

**Desarrollo de un sistema de análisis automático
para la segmentación, clasificación y conteo de
leucocitos en imágenes digitales de frotis de
sangre periférica**



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
DIRECCIÓN DE ESTUDIOS PARA GRADUADOS
PROGRAMA DE MAESTRÍA
INGENIERÍA ELÉCTRICA



**DESARROLLO DE UN SISTEMA DE ANÁLISIS AUTOMÁTICO PARA LA
SEGMENTACIÓN, CLASIFICACIÓN Y CONTEO DE LEUCOCITOS EN
IMÁGENES DIGITALES DE FROTIS DE SANGRE PERIFÉRICA**

AUTOR: Ing. Angel L. Villegas N.

Valencia, julio de 2019



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
DIRECCIÓN DE ESTUDIOS PARA GRADUADOS
PROGRAMA DE MAESTRÍA
INGENIERÍA ELÉCTRICA



**DESARROLLO DE UN SISTEMA DE ANÁLISIS AUTOMÁTICO PARA LA
SEGMENTACIÓN, CLASIFICACIÓN Y CONTEO DE LEUCOCITOS EN
IMÁGENES DIGITALES DE FROTIS DE SANGRE PERIFÉRICA**

AUTOR: Ing. Angel L. Villegas N.
TUTOR: Ing. MSc. Wilmer Sanz.

Valencia, julio de 2019



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
DIRECCIÓN DE ESTUDIOS PARA GRADUADOS
PROGRAMA DE MAESTRÍA
INGENIERÍA ELÉCTRICA



**DESARROLLO DE UN SISTEMA DE ANÁLISIS AUTOMÁTICO PARA LA
SEGMENTACIÓN, CLASIFICACIÓN Y CONTEO DE LEUCOCITOS EN
IMÁGENES DIGITALES DE FROTIS DE SANGRE PERIFÉRICA**

AUTOR: Ing. Angel L. Villegas N.

Trabajo Especial de Grado presentado ante el
Área de Estudios de Postgrado de la
Universidad de Carabobo para optar al título
de Magíster en Ingeniería Eléctrica.

Valencia, julio de 2019



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERIA
DIRECCIÓN DE ESTUDIOS PARA GRADUADOS
SECCIÓN DE GRADO



ACTA DE DISCUSIÓN DE TRABAJO DE GRADO

En atención a lo dispuesto en los Artículos 137, 138 y 139 del Reglamento de Estudios de Postgrado de la Universidad de Carabobo, quienes suscribimos como Jurado designado por el Consejo de Postgrado de la Facultad de Ingeniería, de acuerdo a lo previsto en el Artículo 135 del citado Reglamento, para estudiar el Trabajo de Grado titulado:

“DESARROLLO DE UN SISTEMA DE ANÁLISIS AUTOMÁTICO PARA LA SEGMENTACIÓN, CLASIFICACIÓN Y CONTEO DE LEUCOCITOS EN IMÁGENES DIGITALES DE FROTIS DE SANGRE PERIFÉRICA”

Presentado para optar al grado de **MAGÍSTER EN INGENIERÍA ELÉCTRICA** por el (la) aspirante:

ANGEL LUIS VILLEGAS NOGUERA
V.- 14.436.773

Habiendo examinado el Trabajo presentado, decidimos que el mismo está **APROBADO**.

En Valencia, al primer (01) día del mes de Julio del año dos mil diecinueve.

Prof. Teddy Rojas
C.I.: 4.876.253
Fecha: 01/07/2019
PE: 03/07/2019 /D6.-

Prof. Wilmer Sanz
C.I.: 7.130.476
Fecha: 01/07/2019

Prof. Carlos Jiménez
C.I.: 7.523.966
Fecha: 01/07/2019



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERÍA
DIRECCIÓN DE ESTUDIOS PARA GRADUADOS
PROGRAMA DE MAESTRÍA
INGENIERÍA ELÉCTRICA



VEREDICTO

Nosotros, miembros del jurado designado para la evaluación del Trabajo Especial de Grado
Titulado:

DESARROLLO DE UN SISTEMA DE ANÁLISIS AUTOMÁTICO PARA LA SEGMENTACIÓN, CLASIFICACIÓN Y CONTEO DE LEUCOCITOS EN IMÁGENES DIGITALES DE FROTIS DE SANGRE PERIFÉRICA

Presentado por el Ing. Angel L. Villegas N, para optar al título de **Magíster en Ingeniería Eléctrica**, estimamos que el mismo reúne todos los requisitos para ser considerado como aprobado.

Prof. Wilmer Sanz
C.I: 7.130.496
Presidente

Prof. Teddy Rojas
C.I: 4.876.253
Miembro interno

Prof. Carlos Jiménez
C.I: 7.523.966
Miembro externo

Valencia, julio de 2019



UNIVERSIDAD DE CARABOBO
FACULTAD DE INGENIERIA
DIRECCIÓN DE POSTGRADO



ACTA DE APROBACION DEL PROYECTO DE TRABAJO DE GRADO

Por medio de la presente hacemos constar que el Proyecto de Trabajo de Grado titulado: “**DESARROLLO DE UN SISTEMA DE ANÁLISIS AUTOMÁTICO PARA LA SEGMENTACIÓN, CLASIFICACIÓN Y CONTEO DE LEUCOCITOS EN IMÁGENES DIGITALES DE FROTIS DE SANGRE PERIFÉRICA**” presentado por el ciudadano: **Ángel L. Villegas N.**, alumno regular del Programa de Maestría en **INGENIERIA ELÉCTRICA**, reúne los requisitos exigidos para la inscripción.

El **Prof. Wilmer Sanz**, aceptó la tutoría de este Trabajo, según Constancia anexa.

En Valencia, a los veintisiete días del mes de noviembre del año Dos mil Diecisiete.

Por La Comisión Coordinadora:

Prof. Teddy Rojas
Jefe del Programa

Prof. Wilmer Sanz
Miembro

Profa. Eva Monagas
Miembro

ÍNDICE GENERAL

ÍNDICE GENERAL	iv
ÍNDICE DE FIGURAS	vi
ÍNDICE DE TABLAS	ix
RESUMEN	x
INTRODUCCIÓN	1
CAPÍTULO I	4
EL PROBLEMA	4
1.1. Planteamiento del problema	4
1.2. Objetivos	7
1.2.1. Objetivo general	7
1.2.2. Objetivos específicos	8
1.3. Justificación	8
1.4. Alcance y delimitación	10
CAPÍTULO II	12
MARCO TEÓRICO	12
2.1. ANTECEDENTES	12
2.2. FUNDAMENTOS TEÓRICOS	23
2.2.1. Leucocitos	24
2.2.2. Leucocitos granulados o granulocitos	27
2.2.3. Leucocitos no granulados o agranulocitos	29
2.2.4. Frotis de sangre periférica	31
2.2.5. Tinción del frotis de sangre	34
2.2.6. Conteo diferencial de leucocitos	35
2.2.7. Concepto de imágenes digitales	36
2.2.8. Espacios o modelos de color para imágenes digitales	37
2.2.9. Pre-procesamiento y filtrado de imágenes	46
2.2.10. Generalidades sobre segmentación de imágenes digitales	49
2.2.11. Segmentación utilizando superpíxeles (<i>oversegmentation</i>)	69
2.2.12. Clasificación de imágenes	73
2.2.13. Aprendizaje profundo o <i>Deep Learning</i>	76
2.2.14. Redes neuronales unidireccionales profundas	78
2.2.15. Redes neuronales convolucionales profundas (CNN)	81
2.2.16. Modelos de redes neuronales convolucionales	97
2.2.17. Transferencia de conocimiento (<i>Transfer Learning</i>)	112
CAPÍTULO III	116
3. MARCO METODOLÓGICO	116
3.1. NIVEL DE INVESTIGACIÓN	116
3.2. DISEÑO DE INVESTIGACIÓN	116
3.2.1. Procedimiento metodológico	117
3.3. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS	120
3.4. TÉCNICAS DE PROCESAMIENTO Y ANÁLISIS DE DATOS	121

CAPÍTULO IV	123
RESULTADOS	123
4.1. DESCRIPCIÓN Y PREPARACIÓN DE LOS DATOS EXPERIMENTALES	123
4.1.1. ALL-IDB (Acute Lymphoblastic Leukemia Image Database).....	123
4.1.2. IUMS-IDB.....	124
4.1.3. SMC-IDB	124
4.1.4. LISC: Leukocyte Images for Segmentation and Classification	125
4.1.5. WBC Image Dataset (WBC-ID)	126
4.1.6. Base de datos propia (BDL).....	127
4.1.7. Banco de imágenes de múltiples células.....	128
4.1.8. Preparación y pre-procesamiento de las imágenes.....	128
4.1.9. Aumento de datos y generación de imágenes sintéticas	132
4.2. EXTRACCIÓN DE LEUCOCITOS Y REGIÓN DE INTERÉS.....	134
4.2.1. Ensayos mediante técnicas tradicionales de procesamiento de imágenes.....	134
4.2.2. Propuesta de algoritmo de segmentación combinando Superpíxeles y SVM.....	140
4.2.3. Extracción de leucocitos y región de interés.....	146
4.3. CLASIFICACIÓN DE LOS LEUCOCITOS EN LA SANGRE HUMANA	148
4.3.1. Clasificación usando combinación de CNN y SVM.....	151
4.3.2. Clasificación mediante ajuste de pesos del modelo de red convolucional....	186
4.4. INTERFAZ DE IDENTIFICACIÓN Y CONTEO DE LEUCOCITOS	206
4.5. EVALUACIÓN DEL SISTEMA Y PRUEBAS FINALES.....	207
4.5.1. Funcionamiento general del sistema	208
4.5.2. Resultados con imágenes crudas de múltiples células	213
CONCLUSIONES Y RECOMENDACIONES	215
REFERENCIAS BIBLIOGRÁFICAS	220
APÉNDICES Y ANEXOS	229

ÍNDICE DE FIGURAS

Figura 1. Fotografía de un frotis sanguíneo visto con amplificación 40X.	5
Figura 2. Proceso de formación las células sanguíneas (hematopoyesis).	25
Figura 3. Grupo de leucocitos neutrófilos maduros.	28
Figura 4. Eosinófilo bilobular.....	28
Figura 5. Basófilo con núcleo difícil de distinguir por los grandes gránulos.....	29
Figura 6. Linfocitos: pequeño y grande.....	30
Figura 7. Monocito típico.....	31
Figura 8. Preparación de frotis de sangre periférica en cuña	32
Figura 9. Extensión de gota de sangre para el frotis sanguíneo	33
Figura 10. Representación del espacio de color RGB.....	38
Figura 11. Modelos de color HSL y HSV	39
<i>Figura 12.</i> Representación de los espacios de color HSV, HSL y HSV.	39
Figura 13. Curvas de concordancia de color de los espacios CIE-RGB y CIE- XYZ	42
Figura 14. Representación esférica del espacio LAB.....	44
Figura 15. Frotis sanguíneo visualizado en diferentes espacios de color.....	45
Figura 16. Ejemplo de una neurona artificial	61
Figura 17. Funciones de activación comúnmente usadas en las ANN.....	61
Figura 18. Distribución de las capas de neuronas en una red neuronal típica.....	62
Figura 19. Representación de diferentes topologías de redes neuronales	64
Figura 20. Proyección de los datos en una SVM.	65
Figura 21. Hiperplanos de separación en un espacio bidimensional.....	67
Figura 22. Comparación visual de superpíxeles generados por diferentes métodos.....	70
Figura 23. Reducción del área de búsqueda en torno al centro del superpíxel.	71
Figura 24. Comparación de algoritmos de superpíxeles en frotis sanguíneo	72
Figura 25. Métodos y exactitud alcanzada al clasificar leucocitos	75
Figura 26. Matriz de confusión de un clasificador binario.....	76
<i>Figura 27.</i> Representación de red MLP con dos capas ocultas	79
Figura 28. Arquitectura de la red convolucional LeNet-5.....	82
Figura 29. Arquitectura general de una CNN simple	83
Figura 30. Conectividad local en las redes convolucionales.....	86
Figura 31. Operaciones de sub-muestreo Max-Pool y Avg-Pool.....	88
Figura 32. Representación simplificada de una red neuronal convolucional.....	90
Figura 33. Convolución bidimensional entre una imagen de entrada I y un filtro K.....	91
Figura 34. Volumen de salida en una capa de convolución.	92
Figura 35. Función de activación ReLU y otras usadas en distintas redes neuronales	94
Figura 36. Volumen de salida en una capa de agrupamiento	95
Figura 37. Especificación de capas FC y de salida en una CNN típica.....	96
Figura 38. Arquitectura simplificada red AlexNet	98
Figura 39. Arquitectura simplificada de redes VGG-16 y VGG-19	101
Figura 40. Comparación de requerimientos de memoria y tiempo de inferencia en distintos modelos de redes neuronales convolucionales	102
Figura 41. Comparación de capa de convolución lineal y micro red Mlpconv.....	103

Figura 42. Comparación capas fully connected (FC) y global average pooling GAP	104
Figura 43. Diseño de módulo inception con reducción dimensional	105
Figura 44. Esquema de red GoogleNet.....	106
Figura 45. Comparación del error de entrenamiento y prueba en redes de 20 y 56 capas al clasificar imágenes de la base de datos CIFAR-10	107
Figura 46. Conceptualización del bloque residual.	108
Figura 47. Características de las arquitecturas ResNet utilizadas para clasificar imágenes de la base de datos ImageNet.	109
Figura 48. Red residual de 34 capas de profundidad ResNet-34	109
Figura 49. Estructura Módulo fire usado para la construcción de la red SqueezeNet.....	111
Figura 50. Arquitectura SqueezeNet	111
Figura 51. Procesos de entrenamiento en <i>machine learning</i> y <i>transfer learning</i>	113
Figura 52. Mejoramiento del aprendizaje por medio de transferencia de conocimiento ..	114
Figura 53. Formas de realizar transferencia de aprendizaje en redes convolucionales.	115
Figura 54. Imágenes base datos ALL-IDB.....	124
Figura 55. Ejemplos de imágenes base datos IUMS-IDB.....	124
Figura 56. Ejemplos de imágenes base datos SMC-IDB	125
Figura 57. Ejemplos de imágenes base datos LISC	126
Figura 58. Ejemplos de imágenes base datos Zheng y otros (2018) – Dataset1	126
Figura 59. Ejemplos de imágenes base datos Zheng y otros (2018) – Dataset2	127
Figura 60. Ejemplos de células encontradas en la base de datos de imágenes propias.....	128
Figura 61. Distribución de los tipos de leucocitos en cada conjunto de imágenes	130
Figura 62. Distribución total de leucocitos en datos disponibles	131
Figura 63. Ejemplo de imágenes aumentadas a partir de la imagen superior izquierda ...	133
Figura 64. Métodos empleados para la segmentación de leucocitos.....	134
Figura 65. Segmentación usando métodos del valor umbral global con éxito	136
Figura 66. Segmentación usando métodos del valor umbral global sin éxito	137
Figura 67. Homogenización del color.	138
Figura 68. Imágenes con apariencia artificial posterior a la transferencia de color	138
Figura 69. Mejora del color y segmentación de la imagen resultante	139
Figura 70. Etiquetado de regiones para entrenamiento de SVM para segmentación.....	141
Figura 71. Resultados segmentación y variación de la cantidad de superpíxeles	143
Figura 72. Resultado de la segmentación en imágenes de calidad buena o media	144
Figura 73. Resultados segmentación en imágenes con múltiples células o baja calidad ..	145
Figura 74. Segmentación incorrecta del citoplasma en imágenes de baja calidad.....	145
Figura 75. Interfaz gráfica de usuario desarrollada para la segmentación de leucocitos ..	146
Figura 76. Resultado del proceso de segmentación y extracción final de leucocitos.....	147
Figura 77. Imágenes de leucocitos provenientes de diferentes bases de datos	149
Figura 78. Diagrama representativo estructura CNN. Nagpal y otros (2018, pág. 16).....	152
Figura 79. Exactitud global en el experimento #1.....	164
Figura 80. Exactitud global en el experimento #2.....	164
Figura 81. Exactitud global en el experimento #3.....	164
Figura 82. Exactitud global en el experimento #4.....	165
Figura 83. Comparación exactitud global en experimentos #1 al #4.	165
Figura 84. Gráficos exactitud global y promedio de verdaderos positivos	182

Figura 85. Comparación exactitud y número de operaciones necesarias para evaluar una imagen en diferentes modelos de redes neuronales convolucionales.....	189
Figura 86. División de la red GoogleNet para su entrenamiento	190
Figura 87. Ensayo entrenamiento en red AlexNet.....	193
Figura 88. Ensayos con diferentes valores de tasa de aprendizaje usando red AlexNet...	195
Figura 89. Ensayos con diferentes valores de momento usando red AlexNet	196
Figura 90. Ensayo entrenamiento usando red GoogleNet.....	196
Figura 91. Ensayo entrenamiento usando red ResNet-18	197
Figura 92. Ensayo entrenamiento usando red ResNet-50	197
Figura 93. Ensayo entrenamiento usando red SqueezeNet	198
Figura 94. Entrenamiento de clasificador red AlexNet usando datos aumentados	199
Figura 95. Entrenamiento de clasificador red GoogleNet usando datos aumentados	200
Figura 96. Entrenamiento de clasificador red ResNet-18 usando datos aumentados.....	201
Figura 97. Entrenamiento de clasificador red ResNet-50 usando datos aumentados.....	202
Figura 98. Entrenamiento de clasificador red SqueezeNet usando datos aumentados.....	203
Figura 99. Comparación exactitud global en experimento #7.....	204
Figura 100. Interfaz gráfica del sistema de análisis automático para la segmentación, clasificación y conteo de leucocitos en imágenes digitales de frotis de sangre	207
Figura 101. Resultado clasificación imágenes ALL-IDB1	208
Figura 102. Resultado clasificación imágenes ALL-IDB2	209
Figura 103. Resultado clasificación imágenes IUMS-IDB	209
Figura 104. Resultado clasificación imágenes SMC-IDB.....	210
Figura 105. Resultado clasificación imágenes LISC.....	211
Figura 106. Resultado clasificación imágenes WBC-ID.....	211
Figura 107. Resultado clasificación imágenes WBC-ID (DataSet2)	212
Figura 108. Resultado clasificación imágenes BDL	212
Figura 109. Resultado clasificación imágenes BDL con criterios de rechazo	213
Figura 110. Conteo de leucocitos realizado por el sistema automático propuesto.....	214

ÍNDICE DE TABLAS

Tabla 1. Coloraciones de uso frecuente en la preparación del frotis de sangre periférica ..	34
Tabla 2. Filtros de uso frecuente en procesamiento de imágenes.....	49
Tabla 3. Algoritmos de clasificación usados para la identificación de leucocitos	74
Tabla 4. Método de transferencia de conocimiento y capas a utilizar en la CNN.....	115
Tabla 5. Clasificación de las imágenes extraídas de los datos experimentales	129
Tabla 6. Distribución de leucocitos en los datos disponibles	132
Tabla 7. Distribución de conjuntos de datos para entrenamiento y prueba	133
Tabla 8. Técnicas o algoritmos utilizados en la segmentación de leucocitos.....	135
Tabla 9. Capas utilizadas como extractor de rasgos en diferentes modelos de CNN.....	153
Tabla 10. Resultados Exp. #1. Datos balanceado con descomposición 1 a todos.....	156
Tabla 11. Resultados Exp. #2. Datos balanceado con descomposición 1 a 1.....	158
Tabla 12. Resultados Exp. #3. Datos desbalanceado con descomposición 1 a todos.....	160
Tabla 13. Resultados Exp. #4. Datos desbalanceado con descomposición 1 a 1	162
Tabla 14. Valores máximos y mínimos de exactitud global en experimentos #1 al #4	166
Tabla 15. Comparación promedio de verdaderos positivos vs exactitud en red AlexNet.	167
Tabla 16. Métricas de evaluación para clasificadores del Experimento #1	169
Tabla 17. Métricas de evaluación para clasificadores del Experimento #2.....	171
Tabla 18. Métricas de evaluación para clasificadores del Experimento #3.....	173
Tabla 19. Métricas de evaluación para clasificadores del Experimento #4.....	175
Tabla 20. Comparación de indicadores del Experimento #1	177
Tabla 21. Comparación de indicadores del Experimento #2.....	177
Tabla 22. Comparación de indicadores del Experimento #3	178
Tabla 23. Comparación de indicadores del Experimento #4.....	178
Tabla 24. Comparación de indicadores promedio para redes AlexNet y ResNet-50	178
Tabla 25. Diferencias de los indicadores de funcionamiento del clasificador	180
Tabla 26. Resultados Exp. #5. Clasificador usando datos aumentados.....	181
Tabla 27. Métricas de evaluación para clasificadores del Exp. #5 con esquema OVA	183
Tabla 28. Métricas de evaluación para clasificadores del Exp. #5 con esquema OVO	184
Tabla 29. Comparación de indicadores Exp. #5 con esquema descomposición OVA.....	185
Tabla 30. Comparación de indicadores Exp. #5 con esquema descomposición OVO.....	185
Tabla 31. Capas a ser retenidas, ajustadas o completamente reentrenadas en las CNN....	189
Tabla 32. Exactitud de los clasificadores entrenados con imágenes aumentadas	204
Tabla 33. Métricas de evaluación para clasificadores del Experimento #7.....	205
Tabla 34. Comparación de indicadores del Experimento #7	206
Tabla 35. Criterios empíricos de clasificación rechazar muestras del conjunto BDL.....	213
Tabla 36. Clasificación de las imágenes y conteo diferencial realizado por experto	214

DESARROLLO DE UN SISTEMA DE ANÁLISIS AUTOMÁTICO PARA LA SEGMENTACIÓN, CLASIFICACIÓN Y CONTEO DE LEUCOCITOS EN IMÁGENES DIGITALES DE FROTIS DE SANGRE PERIFÉRICA

AUTOR: Ing. Angel L. Villegas N.

TUTOR: Ing. MSc. Wilmer Sanz.

FECHA: Julio de 2019

RESUMEN

La presente investigación tuvo como objetivo desarrollar un sistema capaz de identificar, clasificar y contar diferentes tipos de leucocitos en imágenes de frotis sanguíneo. El sistema se fundamenta en el uso de las redes neuronales convolucionales profundas (CNN) que representa uno de los paradigmas más influyentes en la visión por computador y reconocimiento de patrones de los últimos años. Para la segmentación y extracción de las células se utilizó una etapa de pre-procesamiento que incorpora el algoritmo lineal simple de agrupamiento iterativo (SLIC) como generador de superpíxeles asociados a las regiones de la imagen. Luego, mediante una máquina de soporte vectorial (SVM) los superpíxeles son clasificados como parte de núcleo, citoplasma, plasma u otros componentes, usando para ello los valores medios de intensidad en los espacios de color RGB, HSV y LAB. Posteriormente, se extrae la región que contiene al núcleo y citoplasma de las células y se utiliza una CNN para determinar el tipo de leucocito. Para el diseño de este clasificador se partió de 9 modelos públicos de redes convolucionales pre-entrenados con imágenes de la base de datos ImageNet. Estos se adaptaron a la tarea de clasificación de leucocitos mediante el concepto de transferencia de conocimiento evaluando las dos técnicas principales aplicables a las CNN: la extracción automática de rasgos y el ajuste fino de los pesos sinápticos. El sistema, implementado en el ambiente de cómputo científico MATLAB, logró al clasificar en forma individual las seis bases de datos disponibles una exactitud promedio de 99.03%. Los resultados obtenidos evidencian la capacidad de las CNN de extraer representaciones adecuadas de los objetos presentes en las imágenes a partir de los datos sin procesar, al permitir hacer una clasificación similar o superior a otras técnicas que usan rasgos cuidadosamente seleccionados por personal especializado. La transferencia de aprendizaje mediante el ajuste fino en los pesos sinápticos demostró ser más efectiva, en términos de exactitud, para adaptar los modelos a nuevos problemas de clasificación. Esto es en parte atribuible a que estos no habían sido entrenados con imágenes de características similares a las empleadas en este proyecto. Usando una combinación de principios, métodos y tecnologías existentes se desarrolló un sistema automático que exhibe una elevada capacidad para localizar y clasificar de forma robusta imágenes provenientes de diferentes bases de datos. Este sistema mostró buena capacidad de generalización al lograr extraer un 95.86% de las células presentes en un banco adicional de imágenes de validación y clasificarlas con un margen de error en torno al 3% respecto al criterio de un observador humano experto. Esto, junto a su capacidad para realizar el conteo diferencial de leucocitos (100 células) en un tiempo medio aproximado de 1 minuto, le convierte en una alternativa competitiva frente a otras tecnologías del área.

Palabras clave: redes neuronales convolucionales, conteo de leucocitos, clasificación de imágenes

INTRODUCCIÓN

La clasificación y conteo de leucocitos en frotis de sangre periférica es un examen de rutina que aún hoy en día se realiza de forma manual en muchos los laboratorios clínicos. Su correcta ejecución precisa del tiempo valioso de profesionales altamente calificados y su naturaleza repetitiva puede resultar tediosa e inclusive perjudicial, en el largo plazo, para la salud visual del personal que lo realiza. La automatización de esta tarea se ha convertido en un factor importante para los laboratorios modernos por lo que han surgido diversas tecnologías para este fin. Algunas de estas tecnologías requieren de cuantiosas inversiones para su incorporación y operación continua por lo cual su ámbito queda limitado únicamente a grandes centros asistenciales.

Sin embargo, los avances recientes en áreas como la computación e inteligencia artificial han contribuido a que el uso de técnicas modernas de procesamiento de imágenes digitales se convierta en una opción plausible para el desarrollo de este tipo de sistemas, capaz de evolucionar y mejorar a un ritmo mayor que otras técnicas.

Uno de esos avances notables lo constituyen las redes neuronales convolucionales profundas, las cuales, pese a estar basadas en ideas estudiadas antes del inicio de este siglo, han vivido un re-descubrimiento destacado en la última década, producto entre otras cosas del mejoramiento del hardware computacional, especialmente de las unidades de procesamiento gráfico (GPU por sus siglas en inglés). Esto, junto a la aparición de masivas bases de datos públicas de imágenes y al surgimiento de paquetes de software adecuados para manejarlas, ha devenido en la proyección de estas ideas y en el renacimiento de los conceptos del aprendizaje profundo como una poderosa herramienta para múltiples aplicaciones.

Estas redes neuronales convolucionales son arquitecturas de aprendizaje profundo inspiradas en la estructura de corteza visual de los mamíferos y combinan determinadas características que les han permitido posicionarse como uno de los algoritmos más exitosos al realizar tareas de clasificación e identificación de objetos en imágenes a color. Una de sus características

principales es que tienen la capacidad de manejar datos sin procesar por lo que no es necesario realizar una cuantiosa ingeniería de características previas ya que pueden extraer representaciones a partir de los propios datos.

Siendo que las características particulares del análisis manual del frotis de sangre periférica han hecho que exista interés por desarrollar métodos y tecnologías con el objeto de automatizar esta labor, y que, como resultado de esta automatización, se podrían reducir errores inherentes al factor humano, aumentar la productividad de los laboratorios clínicos sin requerir mayores inversiones, mejorar la velocidad de realización del examen brindando una mejor asistencia al paciente, y ante la existencia de herramientas computacionales aplicables para tal fin como las redes convolucionales, este trabajo propuso el desarrollo de un sistema automático capaz de localizar, clasificar y contar los leucocitos presentes en imágenes digitales como alternativa de apoyo diagnóstico el área hematológica.

El trabajo se encuentra estructurado en cuatro capítulos. En el Capítulo I, se describe el problema en estudio que se desea resolver, establecen los objetivos propuestos, presentan las razones que justificaron la realización del proyecto y define el alcance pretendido con su ejecución. El Capítulo II abarca la revisión de un amplio conjunto de antecedentes junto a la fundamentación teórica sobre la que se apoyó el desarrollo del proyecto. Esta comienza con una descripción de conceptos básicos de hematología, necesarios para comprender los tipos de los leucocitos y sus rasgos típicos, así como, el proceso requerido para la elaboración del frotis de sangre periférica, la naturaleza y características de las imágenes en estudio. Seguidamente se incluyen fundamentos de procesamiento de señales, junto a algunos de los algoritmos empleados o evaluados durante la investigación. Finalmente, se explican las redes neuronales convolucionales profundas, las cuales constituyen la herramienta fundamental empleada para alcanzar los objetivos propuestos. En el Capítulo III se presenta la metodología empleada, señalando el tipo y diseño de la investigación, procedimiento metodológico descrito mediante fases con sus respectivas actividades principales, técnicas e instrumentos de recolección, procesamiento y análisis de los datos. Se incluye una descripción de los equipos computacionales utilizados para llevar a cabo la investigación.

Por su parte, en el Capítulo IV se presentan y discuten los resultados obtenidos. Para ello, se hace uso de numerosas tablas y graficas que muestran e ilustran el comportamiento obtenido durante el entrenamiento y al clasificar los datos tanto de validación como de prueba mediante numerosos indicadores estadísticos. En esta sección se incluye además una descripción detallada de los experimentos realizados, sus resultados e implicaciones. También se presentan las características de la interfaz gráfica de usuario creada para el sistema, indicando la funcionalidad de sus controles y forma de uso. Luego, se compara y analiza el funcionamiento de la propuesta al clasificar todas las bases de datos disponibles contra los resultados documentados por otros investigadores. Posteriormente, se contrastan los resultados emitidos por el sistema y un experto médico en una base de datos adicional de prueba con imágenes variadas, no usadas durante el entrenamiento del clasificador. Finalmente, se presentan las conclusiones obtenidas, recomendaciones para trabajos de investigación futuros y el listado de referencias bibliográficas consultadas. Adicionalmente se incluyen en forma de Apéndice el código fuente de los scripts generados y un Anexo con resultados adicionales no incluidos en el cuerpo principal del trabajo.

CAPÍTULO I

EL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

El frotis de sangre periférica es un examen para el diagnóstico hematológico ampliamente utilizado, el cual consiste obtener una muestra de sangre de las venas del paciente, extenderla sobre una lámina de vidrio y teñirla con el fin de observar imágenes microscópicas en las cuales se pueden identificar típicamente: eritrocitos (glóbulos rojos), plaquetas y leucocitos (glóbulos blancos) (Ramoser, Laurain, Bischof, & Ecker, 2005). Estos últimos constituyen células extraordinariamente versátiles cuyo principal propósito es defender al organismo contra bacterias, virus, hongos y otras sustancias extrañas (Ciesla, 2014). Las muestras normales de sangre periférica contienen típicamente seis tipos diferentes de leucocitos: neutrófilos segmentados, linfocitos, monocitos, eosinófilos, basófilos y neutrófilos banda, mientras que en caso de algunas enfermedades puede observarse la presencia de otras células como: metamielocitos, mielocitos, promielocitos, mieloblastos y eritroblastos (Ramoser y otros, 2005).

El conteo y clasificación de los leucocitos se conoce con el nombre de conteo diferencial de glóbulos blancos (Ciesla, 2014), y es una tarea que, aún hoy en día, es llevada a cabo en muchos casos de forma manual por profesionales del bioanálisis o médicos hematólogos. Estos profesionales requieren de años de estudio y experiencia para realizar satisfactoriamente el proceso ya que la confiabilidad de los resultados obtenidos es dependiente, en gran medida, de las habilidades del personal que realiza el estudio (Martínez-Castro, Reyes-Cadena, & Felipe-Riverón, 2014). Al respecto, Ramirez-Cortes y otros (2010) señalan que este método tradicional de análisis se ve afectado por tres tipos de errores: estadístico, distribucional debido a la localización desigual de las células en el frotis y la identificación errada de células relacionada con la interpretación subjetiva del examinador.

Adicionalmente, el procedimiento puede resultar tedioso debido a su naturaleza repetitiva e incluso a largo plazo llegar a afectar la salud visual del practicante (Martínez-Castro y otros, 2014), pues implica recorrer imágenes microscópicas hasta lograr identificar y contar 100 glóbulos blancos.

En la Figura , se puede observar una fotografía de una imagen típica obtenida al realizar un frotis de sangre periférico. En ella se identifica que la gran mayoría de la células presentes en la muestra, corresponden con eritrocitos y plaquetas (se observan en color gris). La tinción aplicada permite resaltar el núcleo de los leucocitos los cuales se destacan con un color morado que permite facilitar su identificación por parte del personal médico. En la parte derecha se ha resaltado en un círculo rojo aquellas células que corresponden a los leucocitos existentes en la sección de la imagen

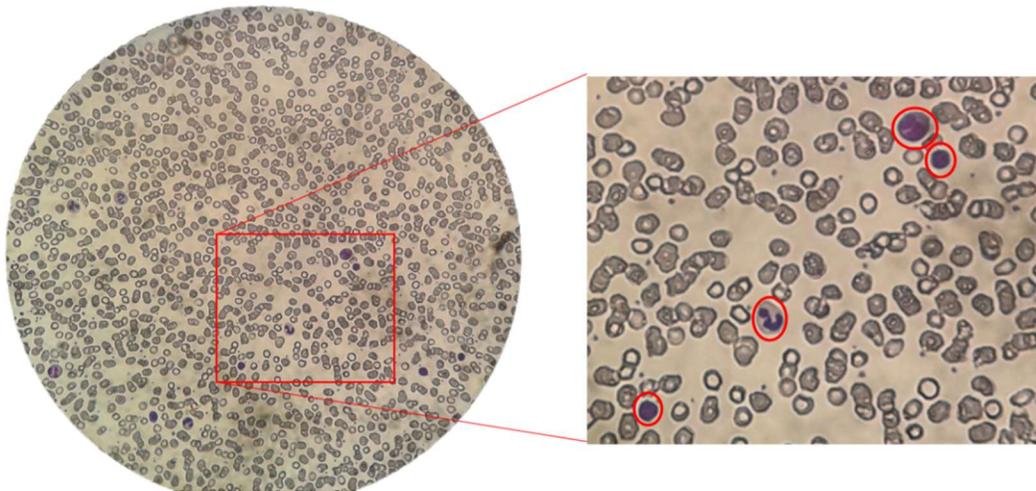


Figura . Fotografía de un frotis sanguíneo visto con amplificación 40X. **Nota.** En el detalle de la derecha se han resaltado los leucocitos presentes en un segmento de la muestra.

Por otra parte, el tiempo requerido para realizar el análisis depende del profesional que los ejecuta, pudiendo estar comprendido entre 3 y 15 minutos por muestra, según indican Kim et al., (2014) y Rezatofghi & Soltanian-Zadeh (2011), respectivamente. Considerando que un laboratorio puede, dependiendo de su volumen de trabajo, recibir entre 50 a 350 muestras diarias para ser procesadas, esto constituye en algunos casos una importante inversión de horas/especialista para lograr la atención de los pacientes.

Es por esto que la identificación y conteo automático de leucocitos en la sangre, se ha convertido en un factor de importancia para los laboratorios modernos, propiciando la aparición de equipos comerciales para este fin.

Existen equipos comerciales que permiten hacer el conteo automatizado de células típicamente basados en tres principios: impedancia eléctrica de Coulter, análisis de imágenes digitales y citometría de flujo (Ruiz, 2016). Esta última ha sido la técnica dominante, empleada por los laboratorios médicos para examinar partículas microscópicas como los elementos sanguíneos durante los pasados 25 años según comenta Habibzadeh y otros (2012). Estos equipos son usados especialmente en laboratorios que procesan altos volúmenes de muestras, debido a su alto costo de adquisición, ya que requieren de un hardware y software especializado para su funcionamiento, diferente a los tradicionales equipos manuales utilizados por muchos profesionales del área. Adicionalmente, en algunos casos requieren del uso de químicos costosos para su operación (Rodrigues, Ferreira, & Monteiro, 2008).

La imagenología médica es un campo relativamente nuevo de la ciencia, que refiere al uso de técnicas y procesos utilizados para crear imágenes del cuerpo humano con propósitos clínicos. Sus usos actuales abarcan diferentes áreas por ejemplo: segmentación de imágenes de ultrasonido, identificación de estructuras cerebrales, localización de tumores y otras patologías o diagnóstico de enfermedades como la leucemia (Nagabhushana & Rajeshwari, 2014).

Los impresionantes avances en esta área, junto con las matemáticas y la computación, se han aprovechado para la construcción de equipos destinados al análisis de células sanguíneas basados en técnicas de procesamiento de imágenes digitales e inteligencia artificial. Estos han incrementado su rendimiento muy rápidamente durante los últimos años, ofreciendo ciertas ventajas potenciales como el acceso a parámetros que no son fácilmente medidos mediante citometría de flujo (Lisicki, 2014) y la posibilidad de un

mejoramiento constante producto del crecimiento de las disciplinas sobre las que se fundamentan.

Algunos de los principales avances en el campo de la visión por computador de la última década, han sido debido al resurgimiento de las redes neuronales convolucionales profundas, las cuales han alcanzado resultados sorprendentes al realizar tareas de clasificación e identificación de objetos en imágenes a color y constituyen un paradigma prometedor para la implementación de sistemas de procesamiento de imágenes y visión computacional, siendo capaces de alcanzar o exceder la exactitud lograda mediante otras tecnologías.

Considerando lo anterior y en vista de sus beneficios potenciales, en este trabajo se propuso desarrollar un sistema automático capaz de identificar, clasificar y contar leucocitos basado en el análisis de imágenes digitales para ser utilizado como herramienta de apoyo diagnóstico en el área hematológica.

Dicho sistema es capaz de reducir el tiempo requerido por el especialista para hacer el conteo diferencial de leucocitos y minimizar los errores producto del factor humano en el procesamiento de las muestras. Para su ejecución no precisa la adquisición de equipos computacionales especializados, siendo factible y accesible su uso en todo tipo de laboratorios.

1.2. OBJETIVOS

1.2.1. Objetivo General

Desarrollar un sistema de análisis automático para la segmentación, clasificación y conteo de leucocitos en imágenes digitales de frotis de sangre periférica como herramienta de apoyo al diagnóstico en el área hematológica.

1.2.2. Objetivos Específicos

- Describir las características morfológicas de las principales células presentes en el frotis sanguíneo que permiten su identificación visual.
- Crear un banco de imágenes digitales con muestras catalogadas por expertos para fines experimentales.
- Analizar diferentes técnicas de segmentación de imágenes digitales aplicables a la identificación de células sanguíneas para seleccionar el algoritmo a ser implementado, considerando como criterios la exactitud del proceso y costo computacional requerido para su ejecución.
- Seleccionar un conjunto de rasgos característicos del núcleo y/o citoplasma de los leucocitos que permitan su identificación mediante un sistema computacional.
- Implementar un algoritmo de clasificación que permita identificar al menos 3 tipos de leucocitos (neutrófilos segmentados, linfocitos y monocitos) presentes en la sangre humana.
- Evaluar el funcionamiento del sistema mediante el cálculo de métricas de validación como la sensibilidad y sensibilidad utilizando muestras de individuos sanos.

1.3. JUSTIFICACIÓN

El uso de imágenes digitales en el diagnóstico médico constituye un área de investigación muy activa en la actualidad, de la cual se han beneficiado numerosos campos, siendo la hematología uno de ellos. Las características particulares del análisis manual de muestras de sangre periférica, al ser una tarea tediosa, lenta y, estar sujeta a variaciones producto del criterio del observador, han hecho que exista interés en la comunidad científica por desarrollar métodos y algoritmos computacionales a fin de automatizar esta labor y reducir errores inherentes al factor humano.

El proyecto abarca la resolución de dos problemas interesantes en el desarrollo de aplicaciones basadas en procesamiento de imágenes, como lo son: (a) la segmentación de un conjunto de objetos en el contexto de imágenes ruidosas, compuestas por numerosos elementos, las cuales, no son necesariamente de alta calidad y poseen condiciones de iluminación cambiantes, y (b) la clasificación o reconocimiento de objetos con características morfológicas distintivas mediante el uso de técnicas vanguardistas con alto impacto en el campo de visión por computador.

Dichas técnicas fueron aplicadas en la automatización de un examen de rutina, el cual es realizado por todos los laboratorios clínicos, como lo es el conteo diferencial de leucocitos. Con esto, se logró verificar que mediante el uso de algoritmos de aprendizaje automático es posible lograr una clasificación de imágenes médicas similar a la realizada por un experto con años de entrenamiento en un tiempo reducido y con pocos errores.

La ejecución del trabajo de investigación permitió definir una metodología para la creación de sistemas identificación y clasificación de células sanguíneas basado en imágenes, utilizando para ello poderosas herramientas de la inteligencia artificial como son las redes neuronales convolucionales de aprendizaje profundo (CNN) y máquinas de soporte vectorial. Su realización aporta resultados experimentales que demuestren como la automatización de ésta tarea reduce el tiempo requerido para la obtención del conteo, permitiendo así al profesional de la salud enfocarse más en el análisis de los resultados y brindar un mejor diagnóstico al paciente.

Los resultados y modelos computacionales obtenidos pueden ser escalados para la identificación de otros tipos de células e incluso para la detección de leucocitos inmaduros u otras alteraciones que se relacionan como enfermedades como la leucemia y otros padecimientos, constituyendo así un aporte interesante para diagnóstico clínico en general. Esta capacidad de permitir el reconocimiento de nuevos tipos de células fue

comprobado durante el desarrollo de la investigación, al expandir el alcance original de la propuesta e incluir finalmente los 4 tipos principales de leucocitos presentes en la sangre.

El trabajo se encuentra adscrito a la línea de investigación “Segmentación y Procesamiento de Imágenes” del Centro Procesamiento de Imágenes de la Facultad de Ingeniería de la Universidad de Carabobo, enmarcado dentro de las áreas prioritarias “Tecnología” y “Salud” de esta institución.

1.4. ALCANCE Y DELIMITACIÓN

Al observar el frotis de sangre periférica en un microscopio es posible identificar diferentes tipos de células dependiendo del estado de salud y enfermedades presentes en el paciente. Debido a la importancia que tiene el conteo e identificación de los glóbulos blanco o leucocitos para el diagnóstico clínico de muchas patologías así como la evaluación de las condiciones generales del paciente, este trabajo se circunscribe únicamente a este tipo de células.

Para esta etapa de la investigación se utilizaron únicamente imágenes de células provenientes de individuos sanos, descartando toda muestra que sea identificada como patológica. Las imágenes, obtenidas mediante la asociación con un laboratorio clínico y a través de bases de datos publicadas por grupos de investigación del área, fueron clasificadas por expertos médicos con más de 15 años de experiencia en la realización de exámenes sanguíneos.

Dentro de los diferentes tipos de leucocitos presentes en una muestra no patológica, se destacan cuatro tipos fundamentales: neutrófilos segmentados, linfocitos, monocitos y eosinófilos, los cuales pueden llegar a constituir cerca del 99% del total general. Por esta razón el sistema propuesto se centró en la identificación de estos tipos de células. La identificación de basófilos fue descartada toda vez que no fue posible conseguir una

cantidad apropiada de imágenes para entrenar un sistema capaz de reconocer sus características de forma consistente y confiable.

Para la implementación de los algoritmos se utilizó el programa MATLAB, ampliamente aceptado en la comunidad científica debido a la gran cantidad de herramientas en el área de tratamiento de imágenes y aprendizaje automático que ofrece, lo cual permitió reducir el tiempo requerido para la codificación y prueba del sistema propuesto.

CAPÍTULO II

MARCO TEÓRICO

En esta sección se describen algunos aspectos teóricos que servirán de sustento a la investigación. Se describen los antecedentes del trabajo y presentan una serie de conceptos sobre hematología y procesamiento digital de imágenes.

2.1. ANTECEDENTES

El uso de imágenes digitales como medio de apoyo al diagnóstico clínico hematológico es un área activa de investigación desde hace más de diez años, según puede evidenciarse en la existencia de diversos trabajos publicados recientemente en numerosas revistas y eventos de difusión científica en todo el mundo. Algunos de estos trabajos previos fueron consultados como punto de partida al para la realización de la presente investigación y se describen a continuación.

En el trabajo de Ramoser y otros (2005), se expone que la detección y clasificación de los leucocitos en el extendido de sangre periférica es una tarea de rutina en el diagnóstico médico, y presenta un proceso automatizado para la segmentación de leucocitos el cual es robusto ante las variaciones de apariencia de las células y calidad de las imágenes utilizadas. El enfoque utilizado para la clasificación de los leucocitos consistió de dos partes: (a) la segmentación del leucocito, citoplasma y núcleo, y (b) la clasificación de las células ya segmentadas mediante el uso de un conjunto de características estadísticas (18 de color y 8 morfológicas) que describen las propiedades del citoplasma y el núcleo. En el documento se señala que “los pasos más cruciales son la correcta segmentación y selección de rasgos discriminatorios mientras que la selección del método de clasificación parece tener poca influencia en la exactitud de clasificación obtenible”. Se propone un método adaptativo para la segmentación de leucocitos compuesto de cuatro etapas: (a) detección del fondo de la imagen, regiones de potenciales núcleos celulares y otras regiones (ejemplo, eritrocitos o citoplasma),

(b) supresión del color del fondo, (c) segmentación probabilística del leucocito, y (d) segmentación del núcleo/citoplasma de cada célula. En ambos procesos de segmentación se utilizó el algoritmo de agrupamiento k-medias. Para la etapa de clasificación fue utilizada una máquina de soporte vectorial (SVM) empleando un vector de rasgos de 26 elementos. Los autores utilizaron un conjunto de 1166 imágenes de 558x750 píxeles, capturadas de un microscopio con amplificación de 63X, en las cuales se encontraban muestras de 13 clases diferentes de células, logrando un 95% de éxito en el proceso de segmentación. Usando únicamente las imágenes segmentadas correctamente, los autores lograron índices de aciertos en la clasificación desde 64.5% hasta 99.5%, dependiendo del tipo de célula y de tipo de entrenamiento (con o sin umbral de rechazo). Las células clasificadas con mayor exactitud fueron: eosinófilos (98,0% a 99,5%), linfocitos (92.8% a 97.8%), eritroblastos (93.9% a 96.7%), monocitos (88.4% a 96.1%) y basófilos (87.5% a 93.8%). Los resultados obtenidos por estos investigadores fueron empleados como referencia de comparación para la validación de los resultados de la presente investigación.

El trabajo Ruggero, Piuri y Scotti (2011) surgió como iniciativa para solventar uno de los problemas encontrados al momento de comparar resultados de diferentes estudios del uso de procesamiento de imágenes aplicado al frotis sanguíneo: la falta de datos públicos comunes para evaluar el funcionamiento de los algoritmos de segmentación y clasificación. El proyecto propuso la creación de un conjunto de datos de muestras sanguíneas, específicamente diseñado para la evaluación y comparación dichos algoritmos. Para cada imagen se incluye una clasificación realizada por oncólogos expertos que indica si la fotografía pertenece a un paciente con leucemia (contiene células inmaduras denominadas blastos) o a un individuo sano. El objeto de esta iniciativa fue crear una herramienta para los investigadores del área de procesamiento de imágenes y reconocimiento de patrones para estimular el desarrollo de nuevos estudios en este importante campo de investigación, especialmente para apoyar el desarrollo de técnicas para la rápida detección de la leucemia linfoblástica aguda

(ALL). Representó un valioso antecedente para la presente investigación ya que ofreció un conjunto de datos experimentales que permitieron la comparación de los resultados obtenidos con otros trabajos reportados en la literatura del área.

Putzu y Di Ruberto (2013) señalan que el uso de técnicas de procesamiento de imagen ha crecido rápidamente en los últimos años y que las mismas, ayudan a realizar el conteo de células en la sangre humana y simultáneamente proveen información de la morfología celular. Se resalta que el conteo y análisis de las células sanguíneas permite la evaluación y diagnóstico de muchas enfermedades, destacando cómo el análisis de los glóbulos blancos es un punto de especial interés para los hematólogos, y que el análisis morfológico de estos es realizado de forma manual por operadores entrenados, trayendo consigo numerosas desventajas como lo son la lentitud del análisis y exactitud no estándar. En el trabajo se comenta que existen equipos comerciales que permiten contar el número de las diferentes tipos de células presentes en el extendido periférico haciendo uso de técnicas de citometría de flujo para medir características físicas y/o químicas de las células al pasar por detectores ópticos o medidores de impedancia a fin de lograr su identificación. En este sentido, los autores afirman que pese a ser los resultados de la cuantificación muy precisos, las anomalías morfológicas de las células no son detectadas por la máquina haciendo entonces necesario un análisis subsecuente de la muestra de sangre bajo el microscopio. Los leucocitos son un conjunto de células diferentes entre sí, las cuales exhiben variaciones notables en su forma, dimensiones y bordes siendo estas diferencias las que pueden representar complicaciones para su procesamiento y análisis. El artículo presenta un método automatizado de identificación de glóbulos blancos (sin realizar una clasificación de los mismos) a partir de imágenes microscópicas en el cual las etapas de pre-procesamiento y segmentación no se encuentran separadas sino que se utilizan métodos de pre-procesamiento insertos entre las diferentes etapas de segmentación. La propuesta consta de cinco etapas: (1) Identificación de leucocitos y remoción del fondo de la imagen, (2) Identificación de grupos de leucocitos adyacentes, (3) Separación de

leucocitos adyacentes, (4) Eliminación de componentes anormales y células en los bordes de la imagen, y (5) Segmentación de núcleo y citoplasma. Este proyecto fue desarrollado completamente utilizando el ambiente MATLAB, en especial el Toolbox de procesamiento de imágenes. Para las pruebas se utilizó una muestra de 33 imágenes de 1712x1368 píxeles extraídas de una base de datos de acceso público denominada ALL-IDB1, la cual consiste de 108 muestras de sangre y puede ser obtenida en la dirección (<https://homes.di.unimi.it/scotti/all/>). El uso de esta base de datos constituyó un buen aporte para la presente investigación, ya que permitió comparar sus resultados bajo un marco referencial común al utilizar imágenes accesibles para diferentes grupos, como es la intención del autor de la presente investigación. Sus resultados señalan que fue posible identificar un total de 245 leucocitos de un total de 267 con una exactitud promedio de 92%. De este trabajo también se tomó la idea de intercalar etapas de procesamiento y segmentación.

Sarrafzadeh, Rabbani, Talebi y Banaem (2014) en realizaron la detección y clasificación de 4 tipos de leucocitos, basándose en el análisis de características de color, geométricas y de textura consideradas por los especialistas médicos para diferenciar estas células. A diferencia de otros trabajos la segmentación no se realizó utilizando el espacio de color RGB o imágenes en escala de grises sino que se usó una transformación al espacio $L^*a^*b^*$. Las componentes de color a y b fueron usadas como rasgos para aplicar el algoritmo de agrupamiento C-medios difuso (FCM), posteriormente operaciones morfológicas fueron aplicadas para lograr la segmentación tanto del núcleo como del citoplasma. El vector de rasgos usado constó de 25 elementos, distribuidos de la siguiente forma: 6 rasgos de color extraídos tanto del núcleo como del citoplasma, 6 características geométricas únicamente del núcleo, 6 valores estadísticos del citoplasma y 7 momentos invariantes del citoplasma. Para la asignación de clases, se entrenó de manera independiente 4 clasificadores basados en SVM con arquitectura uno a uno. Se emplearon imágenes microscópicas de 10 pacientes las cuales incluían un total de 149 leucocitos alcanzando una tasa de acierto

del 93% en el peor de los casos (identificación de monocitos) hasta un 99.9% para el caso más favorable (eosinófilos), esta base de datos denominada IUMS-IDB ha sido publicada de libre acceso y fue utilizada en el presente trabajo. Adicionalmente, se tomó la idea de emplear componentes resultantes de la transformación al espacio $L^*a^*b^*$ como parte los rasgos usados por el algoritmo de segmentación.

Bhavnani y otros (2016) realizaron la segmentación y conteo de glóbulos blancos y rojos a partir de imágenes microscópicas de muestras sanguíneas. No se realizó una clasificación de los leucocitos sino la determinación del número total de células pertenecientes a esta categoría. Para el procesamiento, estos autores optaron por extraer únicamente la componente verde de las imágenes RGB y realizar un ajuste del contraste, en lugar del enfoque usual de conversión a escala de grises de la imagen completa. La segmentación se realizó utilizando el algoritmo de umbralización de Otsu y operaciones morfológicas, mientras que el conteo de los glóbulos blancos se realizó en base a las características geométricas de la célula especialmente el área. Para el conteo de eritrocitos se examinó el uso de dos métodos basados en las transformadas Watershed y Hough. Los algoritmos desarrollados en MATLAB fueron probados usando la base de datos ALL-IDB1 logrando una exactitud mayor al 90%. Los investigadores concluyeron que el uso de la transformada Watershed era más adecuado en el caso de anomalías morfológicas de las células que la transformada circular de Hough. Este trabajo fue empleado durante la validación de resultados como uno de los puntos de referencia al comparar la exactitud obtenida sobre la base de datos ALL-IDB1.

El artículo de Reinhard, Ashikhmin, Gooch, & Shirley (2001) describe un método de transferencia de color que emplea un análisis estadístico simple y permite imponer las características de color de una imagen sobre otra. Se basa en el emparejamiento de la media y desviación estándar de cada canal de color en las imágenes fuente y destino. Por medio de éste procedimiento se logra realizar una corrección de color, de forma

que la imagen destino adquiere las características de otra imagen usada como fuente. Para su implementación, las imágenes RGB deben ser convertidas al espacio de color LAB ya que, a diferencia del espacio RGB, existe poca correlación entre los ejes en dicho espacio, lo cual permite aplicar diferentes operaciones en los canales de color de forma relativamente independiente, sin que ocurran artefactos inter-canal indeseados. Ya que uno de los problemas mencionados en la literatura de forma constante al implementar sistemas de identificación de glóbulos blancos son las variaciones en la iluminación y colores de las imágenes empleadas, en el presente proyecto de investigación se evaluó la utilización de un esquema similar.

Zhang y otros (2014) exponen que la segmentación de glóbulos blancos es un reto debido a la diversidad morfológica de estas células y la complejidad e incertidumbre del fondo de las imágenes de frotis sanguíneo y propone un método novedoso para segmentación del citoplasma y núcleo de los linfocitos para ser usado en aplicaciones de citología. El método incluye un paso de ajuste del color previo a la segmentación derivado del algoritmo propuesto por Reinhard y otros (2001), este ajuste permitió mejorar la exactitud de la segmentación de acuerdo a las conclusiones del trabajo. Se utilizó el agrupamiento k-medias tres veces aplicado en diferentes espacios de color (RGB, HSI y CMYK) con el fin de segmentar de forma exitosa el núcleo y el citoplasma. Para la verificación del funcionamiento del sistema se utilizó una base de datos compuesta por 300 imágenes en las que se encuentran 690 leucocitos, alcanzando un 95.1% y 91.3% de precisión en la segmentación del núcleo y citoplasma, respectivamente. Las imágenes de la base de dato presentaron algunas variaciones en las condiciones de iluminación, el grosor del frotis y tiempo de secado ya que fueron adquiridas por diferentes hematólogos sin embargo el algoritmo fue lo suficientemente robusto para arrojar buenos resultados. La mayoría de las investigaciones en la segmentación de leucocitos se enfocan en los algoritmos usados y pocos prestan atención al ajuste del color, a pesar de usar métodos sensitivos a estas variaciones; esto aunado a que las imágenes capturadas en el ambiente clínico no siempre tienen las

condiciones adecuadas para garantizar una buena consistencia del color, limita su aplicabilidad en la práctica y condujo a los autores a concluir que el ajuste de color es un paso necesario. De este trabajo se extrajo la idea de utilizar simultáneamente la información proveniente de la imagen representada en distintos espacios de color como medio para aumentar el éxito.

Mostafa y otros (2012) presentan una técnica para la segmentación automática del núcleo de glóbulos blancos basada en el mejoramiento del contraste de la imagen en escala de grises y filtrado, posteriormente se aplica el algoritmo de Otsu y un conjunto de operaciones morfológicas. Para la eliminación de objetos pequeños que presentan niveles de grises similares al núcleo de los leucocitos, se incorporó un criterio de tamaño mínimo el cual descarta aquellos objetos que por su reducido tamaño no pueden corresponder al núcleo de un glóbulo blanco. Las pruebas fueron llevadas a cabo en un conjunto de 365 imágenes con un rendimiento promedio del 79.7% de aciertos que se eleva hasta un 86.3% al segmentar monocitos. La base de datos de este proyecto y el propio código fuente desarrollado en MATLAB fueron colocados por los autores para su descarga pública en el sitio de Web de MathWorks con el fin de ser usado para comparación y reproducción. A partir de la exactitud lograda, los autores analizan que el algoritmo empleado tiene limitaciones y que las razones principales de la inexactitud se encuentran en las condiciones no homogéneas de iluminación y teñido de las muestras las cuales afectan el contraste de las imágenes. Estos problemas son ampliamente comentados por otros autores. El acceso público a las imágenes de este proyecto permitió comparar con los resultados obtenidos en el presente trabajo, además de formar parte de los datos usados para el entrenamiento del sistema de clasificación propuesto.

Nagabhushana y Rajeshwari (2014) se enfocan el uso de técnicas de procesamiento de imágenes clásicas en la clasificación de células sanguíneas para el diagnóstico de enfermedades, tratando específicamente el caso de la leucemia linfoblástica aguda

(ALL por sus siglas en inglés) debido a que esta enfermedad puede ser detectada mediante el análisis de los glóbulos blancos. En el trabajo se utilizaron técnicas tradicionales de umbralización (algoritmo de Otsu), detección de borde mediante método de Canny junto con operaciones morfológicas para la segmentación del núcleo y un clasificador basado en máquinas de soporte vectorial para determinar si pertenecía a un leucocito normal o inmaduro (mieloblasto). Como rasgos característicos para el clasificador se utilizó: área, perímetro, redondez, solidez, excentricidad, elongación y factor de forma resaltando que el clasificador logró un 88% de exactitud en su labor utilizando un vector de rasgos bastante reducido en comparación de otros trabajos. Como datos de entrada al sistema los investigadores utilizaron 144 imágenes de la base de datos ALL-IDB1, ya descrita anteriormente (Ruggero, Piuri, & Scotti, 2011). Entre sus conclusiones los autores destacan que los resultados obtenidos les alientan a seguir trabajando en mejoras como la segmentación independiente de la tinte usada en el frotis y clasificación de tipos de leucemia. Este trabajo se utilizó como referencia en la comparación de resultados. También se tomó la idea de conformar un vector de rasgos característicos de dimensión muy reducida, para buscar la mayor eficiencia computacional.

Rodrigues y otros (2008) centraron su esfuerzo en comparar dos diferentes técnicas para la segmentación y clasificación de leucocitos, una basada en aprendizaje supervisado y otra en aprendizaje no supervisado en la búsqueda de lograr un balance entre la especialización y generalización de la red neuronal. Para ello primero analizaron el uso de un sistema que emplea exclusivamente redes neuronales entrenadas mediante el algoritmo de retro-propagación (BPNN). Este contiene una etapa de segmentación en la cual se entrena a la red para determinar si el píxel central de una sub-imagen pertenece a un glóbulo blanco; a fin de lograr esto fue creado, de forma manual con ayuda de un experto, un conjunto de patrones de salida que indican la pertenencia de cada píxel o no a un leucocito. La red se entrena recorriendo la imagen con una ventana de 25x25 píxeles, siendo finalmente capaz de segmentar y aislar las

regiones pertenecientes a las células en estudio. Para la clasificación existe otra red neuronal, la cual utiliza como parámetros de entrada un conjunto de rasgos de forma, textura y geométricos (un total de 21) extraídos del núcleo y citoplasma (estos son separados mediante el análisis heurístico del histograma). Usando este enfoque los autores clasificaron en forma exitosa un 81% de los 5 tipos principales de leucocitos, siendo los linfocitos con 88.7% el grupo mejor identificado y los monocitos con 73% el grupo de menor acierto. Sin embargo, los autores señalaron que el rendimiento de este método era muy sensible al conjunto de entrenamiento usado y que esa tasa de aciertos se debía al uso de imágenes similares en brillo, color y posición estadística de las células por lo que no se podía garantizar la estabilidad al utilizar otras imágenes. Se contrasta entonces el rendimiento al usar un sistema híbrido que emplea una red neuronal de pulso acoplado (PCNN) junto a una máquina de soporte vectorial (SVM). Aquí el elemento distintivo es el uso de una PCNN, la cual utiliza un proceso no supervisado y es capaz de extraer información visual esencial a partir de la propia imagen. Con este segundo enfoque los investigadores buscaban mejorar la capacidad de generalización por considerar que ese era la principal debilidad del primer sistema. Los resultados presentados en el artículo indican que efectivamente se logró incrementar los aciertos en la clasificación alcanzando valores entre 80% y 91.9% según el tipo de leucocito. Sin embargo, se señala un notable incremento del costo computacional y tiempo de procesamiento requerido, especialmente al usar imágenes en color siendo ésta su principal debilidad. Los autores concluyen que aun utilizando imágenes en escala de grises, el costo computacional del segundo enfoque sigue siendo mucho mayor que el del primero pero que su rendimiento y la robustez justifica el uso de técnicas de aprendizaje no supervisado para el proceso de segmentación. También argumentan que esto se debe a la naturaleza iterativa del algoritmo de funcionamiento del PCNN y puede ser mejorado, ya sea por medio de implementaciones paralelizadas o simplemente reemplazando la PCNN por otros algoritmos no supervisados. El aporte de este trabajo se encuentra justamente en esta conclusión, que permite descartar el uso

de algoritmos de segmentación no supervisados cuya implementación resultase computacionalmente compleja.

El trabajo de Di Ruberto y otros (2016) presentó un sistema de conteo del número total de leucocitos en frotis de sangre periférica donde se introdujo un nuevo enfoque para la segmentación de los glóbulos blancos al utilizar técnicas de aprendizaje supervisado en lugar de técnicas basadas en la umbralización. Los autores aseguran que al obtener una exactitud promedio del 99.73% en el conteo de los leucocitos usando 3 bases de datos de acceso público con imágenes de diferente coloración y condiciones de iluminación, su sistema supera el estado del arte en el área. Se considera que el principal aporte del artículo es la creación de un sistema que opera de forma independiente al conjunto de datos usado capaz de realizar el análisis de forma robusta en cualquier imagen de entrada; los investigadores señalan que esto ha sido posible debido al uso de un esquema de aprendizaje automatizado para el proceso de segmentación. Este esquema se inicia con la identificación de 3 áreas de interés (o clases) en la imagen correspondientes a glóbulos blancos, glóbulos rojos y plasma. Posteriormente, la información de color del píxel junto con algunos parámetros estadísticos (promedio, desviación estándar, uniformidad y entropía), calculados en torno a una vecindad de 3x3 elementos, son utilizados como rasgos descriptores para la máquina de soporte vectorial. En sus conclusiones los autores señalan que el método propuesto en su investigación está basado en muchos conocimientos ya documentados en la literatura pero que fueron combinados para construir en esencia un procedimiento completamente novedoso, en el cual la mayor innovación fue producto del uso de un clasificador que hace uso del método del K-vecino más cercano y una máquina de soporte vectorial para la segmentación. Adicionalmente, se incorporó el uso de la transformada circular de Hough, muy adecuada para detectar y contar formas circulares especialmente en el caso de aglomeración de los leucocitos. Con respecto al tiempo requerido para realizar la segmentación, los investigadores acotan que en sus experimentos fue en promedio 8.6 ± 1.4 segundos, pero aclaran que este tiempo

depende de muchos factores incluyendo por supuesto las características del hardware y sistema operativo del computador utilizado, señalando haber usado dos máquinas con equipadas con procesadores Intel® Core™ i7 – 3.10 GHz y 4.0 GB RAM, e Intel® Core™ i7 – 2.10 GHz y 16.0 GB RAM, respectivamente. Pese al éxito de su trabajo sugieren que la extensión del algoritmo a distintos espacios de color pudiera hacer el proceso de segmentación más simple y efectivo para todo tipo de imágenes. El éxito obtenido en el trabajo de estos autores, junto a la capacidad de su propuesta de ser extendida y probada con distintos tipos de imágenes adquiridas usando técnicas de tinción, equipos de captura y condiciones de iluminación diferentes, fue un aporte fundamental para definir la metodología a utilizar para segmentar las células.

Lisicki (2014) describe el desarrollo de un sistema computacional que permite realizar el conteo de diferentes tipos de células sanguíneas basado en métodos computacionales con variados niveles de intervención humana. El autor compara las técnicas de k-vecino más cercano, máquinas de vectores de soporte (SVM) y redes neuronales convolucionales de aprendizaje profundo. Con respecto a este último enfoque, el autor indica que tiene la ventaja de encontrar soluciones rápidas a problemas complejos que involucran soluciones multidimensionales. En las conclusiones del trabajo se considera que una contribución del mismo es la verificación de los algoritmos modernos de aprendizaje automático pueden aprender a caracterizar las imágenes a partir de datos crudos y obtener una clasificación similar o superior a algoritmos que utilizan vectores de rasgos cuidadosamente seleccionados por personal especializado.

Finalmente, en Habibzadeh y otros (2013) se presentó un trabajo centrado en la clasificación y conteo de los cinco tipos principales de leucocitos usando imágenes de baja resolución. En particular se describe únicamente la implementación de tres tipos de clasificadores, los cuales procesan imágenes previamente segmentadas por los autores. Se analizan los resultados obtenidos mediante clasificadores basados en: (a) SVM con rasgos estándares calculados por medio del histograma de la imagen ,

específicamente los cuatro primeros momentos centrales (media, desviación estándar, oblicuidad y curtosis); (b) SVM con rasgos extraídos usando K-PCA (análisis de componentes principales con kernel) , y (c) red neuronal convolucional con extracción automática de rasgos, esto último fue resaltado por los autores como una gran ventaja. Este trabajo constituyó uno de los pioneros en la aplicación exitosa de las redes neuronales convoluciones en el análisis de frotis de sangre periférica y mostró que usando un modelo pre-existente de red convolucional, en este caso la arquitectura LeNet-5 (LeCun, Bottou, Bengio, & Haffner, 1998), era posible clasificar este tipo de imágenes con la misma o mayor exactitud que el estado de arte y tendencias dominantes del momento como las SVM.

2.2. FUNDAMENTOS TEÓRICOS

La sangre es una mezcla compleja de plasma, glóbulos blancos, glóbulos rojos y plaquetas que realiza una gran variedad de funciones mientras recorre el cuerpo humano. La mayoría de las células que la componen se producen en la médula ósea, a partir de un mismo tipo de célula no especializada denominada célula madre o precursora, sin embargo algunas células son producidas en los ganglios linfáticos, timo o bazo (Frenkel, 2003).

El plasma es el componente líquido de la sangre y constituye más de la mitad de su volumen total. Está compuesto por agua, electrolitos y proteínas. Las células sanguíneas, hormonas y demás componentes de la sangre se encuentran en suspensión en el plasma, pudiendo así ser transportados por este a todo el cuerpo.

Los glóbulos rojos o eritrocitos constituyen aproximadamente el 40% del volumen sanguíneo y contienen hemoglobina cuya función principal es transportar el oxígeno desde los pulmones hacia todos los tejidos del cuerpo. Son los responsables del color característico de la sangre.

Las plaquetas o trombocitos son partículas o fragmentos celulares carentes de núcleo mucho más pequeñas que las otras células sanguíneas cuya función principal es intervenir en la homeostasis de la sangre mediante el proceso de coagulación así como favorecer la reparación y regeneración de los tejidos en caso de heridas. Se encuentran presentes en la sangre en menor proporción a los eritrocitos (1 plaqueta por cada 20 eritrocitos) (Frenkel, 2003).

Los glóbulos blancos o leucocitos se encuentran en menor cantidad que los glóbulos rojos con una proporción aproximada de 1 a 660. Los leucocitos son responsables principalmente de la defensa del cuerpo ante infecciones bacterianas, afecciones virales, hongos y otros patógenos. Para ello, poseen gránulos que contienen enzimas capaces de digerir y destruir los microorganismos invasores (Ciesla, 2014).

El proceso de producción, desarrollo, diferenciación y maduración de las células sanguíneas se denomina hematopoyesis (Ciesla, 2014), las etapas por las que atraviesa cada tipo de célula pueden observarse en la Figura . El término hematopoyesis comprende e incluye los procesos de eritropoyesis, trombopoyesis y leucopoyesis, encargados del desarrollo de glóbulos rojos, plaquetas y glóbulos blancos respectivamente. En la Figura se ha omitido la descripción de las citoquinas e interleucinas (sustancias encargadas de promover la transformación de las células de un linaje específico) involucradas en el desarrollo de cada tipo de célula por considerar que dicho conocimiento escapa a los objetivos de la presente investigación.

2.2.1. Leucocitos

Los leucocitos componen menos del 1% del volumen total de la sangre y su presencia en la misma puede durar desde horas hasta un par de días únicamente. Estas células realizan la mayoría de sus funciones en los tejidos del cuerpo a través de los cuales pueden desplazarse y residir por 2 a 5 días. La mayoría de los glóbulos blancos existentes en un individuo se encuentran almacenados en la médula ósea y el bazo, de

donde pueden ser liberados y transportados rápidamente hacia los tejidos en caso de ser requeridos. Los leucocitos que aparecen en la circulación sanguínea forman a su vez parte de dos reservorios diferentes: la reserva marginal del endotelio vascular y el reservorio circulante del torrente sanguíneo (Ciesla, 2014).

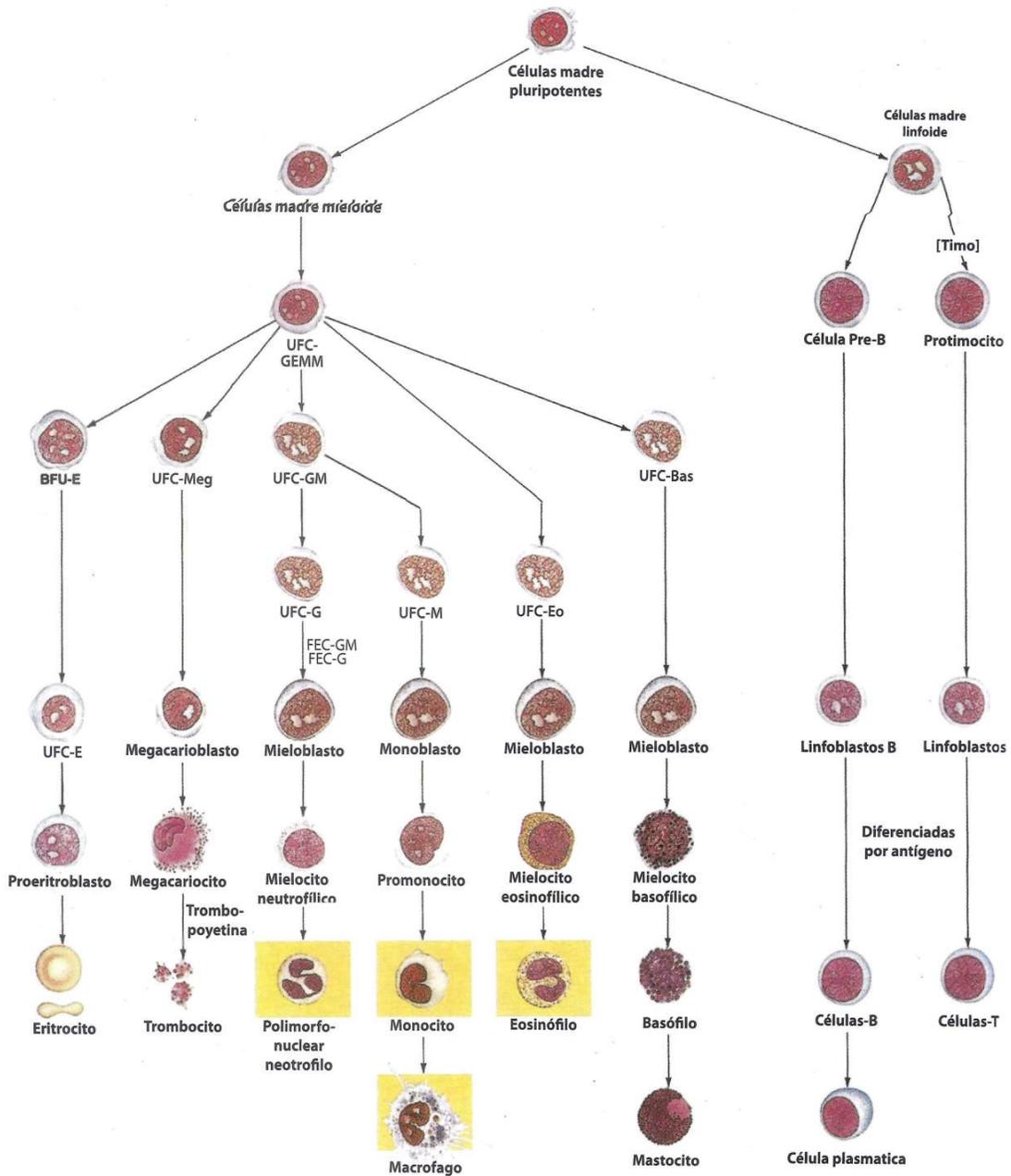


Figura . Proceso de formación las células sanguíneas (hematopoyesis). Adaptado de Ciesla, 2014 p.20

La leucopoyesis o proceso de formación y maduración de los glóbulos blancos es un proceso complejo debido a que en su estado maduro estas células pueden pertenecer a cinco tipos diferentes: neutrófilos, linfocitos, monocitos, eosinófilos y basófilos. Adicionalmente, en su desarrollo los leucocitos pueden pasar por cuatro o cinco (dependiendo del tipo) estadios intermedios. Cielsa (2014) indica que las características claves que permiten identificar a los leucocitos en sus etapas maduras e inmaduras son las siguientes:

- El tamaño celular.
- La proporción entre el núcleo y el citoplasma.
- El patrón de cromatina (coloración).
- Presencia o ausencia de nucléolos.
- Calidad citoplasmática.
- Presencia de gránulos.

La identificación de linfocitos en sus etapas inmaduras es un área sumamente amplia e interesante para investigación con implicaciones en el diagnóstico médico, sin embargo escapa del objeto del presente trabajo.

Según su origen los leucocitos pueden pertenecer a tres series distintas: mielocítica, monocítica o linfocítica. Las primeras dos series derivan únicamente de la médula ósea mientras que la tercera deriva tanto de los órganos linfoides primarios (médula y timo) como de los órganos linfoides secundarios (bazo, ganglios linfáticos, placas de Peyer del tracto gastrointestinal y las amígdalas).

Otra forma de clasificar los leucocitos y que reviste un mayor interés para este proyecto es según la presencia o ausencia de gránulos en el citoplasma en la célula. De esta manera, los leucocitos se pueden clasificar como: granulocitos o agranulocitos.

2.2.2. Leucocitos granulados o granulocitos

Pertenecen a la serie mielocítica debido a que se forman exclusivamente en la médula ósea. En su mayoría presentan un núcleo celular polilobulado con presencia de varios segmentos interconectados por filamentos delgados de cromatina. Por esta razón también se les denomina glóbulos blancos polimorfonucleados. El citoplasma de este tipo de leucocito puede presentar grandes gránulos característicos, siendo esto un rasgo distintivo para su identificación. Dentro de esta categoría se encuentran agrupados los siguientes tipos de glóbulos blancos:

- **Neutrófilos segmentados:** los neutrófilos son el tipo más común de leucocitos, representando normalmente entre un 50% y 70% de total de glóbulos blancos presentes en la sangre. Estas células poseen un diámetro característico comprendido entre 9 a 15 micrómetros. Poseen como función defender al cuerpo ante infecciones bacterianas y micóticas siendo importantes agentes funcionales al tener la capacidad de fagocitar gérmenes patógenos y materiales extraños. Los neutrófilos pueden abandonar los vasos sanguíneos desplazándose por el tejido conectivo y son el primer tipo de célula inmune que responde y se dirige a las infecciones. Su periodo de maduración en la medula ósea es de unos 4 a 10 días y luego de estar en circulación por algunas horas en la sangre migran a los tejidos del sistema respiratorio, al tracto digestivo o al urogenital. Una vez liberados por la medula ósea su tiempo de supervivencia es de pocos días. La Figura muestra una fotografía de un grupo de neutrófilos segmentados en un caso de leucocitosis donde puede observarse la característica presencia del núcleo lobulado en ellos. En el citoplasma se observan gránulos secundarios finos similares al polvo.
- **Eosinófilos:** son un poco más grandes que los neutrófilos con un diámetro entre 10 a 16 micrómetros. Se encargan de combatir infecciones parasitarias y también se observa su aumento en procesos alérgicos. El núcleo típicamente es bi-lobular

y en el citoplasma se observan gránulos relativamente gruesos, uniformemente redondos y grandes. Constituyen entre el 0% y 4% de todos los leucocitos en la sangre. La Figura muestra un eosinófilo con núcleo bilobulado y se puede observar la gran presencia de gránulos en la célula.

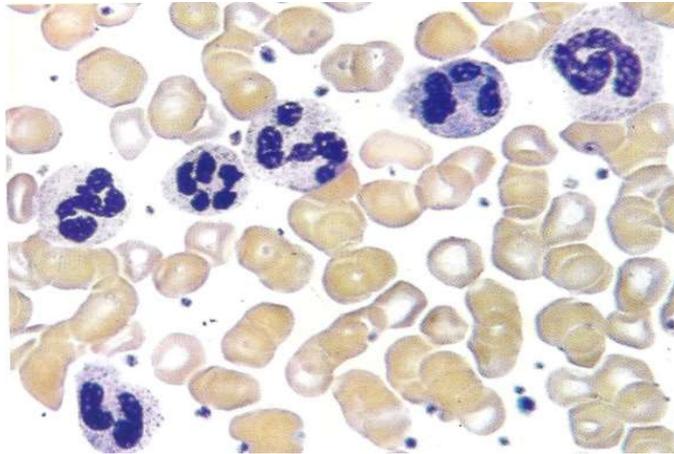


Figura . Grupo de leucocitos neutrófilos maduros. McDonald, Paul y Cruickshank (2001, pág. 81)

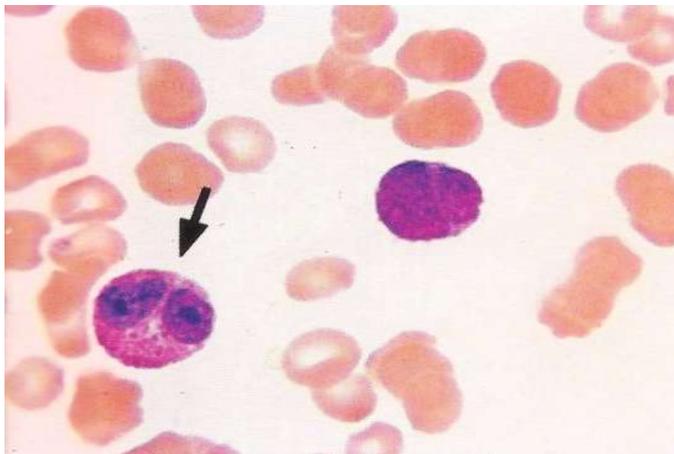


Figura . Eosinófilo bilobular. Cielsa (2014, pág. 133)

- **Basófilos:** son el tipo menos común de granulocitos y representan menos del 0.3% del total de leucocitos. Intervienen en la respuesta alérgica y antigénica al controlar la liberación de histamina causando vasodilatación que promueve el flujo de sangre hacia los tejidos. En el citoplasma de estas células se observan grandes

gránulos que parecen oscurece al núcleo pudiendo llegar a ocultarlo casi totalmente como se muestra en la fotografía de la Figura .

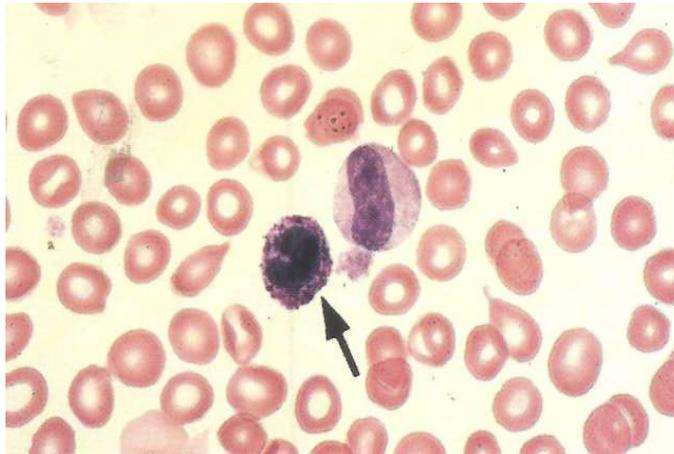


Figura . Basófilo con núcleo difícil de distinguir por los grandes gránulos. Cielsa (2014, pág. 133)

2.2.3. Leucocitos no granulados o agranulocitos

Se les denomina leucocitos mononucleares ya que se caracterizan por tener un núcleo no lobulado. En el citoplasma pueden observarse gránulos muy pequeños. A esta categoría pertenecen:

- **Linfocitos:** constituyen el segundo tipo de leucocito más común en la sangre, representando entre el 20% y 44% del total. Sin embargo, la mayoría de ellos se encuentra en el sistema linfático y no en el circulatorio. Son parte fundamental del sistema inmune pues poseen la capacidad de reconocer y atacar antígenos presentes en el cuerpo. Un linfocito tiene la capacidad de actuar específicamente para un tipo de antígeno, de manera que se activa únicamente cuando detecta la presencia del antígeno particular que lo estimula y comenzar a producir sustancias químicas para destruirlo. Esto ocurre especialmente ante la presencia de virus. Su tamaño varía entre 7 a 12 micrómetros (linfocito pequeño) hasta 15 a 18 micrómetros de diámetro (linfocito grande). El núcleo de los linfocitos es generalmente de forma circular u ovalada, denso y sin gránulos, encerrado por un citoplasma casi del mismo tamaño del propio núcleo. Existen dos tipos principales

de linfocitos: B y T, formados ambos a partir de células madre de la médula ósea. Sin embargo, los linfocitos T abandonan rápidamente la médula para realizar su proceso de maduración en el timo. La Figura muestra una comparación entre un linfocito pequeño y otro grande.

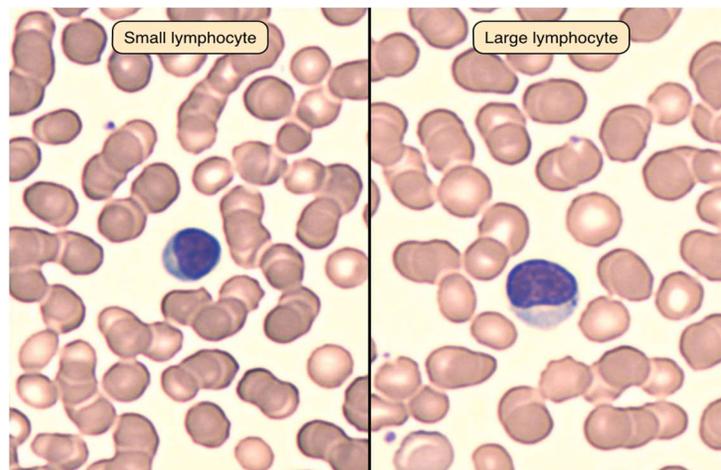


Figura . Linfocitos: pequeño (izquierda) y grande (derecha). Recuperado de “Systems Cell Biolog”: http://medcell.med.yale.edu/systems_cell_biology/blood_lab.php

- **Monocitos:** Son las células sanguíneas de mayor tamaño, pudiendo tener un diámetro entre 12 y 20 micrómetros. El núcleo celular es grande, frecuentemente arriñonado, a veces lobulado, y por lo general, situado algo excéntricamente. Su citoplasma contiene gránulos muy finos. Se forman en la médula ósea y permanecen pocas horas en el torrente sanguíneo antes de dirigirse a otros órganos y tejidos donde son almacenados. Cumplen una función de fagocitosis, no solamente de los agentes patógenos, sino también de otras células sanguíneas dañadas o muertas y eventualmente se transforman en macrófagos de tejidos, encargados de remover células muertas y atacar microorganismos. Constituyen entre el 2% y el 9% de los leucocitos en la sangre. En la Figura se puede observar un típico monocito con núcleo de forma arriñonada.



Figura . Monocito típico. Recuperado de “Systems Cell Biolog”:
http://medcell.med.yale.edu/systems_cell_biology/blood_lab.php

2.2.4. Frotis de sangre periférica

Una extensión o frotis sanguíneo consiste en recubrir parcialmente un portaobjetos con una gota de sangre, de tal manera que las células de ésta se dispongan formando una sola capa de ellas.

El frotis sanguíneo es esencial en el estudio clínico hematológico. Este análisis clínico arroja datos importantes como la cantidad, morfología y dimensión de las células componentes de la sangre. Lo anterior favorece el esclarecimiento de patologías sanguíneas. Además, el frotis sanguíneo puede arrojar información acerca de patógenos existentes en el torrente sanguíneo.

La importancia de un buen extendido y su correcta tinción son fundamentales a la hora de la observación al microscopio, ya que un frotis mal hecho entrega muy poca información tanto cuantitativa como cualitativa. Su preparación se realiza colocando una gota de sangre sobre un extremo de un portaobjetos o lámina de vidrio limpio, el tamaño de ésta es importante ya que define el grosor o longitud del frotis resultante (Rodak & Carr, 2014), y expandiéndola usando un segundo portaobjetos. Durante este proceso el técnico debe sostener con firmeza el portaobjeto extensor en un ángulo

constante de 30 a 45 grados (Figura - A) y hacer que toque la gota de sangre hasta que esta se esparza completamente por la superficie en contacto entre ambas láminas (Figura - B). Finalmente, el extensor se desliza con rapidez y suavidad hacia el extremo libre del portaobjetos que tiene la gota de sangre hasta expandirla completamente (Figura - C).

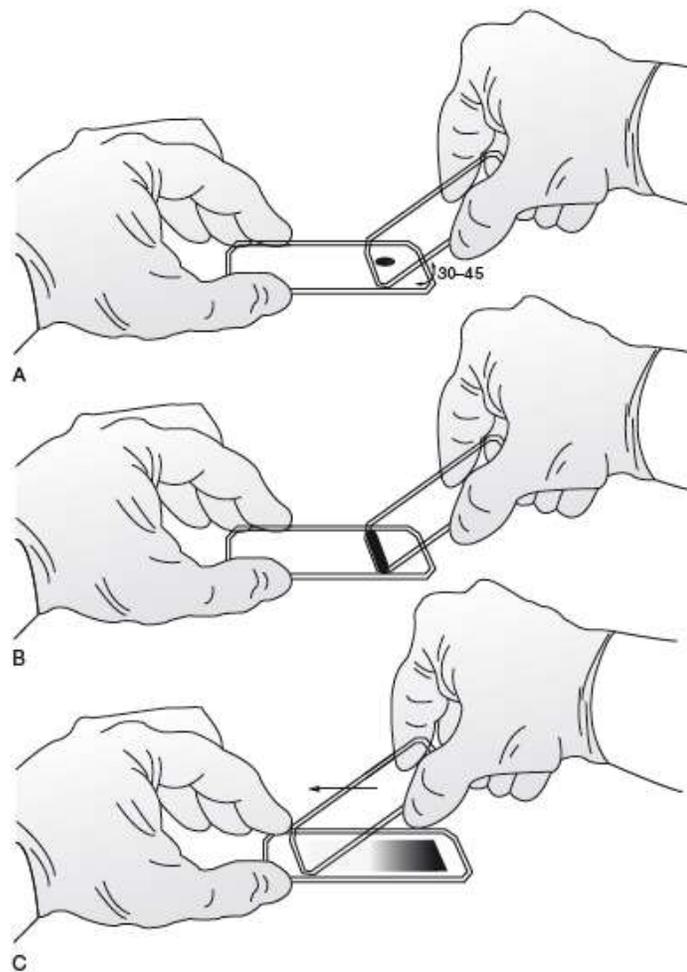


Figura . Preparación de frotis de sangre periférica en cuña. **Nota.** **A:** Mantener ángulo correcto al hacer extensión. **B.** Extensión de la sangre totalmente en la superficie de contacto entre los portaobjetos. **C.** Resultado esperado al desplazar el portaobjeto extensor. Tomado de Rodak & Carr (2014, pág. 3).

El procedimiento requerido para efectuar correctamente el extendido de sangre periférica, junto con diversas consideraciones pertinentes a garantizar su calidad, puede

ser consultado por el lector los trabajos de Rodak & Carr (2014, págs. 2-5) , y Cielsa (2014, págs. 306-307).

Al extender la gota de sangre sobre la lámina, se observan tres zonas claramente diferenciadas en las cuales puede variar la distribución de los leucocitos, estas son: cabeza, cuerpo y cola (ver Figura).

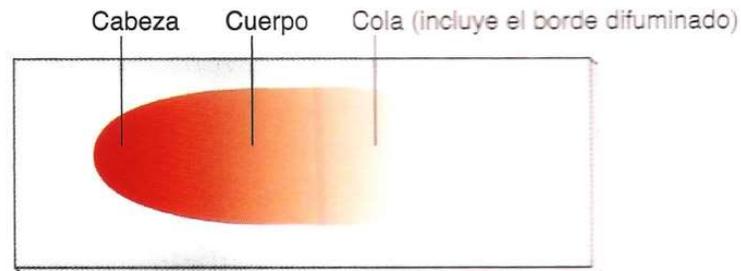


Figura . Extensión de gota de sangre para el frotis sanguíneo. Cielsa (2014, pág. 307)
 La zona en la cual inicialmente se depositó la gota (denominada cabeza) es excesivamente gruesa y no puede ser valorada adecuadamente. A continuación se encuentra el cuerpo de la extensión, que corresponde a la región intermedia del frotis y en ella existe un reparto equilibrado de las células; es la zona ideal para la observación microscópica. Por último, al final de la extensión, se encuentran las “barbas” o la “cola”, es una zona fina dónde las células adoptan una disposición de cordones.

Las características que definen una buena extensión son las siguientes:

- Debe cubrir cerca de $2/3$ a $3/4$ partes de la longitud partes del portaobjetos.
- Ha de ser fina y homogénea.
- La cabeza debe estar cerca de uno de los extremos del portaobjetos.
- La cola debe estar cercana al otro extremo, sin llegar a éste.
- Los bordes laterales de la extensión deben visualizarse y estar separados de los bordes del portaobjeto por 1 mm aproximadamente.
- Cuando el portaobjetos se observa a la luz, la cola del frotis debe adoptar un aspecto en “arco iris”.
- Se toma y se extiende la totalidad de la gota de sangre.

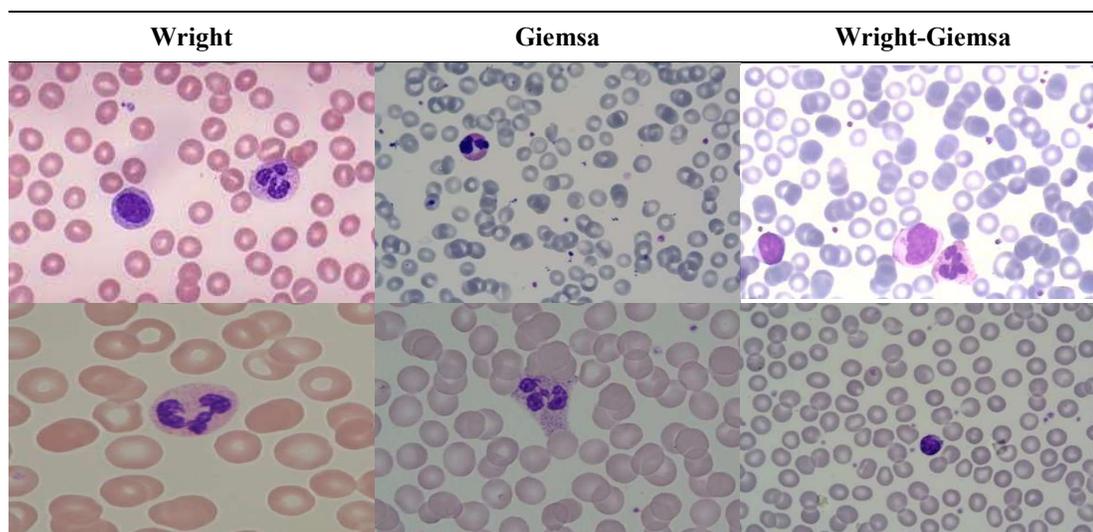
2.2.5. Tinción del frotis de sangre

El objetivo de la tinción o coloración del frotis de sangre periférica es identificar las células y reconocer con facilidad la morfología a través del microscopio. Las tinciones de Wright, Giemsa y Wright-Giemsa, entre otras; son ampliamente utilizadas para los frotis de sangre periférica y de médula ósea. Estas coloraciones policrómicas, denominadas tipo Romanowsky, contienen combinaciones de componentes químicos con pH ácido (eosina) y alcalino (azul de metileno, azure A, azure B). Los colores resultantes presentan variaciones entre los diferentes laboratorios según el método utilizado como puede evidenciarse en la Tabla .

Al aplicar la tinción, ocurren reacciones que dependen del pH entre los componentes de las células y los químicos colorantes utilizados. El azul de metileno es básico y reacciona con las partes con los componentes celulares ácidos como el ARN (ácido ribonucleico) tiñéndolos de azul. Por su parte, la eosina es ácida y tiñe de rojo los componentes celulares básicos como los gránulos de los eosinófilos. Los neutrófilos segmentados poseen gránulos citoplasmáticos finos con pH neutro y por tanto aceptan algunas de características de ambos colorantes (Rodak & Carr, 2014).

Tabla

Coloraciones de uso frecuente en la preparación del frotis de sangre periférica



Nota. Imágenes extraídas de las bases de datos utilizadas en el presente trabajo, las cuales, se describen en el capítulo 4

Un frotis teñido de manera óptima presenta las siguientes características (Rodak & Carr, 2014) :

- Los eritrocitos deben ser de color rosado a salmón (Wright) o grisáceo (Giemsa o Wright-Giemsa)
- Los núcleos son de color azul oscuro o violeta.
- Los gránulos citoplasmáticos de los neutrófilos son de color lavanda a lila.
- Los gránulos citoplasmáticos de los basófilos son de color azul oscuro a negro.
- Los gránulos citoplasmáticos de los eosinófilos son de color rojos a anaranjados.
- El citoplasma de los monocitos es azul grisáceo.
- El citoplasma de los monocitos es rosado pálido (Wright) o rosado grisáceo (Giemsa o Wright-Giemsa)
- La zona entre las células debe ser incolora, clara, limpia y sin colorante precipitado.

2.2.6. Conteo diferencial de leucocitos

El conteo diferencial de glóbulos blancos es una evaluación de los leucocitos maduros en la circulación periférica y aunque proporciona solo una indicación instantánea de la población de glóbulos blancos en un instante particular del tiempo, ofrece una valiosa información sobre el perfil hematológico de una persona y la respuesta hematológica de la misma a cualquier circunstancia. Generalmente el diferencial es realizado en un frotis periférico bien distribuido y bien teñido.

Por medio de este examen, se busca obtener información de la distribución relativa de los diferentes leucocitos en la sangre. Los resultados obtenidos son gran utilidad para el diagnóstico de múltiples enfermedades, orientan en la elección del tratamiento a seguir en los pacientes y pone de manifiesto la existencia de efectos secundarios en pacientes que reciben quimioterapia o radioterapia.

La estimación de glóbulos blancos se realiza utilizando un aumento de 40X o 100X en el microscopio y se contabiliza un total de 100 células, recorriendo el extendido en un movimiento de zigzag. El porcentaje y la identificación de cada tipo de célula son registrados a medida que se realiza el conteo y los resultados son comparados con los rangos de referencia de acuerdo a la edad del individuo.

En determinadas circunstancias se recomienda realizar el conteo diferencial de 200 células en lugar de las 100 requeridas por el procedimiento estándar (Ciesla, 2014, pág. 138). Estos casos incluyen: conteo global de glóbulos blancos mayor a $35,0 \times 10^9$ / litro, presencia muy elevada ($>40\%$) o muy reducida ($<17\%$) de linfocitos, conteo de monocitos mayor al 12%, observación de células inmaduras (blastos) por primera vez en un paciente.

2.2.7. Concepto de imágenes digitales

Una imagen es una representación bidimensional de una escena del mundo tridimensional, que se obtiene como resultado de la adquisición de una señal proveniente de un sensor capaz de convertir la información del espectro electromagnético en una representación numérica (Sánchez-Salmerón & Ricolfe-Viala, 2016). De forma genérica, una imagen digital se define entonces como una matriz de dimensiones $N \times M$, donde cada elemento (denominado píxel) cuantifica, en un número finito de bits, un nivel de información asociado a una región del objeto representado.

Dicho de otra forma, una imagen bidimensional es una función que a cada par de coordenadas (x, y) asocia un valor relativo a alguna propiedad del punto que representa. En las imágenes en escala de grises ésta información es el brillo, mientras que, en las imágenes a color la información corresponde a la intensidad de cada una de las componentes de una base de color (Vélez, Moreno, Sánchez, & Sánchez, 2003). Los valores de cada píxel representan desde niveles oscuros de luminosidad hasta valores

claros. El nivel más oscuro es el valor más bajo del intervalo y viene representado por el negro y el nivel más claro se representa con el blanco y es el valor más alto.

2.2.8. Espacios o modelos de color para imágenes digitales

Un espacio o modelo de color es una forma de representación mediante la cual se puede especificar, crear o visualizar cualquier color de forma normalizada mediante un sistema de coordenadas, generalmente tridimensionales. La especificación numérica de un color en un espacio particular, se efectúa mediante tres o cuatro cantidades (luminancias) las cuales lo definen de forma cualitativa y cuantitativamente. No todos los espacios de colores pueden representar exactamente los mismos colores y los sub-espacios que en ellos se definen coinciden en mayor o menor manera, con las capacidades del ojo humano para percibir el color de los objetos. A continuación se describen algunas características de los espacios de color de uso más frecuente.

2.2.8.1. Espacio de color RGB

Es sin lugar a duda el modelo de color más comúnmente utilizado ya que es empleado por los periféricos informáticos como pantallas, cámaras fotográficas y escáneres (Martín, García, & Armingol, 2016). Se compone de tres señales de luminancia cromática: roja, verde y azul (R, G, B) las cuales se mezclan para obtener una gran variedad de posibles colores, siendo así un modelo de color aditivo. Los píxeles de una imagen representada usando el modelo RGB usualmente tienen una profundidad de 8 bits, por lo cual, existe un total de 256 posibles valores de intensidad para cada componente de color cuyo rango abarca de 0 a 255. Cuando todas las componentes tienen el valor mínimo de intensidad, el color resultante es negro; por el contrario, cuando sus valores son máximos el resultado es blanco.

La ubicación de los colores en el espacio se representa mediante un cubo debido a que el modelo de color está basado en un sistema de ejes cartesiano (ver Figura). Las esquinas del cubo corresponden los tres colores primarios (rojo, verde, azul), los tres

colores secundarios (cian, magenta, amarillo), el blanco y el negro, estando éste último ubicado en el origen. La diagonal entre el blanco y el negro representa la escala de grises, la cual es ampliamente usada para el procesamiento de imágenes.

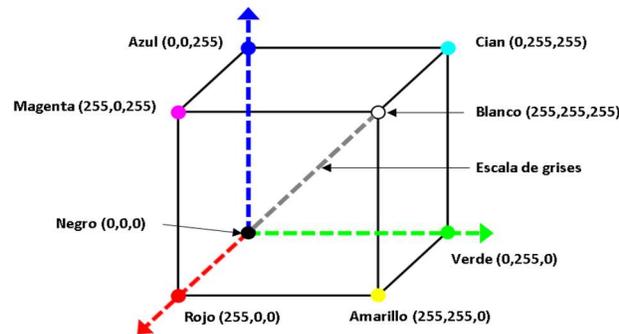


Figura . Representación del espacio de color RGB

2.2.8.2. Espacio de color CMYK

CMYK es un modelo de color sustractivo (a diferencia de RGB) que se basa en la mezcla de pigmentos de cuatro colores (C = Cian, M = Magenta, Y = *Yellow* y K = *Black*), lo cual, le hace adecuado para los sistemas de impresión de inyección de tinta y fotocopiado, al permitir utilizar el negro en la conformación de diferentes colores con un significativo ahorro en el uso de las tintas pigmentadas. La conversión más común del modelo RGB a CMYK se realiza, según lo indicado por Pratt (2001), utilizando la ecuación (). Los valores de u y b son coeficientes que permiten ajustar la imagen resultante, en el caso más simple se colocan en 1.0.

$$\begin{aligned} C &= 255 - R - uK_b \\ M &= 255 - G - uK_b \\ Y &= 255 - B - uK_b \\ K &= bK_b \end{aligned} \quad ()$$

Donde:

$$\begin{aligned} K_b &= \min \{255 - R, 255 - G, 255 - B\} \\ 0.0 &\leq u \leq 1.0 \\ 0.0 &\leq b \leq 1.0 \end{aligned}$$

2.2.8.3. Espacios de color HSL, HSV y HSI

Estos espacios de color se basan en la forma en la cual los seres humanos describen el color de un objeto, caracterizándolo en términos del matiz o tono (*Hue*), saturación (*Saturation*) y brillo (*Brightness*) de forma similar a la utilizada por los artistas en sus

composiciones mediante los conceptos de matices, tintes, sombras y tonos. Constituyen espacios cilíndricos donde H corresponde al ángulo medido sobre el eje vertical central, S señala el radio y la componente restante (L, V o I, según el caso) indica la altura sobre la base como se indica en la Figura .

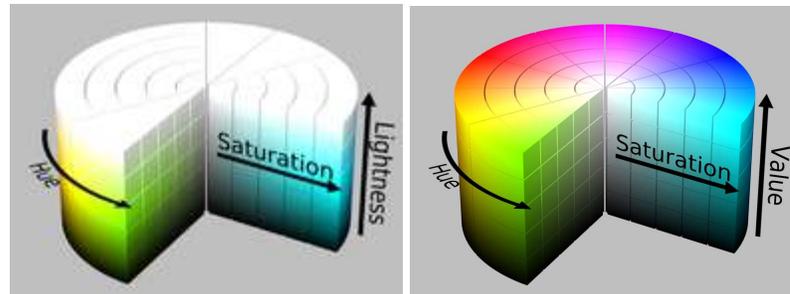


Figura . Modelos de color HSL y HSV

Sin embargo, no todas las combinaciones de valores en estas coordenadas tienen una correspondencia válida en el espacio RGB y por ello se suelen representar usando un subconjunto del espacio total. Por ejemplo los modelos HSI y HSL son comúnmente representados mediante una bipirámide hexagonal donde el vértice superior corresponde al color blanco y el inferior al negro; mientras que, el HSV se presenta como un cono o un cono hexagonal como se muestra en la *Figura .*

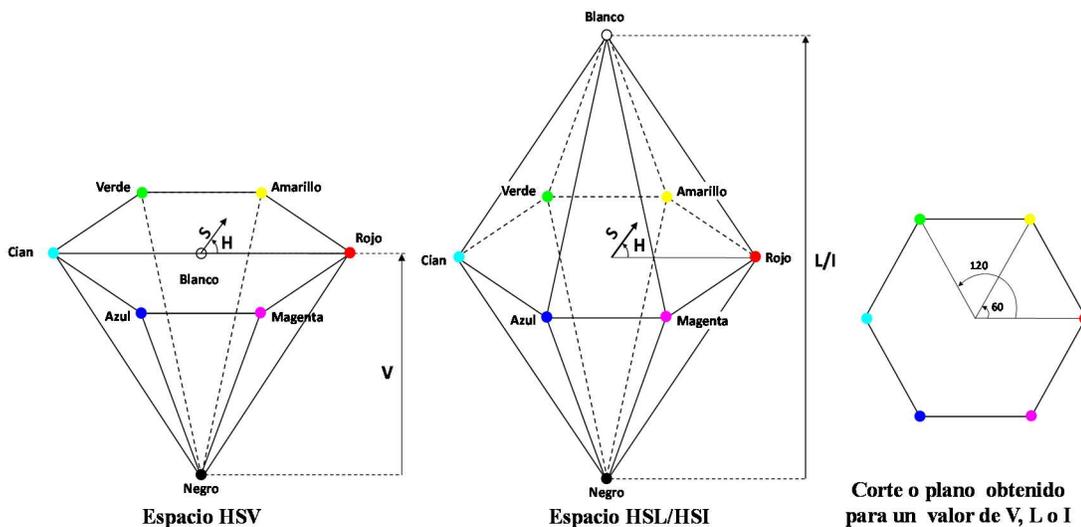


Figura . Representación de los espacios de color HSV, HSL y HSI.

La componente tono (H) es un atributo que describe al color propiamente dicho, siendo una medida de la pureza del mismo y se relaciona con la longitud de onda que es percibida por el observador. Se puede obtener a partir de los valores normalizados de las componentes RGB mediante la ecuación () (Gonzalez & Woods, 2002, p. 299). Típicamente, se representa mediante un ángulo comprendido de 0 a 360 grados (donde 0, 60 y 120 grados corresponden a los colores rojo, amarillo y verde, respectivamente); sin embargo, también puede emplearse un rango normalizado de 0 a 1 como ocurre en el paquete informático MATLAB al usar la función de conversión *rgb2hsv*.

$$r = \frac{R}{255}; g = \frac{G}{255}; b = \frac{B}{255}$$

$$H = \begin{cases} \theta, & b \leq g \\ 360 - \theta, & b > g \end{cases} \quad ()$$

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(r-g) + (r-b)]}{\sqrt{(r-g)^2 + (r-b)(g-b)}} \right\}$$

La saturación (S) indica el grado de dilución del color con blanco de manera que refleja la intensidad o palidez del mismo. Los colores puros (también llamados tonos) se observan muy vivos o intensos, estando completamente saturados. Esta componente tiene una expresión distinta en cada uno de los espacios de color mencionados, la cual puede obtenerse a partir de las expresiones mostradas en la ecuación (). Se suele expresar en términos porcentuales (en rango de 0 a 100) o normalizada de 0 a 1.

$$\text{HSV}$$

$$S = \begin{cases} 0, & \max(r, g, b) = 0 \\ \frac{\max(r, g, b) - \min(r, g, b)}{\max(r, g, b)}, & \max(r, g, b) \neq 0 \end{cases}$$

$$\text{HSL}$$

$$\Delta = \max(r, g, b) - \min(r, g, b)$$

$$S = \begin{cases} 0, & \Delta = 0 \\ \frac{\Delta}{1 - |\max(r, g, b) + \min(r, g, b) - 1|}, & \Delta \neq 0 \end{cases} \quad ()$$

$$\text{HSI}$$

$$S = 1 - \frac{3 \cdot \min(r, g, b)}{r + g + b}$$

Finalmente, el brillo de un color es un descriptor subjetivo que es prácticamente imposible de medir (Gonzalez & Woods, 2002). Este representa la claridad u oscuridad con la cual es percibido el color. Para los espacios HSI y HSL, su tercera componente abarca la noción acromática de intensidad y se relaciona con la medición de la energía presente en el color, mientras que, el modelo HSV la coordenada V mide la distancia desde un tono cualquiera hasta el negro (Smith, 1978). Estas definiciones implican que mientras los colores puros como rojo, blanco y amarillo tienen en el sistema HSV el mismo valor ($V=100\%$ ya no poseen componentes de negro), sus valores de intensidad en el sistema HSI (rojo = 33.33%, blanco = 100% y amarillo = 66.67%) sean completamente diferentes. En los tres espacios la última componente se expresa típicamente en términos porcentuales o de forma normalizada en el rango de 0 a 1. Las expresiones empleadas para obtener esta tercera componente (V, L o I) partiendo modelo RGB, se muestran en la ecuación ().

$$\begin{array}{l}
 \text{HSV} \\
 V = \max(r, g, b) \\
 \text{HSL} \\
 L = \frac{\max(r, g, b) + \min(r, g, b)}{2} \\
 \text{HSI} \\
 I = \frac{r + g + b}{3}
 \end{array} \quad ()$$

En los tres modelos la información del brillo, luminosidad o intensidad se encuentra desacoplada en mayor o menor medida (de forma especialmente apreciable en el espacio HSI) de las componentes que definen propiamente al color, mostrando un aspecto favorable de cara a realizar la segmentación de las imágenes en atención al tono o tinte del color, con mayor invariancia a la iluminación (Martín, García, & Armingol, 2016).

2.2.8.4. Espacios de color XYZ y $L\alpha\beta$ (LAB)

La capacidad de percepción del color humana se debe a la existencia de tres tipos de receptores fotosensibles denominados conos S, M y L, los cuales exhiben una respuesta ante la energía incidente con picos máximos en longitudes de ondas distintas (en torno

a 420-440 nm, 530-550 nm y 560-580 nm, respectivamente). Al observar un objeto iluminado los tres tipos de células emitirán un determinado nivel de respuesta cuya combinación será empleada en la interpretación del color percibido. Esto constituye la base de la teoría tricromática de la visión en color, la cual establece que es posible igualar la respuesta visual humana ante una fuente de luz determinada mediante una combinación, en las cantidades apropiadas, de tres fuentes lumínicas diferentes.

Esto fue estudiado durante los años 1920, cuando investigadores midieron la repuesta perceptiva de distintos sujetos ante estímulos luminosos de diferente intensidad y longitud de onda. Los resultados de éstos experimentos derivaron en la creación de un conjunto de curvas normalizadas de concordancia de color (*color matching curves*) también conocidas con el nombre de observador estándar (ver Figura), las cuales definen los coeficientes de combinación que permiten, a partir de los tres colores base del sistema, reproducir todos los colores percibidos por el observador humano promedio

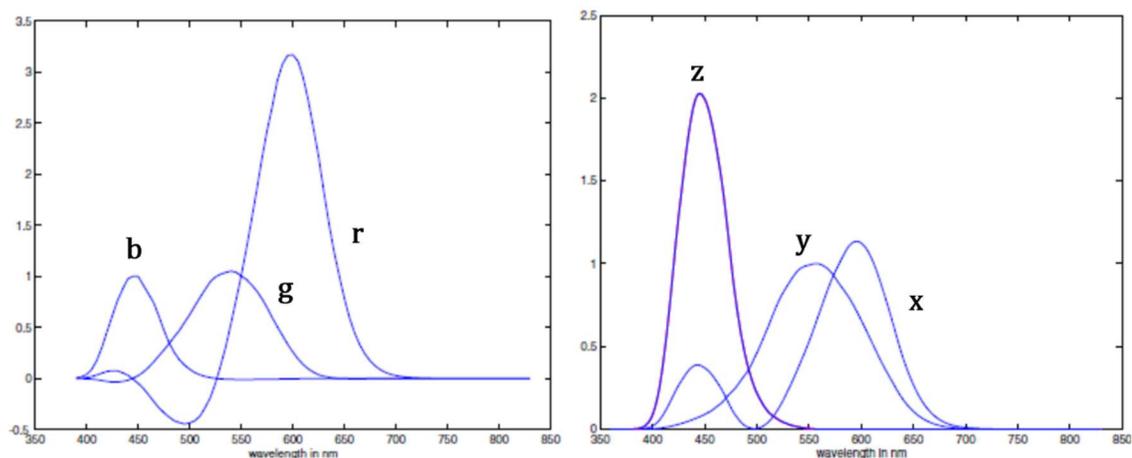


Figura . Curvas de concordancia de color de los espacios CIE-RGB y CIE- XYZ. **Nota.** En las curvas correspondientes al espacio RGB (izquierda) se observa la existencia de valores negativos lo cual implica que la igualación de ciertos colores requeriría una combinación sustractiva. Por el contrario, en el espacio CIE-XYZ (derecha) son positivos. Adaptado de Forsyth & Ponce (2011, pág. 64)

Como puede verse en la Figura , en el modelo de color RGB aparecen coeficientes negativos para colores de determinadas longitudes de onda, lo cual supone según

Martín y otros (2016) una fuente de error importante en los cálculos colorimétricos (comparación de un color monocromático con el blanco de referencia). Por esta razón, la CIE (*Commission Internationale de l'Eclairage* – Comisión Internacional de Iluminación), autoridad en internacional en la materia, estableció en 1931 un espacio de color llamado XYZ con la finalidad de evitar los coeficientes negativos. En éste, la ordenada Y (luminosidad) es perpendicular al plano definido por XZ (romaticidad) de luminancia nula. La transformación entre RGB y XYZ, se puede realizar mediante las expresiones de la ecuación () (Martín y otros, 2016).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.431 & 0.342 & 0.178 \\ 0.222 & 0.707 & 0.071 \\ 0.020 & 0.130 & 0.939 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 3.063 & -1.393 & -0.476 \\ -0.969 & 1.876 & 0.042 \\ 0.068 & -0.229 & 1.069 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad ()$$

El espacio XYZ permite describir la totalidad de los colores percibidos por un ser humano promedio. Sin embargo no es perceptualmente uniforme, es decir, un cambio en una coordenada no produce el mismo cambio perceptual independientemente de su posición en el espacio. Por ello, a partir de las componentes XYZ se definieron los espacios CIE ($L^*u^*v^*$) y CIE- $(L^*a^*b^*)$, también llamado $L\alpha\beta$ o LAB, los cuales se caracterizan por ser uniformes. En los tres sistemas la información correspondiente al tono, tinte o cromatismo se encuentra separada de la luminosidad o intensidad del color, esto los hace especialmente útiles para ser empleado por algoritmos de segmentación de imágenes.

El modelo de color CIELAB fue derivado en 1976 a partir del espacio XYZ. Este determina el color en un sistema de coordenadas tridimensionales (ver Figura) donde L representa la componente de luminosidad y abarca un rango de 0 para el negro y 100 para el blanco, A representa la referencia respecto al eje formado por los colores opuestos rojo-verde y B respecto a la relación amarillo-azul. En este espacio la información colorimétrica está referida a un blanco de referencia y se utilizan relaciones no lineales que pretenden imitar la respuesta logarítmica del ojo humano.

LAB es empleado como base para la transferencia de color entre imágenes tanto en la propuesta de Reinhard y otros (2001), como en la de Grundland y Dodgson (2004).

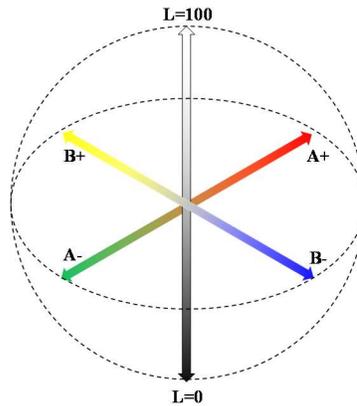


Figura . Representación esférica del espacio LAB

Para la transformación desde el espacio XYZ al espacio LAB se pueden emplear las expresiones de la ecuación () (Forsyth & Ponce, 2011). Los valores de X_n , Y_n y Z_n corresponden con la referencia del color blanco en el espacio XYZ.

$$\begin{aligned}
 L^* &= 116 \sqrt[3]{\frac{Y}{Y_n} - 16} \\
 a^* &= 500 \left[\sqrt[3]{\frac{X}{X_n} - \sqrt[3]{\frac{Y}{Y_n}}} \right] \\
 b^* &= 200 \left[\sqrt[3]{\frac{Y}{Y_n} - \sqrt[3]{\frac{Z}{Z_n}}} \right]
 \end{aligned}
 \quad ()$$

En el paquete informático MATLAB puede usarse al usar la función de conversión *rgb2lab* para transformar una imagen estándar *srgb* al espacio LAB, en este caso el programa maneja de forma automática la conversión intermedia al espacio XYZ. En este trabajo las conversiones a LAB se realizaron usando como blanco de referencia el obtenido mediante iluminación estándar CIE D65.

En el procesamiento de imágenes digitales, es común el uso de transformaciones entre los diferentes espacios de color con el fin de resaltar características de los objetos presentes las cuales pueden o no ser visibles en función del espacio de color utilizado.

Como ejemplo de ello, en la Figura se puede ver una fotografía de frotis sanguíneo visualizada en los espacios de color RGB, HSV, HSV, LAB y CMYK. En ella se observa que diferentes partes de la célula como el núcleo son más visibles en determinadas componentes del espacio de color.

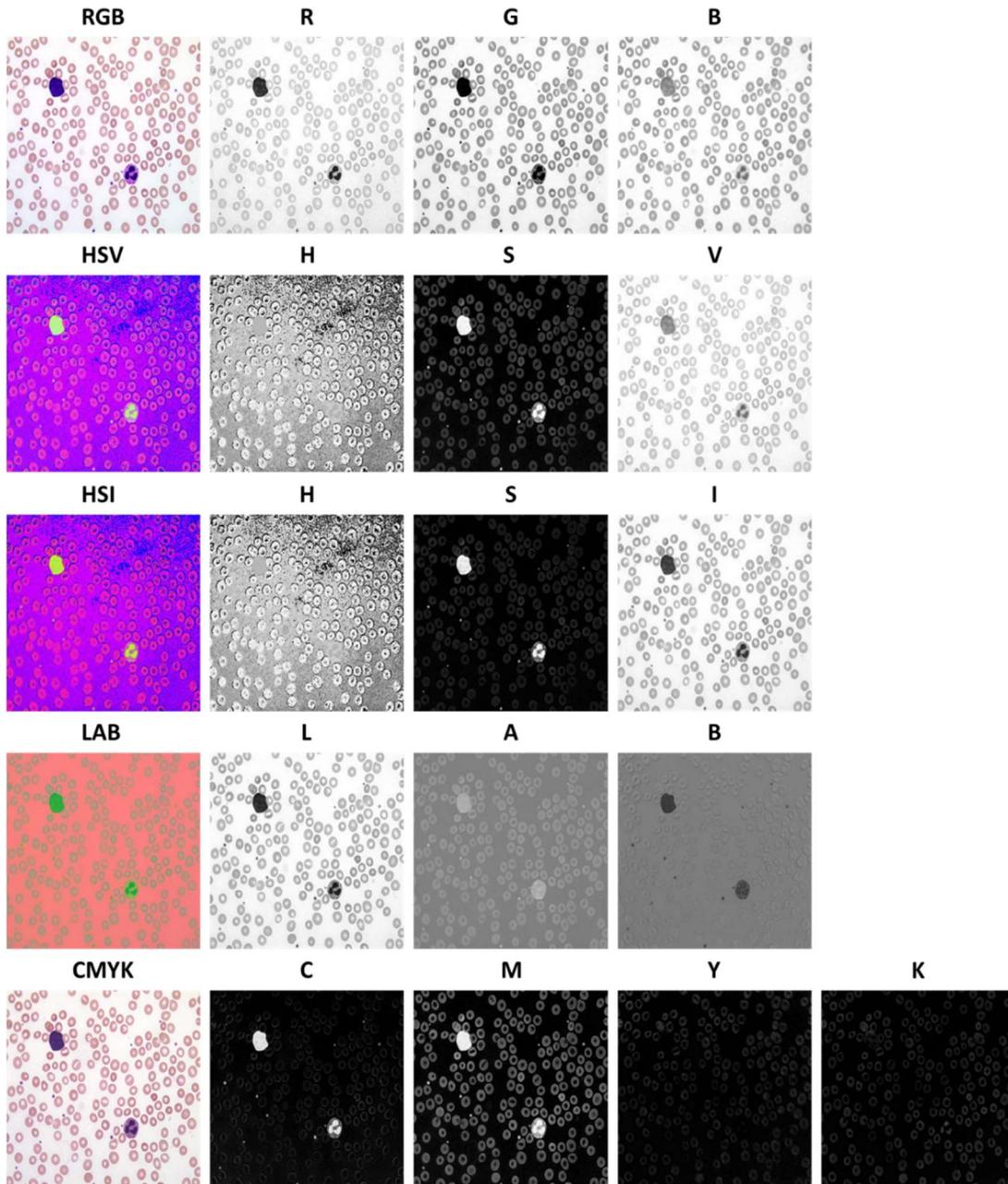


Figura . Frotis sanguíneo visualizado en diferentes espacios de color

2.2.9. Pre-procesamiento y filtrado de imágenes

En esta sección se describirán operaciones previas que se suelen realizar en imágenes digitales con el fin de obtener mejoras y realces necesarios para el funcionamiento adecuado de los algoritmos de procesamiento más complejos. También se presentan operaciones básicas que constituyen en sí mismas algoritmos de procesamiento clásicos de baja complejidad.

Para realizar mejoras en las imágenes digitales, existe una gran cantidad de técnicas y métodos tanto en el dominio espacial como de la frecuencia, pudiendo operar directamente sobre los píxeles con expresiones simples o requiriendo mayor complejidad matemática y computacional para su aplicación. No se pretende profundizar en ninguno de estos conceptos sino brindar una información general al respecto.

2.2.9.1. Transformaciones puntuales

Son transformaciones que afectan únicamente al píxel involucrado en la operación sin provocar ninguna modificación en la intensidad de los píxeles adyacentes (vecindad). Para ello hacen uso de un operador que indica como ocurrirá la modificación, uno a uno, de los píxeles como resultado de aplicar una determinada función sobre su intensidad en la imagen original, es decir $S(i, j) = f(E(i, j))$, donde $E(i, j)$ es la imagen de entrada, $S(i, j)$ la imagen de salida y $f()$ la operación aplicada. Entre los operadores típicos se pueden enumerar (Herrera, Guijarro, & Guerrero, 2016) :

- Identidad: $S(i, j) = E(i, j)$
- Inverso: $S(i, j) = 255 - E(i, j)$
- Umbral : $S(i, j) = \begin{cases} 0, & E(i, j) \leq p \\ 255, & E(i, j) > p \end{cases}$
- Intervalo de umbral binario: $S(i, j) = \begin{cases} 0, & p_1 < E(i, j) < p_2 \\ 255, & E(i, j) \leq p_1 \text{ o } E(i, j) \geq p_2 \end{cases}$

- Intervalo de umbral binario invertido: $S(i, j) = \begin{cases} 255, & p_1 < E(i, j) < p_2 \\ 0, & E(i, j) \leq p_1 \text{ o } E(i, j) \geq p_2 \end{cases}$
- Umbral escala de grises: $S(i, j) = \begin{cases} E(i, j), & p_1 < E(i, j) < p_2 \\ 255, & E(i, j) \leq p_1 \text{ o } E(i, j) \geq p_2 \end{cases}$
- Umbral escala de grises invertido: $S(i, j) = \begin{cases} 255 - E(i, j), & p_1 < E(i, j) < p_2 \\ 255, & E(i, j) \leq p_1 \text{ o } E(i, j) \geq p_2 \end{cases}$
- Extensión del nivel de gris:

$$S(i, j) = \begin{cases} (E(i, j) - p_1) \frac{255}{p_2 - p_1}, & p_1 < E(i, j) < p_2 \\ 0, & E(i, j) \leq p_1 \text{ o } E(i, j) \geq p_2 \end{cases}$$

- Reducción del niveles de gris:

$$S(i, j) = \begin{cases} 0, & E(i, j) \leq p_1 \\ S_1(i, j), & p_1 < E(i, j) \leq p_2 \\ S_2(i, j), & p_1 < E(i, j) \leq p_3 \\ \dots & \\ S_n(i, j), & p_{n-1} < E(i, j) \leq 255 \end{cases}$$

Dentro de esta categoría también se pueden agregar las operaciones de adición, sustracción, multiplicación y división de imágenes, así como transformaciones lógicas conjunción (*and*), disyunción (*or*), inversión (*not*), *xor*, u otras derivadas en imágenes binarias.

Las operaciones por puntos pueden ser realizadas para realizar mejoras en la calidad de la imagen siendo especialmente importantes al efectuar el ajuste (aumento o disminución) del contraste y/o la ecualización del histograma.

2.2.9.2. Transformaciones geométricas

Este tipo de transformaciones describen cambios en la posición de los píxeles en lugar de modificaciones en sus intensidades, esto implica que su distribución respecto al sistema de coordenadas de la imagen sufre modificaciones como resultado de la operación. Las transformaciones geométricas más usuales son (Vélez y otros, 2003):

- **Traslación:** consiste en el desplazamiento de los píxeles por una determinada cantidad horizontal (d_x) y/o vertical (d_y) de forma que el píxel (i, j) de la imagen original alcanza la posición $(i + d_x, j + d_y)$, es decir: $S(i + d_x, j + d_y) = E(i, j)$.
- **Rotación:** implica el giro en un determinado ángulo (θ) de los píxeles en torno al origen de coordenadas. La posición del píxel (i, j) luego de la rotación será determinada por:

$$x = i \cdot \cos \theta - j \cdot \sin \theta ; y = i \cdot \sin \theta + j \cdot \cos \theta$$

- **Escalado:** consiste en la variación del tamaño de la imagen original en cualquiera de sus dimensiones de acuerdo a un factor de escala (s_x, s_y) . Si el factor es menor que 1, se produce una reducción de la imagen mientras que valores mayores implican su aumento. Las coordenadas del punto (i, j) luego de su aplicación se obtienen como:
 $x = s_x \cdot i , y = s_y \cdot j$

Para realizar este tipo de operaciones se suele expresar los puntos en coordenadas homogéneas y describir las transformaciones como productos de matrices de manera que se pueden agrupar para obtener una única matriz capaz de representar varias operaciones simultáneamente.

2.2.9.3. *Filtrado de imágenes*

Los filtros son transformaciones de intensidad en las cuales el valor de la salida es función de la aplicación de una operación que involucra a los píxeles de su entorno cercano o vecindad. Por tanto, la transformación de la imagen se produce por la combinación de píxeles, en lugar de realizar una operación píxel a píxel.

La implementación de estos se realiza mediante la operación de convolución entre la imagen y una máscara de filtrado (una matriz mucho menor a la imagen) la cual define a partir de sus coeficientes, el tipo de filtro y el efecto que tendrá sobre la imagen. Por esta razón, se les conoce a estas operaciones con el nombre genérico de filtros de

convolución. Los filtros pueden ser usados para la eliminación de ruido, suavizado (pasa bajo), realce de zonas borrosas (pasa alto) e incluso para la detección de la presencia de bordes en los objetos que componen la imagen (operadores de Prewit, Sobel, Laplaciano, entre otros). Las máscaras de convolución pueden describir filtros lineales o no-lineales. La Tabla muestra un resumen de algunos tipos de filtros de uso común en el procesamiento de imágenes.

Tabla

Filtros de uso frecuente en procesamiento de imágenes

Tipo de filtro	Efecto de su aplicación	Máscaras Representativas
Pasa bajo	Suavizado por promedio de píxeles adyacentes. Reduce el ruido en la imagen.	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
	Puede hacer borrosos los bordes de los objetos.	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$
Realce (<i>sharpen</i>)	Detectan y resaltan detalles de la imagen. Mejora la nitidez.	$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
Sobel	Obtiene el gradiente direccional en las direcciones vertical u horizontal para detectar bordes.	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
Prewit	Similar al anterior	$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Laplaciano	Acentúa las zonas que tienen gran discontinuidad. Utiliza la segunda derivada en lugar de la primera.	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

2.2.10. Generalidades sobre segmentación de imágenes digitales

La segmentación es uno de los principales pasos en el procesamiento de imágenes y es un proceso mediante el cual la imagen es subdividida en múltiples regiones homogéneas, cada una de las cuales representa algún tipo de información en la forma de color, intensidad o textura con el fin de facilitar su posterior análisis y reconocimiento automático (Vélez y otros, 2003).

Segmentar una imagen consiste en la asignación de un valor único (a cada píxel perteneciente a una determinada región dentro de la imagen con el objetivo de diferenciarlo de los píxeles que pertenecen a diferentes objetos. Las propiedades utilizadas para realizar la segmentación pueden ser varias, por lo que la selección de una técnica particular de su realización así como el nivel de segmentación deseado, se realiza luego de analizar el dominio del problema (Rajeshwar, Priyanka, & Swapna, 2012).

En general, el proceso de la segmentación suele resultar complejo debido, por un lado, a que no se tiene una información adecuada de los objetos a extraer y, por otro, a que en la escena a segmentar aparece normalmente ruido (Vélez y otros , 2003) . Es por esto que el uso de conocimiento sobre el tipo de imagen a segmentar o alguna otra información de alto nivel puede resultar muy útil para conseguir el éxito.

Se han desarrollado muchas técnicas de segmentación, basadas en las diferentes características de los objetos que pueden presentarse en las imágenes como: presencia de bordes, textura o rasgos colorimétricos, siendo un área del conocimiento sumamente extensa. Algunas técnicas de segmentación de uso frecuente en el procesamiento de imágenes, algunas de las cuales han sido aplicadas previamente para la identificación de leucocitos en la sangre, se mencionan a continuación:

- **Segmentación basada en umbralización**

Es un proceso que permite convertir una imagen de niveles de gris o de color en una imagen binaria, de tal forma que los objetos de interés se etiqueten con un valor distinto al de los píxeles del fondo. Es una técnica de segmentación rápida, que tiene un coste computacional bajo y que puede ser realizada en tiempo real (Vélez y otros, 2013). Su enfoque es simple pero potente, permitiendo separar los objetos de interés del fondo de la imagen mediante la existencia de un nivel de intensidad umbral que sirve como criterio de asignación de pertenencia de los píxeles. En su versión más

simple, conocida como umbralización fija o global, se requiere definir un único valor umbral que aprovecha la diferencia de contraste entre los objetos presentes para realizar la segmentación. Este criterio es aplicable cuando el histograma es del tipo bimodal, es decir con dos lóbulos claramente diferenciados.

En casos más complejos, el nivel de umbral no puede ser fijo debido a las características propias de la imagen, requiriéndose del uso de múltiples valores para segmentar objetos de características distintas. Esto se conoce como umbralización adaptativa. En todo caso, la selección del valor óptimo de umbral es un problema asociado a todas las versiones posibles de la técnica y para su elección se suele recurrir a métodos que involucran el análisis estadístico del histograma de la imagen, siendo el algoritmo de Otsu (Otsu, 1979) la solución clásica por excelencia. Sezgin y Sankur (2004), presentan un extenso análisis comparativo de 40 métodos de umbralización de imágenes, describiendo brevemente la idea principal que sustenta al algoritmo, categorizándolos de acuerdo al tipo de información utilizada y resumiendo las ecuaciones empleadas para calcular los valores umbrales.

Este tipo de técnicas, basadas en el análisis del histograma para la determinación de umbrales, se puede emplear para la localización de puntos iniciales o semillas usadas como punto de partida de otros algoritmos (por ejemplo de agrupamiento o de crecimiento de regiones) o como una segmentación base que sea posteriormente refinada mediante una técnica más compleja.

- **Segmentación basada en crecimiento de regiones**

Agrupar un conjunto de técnicas que utilizando criterios de similitud y proximidad entre los píxeles de la imagen. La homogeneidad (o falta de ella) entre elementos adyacentes es el criterio utilizado para unir (o dividir) regiones de la imagen. Dicha homogeneidad se puede definir a partir de criterios como: el nivel de gris, el color, la forma, entre otras. Según explica Vélez y otros (2003), las técnicas basadas en la

unión o crecimiento de regiones parten de píxeles iniciales (semillas) a partir de los cuales la región crece por agrupamiento de píxeles con características similares. Se debe definir criterios que permiten la agregación de los píxeles a la región. Los métodos de división de regiones operan de inversa, partiendo de una única región conformada por toda la imagen. Si dicha región no satisface el criterio de homogeneidad establecido, la región inicial se divide, de manera secuencial, en subregiones de las que se estudia su homogeneidad y se repite el proceso hasta conseguir sub regiones constituidas por píxeles de características homogéneas.

- **Segmentación basada en agrupamiento**

El termino agrupamiento o *clustering* define el proceso mediante el cual un conjunto de datos es dividido en conjuntos (clústeres) o clases que comparten rasgos semejantes. Son métodos no supervisados de manera que utilizan los propios datos de origen disponibles para caracterizar las propiedades que distinguen a cada clase y deben recorrer iterativamente los datos hasta optimizar las regiones que conforman cada conjunto. Para determinar la pertenencia de un determinado elemento, los algoritmos de agrupamiento utilizan métricas de distancia como medidas de similitud, estas miden la diferencia entre variables en el espacio de características que describen al objeto. Los criterios usados como características para el agrupamiento dependen completamente de la aplicación. Tienen aplicación tanto en la segmentación como en la clasificación de imágenes.

Desde sus inicios la segmentación de imágenes ha estado íntimamente vinculada con el agrupamiento perceptual, mediante el cual un conjunto de puntos dentro de la imagen puede catalogarse como pertenecientes a una misma categoría. Es relativamente fácil tomar un algoritmo de agrupamiento cualquiera y a partir de allí construir un segmentador de imágenes y mucha de la literatura del área está relacionada con ello (Forsyth & Ponce, 2011). De esta forma, resulta intuitivo y

natural pensar en el uso de algoritmos de agrupamiento para definir conjuntos de píxeles que compartan similitudes y por tanto pertenezcan o formen parte de un mismo objeto. Los píxeles se pueden agrupar debido a que tengan el mismo color, textura y/o por su cercanía

La mayor dificultad al utilizar métodos de agrupamiento en la segmentación de imágenes puede hallarse en la necesidad de iterar varias veces a través de todos los píxeles, lo cual puede resultar costoso computacionalmente. También ocurre que aunque el algoritmo converge a una solución, no existe garantía de que ésta sea óptima.

El algoritmo *k*-medias (*k-means*) constituye uno de los más empleados y representativos del área. En este se sub-divide una imagen en *k* regiones, donde *k* es un parámetro debe ser especificado por el usuario. Típicamente al agrupamiento resultante es dependiente de los valores o intensidades de los píxeles pero no de su localización dentro de la imagen. De forma breve puede resumirse su aplicación en tres pasos:

1. Inicialización de los centroides de cada conjunto: en cada uno de los *k* clústeres se seleccionan de forma manual, aleatoria o usando alguna heurística valores iniciales que tentativamente representan al conjunto.
2. Asignación de los píxeles: en el espacio de características se calcula la distancia entre cada píxel y la media del conjunto, este valor permite asignarlo al clúster a cuyo centroide sea más cercano. Esto se realiza para todos los píxeles que componen la imagen.
3. Recalculo de parámetros: se calculan nuevamente los centroides de cada conjunto a partir los píxeles asignados a ellos. Los pasos 2 y 3 se repiten hasta alcanzar la convergencia, lo cual ocurre cuando ningún píxel cambia de un conjunto al otro.

Otros algoritmos que pertenecen a esta categoría son *Expectation-Maximization* (EM) y *Mean-Shift Clustering*. El primero de ellos (EM) guarda similitudes importantes con k-medias, en el sentido de que requiere la indicación externa del número de conjuntos (k) y comprende los mismos tres pasos: inicialización de conjuntos, asignación de los píxeles a los clústeres y recalcado de parámetros, repitiendo los últimos dos de forma iterativa. Sin embargo, en el algoritmo *Expectation-Maximization* los conjuntos son representados por combinaciones de distribuciones probabilísticas (generalmente con distribución normal) y no solo por un valor medio. Por otra parte, la asignación de los píxeles también es probabilística, de modo que, cada uno no es simplemente asociado a una clase, sino que se puede obtener la probabilidad de pertenencia a cada uno de los conjuntos finales (*soft classifier*).

EM se fundamenta en que existe un modelo analítico para describir los datos y se asume como conocida la forma funcional del modelo pero se desconocen los valores de los parámetros que lo caracterizan. El algoritmo busca entonces estimar la máxima verosimilitud (*maximum likelihood*) para dichos parámetros. Esto lo hace desde la premisa de que se tiene un conjunto incompleto de datos en un espacio multidimensional por lo tanto es capaz de manejar la información faltante y lograr una estimación adecuada tanto para los describir los datos existentes como para predecir sobre los faltantes.

Tanto k-medias como EM requieren la especificación externa del número de conjuntos deseado, y mientras esto es adecuado para ciertos casos o tipos de problemas, es frecuente que dicha cantidad sea dependiente de la propia imagen y no un valor predefinido, por lo que los algoritmos de agrupamiento que pueden determinar automáticamente la cantidad de clústeres son más útiles para la segmentación de imágenes arbitrarias (Liu, Soran, Petrie, & Shapiro, 2011). Esto ocurre en el algoritmo *mean-shift*, el cual es una técnica de análisis no paramétrico que permite localizar los máximos de la función de densidad de los datos en el espacio

de características. Esta función de densidad se puede considerar como una estimación empírica de la función densidad probabilística asociada al modelo (Comaniciu & Meer, 2002). Por lo tanto, las áreas de mayor densidad de datos corresponden a los máximos locales de la función de densidad probabilística, esto es, las modas y en torno a éstas se establece la ubicación de los clústeres.

Esto se hace empleando el método de la ventana de Parzen como estimador de la densidad de probabilidad en un kernel (K) centrado una ventana o ancho de banda h . Para cada punto (x) del espacio de características, se calcula la media de los elementos que caen dentro de área definida bajo el kernel (vecindad definida por el tamaño ventana) y se desplaza al punto (x) hacia la media de manera iterativa hasta lograr la convergencia. Este desplazamiento lo hace en la dirección del vector de *mean-shift*, el cual es definido como lo indica la ecuación (1), de manera que en cada repetición el valor de x se actualiza a $x = mh(x)$.

$$mh(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) \cdot x_i}{\sum_{x_i \in N(x)} K(x_i - x)} - x \quad (1)$$

Donde:

$N(x)$ = vecindad de puntos en torno a x (definido por el tamaño de la ventana h)

K = función kernel (generalmente Gaussiano)

El desplazamiento del punto se realiza entonces en la dirección ascendente del gradiente hasta alcanzar un punto en el cual este se hace cero (o inferior a un umbral pre-definido), es decir, en la ubicación del máximo local más cercano.

La principal ventaja de la técnica es que no presupone ni el número, ni la forma de los conjuntos de datos permitiendo analizar espacios de rasgos más complejos con clústeres de forma y cantidad arbitraria.

- **Segmentación basada teoría de grafos**

Estas técnicas de aparición relativamente reciente han generado mucho interés en los investigadores del área de visión por computadora. Parten de la idea general de

representar la imagen en forma de un grafo ponderado no orientado donde cada píxel es asociado a un nodo o vértice y los pesos de las aristas representan la similaridad entre ellos. Cuando dos píxeles adyacentes son similares entre sí el peso de la arista que conecta sus nodos respectivos será un valor grande; mientras que, si los píxeles son diferentes, será pequeño. Esta similaridad puede ser medida en función de diferentes características ya sea intensidad, color o textura, entre otras. El proceso de segmentación se lleva a cabo típicamente cortando el grafo en las aristas de valores relativamente bajos y manteniendo juntos los elementos unidos por aristas con pesos elevados (Forsyth & Ponce, 2011). Por lo tanto, los vértices que comparten características semejantes forman componentes (regiones) unidas relativamente homogéneas al ser separados de aquellos que son disímiles.

A partir de estos conceptos se han desarrollados diferentes algoritmos para la segmentación de imágenes, los cuales emplean múltiples criterios para efectuar esos cortes usando el peso de las aristas como indicador del costo de separar los conjuntos.

De esta forma un grafo $G = (V, E)$, donde V representa los nodos y E las aristas, puede ser particionado en dos conjuntos separados A y B simplemente removiendo las aristas que los conectan. El grado de disimilitud o corte (*cut*) entre las partes separadas puede ser calculado como la suma total de los pesos de las aristas que han sido removidas, esto es $cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$; y una bipartición óptima del grafo se obtiene si este valor es minimizado (Shi & Malik, 2000).

El valor mínimo del corte (*minimum cut*) de un grafo puede ser eficientemente calculado y fue utilizado criterio en la década de 1990 por Wu y Leahy (1993) para plantear su algoritmo de segmentación. Estos autores propusieron dividir un grafo en k sub-grafos utilizando de manera recursiva el cálculo de los cortes mínimos que bisecan los segmentos existentes. Su propuesta aunque intuitiva para ilustrar la idea general, demostró tener un sesgo hacia encontrar componentes pequeños aislados

(Peng, Zhang, & Zhang, 2013), lo cual era predecible dada la definición del costo del corte empleada.

Esta situación fue considerada por Shi y Malik (2000) en el planteamiento del conocido algoritmo de cortes normalizados (*NCut*) en el cual se prevé la definición de condiciones para explícitamente lograr una partición de los conjunto más equilibrada, tratando de que cada uno de ellos sea lo razonablemente grande. Para ello, se mide tanto la disimilitud entre los diferentes conjuntos o regiones como la similaridad total dentro de los elementos pertenecientes a cada conjunto. Los autores definieron entonces al costo del corte normalizado *NCut* como la medida a ser minimizada al separar los sub-conjunto *A* y *B* (ambos parte del conjunto *V*) según la ecuación ().

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} - \frac{cut(A, B)}{assoc(B, V)} \quad ()$$

Donde:

$cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$ (pesos de las aristas removidas para separar A y B)

$assoc(A, V) = \sum_{u \in A, v \in V} w(u, v)$ (conexión de los nodos de A a todos los nodos en V)

$assoc(B, V) = \sum_{u \in B, v \in V} w(u, v)$ (conexión de los nodos de B a todos los nodos en V)

De esta forma, los cortes que dividen pequeñas componentes aisladas no tienen valores *Ncut* bajos ya que justamente tienen pocas conexiones al resto del conjunto *V*. Dicho de otra forma, el algoritmo debe buscar los valores mínimos del numerador (el corte propiamente dicho) tratando de maximizar el denominador (relacionado con la cantidad de elementos existente en área particionada). La minimización de la expresión resulta ser un problema computacional complejo, por lo cual los autores desarrollaron una solución aproximada mediante la discretización del autovector con el segundo menor autovalor del sistema descrito por la ecuación ().

$$(D - W) x = \lambda D x \quad ()$$

Donde:

$D = \text{Matriz diagonal donde } D_{u,u} = \sum_{v \in V} w(u, v)$ (*degree matrix* - conexión total de cada nodo)

$W = \text{Matriz de adyacencia donde } W_{u,v} = w(u, v)$

Nota. Se muestra la solución para el caso de partición en 2 conjuntos, en Shi & Malik (2000) se presenta la generalización para *k* conjuntos

La discretización se hace comparando cada componente del autovector con un valor umbral, el cual se busca de forma iterativa minimizando al *NCut* resultante en cada repetición. Usando el autovector discretizado se particiona el grafo. En las secciones resultantes se ejecuta el algoritmo recursivamente.

Durante el presente trabajo de investigación se evaluó la posibilidad de utilizar el algoritmo *NCut* empleando para ello el código fuente publicado por sus autores y disponible en la URL <http://www.cis.upenn.edu/~jshi/software/>. La idea fue descartada al observar que el tiempo requerido para procesar una imagen 160x160 píxeles era muy elevado, siendo mayor a 9.65 segundos.

El árbol de expansión mínima (*minimal spanning tree*), MST por sus siglas en inglés, es un concepto importante en la teoría de grafos. Un árbol de expansión T de un grafo G , está conformado por todos los vértices de G y un sub-conjunto de sus aristas, seleccionadas de manera que todos vértices queden conectados pero no existan ciclos (camino cerrados). El MST se define entonces como aquel con los menores pesos de todos los árboles de expansión posibles.

Felzenszwalb y Huttenlocher (2004) presentaron un método eficiente de segmentación basado en el algoritmo de Kruskal para cálculo MST, donde, las aristas del grafo son inicialmente ordenadas de forma no decreciente en función de sus pesos y posteriormente la segmentación se realiza por medio de un algoritmo de fusión regiones. Propusieron evaluar la existencia de una frontera entre dos componentes en una segmentación (dos regiones de la imagen) haciendo uso tanto de la diferencia entre las componentes (externa) como de la disimilitud dentro de cada sub-grafo (interna). En función de la comparación entre estas diferencias interna y externa se define si los nodos involucrados deben estar unidos o separados, de forma que la salida del algoritmo es un bosque compuesto por múltiples árboles mínimos disjuntos, en el cual cada árbol representa una sección segmentada.

Según Peng y otros (2013), la clave de este algoritmo es la umbralización adaptativa empleada ya que, en lugar de un valor constante k , el umbral es variable y está definido en relación al tamaño de los clústeres como $\tau = \frac{k}{|C|}$. Esto permite a dos componentes disjuntas (C_1, C_2) fusionarse si el menor enlace entre ellas ($|e_t|$) es menor que el máximo peso en cualquiera de las componentes del MST ($Int(C)$) sumado al valor umbral (τ). La definición formal del criterio de fusión es dada por la ecuación (1).

$$|e_t| \leq \min\left(Int(C_1) + \frac{K}{|C_1|}, Int(C_2) + \frac{K}{|C_2|}\right) \quad (1)$$

Donde:

$$Int(C) = \max_{u \in MST(C, E)} (w(u)) \quad (\text{Distancia interna mínima dentro del componente } C)$$

$$|e_t| = \min_{v_i \in C_1, v_j \in C_2} (w(v_i, v_j)) \quad (\text{Arista con el menor peso que conecta } C_1 \text{ con } C_2)$$

$|C|$ = Tamaño del conjunto C . K = Constante de ajuste de escala

Nota. Se considera que se ha representado la imagen como un grafo $G = (V, E)$

Así, el algoritmo descrito por Felzenszwalb y Huttenlocher (2004) utiliza como entrada un grafo ponderado unidireccional $G = (V, E)$, con n vértices y m aristas, y su salida es una segmentación de V en las componentes $S = (C_1, C_2, \dots, C_r)$ la cual se obtiene mediante el procedimiento descrito a continuación:

1. Ordenar E en $\pi = (o_1, o_2, \dots, o_m)$ de forma no decreciente en función de los pesos de cada arista.
2. Iniciar el algoritmo con una segmentación S^0 , donde cada vértice v_i está en su propia componente. Es decir, inicialmente cada píxel es considerado una región aislada.
3. Construir S^q a partir de S^{q-1} de la forma que se indica a continuación. Siendo C_i^{q-1} y C_j^{q-1} las componentes de S^{q-1} que contienen los nodos v_i y v_j , respectivamente. Si $C_i^{q-1} \neq C_j^{q-1}$ y $w(o_q) \leq \min\left(Int(C_i^{q-1}) + \frac{K}{|C_i^{q-1}|}, Int(C_j^{q-1}) + \frac{K}{|C_j^{q-1}|}\right)$, entonces S^q se obtiene a partir de S^{q-1} mediante la fusión de C_i^{q-1} y C_j^{q-1} ; de lo

contrario $S^q = S^{q-1}$. Es decir, si v_i y v_j se encuentran en componentes disjuntas de S^{q-1} , y $w(o_q)$ es pequeño comparado con la diferencia interna entre ellas, entonces las componentes se fusionan.

4. El paso anterior se repite para $q = 1, 2, \dots, m$.
5. La salida será dada por $S = S^m$.

Peng y otros (2013) ofrecen información complementaria sobre esta y otras técnicas de segmentación basadas en teoría de grafos.

• Segmentación usando redes neuronales y máquinas de soporte vectorial

Las redes neuronales artificiales (*Artificial Neural Networks –ANN*) tienen su origen e inspiración en las neuronas biológicas, de manera que estas imitan el hardware del sistema nervioso, con la intención de construir sistemas de procesamiento de información paralelos, distribuidos que puedan tener cierta inteligencia (Martín del Brío & Sanz, 2002). Se componen de muchos elementos de procesamiento básico (PE) o neuronas que son elementos de cálculo simples, los cuales, determinan el valor de una única salida, a partir de un conjunto de entradas que provienen del exterior (ver modelo representativo en la Figura). Cada neurona o nodo de una ANN es capaz de llevar a cabo cálculos elementales y el aprendizaje se alcanza a través de la adaptación de pesos asignados a las conexiones entre nodos. El modelo general de una neurona artificial, incluye los siguientes elementos:

- Conjunto de entradas, x_j .
- Pesos sinápticos de la neurona i , w_{ij} . Representan la intensidad de la interacción entre cada entrada x_j y el nodo i .
- Regla de propagación, σ . Proporciona el valor del potencial postsináptico de la neurona i en función de las entradas y de los pesos. Matemáticamente se puede escribir como $h_i = \sigma(w_{ij}, x_j)$. Típicamente corresponde a una combinación

lineal de los pesos ($h_i = \sum_j w_{ij} \cdot x_i$) o su distancia euclídea respecto al vector de

$$\text{entradas } h_i = \sqrt{\sum_j (x_i - w_{ij})^2}.$$

- Función de activación, f_i . Determina el estado de activación de la neurona a partir del potencial post-sináptico (h_i) y un valor umbral θ (también llamado sesgo o *bias*) $a_i = f_i(h_i - \theta_i)$. Algunas veces también involucra al estado anterior de la neurona $a_i(t) = f(a_i(t-1), h_i(t))$. Habitualmente la función de activación es de tipo sigmoidea, gaussiana o de rectificación (ReLU). La Figura muestra algunas funciones de activación de uso frecuente.
- Función de salida, F_i . Proporciona la salida real de la neurona y_i y se obtiene a partir del estado de activación. Muy frecuentemente es la función identidad modo que $y_i = a_i$.

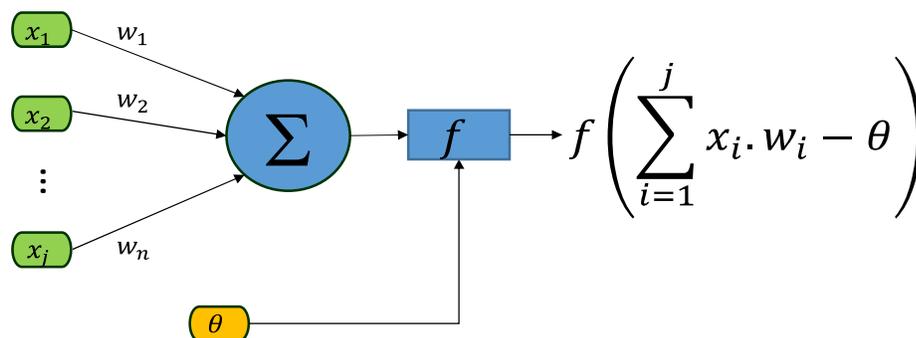


Figura . Ejemplo de una neurona artificial. La salida de la función f indicará el estado de activación.

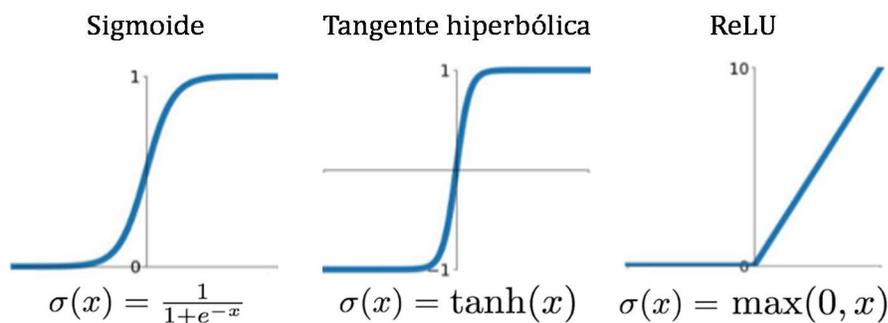


Figura . Funciones de activación comúnmente usadas en las ANN.

Un conjunto de neuronas agrupadas conforman una capa, que puede ser de tres tipos:

- Entrada: son las unidades de procesamiento se comunican con el exterior y reciben información proveniente del entorno.
- Salida: envían respuestas hacia el exterior de manera similar a las neuronas motoras del cuerpo humano.
- Ocultas: no se conectan al exterior y su función es brindarle a la red neuronal, los grados de libertad necesarios para adaptarse y resolver los problemas más complejos. Cada capa oculta puede tener diferente cantidad de unidades de procesamiento. Una red neuronal típicamente tiene una o más capas ocultas.

Existen múltiples criterios para clasificar las redes neuronales siendo común hacerlo en función de: (a) la cantidad de capas que posea (monocapa o multicapa), (b) el flujo de los datos a través de la red, pudiendo estas ser unidireccionales (*feedforward*) o recurrentes (*feedback*), (c) el paradigma de entrenamiento empleado (supervisado, no-supervisado, híbrido o por refuerzo), (d) la forma de asociación de la información (memoria hetero-asociativa o auto-asociativa), o la cantidad de capas ocultas (redes superficiales – *shallow* o profundas – *deep*), entre otros. La Figura muestra una red neuronal típica con una arquitectura unidireccional de tres capas.

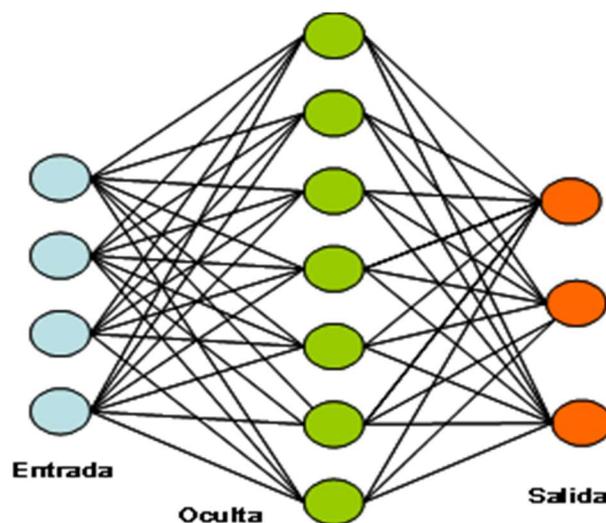


Figura . Distribución de las capas de neuronas en una red neuronal típica

En función del estudio el tipo de neurona empleada (selección de las funciones de propagación y activación), la arquitectura o topología de interconexión de las unidades de procesamiento y algoritmo de aprendizaje empleado se han definido una gran variedad de modelos o tipos diferentes de redes neuronales, algunos de las más conocidas son: el perceptrón multicapa (MLP), redes de base radial (RBF), mapas auto-organizativos (SOM) de Kohonen, redes de Hopfield y las redes convolucionales, las cuales que fueron el tipo empleado para el presente trabajo. Un trabajo gráfico de fácil interpretación que permite observar la gran variedad de redes neuronales existentes en la actualidad fue realizado y publicado en 2016 por Fjodor Van Veen en el sitio web de la organización “the Asimov Institute”. La ilustración denominada “The Neural Network Zoo” se reproduce en la Figura . El sitio Web (<http://www.asimovinstitute.org/neural-network-zoo/>) permite además acceder a los artículos originales de cada autor siendo un recurso valioso para el lector interesado en la evolución histórica de las redes neuronales.

Diferentes topologías de redes neuronales artificiales han sido usadas por más de 25 años para el procesamiento de imágenes. Al respecto, el trabajo de Egmont-Petersen y otros (2002) analiza y clasifica más de 200 aplicaciones procesamiento de imágenes por medio de ANN de acuerdo a dos criterios: (1) tipo de tarea realizada (pre-procesamiento/filtrado, reducción de datos/ extracción de características, segmentación, detección y reconocimiento de objetos, comprensión semántica de imágenes y optimización) y (2) nivel de abstracción de los datos de entrada procesados por la red neuronal (píxel, rasgos locales, estructuras básicas, conjuntos de objetos y caracterización de escenas. De acuerdo a sus hallazgos, el mayor uso de las ANN fue en el reconocimiento y detección de objetos (41.28%). Sin embargo, en el 25.53% de los casos la red fue utilizada para la segmentación usando como entrada directamente los píxeles crudos de la imagen o rasgos locales de bajo nivel en el 65% y 31.67% de las aplicaciones, respectivamente. Esto indica que el uso más frecuente de las ANN como herramienta de segmentación es utilizarles para clasificar

directamente los píxeles, sin embargo, ello usualmente resulta en un espacio de entrada de dimensiones muy grandes lo cual involucra un gran costo computacional.

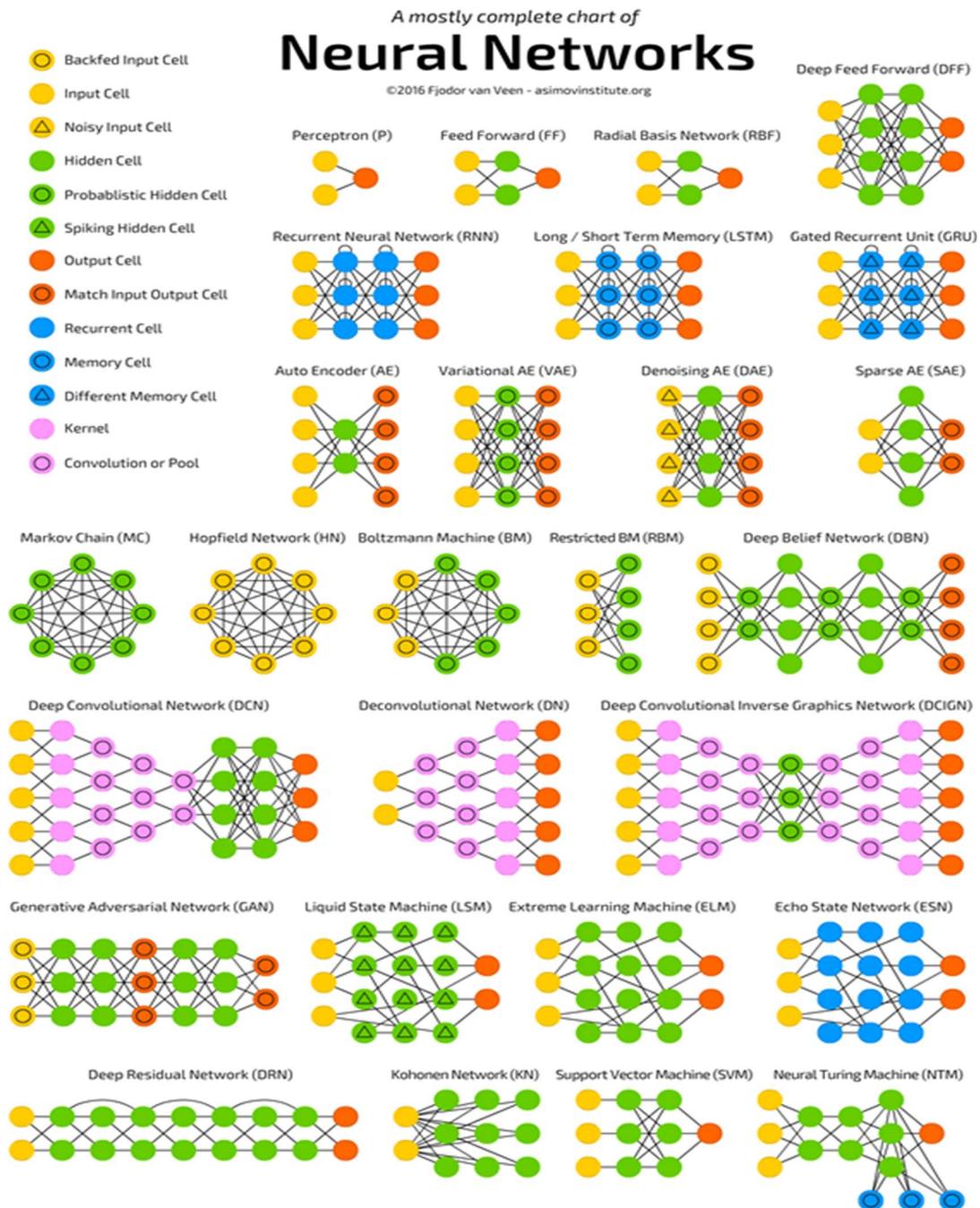


Figura . Representación de diferentes topologías de redes neuronales. Recuperado de “The Neural Network Zoo”, por the Asimov Institute (2016) <https://www.asimovinstitute.org/>. **Nota.** El autor incluye una aclaratoria en la Web de origen señalando sus razones para incluir las SVM en su representación.

Otro modelo de aprendizaje automático ampliamente empleado en la clasificación y segmentación de imágenes son las máquinas de vectores de soporte (*Support Vector Machines –SVM*) propuestas por Vapnik y Cortes (1995). Se fundamentan en la teoría del aprendizaje estadístico y han constituido una de las herramientas más versátiles para el *machine learning*. Originalmente fueron concebidas para resolver problemas de clasificación binaria pero dadas sus características han sido extendidas para aplicarse en muchos otros tipos de problemas como regresión, agrupamiento o multi-clasificación. Se han empleado con éxito en aplicaciones muy diversas como el reconocimiento de caracteres, categorización de texto, procesamiento de lenguaje natural, reconocimiento de voz y textos manuscritos, análisis de series temporales, detección de objetos, segmentación y clasificación de imágenes digitales. En esta última área, han sido uno de los algoritmos más empleados en los últimos años.

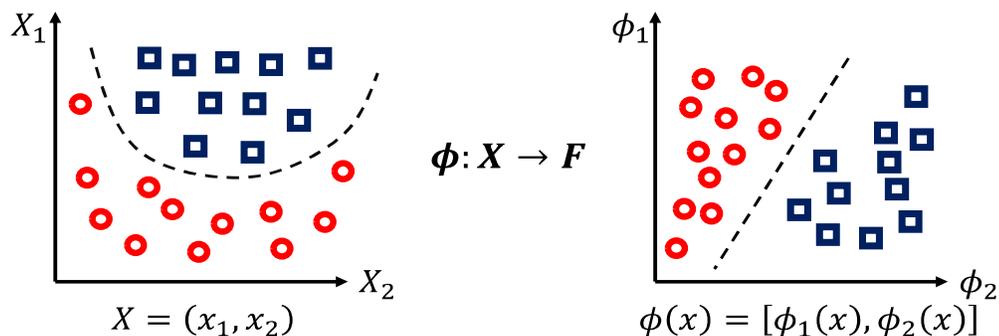


Figura . Proyección de los datos en una SVM. La selección de una función de transformación ϕ adecuada puede lograr que un conjunto no linealmente separable en el espacio de entradas X lo sea en el de características F .

Las SVM pertenecen a la categoría de los clasificadores lineales, puesto que inducen separadores lineales o hiperplanos, ya sea en el espacio original de los datos de entrada (si estos son separables o cuasi-separables) o en un espacio transformado o espacio de características (Carmona, 2016). Justamente es esta la idea fundamental detrás de las SVM, proyectar mediante una función de transformación (*kernel*) los datos no linealmente separables en el espacio de entrada a un espacio de características con una dimensión superior (dimensión de Vapnik-Chervonenkis) en el cual es posible hallar un hiperplano de separación adecuado para resolver el

problema (ver Figura). Las funciones kernel usadas pueden ser de tipo lineal, polinómico, gaussiano o sigmoïdal (Carmona, 2016).

La mayoría de los métodos de aprendizaje supervisado se basan en minimizar el error observado en la respuesta del modelo ante la presencia de los ejemplos o datos de entrenamiento (riesgo empírico). Ello no implica que el modelo pueda generalizar de forma correcta, es decir, que logre una respuesta adecuada al observar tanto los datos de entrenamiento como los datos de prueba (riesgo esperado). En las SVM se busca la minimización del riesgo estructural, este implica la reducción tanto del riesgo empírico como de la capacidad de la función, estando está relacionada con su complejidad y por tanto con la dimensión de Vapnik-Chervonenkis. La idea es entonces seleccionar un hiperplano de separación que sea equidistante de los ejemplos más cercanos de cada clase en el espacio de características para, de esta forma, conseguir un margen de separación máximo en cada conjunto. Este hiperplano de separación máxima logra simultáneamente reducir el error de separación del conjunto de entrenamiento y maximiza la capacidad de generalización del clasificador al evitar en gran medida el sobreajuste, logrando con ello la minimización del riesgo estructural.

El hiperplano de separación óptimo fue definido por Vapnik y Cortes (1995) como “la función de decisión lineal con máximo margen (de separación) entre los vectores de dos clases” y para su construcción únicamente se tiene que considerar una pequeña cantidad de los datos de entrenamiento, aquellos ejemplos que se encuentran justo sobre la frontera definida por este margen de separación máxima, los cuales se denominan vectores de soporte. La Figura ilustra el concepto, en ella se puede observar como pudieran existir infinitos hiperplanos capaces de separar los conjuntos de datos (izquierda) pero solo uno maximiza el margen o distancia entre los puntos de datos más cercanos de los conjuntos (derecha).

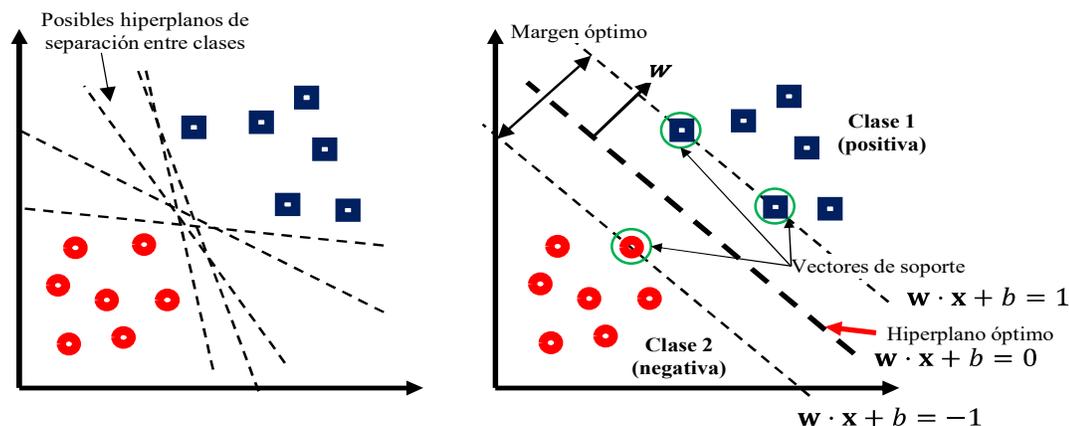


Figura . Hiperplanos de separación en un espacio bidimensional. Izquierda: múltiples hiperplanos no óptimos que pudieran separar los datos. Derecha: Los vectores de soporte señalados definen el máximo margen de separación entre los conjuntos y el hiperplano óptimo.

Los conjuntos de datos mostrados por la Figura corresponden con un caso en el cual todos los ejemplos de entrenamiento (x_i) son perfectamente separables (sin error) mediante el hiperplano definido por $\mathbf{w} \cdot \mathbf{x} + b = 0$, conocido como hiperplano de separación de margen duro, el cual se obtiene minimizando la ecuación () bajo la restricción impuesta por () (Byun & Lee, 2002).

$$\min_{\mathbf{w}, b} \Phi(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2} \quad ()$$

$$\mathbf{y}_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0, \forall_i \quad ()$$

Siendo:

\mathbf{x}_i = conjunto de ejemplos de entrenamiento

$\mathbf{y}_i \in \{-1, +1\}$ = etiquetas de asignación de clase de los elementos \mathbf{x}_i

Sin embargo, en las aplicaciones reales los datos se caracterizan por poseer ejemplos ruidosos y no ser perfecta y linealmente separables. La estrategia para este tipo de problemas es relajar el grado de separabilidad de los datos, permitiendo que haya errores de clasificación en algunos de los ejemplos del conjunto de entrenamiento. Sin embargo, sigue siendo un objetivo el encontrar un hiperplano óptimo para el resto de ejemplos que sí son separables (Carmona, 2016). Esto puede lograrse introduciendo un conjunto de variables denominadas de holgura (ξ_i), las cuales permiten cuantificar de forma implícita la cantidad de ejemplos no-separables (con

error) que se pueden admitir. De esta forma, el hiperplano de separación (ahora denominado de margen blando) queda definido por ()

$$\mathbf{y}_i(\mathbf{x}_i \cdot \mathbf{w} + \mathbf{b}) - \mathbf{1} \geq 1 - \xi_i, \forall_i \quad ()$$

Las variables de holgura con valor cero corresponden a datos separables; entre cero y menor que uno, con datos no-separables pero bien clasificados; y mayor que uno, denotan ejemplos no separables y mal clasificados. La sumatoria de estas define un límite superior para la cantidad de errores de clasificación y la función a optimizar se modifica para considerar su presencia, quedando de acuerdo a lo indicado en la ecuación ().

$$\min_{\mathbf{w}, \mathbf{b}} \Phi(\mathbf{w}, \xi) = \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^n \xi_i \quad ()$$

Donde n corresponde al total de pares de datos y etiquetas del conjunto de entrenamiento y C se denomina parámetro de regularización, siendo un valor elegido por el usuario que controla el compromiso entre el tamaño del margen (grado de sobreajuste del clasificador) y la cantidad de ejemplos clasificados de forma incorrecta. Valores altos de C penalizan la existencia de errores.

En su definición los clasificadores SVM son de tipo binario y por lo tanto sólo pueden manejar dos clases, pero pueden extenderse para ser empleados en aplicaciones multiclase mediante técnicas de descomposición, las cuales interpretan el problema original como pares de problemas binarios más simples de resolver. Cada clasificador afronta posteriormente un problema de forma independiente y las salidas obtenidas se combinan mediante una estrategia de agregación que puede estar basada en el voto simple de la clase asignada, el voto ponderado en función del grado de confianza o mediante una estimación de la probabilidad condicional. Por su parte, las estrategias de descomposición más comúnmente empleadas son de acuerdo a Galar y otros (2011):

- Estrategia uno contra uno (1 vs 1, OVO): un problema de k clases se divide en $\frac{k(k-1)}{2}$ problemas binarios los cuales son afrontados por clasificadores independientes. Cada uno emite un voto a favor de una clase, generalmente acompañado de un índice de confianza. Luego sus salidas son combinadas para predecir la asignación de clase. Esta asignación puede ser simplemente contar los votos de cada clase, determinar la clase con el mayor índice de confianza u otros métodos más avanzados.
- Estrategia uno contra todos (1 vs all, OVA): cada conjunto de datos perteneciente a una clase se compara con el resto (unión de todos los que no pertenecen a ella), es decir que si existen k clases, se tendrán k clasificadores. La asignación de pertenencia de clase corresponderá al clasificador que presente la mayor respuesta positiva.

2.2.11. Segmentación utilizando superpíxeles (*oversegmentation*)

El termino superpíxel fue introducido por Ren & Malik (2003) para definir la agrupación de píxeles en regiones perceptualmente similares como etapa de pre-procesamiento. Su desarrollo se fundamenta en dos razones fundamentales (a) los píxeles no son entidades naturales sino una mera consecuencia de la representación discreta de las imágenes y (b) la cantidad de píxeles presente en una imagen es grande, incluso a resoluciones moderadas, lo cual limita la factibilidad computacional de utilizar muchos algoritmos. Los superpíxeles son entidades más naturales que mantienen la mayoría de la estructura perceptual necesaria para realizar la segmentación en la escala de interés reduciendo notablemente el número de primitivas para las etapas de procesamiento subsecuentes.

Los métodos para generar superpíxeles se categorizan como basados en grafos o en el gradiente ascendente, cada uno con sus propias ventajas e inconvenientes que pueden adaptarse mejor a una aplicación en particular (Achanta, y otros, 2012). Por ejemplo, Ren & Malik (2003) se basaron en *NCut* para realizar la sobre segmentación de las

imágenes generando 200 superpíxeles, pero el algoritmo de Felzenszwalb & Huttenlocher (2004) también puede emplearse a este fin. En general, los superpíxeles deben ser fáciles de calcular, eficientes en el uso de la memoria, simples de utilizar y adherirse bien a las fronteras presentes en la imagen (Achanta, y otros, 2012). También es recomendable poder controlar la cantidad de superpíxeles resultantes y que estos sean compactos, regulares y suaves (Stutz, Hermans, & Leibe, 2016). En la Figura se muestra una comparación entre los superpíxeles generados mediante tres algoritmos diferentes.

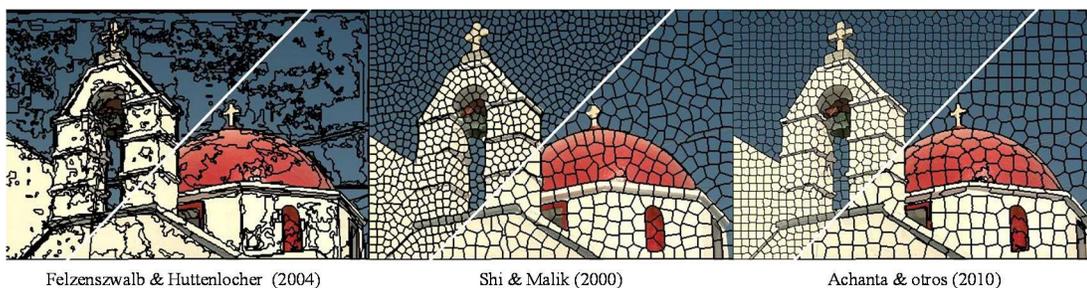


Figura . Comparación visual de superpíxeles generados por diferentes métodos. En la parte inferior de cada imagen indica el método empleado. Recuperado de “SLIC Superpixels”, <https://ivrl.epfl.ch/research-2/research-current/research-superpixels/>.

2.2.11.1. Algoritmo Simple linear iterative clustering (SLIC)

El algoritmo lineal simple de agrupamiento iterativo (SLIC) es un método de generación de superpíxeles rápido, eficiente y que ofrece gran adherencia a las fronteras de la imagen. Constituye una adaptación del algoritmo k-medias para la generación de superpíxeles con dos diferencias importantes:

- El número de cálculos de distancia requeridos en la optimización se reduce drásticamente al limitar el espacio de búsqueda a una región proporcional al tamaño de los superpíxeles y no al de la imagen entera (Figura).
- Una medida de distancia ponderada combina el color y la proximidad espacial, a la vez que proporciona control sobre el tamaño y la compacidad de los superpíxeles.

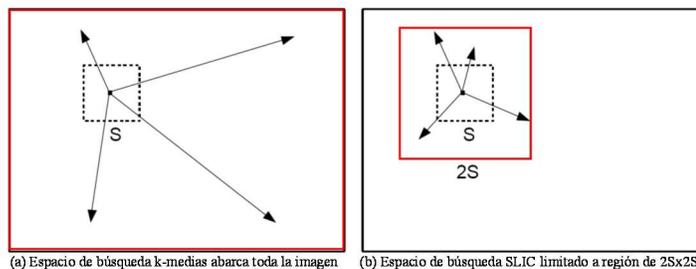


Figura . Reducción del área de búsqueda en torno al centro del superpixel.
Modificado de Achanta y otros, (2012)

SLIC funciona mediante la agrupación de píxeles basada en su similitud de color y proximidad en el plano de la imagen. Esto se realiza en el espacio de 5 dimensiones $[labxy]$, donde $[lab]$ representa al píxel en el modelo de color CIELAB y $[xy]$ su posición en la imagen.

El único parámetro de entrada al algoritmo es k , que indica la cantidad de superpíxeles deseados. Los superpíxeles formados serán de un tamaño similar, dado por N / k para una imagen de N píxeles. El procedimiento de agrupamiento comienza por la creación de una malla o rejilla con k segmentos regulares, espaciados por $S = \sqrt{(N / k)}$ píxeles, en cuyos centros se colocarán las semillas de los clústeres. Con la finalidad de evitar colocar el centroide de un clúster en un píxel correspondiente a un borde de la imagen o sobre un píxel con ruido, estos se mueven a la posición con el menor gradiente en una vecindad de 3×3 respecto a su localización original.

Seguidamente, cada píxel en la imagen es asociado con el centroide del clúster más cercano con cuya área de búsqueda se solape. Esta área de búsqueda está definida como la vecindad de $2S \times 2S$ píxeles alrededor del centro del superpíxel en el plano xy . Luego que todos los píxeles han sido asociados con su respectivo centroide, este se recalcula como el promedio del vector $[labxy]^T$ de todos píxeles pertenecientes a él y se repite el proceso de forma iterativa hasta alcanzar la convergencia. Dicho procedimiento es en esencia igual a ejecutar el algoritmo k-medias restringido al área de búsqueda en torno al centro del superpíxel (ver Figura).

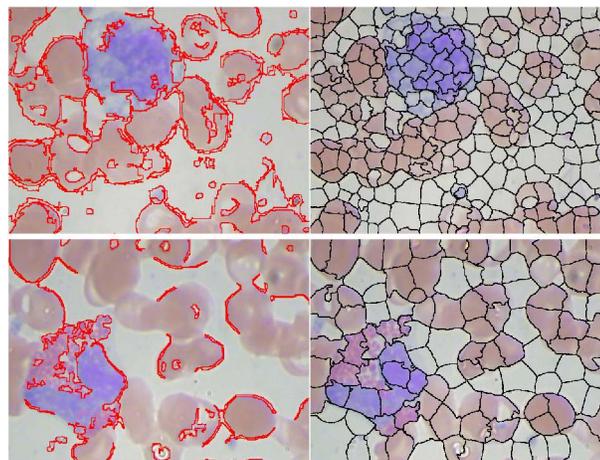
Para medir la cercanía entre un píxel $j [l_j, a_j, b_j, x_j, y_j]^T$ y un centroide de un clúster $C_i = [l_i, a_i, b_i, x_i, y_i]^T$ este método utiliza una definición de distancia ponderada que combina la similitud de color y la proximidad espacial según se muestra en la ecuación (9). En esta se observa que la distancia en el plano xy ha sido normalizada por el espaciado entre los clústeres S . Esto se hace porque el tamaño del superpíxel es dependiente de la imagen por tanto, si esta es muy grande los valores de distancia espacial tendrán mayor importancia relativa que la proximidad del color, resultando en superpíxeles que no respetan las fronteras entre regiones. La definición también incluye un parámetro m , el cual controla cuan compactos son los superpíxeles generados. Para valores pequeños de m , la métrica de distancia enfatiza en la adherencia a las fronteras de la imagen pero los conjuntos resultan menos regulares en su forma y tamaño, mientras que un mayor valor de m se traducen en superpíxeles más compactos.

$$D_s = d_{lab} + \frac{m}{S} d_{xy} \quad (9)$$

Donde:

$$d_{lab} = \sqrt{(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2} \quad d_{xy} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Achanta y otros (2012) ofrece una comparación entre diferentes métodos de sobre-segmentación mediante superpíxeles en el cual se destacan las bondades del algoritmo *Simple linear iterative clustering* (SLIC) usado en el presente proyecto.



Felzenszwalb & Huttenlocher (2004)

Achanta & otros (2010)

Figura . Comparación de algoritmos de superpíxeles en frotis sanguíneo

En la Figura se comparan los superpíxeles obtenidos al procesar mediante dos métodos distintos imágenes frotis de sangre periférica pertenecientes a una de las bases de datos usadas. Se observa una mejor adherencia y una representación más compacta del leucocito al emplear el método SLIC.

2.2.12. Clasificación de imágenes

Los algoritmos de clasificación de utilizan para distinguir entre objetos diferentes de un conjunto predefinido llamado universo de trabajo. Normalmente, el universo de trabajo se considera dividido en una colección K de clases ($\alpha_1, \alpha_2 \dots \alpha_K$), perteneciendo los diferentes objetos a algunas de estas clases (Vélez y otros, 2003). Para realizar el reconocimiento automático de los objetos se hace una transformación que convierte un objeto del universo de trabajo en un vector X cuyas N componentes se llaman características discriminantes o rasgos característicos. Estas características deben permitir discriminar a cual clase puede pertenecer cualquier objeto del universo de trabajo. La determinación de las N características discriminantes es un proceso nada trivial que suele requerir del uso de la imaginación y análisis exhaustivo de los elementos que componen cada clase. En general, suelen usarse características como los momentos de los objetos a reconocer o alguna transformación de los mismos, rasgos estadísticos o cualquier información asociada a los objetos que pueda ser obtenida mediante algún procedimiento algorítmico.

En el reconocimiento de patrones existen dos paradigmas claramente diferentes: supervisado y no supervisado. En el aprendizaje supervisado se dispone de un conjunto de datos junto con sus etiquetas, indicando a qué clase pertenece cada objeto. Se busca entonces conseguir un modelo capaz de etiquetar automáticamente nuevos datos que no se hayan empleado en el ajuste del modelo de clasificación (proceso de entrenamiento).

En las técnicas de aprendizaje no supervisado sólo se dispone de los datos sin las etiquetas de clase. Tratan de modelar los datos en sí mismos, lo que se traduce

normalmente en descubrir grupos por medio de un agrupamiento de elementos con características similares (*clustering*), es decir, determinar si hay ejemplos similares en el conjunto de datos y qué características hacen que estos datos sean similares dentro del grupo y diferentes del resto (García-Ordás, Alaiz-Rodríguez, & Alegre, 2016).

En la revisión realizada como parte del presente trabajo, se encontró que en las aplicaciones de identificación de leucocitos y otras células sanguíneas en imágenes digitales, se han empleado diferentes algoritmos como se muestra en la Tabla .

Tabla

Algoritmos de clasificación usados para la identificación de leucocitos

Algoritmo o método	Autores	Exactitud media
K-vecinos más cercanos (KNN)	Martínez-Castro et al. (2014)	89.8%
	Ramírez-Cortes et al. (2010)	71%
	Lisicki (2014)	53%
	Ramoser et al. (2005)	86.4%
	Ramírez-Cortes et al. (2010)	71.6%
	Rezatofghi y Soltanian-Zadeh (2011)	96%
Máquinas de vectores de soporte (SVM)	Sarrafzadeh et al. (2014)	97.73%
	Habibzadeh et al. (2012)	95.2%
	Rodrigues et al. (2008)	85.44%
	Putzu y Di Ruberto (2013)	92%
	Lisicki (2014)	45%
	Rezatofghi & Soltanian-Zadeh (2011)	90.0%
Redes Neuronales Artificiales (MLP)	Ramírez-Cortes et al. (2010)	87.6%
	Rodrigues et al. (2008)	80.34%
	Habibzadeh et al. (2012)	85%
Redes neuronales convolucionales (CNN)	Habibzadeh et al. (2013)	84.7%
	Lisicki (2014)	88%
	Jiang et al. (2018)	83%
	Shahin et al. (2019)	96.1%
Clasificador Bayesiano ingenuo	Mathur et al. (2013)	92.72%
	Prinyakupt & Pluempitiwiriwawej (2015)	94.1%
Clasificador de distancia euclídea mínima	Ramírez-Cortes et al. (2010)	77.7%

De forma complementaria, en la Figura se reproduce una tabla comparativa extraída del trabajo de Saraswat & Arya (2014) en la cual se muestra el método usado, exactitud alcanzada y rasgos tanto colorimétricos, geométricos y de textura empleados para clasificar leucocitos.

Classifier	Extracted features	Accuracy	Year	Reference
Bayesian classifier	Eccentricity, compactness, area-ratio, number of nucleus lobes, average value of each color components, energy, entropy, correlation, coarseness, and busyness	82.3%	2003	Sinha and Ramakrishnan (2003)
	Area of the nucleus, location of its pattern spectrum's peak, first, and second granulometric moments of the pattern spectrum	77%	2007	Theera-Umpon and Dhompongsa (2007)
	Area, perimeter, compactness, eccentricity, orientation, solidity, form factor, and roundness	83.2%	2010	Ghosh et al. (2010)
k-Nearest neighbor	Eccentricity, compactness, area-ratio, number of nucleus lobes, average value of each color components, energy, entropy, correlation, coarseness, and busyness	70.6%	2003	Sinha and Ramakrishnan (2003)
	Area, perimeter, convex area, solidity, major axis length, orientation, filled area, eccentricity, the ratio between the cell and nucleus areas, rectangularity, circularity, number of lobes, mean intensity of the cytoplasm	86%	2004	Piuri and Scotti (2004)
	Shape, presence of nuclei, perimeter, area, nucleus/cytoplasm ratio, mean, variance, and standard deviation	95%	2011	Rajendran et al. (2011)
Neural network	Eccentricity, compactness, area-ratio, number of nucleus lobes, average value of each color components, energy, entropy, correlation, coarseness and busyness	94.1%	2003	Sinha and Ramakrishnan (2003)
	Area, perimeter, convex area, solidity, major axis length, orientation, filled area, eccentricity, the ratio between the cell and nucleus areas, rectangularity, circularity, number of lobes, mean intensity of the cytoplasm	92%	2004	Piuri and Scotti (2004)
	Area of the nucleus, location of its pattern spectrum's peak, first and second granulometric moments of the pattern spectrum	77%	2007	Theera-Umpon and Dhompongsa (2007)
	Nucleus area, perimeter, number of the separated parts of nucleus, mean, variance of the nucleus boundaries, roundness of nucleus, averaged nucleus color, fourteen features based on co-occurrence matrix, and two features based on Local Binary Pattern	96%	2011	Rezatofghi and Soltanian-Zadeh (2011)
Support vector machine	Eccentricity, compactness, area-ratio, number of nucleus lobes, average value of each color components, energy, entropy, correlation, coarseness and busyness	94.1%	2003	Sinha and Ramakrishnan (2003)
	Contrast, correlation, cluster prominence, cluster shade, dissimilarity, energy, entropy, homogeneity, maximum probability, variance, sum average, sum variance, sum entropy, difference variance, difference entropy, and Normalized inverse	78%	2010	Kuse et al. (2010)
	Nucleus area, perimeter, number of the separated parts of nucleus, mean, variance of the nucleus boundaries, roundness of nucleus, averaged nucleus color, fourteen features based on co-occurrence matrix, and two features based on Local Binary Pattern	96%	2011	Rezatofghi and Soltanian-Zadeh (2011)

Figura . Métodos y exactitud alcanzada al clasificar leucocitos. Tomado de Saraswat & Arya (2014)

Los clasificadores basados en SVM y ANN han logrado la mayor exactitud, siendo las opciones más utilizadas por los investigadores del área. Por ello, en este trabajo la clasificación de las imágenes se realizó mediante máquinas de soporte vectorial (SVM) descritas anteriormente y redes neuronales convolucionales las cuales se presentarán más adelante en este capítulo.

2.2.12.1. *Matriz de confusión y evaluación de un clasificador de imágenes*

Es una herramienta sencilla que permite resumir y evaluar la calidad de la clasificación realizada. Su uso proporciona un medio para realizar ajustes en el diseño del clasificador y para comparar el rendimiento entre diferentes tipos de clasificadores. La matriz tendrá una dimensión $N \times N$, siendo N el número de clases que se pueden

identificar. En la diagonal principal se coloca la totalización de los elementos correctamente clasificados para cada conjunto. Para el caso de un clasificador binario (2 valores posibles), la matriz de confusión tendrá la forma mostrada en la Figura . Su importancia radica en que a partir de los valores de TP, TN, FN y FP obtenidos de ella, pueden calcularse diferentes métricas de rendimiento del clasificador:

- Sensitividad o tasa de aciertos positivos: TP/P .
- Tasa de falsos positivos: FP/N
- Tasa de falsos negativos: FN/P
- Especificidad o tasa de aciertos negativos: TN/N
- Precisión: $TP / (TP + FP)$
- Tasa de errores: $(FP + FN) / (P + N)$

		Predicción	
		u_1	u_0
Clase Real	u_1	TP	FN
	u_0	FP	TN

Figura . Matriz de confusión de un clasificador binario. García-Ordás et al. (2016). **Nota:** TP = elementos de la clase u_1 clasificados correctamente. TN = elementos de la clase u_0 . FN = elementos de la clase u_1 clasificados como pertenecientes a u_0 . FP = elementos de la clase u_0 clasificados como pertenecientes a u_1 . Total de elementos positivos $P = FN + TP$. Total de elementos negativos $N = FP + TN$.

2.2.13. Aprendizaje profundo o *Deep Learning*

El rendimiento alcanzado por un método de aprendizaje automático depende en gran medida de la elección de la representación (o características) de los datos sobre los cuales se aplica (Bengio, Courville, & Vincent, 2013). Por ello, durante décadas la construcción de un sistema de reconocimiento de patrones o de aprendizaje automático dependía de una cuidadosa ingeniería y considerable experticia para diseñar algoritmos de extracción de rasgos que transformasen los datos no procesados (por ejemplo valores de intensidad de los píxeles de una imagen) en una representación interna significativa o vector de características a partir del cual se pudiera detectar o clasificar patrones existentes en la entrada (LeCun y otros, 2015). La importancia y dependencia de esa

labor manual destaca como una debilidad de los muchos algoritmos de aprendizaje, ya que no son capaces de extraer y organizar la información discriminativa de los datos, siendo esto una de las premisas pretendidas por la inteligencia artificial (Bengio y otros, 2013).

Dentro del mundo del aprendizaje automático, el aprendizaje de características (*feature learning*) o aprendizaje de representación (*representation learning*) es un conjunto de técnicas que permiten a una máquina descubrir las representaciones necesarias para la detección y clasificación de las características a partir de los datos sin procesar (LeCun, Bengio, & Hinton, 2015). Las representaciones aprendidas de esta forma frecuentemente resultan en sistemas con mucho mejor rendimiento que el que puede ser obtenido mediante representaciones diseñadas de forma manual. Ellas también permiten a los sistemas de inteligencia artificial adaptarse rápidamente a la realización de nuevas tareas con intervención humana mínima. Un algoritmo de aprendizaje de características puede encontrar un buen conjunto de rasgos para una tarea simple en minutos, o en un lapso de horas o meses si la tarea es compleja. El diseño manual de un algoritmo de extracción de rasgos para una tarea compleja requiere una gran cantidad de esfuerzo y tiempo por parte de expertos humanos; pudiendo implicar décadas de trabajo de una comunidad entera de investigadores (Goodfellow, Bengio, & Courville, 2016).

Bajo el nombre de aprendizaje profundo o *deep learning* se conoce a un conjunto de métodos de aprendizaje de características que utilizan múltiples niveles de representación con diferentes niveles de abstracción y se obtienen mediante la combinación de módulos más simples o menos abstractos.

De esta forma, *deep learning* enfrenta el problema de hallar una representación mediante su descomposición o expresión en términos de otras representaciones más simples, permitiendo al computador construir conceptos complejos a partir de otros

elementales. De esta forma, un sistema de aprendizaje profundo puede representar el concepto de un objeto o persona en una imagen mediante la combinación de elementos más simples, como esquinas o contornos, los cuales a su vez se definen en términos de bordes (Goodfellow et al., 2016).

Esto implica la existencia de múltiples capas de representaciones las cuales inician en un nivel muy genérico (bordes detectados a partir de los píxeles de una imagen por ejemplo) y se van especializando pasando por detectar elementos simples (como líneas o esquinas), formas comunes (círculos y cuadros), para finalmente combinarse en objetos o grupos de objetos en las capas finales. LeCun y otros (2015) señala que el aspecto clave del aprendizaje profundo es que esas capas de rasgos no son diseñadas por expertos humanos sino que se aprenden a partir de los propios datos utilizando un procedimiento de entrenamiento de propósito general.

El aprendizaje profundo ha probado ser útil en muchas disciplinas de software, incluyendo visión por computador, procesamiento de voz y audio, procesamiento de lenguaje natural, robótica, bioinformática y química, video juegos, motores de búsqueda, publicidad en línea y finanzas.

2.2.14. Redes neuronales unidireccionales profundas

Durante el desarrollo de las redes neuronales artificiales, la inclusión de capas intermedias (ocultas) entre las capas de entrada y salida, dio origen a la arquitectura de red conocida como red unidireccional profunda (*Deep feedforward network* – DFFN o simplemente *feedforward network*) también conocida como perceptrón multicapa (MLP). Este tipo de redes fueron popularizadas desde mediados de la década de los 80 y son extensiones de la simple red perceptrón propuesta por Rosenblatt en 1959 (Martín del Brío & Sanz, 2002). Al caso particular de una red unidireccional de una única capa de neuronas ocultas se le denomina red superficial (*shallow network*). Las neuronas de las capas ocultas de un MLP siempre son no lineales y la recomendación

en las redes modernas es utilizar la función de activación ReLU (*rectified linear unit*) (Goodfellow et al., 2016) Una característica resaltante de las MLP es que están masivamente interconectadas, así que un elemento en una capa cualquiera tiene cierto grado de influencia en todos los elementos de la siguiente capa (ver Figura).

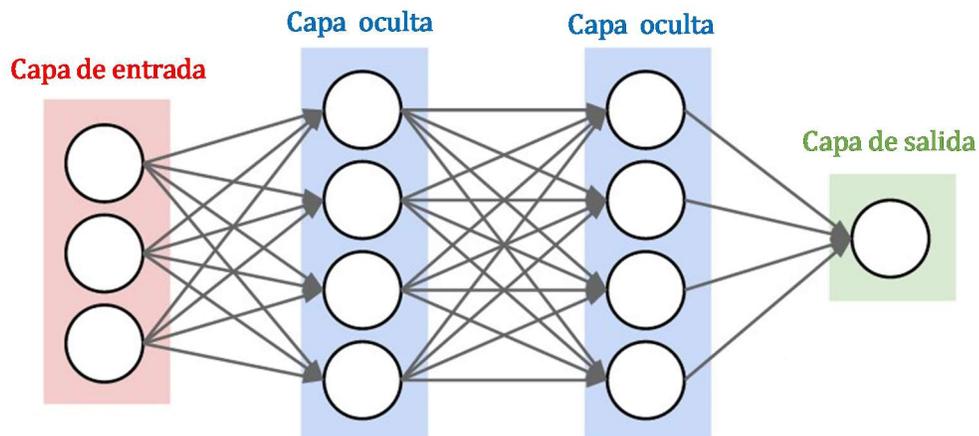


Figura . Representación de red MLP con dos capas ocultas. Modificado de “CS231n: Convolutional Neural Networks for Visual Recognition.” <http://cs231n.github.io/neural-networks-1/>

Este tipo de red neuronal utiliza un paradigma de aprendizaje supervisado ya que requiere de la presentación de los datos junto con un patrón deseado y se entrena normalmente mediante un algoritmo conocido como retropropagación de errores (*backpropagation*). El conjunto de arquitectura MLP y aprendizaje por retropropagación suele llamarse red de retropropagación (BP) (Martín del Brío & Sanz, 2002).

Las redes MLP han sido empleadas satisfactoriamente para resolver múltiples tipos de problemas, demostrando ser capaces de representar complejos mapeos o relaciones entre sus entradas y salidas y de abordar problemas de clasificación complejos de forma eficaz. Una MLP permite aproximar una función cualquiera $y = f^*(x)$, definiendo un mapeo $y = f(x, \theta)$ y aprendiendo los valores de los parámetros θ que resulten en la mejor aproximación de f^* . Por esta capacidad se le considera como un aproximador universal de funciones.

En efecto, el teorema de aproximación universal establece que una red MLP con una capa de entrada y salida lineal, y al menos una capa oculta con función de activación no lineal (por ejemplo de tipo sigmoide) puede aproximar cualquier función en un intervalo dado con un cierto nivel de error siempre que la capa oculta de la red tenga suficientes neuronas (Martín del Brío & Sanz, 2002).

Este teorema señala que sin importar cual función se trate de representar, un MLP lo suficientemente grande tendrá capacidad para hacerlo, sin embargo, ello no implica que el algoritmo de entrenamiento esté en capacidad de aprender la función deseada satisfactoriamente, ya sea porque no pueda hallar los valores necesarios de los parámetros que corresponden (no exista convergencia) o porque estos sean incorrectos como resultado del sobreajuste. En este sentido, es suficiente con que la red pueda tener únicamente una capa oculta para que pueda representar cualquier función pero ésta pudiera ser inimaginablemente grande y no ser posible de entrenar de forma adecuada o generalizar de manera incorrecta. En muchas circunstancias, el uso de modelos más profundos (más capas ocultas) puede reducir el número de neuronas requeridas para representar la función deseada y mejorar el error de generalización (Goodfellow et al., 2016).

Las MLP son consideradas como el modelo de aprendizaje profundo excelencia y han sido de extrema importancia en el campo del aprendizaje automático, formando parte esencial de muchas aplicaciones comerciales y constituyendo un bloque fundacional para el desarrollo de otras arquitecturas como las redes recurrentes, empleadas en el procesamiento de lenguaje natural, y las redes convolucionales, usadas para el reconocimiento de objetos en imágenes (Goodfellow y otros, 2016). Estas últimas son un tipo especializado de red *feedforward* las cuales han permitido atacar un aspecto que durante muchos años limitó el uso de las redes neuronales en favor de otras técnicas del área (como por ejemplo las máquinas de vectores de soporte), y es que, las arquitecturas tradicionales como el MLP no permiten conservar la información espacial

de los píxeles al momento de ser procesados, es decir, no existe invariancia ante las traslaciones o distorsiones locales de los datos de entrada. Esto último es señalado por LeCun y otros (1998) como la mayor deficiencia de este tipo de redes para aplicaciones de procesamiento de imágenes o voz.

2.2.15. Redes neuronales convolucionales profundas (CNN)

2.2.15.1. *Concepto y arquitectura general*

Las redes neuronales convolucionales profundas (DCNN o CNN) son modelos supervisados de aprendizaje utilizados principalmente para la clasificación de imágenes. El diseño de su arquitectura base se remonta al trabajo de Fukushima (1980), el cual describía un potente modelo llamado “Neocognitron” usado para el procesamiento de imágenes inspirado en la estructura del sistema visual de los mamíferos. Este sistema, sin embargo, carecía de un algoritmo eficiente de entrenamiento. LeCun y otros (1989) propusieron y probaron que el algoritmo de retropropagación del error, anteriormente empleado para entrenar redes perceptrón multicapa, podía ser empleado para entrenar este tipo de redes, creando la arquitectura básica de lo actualmente son las redes convolucionales profundas. Su enfoque fue muy exitoso para tareas como el reconocimiento óptico de caracteres (OCR) y procesamiento de textos manuscritos. El modelo de red LeNet-5 (Figura), descrito en LeCun y otros (1998), fue utilizado para la construcción de máquinas lecturas de cheques y se convertiría en el primer sistema comercial exitoso en emplear dicha tecnología. De acuerdo a Goodfellow et al. (2016), en la actualidad, las redes neuronales profundas compiten y superan a sistemas de inteligencia artificial basados en otras tecnologías de aprendizaje automático.

Las redes neuronales convolucionales se consideran una extensión del perceptrón multicapa y conservan características similares en cuanto a que: (a) poseen unidades de procesamiento (neuronas) agrupadas en capas con pesos que pueden ser entrenados, (b) las neuronas efectúan una operación de combinación de sus entradas y se activan

mediante una función no-lineal, (c) permiten encontrar una relación funcional entre los datos de entrada y sus salidas y (d) son entrenadas mediante el algoritmo estándar de retropropagación, mediante el cual los parámetros (pesos de asociados a las unidades de procesamiento) son ajustados en cada iteración de entrenamiento hasta que el error observado es minimizado.

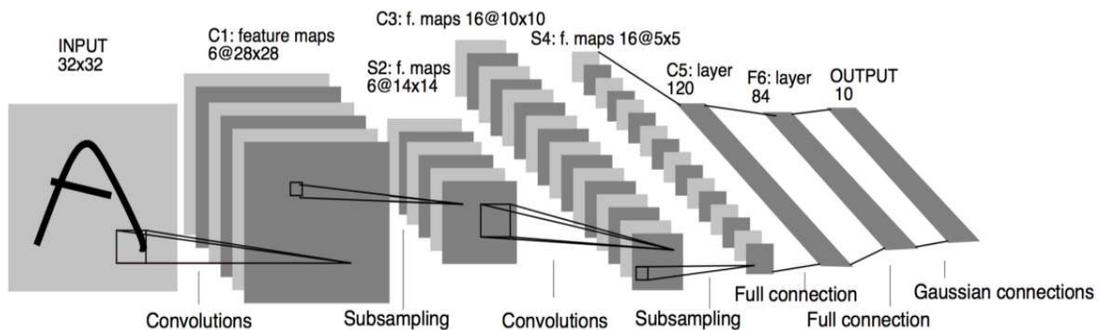


Figura . Arquitectura de la red convolucional LeNet-5. Tomado de LeCun y otros (1998)

Sin embargo, entre ambas topologías también existen diferencias importantes. Una de ellas es que mientras las MLP poseen capas bidimensionales y sus datos de entrada se organizan como un vector, en las CNN la estructura de las capas es tridimensional y los datos se representan como volúmenes. Esto se encuentra implícito en su arquitectura, pensada para ser empleada en aplicaciones de procesamiento de imágenes. Otra diferencia notable entre las MLP y las CNN se refiere a la conectividad entre las neuronas. En una MLP cada unidad de procesamiento de una capa oculta, está completamente conectada a todas las neuronas de capa anterior y no comparte conexiones o pesos con otras unidades de su misma capa. Por el contrario, en las CNN se utiliza el concepto de conectividad local, es decir que la activación de una neurona involucra únicamente a un sub-conjunto de las unidades de procesamiento de la capa anterior (campo receptivo). Adicionalmente, esta conexión es compartida entre todas las neuronas de presentes en la capa (pesos compartidos). Ambos factores hacen que el número de parámetros empleados por una CNN para procesar una imagen sea significativamente menor que el requerido al usar MLP y ello constituye buena parte de su éxito.

Las redes convolucionales profundas son de naturaleza jerárquica, es decir, las características o patrones reconocidos por la red evolucionan a través de su estructura de capas. Parten desde rasgos primitivos, o de bajo nivel de abstracción, en las capas más superficiales o iniciales, hasta combinarse para aprender interacciones complejas en las capas más profundas o finales de la red, las cuales, representan rasgos con niveles de abstracción mayores. Esto corresponde directamente con el paradigma principal del *Deep Learning* según el cual las características o patrones reconocidos por el sistema se aprenden automáticamente a partir de los datos, en contraposición al diseño manual típico del *machine learning* tradicional.

La arquitectura básica de una CNN está constituida por una secuencia de capas de neuronas apiladas en un patrón similar al mostrado en la Figura . Pese a no existir un orden fijo establecido para colocar las capas, se observa que dicha estructura funcionalmente comprende dos partes principales: una encargada de extraer las características de los objetos presentes en la imagen y otra empleada para clasificarles.

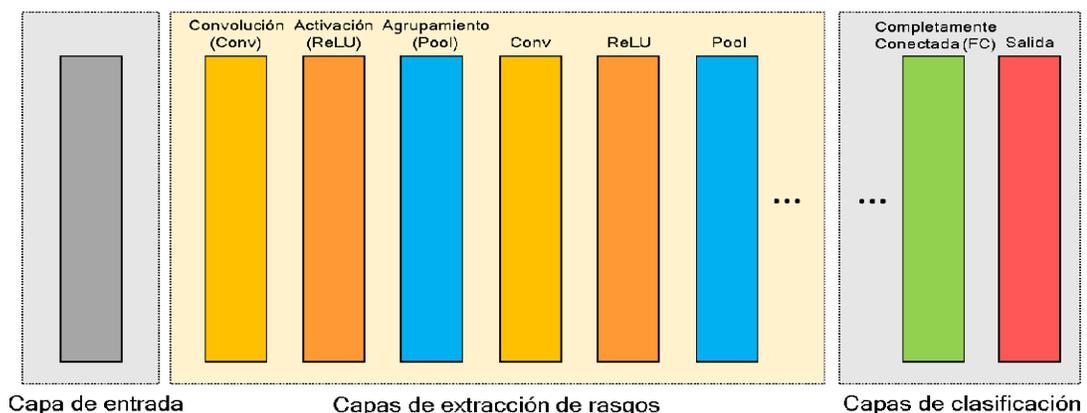


Figura . Arquitectura general de una CNN simple

La etapa de extracción de rasgos es la parte de la red neuronal convolucional que contiene sus elementos más significativos. En lugar de emplear capas de neuronas ocultas completamente conectadas, se realiza por medio de una estructura especial que alterna tres tipos de capas llamadas de convolución (Conv), activación (ReLU) y agrupamiento (Pool), respectivamente. Estas se colocan una a continuación de la otra

en forma de bloque, el cual es repetido en función del tamaño (profundidad) deseado de la red. En algunas representaciones gráficas de redes neuronales convolucionales, es frecuente omitir la capa ReLU y solamente indicar la función de activación empleada. Las capas de clasificación generalmente utilizan una o más capas de neuronas completamente conectadas (FC) las cuales combinan los rasgos de alto nivel para detectar características complejas y clasificar los objetos presentes en la imagen. La salida de la red incluye una capa final, la cual, utiliza la función exponencial normalizada (*softmax*) para asignar a cada neurona el valor de probabilidad normalizada de cada clase. De esta forma, cada neurona indicará la probabilidad de que la imagen en su entrada pertenezca a una de las clases para las cuales se ha entrenado la CNN.

2.2.15.2. Propiedades de las redes convolucionales

En las redes convolucionales se incorporan tres ideas fundamentales para reducir el número total de parámetros a entrenar y garantizar cierto grado de invarianza ante transformaciones geométricas de las imágenes, tales como desplazamiento, cambio de escala o distorsión. Estas ideas son: uso de campos receptivos locales, existencia de pesos compartidos y aplicación de sub-muestreo espacial o temporal (LeCun y otros, 1998). A continuación se explora brevemente cada uno de estos conceptos.

Campos receptivos locales

Wandell, Dumoulin, & Brewer (2007) señala que gran parte de la corteza visual en los mamíferos está organizada en mapas de campo visual o áreas visuales. Estos mapas exhiben una organización fija con respecto a la retina, por lo que se les denomina mapas retinotópicos, es decir que la imagen presentada se mantiene topográficamente intacta respecto a la retina en la superficie cortical, sin importar que el campo visual se desplace con el movimiento ocular. Esto resulta en una correspondencia uno a uno entre un punto en la retina y los centros de procesamiento (neuronas) para ese punto en los mapas de campo visual, de modo que, los puntos vecinos en la retina también se

procesan cerca en los mapas de campo visual (Eickenberg, Gramfort, Varoquaux, & Thirion, 2016). Por lo tanto, las neuronas cercanas (en la corteza visual) poseen campos receptivos en ubicaciones cercanas en la imagen.

Rawat y Wang (2017) comenta que la inspiración biológica de las CNN se fundamenta en los experimentos llevados a cabo por Hubel y Wiesel en la década de 1960. Estos pioneros de la neurociencia descubrieron que las neuronas en las primeras etapas de la corteza visual presentan una fuerte respuesta ante patrones de entrada simples específicamente orientados (como barras o bordes), mientras que, ignoraban otros más complejos. Estos patrones complejos por el contrario, generaban fuertes activaciones en las neuronas localizadas en etapas posteriores del sistema visual.

Sus observaciones respaldaron la idea de que ciertos mapas de campo visual están compuestos por un conjunto de neuronas con selectividad de estímulo común (Wandell et al., 2007), lo cual es uno de los postulados teóricos de la neurociencia visual. Según esto las regiones corticales se especializan en funciones perceptuales particulares, es decir que un mapa de campo visual específico sirve para ciertas especializaciones perceptivas.

Hubel y Wiesel también encontraron que la corteza visual consistía de células simples con campos receptivos locales y células complejas cuya respuesta es invariante ante entradas desplazadas o distorsionadas. Estas nociones clásicas inspiraron directamente las funciones de las capas de convolución y agrupamiento de las CNN cuya arquitectura general recuerda la jerarquía existente en la vía ventral de la corteza visual (LeCun et al., 2015)

Como se mencionó anteriormente, a diferencia de las DFFN donde la activación de una neurona depende de todas las neuronas de la capa anterior (conectividad completa o global), la arquitectura de las CNN obliga la extracción de rasgos locales, limitando los

campos receptivos de las unidades de procesamiento, es decir, que cada una se conecta únicamente a una pequeña región del volumen de entrada. La diferencia entre ambos enfoques muestra en la Figura . La extensión espacial de esta región (ancho y alto) se define mediante un hiperparámetro de diseño denominado campo receptivo de la neurona (F), en clara alusión a su equivalente biológico. Una única neurona procesa esa porción de la entrada mediante una operación de convolución, donde este hiperparámetro (F) define el tamaño del filtro o *kernel* a utilizar. Este filtro tendrá, sin embargo, la profundidad del volumen de los datos de entrada. El filtro se desplaza sobre la entrada para calcular el conjunto de activaciones en la capa oculta (ver Figura - derecha) dando origen a un mapa de activación o mapa de características (*feature map*) como respuesta específica a la presencia de un rasgo o patrón en diferentes ubicaciones de la entrada. El desplazamiento del *kernel* se define mediante un hiperparámetro llamado paso del filtro o *stride* (S).

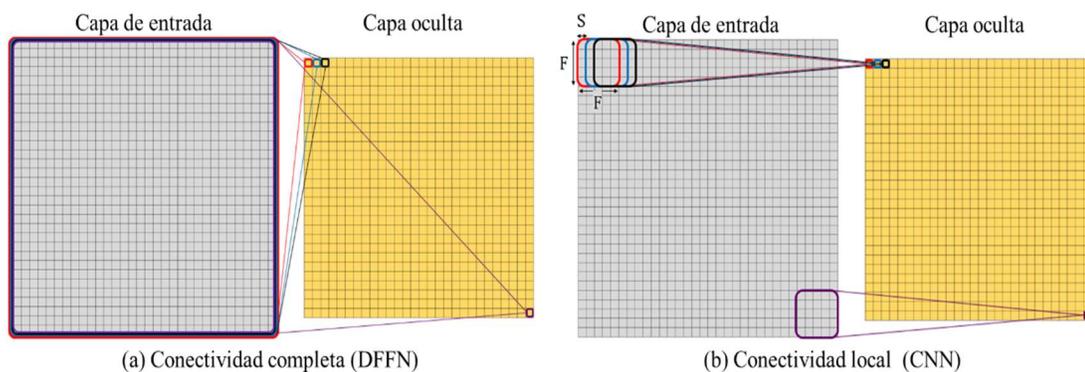


Figura . Conectividad local en las redes convolucionales. **Nota.** En las redes DFFN o MLP (izquierda) cada neurona de la capa oculta se conecta a todos los elementos de la entrada, por el contrario en una CNN la activación de la neurona depende únicamente de una pequeña región de la entrada

El uso de la conectividad local tiene un gran efecto de reducción sobre la cantidad de parámetros entrenables totales existentes en las capas de neuronas ocultas del modelo. Por ejemplo, para una imagen RGB de entrada cuyas dimensiones sean 225×225 píxeles en una red MLP cada neurona tendría un total de 151876 parámetros. En una CNN que utilice un campo receptivo (F) igual a 5, cada neurona requerirá únicamente de 76 pesos involucrados en su función de activación.

Pesos compartidos

Debido a que cada mapa de activación es generado a partir de un único filtro, todas las unidades de procesamiento contenidas en éste comparten sus pesos sinápticos. Esto representa una de las características fundamentales de este tipo de redes y permite que existan unidades de detección replicadas compartiendo la misma configuración logrando que las características puedan ser detectadas sin importar su posición en el campo visual o espacio de entrada. Esto quiere decir que, la forma de compartir los parámetros en la CNN, hace que los mapas de activación se generen por medio de una función con una propiedad llamada equivarianza a la traslación. Decir que una función es equivariante significa que si la entrada cambia, la salida lo hace de la misma forma. Por lo tanto, si una traslación es aplicada a la imagen de entrada, el mapa de características contendrá visualmente la misma transformación.

Otro efecto importante de esta configuración es que reduce notablemente la cantidad de parámetros entrenables (pesos sinápticos) del modelo, incrementando la eficiencia del aprendizaje. Para apreciar la escala de la reducción de parámetros producto de esta característica, considere una red convolucional cuya entrada es una imagen RGB de 225 x 225 píxeles y contiene una capa oculta. Si esta capa oculta está conformada por 10 mapas de características generados usando filtros de 5x5, cada mapa de activación requerirá un total de $5 \times 5 \times 3 + 1 = 76$ parámetros, necesiándose en total $76 \times 10 = 760$ parámetros. Por otro lado, si se utilizase una red neuronal con arquitectura MLP, su capa de entrada tendría un total de $255 \times 255 \times 3 = 151875$ nodos y para conectarla con una capa oculta de 10 neuronas se requerirían 1518760 parámetros.

Sub-muestreo o agrupamiento (pooling)

El propósito de los filtros de convolución empleados en una CNN es producir mapas de características, los cuáles, se activarán en mayor o menor medida al pasar sobre regiones de la imagen de entrada que contengan las características buscadas, como bordes, líneas, contornos, entre otras. Para detectar la gran cantidad de rasgos que

podieran aparecer en la imagen se suele generar un número elevado de mapas, lo que puede implicar utilizar decenas e incluso cientos de ellos. Esto significa que el volumen de los datos pudiera hacerse muy elevado debido al crecimiento en la profundidad de los mismos. Por ello, en las CNN tradicionales se incorpora una operación de sub-muestreo, también denominado agrupamiento o *pooling* cuya función es reducir la dimensionalidad espacial del volumen de datos, manteniendo su profundidad y preservando la mayor cantidad de información relevante necesaria para detectar la presencia de las características.

La operación se aplica sobre los mapas de activación y consiste en un promediado local realizado sobre una ventana de dimensiones pre-definidas. Para ello, se particiona la entrada en un conjunto de sub-regiones no solapadas y se calcula el valor máximo o promedio en cada una, esto resulta en una salida única para cada sub-región del espacio de entrada (ver Figura). La función más comúnmente empleada para realizar el sub-muestreo es el valor máximo, implementándose por medio de un filtro con dimensión 2x2 y paso igual a 2, valores que permiten reducir el volumen de salida a la mitad (Patterson & Gibson, 2017).

Al utilizar esta operación de agrupamiento efectivamene se reduce el número de neuronas ocultas y, por lo tanto, el cálculo de la subsiguiente capa de convolución es más eficiente. Esto se realiza sin necesidad de incluir parámetros entrenables en la red ya que la función empleada es conocida y no depende de ninguna relación entre los datos de entrada y salida.

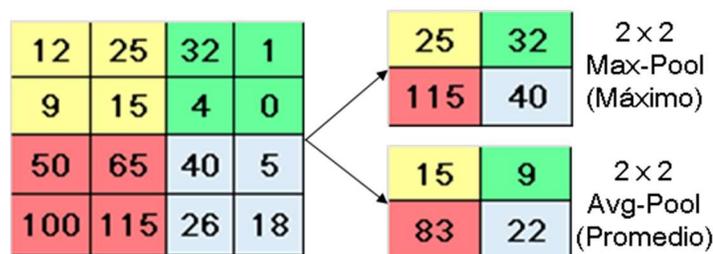


Figura . Operaciones de sub-muestreo Max-Pool y Avg-Pool

La incorporación del operador introduce invarianza a pequeñas traslaciones del objeto, sin embargo, ya que su uso implica pérdida de información respecto a la localización exacta de los patrones en la imagen. Por esto y, debido al constante incremento de las capacidades de procesamiento de los computadores, algunas arquitecturas de CNN modernas como ResNet (He, Zhang, Ren, & Sun, 2016) y GoogLeNet (Szegedy, y otros, 2015) han reducido la cantidad de capas de agrupamiento presentes en su estructura.

La incorporación de estos conceptos en el diseño de la red neuronal hace que las características aprendidas por éstas sean invariantes ante el cambio de posición o traslación de los objetos, pudiendo ser detectadas en cualquier localización dentro de la imagen pero no poseen, en general, invarianza ante la rotación. Es posible lograr cierto nivel invarianza rotacional con el apropiado incremento de los datos de entrenamiento (Patterson & Gibson, 2017), ello implica mostrar a la red durante la fase de entrenamiento varias veces el mismo patrón pero rotado aleatoriamente.

2.2.15.3. Capas usadas en el diseño de las CNN

Como se comentó anteriormente una red neuronal convolucional profunda típica, se estructura apilando en secuencia capas que transforman un volumen tridimensional de entrada en otro de salida usando una función diferenciable (Patterson & Gibson, 2017). El volumen de los datos evoluciona a medida que atraviesan las distintas capas de la red, como se ilustra en la Figura , se observa que en las capas iniciales este es mucho menos profundo que en las capas finales y el efecto del sub-muestreo en la reducción dimensional posterior a cada capa de convolución.

Los tipos de capas que se encuentran en la estructura de una red neuronal convolucional se describe a continuación:

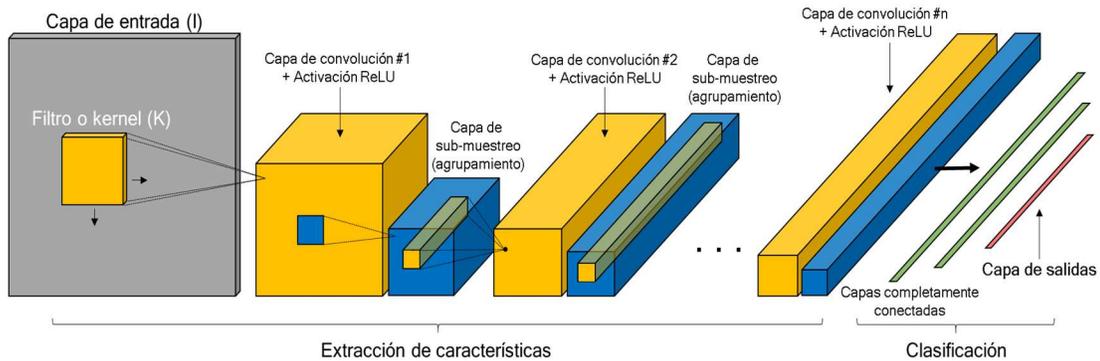


Figura . Representación simplificada de una red neuronal convolucional. **Nota.** Se ha simplificado la representación considerando que la capa de activación (ReLU) forma parte de la capa de convolución

Capa de entrada

Esta capa representa la entrada de datos a una red neuronal convolucional y define el ancho, alto y número de canales de color que componen las imágenes a ser procesadas. Típicamente el número de canales es tres, debido a que utilizan imágenes a color RGB. En el paquete informático MATLAB al especificar esta capa, mediante la función *imageInputLayer*, se pueden incluir operaciones de pre-procesamiento como normalización, reflexión horizontal o cortes aleatorios (The MathWorks, Inc, 2019).

Capa de convolución

Este es el bloque constructivo de una CNN como su nombre lo sugiere y por tanto la capa de mayor importancia. La convolución en su forma más general es una operación entre dos funciones con argumentos reales definida como $s(t) = \int x(a) \cdot w(t - a) da$, y típicamente se denota mediante un asterisco $s(t) = (x * w)(t)$. En el contexto de las redes neuronales convolucionales, el primer argumento de la operación (la función x) es un arreglo multidimensional de datos denominado entrada (I) y el segundo (función w), llamado *kernel* (K) o filtro, es usualmente un tensor de parámetros que son ajustados durante el entrenamiento del modelo. Adicionalmente, la operación se lleva a cabo de forma discreta, con una extensión de los datos finita y, si se considera que tanto la entrada (I) como el *kernel* sean bidimensionales, puede ser escrita como lo indica la ecuación θ (Goodfellow y otros , 2016). En la cual, I es la imagen (2D) de

entrada y K es la matriz de coeficientes del filtro (también llamada máscara de filtrado) de dimensión $F_w \times F_h$ (en general, $F_w = F_h = F$).

$$S(i, j) = (I * K)(i, j) = \sum_{m=0}^{F_w} \sum_{n=0}^{F_h} I(m, n) K(i - m, j - n) \quad ()$$

Esta expresión, conocida como convolución bidimensional discreta, es ampliamente utilizada en el procesamiento de imágenes digitales y permite calcular el píxel de salida $S(i, j)$ mediante una suma ponderada de los elementos vecinos. La Figura ilustra el cálculo del píxel de salida $S(1,1)$ para el caso de una imagen de 5×5 y un filtro típico de 3×3 ; los elementos de igual color representan los productos involucrados en aplicación de (). Dicha ecuación implica que el *kernel* es reflejado, tanto en sentido vertical como horizontal, lo cual equivale a una rotación de 180 grados en torno a su centro como lo evidencia la Figura (b).

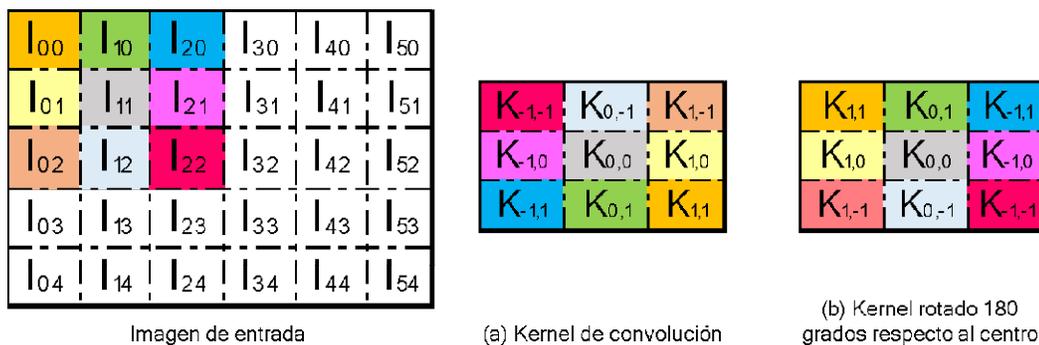


Figura . Convolución bidimensional entre una imagen de entrada I y un filtro K .

Esto es aprovechado al momento de implementar computacionalmente la convolución, procediendo primero a realizar la rotación del *kernel* y luego el producto escalar entre éste y la imagen de entrada. Esta última operación matemática, indicada por la ecuación () y conocida como correlación cruzada tiene especial importancia para las redes neuronales convolucionales.

$$S(i, j) = (K \star I)(i, j) = \sum_m \sum_n I(i + m, j + n) K(m, n) \quad ()$$

A pesar de utilizarse en la literatura el nombre convolución para referirse a la operación matemática empleada en el cálculo de la pre-activación de las neuronas en este tipo de redes, Goodfellow y otros (2016) aclara que, en la práctica, estos modelos se implementan utilizando la correlación cruzada en lugar de la convolución propiamente dicha. Sin embargo, ello no presupone ningún problema en el marco del aprendizaje automático ya que el algoritmo usado aprenderá los valores adecuados sin importar que utilice el *kernel* invertido o no.

Para las redes neuronales convolucionales la operación de convolución (o correlación cruzada, según sea la implementación) se realiza sobre una entrada tridimensional representada mediante un tensor de tercer orden y varios filtros también tridimensionales. La aplicación de cada filtro a la entrada da origen a un mapa de activación y estos conforman un volumen cuyas dimensiones están determinadas por cuatro hiperparámetros: campo receptivo - F, cantidad de filtros usados en la capa- k, paso del filtro - S y relleno de ceros - P, como se muestra en la Figura .

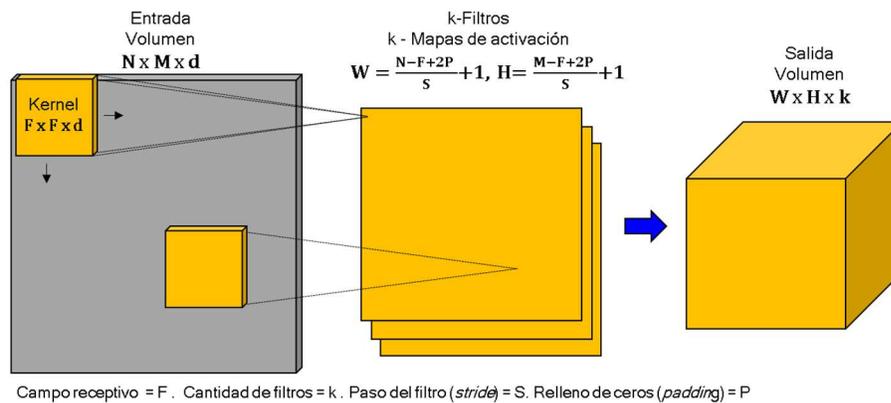


Figura . Volumen de salida en una capa de convolución

El tamaño del filtro (F) corresponde al campo visual receptivo de las neuronas en el mapa de activación. El paso del filtro (S) se refiere a la cantidad de píxeles que se desplaza el filtro al moverse sobre la imagen, como fue indicado anteriormente. La profundidad de los datos de salida corresponde con la cantidad de filtros empleados (k). El parámetro *padding* (P) se utiliza en aquellos casos que se deba rellenar los

bordes de la entrada con ceros a fin de adaptarla para permitir que la convolución sea válida y poder hacer el barrido completo sobre las dimensiones de la entrada. Su uso también posibilita ajustar el tamaño espacial del volumen de salida, ya que, sin su incorporación, éste disminuiría progresivamente conforme se avanza hacia las capas más profundas, limitando por tanto la cantidad de capas ocultas del modelo. Finalmente, la cantidad de parámetros entrenables de una capa de convolución incluyendo pesos (coeficientes de los filtros) y umbrales se puede calcular como $((F \cdot F \cdot d) + 1) \cdot k$.

El fundamento al entrenar este tipo de redes consiste en que los coeficientes de estos filtros no son conocidos sino que se ajustan mediante el aprendizaje, de forma que, cada filtro extraiga las correspondientes características buscadas partiendo de los propios datos.

Así una vez concluido el entrenamiento, la operación de convolución actúa como un detector de rasgos, el cual produce su mayor activación ante determinados patrones locales de entrada, es decir cuando estos patrones aparecen dentro de su campo receptivo. A medida que la red se hace más profunda (contiene más capas de convolución) los filtros pueden reconocer combinaciones no lineales de características más complejas, implicando que la profundidad de la red resulta un factor importante en su rendimiento.

Capa de activación (ReLU)

Las capas de convolución y completamente conectadas son usualmente seguidas por capas de activación, también conocidas como capas ReLU. Estas aplican la función de rectificación lineal ReLU, dada por $f(x) = \max(0, x)$, la cual es una operación usada para introducir no linealidad en la red y lograr que aprenda características más discriminativas. En el pasado las funciones de activación como sigmoide y tangente hiperbólica fueron utilizadas para este propósito, sin embargo, se ha observado que la

implementación computacional de ReLU es más rápida y su uso reduce el tiempo de entrenamiento significativamente. Por estas razones, se ha convertido en una opción muy empleada en la actualidad. Otras funciones similares (ver Figura) han sido propuestas con éxito como alternativas o mejoras de ReLU sin lograr aún la misma aceptación. Nwankpa, Ijomah, Gachagan y Marshall (2018) presenta un completo resumen de 21 funciones de activación empleadas en el contexto del *deep learning*, en el cual se observa la predominancia de ReLU en las principales arquitecturas de redes convolucionales.

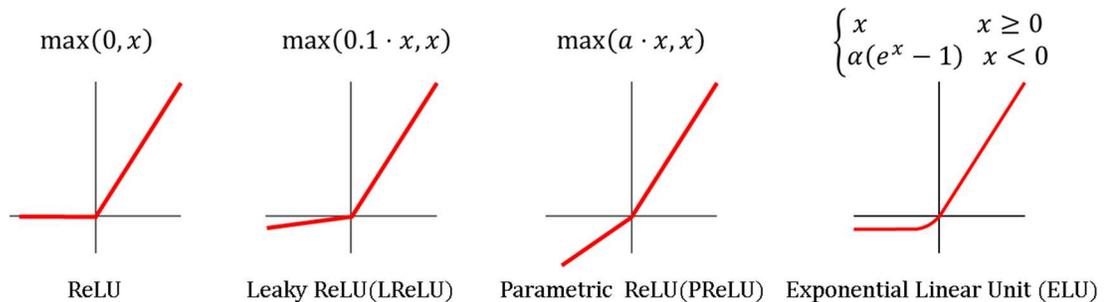


Figura . Función de activación ReLU y otras usadas en distintas redes neuronales

Capa de agrupamiento o sub-muestreo

Las capas de agrupamiento (*Pool*) son utilizadas luego de las capas de convolución y ReLU para la reducción progresiva de la dimensionalidad de los mapas de características y, consecuentemente, la disminución de la cantidad de cálculos realizados por las etapas subsecuentes. Generalmente, se escoge un tamaño (F) y paso de filtro (S) y, posteriormente, se aplica al espacio de entrada una operación como máximo o promedio, lo cual resulta en una única salida para la subregión de entrada. Esto es hecho para el mapa de características entero mediante el deslizamiento del filtro sobre él. El objeto de ésta operación es codificar la información más representativa de la región mientras se preservan los detalles espaciales relativos. Adicionalmente, también previene el sobreajuste. La Figura muestra la reducción dimensional (más no en la profundidad) producto de la capa de sub-muestreo.

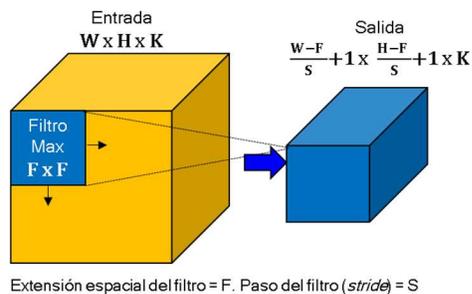


Figura . Volumen de salida en una capa de agrupamiento

Capas completamente conectadas

En una red neuronal convolucional, luego del conjunto de capas de convolución y agrupamiento, típicamente se encuentra una o más capas completamente o densamente conectadas, las cuales realizan una combinación de los rasgos con mayor nivel de abstracción y permiten obtener la salida de la red al formar parte de la etapa de clasificación.

Estas capas funcionan como una red MLP tradicional, donde cada elemento es considerado un nodo independiente de la ANN y se encuentra completamente conectado con todas las salidas que conforman el volumen de la capa anterior. Para ello, el volumen de entrada es transformado en un vector y cada unidad de procesamiento realiza una suma ponderada de su producto con todos sus parámetros de entrada y un valor umbral independiente (*bias*). Esto implica que estas capas contienen normalmente muchos más parámetros entrenables que todas las capas convolucionales que le anteceden.

En las capas completamente conectadas se utiliza a ReLU como función de activación (salvo en la última de ellas que emplea la función *softmax* y constituye parte de la capa de salida). También es frecuente que, con el objetivo de prevenir el sobreajuste (*overfitting*), se introduzca en estas capas un método de regularización denominado abandono o *dropout*. Esta estrategia presentada por Hinton y otros (2012) consiste en omitir aleatoriamente un porcentaje estadístico de las neuronas de la red durante la etapa de entrenamiento y se utiliza para prevenir co-adaptaciones complejas en los

datos de entrenamiento, de manera tal que una neurona no dependa completamente de que otras estén presentes.

Al implementar modelos de CNN usando el paquete informático MATLAB, este tipo de capas se definen mediante la función *fullyConnectedLayer* y tanto la función de activación como el uso de *dropout* se especifican como capas adicionales según se indica en la Figura . En ella se puede observar como las capas densamente conectadas llamadas ‘fc6’ y ‘fc7’ van seguidas de una capa de activación ReLU y otra de regularización con abandono del 50%, mientras, la capa ‘fc8’ forma parte de la salida de la red, utiliza activación *softmax* y no incluye *dropout*.

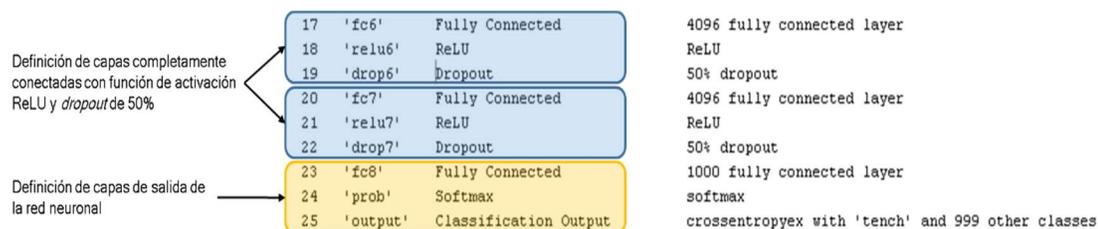


Figura . Especificación de capas FC y de salida en una CNN típica

Capa de salida

En la mayoría de las arquitecturas de CNN modernas, la última capa completamente conectada posee una dimensión de salida igual al número de clases que ésta puede identificar y los valores del vector de salidas corresponden a las probabilidades asociadas con cada clase. A fin de lograr esto, la capa final utiliza como activación la función *softmax* para asignar a cada neurona su valor.

La función exponencial normalizada o *softmax* calcula la distribución probabilística de un vector de números reales y, al ser usada de función de activación en las ANN, le asigna a cada neurona el valor de probabilidad normalizada. La clase objetivo (o valor verdadero) se identifica a la como aquella de exhibe el mayor valor de probabilidad de pertenencia. *Softmax* produce una salida cuyo rango de valores está comprendido entre 0 y 1, cuya suma sobre todas neuronas de salida es igual a 1 y se calcula mediante la ecuación ().

$$f(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \quad ()$$

Esta función *softmax* es usada de forma frecuente en modelos de clasificación multiclase. En efecto, aparece en casi todas las capas de salida de las arquitecturas de redes convolucionales, por ejemplo Nwankpa y otros (2018) muestra que ha sido empleada en 10 de las 11 redes consideradas como avances significativos de la última década.

Al momento de su implementación con MATLAB, la capa de salida de la red convolucional se especifica como un grupo compuesto por una capa completamente conectada (*fullyConnectedLayer*), seguida de una capa de activación *softmax* (*softmaxLayer*) y finalmente una capa de clasificación (*classificationLayer*). Esta última se encarga de calcular la función de pérdida de entropía categórica cruzada (*cross-entropy loss*) a partir de las probabilidades de cada clase dadas por *softmax*, cuyo valor se utiliza como medida de error de la salida del clasificador y se busca minimizar durante el entrenamiento.

2.2.16. Modelos de redes neuronales convolucionales

Debido a la flexibilidad en la arquitectura de las redes convolucionales, los investigadores han creado gran variedad de modelos capaces de realizar diversas tareas de extracción de rasgos y clasificación. Algunos desarrollos relevantes del área se describen en la siguiente sección, incluyendo los modelos de redes convolucionales usadas durante el presente Trabajo Especial de Grado.

2.2.16.1. AlexNet

Las redes neuronales convolucionales ya habían sido utilizadas en años anteriores para vencer en competiciones de algoritmos de aprendizaje automático y visión por computador sin lograr mucho impacto, sin embargo el intenso interés existente en la

actualidad por el uso del aprendizaje profundo en aplicaciones comerciales comenzó cuando Krizhevsky, Sutskever y Hinton (2012) ganaron la competencia de reconocimiento de objetos “*ImageNet Large Scale Visual Recognition Challenge (ILSVRC)*” del año 2012. Esto fue realizado utilizando AlexNet , una red neuronal convolucional de 8 capas de profundidad la cual fue inicialmente entrenada usando 1.2 millones de imágenes de la base de datos ImageNet y que puede clasificar imágenes en 1000 categorías de objetos como teclado, lápiz, diferentes tipos de animales, vehículos, trenes, aviones, entre otras.

En la Figura se muestra un esquema simplificado de la arquitectura de AlexNet. Se observa la existencia de 5 capas de convolución resaltadas en color amarillo y 3 capas de neuronas completamente conectadas en color verde. Para las capas de convolución y agrupamiento (mostradas en azul) se indican los valores de los hiperparámetros que definen el volumen de los datos resultantes en la salida de cada capa (mostrados en la parte superior), estos son: campo visual o tamaño del filtro F , paso del filtro (*stride*) S , relleno de cero (*padding*) P y profundidad o cantidad de filtros K .

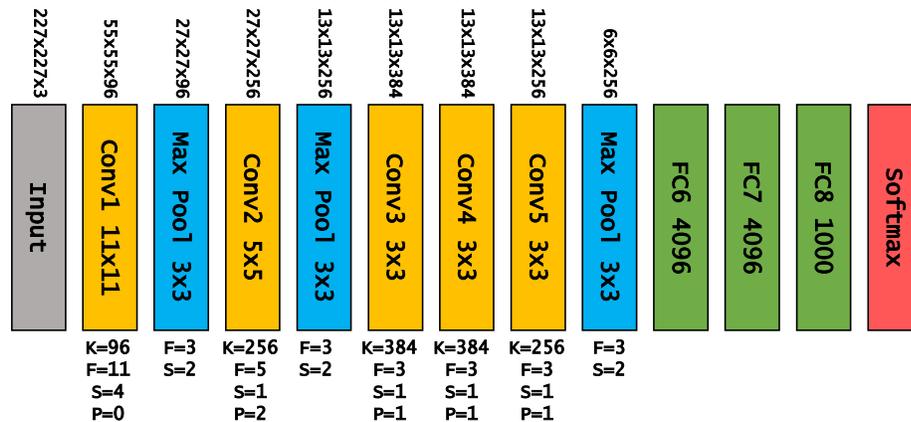


Figura . Arquitectura simplificada red AlexNet. En la figura se han omitido las capas de normalización posteriores a las primeras dos capas de agrupamiento.

La arquitectura de la red AlexNet implica la existencia de 60 millones de parámetros y para evitar el sobreajuste sus autores recurrieron a dos estrategias distintas. La primera fue aumentar artificialmente el conjunto de imágenes de entrenamiento (*data*

augmentation) empleando para ello operaciones de traslación y reflexión horizontal, así como variaciones aleatorias en la intensidad y color de las imágenes. La segunda estrategia fue la inclusión del método de regularización conocido como abandono, aplicado con un valor de probabilidad de 50%, en las capas completamente conectadas FC6 y FC7.

Krizhevsky y otros (2012), inspirados en un trabajo previo de Nair y Hinton, introdujeron al campo de las CNN el uso de la función de activación no lineal ReLU, en lugar de funciones sigmoide o tangente hiperbólica. Dicha función les permitió entrenar al modelo de red neuronal en un tiempo considerablemente menor, estableciéndola como el estándar posteriormente empleado por otros investigadores.

El éxito logrado por esta red convolucional marcó una clara influencia en el área, siendo algunos de los modelos destacados la competencia ImageNet en los años subsiguientes: ZFNet (Zeiler & Fergus, 2014) y VGGNet (Simonyan & Zisserman, 2014), básicamente mejoras de ella. El primero de ellos (ZFNet), utilizó una modificación de la arquitectura AlexNet incluyendo dos cambios significativos: (a) Reducción del campo receptivo en la primera capa de convolución, pasando de 11x11 con stride de 4 píxeles, a una dimensión de 7x7 con stride de 2 píxeles y (b) Aumento de la cantidad de filtros empleados en la tercera, cuarta y quinta capa de convolución yendo de 384,384 y 256 a 512, 1024 y 512, respectivamente. El segundo se describe con mejor detalle a continuación.

2.2.16.2. VGGNet

La arquitectura VGGNet propuesta por investigadores de la Universidad de Oxford planteó el uso filtros convolucionales de menor tamaño así como, una profundidad mucho mayor en la estructura de la red neuronal como herramientas para incrementar la capacidad de representación del modelo y lograr superar ampliamente los resultados obtenidos por AlexNet y ZFNet.

Simonyan y Zisserman (2014) evaluaron redes con 11, 13, 16 y 19 capas de profundidad, siendo las últimas capaces de producir resultados iguales o superiores al estado del arte al momento de su publicación. En el marco de la ILSVRC-2014, la tasa de error en la tarea de clasificación lograda por algunas versiones de VGGNet (aproximadamente 8%) implicó un avance notable frente a lo alcanzado por AlexNet (16.4%) y ZFNet (11.7%). Su modelo fue ligeramente superado en esa ocasión por la novedosa arquitectura GoogLeNet de Szegedy y otros (2015).

Todas las capas de convolución de la arquitectura VGG son iguales con campo receptivo de 3x3, *stride* y *padding* de 1 píxel. Las capas de agrupamiento (5 en total) tienen dimensión de 2x2 y son aplicadas con paso (*stride*) de 2 píxeles. La pila de capas convolucionales es seguida por 3 capas completamente conectadas: las dos primeras de 4096 elementos y la tercera de 1000 neuronas de acuerdo a los requerimientos de clasificación de la competición ILSVRC. La capa de salida de la red es del tipo *softmax*. Esta configuración en las capas finales del modelo VGGNet es la misma en todas sus versiones. La Figura muestra una representación esquemática de las redes VGG-16 (arriba) y VGG-19 (abajo).

Una de las características significativas de la arquitectura VGGNet es que al utilizar capas apiladas de filtros de 3x3 mostró un campo receptivo efectivo igual al obtenido con filtros de 7x7 pero con un costo computacional significativamente menor (tanto en número de parámetros como en uso de memoria), ello señaló la tendencia para los modelos posteriores de redes convolucionales.

La arquitectura de muchas capas de este modelo de red implica la existencia 133 a 144 millones de parámetros y la representación de capa imagen ocupa gran cantidad de memoria, por ello, para el entrenamiento de cada modelo los autores requirieron del uso de 4 GPUs NVIDIA® Titan Black (6GBytes) operado en paralelo durante un lapso de 2 a 3 semanas. La Figura reproduce dos gráficos comparativos entre diferentes

arquitecturas de redes convolucionales elaborados por Canziani, Paszke, y Culurciello (2017), en los cuales se pueden apreciar los requerimientos de memoria (izquierda) y el tiempo de inferencia de la red (derecha). Se extrae que las redes VGG-16 y VGG-19 requieren, para procesar un lote de 16 imágenes, de aproximadamente 1.7 a 1.9 GBytes de memoria, mientras que el modelo de AlexNet precisa de poco más de 300 MBytes. También se puede observar como el tiempo de inferencia (tiempo usado por la red para procesar una imagen en sentido directo) de ambas versiones de VGGNet fue superior a todas las arquitecturas evaluadas por estos autores.

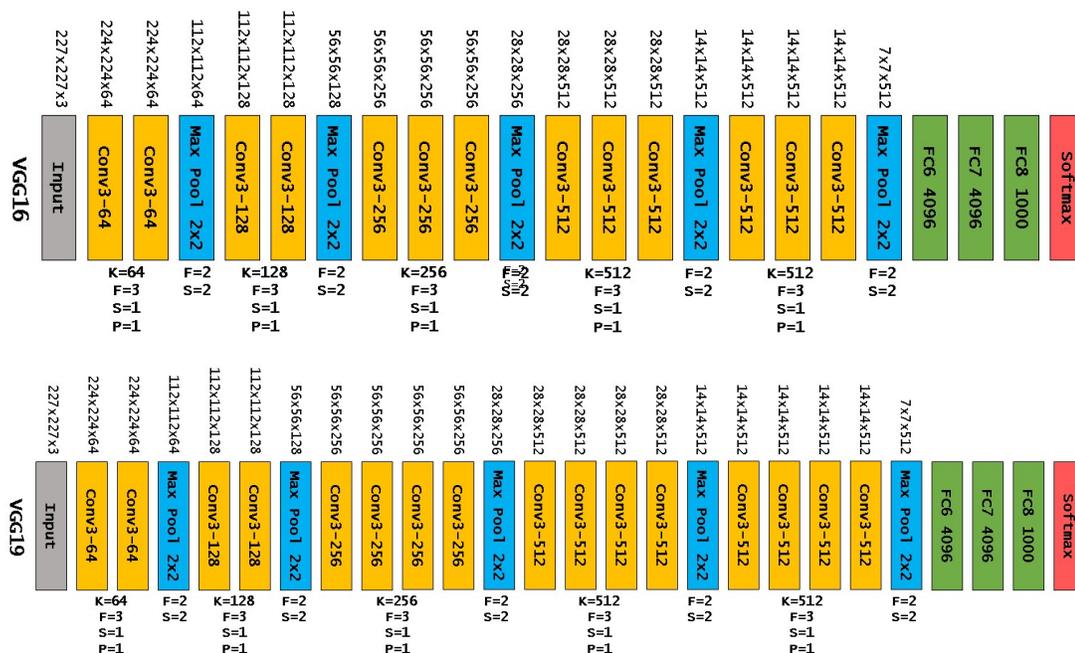


Figura . Arquitectura simplificada de redes VGG-16 y VGG-19.

2.2.16.3. GoogLeNet

La propuesta de Szegedy y otros (2015) también partió de la idea fundamental de incrementar la profundidad de la red convolucional. Pero en lugar de simplemente apilar más capas en serie generando un modelo computacionalmente complejo de manejar (como ocurre con las redes VGGNet), estos investigadores se enfocaron en lograr mayor profundidad de una manera eficiente. Para ello incorporaron el concepto de modulo “Inception” como parte estructural de la arquitectura. Adicionalmente, en

la red GoogLeNet se eliminan las capas completamente conectadas de la etapa de salida de la arquitectura. Con ello, a pesar de constituir una arquitectura mucho más profunda y compleja que sus predecesoras, el número de parámetros o neuronas a entrenar se pudo reducir a 5 millones, es decir, unas 12 y 28 veces menos que AlexNet y VGGNet, respectivamente. En términos de exactitud, GoogleNet superó ligeramente a VGGNet en la ILSVRC-2014, cuando ambas fueron presentadas, al alcanzar un error de clasificación menor al 7%.

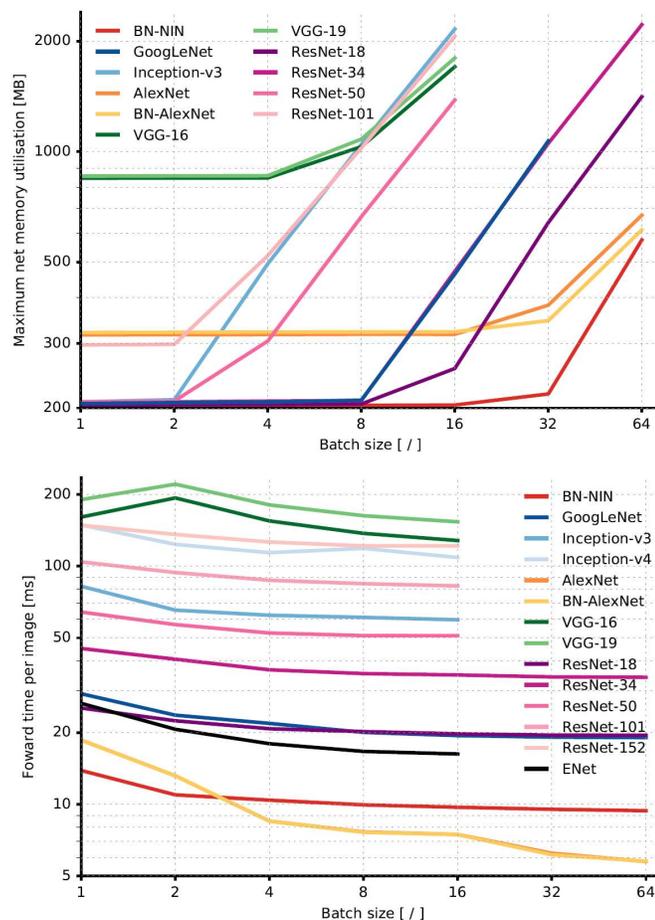


Figura . Comparación de requerimientos de memoria y tiempo de inferencia en distintos modelos de redes neuronales convolucionales. Tomado de Canziani et al.,(2017)

La red se fundamenta en las ideas propuestas anteriormente por Lin, Chen y Yan (2013) de “red dentro de la red” (*Network in Network - NiN*), las cuales se resumen en dos conceptos relevantes:

- Módulos no lineales en lugar de simple convolución:** Los filtros de convolución usados en las CNN para generar los mapas de características son operadores lineales, por tanto, aprenden rasgos con bajo nivel de abstracción y pueden ser reemplazados por estructuras no-lineales capaces de aprender rasgos más robustos. Lin y otros (2013) reemplazaron los filtros lineales por redes neuronales del tipo perceptrón multicapa que operan sobre el mismo campo receptivo siendo llamados capas Mlpconv (ver Figura). Estos son desplazados de la misma forma que los filtros convolucionales para generar los mapas de características pero son capaces de extraer mejores rasgos mejorando la abstracción y exactitud del modelo.

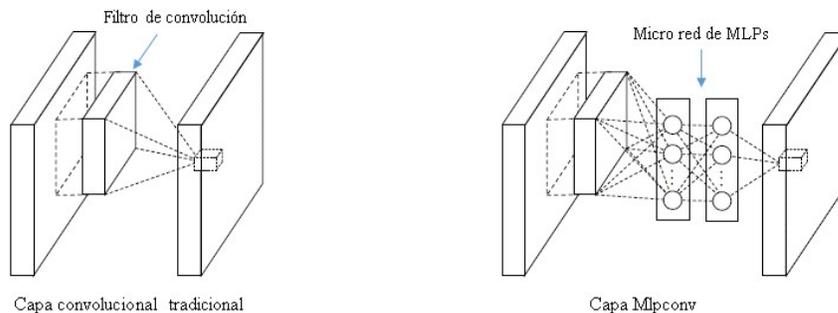


Figura . Comparación de capa de convolución lineal y micro red Mlpconv.
Modificado de Lin y otros (2013)

- Agrupamiento global promedio (*Global Average Pooling*):** En las redes convolucionales existentes hasta ese momento, la última capa de convolución era vectorizada y utilizada para alimentar varias capas de neuronas completamente conectadas las cuales eran usadas para realizar la clasificación de la imagen introducida a la red. Esta topología utiliza las capas de convolución como un extractor de rasgos previo a un clasificador tradicional basado en redes neuronales artificiales, situación que genera dos problemas fácilmente observables:
 - La mayoría de los parámetros entrenables que componen la red están contenidos únicamente en esta etapa junto con su costo computacional asociado.
 - Las capas completamente conectadas son propensas a sobreajuste comprometiendo la capacidad de generalización de la red en conjunto.

Aunque esto último ya había sido atacado mediante el concepto de abandono (*dropout*), Lin y otros (2013) propusieron una alternativa que sería utilizada por GoogLeNet y otras arquitecturas posteriores mediante la definición y uso de las capas GAP (*global average pooling*).

La idea consiste en modificar la arquitectura de la red y generar un mapa de características para cada categoría a clasificar. Posteriormente, se promedia cada mapa de forma individual reduciendo su dimensión a un único elemento y estos promedios se concatenan y forman un vector que alimenta la capa de salida *softmax*. Este enfoque ilustrado por la ver Figura brinda tres grandes beneficios: (a) impone correspondencia entre los mapas de características y las categorías a los que pertenecen al momento de entrenar el modelo, (b) elimina la necesidad de ajustar los millones de parámetros existentes en las capas completamente conectadas y (c) mejora la robustez ante traslaciones de la imagen de entrada debido a que el promedio no es afectado por la ubicación espacial del objeto.

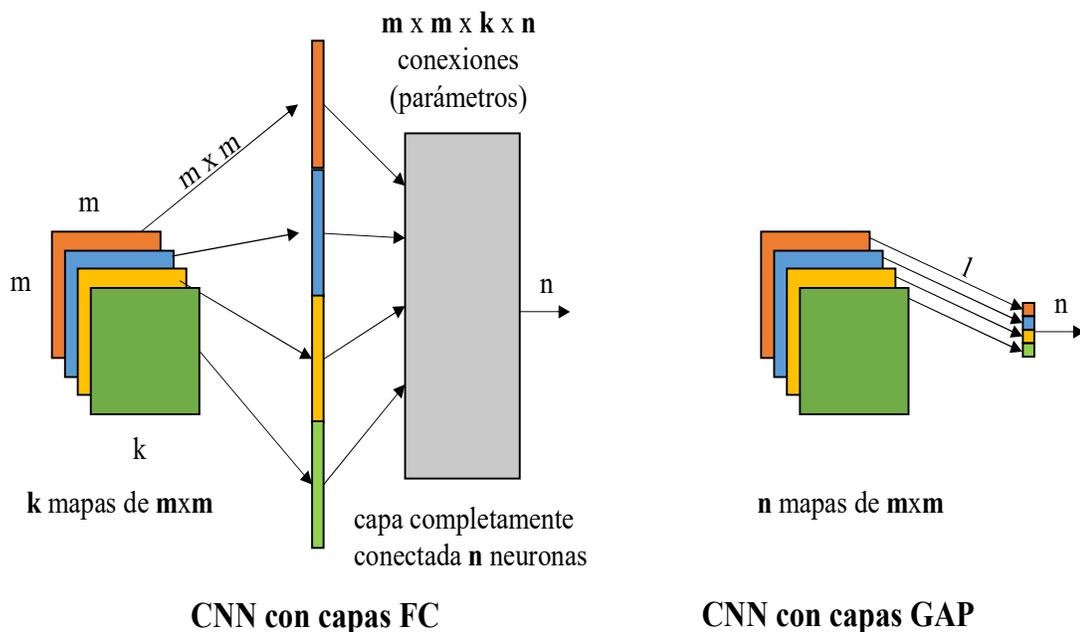


Figura . Comparación capas *fully connected* (FC) (AlexNet, ZFNet, VGGNet) y *global average pooling* GAP (NiN, GoogLeNet).

Usando la idea básica subyacente en los módulos Mlpconv, el grupo de Szegedy y otros (2015) desarrollo el concepto de los módulos *inception* que constituyen el elemento característico de GoogLeNet. Estos módulos están conformados por filtros de diferentes dimensiones los cuales operan de forma paralela sobre una misma entrada. Las salidas se concatenan y son usadas como entrada para el siguiente módulo. Debido a esta operación multiescala, la arquitectura puede aprender simultáneamente detalles muy finos (por la presencia de filtros de reducidas dimensiones) o gruesos (mediante los filtros de mayor tamaño), y características espaciales de la imagen, algo que no es posible de realizar utilizando filtros de un único tamaño.

Funcionalmente los módulos *inception* de GoogLeNet incorporan filtros de convolución de tamaños: 1x1, 3x3 y 5x5 junto a una capa de sub-muestreo de 3x3 como se aprecia en la Figura . Es de hacer notar que la inclusión de bloques de reducción dimensional (mostrados en color amarillo) fue realizada con el propósito de: (a) reducir el costo computacional de realizar las operaciones de convolución de 3x3 y 5x5 en capas con muchos filtros, (b) controlar el volumen de salida resultante en cada módulo (el cual se incrementaría notablemente con la presencia de la capa de agrupamiento concatenada a las salidas de los bloques convolucionales), y (c) incorporar no-linealidad en la operación al utilizar como función de activación la rectificación ReLU.

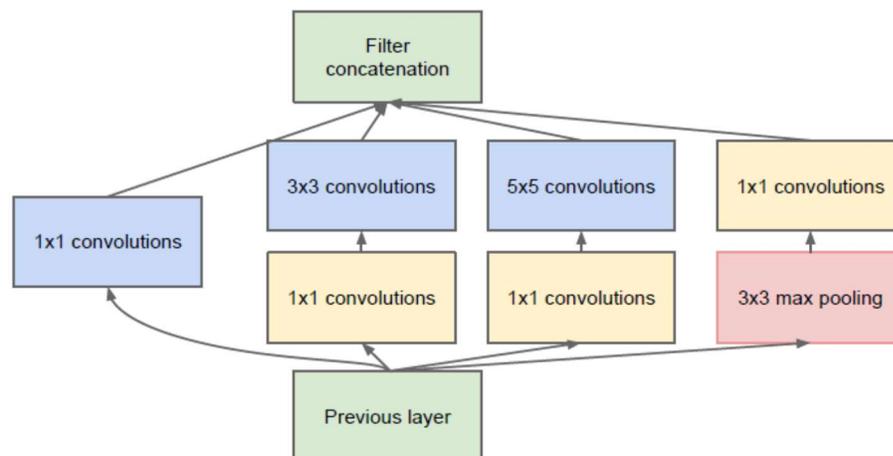


Figura . Diseño de módulo *inception* con reducción dimensional.
Tomado de Szegedy y otros (2015)

La red GoogLeNet posee una profundidad de 22 capas con parámetros entrenables y su estructura total se compone de unos 100 bloques (o capas) independientes. La Figura muestra la arquitectura de la red GoogLeNet resaltando sus tres partes distintivas:

- Tallo de la red: compuesto por 3 capas convolucionales y 2 de agrupamiento.
- Bloque central: formado 9 módulos *Inception* apilados y 2 capas de agrupamiento
- Clasificador de salida: consistente en una capa de agrupamiento promedio (*global average pooling*), una capa completamente conectada de 1000 elementos y capa de salida *softmax*.

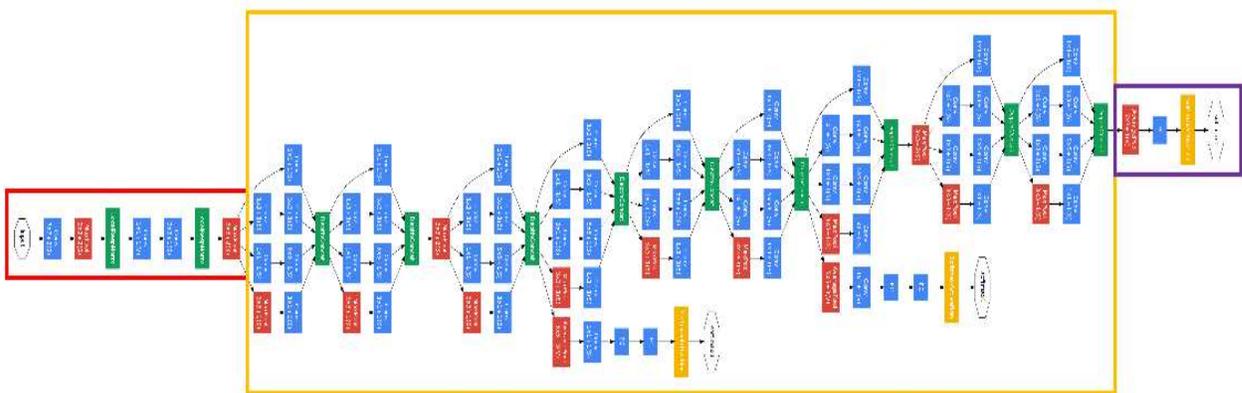


Figura . Esquema de red GoogleNet. Modificado de Szegedy y otros (2015)

Este enfoque de redes profundas con eficiencia computacional capaces de ser soportadas aún en dispositivos móviles ha marcado la tendencia en el área desde ese momento.

2.2.16.4. ResNet

He, Zhang, Ren y Sun (2016) observaron cómo simplemente apilar más capas para hacer la red más profunda no permitía, a partir de cierto punto, incrementar la exactitud al clasificar imágenes. Este incremento trae incluso problemas para la convergencia durante el entrenamiento en redes profundas compuestas de decenas de capas apiladas y en aquellas que logran hacerlo, se observa un problema de degradación el cual se manifiesta primero en una saturación de la exactitud y en su posterior desmejora,

situación que se magnifica al incrementar las capas de modelo, como se ilustra en la Figura .

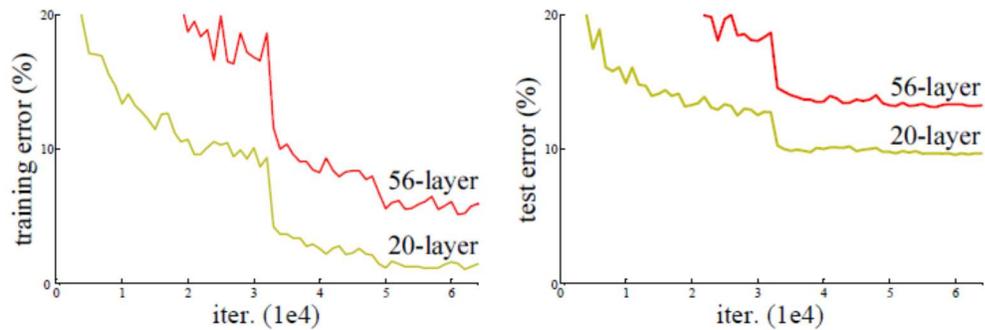


Figura . Comparación del error de entrenamiento y prueba en redes de 20 y 56 capas al clasificar imágenes de la base de datos CIFAR-10. La red más profunda mostró un mayor error al contrario de lo esperado. Tomado de He y otros (2016).

De acuerdo a estos autores, esta degradación no era producto del sobreajuste (*overfitting*) de la red, sino un problema de entrenamiento del modelo debido al desvanecimiento del gradiente del error (requerido por el algoritmo *backpropagation*) al atravesar las diferentes capas de la red. De esta forma la información del error se pierde, imposibilitando la optimización de las capas iniciales de la red y limitando en la práctica el número máximo de capas que puede tener la red convolucional.

La propuesta de He y otros (2016) a esta situación permitiría el surgimiento de una nueva generación de redes convolucionales mucho más profundas, denominadas redes residuales (ResNet), en las cuales es posible apilar cientos de capas y aún mantener un costo computacional menor que la red VGGNet. Usando estas redes lograron ganar la ILSVRC-2015 obteniendo un error menor al 4%, superando incluso lo obtenido por observadores humanos.

La idea principal tras el modelo ResNet es el concepto del bloque residual, el cual, modifica sustancialmente la función de mapeo (transferencia) que deben aprender los filtros convolucionales de la red. En una CNN tradicional la salida “z” de un conjunto

de capas se obtiene a partir de la entrada “x” mediante una función no lineal de mapeo tal que: $z = H(x)$, la cual se aproxima durante el entrenamiento del modelo.

La hipótesis propuesta por He y otros (2016) afirma que, en lugar de aprender directamente la representación de $H(x)$, es más simple para el sistema aproximar una función en diferencia $F(x)$ tal que $F(x) = H(x) - x$, ello implica que el bloque convolucional aprende a representar la diferencia existente entre $H(x)$ y la entrada “x”. Esto se logra mediante la existencia una conexión de *bypass* y una operación de adición como se ilustra en la Figura .

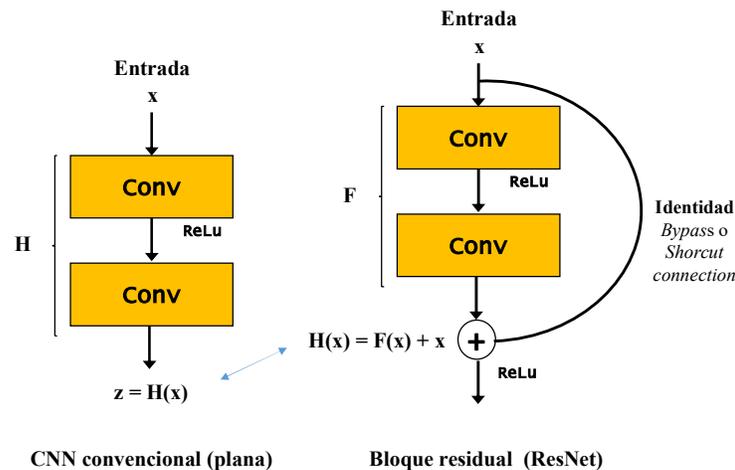


Figura . Conceptualización del bloque residual

Esta modificación permitió a los autores desarrollar las redes convolucionales ultra densas de 18, 34, 50, 101 y 152, cuyas arquitecturas se resumen en la Figura . El diseño básico de las redes ResNet resulta entonces bastante simple y consiste en apilar capas formadas por estos bloques residuales, haciendo la red simplemente más profunda y no más compleja. Como ejemplo de ello, la Figura reproduce la arquitectura ResNet-34, la cual, al evaluarse sobre los datos de prueba de la base ImageNet, fue capaz de producir resultados notablemente mejores que GoogLeNet y VGG-16. Incluso la versión menos profunda (ResNet-18) exhibe un rendimiento al menos igual a su

contraparte tradicional (sin bloques residuales) con la ventaja de lograr la convergencia del modelo más rápidamente (He y otros, 2016).

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figura . Características de las arquitecturas ResNet utilizadas para clasificar imágenes de la base de datos ImageNet. La más profunda de ellas (ResNet-152) implica una complejidad menor que el modelo VGG-16 (15×10^{19} FLOPs). Las versiones con mayor número de capas (50, 101 y 152) incluyen bloques de reducción dimensional (convolución 1x1) para reducir el tiempo de entrenamiento al controlar la complejidad del modelo. Tomado de He y otros (2016)

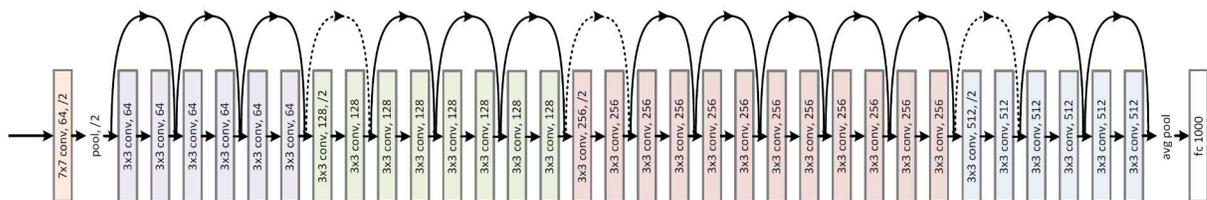


Figura . Red residual de 34 capas de profundidad ResNet-34. Tomado de He y otros (2016)

2.2.16.5. SqueezeNet

SqueezeNet es un tipo de red convolucional presentado por Iandola y otros (2016) cuyo diseño, en lugar de buscar mejorar la exactitud alcanzada por la red, se enfoca en la reducción del modelo empleado. Esto implica minimizar tanto del número de parámetros entrenables como la complejidad computacional de implementar la arquitectura (cantidad de operaciones requeridas para ejecutar la convolución).

Lograr la reducción en la arquitectura de la red sin comprometer su exactitud se traduce en al menos tres ventajas apreciables: (a) facilita un entrenamiento distribuido más

eficiente al requerir menor comunicación entre los servidores, (b) reduce el ancho de banda requerido para exportar o distribuir a los clientes actualizaciones de los productos comerciales basados en CNN haciendo posible la actualización frecuente incluso vía inalámbrica (*over-the air*) y (c) posibilita el despliegue del modelo de la red en FPGA y otros dispositivos embebidos al poder ajustarse a las restricciones de memoria presentes en dichas plataformas. Con estos beneficios en mente, los autores propusieron la arquitectura de SqueezeNet, una red capaz de igualar la exactitud lograda por AlexNet utilizando un modelo con 50 veces menos parámetros. Esto fue logrado usando principalmente los conceptos ya incluidos en las redes NiN (Lin y otros, 2013) y GoogLeNet (Szegedy y otros, 2015), es decir: modificaciones en la microarquitectura (bloques constructivos de la red) y la sustitución de las capas completamente conectadas por capas de agrupamiento promedio (GAP). Respecto a esto último al igual el enfoque usado en para NiN, SqueezeNet no incluye ninguna capa completamente conectada.

En el diseño de la microarquitectura de esta red se incorporaron dos estrategias destinadas a la reducción del modelo: (a) reemplazar la mayor cantidad posible de filtros de 3x3 por filtros de 1x1 y (b) reducir la dimensionalidad del volumen de entrada antes aplicar los filtros de 3x3. Estas estrategias se integran en el bloque constructivo de SqueezeNet denominado módulo *Fire* y se muestra en la Figura . Este se compone de dos capas convolucionales llamadas *squeeze* y *expand*. La primera de ellas está compuesta por un total de s_{1x1} filtros de convolución de 1x1, mientras que la segunda incluye e_{1x1} filtros de 1x1 junto a e_{3x3} filtros de 3x3, cuyas salidas finalmente se concatenan.

Finalmente, la arquitectura SqueezeNet en su versión original (ver Figura - arriba) incluye un total de ocho módulos *fire*, tres capas de sub-muestreo (*max pooling*), dos capas convolucionales, una capa de agrupamiento global y capa de salida *softmax*. La versión de la red utilizada en el presente trabajo de investigación y disponible para el

paquete informático MATLAB (ver Figura - abajo), corresponde una actualización denominada SqueezeNet v1.1. Esta incorpora ligeras modificaciones respecto a la arquitectura original en el tamaño y cantidad de filtros de la primera capa de convolución y la ubicación de las capas de agrupamiento (*maxpool*). El modelo resultante requiere 2.4 veces menos operaciones de cómputo sin perder exactitud según describe el repositorio de sus autores disponible en la URL: https://github.com/DeepScale/SqueezeNet/tree/master/SqueezeNet_v1.1.

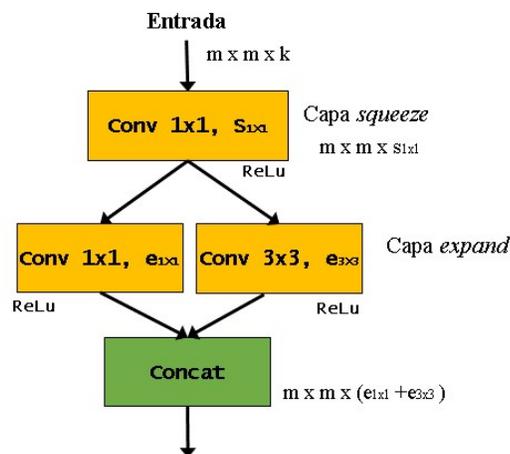


Figura . Estructura Módulo *fire* usado para la construcción de la red SqueezeNet

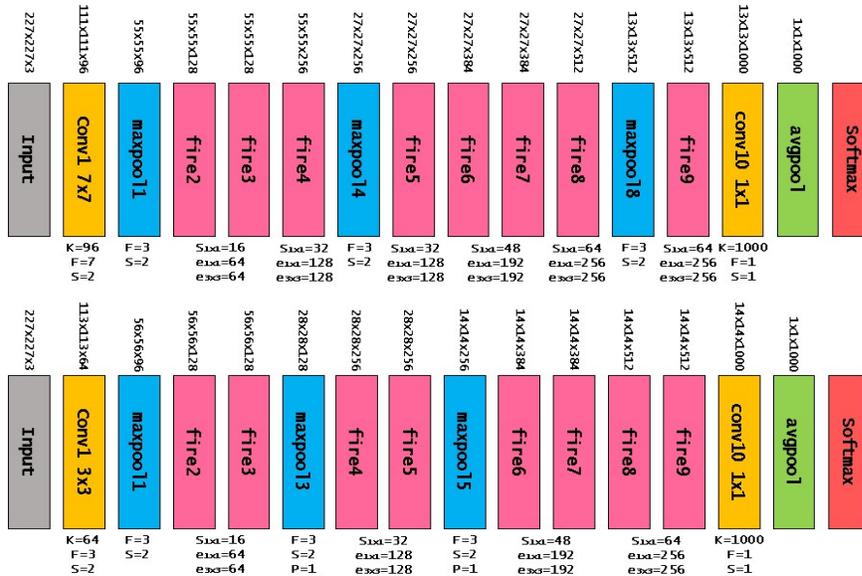


Figura . Arquitectura SqueezeNet. Arriba: SqueezeNet V1.0 publicada por (Iandola y otros, 2016). Abajo: SqueezeNet V1.1 implementada en MATLAB.

En Rawat y Wang (2017) se presenta una revisión exhaustiva de las CNN usadas para la clasificación de imágenes abarcando desde su progresión histórica y contribución en el renacimiento de los postulados del aprendizaje profundo hasta su rápido avance en los últimos años, tendencias y desafíos aún abiertos a la investigación.

2.2.17. Transferencia de conocimiento (*Transfer Learning*)

La minería de datos y el aprendizaje automático han alcanzado un éxito significativo en muchos campos de la ingeniería de conocimiento como la clasificación, regresión y agrupamiento de datos. Sin embargo, muchos métodos de aprendizaje automático trabajan bien únicamente sobre la concepción de que los datos de entrenamiento y prueba se encuentran en el mismo espacio de características y poseen la misma distribución. Cuando esta cambia, los modelos estadísticos deben ser reconstruidos desde cero utilizando nuevos datos, no siendo esto siempre posible en las aplicaciones del mundo real (Pan & Yang, 2010). Para estos los casos en los cuales los dominios, tareas de aprendizaje y/o distribución de los datos cambian, se han desarrollado técnicas que permiten reducir la necesidad y esfuerzo de recolectar nuevamente datos de entrenamiento a la vez que minimizan el tiempo y recursos requeridos para lograr el ajuste de los modelos computacionales. Estas se agrupan bajo el nombre de transferencia de conocimiento o transferencia de aprendizaje (*knowledge transfer* o *transfer learning*)

El término transferencia de conocimiento (o de aprendizaje), se refiere entonces a la capacidad de utilizar un modelo computacional pre-entrenado para realizar una tarea determinada como punto inicial de la construcción y entrenamiento de un modelo especializado en otra tarea. Esta capacidad aparentemente inherente al proceso de aprendizaje humano, mediante la cual la extracción de conocimiento relevante producto de las experiencias de aprendizaje pasadas se aplica al enfrentar nuevas tareas facilitando su realización, no está presente en la mayoría de los algoritmos de aprendizaje automático comúnmente utilizados, los cuales aprenden a realizar tareas

aisladas entre sí (Torrey & Shavlik, 2009). La Figura ilustra la diferencia fundamental en los procesos de aprendizaje comúnmente empleados por los algoritmos de *machine learning* frente al enfoque basado en *transfer learning*.

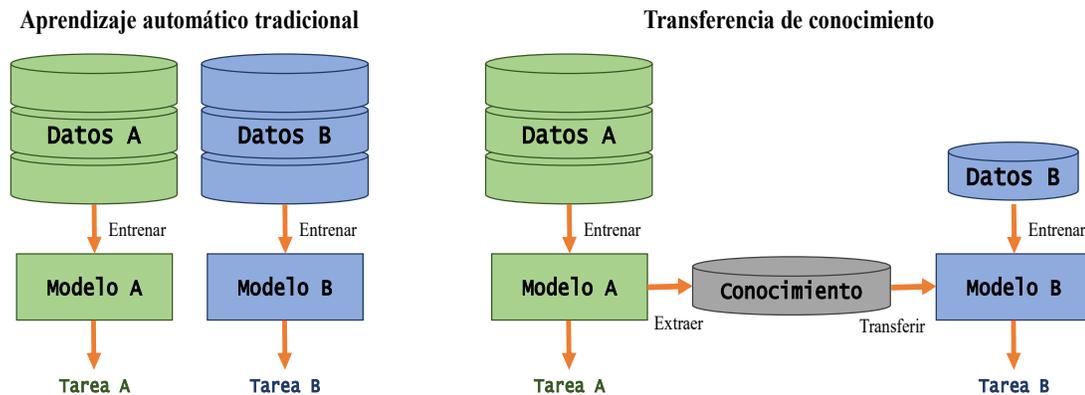


Figura . Procesos de entrenamiento usados en *machine learning* y mediante *transfer learning*

Torrey y Shavlik (2009) describen tres beneficios obtenidos al utilizar transferencia de conocimiento de forma exitosa:

- **Rendimiento inicial más alto:** esto es la exactitud alcanzada por el modelo computacional utilizando únicamente el conocimiento transferido y antes de realizar cualquier entrenamiento posterior. En el caso de las redes neuronales esto se traduce a que la inicialización de los pesos sinápticos usando los datos de un modelo previamente entrenado resulta mejor que utilizar valores aleatorios.
- **Convergencia más rápida:** la pendiente de la curva de aprendizaje resulta más pronunciada y por tanto el tiempo requerido para el entrenamiento se reduce.
- **Mayor exactitud final:** una vez entrenado el modelo, el nivel de rendimiento obtenido es superior al alcanzado sin utilizar *transfer learning*.

Para ilustrar lo anterior, en la Figura se compara el rendimiento obtenido por un sistema entrenado desde cero (línea sólida) y otro en el cual se ha usado transferencia de aprendizaje (línea punteada). Los beneficios mencionados anteriormente se manifiestan al observar que este último presenta un valor inicial, final y pendiente mayor.

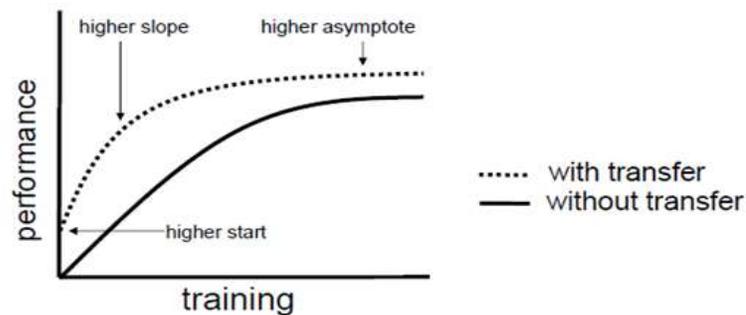


Figura . Mejoramiento del aprendizaje por medio de transferencia de conocimiento.
Tomado de Torrey y Shavlik (2009)

Al emplear redes convolucionales, la transferencia de conocimiento puede hacerse de dos maneras distintas: usando la red como extractor de rasgos o mediante el ajuste de pesos (*fine tuning*) del modelo. La idea conceptual detrás de cada uno de estos enfoques se ilustra en la Figura . Cuando se hace *fine tuning*, los pesos de algunas capas iniciales del modelo pre-entrenado de CNN se preservan o retienen mientras que se ajustan (re-entrenan) los pesos de las capas finales. Esto se debe a que los rasgos extraídos por las primeras capas son genéricos y aplicables a diferentes tareas, mientras que aquellos extraídos por las capas finales son más específicos y por tanto pueden ser especializados para representar al nuevo conjunto de datos. En contraposición, al usar la red convolucional como extractor de rasgos no se realiza ningún ajuste de los parámetros del modelo, ya que este enfoque se basa en la idea de utilizar la capacidad de representación de la red para extraer rasgos de cualquiera de sus capas y emplearlos para entrenar un clasificador especializado en la nueva tarea.

Se puede definir el tipo de transferencia de conocimiento empleada en función de dos factores: (a) el tamaño del conjunto de imágenes disponible para entrenar al sistema en la nueva tarea y (b) la similitud existente entre éstas y el conjunto de entrenamiento original del modelo, pudiendo identificarse cuatro posibles escenarios mostrados en la Tabla (Li, Johnson, & Yeung, 2017).

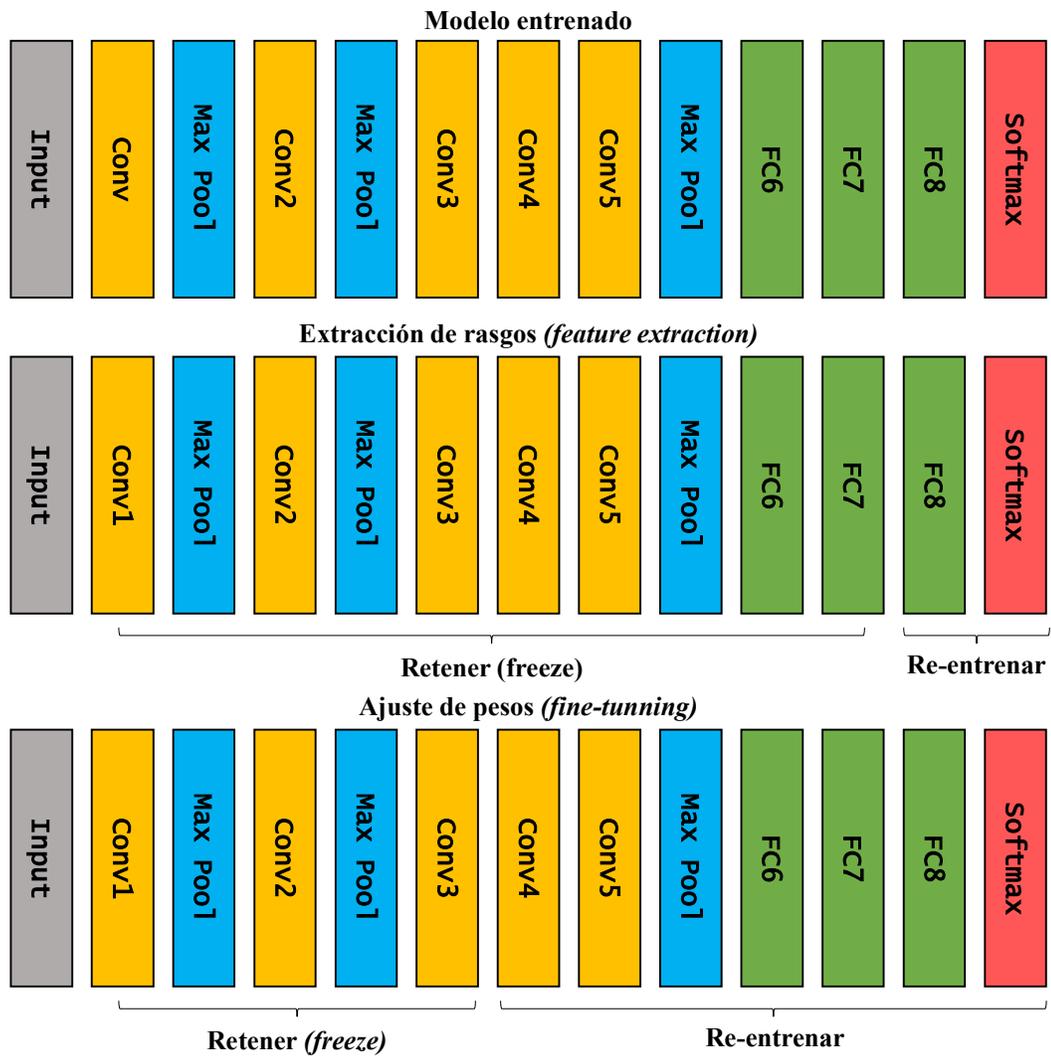


Figura . Formas de realizar transferencia de aprendizaje en redes convolucionales. Arriba: modelo previamente entrenado. Centro: CNN como extractor de rasgos. Abajo: ajuste de pesos.

Tabla

Método de transferencia de conocimiento y capas a utilizar en la red convolucional

Tamaño del conjunto para la nueva tarea	Semejanza con datos originales de entrenamiento	
	Alta	Baja
Pequeño	Extracción de rasgos en las capas finales	Extracción de rasgos en las capas medias o iniciales
Grande	Ajuste de pesos solamente algunas capas (finales)	Ajuste de pesos en más capas (finales y medias)

Nota. Adaptado de Li et al. (2017), *Convolutional Neural Networks for Visual Recognition: Training Neural Networks, Part 2.*

CAPÍTULO III

MARCO METODOLÓGICO

3.1. NIVEL DE INVESTIGACIÓN

El presente Trabajo Especial de Grado titulado “Desarrollo de un sistema de análisis automático para la segmentación, clasificación y conteo de leucocitos en imágenes digitales de frotis de sangre periférica”, es una investigación de alcance descriptivo debido a que busca especificar las propiedades y características del objeto sometido a análisis (Hernández, Fernández, & Baptista, 2014, p. 92). En función del propósito del trabajo, se considera una investigación aplicada ya que utiliza conocimientos existentes en pro de la solución del problema observado. Tiene un carácter cuantitativo, debido a que la solución propuesta se evaluó numéricamente utilizando medidas estadísticas para determinar la calidad de los resultados obtenidos. Se enmarca, utilizando como referencia el “Manual de Trabajos de Grado de Especialización, Maestría y Tesis Doctorales de la Universidad Pedagógica Experimental Libertador” (Universidad Pedagógica Experimental Libertador, 2003, p. 14), en la modalidad de Proyectos Especiales, definidos estos como:

“Trabajos que lleven a creaciones tangibles, susceptibles de ser utilizadas como soluciones a problemas demostrados, o que respondan a necesidades e intereses de tipo cultural. Se incluyen en esta categoría los trabajos de elaboración de libros de texto y de materiales de apoyo educativo, el desarrollo de software, prototipos y de productos tecnológicos en general, así como también los de creación literaria y artística”.

3.2. DISEÑO DE INVESTIGACIÓN

El diseño de investigación es la estrategia utilizada para responder al problema planteado (Arias, 2006, p. 27); siendo en este caso un diseño de campo, debido a que se utilizaron imágenes de muestras sanguíneas obtenidas de pacientes reales en un ambientes clínicos; se observaron los procedimientos, técnicas y equipos empleados en el laboratorio; se consultó y entrevistó a expertos en las áreas de hematología

y bioanálisis. Los datos obtenidos fueron utilizados durante el desarrollo, pruebas, validación, comparación y análisis de los resultados finales.

3.2.1. PROCEDIMIENTO METODOLÓGICO

Para el cumplimiento de los objetivos propuestos, el trabajo de investigación se desarrolló en seis etapas o fases, las cuales se explican a continuación:

Fase I - Descripción de las características morfológicas de las principales células presentes en el frotis sanguíneo que permiten su identificación visual. Esta etapa abarcó un estudio bibliográfico destinado a describir el proceso de preparación del frotis de sangre periférica y señalar los elementos característicos que permiten distinguir cada uno de los tipos de células que se observan en éste. Implicó la realización de las siguientes actividades:

- Arqueo de información en libros y atlas de hematología, artículos científicos y otros medios electrónicos sobre los tipos de células sanguíneas, concentraciones y rasgos fisiológicos normales.
- Revisión del proceso experimental de preparación y análisis manual del frotis periférico usado en los laboratorios clínicos.
- Arqueo de información sobre el proceso de tinción y su influencia en las imágenes observadas en el microscopio.
- Tabulación de las características físicas más relevantes de los tipos de células observables en el frotis sanguíneo haciendo especial hincapié en los leucocitos.

Fase II - Creación de un banco de imágenes digitales con muestras catalogadas por expertos. Esta fase comprendió las actividades destinadas a proveer y catalogar el conjunto de imágenes experimentales que fueron empleadas para entrenar y verificar el sistema computacional desarrollado, entre ellas:

- Búsqueda y catalogación de bancos de imágenes experimentales de acceso gratuito de frotis sanguíneo disponibles en la red.
- Revisión de estudios previos con el fin de evaluar resolución, dimensiones y otras especificaciones asociadas a las imágenes empleadas en este tipo de investigaciones.
- Adecuación del equipo de microscopía disponible en el laboratorio para la captura de imágenes de frotis sanguíneo.
- Preparación del frotis y/o selección las láminas pertenecientes a individuos sanos.
- Captura de imágenes en instalaciones del laboratorio.
- División de las imágenes disponibles en sub-conjuntos de entrenamiento y prueba para las operaciones de segmentación y clasificación.
- Clasificación con presencia del experto médico de las imágenes utilizadas en el proyecto y eliminación de muestras defectuosas.

Fase III - Análisis de técnicas de segmentación de imágenes digitales utilizadas en la identificación células sanguíneas para seleccionar el algoritmo a ser implementado. Esta etapa abarcó las siguientes actividades:

- Arqueo de información en libros, revistas científicas y otros medios electrónicos sobre diferentes técnicas de segmentación y sus fundamentos matemáticos.
- Estudio de diferentes técnicas comúnmente empleadas para la segmentación de imágenes digitales y su implementación utilizando MATLAB.
- Arqueo de información sobre técnicas de segmentación de imágenes aplicadas específicamente al frotis de sangre periférica.
- Elaboración de pruebas de segmentación mediante técnicas clásicas documentadas en la literatura.
- Selección de algoritmo de segmentación a utilizar considerando como criterios relevantes la exactitud del resultado obtenido, costo computacional asociado y factibilidad de adaptación ante imágenes de características distintas.

- Estudio de las herramientas, librerías y funciones requeridas para la codificación e implementación del algoritmo seleccionado en ambiente MATLAB.
- Captura de imágenes explicativas del proceso de segmentación y documentación de resultados.

Fase IV – Selección y extracción de un conjunto de rasgos característicos de los leucocitos que permitan su identificación mediante un sistema computacional. En esta etapa de la investigación, se realizó la selección de los rasgos a utilizar para identificar los leucocitos en la imagen así como el establecimiento de los pasos requeridos para su extracción para su posterior clasificación. Comprendió la realización de las siguientes actividades:

- Análisis bibliográfico de rasgos utilizados en trabajos previos.
- Establecimiento de la metodología de extracción de rasgos usando redes neuronales convolucionales.
- Estudio de las bibliotecas y funciones requeridas para su implementación en MATLAB
- Codificación y pruebas del algoritmo de extracción.

Fase V- Implementación de un algoritmo de clasificación para la identificación automática de cuatro (04) tipos de leucocitos (neutrófilos segmentados, linfocitos y monocitos y eosinófilos) mediante técnicas de aprendizaje automático. En esta etapa se realizaron las siguientes actividades:

- Selección y estudio de las técnicas de clasificación a ser empleadas. Para este trabajo, se implementaron clasificadores basados tanto en máquinas de soporte vectorial como en redes neuronales convolucionales de aprendizaje profundo.
- Elaboración de códigos e interfaz gráfica de usuario para la carga de imágenes de entrenamiento y evaluación.
- Selección y pre-procesamiento de las imágenes utilizadas para el entrenamiento.

- Elaboración y codificación del algoritmo.
- Aplicación del algoritmo en un conjunto de datos pre-seleccionados y documentación de resultados de aciertos y fallas.
- Elaboración de matriz de confusión y cálculo de métricas de funcionamiento en las diferentes variantes implementadas para la clasificación de las imágenes para su análisis posterior.

Fase VI - Evaluación del funcionamiento del sistema de clasificación utilizando muestras de individuos sanos. En esta etapa se validaron los resultados obtenidos, abarcando las siguientes actividades:

- Cálculo de métricas para la evaluación del funcionamiento del sistema de clasificación ante diferentes conjuntos de imágenes.
- Comparación de los resultados obtenidos por el sistema computacional propuesto y los indicados por un experto del área médica al realizar el conteo de glóbulos blancos en imágenes con múltiples células presentes.
- Estudio comparativo de los resultados experimentales con los indicados por otros trabajos de investigación y establecimiento de conclusiones.

3.3. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS

En función de los objetivos planteados, los datos recolectados se obtuvieron mediante el uso de las siguientes técnicas:

- Entrevistas no estructuradas con expertos del área médica con respecto a los procedimientos empleados para el análisis manual de las muestras en su campo de ejercicio.
- Observación en campo del proceso de conteo de glóbulos blancos y captura de imágenes para la construcción de una de las bases de datos a utilizar.
- Recopilación de bases de datos de imágenes de frotis sanguíneo publicadas por grupos de investigación del área.

- Análisis documental de material bibliográfico en las áreas de: hematología, procesamiento de imágenes digitales, aprendizaje automático y programación en MATLAB; obtenido de libros, manuales, artículos de revistas, memorias de congresos, informes de proyectos, trabajos de grado y otras fuentes.

Para la recolección de los datos de campo fue necesario del uso de los siguientes instrumentos:

- Cámara fotográfica digital y microscopio óptico para la adquisición de imágenes.
- Acople mecánico para sujeción de cámara digital al microscopio.
- Contador manual de células sanguíneas.
- Computador personal para realizar el registro de las células observadas en las imágenes por parte del experto médico.

3.4. TÉCNICAS DE PROCESAMIENTO Y ANÁLISIS DE DATOS

Se solicitó al experto médico describir verbalmente las características morfológicas, geométricas y/o colorimétricas relevantes de las imágenes empleadas, a fin de comprender los rasgos distintivos utilizados durante el proceso de identificación de leucocitos. Las células fueron observadas de manera individual por expertos del área y las imágenes separadas según el tipo de leucocito presente. En los casos en los cuales la fotografía contenía más de una célula, estas fueron separadas manualmente antes de su clasificación. Los datos recolectados fueron visualizados mediante el uso de tablas, gráficos de barras y circulares para facilitar su interpretación.

Se utilizó el programa MATLAB para crear programas destinados a:

- Segmentar el núcleo y citoplasma de los leucocitos y extraer la región de interés que contiene a la totalidad de la(s) célula(s) presentes en una imagen.
- Incrementar el volumen de datos (*data augmenting*) mediante la generación de imágenes artificiales obtenidas por medio de operaciones de rotación, traslación, ajustes de brillo y contraste, entre otras.

- Utilizar redes neuronales convolucionales para la clasificación de leucocitos mediante dos técnicas del área:
 - Extracción automática de rasgos.
 - Ajuste de pesos mediante el re-entrenamiento de la red neuronal.

Para la obtención de resultados se definió un conjunto de 7 experimentos destinados a evaluar el funcionamiento de diferentes configuraciones o variantes del clasificador de leucocitos basado en imágenes, utilizando para ello la matriz de confusión del sistema como indicador general de la operación del clasificador al someterse a datos de entrenamiento, validación y prueba. A partir de esos resultados construyeron tablas y gráficos descriptivos con los valores de exactitud, precisión, sensibilidad, especificidad, tasas de acierto y falla obtenidas de cada clase para así efectuar un análisis comparativo entre los resultados obtenidos y los indicados por expertos del área en un conjunto de imágenes de prueba. Finalmente se realizó un breve análisis comparativo con algunos resultados presentados en otros trabajos de investigación.

Para el desarrollo de los algoritmos computacionales descritos en la presente investigación fue utilizado un computador personal con sistema operativo Windows® 7 de 64bits, equipado con CPU Intel® Core™ i3 2100–3.10 Ghz, 8 GB de memoria RAM DDR3 y tarjeta aceleradora grafica NVIDIA ® GeForce GTX780 de 3 GB de memoria DDR5. Se verificó que la solución propuesta es capaz de trabajar satisfactoriamente en un equipo de menores prestaciones, con mismo procesador pero utilizando únicamente 4 GB de memoria RAM DDR3 y tarjeta de video integrada Intel Graphics® serie 4000.

En las pruebas finales también se utilizó un computador con sistema operativo Windows® 10, equipado con procesador Intel® Core™ i5 3230M-2.60 Ghz, 4GB de memoria RAM DD3 y tarjeta de video integrada Intel® HD Graphics 4000.

CAPÍTULO IV

RESULTADOS

4.1. DESCRIPCIÓN Y PREPARACIÓN DE LOS DATOS EXPERIMENTALES

Para el desarrollo del presente trabajo se utilizó un total de información proveniente de seis (6) bases de datos con imágenes de leucocitos captadas bajo condiciones de iluminación, tinción, resolución, nitidez, coloración, enfoque y aumento diferentes, ello con el fin de lograr una representación adecuada de diversas variantes que podrían degradar la calidad de la identificación de las células en estudio. A continuación se describen las características más importantes de cada conjunto de imágenes.

4.1.1. ALL-IDB (*Acute Lymphoblastic Leukemia Image Database*)

Esta base de datos, descrita por Ruggero y otros (2011), se obtuvo gracias al profesor Fabio Scotti de la Universidad de Milán, Italia (<https://homes.di.unimi.it/scotti/all/>). Las imágenes fueron capturadas con una cámara Canon PowerShot G5, a una resolución de 2594 x 1944, con diferentes niveles de magnificación del microscopio en el rango de 300X a 500X. Los datos se encuentran divididos en dos conjuntos: ALL_IDB1 y ALL_IDB2, almacenados en formato JPG y TIFF, respectivamente.

ALL-IDB1 puede ser usada para probar algoritmos de segmentación, procesamiento de imágenes o sistemas de clasificación. Se compone de 108 imágenes que contienen unas 39000 células sanguíneas.

El conjunto ALL-IDB2 fue diseñado para probar sistemas de clasificación y contiene 260 imágenes extraídas de células pertenecientes al conjunto ALL-IDB1 las cuales fueron centradas y cortadas aun tamaño de 256x256 pixeles. La Figura muestra un ejemplo de las imágenes disponibles en esta base de datos.

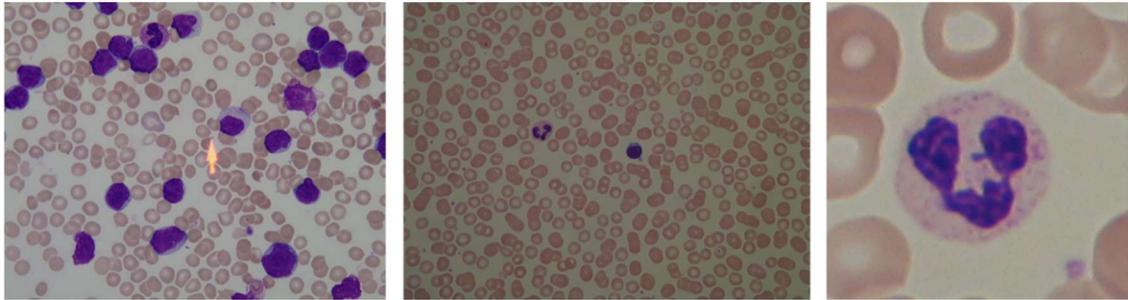


Figura . Imágenes base datos ALL-IDB. **Izquierda:** Paciente con ALL (conjunto ALL-IDB1). **Centro:** Individuo sano (conjunto ALL-IDB1). **Derecha:** Célula del conjunto ALL-IDB2

4.1.2. IUMS-IDB

La base de datos IUMS-IDB (Sarrafzadeh et al., 2014) es de libre acceso y pertenece a la Universidad de Ciencias Médicas de Irán (IRAN University of Medical Science). Contiene un total de 195 imágenes de 10 individuos, capturadas con una cámara Nikon V1 montada sobre un microscopio óptico Nikon modelo ECLIPSE 50i con lente de aumento de 100X. Las fotografías fueron originalmente almacenadas en formato JPG con 3872x2592 píxeles y 300 dpi de resolución. En el sitio web de origen (<http://misp.mui.ac.ir/fa/download>) puede descargarse estas imágenes de alta resolución o una versión de menor resolución 775 x 519 píxeles, esta fue la versión utilizada en el presente trabajo. La Figura muestra ejemplos de imágenes disponibles en esta base de datos.

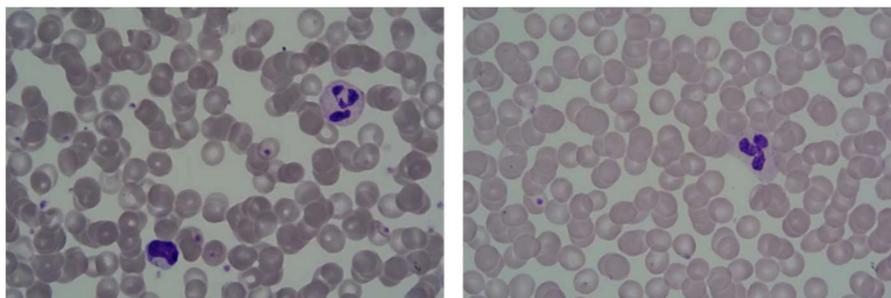


Figura . Ejemplos de imágenes base datos IUMS-IDB

4.1.3. SMC-IDB

Este conjunto, publicado y utilizado por Mohamed, Far & Guaily (2012), se compone de un total de 367 imágenes de 640 x 480 píxeles y puede ser descargado desde el sitio Web de Mathworks (www.mathworks.com/matlabcentral/_leexchange/36634). La

base de datos incluye además la segmentación manual de los núcleos de los leucocitos presentes. En la Figura se muestra un ejemplo de las imágenes disponibles en este conjunto de datos (izquierda) y la segmentación provista por los autores (derecha).

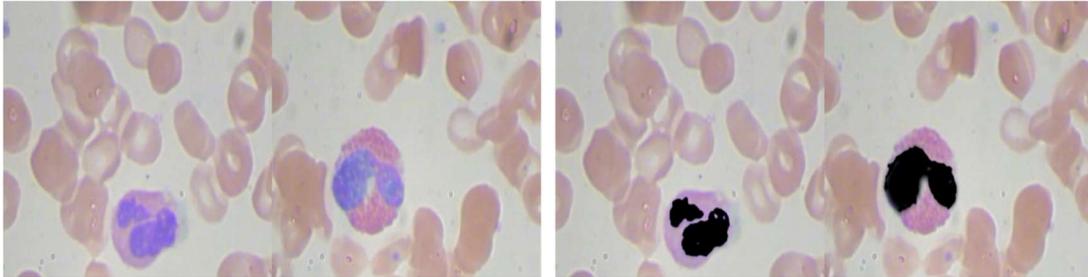


Figura . Ejemplos de imágenes base datos SMC-IDB

4.1.4. LISC: *Leukocyte Images for Segmentation and Classification*

La base de datos LISC (Rezatofghi & Soltanian-Zadeh, 2011) incluye imágenes de frotis sanguíneo de individuos sanos y fue creada para permitir la evaluación comparativa de diferentes técnicas de segmentación del núcleo y citoplasma, y reconocimiento de leucocitos en imágenes hematológicas. Los datos fueron adquiridos de 8 individuos sanos usando una cámara digital Sony modelo No. SSCD50AP y un microscopio óptico Axioskope 40 con lente de aumento de 100X siendo almacenadas como archivos .BMP con resolución de 720 x 576 píxeles. El total de 400 imágenes se encuentra dividido en un conjunto principal de 250 muestras y un conjunto adicional de 150 muestras. Este último contiene imágenes de uno o más leucocitos en cada fotografía y se utilizó para verificar el funcionamiento de la propuesta presentada en el presente trabajo.

Para el conjunto principal los autores proveen la clasificación del tipo de célula (basófilos, eosinófilos, linfocito, monocito o neutrófilo segmentado) realizada por un médico hematólogo y la máscara de segmentación correspondiente al área total ocupada por cada leucocito (realizado esto manualmente por un experto). En este conjunto se encuentra un grupo de 8 imágenes que contiene múltiples células. La Figura muestra las imágenes típicas encontradas en esta base de datos.

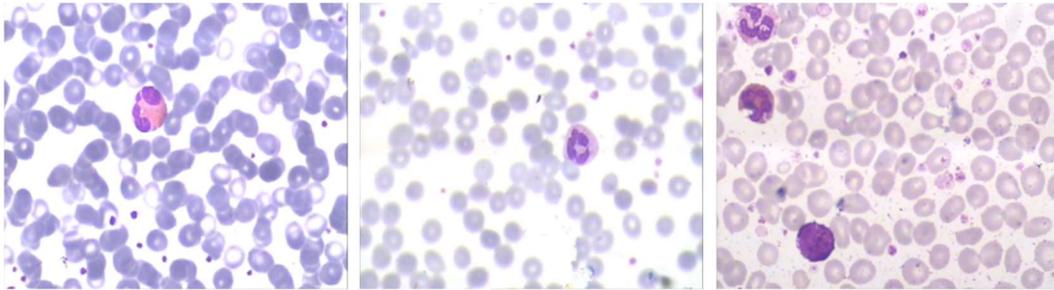


Figura . Ejemplos de imágenes base datos LISC

4.1.5. WBC Image Dataset (WBC-ID)

Esta base de datos utilizada por Zheng, Wang, Wang y Liu Zheng (2018), contiene un total de 400 imágenes en formato de mapa de bits (BMP) y se encuentra dividida en dos conjuntos: DataSet1 y DataSet2. Para ambos los autores proveen la segmentación manual del núcleo y citoplasma de cada célula.

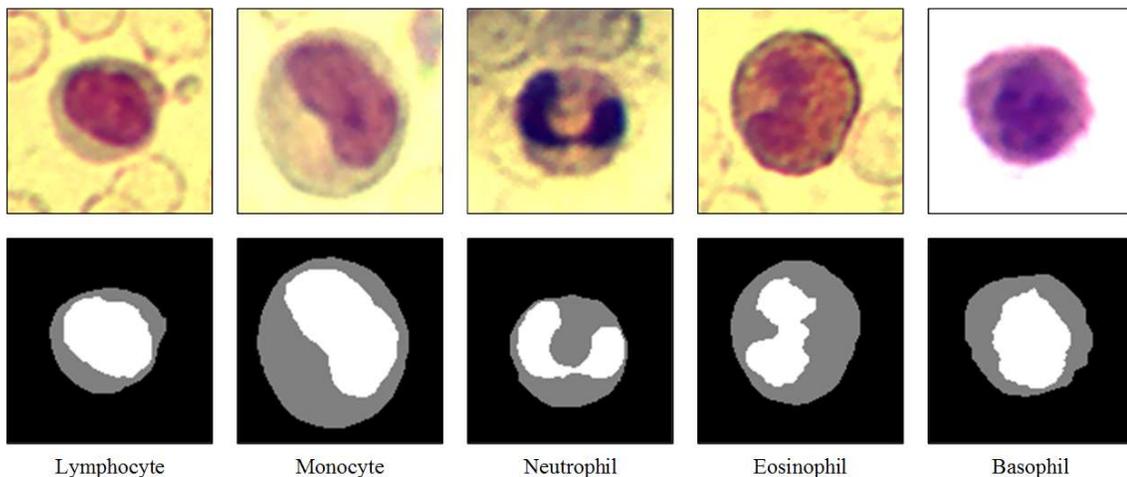


Figura . Ejemplos de imágenes base datos usada por Zheng y otros (2018) – DataSet1. Arriba: Imagen de diferentes células. Abajo: Segmentación de núcleo y citoplasma correspondiente. Recuperado de sitio Web del proyecto https://github.com/zxaoyou/segmentation_WBC

El DataSet1 fue obtenido de la empresa Jiangxi Tecom Science Corporation y contiene 300 imágenes de baja resolución (120 x 120 píxeles). Las fotografías fueron capturadas utilizando un microscopio óptico motorizado con auto enfoque modelo N800-D y una cámara Motic Moticam Pro 252A. Posee la particularidad de utilizar un colorante hematológico completamente diferente al usado normalmente para la tinción del frotis sanguíneo, lo cual dio origen a imágenes con fondo generalmente amarillo, en lugar de

blanco como se puede observar en la Figura . Para efectos de la presente investigación, el DataSet1 fue dividido en dos sub-conjuntos: JTSC-I y JTSC-II en función del color del fondo, teniendo 200 y 100 imágenes respectivamente.

El DataSet2 compuesto por 100 imágenes de 300x300 píxeles fue obtenido desde la página web de CellaVision (<http://blog.cellavision.com>). Las imágenes poseen una coloración generalmente púrpura como se observa en la Figura .

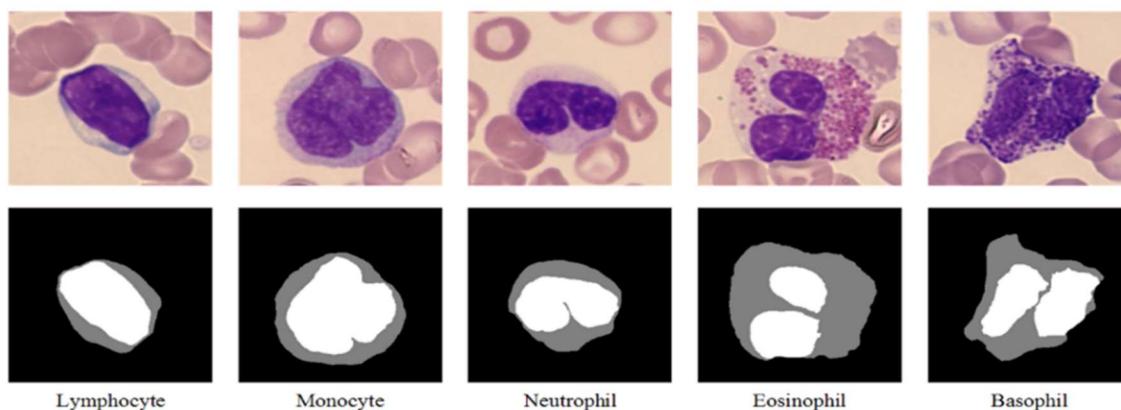


Figura . Ejemplos de imágenes base datos usada por Zheng y otros (2018) – Dataset2. Arriba: Imagen de diferentes células. Abajo: Segmentación de núcleo y citoplasma correspondiente. Recuperado de sitio Web del proyecto https://github.com/zxaoyou/segmentation_WBC

4.1.6. Base de datos propia (BDL)

La base de datos BDL fue creada como aporte del presente trabajo, utilizando un microscopio marca OMAX, modelo M82E con lente de magnificación 100X y una cámara G.E Modelo E1235. Las imágenes resultantes se encuentran en diferentes tamaños desde 150x150 hasta 1024x1024 píxeles; son ruidosas, con poca nitidez, grandes variaciones en el color, contraste e iluminación y en muchos casos desenfocadas. Su importancia para este proyecto radicó en permitir evaluar el desempeño del sistema propuesto ante imágenes de baja calidad, como las que se podrían capturar en un ambiente de trabajo real, sin utilizar equipos de adquisición especializados y compararlo con estudios previos que se han centrado en imágenes de

calidad similar como el trabajo de Habibzadeh y otros (2013). En la Figura son mostradas algunas fotografías extraídas de este conjunto de datos.

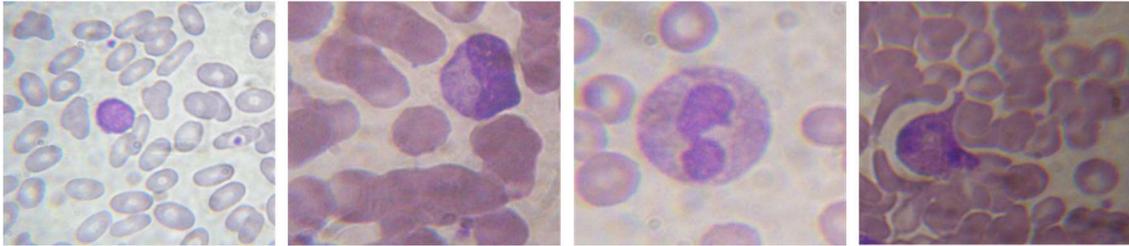


Figura . Ejemplos de células encontradas en la base datos de imágenes propias (BDL)

A manera resumen, la Tabla muestra imágenes típicas encontradas en cada conjunto de datos y se pueden apreciar las notables diferencias existentes entre imágenes que corresponde al mismo tipo de leucocito.

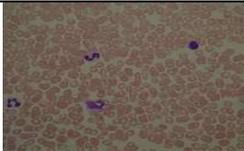
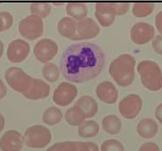
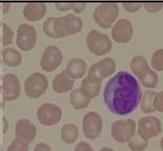
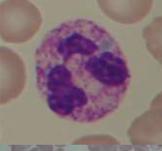
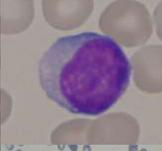
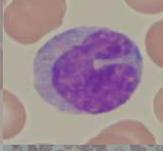
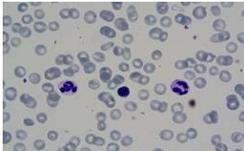
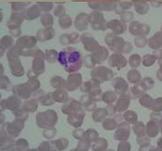
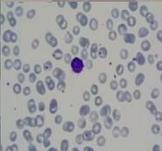
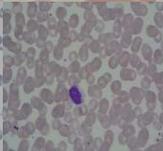
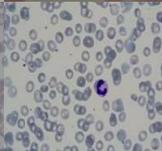
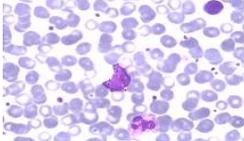
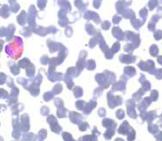
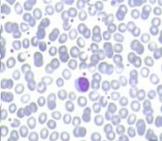
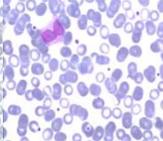
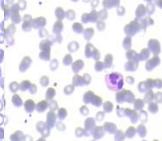
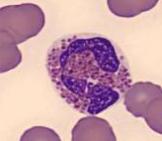
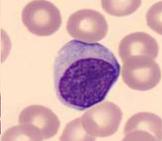
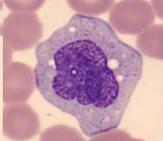
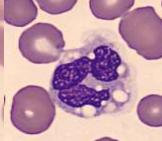
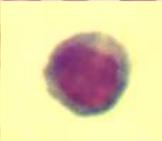
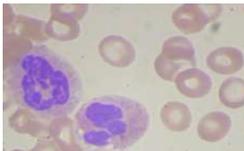
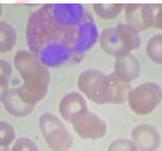
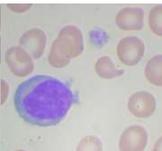
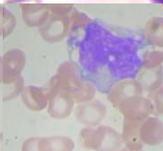
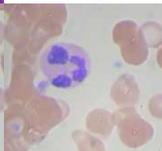
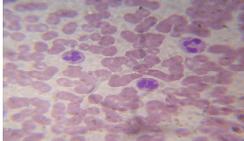
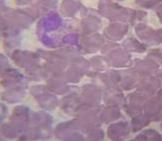
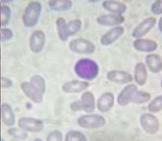
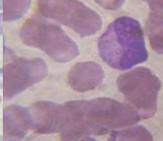
4.1.7. Banco de imágenes de múltiples células

Está compuesto por cuatro conjuntos de imágenes (M-ALL-IDB, M-LISC, M- IUMS y M-BDL). Cada uno de ellos contiene fotografías con uno o más leucocitos presentes, muchas de las cuales no fueron utilizadas durante el entrenamiento del clasificador. Aquellas células que si intervinieron en el entrenamiento se les aplicó algún pre-procesamiento manual para este fin, por lo cual se consideró válido usarles nuevamente. Este banco de datos fue seleccionado para hacer las pruebas finales y comparar la salida obtenida por el sistema de clasificación contra el criterio del experto humano.

4.1.8. Preparación y pre-procesamiento de las imágenes

Todas las imágenes de leucocitos utilizadas en este trabajo corresponden a individuos sanos, aquellas células consideradas como patológicas o provenientes de pacientes con diagnostico o sospecha de diagnóstico de algún padecimiento capaz de alterar la morfología de sus glóbulos blancos fueron inmediatamente descartadas.

Tabla
Clasificación de las imágenes extraídas de los datos experimentales

Conjunto	Múltiples células	Eosinófilos	Linfocitos	Monocitos	Neutrófilos segmentados
ALL-IDB1					
ALL-IDB2	No Aplica				
IUMS-IDB					
LISC					
WBC-ID DataSet2	No aplica				
WBC-ID DataSet1 JTSC-I	No aplica				
WBC-ID DataSet1 JTSC-II	No aplica				
SMC-IDB					
BDL					

De forma manual, con ayuda de un experto del área médica, se inspeccionó visualmente las fotografías clasificando y etiquetando todas las células presentes en cada una de ellas. Aquellas muestras en las cuales el experto manifestó tener dudas debido a mala coloración del frotis o debido a poca calidad de la imagen (desenfoco) o pobre iluminación fueron eliminadas. En las fotografías que contenían más de un tipo de célula, estas fueron separadas manualmente para organizarlas en categorías individuales. No se aplicó ningún recorte o centrado específico a las imágenes resultantes. La composición de las imágenes obtenidas en cada conjunto de datos se muestra en la Figura .

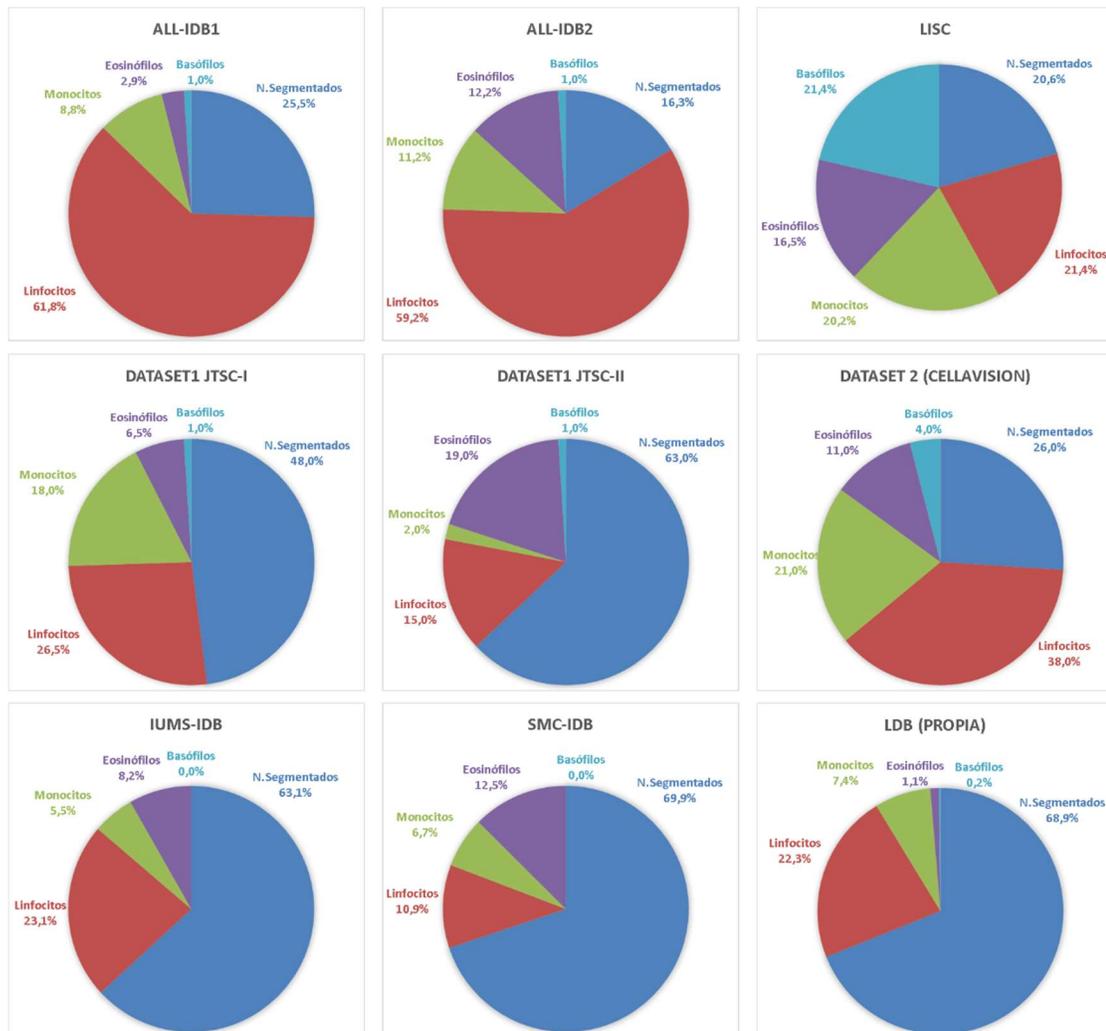


Figura . Distribución de los tipos de leucocitos en cada conjunto de imágenes

Al analizar la distribución de los tipos de células existentes en las imágenes disponibles (Figura), se optó por seleccionar 4 tipos de leucocitos a ser reconocidos por el sistema propuesto: Neutrófilos segmentados, linfocitos, monocitos y eosinófilos, siendo descartada la clasificación de basófilos por la falta de muestras suficientes (un 4.3% del total provenientes casi en su totalidad de un mismo conjunto de datos) para poder entrenar un sistema computacional capaz de reconocerlos. La distribución resultante del conjunto de datos guarda correspondencia con la cantidad de células presentes en la sangre humana según la literatura consultada.

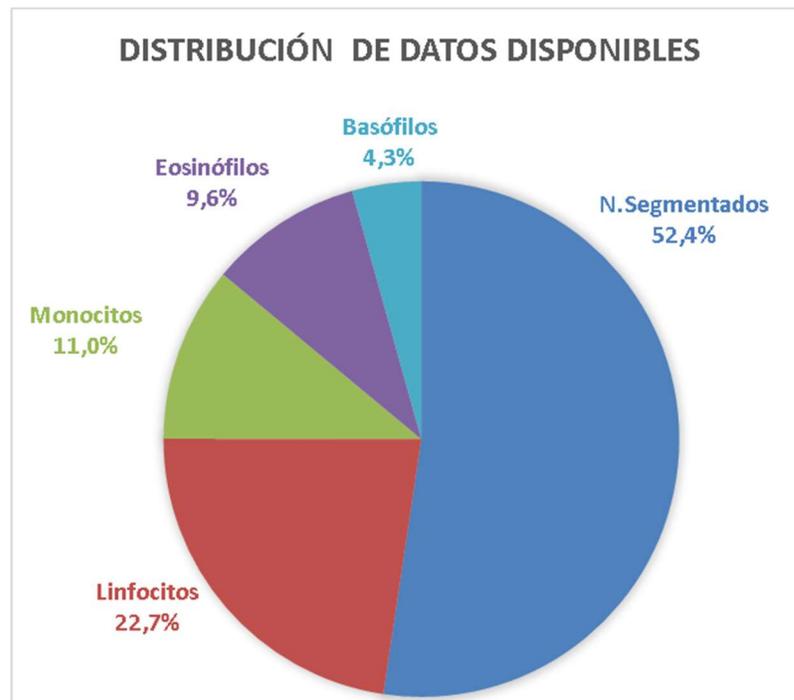


Figura . Distribución total de leucocitos en datos disponibles

Finalmente, se extrajo de las bases de datos mencionadas anteriormente un total de 1940 imágenes individuales de leucocitos distribuidos como se indica en la Tabla . El conjunto de datos disponible fue dividido de forma aleatoria utilizando 210 imágenes para la prueba de funcionamiento del clasificador (10.83%) y el resto el proceso de entrenamiento (89.12%).

Tabla
Distribución de leucocitos en los datos disponibles

Conjunto de datos experimentales	Neutrófilos segmentados	Linfocitos	Monocitos	Eosinófilos	Total
ALL-IDB1	26	63	9	3	101
ALL-IDB2	16	58	11	12	97
LISC	51	53	50	41	195
WBC-ID DataSet1 JTSC-I	96	53	36	13	198
WBC-ID DataSet1 JTSC-II	63	15	2	19	99
WBC-ID DataSet2	26	38	21	11	96
IUMS-IDB	146	59	14	33	252
SMC-IDB	262	41	25	47	375
BDL	364	118	39	6	527
Total	1050	498	207	185	1940
	54.1%	25.7%	10.7%	9.5%	100%

4.1.9. Aumento de datos y generación de imágenes sintéticas

Una vez aplicado el algoritmo de segmentación (que será descrito más adelante en la sección 4.2 del presente documento) y extraídas todas las imágenes correspondientes a las células (o posibles células) que serían utilizadas para entrenar al clasificador, las mismas fueron evaluadas nuevamente por el experto médico a fin de descartar falsos positivos como plaquetas u otras regiones segmentadas de manera incorrecta.

Para la clasificación de los diferentes tipos de leucocitos en este trabajo, se optó por utilizar aprendizaje profundo empleando redes neuronales convolucionales, lo cual requiere disponer de miles de muestras pertenecientes a cada tipo de leucocito. La naturaleza de las imágenes médicas empleadas en este estudio, hace que este requerimiento sea difícil de satisfacer. Por esto fue necesario utilizar técnicas de aumento de datos para generar un conjunto de entrenamiento mucho mayor. Realizando cambios en la posición, orientación, intensidad y contraste, se pudo lograr un total de 140 variaciones a partir de cada imagen. La Figura muestra un ejemplo del resultado obtenido al aplicar este proceso.

Las operaciones aplicadas a las imágenes del conjunto de entrenamiento fueron las siguientes:

- Ajuste de brillo.

- Ajuste de contraste.
- Ecualización de histograma.
- Reflexión horizontal (*flip*).
- Rotación.
- Traslación y rotación aleatoria combinadas.

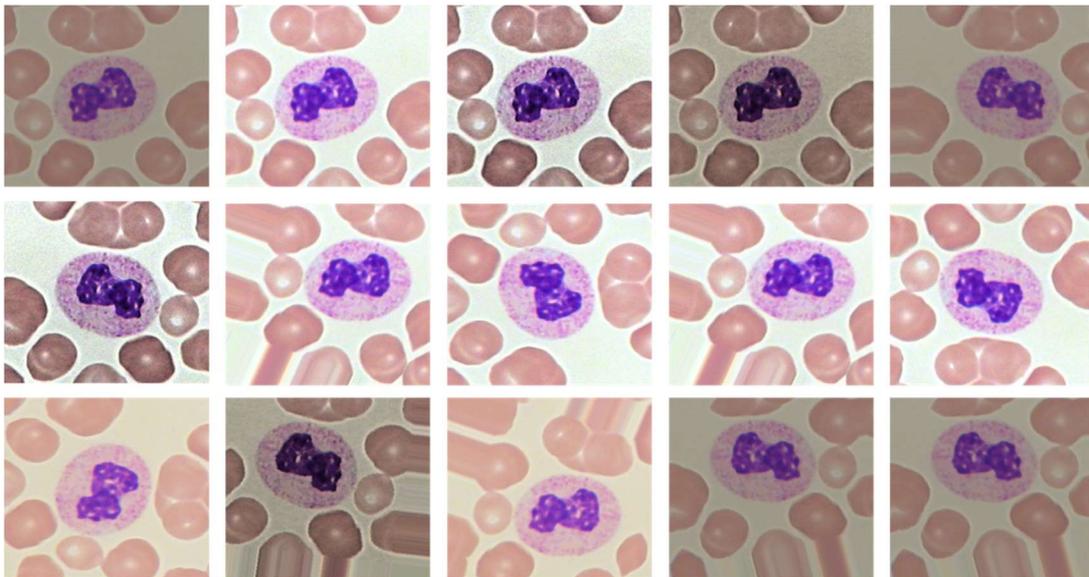


Figura . Ejemplo de imágenes aumentadas a partir de la imagen superior izquierda.

Este conjunto aumentado de imágenes fue utilizado para mejorar el entrenamiento del clasificador de leucocitos al permitir presentar al sistema miles de imágenes de cada clase con múltiples posibles representaciones de la misma célula. Los datos utilizados para entrenamiento y validación quedaron finalmente distribuidos en tres conjuntos como se indica en la Tabla .

Tabla

Distribución de conjuntos de datos para entrenamiento y prueba del clasificador

Conjunto datos	Neutrófilos Segmentados	Linfocitos	Monocitos	Eosinófilos	Total
Entrenamiento no aumentado (CT1)	945	450	186	163	1744
Entrenamiento Aumentado (CTA)	43700	34160	24050	20860	122770
Prueba-C	114	53	21	22	210

4.2. EXTRACCIÓN DE LEUCOCITOS Y REGIÓN DE INTERÉS

En esta sección se describen los ensayos realizados a fin de establecer un procedimiento para lograr la extracción de los leucocitos presentes en el frotis sanguíneo, comenzando por un abordaje tradicional basado en técnicas de procesamiento de imágenes clásicas hasta presentar el enfoque y resultados del método propuesto.

4.2.1. Ensayos mediante técnicas tradicionales de procesamiento de imágenes

El problema de segmentación del núcleo y/o citoplasma de leucocitos en frotis de sangre periférica ha sido abordado utilizando diferentes combinaciones de enfoques, metodologías y algoritmos, los cuales pueden categorizarse en cuatro grupos según el método empleado: basados en umbralización, reconocimiento de patrones, modelos deformables y metaheurísticos (Saraswat & Arya, 2014). La Figura reproduce una tabla comparativa publicada por estos autores. Otras fuentes consultadas durante el desarrollo del presente trabajo de investigación han utilizado los métodos presentados en la Tabla .

S.no.	References	Category	Sub-category	Comments
1.	Pan et al. (2012), Malpica et al. (1997), and Wu et al. (2006)	Threshold-based method	Region growing Watershed method Otsu's threshold	Fast and reliable methods for uniform images but inconsistent for the images having considerable variability
2.	Glotsos et al. (2004), Guo et al. (2007), Shitong and Min (2006), Mohapatra et al. (2011), Umpon (2005), and Sinha and Ramakrishnan (2003)	Pattern recognition-based method	Support Vector Machine (SVM) Artificial Neural Network (ANN) k-Means clustering Fuzzy c-means Expectation-maximization	Categorized as unsupervised or supervised methods. Unsupervised methods produce poor results for the images having complex background. Supervised methods are based on learning and classification of objects
3.	Rezatofghi and Soltanian-Zadeh (2011), Yang et al. (2005)	Deformable models	Active contours (level sets and snakes)	Major limitation of multiple overlapping objects. Presence of irrelevant objects complicates the possibility of obtaining a viable segmentation
4.	Osowski et al. (2009), Karkavitsas and Rangoussi (2005), Saraswat et al. (2013), Nakib et al. (2007), and Saraswat and Arya (2014)	Metaheuristic-based method	Genetic algorithm Differential evolution Artificial bee colony Microcanonical annealing Non-dominated sorting genetic algorithm-II	Viewed segmentation problem as optimization problem and they tried to find the global optima to segment the objects. Use of metaheuristic methods for leukocyte segmentation is limited

Figura . Métodos empleados para la segmentación de leucocitos (Saraswat & Arya, 2014)

Tabla*Técnicas o algoritmos utilizados en la segmentación de leucocitos*

Autor	Técnica(s)/algoritmo(s) de segmentación
Ramoser y otros (2005)	Agrupamiento K-medios
Rodrigues y otros (2008)	Red neuronal artificial de retropropagación y red neuronal de pulso acoplado
Rezatofighi & Soltanian-Zadeh (2011)	Ortogonalización Gram-Schmidt, umbralización basada en histograma y contornos activos (<i>snakes</i>)
Mostafa y otros (2012)	Umbralización de Otsu y operaciones morfológicas
Habibzadeh y otros (2012)	Umbralización Niblack, detección de borde mediante método de Canny y Transformada Wavelet Discreta Compleja de Árbol Dual
Putzu y Di Ruberto (2013)	Umbralización algoritmo de Zack en espacio de color CMYK
Mathur y otros, (2013)	Contornos activos y umbralización en espacio color HSV
Nagabhushana y Rajeshwari (2014)	Umbralización de Otsu, detección de borde mediante método de Canny y operaciones morfológicas
Sarrafzadeh y otros (2014)	C-medios difusos en espacio color LAB y operaciones morfológicas
Bhavnnani y otros (2016)	Umbralización de Otsu y operaciones morfológicas en componente G
Zhang y otros (2014)	Agrupamiento K-medios simultaneo en RGB, HSI y CMYK
Liu y otros, (2015)	Agrupamiento <i>Mean Shift</i> , transformada Watershed y operaciones morfológicas
Di Ruberto y otros (2016)	Máquina de soporte vectorial
Zheng y otros (2018)	K-medios en espacio HSI junto SVM

La segmentación basada en umbralización es rápida y confiable cuando se utiliza con conjuntos de imágenes uniformes, pero produce resultados inconsistentes si existe variabilidad en los datos (Gurcan, Boucheron, Madabhushi, Rajpoot, & Yener, 2009). Para comprobar el funcionamiento de estas técnicas clásicas sobre los datos experimentales disponibles, se realizaron ensayos con los métodos de umbralización de Otsu (Otsu, 1979) y del triángulo (Zack, Rogers, & Latt, 1977).

Las figuras 66 y 67 muestran la aplicación de estas técnicas mediante un único valor umbral global en diferentes imágenes. Se experimentó utilizando imágenes de leucocitos en escala de grises así como las componentes individuales de los espacios de color RGB, HSV, LAB y CMYK. En la Figura se muestran ejemplos de dos leucocitos que pudieron ser segmentados exitosamente mediante estos algoritmos. Se aprecia que para estas imágenes el núcleo de las células pudo ser extraído con relativa facilidad a partir de las componentes S y A de los espacios de color HSV y LAB,

respectivamente. Igualmente las regiones pertenecientes a las células completas se pueden extraer directamente de varias componentes de los espacios HSV, LAB y CMYK. Para la célula mostrada en la parte superior de la figura 26 fue posible utilizar la imagen en escala de grises o las componentes R y B para extraer el leucocito con éxito, al contrario de lo observado en la imagen de la parte inferior. En estas imágenes el algoritmo de Otsu fue más efectivo para segmentar al núcleo, mientras que, el algoritmo del triángulo lo fue para la totalidad de la célula.

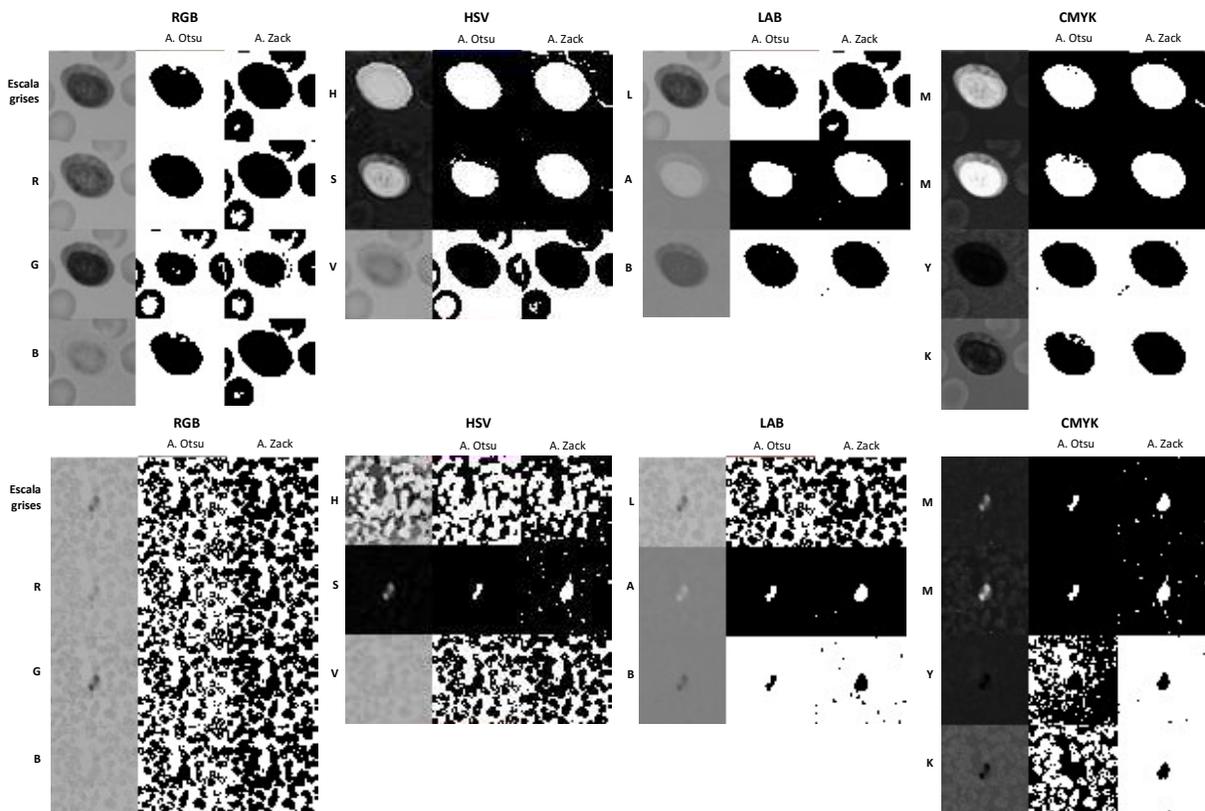


Figura . Segmentación usando métodos del valor umbral global con éxito

Por su parte, en la Figura se muestran ejemplos de dos leucocitos que no pudieron ser segmentados de manera adecuada mediante estos algoritmos. En ellas se aprecia que las observaciones realizadas para la figura anterior no son válidas, haciendo imposible generalizar la utilización de estos métodos de forma directa. En el anexo “A” se incluyen imágenes adicionales, las cuales soportan la inconsistencia en los resultados obtenidos al procesar datos provenientes de distintas fuentes.

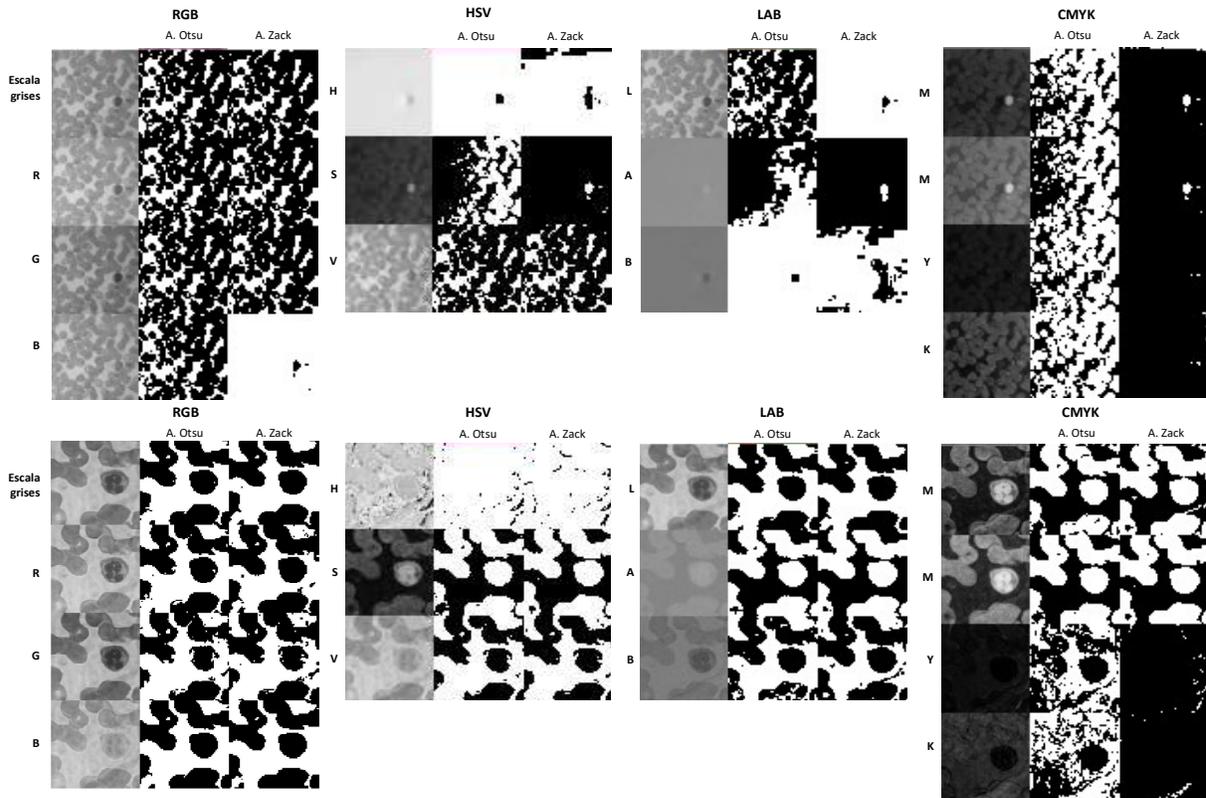


Figura . Segmentación usando métodos del valor umbral global sin éxito

En búsqueda de una solución a dicho problema se utilizó el algoritmo de transferencia de color entre imágenes descrito por Reinhard y otros (2001), previamente utilizado tanto por Zhang y otros (2014) como por Mathur, Tripathi y Kuse (2013) para homogenizar el color de frotis sanguíneo como etapa de pre-procesamiento. Para ello, se utilizó una muestra de 4 a 11 fotografías pertenecientes a cada base de datos, como conjunto destino a colorear “T” y se seleccionaron 9 imágenes fuente a ser usadas como patrón de color “S”.

Como se observa en la Figura , las imágenes “T” adquieren características de color similares a las del conjunto fuente “S”. En el anexo “B” se han incluido los resultados de las pruebas realizadas.

A partir de las características de la imagen fuente empleada, el uso de esta técnica permitió obtener mejoras en cuanto al brillo y contraste en la imagen resultante, así como en algunos casos resaltar el núcleo o cambiar completamente el color de los glóbulos rojos. Sin embargo, la aplicación del algoritmo de transferencia de color generó en muchos casos una coloración artificial (ver Figura) y no produjo mejoras significativas al intentar segmentar mediante las técnicas mencionadas (ver Figura).

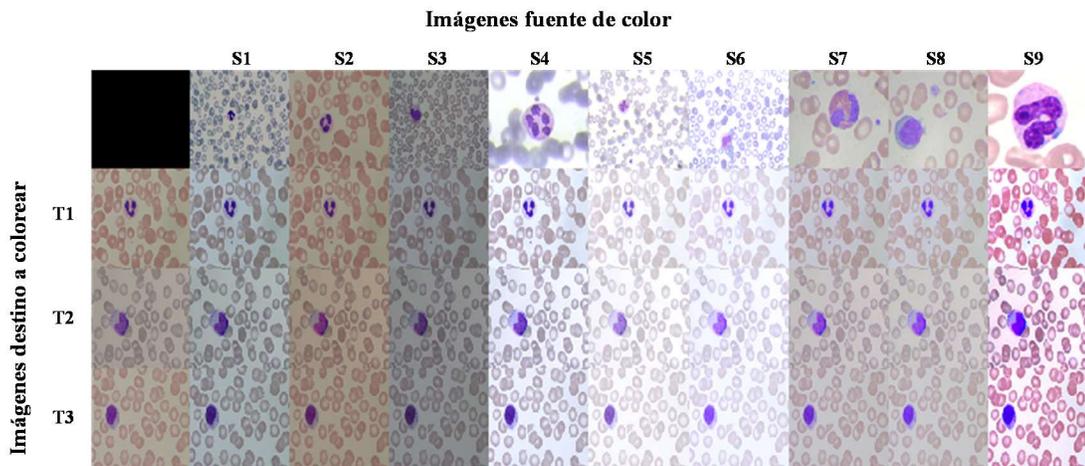


Figura . Homogenización del color.

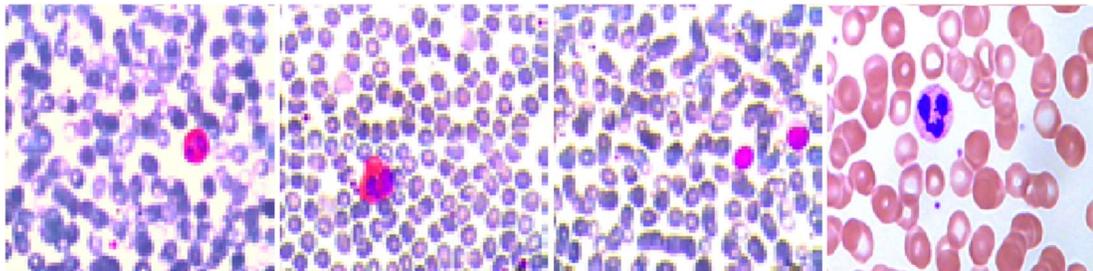


Figura . Imágenes con apariencia artificial posterior a la transferencia de color

Para el caso mostrado en la parte superior de la Figura , efectivamente se logró un realce del color y contraste en la imagen, el cual, permite al observador humano localizar más fácilmente el núcleo y citoplasma del leucocito, esto no ha implicado que el algoritmo de umbral global brinde un mejor resultado para segmentar directamente esos elementos.

Autores que han usado este enfoque para la segmentación de leucocitos como por ejemplo Mohamed y otros (2012) han encontrado limitaciones en la exactitud alcanzada y adicionalmente han requerido completar su metodología mediante la inclusión de criterios de rechazo de píxeles basados en un conocimiento *a priori* de las imágenes estudiadas o el uso de operaciones morfológicas sobre el resultado de la umbralización. Estas últimas requieren de la definición de elementos estructurantes cuya forma y tamaño son dependientes de las dimensiones de la imagen y de la propia célula que desea segmentar y por ello no es posible definirle un valor óptimo que funcione con imágenes variadas, en resolución y dimensiones, como se pretendió en el presente trabajo. La obtención de resultados de buena calidad en estos casos, depende de un ensayo empírico basado en el análisis de las características de la imagen en estudio. Por ello, se optó por abordar el problema con un enfoque diferente, el cual se describe en la siguiente sección de este trabajo.

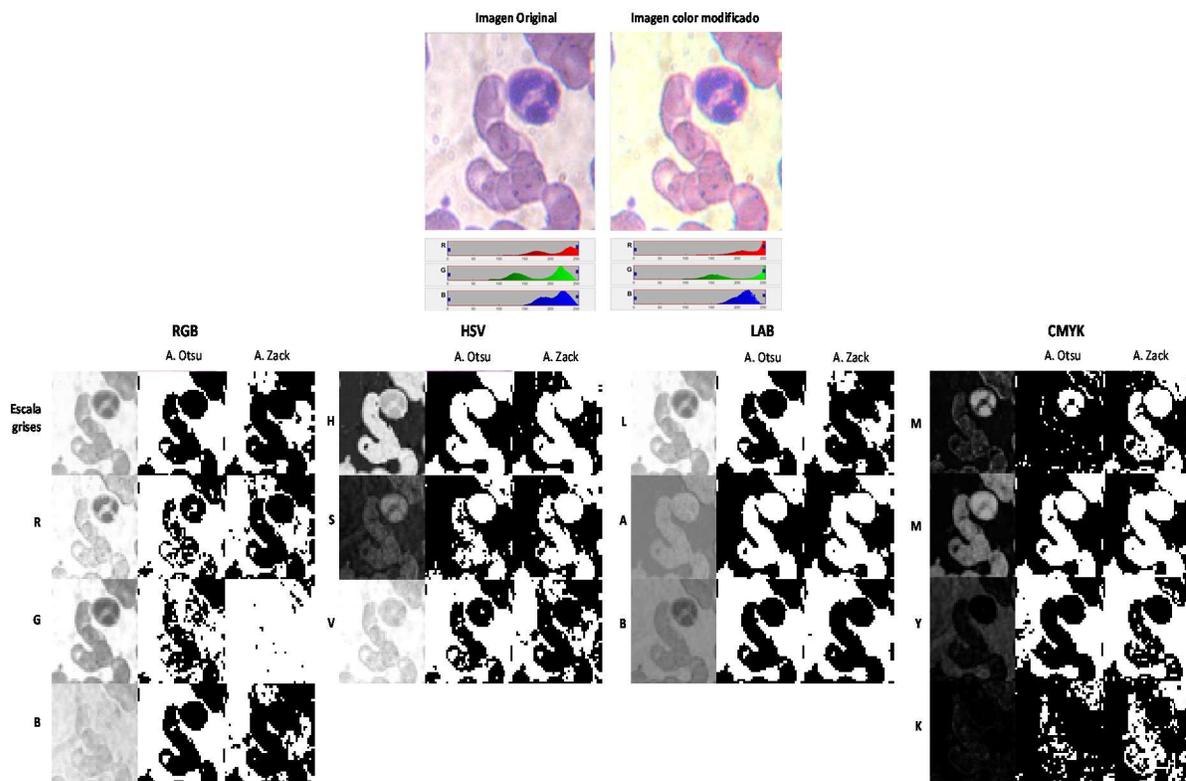


Figura . Mejora del color y segmentación de la imagen resultante. Arriba: efecto de aplicar el algoritmo de transferencia de color. Abajo: segmentación sin éxito usando métodos de umbral global.

El código fuente utilizado para la transferencia de color entre imágenes y segmentación usando técnicas tradicionales, se encuentra disponible en el apéndice A-1 “Scripts pruebas iniciales”.

4.2.2. Propuesta de algoritmo de segmentación combinando Superpíxeles y SVM

Ante el panorama descrito en la sección anterior y, considerando que, otros investigadores han demostrado que los métodos basados en aprendizaje supervisado, como las máquinas de soporte vectorial y redes neuronales artificiales, generan mejores resultados al ser utilizados para la segmentación de imágenes (Saraswat & Arya, 2014), se propuso implementar un bloque de extracción que permite segmentar e identificar las regiones pertenecientes al núcleo y citoplasma de los leucocitos basado en máquinas de soporte vectorial. La solución propuesta se inspira en la simplicidad del modelo presentado por Di Ruberto y otros (2016), pero incluye modificaciones orientadas tanto a permitir un procesamiento de la imagen en menor tiempo y con recursos computacionales limitados, como a incorporar el procesamiento simultáneo en varios espacios de color.

Para cada conjunto de imágenes dentro de las diferentes bases de datos, se seleccionó un grupo entre 4 y 8 imágenes representativas y se extrajo manualmente muestras pertenecientes a 4 regiones: Núcleo, Citoplasma, Plasma y Otro. Dichos píxeles fueron utilizados posteriormente como etiquetas descriptoras de su región de pertenencia. La Figura muestra el proceso de etiquetado de regiones utilizando la herramienta “Image Labeler” incorporada en MATLAB. Cada uno de los cuadros de colores señala la pertenencia de una región de píxeles a un elemento de la imagen.

Utilizando los píxeles pertenecientes a cada región se entrenó un clasificador SVM independiente y especializado para las imágenes de cada base de datos, el cual, logra identificar la pertenencia de un píxel (núcleo, citoplasma, plasma u otro) en función de las características particulares de las imágenes en ella contenida. Esto facilitó agregar

cierta modularidad a la solución planteada, permitiendo a partir de unas pocas muestras (incluso una sola imagen resulta suficiente) adaptar el proceso de segmentación a bancos de datos con características distintas.

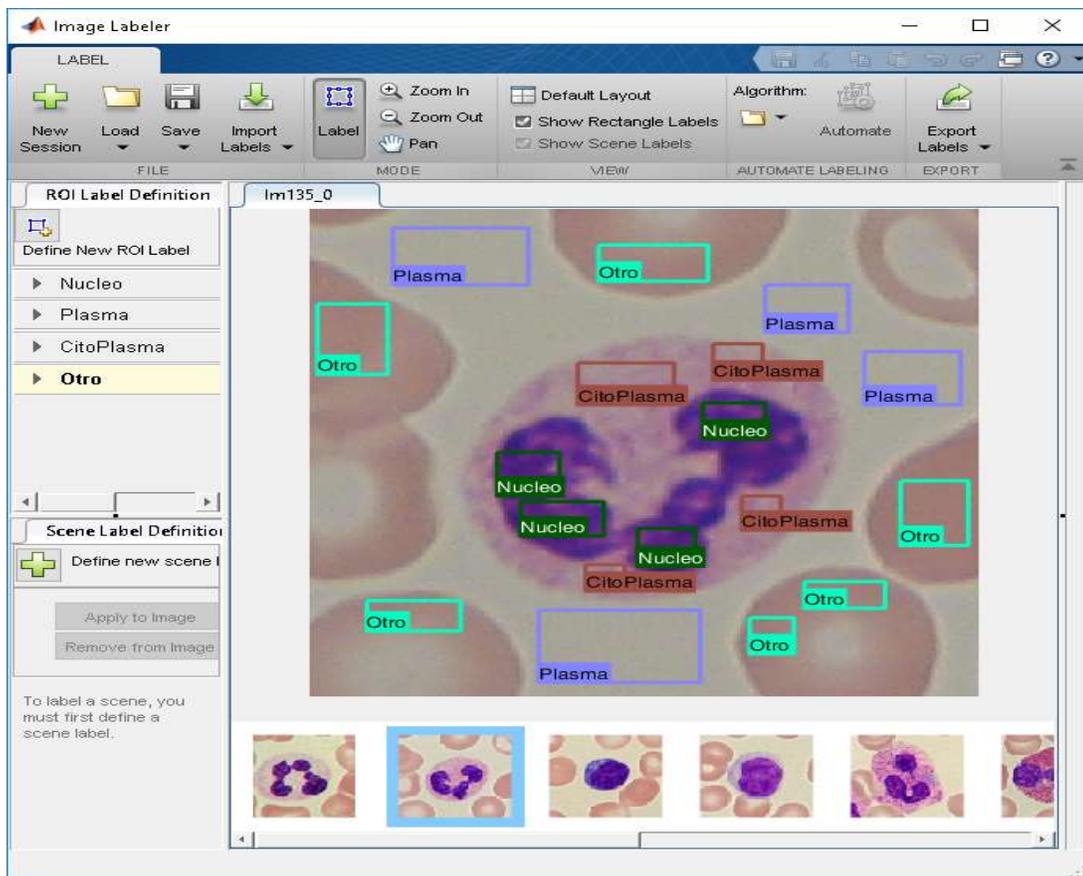


Figura . Etiquetado de regiones para entrenamiento de SVM usado en la segmentación. A partir de las características de los píxeles presentes en las regiones etiquetadas, el sistema es capaz de aprender un modelo para posteriormente segmentar correctamente imágenes similares.

El vector de rasgos para la máquina de soporte vectorial fue construido utilizando las intensidades de cada píxel en los espacios de color RGB, LAB y HSV de forma simultánea, conformando así un vector de 9 elementos. No se incorporó información pertinente a la vecindad de cada píxel por considerar que ello comprometería la velocidad de ejecución al momento de segmentar las imágenes.

Considerando esto último, en lugar de recorrer la imagen completa y clasificar cada píxel individualmente, se optó por incorporar una etapa de pre-procesamiento que permitiera reducir drásticamente el costo computación de realizar la segmentación.

Esto fue realizado mediante el uso de una técnica de sobre-segmentación (*over segmentation*) para descomponer previamente cada imagen en un total entre 200 a 1000 subregiones (ajustado su número de forma empírica-experimental) mediante el algoritmo *Simple Linear Iterative Clustering* (SLIC). Cada una de las regiones obtenidas al aplicar el algoritmo (llamadas superpíxeles) agrupa píxeles con características suficientemente similares como para obtener información representativa promedio capaz de representar toda la región como único píxel. SLIC es solamente uno de muchos algoritmos existentes para la segmentación de imágenes mediante la generación de superpíxeles. Fue seleccionado por su alta eficiencia y capacidad de generar un número conocido y controlado de regiones con áreas relativamente homogéneas. Al ser comparado con algoritmos similares (*Normalized cuts, Minimum spanning tree, Graph cuts, Mean-shift, Quick shift, Watershed, Turbopixel*), SLIC demostró ser superior en casi todos los aspectos permitiendo segmentar imágenes complejas de 320x240 píxeles en un tiempo menor a 1 segundo utilizando un computador de muy modestas prestaciones (procesador Intel® Dual Core™ 2.26 GHz con 2 GB RAM). Esto lo logra al reducir significativamente la complejidad del grafo resultante, haciendo manejable la segmentación (Achanta, y otros, 2012). Al verificar experimentalmente esta información, logrando segmentar de manera rápida y efectiva las imágenes usadas en este trabajo, se consideró conveniente su utilización definitiva.

Como cada superpíxel representa a toda una región, el clasificador de píxeles (SVM), utilizado para segmentar la imagen del frotis sanguíneo, opera sobre un conjunto limitado (máximo 2000 elementos) en lugar de hacerlo sobre todos los píxeles de la imagen (por ejemplo 1048576 elementos para una imagen de 1024x1024) permitiendo una gran reducción del costo computacional total.

Las figuras 71, 72 y 73 muestran resultados obtenidos al aplicar el algoritmo de segmentación implementado a diferentes imágenes. En el lado izquierdo se presenta la imagen segmentada en regiones y del lado derecho se observa la asignación a una categoría (núcleo, citoplasma, plasma u otro) realizada por la SVM.

En la Figura puede verse el resultado obtenido al variar la cantidad de superpíxeles usados para la segmentación. Experimentalmente se determinó que valores mayores de 300 y menores de 2000 eran adecuados para la mayoría de las imágenes contenidas en los datos utilizados. Para algunas imágenes (donde la célula ocupa gran parte de la misma) valores pequeños como 80 superpíxeles fueron suficientes para lograr regiones fácilmente separables. Por su parte, valores mayores a 2000 hacen que el algoritmo se ejecute notablemente más lento sin ofrecer mayor exactitud en la segmentación realizada.

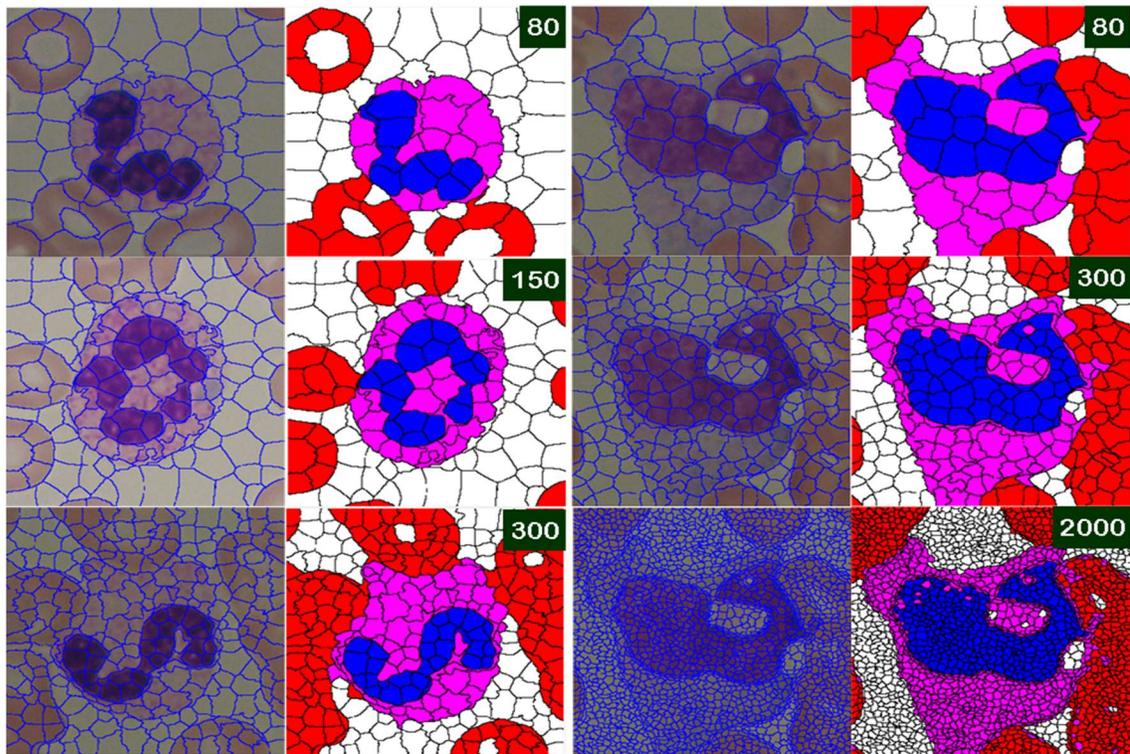


Figura . Resultados segmentación y variación de la cantidad de superpíxeles

En la Figura se muestran resultados obtenidos al segmentar fotografías consideradas de buena o media calidad que contienen una única célula en ellas. En todos los casos se observó que el sistema logró obtener una adecuada separación tanto del núcleo como del citoplasma.

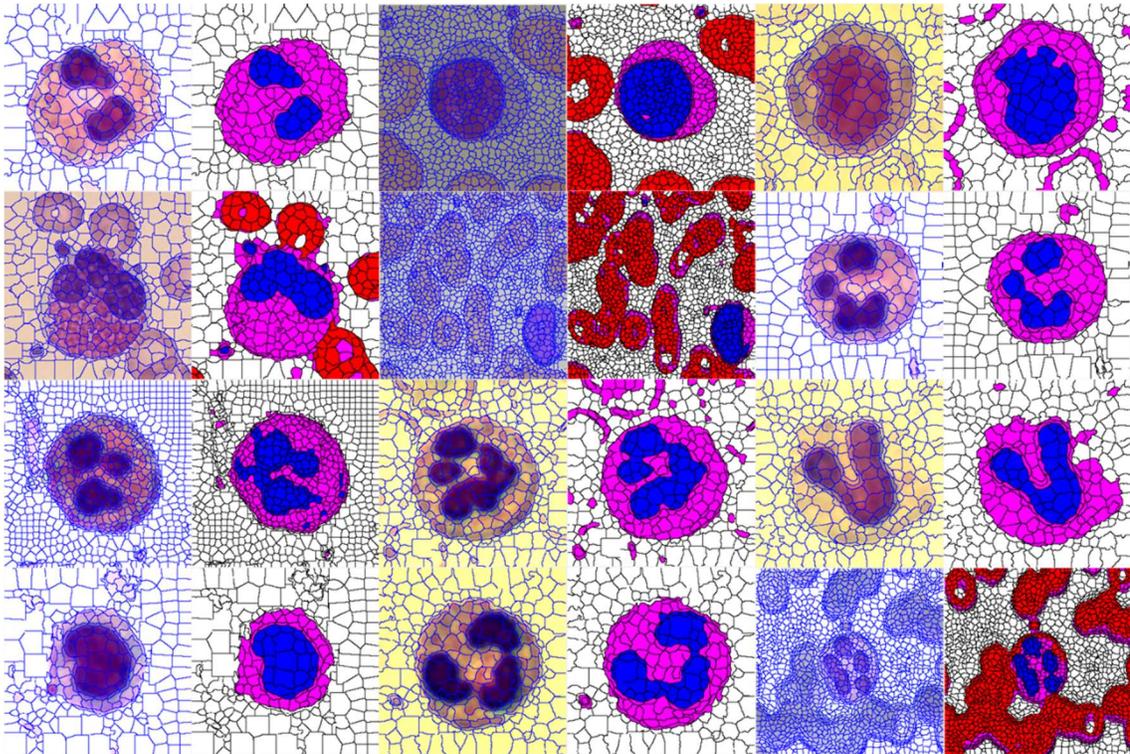


Figura . Resultado de la segmentación en imágenes de calidad buena o media

En la Figura se aprecian resultados obtenidos al segmentar fotografías de baja calidad o con múltiples células presentes. Nuevamente, fue posible lograr una identificación correcta del núcleo en todas las células observadas.

En algunos casos pertenecientes a la base de datos de menor calidad, como los mostrados en la Figura , se pudo observar que el citoplasma fue segmentado de manera incorrecta o incompleta. Esto sin embargo, es perfectamente atribuible a la poca calidad de la propia imagen fuente, en la cual, resulta difícil inclusive para el ojo humano discriminar la coloración del citoplasma y de los otros elementos presentes como plasma y los glóbulos rojos.

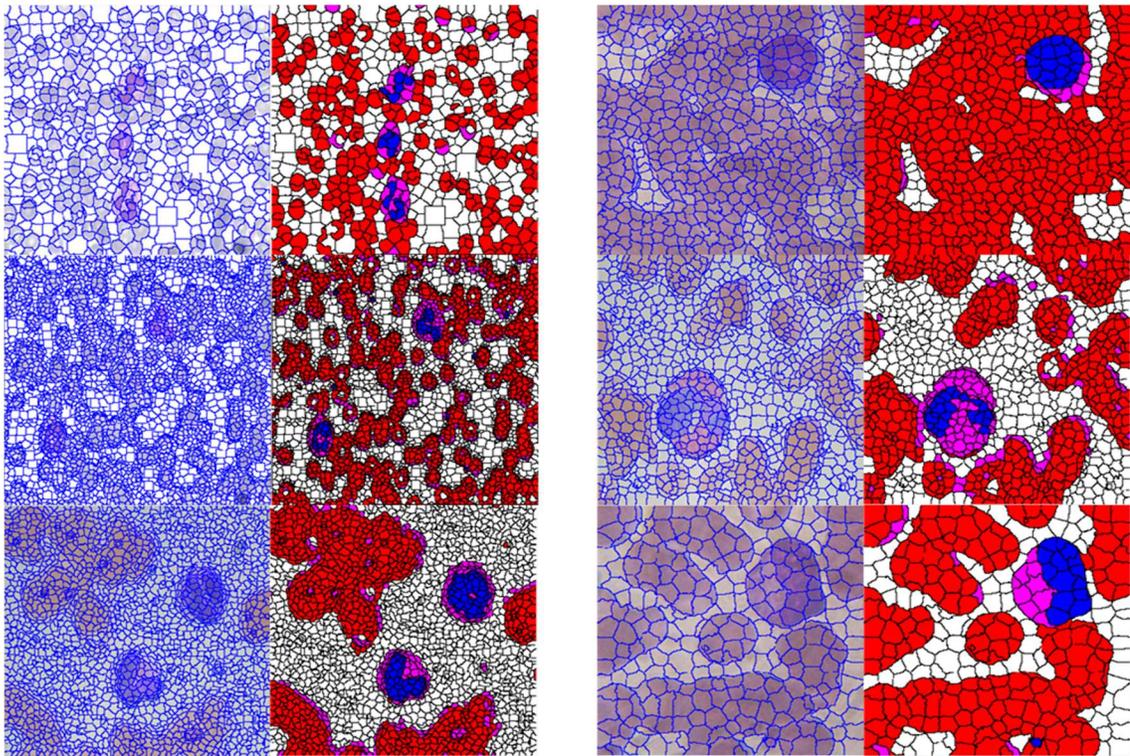


Figura . Resultados segmentación en imágenes con múltiples células o baja calidad

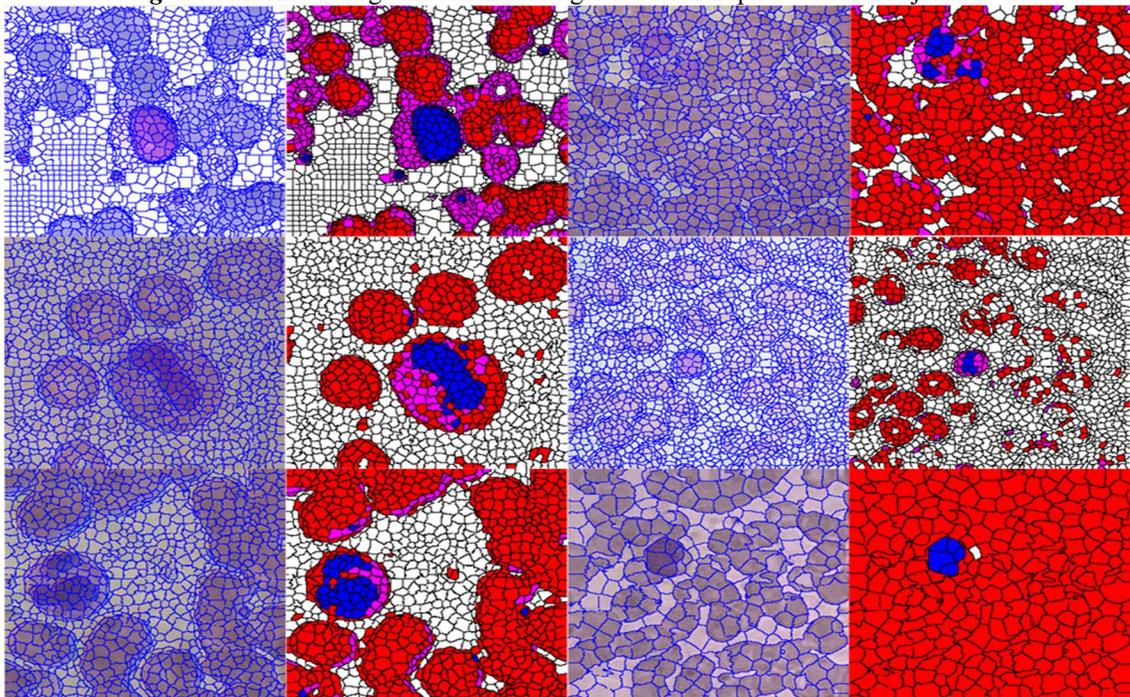


Figura . Segmentación incorrecta del citoplasma en algunas imágenes de baja calidad

4.2.3. Extracción de leucocitos y región de interés

Se creó la interfaz gráfica de usuario (GUI) mostrada en la Figura mediante la cual pueden realizarse las siguientes acciones:

- Seleccionar una imagen o un directorio de imágenes para ser procesadas.
- Cambiar manualmente la cantidad de superpíxeles o regiones a utilizar.
- Aplicar el algoritmo de segmentación descrito anteriormente.
- Extraer la célula completa junto a su vecindad y almacenar cada imagen resultante de forma individual.

A medida que avanza el proceso de segmentación se puede visualizar:

- Imagen original.
- Sobre-segmentación mediante superpíxeles.
- Imagen de color promedio en cada superpíxel.
- Segmentación resultante.
- Mascara binaria para extracción de célula.

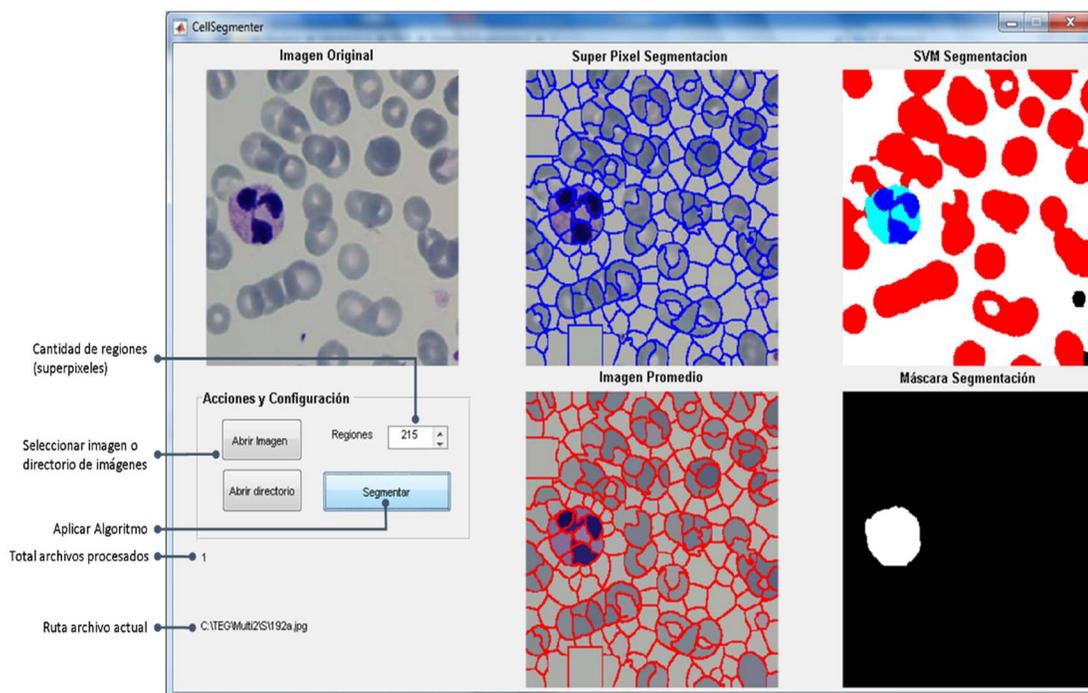


Figura . Interfaz gráfica de usuario desarrollada para la segmentación de leucocitos

La máscara binaria para la extracción de los leucocitos se conformó mediante la unión de las regiones clasificadas como posibles núcleos celulares junto a las porciones de citoplasma que se encuentren conexas a estas. Las áreas de interés resultantes fueron ordenadas de mayor a menor en función del área ocupada en la imagen y se fusionaron aquellas regiones solapadas. Posteriormente se eliminaron aquellos elementos de tamaño menor a un umbral establecido experimentalmente. Este valor umbral se definió igual al 35% del área del mayor objeto detectado, en función de la relación existente entre el tamaño de los diferentes tipos de leucocitos. Finalmente se extrajo de la imagen original toda el área perteneciente a la célula y a su vecindad (ver detalle en Figura), la cual es almacenada en disco para su uso posterior. La extracción se realizó utilizando un cuadrado concéntrico al centroide de la célula y lados iguales al doble de la longitud de su eje mayor.

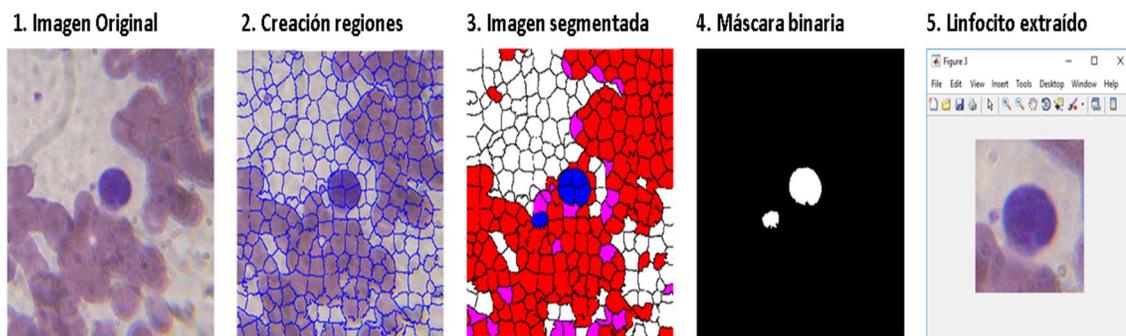


Figura . Resultado del proceso de segmentación y extracción final de leucocitos

El código fuente utilizado para la segmentación de las imágenes de este proyecto se encuentra disponible en el apéndice A-2 “Scripts segmentación” y se compone de los archivos:

- TrainSeg.m: archivo para entrenamiento del clasificador de píxeles.
- CellSegmenter.m: implementación del algoritmos de segmentación e interfaz gráfica de usuario. Trabaja en conjunto con archivo CellSegmenter.fig
- ExtraerCell.m: archivo para extraer célula entera. Requiere del uso de:
 - RemoveIncluded.m: *script* para eliminación de áreas solapadas.
 - CellCrop.m: *script* para recorte de área de interés en imagen origen.

En las pruebas realizadas al conjunto completo de datos indicaron que el 100% de los núcleos de los leucocitos presentes en las fotografías pudo ser detectado. En algunos casos (menos del 3%), regiones pertenecientes a otros tipos de células sanguíneas (especialmente plaquetas) o irregularidades del frotis fueron detectadas como posibles núcleos de leucocitos.

4.3. CLASIFICACIÓN DE LOS LEUCOCITOS EN LA SANGRE HUMANA

Para la clasificación de leucocitos usando imágenes se han empleado diferentes métodos con enfoques probabilísticos (p. ej. clasificador Bayesiano), geométricos (p.ej. SVM y ANN) o basados en similaridad de píxeles (p. ej. K-vecinos más cercanos) (Saraswat & Arya, 2014). Recientemente algunos trabajos han explorado el uso de clasificadores basados en modelos simples de redes neuronales convolucionales para esta tarea Habibzadeh y otros (2013), (Lisicki, 2014), Jiang y otros (2018), sin lograr un éxito comparable o superior al logrado con técnicas tradicionales.

Las redes neuronales convolucionales son arquitecturas de aprendizaje profundo inspiradas en el mecanismo de percepción de los seres vivos que ha tenido un gran desarrollo en los años recientes (Jiang y otros, 2018). En su diseño combinan características que les permiten tener cierta invariancia ante imágenes desplazadas (traslación), escaladas o distorsionadas (LeCun y otros, 1998), logrando así reconocer patrones con gran variabilidad.

En general las CNN usan datos crudos sin usar pre-procesamiento y no requieren segmentación previa siendo esto uno de sus principales atributos. Sin embargo, el problema de clasificación de leucocitos en frotis de sangre periférica puede implicar trabajar con imágenes tan diferentes como las mostradas en la Figura , donde una sola fotografía puede contener una o más células distintas. Entrenar satisfactoriamente un modelo de red convolucional directamente con imágenes tan dispares como estas es, en opinión del autor y de acuerdo a los resultados de otros grupos de investigación, difícil de lograr.

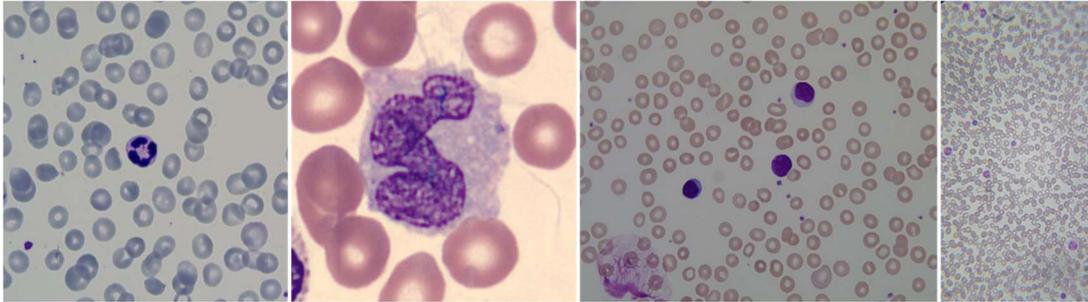


Figura . Imágenes de leucocitos provenientes de diferentes bases de datos. Se aprecia la marcada variabilidad en color, posición y tamaño que presentan las células en estudio.

Al utilizar modelos de redes neuronales convolucionales modernos es poco común el entrenamiento de la red desde cero. Este enfoque requiere del uso de cientos de miles o incluso millones de imágenes para lograr un resultado exitoso debido al enorme número de parámetros que deben ser calculados. Además, el proceso es computacionalmente intensivo requiriendo incluso semanas enteras de cálculo ininterrumpido con equipos de gran poder de cómputo operando en paralelo.

Por esta razón, la mayoría de las aplicaciones con redes neuronales convolucionales emplean la transferencia del aprendizaje o transferencia de conocimiento. De forma general “*transfer learning*” es el mejoramiento del aprendizaje de una nueva tarea por medio de la transferencia del conocimiento logrado de una tarea similar que ha sido aprendida previamente (Torrey & Shavlik, 2009). Esto quiere decir que mediante este método un modelo de red neuronal desarrollado para cumplir una tarea específica es reutilizado como punto de inicio para la construcción de un modelo destinado a otra aplicación.

El uso de este enfoque implica disponer de un modelo de red que ha sido previamente entrenado con un conjunto de imágenes muy grande, por ejemplo la base de datos ImageNet (<http://www.image-net.org/>).

En general, existen tres escenarios en los cuales se utiliza la transferencia de conocimiento con las redes neuronales convolucionales los cuales son:

- **Inicialización de pesos a partir de modelo pre-entrenado:** Como se mencionó anteriormente, el entrenamiento de modelos modernos de redes convolucionales desde cero es una labor computacional muy intensa por lo cual en la práctica es conveniente utilizar un modelo previamente entrenado y a partir de este, re-entrenar a la red utilizando las imágenes del conjunto que se pretende clasificar. Yosinski, Clune, Bengio y Lipson (2014) encontraron que una red inicializada con un conjunto A y re-entrenada para clasificar un conjunto B, generalizaba mejor que al entrenar la red desde cero para el conjunto B, dando evidencia de las ventajas del método frente a la inicialización aleatoria de las matrices de pesos. En todo caso, este enfoque requiere disponer de un conjunto de entrenamiento muy grande.
- **Empleándolas como extractores de rasgos fijos (*feature extraction*):** En este caso, no se reentrena la red neuronal sino que se utilizan los mapas de características extraídos por ella como entrada o vector de rasgos para otro algoritmo de clasificación, por ejemplo una máquina de soporte vectorial. Su uso es recomendado cuando la base de datos de los objetos nuevos objetos a clasificar es pequeña (decenas o cientos de imágenes por clase). Es el enfoque que requiere menor poder y tiempo de cómputo y aprovecha notablemente las capacidades de representación de las imágenes que poseen las redes neuronales convolucionales.
- **Mediante ajuste de pesos (*fine-tune*):** La idea general se basa en la modificación y re-entrenamiento de la red convolucional para adaptarla a la nueva tarea de clasificación. Su uso es adecuado cuando se dispone de miles o cientos de miles de imágenes de cada objeto que se pretenda clasificar. Es posible entrenar nuevamente todas las capas de la red o mantener fijas las capas iniciales y re-entrenar únicamente las capas superiores (últimas). Haciendo esto último, se reduce el sobreajuste y tiempo de entrenamiento necesario.

Para este trabajo se propuso experimentar tanto con la metodología de ajuste de pesos como con la extracción de rasgos mediante CNN.

4.3.1. Clasificación usando combinación de CNN y SVM

4.3.1.1. Extracción automática de rasgos usando CNN

Las redes de aprendizaje profundo se caracterizan por poseer decenas de capas apiladas, las cuales, al ser entrenadas aprenden representaciones de formas cada más vez complejas que se encuentran presentes en la imagen. De esta forma, las primeras capas de la red son capaces de detectar características simples o genéricas como bordes, líneas horizontales o verticales; mientras que, las últimas capas detectan formas bien definidas propias de los objetos que se pretende identificar.

Las características o rasgos descriptivos que logra identificar una CNN no son conocidos *a priori*, ya que dependen del ajuste de las matrices de pesos (también llamados filtros o *kernels*) que se actualizan durante el proceso de entrenarla usando miles de imágenes. Una vez que la red ha sido entrenada, cada capa posee la capacidad de elaborar una representación con características distintas que pueden ser extraídas en cualquier momento.

La Figura conceptualiza en forma gráfica la estructura de una red neuronal convolucional, en ella se observa con claridad la existencia de diversos mapas de características (*feature maps*). Estos mapas de características se activan de forma diferente en respuesta a la imagen de entrada aplicada a la red y permiten extraer información de la forma o estructura del objeto observado siendo invariante traslación del mismo.

De esta forma, la red convolucional pasa a convertirse en un extractor de rasgos muy versátil cuya salida puede usarse como entrada para otro modelo o algoritmo de aprendizaje automático. Así entonces, se hace necesario entrenar únicamente el nuevo

modelo de aprendizaje para la etapa de salida del sistema, ajustándolo para realizar la tarea específica (en este caso el reconocimiento del tipo de leucocito presente en la imagen). Ello supone un enorme ahorro en cuanto al tiempo de procesamiento, poder de cálculo y conjunto de imágenes requeridos para resolver el problema planteado.

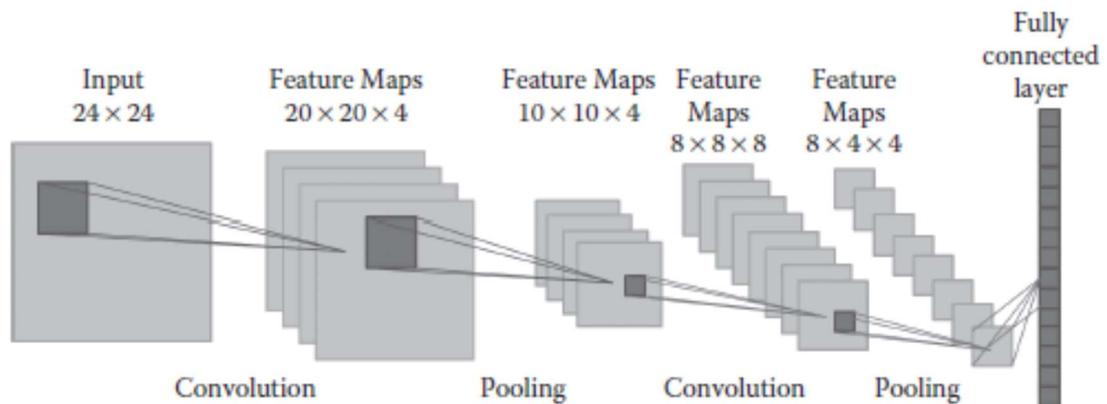


Figura . Diagrama representativo estructura CNN. Nagpal y otros (2018, pág. 16).

Dicho enfoque ha sido utilizado en variedad de aplicaciones como por ejemplo: la clasificación de desechos sólidos (Özkaya & Seyfi, 2018), reconocimiento de lenguaje de señas (Oliveira y otros, 2017), objetivos en imágenes de sonar (Zhu, Isaacs, Fu, & Ferrari, 2018), daño estructural en edificaciones (Gao & Mosalam, 2018), identificación de género (Van de Wolfshaar, Karaaba, & Wiering, 2015), análisis morfológico de bucles capilares papilares intraepiteliales para el diagnóstico de cáncer esofágico (Xue, Zhang, Feng, & Wang, 2016), clasificación de imágenes histológicas para la detección de cáncer (Xu, y otros, 2017) e identificación de leucocitos (Shahin, Guo, Amin, & Sharawi, 2019); entre otras.

Si el nuevo conjunto de imágenes a clasificar es similar al conjunto de datos usado para entrenar el modelo de la CNN, entonces se emplea para la extracción de rasgos una de las capas superiores (finales) de lo contrario resulta conveniente utilizar las capas anteriores (iniciales). Esto fue comprobado experimentalmente para varios modelos de redes convolucionales en los ensayos realizados.

4.3.1.2. *Procedimiento empleado para extraer rasgos usando CNN*

Para la realizar extracción de rasgos mediante la red convolucional se utilizó el procedimiento que se describe a continuación:

- Se instalaron modelos pre-entrenados de las redes convolucionales: AlexNet, Densenet201, GoogleNet, Inception-ResNetV2, InceptionV3, Resnet-18, Resnet-50, Resnet-101 y SqueezeNet. Los modelos empleados en este trabajo fueron previamente entrenados para reconocer 1000 categorías distintas usando un subconjunto 1.2 millones de imágenes de la base de datos ImageNet.
- Se identificó en cada modelo algunas la(s) capa(s) que podían emplearse como extractores de rasgos para el clasificador, lo cual es indicado en la Tabla . En aquellos casos en los que se indica más de una capa, es porque se realizaron ensayos para evaluar las diferencias existentes al extraer los rasgos usando capas más internas de la red convolucional.

Tabla

Capas utilizadas como extractor de rasgos en diferentes modelos de redes convolucionales

Modelo de red	Total capas	Capa extracción de rasgos	Número capa extracción
AlexNet	25	pool5	16
		fc6	17
		fc7	20
		fc8	23
Densenet201	709	avg_pool	706
		fc1000	707
GoogleNet	144	pool5_7x7_s1	140
		loss3-classifier	142
inceptionresnetv2	825	avg_pool	822
		predictions	823
inceptionV3	316	avg_pool	313
		predictions	314
ResNet-18	72	pool5	69
		fc1000	70
ResNet-50	177	avg_pool	174
		fc1000	175
Resnet-101	347	pool5	344
		fc1000	345
SqueezeNet	68	pool10	66

- En cada modelo de red, se calculó la activación producida en la capa seleccionada y esta fue utilizada como entrada para entrenar un clasificador multiclases SVM con *kernel* lineal empleando las imágenes del conjunto de entrenamiento no aumentado CT1 (ver detalles en la Tabla , página 153), con las siguientes variantes:
 - Conjunto de datos balanceados con 163 muestras por clase utilizando:
 - a) Estrategia de descomposición 1-vs-todos (OVA) (Experimento # 1).
 - b) Estrategia de descomposición 1-vs-1 (OVO) (Experimento # 2).
 - Conjunto de datos desbalanceado (usando todos los datos disponibles)
 - a) Estrategia de descomposición 1-vs-todos (Experimento # 3).
 - b) Estrategia de descomposición 1-vs-1 (Experimento # 4).
- Con el clasificador entrenado, se calcularon la exactitud y matriz de confusión para los siguientes conjuntos de prueba:
 - CE1: 100 muestras por clase (aleatorias) del conjunto CT1.
 - CE2: 1000 muestras por clase del conjunto aumentado CTA.
 - CE3: Conjunto “Prueba –C” de 210 imágenes.
- En cada experimento se obtuvo medida de los tiempos requeridos para:
 - Obtener el mapa de activación promedio de una imagen (T.Act).
 - Entrenamiento del clasificador (T.Ent)
 - Clasificación promedio de una imagen (T.Clas).

Los resultados obtenidos en este punto se muestran desde la Tabla a la 12. En la primera columna de cada tabla se presenta la capa de la red utilizada, seguido de las matrices de confusión normalizadas para cada uno de los conjuntos de datos de prueba y finalmente en las tres últimas columnas contienen las medidas de tiempo arriba mencionadas.

Para evaluar el efecto de incrementar el número de muestras usando imágenes aumentadas, se entrenó nuevamente al clasificador esta vez con 5000 muestras por clase. En función de los resultados obtenidos en los primeros experimentos, los modelos de redes y capas de extracción seleccionadas para el Experimento#5 fueron: AlexNet-pool5, GoogleNet-pool5-7x7_s1, InceptionV3-avg_pool, ResNet-18-pool5, ResNet-50-avg_pool y SqueezeNet-pool10. En modelo del clasificador SVM fue entrenado tanto en esquema 1 vs 1 como 1 vs todos. Este clasificador fue probado únicamente sobre el conjunto CE3 y sus resultados se muestran en la Tabla .

4.3.1.3. Resultados obtenidos por el clasificador SVM

En esta sección se presentan mediante tablas y analizan usando diversos gráficos, los resultados logrados al utilizar la red convolucional como extractor de rasgos y un clasificador SVM para determinar el tipo de leucocito presente en la imagen según los experimentos ya descritos.

Entrenamiento y clasificación con datos no aumentados

Los resultados de los experimentos #1 al #4 se muestran a continuación. En cada tabla de resultados se presenta la matriz de confusión normalizada y expresada en términos porcentuales obtenida al evaluar las imágenes de cada conjunto de prueba (CE1, CE2 y CE3).

Análisis experimentos #1 al #4

Para facilitar el análisis de los resultados obtenidos se construyeron gráficas (ver Figura a Figura) de la exactitud global del clasificador para utilizarlas como criterio inicial de comparación. Adicionalmente se elaboró el gráfico de burbujas de la Figura , el cual permite observar simultáneamente el tiempo relativo de procesamiento asociado a cada red neuronal y la exactitud global alcanzada al clasificar el conjunto CE3.

Tabla
Resultados Experimento #1. Conjunto de datos balanceado con descomposición 1 a todos

Red-capa	CE1	CE2	CE3	T.Act. (ms)	T.Ent. (s)	T.Clas. (ms)	
ALEXNET-pool5	Exact.	89.75%	66.28%	83.33%	4.62	53.61	0.74
		E L M N	E L M N	E L M N			
	E	0.99 0.00 0.01 0.00	0.59 0.09 0.15 0.16	0.91 0.00 0.05 0.05			
	L	0.02 0.87 0.08 0.03	0.04 0.73 0.11 0.13	0.06 0.85 0.06 0.04			
	M	0.03 0.03 0.91 0.03	0.11 0.12 0.64 0.13	0.05 0.14 0.62 0.19			
N	0.08 0.03 0.07 0.82	0.15 0.06 0.11 0.69	0.08 0.04 0.04 0.85				
ALEXNET-fc6	Exact.	89.25%	67.43%	80.00%	5.10	41.96	0.36
		E L M N	E L M N	E L M N			
	E	0.98 0.00 0.01 0.01	0.62 0.10 0.14 0.14	0.82 0.00 0.05 0.14			
	L	0.02 0.87 0.09 0.02	0.05 0.76 0.11 0.08	0.04 0.81 0.08 0.08			
	M	0.02 0.03 0.91 0.04	0.11 0.14 0.63 0.13	0.05 0.14 0.57 0.24			
N	0.09 0.05 0.05 0.81	0.14 0.07 0.10 0.69	0.06 0.05 0.05 0.83				
ALEXNET-fc7	Exact.	89.00%	67.03%	79.05%	5.00	1.30	0.34
		E L M N	E L M N	E L M N			
	E	0.98 0.00 0.02 0.00	0.69 0.06 0.12 0.13	0.77 0.09 0.14 0.00			
	L	0.04 0.86 0.09 0.01	0.08 0.72 0.14 0.06	0.02 0.83 0.09 0.06			
	M	0.02 0.02 0.94 0.02	0.15 0.09 0.66 0.10	0.05 0.10 0.71 0.14			
N	0.11 0.02 0.09 0.78	0.19 0.09 0.11 0.61	0.10 0.04 0.07 0.79				
ALEXNET-fc8	Exact.	88.00%	64.00%	77.14%	4.71	1.13	0.17
		E L M N	E L M N	E L M N			
	E	0.98 0.00 0.01 0.01	0.58 0.13 0.14 0.15	0.73 0.09 0.09 0.09			
	L	0.06 0.85 0.08 0.01	0.05 0.74 0.13 0.08	0.04 0.85 0.08 0.04			
	M	0.03 0.00 0.96 0.01	0.13 0.14 0.63 0.10	0.14 0.00 0.81 0.05			
N	0.17 0.05 0.05 0.73	0.17 0.11 0.11 0.62	0.14 0.05 0.07 0.74				
DENSENET201 - avg_pool	Exact.	89.25%	65.73%	71.43%	41.71	1.08	0.20
		E L M N	E L M N	E L M N			
	E	0.98 0.00 0.00 0.02	0.87 0.05 0.04 0.04	0.87 0.05 0.04 0.04			
	L	0.04 0.90 0.03 0.03	0.11 0.79 0.06 0.04	0.11 0.79 0.06 0.04			
	M	0.05 0.01 0.93 0.01	0.35 0.16 0.44 0.05	0.35 0.16 0.44 0.05			
N	0.13 0.08 0.03 0.76	0.25 0.14 0.08 0.53	0.25 0.14 0.08 0.53				
DENSENET201 - fc1000	Exact.	87.25%	63.13%	67.62%	41.72	0.63	0.14
		E L M N	E L M N	E L M N			
	E	0.95 0.01 0.02 0.02	0.79 0.08 0.07 0.06	0.59 0.05 0.09 0.27			
	L	0.01 0.90 0.05 0.04	0.12 0.75 0.08 0.04	0.02 0.89 0.08 0.02			
	M	0.03 0.04 0.90 0.03	0.33 0.17 0.43 0.08	0.24 0.14 0.52 0.10			
N	0.13 0.09 0.04 0.74	0.20 0.15 0.10 0.56	0.24 0.10 0.04 0.62				
GOOGLNET - pool5-7x7_s1	Exact.	87.00%	60.90%	70.00%	10.39	0.48	0.14
		E L M N	E L M N	E L M N			
	E	0.98 0.00 0.00 0.02	0.68 0.07 0.16 0.10	0.68 0.09 0.09 0.14			
	L	0.03 0.87 0.07 0.03	0.13 0.66 0.14 0.06	0.09 0.72 0.11 0.08			
	M	0.02 0.04 0.93 0.01	0.17 0.15 0.61 0.07	0.05 0.10 0.81 0.05			
N	0.11 0.07 0.12 0.70	0.31 0.11 0.09 0.49	0.15 0.08 0.10 0.68				
GOOGLNET - loss3-classifier	Exact.	86.75%	56.43%	67.14%	10.57	1.54	0.14
		E L M N	E L M N	E L M N			
	E	0.97 0.00 0.01 0.02	0.59 0.14 0.18 0.09	0.73 0.09 0.05 0.14			
	L	0.02 0.89 0.06 0.03	0.13 0.68 0.14 0.05	0.09 0.66 0.19 0.06			
	M	0.05 0.01 0.93 0.01	0.17 0.22 0.56 0.06	0.19 0.05 0.71 0.05			
N	0.11 0.06 0.15 0.68	0.29 0.16 0.13 0.43	0.16 0.08 0.11 0.66				
INCEPTIONV2 - avg_pool	Exact.	87.75%	62.03%	74.29%	60.06	0.72	0.18
		E L M N	E L M N	E L M N			
	E	0.96 0.01 0.01 0.02	0.67 0.21 0.06 0.05	0.59 0.18 0.09 0.14			
	L	0.05 0.82 0.06 0.07	0.05 0.82 0.09 0.04	0.02 0.89 0.06 0.04			
	M	0.03 0.02 0.94 0.01	0.15 0.31 0.49 0.05	0.14 0.10 0.71 0.05			
N	0.12 0.05 0.04 0.79	0.15 0.21 0.15 0.49	0.14 0.09 0.06 0.71				
INCEPTIONV2 - predictions	Exact.	83.75%	52.93%	62.86%	63.93	1.20	0.13
		E L M N	E L M N	E L M N			
	E	0.96 0.02 0.00 0.02	0.49 0.37 0.06 0.09	0.55 0.14 0.14 0.18			
	L	0.07 0.74 0.13 0.06	0.03 0.80 0.11 0.06	0.06 0.79 0.08 0.08			
	M	0.03 0.02 0.93 0.02	0.10 0.43 0.38 0.10	0.19 0.05 0.57 0.19			
N	0.14 0.09 0.05 0.72	0.14 0.27 0.14 0.45	0.18 0.14 0.10 0.58				

Tabla 10 – continuación

Resultados Experimento #1. Conjunto de datos balanceado con descomposición 1 a todos

Red-capa		CE1				CE2				CE3				T.Act. (ms)	T.Ent. (s)	T.Clas. (ms)
	Exact.	86.75%				64.13%				70.95%						
		E	L	M	N	E	L	M	N	E	L	M	N			
INCEPTIONV3 – avg_pool	E	0.97	0.01	0.01	0.01	0.61	0.10	0.21	0.08	0.64	0.23	0.09	0.05	41.16	0.92	0.30
	L	0.03	0.89	0.05	0.03	0.03	0.77	0.16	0.05	0.06	0.79	0.06	0.09			
	M	0.04	0.03	0.91	0.02	0.10	0.15	0.70	0.06	0.05	0.14	0.76	0.05			
	N	0.16	0.09	0.05	0.70	0.15	0.16	0.20	0.49	0.18	0.10	0.05	0.68			
INCEPTIONV3 – predictions	Exact.	82.75%				56.95%				64.29%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.95	0.02	0.02	0.01	0.51	0.11	0.26	0.12	0.45	0.23	0.23	0.09	42.80	0.85	0.20
	L	0.04	0.83	0.07	0.06	0.04	0.67	0.21	0.08	0.08	0.68	0.13	0.11			
M	0.04	0.03	0.90	0.03	0.12	0.15	0.63	0.10	0.05	0.14	0.71	0.10				
N	0.16	0.14	0.07	0.63	0.16	0.16	0.21	0.47	0.22	0.04	0.09	0.65				
RESNET18 – pool5	Exact.	87.50%				56.13%				67.62%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.97	0.01	0.02	0.00	0.45	0.26	0.17	0.13	0.64	0.18	0.14	0.05	6.89	0.53	0.09
	L	0.02	0.90	0.06	0.02	0.04	0.79	0.08	0.08	0.02	0.72	0.17	0.09			
M	0.02	0.03	0.91	0.04	0.11	0.28	0.50	0.11	0.05	0.14	0.62	0.19				
N	0.12	0.10	0.06	0.72	0.13	0.22	0.15	0.51	0.16	0.14	0.03	0.68				
RESNET18 – fc1000	Exact.	84.50%				54.53%				67.62%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.96	0.02	0.02	0.00	0.49	0.20	0.13	0.19	0.59	0.18	0.18	0.05	6.77	2.22	0.13
	L	0.03	0.88	0.05	0.04	0.05	0.76	0.08	0.11	0.02	0.77	0.11	0.09			
M	0.02	0.03	0.90	0.05	0.15	0.26	0.39	0.20	0.00	0.14	0.57	0.29				
N	0.16	0.11	0.09	0.64	0.14	0.20	0.13	0.54	0.17	0.12	0.04	0.67				
RESNET50 – avg_pool	Exact.	92.25%				66.50%				83.33%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.99	0.00	0.01	0.00	0.76	0.13	0.04	0.07	0.73	0.09	0.14	0.05	14.72	0.74	0.22
	L	0.01	0.94	0.03	0.02	0.07	0.83	0.05	0.05	0.00	0.92	0.06	0.02			
M	0.03	0.01	0.94	0.02	0.17	0.28	0.49	0.06	0.05	0.05	0.76	0.14				
N	0.07	0.07	0.04	0.82	0.15	0.16	0.10	0.59	0.08	0.07	0.03	0.82				
RESNET50 – fc1000	Exact.	88.25%				63.25%				78.10%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.95	0.01	0.03	0.01	0.67	0.17	0.05	0.11	0.50	0.14	0.18	0.18	15.98	0.68	0.14
	L	0.03	0.89	0.05	0.03	0.05	0.82	0.05	0.08	0.00	0.94	0.04	0.02			
M	0.03	0.03	0.92	0.02	0.15	0.29	0.44	0.11	0.10	0.05	0.71	0.14				
N	0.07	0.08	0.08	0.77	0.14	0.18	0.08	0.60	0.09	0.09	0.05	0.77				
RESNET101 – pool5	Exact.	89.25%				62.68%				77.62%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.97	0.01	0.01	0.01	0.82	0.07	0.02	0.10	0.68	0.09	0.09	0.14	21.02	0.74	0.20
	L	0.01	0.87	0.06	0.06	0.21	0.60	0.06	0.13	0.04	0.87	0.06	0.04			
M	0.00	0.02	0.95	0.03	0.28	0.12	0.46	0.14	0.00	0.10	0.81	0.10				
N	0.11	0.04	0.07	0.78	0.21	0.10	0.07	0.63	0.11	0.10	0.05	0.75				
RESNET101 – fc1000	Exact.	87.75%				56.35%				73.33%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.96	0.00	0.01	0.03	0.71	0.12	0.02	0.15	0.64	0.09	0.05	0.23	21.21	0.72	0.13
	L	0.04	0.84	0.06	0.06	0.18	0.63	0.06	0.14	0.08	0.81	0.06	0.06			
M	0.03	0.00	0.95	0.02	0.31	0.21	0.31	0.17	0.19	0.00	0.76	0.05				
N	0.15	0.03	0.06	0.76	0.20	0.12	0.07	0.61	0.15	0.09	0.05	0.71				
SQUEEZENET –pool10	Exact.	87.50%				60.20%				69.05%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.97	0.00	0.01	0.02	0.69	0.03	0.14	0.13	0.64	0.14	0.09	0.14	6.04	1.22	0.17
	L	0.05	0.85	0.07	0.03	0.15	0.62	0.12	0.12	0.04	0.81	0.11	0.04			
M	0.06	0.01	0.90	0.03	0.29	0.07	0.52	0.13	0.14	0.05	0.71	0.10				
N	0.12	0.04	0.06	0.78	0.30	0.04	0.09	0.58	0.20	0.09	0.07	0.64				

Tabla
Resultados Experimento #2. Conjunto de datos balanceado con descomposición 1 a 1

Red-capa	CE1	CE2	CE3	T.Act. (ms)	T.Ent. (s)	T.Clas. (ms)	
ALEXNET -pool5	Exact.	88.50%	66.65%	82.38%	4.68	34.80	1.04
		E L M N	E L M N	E L M N			
	E	0.98 0.00 0.01 0.01	0.61 0.07 0.19 0.14	0.91 0.00 0.05 0.05			
	L	0.01 0.85 0.10 0.04	0.04 0.69 0.14 0.14	0.02 0.83 0.08 0.08			
	M	0.02 0.04 0.92 0.02	0.12 0.08 0.69 0.11	0.00 0.19 0.67 0.14			
N	0.11 0.02 0.08 0.79	0.17 0.03 0.12 0.69	0.13 0.01 0.03 0.83				
ALEXNET -fc6	Exact.	88.00%	70.33%	76.67%	4.73	25.06	0.50
		E L M N	E L M N	E L M N			
	E	0.96 0.00 0.04 0.00	0.65 0.07 0.16 0.12	0.68 0.09 0.18 0.05			
	L	0.02 0.85 0.11 0.02	0.04 0.76 0.11 0.09	0.04 0.79 0.09 0.08			
	M	0.02 0.04 0.92 0.02	0.10 0.10 0.70 0.09	0.00 0.19 0.67 0.14			
N	0.13 0.04 0.04 0.79	0.17 0.03 0.10 0.71	0.11 0.04 0.06 0.79				
ALEXNET -fc7	Exact.	89.25%	67.40%	76.67%	4.70	0.59	0.49
		E L M N	E L M N	E L M N			
	E	0.98 0.00 0.02 0.00	0.68 0.07 0.14 0.11	0.77 0.09 0.14 0.00			
	L	0.03 0.86 0.09 0.02	0.05 0.78 0.12 0.05	0.06 0.85 0.06 0.04			
	M	0.02 0.02 0.95 0.01	0.14 0.12 0.64 0.10	0.14 0.05 0.67 0.14			
N	0.11 0.03 0.08 0.78	0.23 0.06 0.11 0.60	0.14 0.03 0.09 0.75				
ALEXNET -fc8	Exact.	88.25%	64.98%	74.76%	4.77	0.28	0.19
		E L M N	E L M N	E L M N			
	E	0.98 0.00 0.01 0.01	0.58 0.09 0.20 0.13	0.77 0.09 0.09 0.05			
	L	0.04 0.86 0.08 0.02	0.04 0.75 0.14 0.07	0.06 0.83 0.09 0.02			
	M	0.04 0.01 0.94 0.01	0.11 0.13 0.67 0.10	0.10 0.05 0.71 0.14			
N	0.14 0.05 0.06 0.75	0.21 0.07 0.12 0.60	0.15 0.08 0.06 0.71				
DENSENET201 - avg_pool	Exact.	91.00%	67.28%	70.48%	42.04	0.27	0.29
		E L M N	E L M N	E L M N			
	E	0.98 0.00 0.00 0.02	0.88 0.03 0.04 0.05	0.64 0.14 0.05 0.18			
	L	0.02 0.92 0.03 0.03	0.11 0.77 0.07 0.05	0.04 0.89 0.06 0.02			
	M	0.02 0.02 0.94 0.02	0.38 0.11 0.47 0.04	0.14 0.05 0.76 0.05			
N	0.09 0.08 0.03 0.80	0.26 0.09 0.08 0.57	0.21 0.08 0.09 0.62				
DENSENET201 - fc1000	Exact.	90.50%	65.15%	72.38%	42.34	0.21	0.19
		E L M N	E L M N	E L M N			
	E	0.98 0.00 0.00 0.02	0.84 0.05 0.06 0.05	0.68 0.05 0.09 0.18			
	L	0.00 0.90 0.05 0.05	0.12 0.77 0.08 0.04	0.02 0.89 0.08 0.02			
	M	0.02 0.03 0.92 0.03	0.37 0.14 0.44 0.05	0.05 0.10 0.76 0.10			
N	0.08 0.06 0.04 0.82	0.23 0.13 0.09 0.56	0.21 0.07 0.07 0.65				
GOOGLNET - pool5-7x7_s1	Exact.	89.00%	64.85%	74.76%	10.10	0.19	0.26
		E L M N	E L M N	E L M N			
	E	0.98 0.00 0.00 0.02	0.69 0.05 0.18 0.08	0.73 0.09 0.00 0.18			
	L	0.02 0.89 0.07 0.02	0.10 0.72 0.14 0.05	0.08 0.77 0.08 0.08			
	M	0.03 0.01 0.95 0.01	0.15 0.14 0.67 0.05	0.00 0.05 0.86 0.10			
N	0.12 0.06 0.08 0.74	0.26 0.09 0.14 0.51	0.16 0.05 0.07 0.72				
GOOGLNET - loss3-classifier	Exact.	88.75%	60.55%	68.57%	10.64	0.40	0.23
		E L M N	E L M N	E L M N			
	E	0.98 0.00 0.00 0.02	0.66 0.09 0.17 0.09	0.73 0.14 0.00 0.14			
	L	0.01 0.91 0.07 0.01	0.09 0.73 0.14 0.04	0.09 0.70 0.13 0.08			
	M	0.04 0.04 0.91 0.01	0.17 0.22 0.57 0.05	0.05 0.14 0.71 0.10			
N	0.13 0.05 0.07 0.75	0.26 0.09 0.18 0.47	0.17 0.08 0.09 0.67				
INCEPTIONV2 - avg_pool	Exact.	87.25%	62.28%	69.52%	62.26	0.33	0.34
		E L M N	E L M N	E L M N			
	E	0.96 0.01 0.00 0.03	0.71 0.18 0.06 0.06	0.64 0.14 0.05 0.18			
	L	0.03 0.80 0.11 0.06	0.05 0.84 0.08 0.04	0.06 0.77 0.13 0.04			
	M	0.02 0.02 0.95 0.01	0.18 0.32 0.45 0.05	0.14 0.05 0.71 0.10			
N	0.12 0.08 0.02 0.78	0.16 0.23 0.12 0.49	0.14 0.11 0.09 0.67				
INCEPTIONV2 - predictions	Exact.	84.25%	55.78%	65.24%	57.80	0.36	0.24
		E L M N	E L M N	E L M N			
	E	0.96 0.01 0.00 0.03	0.56 0.29 0.07 0.09	0.59 0.14 0.05 0.23			
	L	0.03 0.77 0.15 0.05	0.04 0.80 0.11 0.05	0.06 0.77 0.11 0.06			
	M	0.05 0.02 0.91 0.02	0.13 0.36 0.40 0.12	0.29 0.05 0.52 0.14			
N	0.13 0.09 0.05 0.73	0.13 0.25 0.14 0.48	0.17 0.11 0.10 0.63				

Tabla 11 - continuación

Resultados Experimento #2. Conjunto de datos balanceado con descomposición 1 a 1

Red-capa		CE1				CE2				CE3				T.Act. (ms)	T.Ent. (s)	T.Clas. (ms)
	Exact.	87.75%				63.83%				70.95%						
		E	L	M	N	E	L	M	N	E	L	M	N			
INCEPTIONV3 – avg_pool	E	0.96	0.01	0.03	0.00	0.53	0.07	0.33	0.07	0.68	0.14	0.18	0.00	43.06	0.44	0.40
	L	0.01	0.89	0.07	0.03	0.02	0.76	0.17	0.04	0.04	0.79	0.08	0.09			
	M	0.03	0.03	0.91	0.03	0.07	0.12	0.76	0.05	0.10	0.10	0.71	0.10			
	N	0.13	0.08	0.04	0.75	0.17	0.13	0.21	0.50	0.16	0.12	0.05	0.67			
	Exact.	82.75%				58.93%				70.00%						
		E	L	M	N	E	L	M	N	E	L	M	N			
INCEPTIONV3 – predictions	E	0.96	0.01	0.02	0.01	0.46	0.09	0.36	0.09	0.64	0.18	0.14	0.05	41.68	0.24	0.19
	L	0.02	0.84	0.08	0.06	0.03	0.71	0.20	0.06	0.02	0.75	0.09	0.13			
	M	0.05	0.04	0.89	0.02	0.08	0.15	0.70	0.07	0.10	0.14	0.67	0.10			
	N	0.13	0.10	0.08	0.69	0.16	0.14	0.22	0.49	0.10	0.11	0.10	0.69			
	Exact.	90.00%				58.80%				67.14%						
		E	L	M	N	E	L	M	N	E	L	M	N			
RESNET18 – pool5	E	0.98	0.00	0.02	0.00	0.46	0.25	0.17	0.12	0.64	0.18	0.14	0.05	6.82	0.18	0.14
	L	0.02	0.88	0.05	0.05	0.02	0.82	0.09	0.07	0.04	0.70	0.15	0.11			
	M	0.01	0.02	0.96	0.01	0.07	0.26	0.55	0.11	0.05	0.10	0.76	0.10			
	N	0.13	0.06	0.03	0.78	0.12	0.22	0.14	0.52	0.21	0.11	0.04	0.65			
	Exact.	88.25%				57.53%				66.67%						
		E	L	M	N	E	L	M	N	E	L	M	N			
RESNET18 – fc1000	E	0.98	0.00	0.02	0.00	0.48	0.24	0.16	0.13	0.68	0.18	0.14	0.00	6.72	0.33	0.19
	L	0.04	0.85	0.07	0.04	0.03	0.80	0.10	0.08	0.02	0.74	0.17	0.08			
	M	0.01	0.03	0.93	0.03	0.09	0.28	0.52	0.12	0.10	0.14	0.67	0.10			
	N	0.11	0.08	0.04	0.77	0.12	0.23	0.14	0.51	0.18	0.14	0.04	0.63			
	Exact.	92.25%				66.75%				81.91%						
		E	L	M	N	E	L	M	N	E	L	M	N			
RESNET50 – avg_pool	E	0.97	0.01	0.02	0.00	0.76	0.15	0.04	0.05	0.64	0.18	0.14	0.05	15.02	0.31	0.30
	L	0.01	0.94	0.04	0.01	0.04	0.89	0.04	0.03	0.00	0.94	0.06	0.00			
	M	0.03	0.01	0.96	0.00	0.21	0.30	0.45	0.04	0.05	0.05	0.90	0.00			
	N	0.10	0.05	0.03	0.82	0.18	0.18	0.08	0.56	0.10	0.09	0.04	0.78			
	Exact.	91.00%				61.38%				80.00%						
		E	L	M	N	E	L	M	N	E	L	M	N			
RESNET50 – fc1000	E	0.97	0.00	0.03	0.00	0.62	0.23	0.07	0.08	0.68	0.09	0.18	0.05	16.43	0.23	0.32
	L	0.02	0.93	0.04	0.01	0.04	0.87	0.05	0.04	0.00	0.94	0.06	0.00			
	M	0.02	0.02	0.96	0.00	0.15	0.37	0.42	0.06	0.05	0.05	0.90	0.00			
	N	0.12	0.05	0.05	0.78	0.16	0.20	0.09	0.54	0.11	0.11	0.05	0.74			
	Exact.	89.25%				65.68%				75.71%						
		E	L	M	N	E	L	M	N	E	L	M	N			
RESNET101 – pool5	E	0.97	0.01	0.01	0.01	0.83	0.05	0.03	0.09	0.64	0.18	0.05	0.14	21.03	0.29	0.31
	L	0.01	0.90	0.05	0.04	0.16	0.66	0.08	0.10	0.02	0.89	0.06	0.04			
	M	0.00	0.01	0.97	0.02	0.32	0.08	0.51	0.09	0.05	0.05	0.86	0.05			
	N	0.14	0.03	0.07	0.76	0.20	0.12	0.07	0.62	0.15	0.08	0.07	0.70			
	Exact.	87.75%				58.15%				71.43%						
		E	L	M	N	E	L	M	N	E	L	M	N			
RESNET101 – fc1000	E	0.97	0.00	0.02	0.01	0.75	0.05	0.03	0.17	0.59	0.14	0.09	0.18	21.05	0.28	0.23
	L	0.02	0.85	0.06	0.07	0.17	0.63	0.05	0.15	0.04	0.81	0.08	0.08			
	M	0.01	0.00	0.95	0.04	0.40	0.11	0.33	0.17	0.05	0.00	0.86	0.10			
	N	0.18	0.05	0.06	0.71	0.19	0.13	0.06	0.62	0.17	0.11	0.06	0.67			
	Exact.	87.50%				61.40%				71.43%						
		E	L	M	N	E	L	M	N	E	L	M	N			
SQUEEZENET -pool10	E	0.97	0.00	0.03	0.00	0.72	0.02	0.14	0.11	0.68	0.09	0.23	0.00	5.97	0.31	0.19
	L	0.03	0.88	0.06	0.03	0.15	0.60	0.13	0.12	0.00	0.91	0.08	0.02			
	M	0.02	0.01	0.95	0.02	0.32	0.06	0.51	0.11	0.10	0.05	0.81	0.05			
	N	0.14	0.02	0.05	0.79	0.27	0.02	0.08	0.63	0.25	0.07	0.07	0.61			

Tabla
Resultados Experimento #3. Conjunto de datos desbalanceado con descomposición 1 a todos

Red-capa		CE1				CE2				CE3				T.Act. (ms)	T.Ent. (s)	T.Clas. (ms)
	Exact.	97.75%				64.05%				87.62%						
		E	L	M	N	E	L	M	N	E	L	M	N			
ALEXNET-pool5	E	0.96	0.00	0.00	0.04	0.47	0.10	0.10	0.34	0.64	0.00	0.05	0.32	6.08	162.40	0.71
	L	0.00	1.00	0.00	0.00	0.02	0.79	0.05	0.15	0.00	0.91	0.02	0.08			
	M	0.00	0.01	0.95	0.04	0.09	0.15	0.46	0.31	0.00	0.14	0.57	0.29			
	N	0.00	0.00	0.00	1.00	0.05	0.06	0.05	0.85	0.01	0.03	0.00	0.96			
ALEXNET-fc6	Exact.	97.25%				64.70%				86.67%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.96	0.00	0.00	0.04	0.53	0.08	0.10	0.29	0.68	0.00	0.05	0.27	5.91	125.65	0.32
	L	0.00	1.00	0.00	0.00	0.03	0.78	0.08	0.12	0.00	0.92	0.00	0.08			
M	0.00	0.03	0.93	0.04	0.13	0.14	0.46	0.27	0.00	0.19	0.52	0.29				
N	0.00	0.00	0.00	1.00	0.07	0.05	0.06	0.82	0.02	0.03	0.02	0.94				
ALEXNET-fc7	Exact.	96.50%				65.70%				81.43%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.97	0.01	0.00	0.02	0.56	0.06	0.09	0.29	0.64	0.14	0.05	0.18	5.97	33.55	0.26
	L	0.01	0.96	0.00	0.03	0.04	0.77	0.08	0.11	0.02	0.85	0.04	0.09			
M	0.00	0.03	0.93	0.04	0.12	0.12	0.49	0.26	0.10	0.24	0.29	0.38				
N	0.00	0.00	0.00	1.00	0.06	0.05	0.09	0.80	0.03	0.04	0.01	0.93				
ALEXNET-fc8	Exact.	96.00%				62.88%				80.95%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.95	0.01	0.01	0.03	0.45	0.15	0.12	0.28	0.64	0.05	0.05	0.27	5.79	12.81	0.14
	L	0.01	0.97	0.00	0.02	0.03	0.81	0.05	0.11	0.02	0.91	0.00	0.08			
M	0.01	0.03	0.93	0.03	0.08	0.22	0.47	0.24	0.10	0.24	0.43	0.24				
N	0.00	0.01	0.00	0.99	0.05	0.10	0.07	0.78	0.02	0.07	0.04	0.87				
DENSENET201 – avg_pool	Exact.	97.00%				68.83%				80.95%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.97	0.00	0.00	0.03	0.78	0.07	0.03	0.13	0.55	0.09	0.00	0.36	50.91	3.68	0.16
	L	0.00	0.98	0.01	0.01	0.08	0.83	0.03	0.05	0.00	0.91	0.08	0.02			
M	0.00	0.01	0.96	0.03	0.31	0.16	0.37	0.16	0.10	0.10	0.48	0.33				
N	0.02	0.01	0.00	0.97	0.12	0.08	0.04	0.77	0.04	0.04	0.04	0.88				
DENSENET201 – fc1000	Exact.	96.50%				65.33%				80.95%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.96	0.00	0.01	0.03	0.80	0.04	0.04	0.12	0.55	0.05	0.09	0.32	51.63	9.38	0.12
	L	0.00	0.98	0.01	0.01	0.15	0.73	0.04	0.08	0.00	0.94	0.04	0.02			
M	0.00	0.02	0.95	0.03	0.39	0.14	0.32	0.15	0.05	0.19	0.52	0.24				
N	0.02	0.01	0.00	0.97	0.11	0.06	0.06	0.76	0.06	0.05	0.04	0.85				
GOOGLENET – pool5-7x7_s1	Exact.	96.25%				62.83%				77.62%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.94	0.00	0.00	0.06	0.52	0.09	0.15	0.25	0.45	0.05	0.00	0.50	12.35	5.00	0.11
	L	0.01	0.97	0.00	0.02	0.05	0.73	0.09	0.13	0.08	0.83	0.00	0.09			
M	0.00	0.01	0.95	0.04	0.12	0.16	0.51	0.22	0.05	0.14	0.57	0.24				
N	0.01	0.00	0.00	0.99	0.11	0.07	0.06	0.76	0.07	0.05	0.03	0.85				
GOOGLENET – loss3-classifier	Exact.	95.75%				61.28%				75.71%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.95	0.00	0.00	0.05	0.51	0.13	0.17	0.20	0.50	0.05	0.00	0.45	12.56	34.44	0.11
	L	0.01	0.96	0.00	0.03	0.05	0.71	0.15	0.10	0.08	0.81	0.00	0.11			
M	0.00	0.01	0.94	0.05	0.13	0.17	0.50	0.20	0.05	0.14	0.52	0.29				
N	0.01	0.00	0.01	0.98	0.11	0.08	0.07	0.73	0.05	0.06	0.06	0.82				
INCEPTIONV2 – avg_pool	Exact.	94.75%				67.60%				79.05%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.91	0.02	0.02	0.05	0.65	0.13	0.10	0.11	0.36	0.23	0.09	0.32	88.17	7.51	1.03
	L	0.01	0.95	0.01	0.03	0.10	0.80	0.05	0.05	0.04	0.89	0.02	0.06			
M	0.02	0.00	0.95	0.03	0.19	0.17	0.53	0.10	0.10	0.14	0.52	0.24				
N	0.00	0.02	0.00	0.98	0.06	0.12	0.10	0.73	0.03	0.08	0.02	0.88				
INCEPTIONV2 – predictions	Exact.	92.50%				63.35%				76.19%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.89	0.01	0.03	0.07	0.56	0.15	0.12	0.17	0.32	0.14	0.14	0.41	74.06	21.57	0.16
	L	0.01	0.94	0.01	0.04	0.07	0.77	0.07	0.09	0.06	0.87	0.02	0.06			
M	0.01	0.01	0.90	0.08	0.18	0.20	0.47	0.16	0.10	0.10	0.43	0.38				
N	0.00	0.02	0.01	0.97	0.05	0.14	0.08	0.73	0.02	0.10	0.03	0.86				

Tabla 12 - continuación

Resultados Experimento #3. Conjunto de datos desbalanceado con descomposición 1 a todos

Red-capa		CE1				CE2				CE3				T.Act. (ms)	T.Ent. (s)	T.Clas. (ms)
	Exact.	95.75%				67.55%				82.38%						
		E	L	M	N	E	L	M	N	E	L	M	N			
INCEPTIONV3 – avg_pool	E	0.59	0.23	0.05	0.14	0.55	0.10	0.18	0.17	0.59	0.23	0.05	0.14	50.33	4.57	0.29
	L	0.02	0.89	0.02	0.08	0.02	0.80	0.11	0.08	0.02	0.89	0.02	0.08			
	M	0.10	0.19	0.48	0.24	0.07	0.16	0.58	0.18	0.10	0.19	0.48	0.24			
	N	0.02	0.08	0.00	0.90	0.04	0.10	0.09	0.77	0.02	0.08	0.00	0.90			
INCEPTIONV3 – predictions	Exact.	95.25%				61.05%				76.67%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.93	0.02	0.02	0.03	0.43	0.14	0.21	0.22	0.41	0.27	0.09	0.23	48.81	14.12	0.15
	L	0.01	0.96	0.00	0.03	0.02	0.74	0.13	0.11	0.02	0.83	0.04	0.11			
M	0.01	0.02	0.93	0.04	0.09	0.19	0.53	0.20	0.10	0.24	0.48	0.19				
N	0.01	0.00	0.00	0.99	0.05	0.12	0.09	0.74	0.04	0.06	0.04	0.86				
RESNET18 – pool5	Exact.	92.75%				54.28%				78.10%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.87	0.02	0.00	0.11	0.36	0.22	0.09	0.33	0.45	0.23	0.05	0.27	8.61	8.27	0.11
	L	0.01	0.91	0.01	0.07	0.03	0.77	0.03	0.17	0.02	0.77	0.08	0.13			
M	0.00	0.00	0.95	0.05	0.12	0.30	0.28	0.30	0.00	0.10	0.67	0.24				
N	0.00	0.01	0.01	0.98	0.08	0.12	0.04	0.76	0.02	0.10	0.02	0.87				
RESNET18 – fc1000	Exact.	92.50%				52.83%				73.81%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.87	0.03	0.02	0.08	0.33	0.24	0.10	0.33	0.50	0.27	0.14	0.09	8.35	67.26	0.15
	L	0.01	0.91	0.01	0.07	0.03	0.77	0.04	0.16	0.06	0.72	0.08	0.15			
M	0.00	0.01	0.94	0.05	0.09	0.34	0.27	0.30	0.00	0.19	0.43	0.38				
N	0.00	0.00	0.02	0.98	0.07	0.14	0.05	0.75	0.04	0.09	0.03	0.85				
RESNET50 – avg_pool	Exact.	98.25%				71.20%				89.05%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.98	0.00	0.00	0.02	0.66	0.12	0.05	0.16	0.77	0.05	0.00	0.18	19.49	3.52	0.22
	L	0.00	0.99	0.01	0.00	0.06	0.85	0.03	0.06	0.00	0.92	0.04	0.04			
M	0.00	0.01	0.96	0.03	0.15	0.21	0.51	0.13	0.00	0.14	0.62	0.24				
N	0.00	0.00	0.00	1.00	0.06	0.08	0.04	0.83	0.00	0.04	0.01	0.95				
RESNET50 – fc1000	Exact.	96.75%				64.00%				84.29%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.96	0.00	0.00	0.04	0.57	0.17	0.04	0.23	0.59	0.09	0.00	0.32	19.50	7.02	0.15
	L	0.00	1.00	0.00	0.00	0.04	0.84	0.02	0.10	0.00	0.96	0.02	0.02			
M	0.00	0.01	0.93	0.06	0.19	0.30	0.34	0.17	0.00	0.14	0.48	0.38				
N	0.01	0.00	0.01	0.98	0.06	0.09	0.03	0.82	0.03	0.04	0.04	0.90				
RESNET101 – pool5	Exact.	97.50%				68.58%				85.71%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.97	0.02	0.00	0.01	0.78	0.08	0.02	0.12	0.59	0.18	0.00	0.23	30.45	4.19	0.23
	L	0.00	0.98	0.00	0.02	0.16	0.77	0.02	0.06	0.00	0.96	0.00	0.04			
M	0.01	0.01	0.95	0.03	0.34	0.17	0.38	0.11	0.05	0.14	0.62	0.19				
N	0.00	0.00	0.00	1.00	0.10	0.06	0.02	0.82	0.01	0.05	0.04	0.90				
RESNET101 – fc1000	Exact.	97.00%				63.45%				81.43%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.95	0.02	0.01	0.02	0.61	0.14	0.02	0.23	0.50	0.23	0.05	0.23	28.92	10.81	0.15
	L	0.00	0.98	0.00	0.02	0.09	0.79	0.02	0.10	0.04	0.87	0.02	0.08			
M	0.00	0.01	0.96	0.03	0.24	0.23	0.31	0.23	0.05	0.14	0.67	0.14				
N	0.01	0.00	0.00	0.99	0.07	0.07	0.03	0.83	0.04	0.07	0.02	0.88				
SQUEEZENET –pool10	Exact.	96.50%				63.15%				81.43%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.95	0.00	0.01	0.04	0.55	0.07	0.07	0.32	0.50	0.14	0.09	0.27	7.41	14.13	0.14
	L	0.00	0.98	0.01	0.01	0.07	0.71	0.03	0.20	0.00	0.89	0.02	0.09			
M	0.01	0.00	0.96	0.03	0.17	0.14	0.37	0.32	0.05	0.10	0.62	0.24				
N	0.02	0.00	0.01	0.97	0.04	0.03	0.03	0.90	0.04	0.05	0.03	0.88				

Tabla
Resultados Experimento #4. Conjunto de datos desbalanceado con descomposición 1 a 1

Red-capa		CE1				CE2				CE3				T.Act. (ms)	T.Ent. (s)	T.Clas. (ms)
	Exact.	97.25%				64.50%				87.14%						
		E	L	M	N	E	L	M	N	E	L	M	N			
ALEXNET -pool5	E	0.95	0.00	0.01	0.04	0.41	0.09	0.14	0.36	0.64	0.00	0.05	0.32	5.87	107.50	1.00
	L	0.00	1.00	0.00	0.00	0.01	0.79	0.06	0.15	0.02	0.91	0.02	0.06			
	M	0.00	0.01	0.94	0.05	0.07	0.12	0.50	0.30	0.00	0.14	0.57	0.29			
	N	0.00	0.00	0.00	1.00	0.04	0.04	0.04	0.88	0.01	0.04	0.00	0.96			
ALEXNET-fc6	Exact.	97.25%				66.58%				76.67%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.97	0.01	0.00	0.02	0.47	0.09	0.15	0.30	0.73	0.05	0.09	0.14	6.16	71.21	0.51
	L	0.00	0.99	0.00	0.01	0.01	0.81	0.06	0.11	0.00	0.91	0.00	0.09			
M	0.00	0.02	0.94	0.04	0.08	0.13	0.53	0.26	0.00	0.19	0.57	0.24				
N	0.00	0.01	0.00	0.99	0.05	0.05	0.06	0.85	0.01	0.05	0.01	0.93				
ALEXNET-fc7	Exact.	97.25%				67.50%				83.81%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.98	0.01	0.00	0.01	0.58	0.07	0.15	0.21	0.73	0.14	0.05	0.09	5.84	2.55	0.51
	L	0.00	0.98	0.00	0.02	0.04	0.79	0.08	0.09	0.00	0.92	0.02	0.06			
M	0.00	0.01	0.93	0.06	0.12	0.12	0.54	0.22	0.05	0.19	0.38	0.38				
N	0.00	0.00	0.00	1.00	0.07	0.06	0.07	0.80	0.04	0.04	0.02	0.90				
ALEXNET-fc8	Exact.	96.75%				64.45%				81.43%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.96	0.01	0.01	0.02	0.50	0.09	0.16	0.25	0.64	0.14	0.05	0.18	5.88	2.17	0.20
	L	0.00	0.98	0.00	0.02	0.03	0.81	0.05	0.11	0.00	0.89	0.04	0.08			
M	0.02	0.01	0.94	0.03	0.11	0.15	0.50	0.25	0.19	0.19	0.43	0.19				
N	0.00	0.00	0.01	0.99	0.08	0.07	0.07	0.78	0.04	0.04	0.04	0.89				
DENSENET201 – avg_pool	Exact.	91.00%				71.43%				84.76%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.97	0.00	0.00	0.03	0.81	0.04	0.05	0.10	0.59	0.09	0.00	0.32	55.26	1.36	0.30
	L	0.01	0.98	0.00	0.01	0.09	0.78	0.07	0.05	0.02	0.92	0.04	0.02			
M	0.00	0.01	0.96	0.03	0.31	0.10	0.48	0.12	0.05	0.14	0.62	0.19				
N	0.01	0.00	0.00	0.99	0.11	0.05	0.06	0.79	0.03	0.04	0.04	0.90				
DENSENET201 – fc1000	Exact.	97.50%				68.73%				82.86%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.97	0.00	0.01	0.02	0.79	0.06	0.05	0.10	0.59	0.14	0.05	0.23	53.82	1.58	0.35
	L	0.00	0.97	0.01	0.02	0.09	0.76	0.07	0.08	0.00	0.89	0.08	0.04			
M	0.00	0.00	0.97	0.03	0.30	0.15	0.44	0.12	0.10	0.05	0.62	0.24				
N	0.01	0.00	0.00	0.99	0.11	0.06	0.07	0.76	0.04	0.04	0.04	0.89				
GOOGLENET – pool5-7x7_s1	Exact.	97.25%				67.33%				81.91%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.94	0.00	0.00	0.06	0.57	0.06	0.15	0.23	0.50	0.05	0.00	0.45	12.93	1.12	0.19
	L	0.00	0.98	0.00	0.02	0.03	0.76	0.11	0.10	0.06	0.81	0.00	0.13			
M	0.00	0.00	0.97	0.03	0.10	0.14	0.61	0.16	0.00	0.05	0.81	0.14				
N	0.00	0.00	0.00	1.00	0.11	0.05	0.09	0.76	0.04	0.05	0.03	0.89				
GOOGLENET – loss3-classifier	Exact.	96.25%				63.48%				78.10%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.95	0.00	0.00	0.05	0.50	0.08	0.20	0.22	0.55	0.05	0.00	0.41	13.16	3.96	0.25
	L	0.00	0.98	0.00	0.02	0.04	0.71	0.16	0.09	0.06	0.79	0.02	0.13			
M	0.00	0.01	0.94	0.05	0.10	0.16	0.61	0.13	0.05	0.14	0.52	0.29				
N	0.00	0.01	0.01	0.98	0.09	0.06	0.13	0.73	0.04	0.05	0.04	0.87				
INCEPTIONV2 – avg_pool	Exact.	95.75%				68.08%				78.10%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.95	0.01	0.00	0.04	0.63	0.11	0.10	0.16	0.50	0.09	0.00	0.41	77.32	1.73	0.25
	L	0.01	0.96	0.00	0.03	0.04	0.85	0.04	0.07	0.04	0.85	0.04	0.08			
M	0.03	0.00	0.93	0.04	0.14	0.19	0.49	0.18	0.19	0.00	0.48	0.33				
N	0.00	0.01	0.00	0.99	0.05	0.14	0.07	0.75	0.04	0.09	0.01	0.86				
INCEPTIONV2 – predictions	Exact.	95.75%				62.75%				74.76%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.94	0.01	0.00	0.05	0.57	0.13	0.10	0.20	0.41	0.09	0.00	0.50	81.35	3.89	0.27
	L	0.01	0.96	0.00	0.03	0.04	0.77	0.08	0.12	0.04	0.81	0.04	0.11			
M	0.01	0.00	0.95	0.04	0.14	0.21	0.45	0.20	0.05	0.00	0.62	0.33				
N	0.00	0.02	0.00	0.98	0.06	0.16	0.07	0.72	0.07	0.09	0.04	0.81				

Tabla 13 - continuación

Resultados Experimento #4. Conjunto de datos desbalanceado con descomposición 1 a 1

Red-capa		CE1				CE2				CE3				T.Act. (ms)	T.Ent. (s)	T.Clas. (ms)
	Exact.	97.00%				67.28%				82.38%						
		E	L	M	N	E	L	M	N	E	L	M	N			
INCEPTIONV3 – avg_pool	E	0.95	0.02	0.01	0.02	0.50	0.09	0.23	0.18	0.55	0.18	0.09	0.18	32.54	1.65	0.30
	L	0.01	0.96	0.00	0.03	0.01	0.80	0.13	0.07	0.02	0.83	0.04	0.11			
	M	0.00	0.01	0.97	0.02	0.06	0.15	0.63	0.17	0.05	0.14	0.67	0.14			
	N	0.00	0.00	0.00	1.00	0.04	0.11	0.08	0.77	0.02	0.06	0.02	0.90			
INCEPTIONV3 – predictions	Exact.	96.50%				60.85%				80.95%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.94	0.03	0.01	0.02	0.39	0.11	0.33	0.18	0.55	0.27	0.05	0.14	31.16	1.79	0.21
	L	0.01	0.96	0.00	0.03	0.01	0.72	0.18	0.09	0.02	0.83	0.06	0.09			
M	0.00	0.02	0.96	0.02	0.05	0.16	0.62	0.17	0.05	0.19	0.57	0.19				
N	0.00	0.00	0.00	1.00	0.05	0.15	0.10	0.70	0.04	0.04	0.02	0.89				
RESNET18 – pool5	Exact.	96.25%				56.70%				80.48%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.96	0.02	0.01	0.01	0.39	0.19	0.11	0.31	0.55	0.18	0.09	0.18	8.73	1.35	0.14
	L	0.00	0.97	0.00	0.03	0.01	0.80	0.04	0.16	0.00	0.81	0.06	0.13			
M	0.00	0.00	0.97	0.03	0.07	0.28	0.32	0.34	0.00	0.10	0.67	0.24				
N	0.04	0.01	0.00	0.95	0.06	0.13	0.05	0.76	0.06	0.05	0.01	0.88				
RESNET18 – fc1000	Exact.	97.00%				54.28%				79.52%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.96	0.02	0.02	0.00	0.35	0.22	0.13	0.30	0.55	0.18	0.14	0.14	8.58	6.76	0.20
	L	0.00	0.98	0.00	0.02	0.01	0.77	0.07	0.15	0.00	0.83	0.09	0.08			
M	0.00	0.00	0.96	0.04	0.06	0.31	0.31	0.31	0.00	0.05	0.67	0.29				
N	0.01	0.00	0.01	0.98	0.07	0.16	0.04	0.73	0.06	0.07	0.02	0.85				
RESNET50 – avg_pool	Exact.	98.00%				73.98%				88.10%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.96	0.00	0.00	0.04	0.64	0.10	0.08	0.18	0.59	0.05	0.00	0.36	19.92	1.57	0.31
	L	0.00	1.00	0.00	0.00	0.04	0.87	0.03	0.06	0.00	0.92	0.02	0.06			
M	0.01	0.01	0.96	0.02	0.09	0.19	0.58	0.14	0.05	0.10	0.67	0.19				
N	0.00	0.00	0.00	1.00	0.04	0.07	0.03	0.87	0.01	0.03	0.01	0.96				
RESNET50 – fc1000	Exact.	97.25%				65.48%				85.71%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.95	0.00	0.01	0.04	0.47	0.20	0.08	0.25	0.55	0.05	0.09	0.32	20.95	1.42	0.26
	L	0.00	0.98	0.01	0.01	0.02	0.89	0.02	0.07	0.00	0.89	0.04	0.08			
M	0.01	0.01	0.97	0.01	0.08	0.29	0.44	0.19	0.05	0.10	0.71	0.14				
N	0.01	0.00	0.00	0.99	0.05	0.11	0.02	0.82	0.02	0.04	0.01	0.93				
RESNET101 – pool5	Exact.	98.25%				72.63%				85.24%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.97	0.02	0.00	0.01	0.75	0.07	0.04	0.14	0.59	0.14	0.00	0.27	29.38	1.72	0.32
	L	0.00	0.98	0.00	0.02	0.08	0.82	0.05	0.06	0.00	0.94	0.02	0.04			
M	0.00	0.00	0.98	0.02	0.27	0.11	0.49	0.13	0.00	0.10	0.76	0.14				
N	0.00	0.00	0.00	1.00	0.08	0.05	0.03	0.84	0.05	0.05	0.02	0.88				
RESNET101 – fc1000	Exact.	96.75%				65.13%				79.05%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.95	0.02	0.01	0.02	0.68	0.08	0.04	0.21	0.41	0.14	0.05	0.41	31.41	2.37	0.20
	L	0.00	0.97	0.01	0.02	0.10	0.78	0.04	0.09	0.00	0.87	0.06	0.08			
M	0.01	0.00	0.97	0.02	0.32	0.17	0.34	0.18	0.05	0.10	0.71	0.14				
N	0.02	0.00	0.00	0.98	0.08	0.08	0.03	0.82	0.06	0.05	0.04	0.84				
SQUEEZENET –pool10	Exact.	96.75%				66.15%				80.48%						
		E	L	M	N	E	L	M	N	E	L	M	N			
	E	0.95	0.00	0.03	0.02	0.59	0.04	0.12	0.26	0.50	0.09	0.18	0.23	7.67	2.45	0.18
	L	0.00	0.98	0.00	0.02	0.06	0.73	0.06	0.15	0.00	0.92	0.02	0.06			
M	0.00	0.01	0.96	0.03	0.19	0.11	0.44	0.26	0.00	0.14	0.62	0.24				
N	0.01	0.00	0.01	0.98	0.06	0.02	0.03	0.88	0.08	0.03	0.05	0.84				

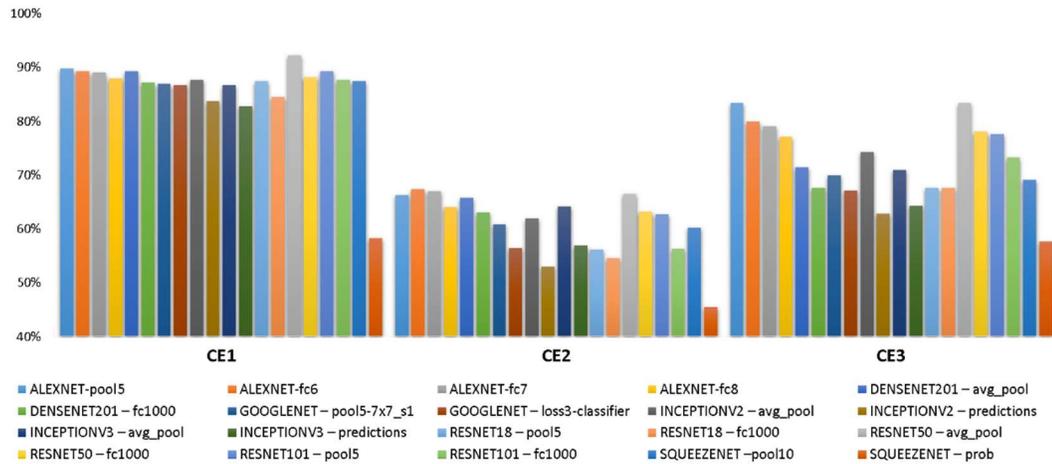


Figura . Exactitud global en el experimento #1

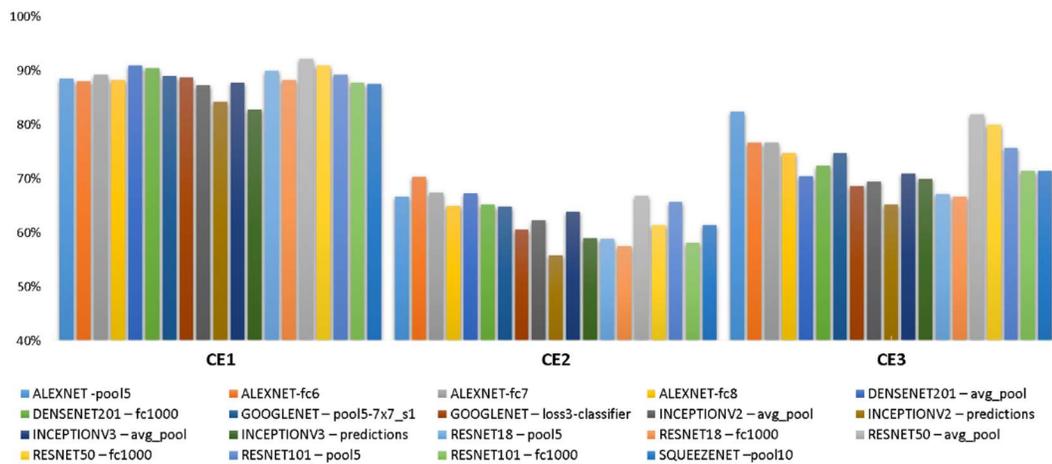


Figura . Exactitud global en el experimento #2

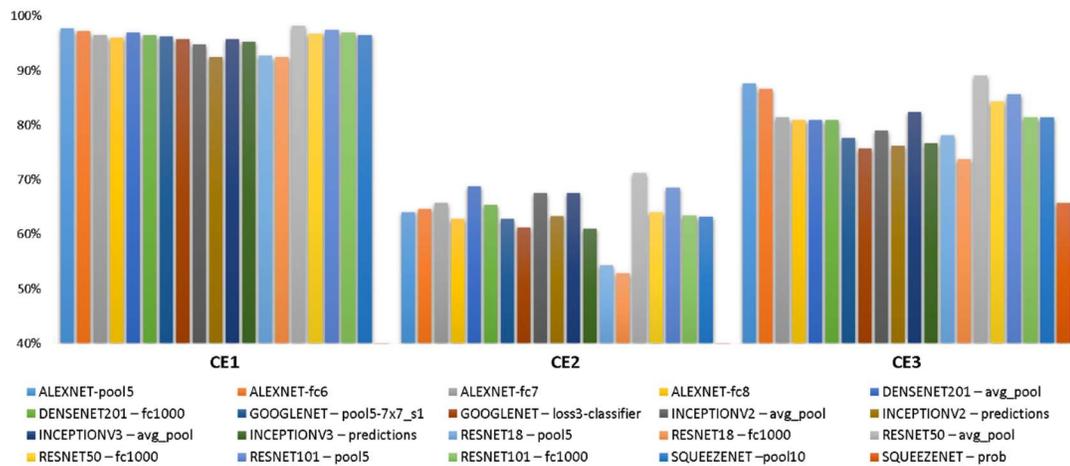


Figura . Exactitud global en el experimento #3

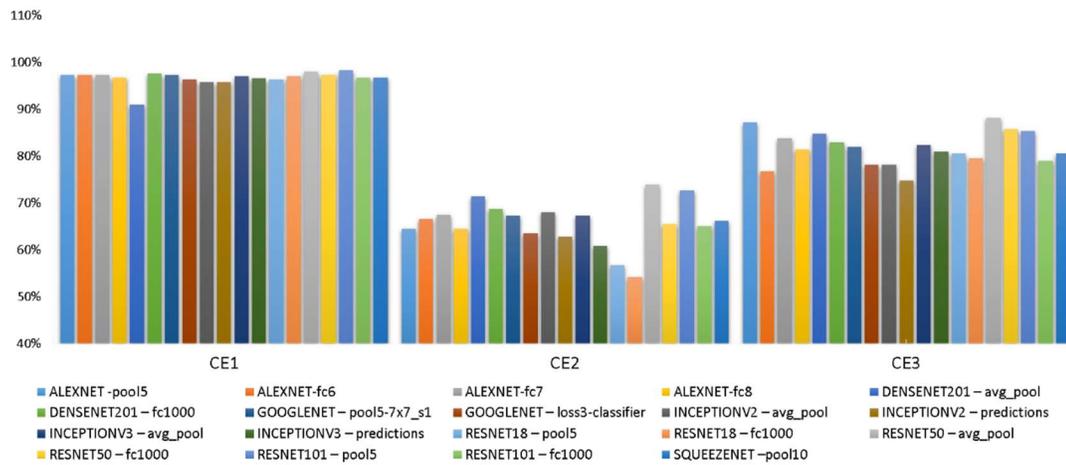


Figura . Exactitud global en el experimento #4

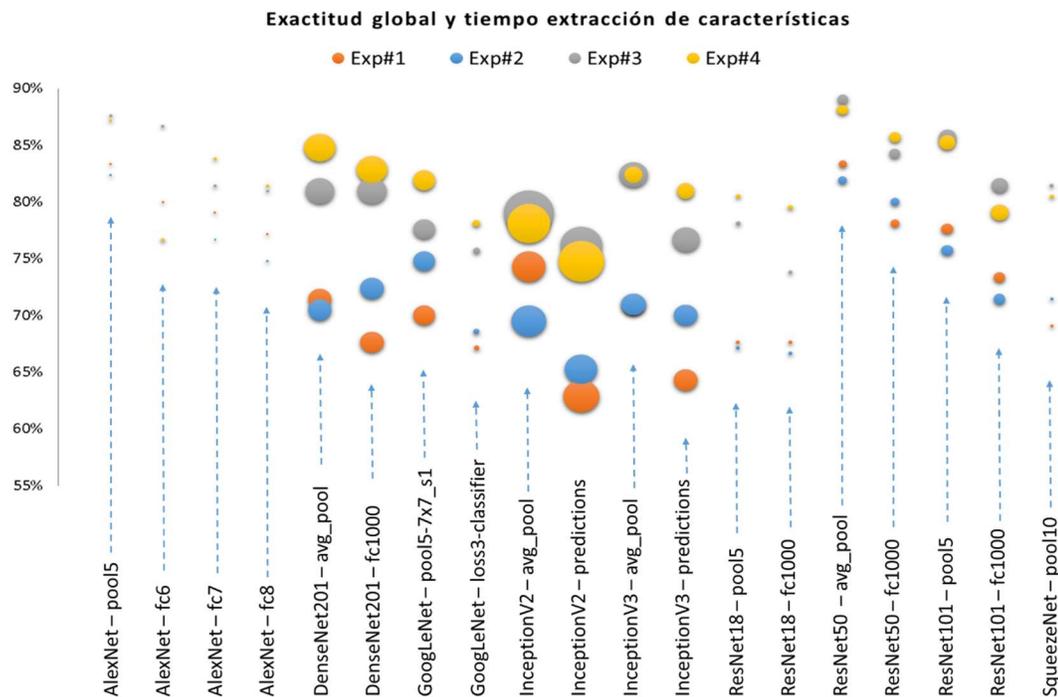


Figura . Comparación exactitud global en experimentos #1 al #4. El eje vertical representa la exactitud global (imágenes clasificadas correctamente / total imágenes). El ancho de los círculos representa el tiempo relativo requerido para extraer los rasgos de cada imagen antes de ser clasificada.

De las matrices de confusión y los gráficos anteriores se extraen las siguientes observaciones:

- Para el conjunto de datos CE1 se lograron, en todos los experimentos, valores muy elevados de aciertos, en general en torno o mayor al 90%. Esto es de esperarse por

ser datos extraídos del conjunto de entrenamiento original. Sin embargo el resultado indica que las imágenes no han sido “memorizadas” completamente por el clasificador incluso para los experimentos #3 y #4, permitiendo suponer que el mismo podrá generalizar correcta.

- Para el conjunto CE2 se observa una exactitud global en torno al 65%. Esto se explica por las grandes variaciones existentes entre este conjunto de prueba (proveniente de la base de datos aumentada) y las imágenes usadas para entrenar al clasificador (no aumentadas). La clasificación de leucocitos y neutrófilos segmentados mostró ser superior a la de eosinófilos y monocitos.
- Para el conjunto CE3 se observan variaciones importantes en la exactitud global lograda yendo desde 62.86% hasta 89.05%. Los máximos y mínimos de cada experimento son mostrados en la Tabla . La mayor exactitud global se obtuvo con la red AlexNet para los experimentos 1 y 2, y con la red ResNet-50 para los experimentos 3 y 4. La red InceptionV2 mostro el menor rendimiento en 3 de los 4 experimentos, junto al mayor costo computacional redes utilizadas.

Tabla

Valores máximos y mínimos de exactitud global en experimentos #1 al #4

Experimento	Mayor exactitud global	Menor exactitud global
# 1	83.33% (AlexNet-pool5)	62.86% (InceptionV2-predictions)
# 2	82.38% (AlexNet-pool5)	65.24% (InceptionV2-predictions)
# 3	89.05% (ResNet-50-avg_pool)	73.18 (ResNet18-fc1000)
# 4	88.1% (ResNet-50-avg_pool)	74.76% (InceptionV2-predictions)

- La mayor exactitud global se ha logrado en todos los experimentos usando las capas más internas de la red neuronal, esto era previsible debido a las marcadas diferencias entre las imágenes usadas en el modelo pre-entrenado y la nueva tarea de clasificación de leucocitos.
- La mayor exactitud se logró con las redes ResNet-50, AlexNet y ResNet-101. Los valores alcanzados por las redes SqueezeNet, ResNet-18 y GoogleNet, aunque inferiores a los del primer grupo de redes, fueron similares entre sí.

- Los tiempos de procesamiento (extracción de los rasgos a partir de las imágenes) con las redes InceptionV2 y DenseNet 201 fueron considerablemente mayores. Los menores tiempos se observaron en las redes AlexNet, SqueezeNet y ResNet-18.
- En general se alcanzó mejor exactitud global en los experimentos #3 y #4. Esto es atribuible al uso de más muestras de neutrófilos segmentados y leucocitos durante el entrenamiento, los cuales constituyen la mayor parte del conjunto de validación. Una revisión más detallada muestra que este aumento en este valor no implica necesariamente un mejor funcionamiento del clasificador, como podría pensarse en primera instancia. En efecto puede corroborarse fácilmente como clasificación acertada de eosinófilos y monocitos resultó severamente afectada en los experimentos #3 y #4. Como ejemplo de ello, en la Tabla se observan los resultados obtenidos al usar la capa “pool5” de la red AlexNet. Se observa que esta variable, por sí sola, no es suficiente para evaluar correctamente el rendimiento del clasificador, especialmente en presencia de conjuntos de entrenamiento y prueba desbalanceados. En este caso particular, el promedio de los verdaderos positivos del clasificador aportó un dato más importante para su análisis.

Tabla

Comparación promedio de verdaderos positivos vs exactitud global en red AlexNet

Experimento	Verdaderos positivos (TP)				Exact. Global	TP. Promedio
	Eosinófilo	Linfocito	Monocito	N.Segmentado		
#1	91%	85%	62%	85%	77.14%	80.75%
#2	91%	83%	67%	83%	82.32%	81.00%
#3	68%	92%	52%	94%	87.62%	76.50%
#4	64%	91%	57%	96%	87.14%	77.00%

Se calculó un conjunto de métricas de uso frecuente en la evaluación de clasificadores, las cuales se presentan a continuación en las tablas 16 a 19. Se incluyen los valores de exactitud, precisión, sensibilidad, especificidad, tasa de falsos positivos y tasa de falsos negativo para cada clase, junto a su valor promedio y la dispersión estándar de cada parámetro. Usando los valores promedios de cada indicador se elaboraron las tablas comparativas 20 a 23. En ellas se ha incluido como indicador adicional el área bajo la

curva de característica operativa del receptor (curva ROC) calculada a partir de los valores puntuales de especificidad y sensibilidad. El color verde señala cuando el indicador alcanza el mejor valor dentro del conjunto y el color rojo representa al peor valor. Analizando los valores presentados en todas estas tablas, se extrae lo siguiente:

- Los clasificadores que mostraron mejor funcionamiento general según los valores de las métricas consideradas fueron aquellos que usaron como vector de rasgos la capa “pool5” de la red AlexNet y la capa “avg_pool” de la red ResNet-50. Este último supera al primero en los experimentos #3 y #4. Ambos exhiben indicadores muy similares en los experimentos #1 y #2.
- El cambio de estrategia del clasificador (OVA u OVO) no implicó diferencias notables en los resultados de los experimentos #1 y #2.
- Para el clasificador basado en la red ResNet-50 el mejor resultado se obtuvo en el experimento #3.
- El uso de un conjunto de datos desbalanceados tuvo un resultado mixto al entrenar los clasificadores. Por un lado algunos indicadores como la exactitud y precisión resultaron favorecidos; otros como especificidad, tasa de falsos negativos y área bajo la curva ROC (AUC-ROC) desmejoraron sus valores, y algunos como la sensibilidad y la tasa de falsos positivos alternaron según el experimento y la red utilizada.

Tabla
Métricas de evaluación para clasificadores del Experimento #1

Red-cap		Exactitud	Precisión	Sensibilidad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error
ALEXNET-pool5	E	0.9286	0.6061	0.9091	0.9309	0.0691	0.0909	0.0714
	L	0.9286	0.8654	0.8491	0.9554	0.0446	0.1509	0.0714
	M	0.9238	0.6190	0.6190	0.9577	0.0423	0.3810	0.0762
	N	0.8857	0.9327	0.8509	0.9271	0.0729	0.1491	0.1143
	Prom	0.9167	0.7558	0.8070	0.9428	0.0572	0.1930	0.0833
	Desv.Std	0.0208	0.1678	0.1284	0.0160	0.0160	0.1284	0.0208
ALEXNET-fc6	E	0.9333	0.6429	0.8182	0.9468	0.0532	0.1818	0.0667
	L	0.9095	0.8269	0.8113	0.9427	0.0573	0.1887	0.0905
	M	0.9048	0.5217	0.5714	0.9418	0.0582	0.4286	0.0952
	N	0.8524	0.8879	0.8333	0.8750	0.1250	0.1667	0.1476
	Prom	0.9000	0.7198	0.7586	0.9266	0.0734	0.2414	0.1000
	Desv.Std	0.0341	0.1682	0.1251	0.0344	0.0344	0.1251	0.0341
ALEXNET-fc7	E	0.9143	0.5667	0.7727	0.9309	0.0691	0.2273	0.0857
	L	0.9143	0.8302	0.8302	0.9427	0.0573	0.1698	0.0857
	M	0.8952	0.4839	0.7143	0.9153	0.0847	0.2857	0.1048
	N	0.8571	0.9375	0.7895	0.9375	0.0625	0.2105	0.1429
	Prom	0.8952	0.7046	0.7767	0.9316	0.0684	0.2233	0.1048
	Desv.Std	0.0269	0.2143	0.0481	0.0119	0.0119	0.0481	0.0269
ALEXNET-fc8	E	0.8714	0.4324	0.7273	0.8883	0.1117	0.2727	0.1286
	L	0.9238	0.8491	0.8491	0.9490	0.0510	0.1509	0.0762
	M	0.9143	0.5484	0.8095	0.9259	0.0741	0.1905	0.0857
	N	0.8333	0.9438	0.7368	0.9479	0.0521	0.2632	0.1667
	Prom	0.8857	0.6934	0.7807	0.9278	0.0722	0.2193	0.1143
	Desv.Std	0.0417	0.2423	0.0585	0.0284	0.0284	0.0585	0.0417
DENSENET201 – avg_pool	E	0.8286	0.3409	0.6818	0.8457	0.1543	0.3182	0.1714
	L	0.9190	0.8103	0.8868	0.9299	0.0701	0.1132	0.0810
	M	0.9000	0.5000	0.6667	0.9259	0.0741	0.3333	0.1000
	N	0.7810	0.9250	0.6491	0.9375	0.0625	0.3509	0.2190
	Prom	0.8571	0.6441	0.7211	0.9098	0.0902	0.2789	0.1429
	Desv.Std	0.0640	0.2703	0.1113	0.0430	0.0430	0.1113	0.0640
DENSENET201 – fc1000	E	0.8000	0.2826	0.5909	0.8245	0.1755	0.4091	0.2000
	L	0.9000	0.7581	0.8868	0.9045	0.0955	0.1132	0.1000
	M	0.9000	0.5000	0.5238	0.9418	0.0582	0.4762	0.1000
	N	0.7524	0.8875	0.6228	0.9063	0.0938	0.3772	0.2476
	Prom	0.8381	0.6070	0.6561	0.8942	0.1058	0.3439	0.1619
	Desv.Std	0.0741	0.2697	0.1592	0.0496	0.0496	0.1592	0.0741
GOOGLENET – pool5-7x7_s1	E	0.8571	0.3947	0.6818	0.8777	0.1223	0.3182	0.1429
	L	0.8667	0.7451	0.7170	0.9172	0.0828	0.2830	0.1333
	M	0.8905	0.4722	0.8095	0.8995	0.1005	0.1905	0.1095
	N	0.7857	0.9059	0.6754	0.9167	0.0833	0.3246	0.2143
	Prom	0.8500	0.6295	0.7209	0.9027	0.0973	0.2791	0.1500
	Desv.Std	0.0451	0.2378	0.0618	0.0186	0.0186	0.0618	0.0451
GOOGLENET – loss3-classifier	E	0.8429	0.3721	0.7273	0.8564	0.1436	0.2727	0.1571
	L	0.8571	0.7447	0.6604	0.9236	0.0764	0.3396	0.1429
	M	0.8619	0.3947	0.7143	0.8783	0.1217	0.2857	0.1381
	N	0.7810	0.9146	0.6579	0.9271	0.0729	0.3421	0.2190
	Prom	0.8357	0.6065	0.6900	0.8963	0.1037	0.3100	0.1643
	Desv.Std	0.0374	0.2670	0.0360	0.0347	0.0347	0.0360	0.0374
INCEPTIONV2 – avg_pool	E	0.8619	0.3939	0.5909	0.8936	0.1064	0.4091	0.1381
	L	0.8952	0.7460	0.8868	0.8981	0.1019	0.1132	0.1048
	M	0.9143	0.5556	0.7143	0.9365	0.0635	0.2857	0.0857
	N	0.8143	0.9310	0.7105	0.9375	0.0625	0.2895	0.1857
	Prom	0.8714	0.6566	0.7256	0.9164	0.0836	0.2744	0.1286
	Desv.Std	0.0438	0.2327	0.1218	0.0238	0.0238	0.1218	0.0438
INCEPTIONV2 – predictions	E	0.8190	0.3000	0.5455	0.8511	0.1489	0.4545	0.1810
	L	0.8524	0.6774	0.7925	0.8726	0.1274	0.2075	0.1476
	M	0.8714	0.4000	0.5714	0.9048	0.0952	0.4286	0.1286
	N	0.7143	0.8462	0.5789	0.8750	0.1250	0.4211	0.2857
	Prom	0.8143	0.5559	0.6221	0.8759	0.1241	0.3779	0.1857
	Desv.Std	0.0701	0.2509	0.1145	0.0221	0.0221	0.1145	0.0701

Tabla 16 – continuación
Métricas de evaluación para clasificadores del Experimento #1

Red-capa		Exactitud	Precisión	Sensibilidad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error
INCEPTIONV3 – avg_pool	E	0.8476	0.3684	0.6364	0.8723	0.1277	0.3636	0.1524
	L	0.8571	0.6885	0.7925	0.8790	0.1210	0.2075	0.1429
	M	0.9238	0.5926	0.7619	0.9418	0.0582	0.2381	0.0762
	N	0.7905	0.9167	0.6754	0.9271	0.0729	0.3246	0.2095
	Prom	0.8548	0.6416	0.7165	0.9051	0.0949	0.2835	0.1452
	Desv.Std	0.0546	0.2272	0.0729	0.0346	0.0346	0.0729	0.0546
INCEPTIONV3 – predictions	E	0.8000	0.2500	0.4545	0.8404	0.1596	0.5455	0.2000
	L	0.8571	0.7347	0.6792	0.9172	0.0828	0.3208	0.1429
	M	0.8667	0.4054	0.7143	0.8836	0.1164	0.2857	0.1333
	N	0.7619	0.8810	0.6491	0.8958	0.1042	0.3509	0.2381
	Prom	0.8214	0.5678	0.6243	0.8843	0.1157	0.3757	0.1786
	Desv.Std	0.0494	0.2906	0.1163	0.0324	0.0324	0.1163	0.0494
RESNET18 – pool5	E	0.8667	0.4118	0.6364	0.8936	0.1064	0.3636	0.1333
	L	0.8190	0.6230	0.7170	0.8535	0.1465	0.2830	0.1810
	M	0.8905	0.4643	0.6190	0.9206	0.0794	0.3810	0.1095
	N	0.7762	0.8851	0.6754	0.8958	0.1042	0.3246	0.2238
	Prom	0.8381	0.5960	0.6620	0.8909	0.1091	0.3380	0.1619
	Desv.Std	0.0508	0.2126	0.0436	0.0278	0.0278	0.0436	0.0508
RESNET18 – fc1000	E	0.8619	0.3939	0.5909	0.8936	0.1064	0.4091	0.1381
	L	0.8429	0.6613	0.7736	0.8662	0.1338	0.2264	0.1571
	M	0.8857	0.4444	0.5714	0.9206	0.0794	0.4286	0.1143
	N	0.7619	0.8636	0.6667	0.8750	0.1250	0.3333	0.2381
	Prom	0.8381	0.5908	0.6506	0.8889	0.1111	0.3494	0.1619
	Desv.Std	0.0537	0.2157	0.0917	0.0241	0.0241	0.0917	0.0537
RESNET50 – avg_pool	E	0.9238	0.6154	0.7273	0.9468	0.0532	0.2727	0.0762
	L	0.9286	0.8167	0.9245	0.9299	0.0701	0.0755	0.0714
	M	0.9333	0.6400	0.7619	0.9524	0.0476	0.2381	0.0667
	N	0.8810	0.9495	0.8246	0.9479	0.0521	0.1754	0.1190
	Prom	0.9167	0.7554	0.8096	0.9443	0.0557	0.1904	0.0833
	Desv.Std	0.0241	0.1574	0.0866	0.0098	0.0098	0.0866	0.0241
RESNET50 – fc1000	E	0.8905	0.4783	0.5000	0.9362	0.0638	0.5000	0.1095
	L	0.9190	0.7813	0.9434	0.9108	0.0892	0.0566	0.0810
	M	0.9143	0.5556	0.7143	0.9365	0.0635	0.2857	0.0857
	N	0.8381	0.9167	0.7719	0.9167	0.0833	0.2281	0.1619
	Prom	0.8905	0.6829	0.7324	0.9250	0.0750	0.2676	0.1095
	Desv.Std	0.0371	0.2020	0.1830	0.0133	0.0133	0.1830	0.0371
RESNET101 – pool5	E	0.9000	0.5172	0.6818	0.9255	0.0745	0.3182	0.1000
	L	0.8952	0.7541	0.8679	0.9045	0.0955	0.1321	0.1048
	M	0.9286	0.6071	0.8095	0.9418	0.0582	0.1905	0.0714
	N	0.8286	0.9239	0.7456	0.9271	0.0729	0.2544	0.1714
	Prom	0.8881	0.7006	0.7762	0.9247	0.0753	0.2238	0.1119
	Desv.Std	0.0423	0.1780	0.0803	0.0154	0.0154	0.0803	0.0423
RESNET101 – fc1000	E	0.8429	0.3590	0.6364	0.8670	0.1330	0.3636	0.1571
	L	0.8952	0.7818	0.8113	0.9236	0.0764	0.1887	0.1048
	M	0.9286	0.6154	0.7619	0.9471	0.0529	0.2381	0.0714
	N	0.8000	0.9000	0.7105	0.9063	0.0938	0.2895	0.2000
	Prom	0.8667	0.6640	0.7300	0.9110	0.0890	0.2700	0.1333
	Desv.Std	0.0567	0.2345	0.0748	0.0337	0.0337	0.0748	0.0567
SQUEEZENET –pool10	E	0.8286	0.3333	0.6364	0.8511	0.1489	0.3636	0.1714
	L	0.8857	0.7544	0.8113	0.9108	0.0892	0.1887	0.1143
	M	0.8952	0.4839	0.7143	0.9153	0.0847	0.2857	0.1048
	N	0.7714	0.9125	0.6404	0.9271	0.0729	0.3596	0.2286
	Prom	0.8452	0.6210	0.7006	0.9011	0.0989	0.2994	0.1548
	Desv.Std	0.0573	0.2610	0.0821	0.0340	0.0340	0.0821	0.0573

Tabla
Métricas de evaluación para clasificadores del Experimento #2

Red-capa		Exactitud	Precisión	Sensibilidad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error	Valor-F
ALEXNET-pool5	E	0.9143	0.5556	0.9091	0.9149	0.0851	0.0909	0.0857	0.6897
	L	0.9333	0.8980	0.8302	0.9682	0.0318	0.1698	0.0667	0.8628
	M	0.9286	0.6364	0.6667	0.9577	0.0423	0.3333	0.0714	0.6512
	N	0.8714	0.9223	0.8333	0.9167	0.0833	0.1667	0.1286	0.8755
	Prom	0.9119	0.7531	0.8098	0.9393	0.0607	0.1902	0.0881	0.7698
	Devv.Std	0.0282	0.1846	0.1022	0.0276	0.0276	0.1022	0.0282	0.1159
ALEXNET-fc6	E	0.8952	0.5000	0.6818	0.9202	0.0798	0.3182	0.1048	0.5769
	L	0.9000	0.8077	0.7925	0.9363	0.0637	0.2075	0.1000	0.8000
	M	0.8905	0.4667	0.6667	0.9153	0.0847	0.3333	0.1095	0.5491
	N	0.8476	0.9184	0.7895	0.9167	0.0833	0.2105	0.1524	0.8491
	Prom	0.8833	0.6732	0.7326	0.9221	0.0779	0.2674	0.1167	0.6938
	Devv.Std	0.0241	0.2242	0.0677	0.0097	0.0097	0.0677	0.0241	0.1528
ALEXNET-fc7	E	0.8714	0.4359	0.7727	0.8830	0.1170	0.2273	0.1286	0.5574
	L	0.9333	0.8824	0.8491	0.9618	0.0382	0.1509	0.0667	0.8654
	M	0.8905	0.4667	0.6667	0.9153	0.0847	0.3333	0.1095	0.5491
	N	0.8381	0.9444	0.7456	0.9479	0.0521	0.2544	0.1619	0.8333
	Prom	0.8833	0.6823	0.7585	0.9270	0.0730	0.2415	0.1167	0.7013
	Devv.Std	0.0397	0.2683	0.0753	0.0352	0.0352	0.0753	0.0397	0.1715
ALEXNET-fc8	E	0.8714	0.4359	0.7727	0.8830	0.1170	0.2273	0.1286	0.5574
	L	0.9000	0.7857	0.8302	0.9236	0.0764	0.1698	0.1000	0.8073
	M	0.9048	0.5172	0.7143	0.9259	0.0741	0.2857	0.0952	0.6000
	N	0.8190	0.9419	0.7105	0.9479	0.0521	0.2895	0.1810	0.8100
	Prom	0.8738	0.6702	0.7569	0.9201	0.0799	0.2431	0.1262	0.6937
	Devv.Std	0.0394	0.2348	0.0565	0.0271	0.0271	0.0565	0.0394	0.1339
DENSENET201 – avg_pool	E	0.8238	0.3256	0.6364	0.8457	0.1543	0.3636	0.1762	0.4308
	L	0.9095	0.7833	0.8868	0.9172	0.0828	0.1132	0.0905	0.8318
	M	0.9095	0.5333	0.7619	0.9259	0.0741	0.2381	0.0905	0.6274
	N	0.7667	0.9221	0.6228	0.9375	0.0625	0.3772	0.2333	0.7435
	Prom	0.8524	0.6411	0.7270	0.9066	0.0934	0.2730	0.1476	0.6584
	Devv.Std	0.0700	0.2648	0.1236	0.0414	0.0414	0.1236	0.0700	0.1733
DENSENET201 – fc1000	E	0.8429	0.3659	0.6818	0.8617	0.1383	0.3182	0.1571	0.4762
	L	0.9190	0.8103	0.8868	0.9299	0.0701	0.1132	0.0810	0.8468
	M	0.9095	0.5333	0.7619	0.9259	0.0741	0.2381	0.0905	0.6274
	N	0.7762	0.9136	0.6491	0.9271	0.0729	0.3509	0.2238	0.7590
	Prom	0.8619	0.6558	0.7449	0.9112	0.0888	0.2551	0.1381	0.6774
	Devv.Std	0.0664	0.2513	0.1058	0.0330	0.0330	0.1058	0.0664	0.1616
GOOGLENET – pool5-7x7_s1	E	0.8667	0.4211	0.7273	0.8830	0.1170	0.2727	0.1333	0.5334
	L	0.9000	0.8200	0.7736	0.9427	0.0573	0.2264	0.1000	0.7961
	M	0.9286	0.6000	0.8571	0.9365	0.0635	0.1429	0.0714	0.7059
	N	0.8000	0.8913	0.7193	0.8958	0.1042	0.2807	0.2000	0.7961
	Prom	0.8738	0.6831	0.7693	0.9145	0.0855	0.2307	0.1262	0.7079
	Devv.Std	0.0553	0.2142	0.0632	0.0296	0.0296	0.0632	0.0553	0.1239
GOOGLENET – loss3-classifier	E	0.8524	0.3902	0.7273	0.8670	0.1330	0.2727	0.1476	0.5079
	L	0.8524	0.7115	0.6981	0.9045	0.0955	0.3019	0.1476	0.7047
	M	0.8905	0.4688	0.7143	0.9101	0.0899	0.2857	0.1095	0.5661
	N	0.7762	0.8941	0.6667	0.9063	0.0938	0.3333	0.2238	0.7638
	Prom	0.8429	0.6162	0.7016	0.8969	0.1031	0.2984	0.1571	0.6356
	Devv.Std	0.0479	0.2303	0.0262	0.0201	0.0201	0.0262	0.0479	0.1188
INCEPTIONV2 – avg_pool	E	0.8571	0.3889	0.6364	0.8830	0.1170	0.3636	0.1429	0.4828
	L	0.8667	0.7193	0.7736	0.8981	0.1019	0.2264	0.1333	0.7455
	M	0.8857	0.4545	0.7143	0.9048	0.0952	0.2857	0.1143	0.5555
	N	0.7810	0.9048	0.6667	0.9167	0.0833	0.3333	0.2190	0.7677
	Prom	0.8476	0.6169	0.6977	0.9006	0.0994	0.3023	0.1524	0.6379
	Devv.Std	0.0460	0.2392	0.0599	0.0141	0.0141	0.0599	0.0460	0.1406
INCEPTIONV2 – predictions	E	0.8238	0.3171	0.5909	0.8511	0.1489	0.4091	0.1762	0.4127
	L	0.8667	0.7193	0.7736	0.8981	0.1019	0.2264	0.1333	0.7455
	M	0.8667	0.3793	0.5238	0.9048	0.0952	0.4762	0.1333	0.4400
	N	0.7476	0.8675	0.6316	0.8854	0.1146	0.3684	0.2524	0.7310
	Prom	0.8262	0.5708	0.6300	0.8848	0.1152	0.3700	0.1738	0.5823
	Devv.Std	0.0561	0.2653	0.1056	0.0239	0.0239	0.1056	0.0561	0.1805

Tabla 17 – continuación
Métricas de evaluación para clasificadores del Experimento #2

Red-capa		Exactitud	Precisión	Sensibilidad	Especificidad	Tasa a Falsos Positivos	Tasa a Falsos Negativos	Error	Valor-F
INCEPTIONV3 – avg_pool	E	0.8619	0.4054	0.6818	0.8830	0.1170	0.3182	0.1381	0.5085
	L	0.8571	0.6885	0.7925	0.8790	0.1210	0.2075	0.1429	0.7368
	M	0.9048	0.5172	0.7143	0.9259	0.0741	0.2857	0.0952	0.6000
	N	0.7857	0.9157	0.6667	0.9271	0.0729	0.3333	0.2143	0.7716
	Prom	0.8524	0.6317	0.7138	0.9037	0.0963	0.2862	0.1476	0.6542
	Devv.Std	0.0493	0.2222	0.0561	0.0263	0.0263	0.0561	0.0493	0.1222
INCEPTIONV3 – predictions	E	0.8952	0.5000	0.6364	0.9255	0.0745	0.3636	0.1048	0.5600
	L	0.8429	0.6667	0.7547	0.8726	0.1274	0.2453	0.1571	0.7080
	M	0.8762	0.4242	0.6667	0.8995	0.1005	0.3333	0.1238	0.5185
	N	0.7857	0.8876	0.6930	0.8958	0.1042	0.3070	0.2143	0.7783
	Prom	0.8500	0.6196	0.6877	0.8984	0.1016	0.3123	0.1500	0.6412
	Devv.Std	0.0480	0.2054	0.0503	0.0217	0.0217	0.0503	0.0480	0.1223
RESNET18 – pool5	E	0.8333	0.3415	0.6364	0.8564	0.1436	0.3636	0.1667	0.4445
	L	0.8381	0.6727	0.6981	0.8854	0.1146	0.3019	0.1619	0.6852
	M	0.9048	0.5161	0.7619	0.9206	0.0794	0.2381	0.0952	0.6154
	N	0.7667	0.8916	0.6491	0.9063	0.0938	0.3509	0.2333	0.7513
	Prom	0.8357	0.6055	0.6864	0.8922	0.1078	0.3136	0.1643	0.6241
	Devv.Std	0.0564	0.2338	0.0570	0.0279	0.0279	0.0570	0.0564	0.1320
RESNET18 – fc1000	E	0.8524	0.3846	0.6818	0.8723	0.1277	0.3182	0.1476	0.4918
	L	0.8238	0.6290	0.7358	0.8535	0.1465	0.2642	0.1762	0.6782
	M	0.8857	0.4516	0.6667	0.9101	0.0899	0.3333	0.1143	0.5385
	N	0.7714	0.9231	0.6316	0.9375	0.0625	0.3684	0.2286	0.7500
	Prom	0.8333	0.5971	0.6790	0.8933	0.1067	0.3210	0.1667	0.6146
	Devv.Std	0.0484	0.2406	0.0434	0.0377	0.0377	0.0434	0.0484	0.1201
RESNET50 – avg_pool	E	0.9048	0.5385	0.6364	0.9362	0.0638	0.3636	0.0952	0.5834
	L	0.9143	0.7692	0.9434	0.9045	0.0955	0.0566	0.0857	0.8474
	M	0.9429	0.6552	0.9048	0.9471	0.0529	0.0952	0.0571	0.7600
	N	0.8762	0.9889	0.7807	0.9896	0.0104	0.2193	0.1238	0.8726
	Prom	0.9095	0.7379	0.8163	0.9443	0.0557	0.1837	0.0905	0.7658
	Devv.Std	0.0275	0.1920	0.1386	0.0352	0.0352	0.1386	0.0275	0.1309
RESNET50 – fc1000	E	0.9048	0.5357	0.6818	0.9309	0.0691	0.3182	0.0952	0.6000
	L	0.9143	0.7692	0.9434	0.9045	0.0955	0.0566	0.0857	0.8474
	M	0.9286	0.5938	0.9048	0.9312	0.0688	0.0952	0.0714	0.7170
	N	0.8524	0.9882	0.7368	0.9896	0.0104	0.2632	0.1476	0.8442
	Prom	0.9000	0.7217	0.8167	0.9390	0.0610	0.1833	0.1000	0.7522
	Devv.Std	0.0332	0.2035	0.1270	0.0360	0.0360	0.1270	0.0332	0.1182
RESNET101 – pool5	E	0.8714	0.4242	0.6364	0.8989	0.1011	0.3636	0.1286	0.5091
	L	0.9048	0.7705	0.8868	0.9108	0.0892	0.1132	0.0952	0.8246
	M	0.9286	0.6000	0.8571	0.9365	0.0635	0.1429	0.0714	0.7059
	N	0.8095	0.9302	0.7018	0.9375	0.0625	0.2982	0.1905	0.8000
	Prom	0.8786	0.6812	0.7705	0.9209	0.0791	0.2295	0.1214	0.7099
	Devv.Std	0.0517	0.2180	0.1208	0.0192	0.0192	0.1208	0.0517	0.1433
RESNET101 – fc1000	E	0.8524	0.3714	0.5909	0.8830	0.1170	0.4091	0.1476	0.4561
	L	0.8810	0.7414	0.8113	0.9045	0.0955	0.1887	0.1190	0.7748
	M	0.9238	0.5806	0.8571	0.9312	0.0688	0.1429	0.0762	0.6923
	N	0.7714	0.8837	0.6667	0.8958	0.1042	0.3333	0.2286	0.7600
	Prom	0.8571	0.6443	0.7315	0.9036	0.0964	0.2685	0.1429	0.6708
	Devv.Std	0.0642	0.2200	0.1240	0.0204	0.0204	0.1240	0.0642	0.1476
SQUEEZENET –pool10	E	0.8238	0.3333	0.6818	0.8404	0.1596	0.3182	0.1762	0.4477
	L	0.9238	0.8136	0.9057	0.9299	0.0701	0.0943	0.0762	0.8572
	M	0.9000	0.5000	0.8095	0.9101	0.0899	0.1905	0.1000	0.6182
	N	0.7810	0.9722	0.6140	0.9792	0.0208	0.3860	0.2190	0.7527
	Prom	0.8571	0.6548	0.7528	0.9149	0.0851	0.2472	0.1429	0.6689
	Devv.Std	0.0663	0.2906	0.1302	0.0575	0.0575	0.1302	0.0663	0.1770

Tabla
Métricas de evaluación para clasificadores del Experimento #3

Red-capa		Exactitud	Precisión	Sensibilidad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error
ALEXNET-pool5	E	0.9571	0.9333	0.6364	0.9947	0.0053	0.3636	0.0429
	L	0.9476	0.8889	0.9057	0.9618	0.0382	0.0943	0.0524
	M	0.9476	0.8571	0.5714	0.9894	0.0106	0.4286	0.0524
	N	0.9000	0.8661	0.9649	0.8229	0.1771	0.0351	0.1000
	Prom	0.9381	0.8864	0.7696	0.9422	0.0578	0.2304	0.0619
	Desv.Std	0.0258	0.0340	0.1947	0.0808	0.0808	0.1947	0.0258
ALEXNET-fc6	E	0.9571	0.8824	0.6818	0.9894	0.0106	0.3182	0.0429
	L	0.9476	0.8750	0.9245	0.9554	0.0446	0.0755	0.0524
	M	0.9381	0.7857	0.5238	0.9841	0.0159	0.4762	0.0619
	N	0.8905	0.8699	0.9386	0.8333	0.1667	0.0614	0.1095
	Prom	0.9333	0.8532	0.7672	0.9406	0.0594	0.2328	0.0667
	Desv.Std	0.0296	0.0453	0.2005	0.0730	0.0730	0.2005	0.0296
ALEXNET-fc7	E	0.9333	0.7000	0.6364	0.9681	0.0319	0.3636	0.0667
	L	0.9048	0.7895	0.8491	0.9236	0.0764	0.1509	0.0952
	M	0.9095	0.6000	0.2857	0.9788	0.0212	0.7143	0.0905
	N	0.8810	0.8618	0.9298	0.8229	0.1771	0.0702	0.1190
	Prom	0.9071	0.7378	0.6752	0.9234	0.0766	0.3248	0.0929
	Desv.Std	0.0215	0.1132	0.2877	0.0711	0.0711	0.2877	0.0215
ALEXNET-fc8	E	0.9381	0.7368	0.6364	0.9734	0.0266	0.3636	0.0619
	L	0.9095	0.7742	0.9057	0.9108	0.0892	0.0943	0.0905
	M	0.9143	0.6000	0.4286	0.9683	0.0317	0.5714	0.0857
	N	0.8571	0.8684	0.8684	0.8438	0.1563	0.1316	0.1429
	Prom	0.9048	0.7449	0.7098	0.9241	0.0759	0.2902	0.0952
	Desv.Std	0.0341	0.1113	0.2221	0.0606	0.0606	0.2221	0.0341
DENSENET201 – avg_pool	E	0.9190	0.6316	0.5455	0.9628	0.0372	0.4545	0.0810
	L	0.9333	0.8421	0.9057	0.9427	0.0573	0.0943	0.0667
	M	0.9095	0.5556	0.4762	0.9577	0.0423	0.5238	0.0905
	N	0.8571	0.8621	0.8772	0.8333	0.1667	0.1228	0.1429
	Prom	0.9048	0.7228	0.7011	0.9241	0.0759	0.2989	0.0952
	Desv.Std	0.0332	0.1527	0.2219	0.0611	0.0611	0.2219	0.0332
DENSENET201 – fc1000	E	0.9143	0.6000	0.5455	0.9574	0.0426	0.4545	0.0857
	L	0.9333	0.8197	0.9434	0.9299	0.0701	0.0566	0.0667
	M	0.9143	0.5789	0.5238	0.9577	0.0423	0.4762	0.0857
	N	0.8571	0.8818	0.8509	0.8646	0.1354	0.1491	0.1429
	Prom	0.9048	0.7201	0.7159	0.9274	0.0726	0.2841	0.0952
	Desv.Std	0.0330	0.1532	0.2129	0.0439	0.0439	0.2129	0.0330
GOOGLENET – pool5-7x7_s1	E	0.8810	0.4348	0.4545	0.9309	0.0691	0.5455	0.1190
	L	0.9095	0.8148	0.8302	0.9363	0.0637	0.1698	0.0905
	M	0.9429	0.8000	0.5714	0.9841	0.0159	0.4286	0.0571
	N	0.8190	0.8220	0.8509	0.7813	0.2188	0.1491	0.1810
	Prom	0.8881	0.7179	0.6768	0.9081	0.0919	0.3232	0.1119
	Desv.Std	0.0525	0.1890	0.1952	0.0879	0.0879	0.1952	0.0525
GOOGLENET – loss3-classifier	E	0.8952	0.5000	0.5000	0.9415	0.0585	0.5000	0.1048
	L	0.9000	0.7963	0.8113	0.9299	0.0701	0.1887	0.1000
	M	0.9190	0.6111	0.5238	0.9630	0.0370	0.4762	0.0810
	N	0.8000	0.8103	0.8246	0.7708	0.2292	0.1754	0.2000
	Prom	0.8786	0.6794	0.6649	0.9013	0.0987	0.3351	0.1214
	Desv.Std	0.0534	0.1502	0.1770	0.0881	0.0881	0.1770	0.0534
INCEPTIONV2 – avg_pool	E	0.9000	0.5333	0.3636	0.9628	0.0372	0.6364	0.1000
	L	0.8905	0.7344	0.8868	0.8917	0.1083	0.1132	0.1095
	M	0.9286	0.6875	0.5238	0.9735	0.0265	0.4762	0.0714
	N	0.8619	0.8696	0.8772	0.8438	0.1563	0.1228	0.1381
	Prom	0.8952	0.7062	0.6629	0.9179	0.0821	0.3371	0.1048
	Desv.Std	0.0275	0.1387	0.2614	0.0614	0.0614	0.2614	0.0275
INCEPTIONV2 – predictions	E	0.8952	0.5000	0.3182	0.9628	0.0372	0.6818	0.1048
	L	0.8905	0.7419	0.8679	0.8981	0.1019	0.1321	0.1095
	M	0.9095	0.5625	0.4286	0.9630	0.0370	0.5714	0.0905
	N	0.8286	0.8305	0.8596	0.7917	0.2083	0.1404	0.1714
	Prom	0.8810	0.6587	0.6186	0.9039	0.0961	0.3814	0.1190
	Desv.Std	0.0358	0.1537	0.2867	0.0808	0.0808	0.2867	0.0358

Tabla 18 – continuación
Métricas de evaluación para clasificadores del Experimento #3

Red-capa		Exactitud	Precisión	Sensibilidad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error
INCEPTIONV3 – avg_pool	E	0.9333	0.7222	0.5909	0.9734	0.0266	0.4091	0.0667
	L	0.8857	0.7231	0.8868	0.8854	0.1146	0.1132	0.1143
	M	0.9381	0.8333	0.4762	0.9894	0.0106	0.5238	0.0619
	N	0.8905	0.8957	0.9035	0.8750	0.1250	0.0965	0.1095
	Prom	0.9119	0.7936	0.7144	0.9308	0.0692	0.2856	0.0881
	Desv.Std	0.0276	0.0858	0.2141	0.0590	0.0590	0.2141	0.0276
INCEPTIONV3 – predictions	E	0.9048	0.5625	0.4091	0.9628	0.0372	0.5909	0.0952
	L	0.8714	0.7097	0.8302	0.8854	0.1146	0.1698	0.1286
	M	0.9048	0.5263	0.4762	0.9524	0.0476	0.5238	0.0952
	N	0.8524	0.6673	0.8596	0.8438	0.1563	0.1404	0.1476
	Prom	0.8833	0.6664	0.6438	0.9111	0.0889	0.3562	0.1167
	Desv.Std	0.0259	0.1556	0.2342	0.0565	0.0565	0.2342	0.0259
RESNET18 – pool5	E	0.9286	0.7692	0.4545	0.9840	0.0160	0.5455	0.0714
	L	0.8571	0.6949	0.7736	0.8854	0.1146	0.2264	0.1429
	M	0.9333	0.6667	0.6667	0.9630	0.0370	0.3333	0.0667
	N	0.8429	0.8462	0.8684	0.8125	0.1875	0.1316	0.1571
	Prom	0.8905	0.7442	0.6908	0.9112	0.0888	0.3092	0.1095
	Desv.Std	0.0471	0.0805	0.1778	0.0783	0.0783	0.1778	0.0471
RESNET18 – fc1000	E	0.9143	0.6111	0.5000	0.9628	0.0372	0.5000	0.0857
	L	0.8333	0.6552	0.7170	0.8726	0.1274	0.2830	0.1667
	M	0.8952	0.4737	0.4286	0.9471	0.0529	0.5714	0.1048
	N	0.8333	0.8435	0.8509	0.8125	0.1875	0.1491	0.1667
	Prom	0.8690	0.6459	0.6241	0.8987	0.1013	0.3759	0.1310
	Desv.Std	0.0420	0.1527	0.1947	0.0697	0.0697	0.1947	0.0420
RESNET50 – avg_pool	E	0.9762	1.0000	0.7727	1.0000	0.0000	0.2273	0.0238
	L	0.9381	0.8448	0.9245	0.9427	0.0573	0.0755	0.0619
	M	0.9476	0.8125	0.6190	0.9841	0.0159	0.3810	0.0524
	N	0.9190	0.9076	0.9474	0.8854	0.1146	0.0526	0.0810
	Prom	0.9452	0.8912	0.8159	0.9531	0.0469	0.1841	0.0548
	Desv.Std	0.0238	0.0826	0.1524	0.0512	0.0512	0.1524	0.0238
RESNET50 – fc1000	E	0.9429	0.8125	0.5909	0.9840	0.0160	0.4091	0.0571
	L	0.9476	0.8500	0.9623	0.9427	0.0573	0.0377	0.0524
	M	0.9238	0.6667	0.4762	0.9735	0.0265	0.5238	0.0762
	N	0.8714	0.8655	0.9035	0.8333	0.1667	0.0965	0.1286
	Prom	0.9214	0.7987	0.7332	0.9334	0.0666	0.2668	0.0786
	Desv.Std	0.0349	0.0908	0.2365	0.0690	0.0690	0.2365	0.0349
RESNET101 – pool5	E	0.9476	0.8667	0.5909	0.9894	0.0106	0.4091	0.0524
	L	0.9286	0.7969	0.9623	0.9172	0.0828	0.0377	0.0714
	M	0.9429	0.7647	0.6190	0.9788	0.0212	0.3810	0.0571
	N	0.8952	0.9035	0.9035	0.8854	0.1146	0.0965	0.1048
	Prom	0.9286	0.8329	0.7689	0.9427	0.0573	0.2311	0.0714
	Desv.Std	0.0237	0.0634	0.1912	0.0497	0.0497	0.1912	0.0237
RESNET101 – fc1000	E	0.9143	0.6111	0.5000	0.9628	0.0372	0.5000	0.0857
	L	0.8905	0.7419	0.8679	0.8981	0.1019	0.1321	0.1095
	M	0.9476	0.7778	0.6667	0.9788	0.0212	0.3333	0.0524
	N	0.8762	0.8929	0.8772	0.8750	0.1250	0.1228	0.1238
	Prom	0.9071	0.7559	0.7279	0.9287	0.0713	0.2721	0.0929
	Desv.Std	0.0312	0.1160	0.1804	0.0500	0.0500	0.1804	0.0312
SQUEEZENET –pool10	E	0.9190	0.6471	0.5000	0.9681	0.0319	0.5000	0.0810
	L	0.9190	0.8103	0.8868	0.9299	0.0701	0.1132	0.0810
	M	0.9333	0.6842	0.6190	0.9683	0.0317	0.3810	0.0667
	N	0.8571	0.8621	0.8772	0.8333	0.1667	0.1228	0.1429
	Prom	0.9071	0.7509	0.7208	0.9249	0.0751	0.2792	0.0929
	Desv.Std	0.0340	0.1019	0.1925	0.0637	0.0637	0.1925	0.0340

Tabla
Métricas de evaluación para clasificadores del Experimento #4

Red-capa		Exactitud	Precisión	Sensibilidad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error
ALEXNET-pool5	E	0.9524	0.8750	0.6364	0.9894	0.0106	0.3636	0.0476
	L	0.9429	0.8727	0.9057	0.9554	0.0446	0.0943	0.0571
	M	0.9476	0.8571	0.5714	0.9894	0.0106	0.4286	0.0524
	N	0.9000	0.8720	0.9561	0.8333	0.1667	0.0439	0.1000
	Prom	0.9357	0.8692	0.7674	0.9419	0.0581	0.2326	0.0643
	Desv.Std	0.0241	0.0082	0.1918	0.0741	0.0741	0.1918	0.0241
ALEXNET-fc6	E	0.9667	0.9412	0.7273	0.9947	0.0053	0.2727	0.0333
	L	0.9238	0.8136	0.9057	0.9299	0.0701	0.0943	0.0762
	M	0.9429	0.8000	0.5714	0.9841	0.0159	0.4286	0.0571
	N	0.9000	0.8908	0.9298	0.8646	0.1354	0.0702	0.1000
	Prom	0.9333	0.8614	0.7835	0.9433	0.0567	0.2165	0.0667
	Desv.Std	0.0283	0.0665	0.1678	0.0597	0.0597	0.1678	0.0283
ALEXNET-fc7	E	0.9476	0.7619	0.7273	0.9734	0.0266	0.2727	0.0524
	L	0.9238	0.8033	0.9245	0.9236	0.0764	0.0755	0.0762
	M	0.9190	0.6667	0.3810	0.9788	0.0212	0.6190	0.0810
	N	0.8857	0.8879	0.9035	0.8646	0.1354	0.0965	0.1143
	Prom	0.9190	0.7799	0.7341	0.9351	0.0649	0.2659	0.0810
	Desv.Std	0.0255	0.0919	0.2515	0.0532	0.0532	0.2515	0.0255
ALEXNET-fc8	E	0.9238	0.6364	0.6364	0.9574	0.0426	0.3636	0.0762
	L	0.9143	0.7966	0.8868	0.9236	0.0764	0.1132	0.0857
	M	0.9095	0.5625	0.4286	0.9630	0.0370	0.5714	0.0905
	N	0.8810	0.8938	0.8860	0.8750	0.1250	0.1140	0.1190
	Prom	0.9071	0.7223	0.7094	0.9297	0.0703	0.2906	0.0929
	Desv.Std	0.0184	0.1504	0.2212	0.0404	0.0404	0.2212	0.0184
DENSENET201 – avg_pool	E	0.9333	0.7222	0.5909	0.9734	0.0266	0.4091	0.0667
	L	0.9381	0.8448	0.9245	0.9427	0.0573	0.0755	0.0619
	M	0.9333	0.6842	0.6190	0.9683	0.0317	0.3810	0.0667
	N	0.8905	0.8957	0.9035	0.8750	0.1250	0.0965	0.1095
	Prom	0.9238	0.7867	0.7595	0.9398	0.0602	0.2405	0.0762
	Desv.Std	0.0223	0.0999	0.1790	0.0453	0.0453	0.1790	0.0223
DENSENET201 – fc1000	E	0.9286	0.6842	0.5909	0.9681	0.0319	0.4091	0.0714
	L	0.9333	0.8545	0.8868	0.9490	0.0510	0.1132	0.0667
	M	0.9143	0.5652	0.6190	0.9471	0.0529	0.3810	0.0857
	N	0.8810	0.8938	0.8860	0.8750	0.1250	0.1140	0.1190
	Prom	0.9143	0.7494	0.7457	0.9348	0.0652	0.2543	0.0857
	Desv.Std	0.0237	0.1528	0.1629	0.0410	0.0410	0.1629	0.0237
GOOGLENET – pool5-7x7_s1	E	0.9143	0.6111	0.5000	0.9628	0.0372	0.5000	0.0857
	L	0.9143	0.8431	0.8113	0.9490	0.0510	0.1887	0.0857
	M	0.9667	0.8500	0.8095	0.9841	0.0159	0.1905	0.0333
	N	0.8429	0.8347	0.8860	0.7917	0.2083	0.1140	0.1571
	Prom	0.9095	0.7847	0.7517	0.9219	0.0781	0.2483	0.0905
	Desv.Std	0.0508	0.1159	0.1715	0.0880	0.0880	0.1715	0.0508
GOOGLENET – loss3-classifier	E	0.9095	0.5714	0.5455	0.9521	0.0479	0.4545	0.0905
	L	0.9000	0.8077	0.7925	0.9363	0.0637	0.2075	0.1000
	M	0.9286	0.6875	0.5238	0.9735	0.0265	0.4762	0.0714
	N	0.8238	0.8182	0.8684	0.7708	0.2292	0.1316	0.1762
	Prom	0.8905	0.7212	0.6825	0.9082	0.0918	0.3175	0.1095
	Desv.Std	0.0460	0.1161	0.1738	0.0928	0.0928	0.1738	0.0460
INCEPTIONV2 – avg_pool	E	0.8952	0.5000	0.5000	0.9415	0.0585	0.5000	0.1048
	L	0.9048	0.7895	0.8491	0.9236	0.0764	0.1509	0.0952
	M	0.9333	0.7692	0.4762	0.9841	0.0159	0.5238	0.0667
	N	0.8286	0.8305	0.8596	0.7917	0.2083	0.1404	0.1714
	Prom	0.8905	0.7223	0.6712	0.9102	0.0898	0.3288	0.1095
	Desv.Std	0.0443	0.1504	0.2117	0.0830	0.0830	0.2117	0.0443
INCEPTIONV2 – predictions	E	0.8857	0.4500	0.4091	0.9415	0.0585	0.5909	0.1143
	L	0.8952	0.7818	0.8113	0.9236	0.0764	0.1887	0.1048
	M	0.9333	0.6842	0.6190	0.9683	0.0317	0.3810	0.0667
	N	0.7810	0.7931	0.8070	0.7500	0.2500	0.1930	0.2190
	Prom	0.8738	0.6773	0.6616	0.8958	0.1042	0.3384	0.1262
	Desv.Std	0.0652	0.1592	0.1907	0.0989	0.0989	0.1907	0.0652

Tabla 19 – continuación
Métricas de evaluación para clasificadores del Experimento #4

Red-capa		Exactitud	Precisión	Sensibilidad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error
INCEPTIONV3 – avg_pool	E	0.9333	0.7500	0.5455	0.9787	0.0213	0.4545	0.0667
	L	0.8905	0.7586	0.8302	0.9108	0.0892	0.1698	0.1095
	M	0.9381	0.7000	0.6667	0.9683	0.0317	0.3333	0.0619
	N	0.8857	0.8879	0.9035	0.8646	0.1354	0.0965	0.1143
	Prom	0.9119	0.7741	0.7365	0.9306	0.0694	0.2635	0.0881
	Desv.Std	0.0276	0.0801	0.1613	0.0532	0.0532	0.1613	0.0276
INCEPTIONV3 – predictions	E	0.9190	0.6316	0.5455	0.9628	0.0372	0.4545	0.0810
	L	0.8857	0.7458	0.8302	0.9045	0.0955	0.1698	0.1143
	M	0.9286	0.6667	0.5714	0.9683	0.0317	0.4286	0.0714
	N	0.8857	0.8947	0.8947	0.8750	0.1250	0.1053	0.1143
	Prom	0.9048	0.7347	0.7105	0.9276	0.0724	0.2895	0.0952
	Desv.Std	0.0223	0.1169	0.1778	0.0454	0.0454	0.1778	0.0223
RESNET18 – pool5	E	0.9190	0.6316	0.5455	0.9628	0.0372	0.4545	0.0810
	L	0.8952	0.7818	0.8113	0.9236	0.0764	0.1887	0.1048
	M	0.9381	0.7000	0.6667	0.9683	0.0317	0.3333	0.0619
	N	0.8571	0.8621	0.8772	0.8333	0.1667	0.1228	0.1429
	Prom	0.9024	0.7439	0.7252	0.9220	0.0780	0.2748	0.0976
	Desv.Std	0.0349	0.0999	0.1486	0.0624	0.0624	0.1486	0.0349
RESNET18 – fc1000	E	0.9190	0.6316	0.5455	0.9628	0.0372	0.4545	0.0810
	L	0.8952	0.7719	0.8302	0.9172	0.0828	0.1698	0.1048
	M	0.9190	0.5833	0.6667	0.9471	0.0529	0.3333	0.0810
	N	0.8571	0.8818	0.8509	0.8646	0.1354	0.1491	0.1429
	Prom	0.8976	0.7172	0.7233	0.9229	0.0771	0.2767	0.1024
	Desv.Std	0.0292	0.1358	0.1444	0.0432	0.0432	0.1444	0.0292
RESNET50 – avg_pool	E	0.9476	0.8667	0.5909	0.9894	0.0106	0.4091	0.0524
	L	0.9524	0.8909	0.9245	0.9618	0.0382	0.0755	0.0476
	M	0.9571	0.8750	0.6667	0.9894	0.0106	0.3333	0.0429
	N	0.9048	0.8790	0.9561	0.8438	0.1563	0.0439	0.0952
	Prom	0.9405	0.8779	0.7846	0.9461	0.0539	0.2154	0.0595
	Desv.Std	0.0241	0.0101	0.1830	0.0694	0.0694	0.1830	0.0241
RESNET50 – fc1000	E	0.9381	0.8000	0.5455	0.9840	0.0160	0.4545	0.0619
	L	0.9333	0.8545	0.8868	0.9490	0.0510	0.1132	0.0667
	M	0.9476	0.7500	0.7143	0.9735	0.0265	0.2857	0.0524
	N	0.8952	0.8833	0.9298	0.8542	0.1458	0.0702	0.1048
	Prom	0.9286	0.8220	0.7691	0.9402	0.0598	0.2309	0.0714
	Desv.Std	0.0230	0.0591	0.1758	0.0592	0.0592	0.1758	0.0230
RESNET101 – pool5	E	0.9286	0.6842	0.5909	0.9681	0.0319	0.4091	0.0714
	L	0.9333	0.8197	0.9434	0.9299	0.0701	0.0566	0.0667
	M	0.9619	0.8421	0.7619	0.9841	0.0159	0.2381	0.0381
	N	0.8810	0.9009	0.8772	0.8854	0.1146	0.1228	0.1190
	Prom	0.9262	0.8117	0.7934	0.9419	0.0581	0.2066	0.0738
	Desv.Std	0.0336	0.0916	0.1544	0.0440	0.0440	0.1544	0.0336
RESNET101 – fc1000	E	0.9000	0.5294	0.4091	0.9574	0.0426	0.5909	0.1000
	L	0.9143	0.8070	0.8679	0.9299	0.0701	0.1321	0.0857
	M	0.9286	0.6250	0.7143	0.9524	0.0476	0.2857	0.0714
	N	0.8381	0.8571	0.8421	0.8333	0.1667	0.1579	0.1619
	Prom	0.8952	0.7046	0.7084	0.9183	0.0817	0.2916	0.1048
	Desv.Std	0.0398	0.1536	0.2105	0.0579	0.0579	0.2105	0.0398
SQUEEZENET –pool10	E	0.9048	0.5500	0.5000	0.9521	0.0479	0.5000	0.0952
	L	0.9429	0.8596	0.9245	0.9490	0.0510	0.0755	0.0571
	M	0.9095	0.5417	0.6190	0.9418	0.0582	0.3810	0.0905
	N	0.8524	0.8807	0.8421	0.8646	0.1354	0.1579	0.1476
	Prom	0.9024	0.7080	0.7214	0.9269	0.0731	0.2786	0.0976
	Desv.Std	0.0374	0.1875	0.1961	0.0418	0.0418	0.1961	0.0374

Tabla
Comparación de indicadores del Experimento #1

	Exactitud	Precisión	Sensitividad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error	AUC-ROC	TPP-Prom	G-Mean
ALEXNET-pool5	91.67%	75.58%	80.70%	94.28%	5.72%	19.30%	8.33%	87.49%	80.70%	87.23%
ALEXNET-fc6	90.00%	71.98%	75.86%	92.66%	7.34%	24.14%	10.00%	84.26%	75.86%	83.84%
ALEXNET-fc7	89.52%	70.46%	77.67%	93.16%	6.84%	22.33%	10.48%	85.42%	77.67%	85.06%
ALEXNET-fc8	88.57%	69.34%	78.07%	92.78%	7.22%	21.93%	11.43%	85.43%	78.07%	85.11%
DENSENET201 – avg_pool	85.71%	64.41%	72.11%	90.98%	9.02%	27.89%	14.29%	81.55%	65.73%	81.00%
DENSENET201 – fc1000	83.81%	60.70%	65.61%	89.42%	10.58%	34.39%	16.19%	77.52%	65.61%	76.60%
GOOGLENET – pool5-7x7_s1	85.00%	62.95%	72.09%	90.27%	9.73%	27.91%	15.00%	81.18%	72.09%	80.67%
GOOGLENET – loss3-classifier	83.57%	60.65%	69.00%	89.63%	10.37%	31.00%	16.43%	79.32%	69.00%	78.64%
INCEPTIONV2 – avg_pool	87.14%	65.66%	72.56%	91.64%	8.36%	27.44%	12.86%	82.10%	72.56%	81.54%
INCEPTIONV2 – predictions	81.43%	55.59%	62.21%	87.59%	12.41%	37.79%	18.57%	74.90%	62.21%	73.82%
INCEPTIONV3 – avg_pool	85.48%	64.16%	71.65%	90.51%	9.49%	28.35%	14.52%	81.08%	71.66%	80.53%
INCEPTIONV3 – predictions	82.14%	56.78%	62.43%	88.43%	11.57%	37.57%	17.86%	75.43%	62.43%	74.30%
RESNET18 – pool5	83.81%	59.60%	66.20%	89.09%	10.91%	33.80%	16.19%	77.65%	66.20%	76.80%
RESNET18 – fc1000	83.81%	59.08%	65.06%	88.89%	11.11%	34.94%	16.19%	76.98%	65.07%	76.05%
RESNET50 – avg_pool	91.67%	75.54%	80.96%	94.43%	5.57%	19.04%	8.33%	87.70%	80.96%	87.44%
RESNET50 – fc1000	89.05%	68.29%	73.24%	92.50%	7.50%	26.76%	10.95%	82.87%	73.24%	82.31%
RESNET101 – pool5	88.81%	70.06%	77.62%	92.47%	7.53%	22.38%	11.19%	85.05%	77.62%	84.72%
RESNET101 – fc1000	86.67%	66.40%	73.00%	91.10%	8.90%	27.00%	13.33%	82.05%	73.00%	81.55%
SQUEEZENET – pool10	84.52%	62.10%	70.06%	90.11%	9.89%	29.94%	15.48%	80.09%	70.06%	79.46%

Tabla
Comparación de indicadores del Experimento #2

	Exactitud	Precisión	Sensitividad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error	AUC-ROC	TPP-Prom	G-Mean
ALEXNET-pool5	91.19%	75.31%	80.98%	93.93%	6.07%	19.02%	8.81%	87.46%	80.98%	87.21%
ALEXNET-fc6	88.33%	67.32%	73.26%	92.21%	7.79%	26.74%	11.67%	82.74%	73.26%	82.19%
ALEXNET-fc7	88.33%	68.23%	75.85%	92.70%	7.30%	24.15%	11.67%	84.28%	75.85%	83.85%
ALEXNET-fc8	87.38%	67.02%	75.69%	92.01%	7.99%	24.31%	12.62%	83.85%	75.69%	83.45%
DENSENET201 – avg_pool	85.24%	64.11%	72.70%	90.66%	9.34%	27.30%	14.76%	81.68%	72.70%	81.18%
DENSENET201 – fc1000	86.19%	65.58%	74.49%	91.12%	8.88%	25.51%	13.81%	82.81%	74.49%	82.39%
GOOGLENET – pool5-7x7_s1	87.38%	68.31%	76.93%	91.45%	8.55%	23.07%	12.62%	84.19%	76.93%	83.88%
GOOGLENET – loss3-classifier	84.29%	61.62%	70.16%	89.69%	10.31%	29.84%	15.71%	79.93%	70.16%	79.33%
INCEPTIONV2 – avg_pool	84.76%	61.69%	69.77%	90.06%	9.94%	30.23%	15.24%	79.92%	69.78%	79.27%
INCEPTIONV2 – predictions	82.62%	57.08%	63.00%	88.48%	11.52%	37.00%	17.38%	75.74%	63.00%	74.66%
INCEPTIONV3 – avg_pool	85.24%	63.17%	71.38%	90.37%	9.63%	28.62%	14.76%	80.88%	71.38%	80.32%
INCEPTIONV3 – predictions	85.00%	61.96%	68.77%	89.84%	10.16%	31.23%	15.00%	79.31%	68.77%	78.60%
RESNET18 – pool5	83.57%	60.55%	68.64%	89.22%	10.78%	31.36%	16.43%	78.93%	68.64%	78.26%
RESNET18 – fc1000	83.33%	59.71%	67.90%	89.33%	10.67%	32.10%	16.67%	78.62%	67.90%	77.88%
RESNET50 – avg_pool	90.95%	73.79%	81.63%	94.43%	5.57%	18.37%	9.05%	88.03%	81.63%	87.80%
RESNET50 – fc1000	90.00%	72.17%	81.67%	93.90%	6.10%	18.33%	10.00%	87.79%	81.67%	87.57%
RESNET101 – pool5	87.86%	68.12%	77.05%	92.09%	7.91%	22.95%	12.14%	84.57%	77.05%	84.23%
RESNET101 – fc1000	85.71%	64.43%	73.15%	90.36%	9.64%	26.85%	14.29%	81.76%	73.15%	81.30%
SQUEEZENET – pool10	85.71%	65.48%	75.28%	91.49%	8.51%	24.72%	14.29%	83.39%	75.28%	82.99%

Tabla
Comparación de indicadores del Experimento #3

	Exactitud	Precisión	Sensitividad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error	AUC-ROC	TPP-Prom	G-Mean
ALEXNET-pool5	93.81%	88.64%	76.96%	94.22%	5.78%	23.04%	6.19%	85.59%	76.96%	85.15%
ALEXNET-fc6	93.33%	85.32%	76.72%	94.06%	5.94%	23.28%	6.67%	85.39%	76.72%	84.95%
ALEXNET-fc7	90.71%	73.78%	67.52%	92.34%	7.66%	32.48%	9.29%	79.93%	67.53%	78.96%
ALEXNET-fc8	90.48%	74.49%	70.98%	92.41%	7.59%	29.02%	9.52%	81.70%	70.98%	80.99%
DENSENET201 – avg_pool	90.48%	72.28%	70.11%	92.41%	7.59%	29.89%	9.52%	81.26%	70.12%	80.49%
DENSENET201 – fc1000	90.48%	72.01%	71.59%	92.74%	7.26%	28.41%	9.52%	82.17%	71.59%	81.48%
GOOGLENET – pool5-7x7_s1	88.81%	71.79%	67.68%	90.81%	9.19%	32.32%	11.19%	79.25%	67.68%	78.40%
GOOGLENET – loss3-classifier	87.86%	67.94%	66.49%	90.13%	9.87%	33.51%	12.14%	78.31%	66.49%	77.41%
INCEPTIONV2 – avg_pool	89.52%	70.62%	66.29%	91.79%	8.21%	33.71%	10.48%	79.04%	66.29%	78.00%
INCEPTIONV2 – predictions	88.10%	65.87%	61.86%	90.39%	9.61%	38.14%	11.90%	76.13%	61.86%	74.78%
INCEPTIONV3 – avg_pool	91.19%	79.36%	71.44%	93.08%	6.92%	28.56%	8.81%	82.26%	71.44%	81.55%
INCEPTIONV3 – predictions	88.33%	66.64%	64.38%	91.11%	8.89%	35.62%	11.67%	77.75%	64.38%	76.59%
RESNET18 – pool5	89.05%	74.42%	69.08%	91.12%	8.88%	30.92%	10.95%	80.10%	69.08%	79.34%
RESNET18 – fc1000	86.90%	64.59%	62.41%	89.87%	10.13%	37.59%	13.10%	76.14%	62.41%	74.89%
RESNET50 – avg_pool	94.52%	89.12%	81.59%	95.31%	4.69%	18.41%	5.48%	88.45%	81.59%	88.18%
RESNET50 – fc1000	92.14%	79.87%	73.32%	93.34%	6.66%	26.68%	7.86%	83.33%	73.32%	82.73%
RESNET101 – pool5	92.86%	83.29%	76.89%	94.27%	5.73%	23.11%	7.14%	85.58%	76.89%	85.14%
RESNET101 – fc1000	90.71%	75.59%	72.79%	92.87%	7.13%	27.21%	9.29%	82.83%	72.80%	82.22%
SQUEEZENET – pool10	90.71%	75.09%	72.08%	92.49%	7.51%	27.92%	9.29%	82.29%	72.08%	81.65%

Tabla
Comparación de indicadores del Experimento #4

	Exactitud	Precisión	Sensitividad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error	AUC-ROC	TPP-Prom	G-Mean
ALEXNET-pool5	93.57%	86.92%	76.74%	94.19%	5.81%	23.26%	6.43%	85.47%	76.74%	85.02%
ALEXNET-fc6	93.33%	86.14%	78.35%	94.33%	5.67%	21.65%	6.67%	86.34%	78.36%	85.97%
ALEXNET-fc7	91.90%	77.99%	73.41%	93.51%	6.49%	26.59%	8.10%	83.46%	73.41%	82.85%
ALEXNET-fc8	90.71%	72.23%	70.94%	92.97%	7.03%	29.06%	9.29%	81.96%	70.95%	81.21%
DENSENET201 – avg_pool	92.38%	78.67%	75.95%	93.98%	6.02%	24.05%	7.62%	84.97%	75.95%	84.49%
DENSENET201 – fc1000	91.43%	74.94%	74.57%	93.48%	6.52%	25.43%	8.57%	84.03%	74.57%	83.49%
GOOGLENET – pool5-7x7_s1	90.95%	78.47%	75.17%	92.19%	7.81%	24.83%	9.05%	83.68%	75.17%	83.25%
GOOGLENET – loss3-classifier	89.05%	72.12%	68.25%	90.82%	9.18%	31.75%	10.95%	79.54%	68.26%	78.73%
INCEPTIONV2 – avg_pool	89.05%	72.23%	67.12%	91.02%	8.98%	32.88%	10.95%	79.07%	67.12%	78.16%
INCEPTIONV2 – predictions	87.38%	67.73%	66.16%	89.58%	10.42%	33.84%	12.62%	77.87%	66.16%	76.98%
INCEPTIONV3 – avg_pool	91.19%	77.41%	73.65%	93.06%	6.94%	26.35%	8.81%	83.36%	73.65%	82.79%
INCEPTIONV3 – predictions	90.48%	73.47%	71.05%	92.76%	7.24%	28.95%	9.52%	81.91%	71.05%	81.18%
RESNET18 – pool5	90.24%	74.39%	72.52%	92.20%	7.80%	27.48%	9.76%	82.36%	72.52%	81.77%
RESNET18 – fc1000	89.76%	71.72%	72.33%	92.29%	7.71%	27.67%	10.24%	82.31%	72.33%	81.70%
RESNET50 – avg_pool	94.05%	87.79%	78.46%	94.61%	5.39%	21.54%	5.95%	86.54%	78.46%	86.16%
RESNET50 – fc1000	92.86%	82.20%	76.91%	94.02%	5.98%	23.09%	7.14%	85.47%	76.91%	85.04%
RESNET101 – pool5	92.62%	81.17%	79.34%	94.19%	5.81%	20.66%	7.38%	86.77%	79.34%	86.45%
RESNET101 – fc1000	89.52%	70.46%	70.84%	91.83%	8.17%	29.16%	10.48%	81.34%	70.84%	80.66%
SQUEEZENET – pool10	90.24%	70.80%	72.14%	92.69%	7.31%	27.86%	9.76%	82.42%	72.14%	81.77%

Tabla
Comparación de indicadores promedio para redes AlexNet y ResNet-50

	Exact.	Prec.	Sen.	Esp.	TFP	TFN	AUC-ROC	TP-A	G-Mean
ALEXNET-EXP#1	91.67%	75.58%	80.70%	94.28%	5.72%	19.30%	87.49%	80.70%	87.23%
ALEXNET-EXP#2	91.19%	75.31%	80.98%	93.93%	6.07%	19.02%	87.46%	80.98%	87.21%
ALEXNET-EXP#3	93.81%	88.64%	76.96%	94.22%	5.78%	23.04%	85.59%	76.96%	85.15%
ALEXNET-EXP#4	93.57%	86.92%	86.92%	76.74%	5.81%	23.26%	85.47%	76.74%	81.67%
RESNET50 - EXP#1	91.67%	75.54%	80.96%	94.43%	5.57%	19.04%	87.70%	80.96%	87.44%
RESNET50 - EXP#2	90.95%	73.79%	81.63%	94.43%	5.57%	18.37%	88.03%	81.63%	87.80%
RESNET50 - EXP#3	94.52%	89.12%	81.59%	95.31%	4.69%	18.41%	88.45%	81.59%	88.18%
RESNET50 - EXP#4	94.05%	87.79%	78.46%	94.61%	5.39%	21.54%	86.54%	78.46%	86.16%

En la Tabla se han colocado los índices promedio obtenidos en los cuatro experimentos por los 2 clasificadores más exitosos. Considerando los 9 indicadores allí mostrados se tiene que:

- Para la red AlexNet :
 - Del experimento #1 al #2 (cambio de estrategia OVA a OVO), hay mejora en 3 indicadores y desmejora en 6. Todas muy poco apreciables
 - Del experimento #1 al #3 (desbalance en los datos de entrenamiento), mejoran 2 indicadores. Los cambios notables ocurrieron en la exactitud (+2.14%), precisión (13.06%), sensibilidad (-3.74%) y tasa de falsos positivos (-3.74%).
 - Del experimento #3 al #4 (cambio de estrategia). Solo mejora la sensibilidad (aumenta +9.96%). La especificidad se reduce notablemente al pasar de 94.22% a 76.74%.
 - Del experimento #2 al #4 se observa mejoría en 4 de los 13 indicadores. La exactitud, precisión y sensibilidad aumentan mientras que la especificidad desciende notablemente del 94.28% a 76.84% (-17.44%).
- Para la red ResNet-50:
 - Del experimento #1 al #2 (estrategia OVA a OVO), mejora ligeramente la tasa de falsos negativos la cual varía en 0.67%, sensibilidad +0.67%, área bajo curva ROC aumenta en 0.34% y otros 3 indicadores. La mayor variación (-1.75%) se registró en la precisión.
 - Del experimento #1 al #3, mejoran todos los indicadores entre ellos exactitud (+2.85%), precisión (+13.58%) y AUC-ROC (+0.75%). Fue el mejor resultado para esta versión del clasificador.
 - Del experimento #3 al #4 (cambio de estrategia) no mejora ningún indicador.
 - Del experimento #2 al # 4 mejoran 4 indicadores. La exactitud aumenta en 3.10% y la precisión lo hace en +14%. Por otra parte, la tasa de falsos negativos y sensibilidad desmejoran ambas en 3.17%, también lo hacen los valores de AUC-ROC, TP-A y G-mean en 1.5%, 3.18% y 1.64%, respectivamente.

La variación experimentada por cada uno de los indicadores de funcionamiento del clasificador se resume en la Tabla , se indica el valor y un icono que representa si el cambio es favorable o desfavorable para el resultado.

Tabla

Diferencias de los indicadores de funcionamiento del clasificador

	Exact.	Prec.	Sen.	Esp.	TFP	TFN	AUC-ROC	TP-A	G-Mean
AlexNet - pool5 + Clasificador SVM									
EXP#1 vs EXP#2	✗ -0.48%	✗ -0.27%	✓ 0.28%	✗ -0.35%	✗ 0.35%	✓ -0.28%	✗ -0.04%	✓ 0.28%	✗ -0.01%
EXP#1 vs EXP#3	✓ 2.14%	✓ 13.06%	✗ -3.74%	✗ -0.06%	✗ 0.06%	✗ 3.74%	✗ -1.90%	✗ -3.74%	✗ -2.07%
EXP#3 vs EXP#4	✗ -0.24%	✗ -1.72%	✓ 9.96%	✗ -17.48%	✗ 0.03%	✗ 0.22%	✗ -0.13%	✗ -0.22%	✗ -3.48%
EXP#2 vs EXP#4	✓ 2.38%	✓ 11.61%	✓ 5.94%	✗ -17.19%	✓ -0.26%	✗ 4.24%	✗ -1.99%	✗ -4.24%	✗ -5.54%
ResNet - avg_pool + Clasificador SVM									
EXP#1 vs EXP#2	✗ -0.72%	✗ -1.75%	✓ 0.67%	✓ 0.00%	✗ 0.00%	✓ -0.67%	✓ 0.34%	✓ 0.68%	✓ 0.36%
EXP#1 vs EXP#3	✓ 2.85%	✓ 13.58%	✓ 0.63%	✓ 0.88%	✓ -0.88%	✓ -0.63%	✓ 0.76%	✓ 0.63%	✓ 0.75%
EXP#3 vs EXP#4	✗ -0.47%	✗ -1.33%	✗ -3.13%	✗ -0.70%	✗ 0.70%	✗ 3.13%	✗ -1.92%	✗ -3.14%	✗ -2.03%
EXP#2 vs EXP#4	✓ 3.10%	✓ 14.00%	✗ -3.17%	✓ 0.18%	✓ -0.18%	✗ 3.17%	✗ -1.50%	✗ -3.18%	✗ -1.64%

Finalmente, se puede resaltar que indicadores importantes como el área bajo la curva ROC, el promedio de verdaderos positivos (diagonal de la matriz de confusión) y la media geométrica (G-Mean) no resultan beneficiados al entrenar usando conjuntos de datos desbalanceados. El cambio de la estrategia de composición del clasificador afectó poco estas métricas al utilizar el conjunto de datos balanceado, siendo ligeramente mejores los resultados al usar el esquema 1 vs todos.

Entrenamiento y clasificación con datos aumentados

Los resultados del experimento #5 se muestran en la Tabla . A partir de ella se construyeron los gráficos de exactitud global y promedio de verdaderos positivos mostrados en la Figura . Nuevamente se obtuvo mejores resultados al utilizar los rasgos extraídos de las redes AlexNet y ResNet-50.

Tabla*Resultados Experimento #5. Clasificador usando conjunto balanceado de datos aumentados*

Red-capá		1 vs todos				1 vs 1			
ALEXNET -pool5	Exact.	88.50%				85.71%			
	T.Ent(s)	1584.91				1371.94			
		E	L	M	N	E	L	M	N
	E	0.73	0.00	0.05	0.23	0.77	0.00	0.05	0.18
	L	0.00	0.94	0.02	0.04	0.00	0.98	0.02	0.00
	M	0.05	0.19	0.71	0.05	0.00	0.24	0.62	0.14
	N	0.05	0.04	0.04	0.87	0.07	0.04	0.04	0.86
GOOGLNET - pool5-7x7_s1	Exact.	79.52%				78.57%			
	T.Ent(s)	1604.68				954.80			
		E	L	M	N	E	L	M	N
	E	0.45	0.09	0.05	0.41	0.45	0.09	0.09	0.36
	L	0.04	0.83	0.08	0.06	0.04	0.81	0.08	0.08
	M	0.10	0.10	0.81	0.00	0.05	0.14	0.76	0.05
	N	0.06	0.05	0.04	0.84	0.02	0.06	0.08	0.84
INCEPTIONV3 - avg_pool	Exact.	78.10%				78.10%			
	T.Ent(s)	1623.93				308.38			
		E	L	M	N	E	L	M	N
	E	0.64	0.09	0.05	0.23	0.59	0.09	0.05	0.27
	L	0.09	0.79	0.02	0.09	0.02	0.79	0.06	0.13
	M	0.00	0.14	0.67	0.19	0.05	0.14	0.67	0.14
	N	0.08	0.05	0.04	0.82	0.06	0.04	0.07	0.83
RESNET18 - pool5	Exact.	81.91%				81.91%			
	T.Ent(s)	1500.04				949.13			
		E	L	M	N	E	L	M	N
	E	0.64	0.18	0.00	0.18	0.73	0.09	0.00	0.18
	L	0.02	0.87	0.06	0.06	0.02	0.91	0.02	0.06
	M	0.05	0.24	0.57	0.14	0.00	0.33	0.57	0.10
	N	0.04	0.06	0.03	0.88	0.04	0.10	0.02	0.84
RESNET50 - avg_pool	Exact.	83.81%				89.05%			
	T.Ent(s)	1083.34				150.39			
		E	L	M	N	E	L	M	N
	E	0.59	0.05	0.05	0.32	0.64	0.05	0.05	0.27
	L	0.00	0.98	0.00	0.02	0.00	0.96	0.02	0.02
	M	0.10	0.14	0.57	0.19	0.05	0.10	0.81	0.05
	N	0.08	0.03	0.03	0.87	0.03	0.03	0.03	0.92
SQUEEZENET - pool10	Exact.	81.43%				76.67%			
	T.Ent(s)	1336.40				943.75			
		E	L	M	N	E	L	M	N
	E	0.64	0.00	0.09	0.27	0.41	0.05	0.27	0.27
	L	0.00	0.91	0.02	0.08	0.00	0.83	0.13	0.04
	M	0.10	0.05	0.71	0.14	0.10	0.05	0.71	0.14
	N	0.09	0.05	0.04	0.82	0.08	0.08	0.03	0.82

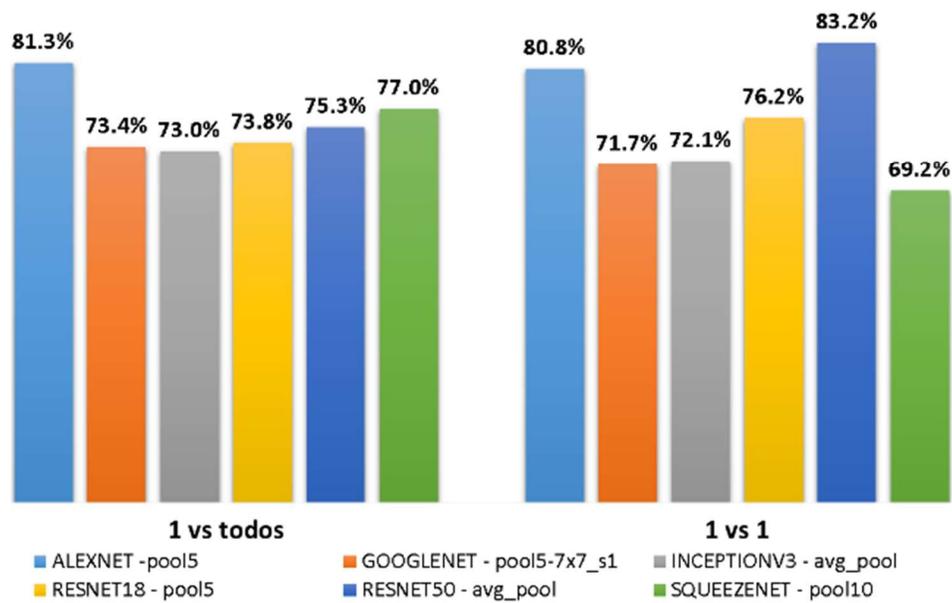
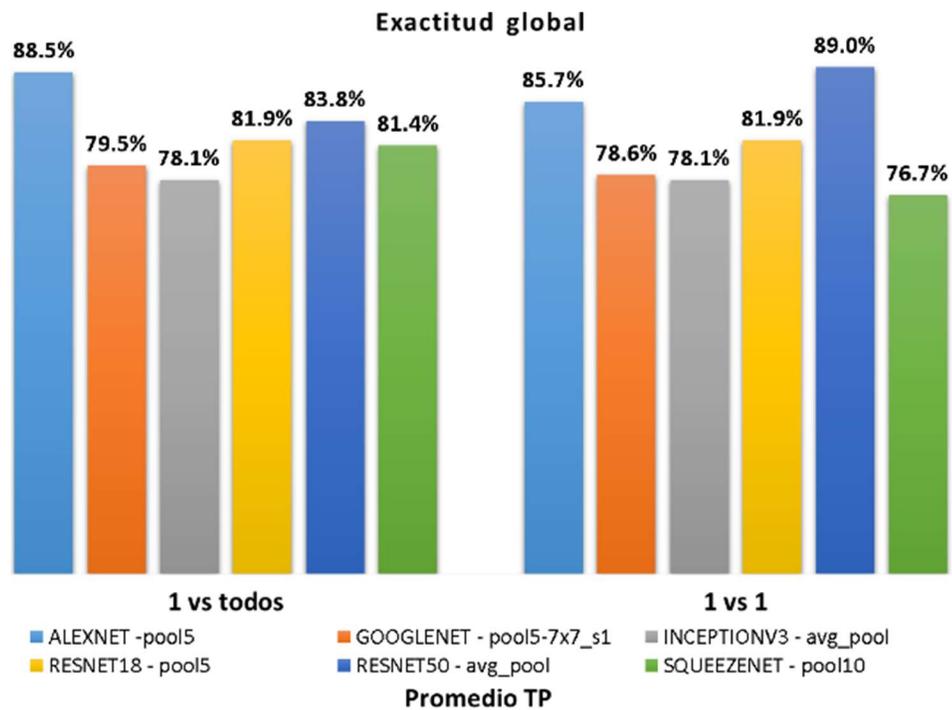


Figura . Gráficos exactitud global (arriba) y promedio de verdaderos positivos (abajo)

Se calcularon las métricas promedio de rendimiento (tablas 27 y 28) y se agruparon los indicadores para realizar la comparación de los clasificadores (tablas 29 y 30).

Tabla
Métricas de evaluación para clasificadores del Experimento #5 con esquema OVA

Red-capa		Exactitud	Precisión	Sensibilidad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error	AUC-ROC
ALEXNET-pool5	E	0.9381	0.6957	0.7273	0.9628	0.0372	0.2727	0.0619	0.8368
	L	0.9429	0.8475	0.9434	0.9427	0.0573	0.0566	0.0571	0.9430
	M	0.9429	0.7143	0.7143	0.9683	0.0317	0.2857	0.0571	0.8317
	N	0.8905	0.9252	0.8684	0.9167	0.0833	0.1316	0.1095	0.8922
	Prom	0.9286	0.7957	0.8133	0.9476	0.0524	0.1867	0.0714	0.8759
	Desv.Std	0.0255	0.1097	0.1113	0.0234	0.0234	0.1113	0.0255	0.0525
GOOGLNET - pool5-7x7_s1	E	0.8905	0.4762	0.4545	0.9415	0.0585	0.5455	0.1095	0.6541
	L	0.9095	0.8148	0.8302	0.9363	0.0637	0.1698	0.0905	0.8817
	M	0.9333	0.6296	0.8095	0.9471	0.0529	0.1905	0.0667	0.8756
	N	0.8571	0.8889	0.8421	0.8750	0.1250	0.1579	0.1429	0.8584
	Prom	0.8976	0.7024	0.7341	0.9250	0.0750	0.2659	0.1024	0.8174
	Desv.Std	0.0322	0.1861	0.1868	0.0336	0.0336	0.1868	0.0322	0.1093
INCEPTIONV3 - avg_pool	E	0.8952	0.5000	0.6364	0.9255	0.0745	0.3636	0.1048	0.7675
	L	0.8952	0.7925	0.7925	0.9299	0.0701	0.2075	0.1048	0.8585
	M	0.9333	0.6667	0.6667	0.9630	0.0370	0.3333	0.0667	0.8013
	N	0.8381	0.8704	0.8246	0.8542	0.1458	0.1754	0.1619	0.8393
	Prom	0.8905	0.7074	0.7300	0.9181	0.0819	0.2700	0.1095	0.8166
	Desv.Std	0.0393	0.1617	0.0924	0.0458	0.0458	0.0924	0.0393	0.0405
RESNET18 - pool5	E	0.9333	0.7000	0.6364	0.9681	0.0319	0.3636	0.0667	0.7849
	L	0.8905	0.7419	0.8679	0.8981	0.1019	0.1321	0.1095	0.8829
	M	0.9286	0.6667	0.5714	0.9683	0.0317	0.4286	0.0714	0.7438
	N	0.8857	0.9091	0.8772	0.8958	0.1042	0.1228	0.1143	0.8865
	Prom	0.9095	0.7544	0.7382	0.9326	0.0674	0.2618	0.0905	0.8245
	Desv.Std	0.0249	0.1076	0.1574	0.0411	0.0411	0.1574	0.0249	0.0715
RESNET50 - avg_pool	E	0.9048	0.5417	0.5909	0.9415	0.0585	0.4091	0.0952	0.7459
	L	0.9619	0.8814	0.9811	0.9554	0.0446	0.0189	0.0381	0.9682
	M	0.9381	0.7500	0.5714	0.9788	0.0212	0.4286	0.0619	0.7479
	N	0.8714	0.8919	0.8684	0.8750	0.1250	0.1316	0.1286	0.8717
	Prom	0.9190	0.7662	0.7530	0.9377	0.0623	0.2470	0.0810	0.8334
	Desv.Std	0.0395	0.1630	0.2038	0.0445	0.0445	0.2038	0.0395	0.1074
SQUEEZENET - pool10	E	0.9048	0.5385	0.6364	0.9362	0.0638	0.3636	0.0952	0.7719
	L	0.9429	0.8727	0.9057	0.9554	0.0446	0.0943	0.0571	0.9302
	M	0.9381	0.6818	0.7143	0.9630	0.0370	0.2857	0.0619	0.8294
	N	0.8429	0.8785	0.8246	0.8646	0.1354	0.1754	0.1571	0.8444
	Prom	0.9071	0.7429	0.7702	0.9298	0.0702	0.2298	0.0929	0.8440
	Desv.Std	0.0461	0.1641	0.1188	0.0449	0.0449	0.1188	0.0461	0.0654

Tabla
Métricas de evaluación para clasificadores del Experimento #5 con esquema OVO

Red-capa		Exactitud	Precisión	Sensibilidad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error	AUC-ROC
ALEXNET-pool5	E	0.9381	0.6800	0.7727	0.9574	0.0426	0.2273	0.0619	0.8601
	L	0.9524	0.8525	0.9811	0.9427	0.0573	0.0189	0.0476	0.9617
	M	0.9333	0.6842	0.6190	0.9683	0.0317	0.3810	0.0667	0.7742
	N	0.8905	0.9333	0.8596	0.9271	0.0729	0.1404	0.1095	0.8927
	Prom	0.9286	0.7875	0.8081	0.9489	0.0511	0.1919	0.0714	0.8722
	Desv.Std	0.0267	0.1261	0.1523	0.0179	0.0179	0.1523	0.0267	0.0779
GOOGLENET - pool5-7x7_s1	E	0.9190	0.6667	0.4545	0.9734	0.0266	0.5455	0.0810	0.6651
	L	0.8952	0.7818	0.8113	0.9236	0.0764	0.1887	0.1048	0.8656
	M	0.9048	0.5161	0.7619	0.9206	0.0794	0.2381	0.0952	0.8375
	N	0.8524	0.8807	0.8421	0.8646	0.1354	0.1579	0.1476	0.8533
	Prom	0.8929	0.7113	0.7175	0.9205	0.0795	0.2825	0.1071	0.8054
	Desv.Std	0.0287	0.1568	0.1784	0.0445	0.0445	0.1784	0.0287	0.0942
INCEPTIONV3 - avg_pool	E	0.9143	0.5909	0.5909	0.9521	0.0479	0.4091	0.0857	0.7501
	L	0.9048	0.8235	0.7925	0.9427	0.0573	0.2075	0.0952	0.8643
	M	0.9095	0.5385	0.6667	0.9365	0.0635	0.3333	0.0905	0.7902
	N	0.8333	0.8559	0.8333	0.8333	0.1667	0.1667	0.1667	0.8333
	Prom	0.8905	0.7022	0.7208	0.9162	0.0838	0.2792	0.1095	0.8095
	Desv.Std	0.0383	0.1608	0.1120	0.0556	0.0556	0.1120	0.0383	0.0499
RESNET18 - pool5	E	0.9429	0.7273	0.7273	0.9681	0.0319	0.2727	0.0571	0.8391
	L	0.8810	0.7059	0.9057	0.8726	0.1274	0.0943	0.1190	0.8890
	M	0.9429	0.8000	0.5714	0.9841	0.0159	0.4286	0.0571	0.7499
	N	0.8714	0.9143	0.8421	0.9063	0.0938	0.1579	0.1286	0.8736
	Prom	0.9095	0.7869	0.7616	0.9328	0.0672	0.2384	0.0905	0.8379
	Desv.Std	0.0387	0.0940	0.1467	0.0523	0.0523	0.1467	0.0387	0.0623
RESNET50 - avg_pool	E	0.9429	0.7778	0.6364	0.9787	0.0213	0.3636	0.0571	0.7892
	L	0.9619	0.8947	0.9623	0.9618	0.0382	0.0377	0.0381	0.9620
	M	0.9571	0.7727	0.8095	0.9735	0.0265	0.1905	0.0429	0.8877
	N	0.9190	0.9292	0.9211	0.9167	0.0833	0.0789	0.0810	0.9189
	Prom	0.9452	0.8436	0.8323	0.9577	0.0423	0.1677	0.0548	0.8895
	Desv.Std	0.0192	0.0802	0.1457	0.0282	0.0282	0.1457	0.0192	0.0735
SQUEEZENET - pool10	E	0.8857	0.4500	0.4091	0.9415	0.0585	0.5909	0.1143	0.6206
	L	0.9048	0.8000	0.8302	0.9299	0.0701	0.1698	0.0952	0.8786
	M	0.8952	0.4839	0.7143	0.9153	0.0847	0.2857	0.1048	0.8086
	N	0.8476	0.8942	0.8158	0.8854	0.1146	0.1842	0.1524	0.8499
	Prom	0.8833	0.6570	0.6923	0.9180	0.0820	0.3077	0.1167	0.7894
	Desv.Std	0.0250	0.2233	0.1957	0.0242	0.0242	0.1957	0.0250	0.1162

Tabla*Comparación de indicadores experimento #5 con esquema descomposición OVA*

	Exactitud	Precisión	Sensitividad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error	AUC-ROC	TP-A	G-Mean
ALEXNET-pool5	92.86%	79.57%	81.33%	94.76%	5.24%	18.67%	7.14%	88.05%	81.34%	87.79%
GOOGLENET - pool5-7x7_s1	89.76%	70.24%	73.41%	92.50%	7.50%	26.59%	10.24%	82.96%	73.41%	82.40%
INCEPTIONV3 - avg_pool	89.05%	70.74%	73.00%	91.81%	8.19%	27.00%	10.95%	82.41%	73.01%	81.87%
RESNET18 - pool5	90.95%	75.44%	73.82%	93.26%	6.74%	26.18%	9.05%	83.54%	73.82%	82.97%
RESNET50 - avg_pool	91.90%	76.62%	75.30%	93.77%	6.23%	24.70%	8.10%	84.54%	75.30%	84.03%
SQUEEZENET - pool10	90.71%	74.29%	77.02%	92.98%	7.02%	22.98%	9.29%	85.00%	77.03%	84.62%

Tabla*Comparación de indicadores experimento #5 con esquema descomposición OVO*

	Exactitud	Precisión	Sensitividad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error	AUC-ROC	TP-A	G-Mean
ALEXNET-pool5	92.86%	78.75%	80.81%	94.89%	5.11%	19.19%	7.14%	87.85%	80.81%	87.57%
GOOGLENET - pool5-7x7_s1	89.29%	71.13%	71.75%	92.05%	7.95%	28.25%	10.71%	81.90%	71.75%	81.27%
INCEPTIONV3 - avg_pool	89.05%	70.22%	72.08%	91.62%	8.38%	27.92%	10.95%	81.85%	72.09%	81.26%
RESNET18 - pool5	90.95%	78.69%	76.16%	93.28%	6.72%	23.84%	9.05%	84.72%	76.16%	84.29%
RESNET50 - avg_pool	94.52%	84.36%	83.23%	95.77%	4.23%	16.77%	5.48%	89.50%	83.23%	89.28%
SQUEEZENET - pool10	88.33%	65.70%	69.23%	91.80%	8.20%	30.77%	11.67%	80.52%	69.24%	79.72%

Estos resultados indican:

- El clasificador que utiliza la red AlexNet mostró su mejor comportamiento al usar el esquema 1 vs todos, aunque la diferencia fue poco significativa.
- El clasificador que utiliza la red ResNet-50 mostró su mejor comportamiento con el esquema 1 vs 1 con mejoras apreciables en todos los indicadores.
- Aumentar la cantidad de datos disponibles para el entrenamiento no implicó un cambio en cuales redes permitieron entrenar los clasificadores más exitosos.
- Utilizar un conjunto mucho mayor de datos de entrenamiento no significó un mejoramiento sustancial con respecto a los resultados logrados en los experimentos anteriores, dando a entender que el modelo del clasificador ya había alcanzado su mejor funcionamiento empleando menos imágenes. Esto reafirma la consideración de que este enfoque, de usar la red como extractor de características, permite alcanzar resultados satisfactorios cuando al nuevo conjunto de entrenamiento es pequeño y que el uso de datos aumentados poco puede aportar en este caso.

- El tiempo de entrenamiento de los clasificadores aumentó considerablemente, como era de esperarse, aunque ello no supuso una mejoría en el resultado.
- La clasificación de los eosinófilos y monocitos fue la menos acertada. En particular para estas células los mejores resultados fueron en los experimentos #1 y #2.

El código fuente utilizado para el entrenamiento de los clasificadores cuyos resultados se presentan en ésta sección se encuentra disponible en el apéndice A-3 “Scripts clasificador SVM”. Cada archivo lleva por nombre Exp_ *n*.m donde *n* indica el número del experimento, así el archivo “Exp_1.m” es el archivo para entrenamiento y prueba del experimento # 1. Para repetir los experimentos solamente se deben ajustar los directorios de origen de las imágenes, seleccionar el modelo pre-entrenado de red (el cual debe instalarse en el computador con anterioridad) y seleccionar la capa a utilizar como extractor de rasgos.

4.3.2. Clasificación mediante ajuste de pesos del modelo de red convolucional

4.3.2.1. *Ajuste de pesos (fine-tune) de redes neuronales convolucionales*

El abordaje usualmente empleado al hacer transferencia de aprendizaje consiste en entrenar una red base, copiar sus primeras capas a una nueva red, luego inicializar de forma aleatoria las capas finales y finalmente éstas se entrenan para realizar la nueva tarea. Al momento de realizar el entrenamiento se puede elegir si la retropropagación del error se hace hacia todas las capas de la red (incluso las copiadas inicialmente) o si, por el contrario, únicamente afectará las capas finales dejando las capas iniciales sin cambios (congeladas) durante el aprendizaje de la nueva tarea (Yosinski y otros, 2014).

Yosinski et al. (2014) indican que la elección de permitir el ajuste de pesos en las primeras capas de la nueva red depende del tamaño del conjunto de datos disponibles y el número de parámetros existentes en éstas. Si nuevos datos para entrenamiento son pocos y la red tiene un gran número de parámetros, permitir el ajuste de pesos en las

capas iniciales puede conducir a un sobreajuste del modelo y la pérdida de su capacidad de generalización o predicción correcta sobre nuevos datos. Por ello, en estas condiciones es usual mantener las primeras capas “congeladas” y no actualizarlas durante el entrenamiento. Por otro lado, si el conjunto de datos es grande o el número de parámetros de la red es pequeño, de manera que el sobreajuste no constituye un problema, entonces se puede permitir el ajuste de pesos en todas las capas para mejorar el rendimiento obtenido.

El enfoque de ajuste de pesos está presente en variedad de aplicaciones como por ejemplo: la clasificación de dígitos numéricos y rostros (Hussain, Bird, & Faria, 2018), identificación de objetivos en movimiento en imágenes de radar (Hadhrami, Mufti, Taha, & Werghi, 2018), clasificación de imágenes histológicas para detección de cáncer de mama (Shallu & Mehra, 2018), separación de tallos de cebolla y malas hierbas (Bäckström, 2017), reconocimiento de micro invertebrados en reservorios de agua (Quintero, Merchán, Cornejo, & Sánchez Galán, 2018), detección de pólipos y embolia pulmonar, clasificación de imágenes colonoscópicas y segmentación de barrera para medición del grosor íntima media de la arteria carótida (Tajbakhsh, y otros, 2016) ; entre otras.

Este método es más lento y requiere mayor esfuerzo que la simple extracción de rasgos, pero debido a que se pueden extraer características especializadas para la nueva tarea, la red resultante es frecuentemente más exacta. En efecto, algunos autores han encontrado mejores resultados al utilizar el ajuste de pesos como técnica de transferencia de aprendizaje sobre la extracción directa de características de la red convolucional como lo demuestran Gao y Mosalam (2018), Van de Wolfshaar y otros (2015) o Shahin y otros (2019).

4.3.2.2. Selección de los modelos de CNN

En esta etapa fueron utilizados los siguientes modelos de redes convolucionales: AlexNet, ResNet-50, GoogleNet, ResNet-18 y SqueezeNet. Los primeros dos se seleccionaron en función de los resultados obtenidos en la sección anterior, mientras que la red GoogleNet se incluyó como referencia (*gold standard*) al ser el modelo de CNN ganador la ILSVRC del año 2014. Sharma, Jain y Mishra (2018) compararon previamente el funcionamiento de estos 3 modelos de red convolucional al clasificar imágenes provenientes de las bases de datos CIFAR10 y CIFAR100, encontrando que ResNet-50 lograba mejor rendimiento en la primera mientras GoogleNet lo hacía en la segunda.

Por su parte, ResNet-18 y SqueezeNet son redes de tipo residual bastante compactas y desarrollo reciente, las cuales se han incluidos para comparar los resultados obtenidos al usar redes que poseen menor número de parámetros. Bäckström (2017) comparó los resultados de las redes AlexNet y SqueezeNet al realizar la clasificación de tallos de cebolla y malas hierbas, según sus resultados este último modelo de red logró resultados superiores pese a tener 50 veces menos parámetros que AlexNet y por tanto un menor costo computacional mucho menor. Otros modelos de redes convolucionales ampliamente utilizados como VGG16 (Simonyan & Zisserman, 2014) o ResNet-152 (He, Zhang, Ren, & Sun, 2016) fueron descartados debido al alto costo computacional que implica su utilización.

La Figura reproduce una ilustración publicada por Canziani y otros (2017) en la cual se compara la exactitud global, número de operaciones necesarias para clasificar una imagen y el total de parámetros que contiene la red. Se observa que las redes VGG-19, VGG-16 y ResNet-152 requieren el mayor número de operaciones de los modelos analizados por estos autores. La red AlexNet a pesar de tener una cantidad de parámetros similar a ResNet-152, requiere un costo computación muy inferior al clasificar las imágenes.

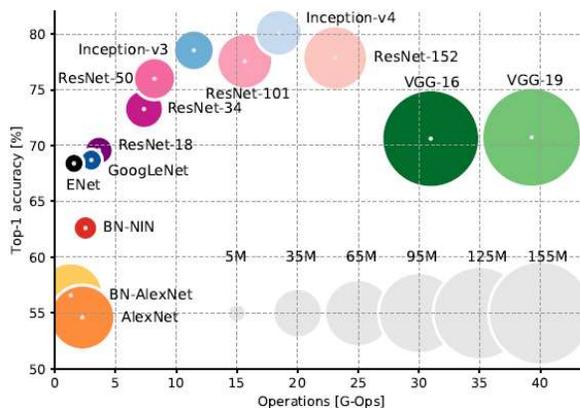


Figura . Comparación exactitud y número de operaciones necesarias para evaluar una imagen en diferentes modelos de redes neuronales convolucionales. El tamaño del círculo es proporcional a la cantidad de parámetros existentes en la red. Tomado de Canziani y otros (2017) .

4.3.2.3. Procedimiento empleado para ajustar el modelo de la red convolucional

Cada uno de los modelos pre-entrenados de red se dividió en tres regiones: tallo principal o sección inicial de la red, sección intermedia y salida, como se ejemplifica en la Figura para la red GoogLeNet. El tallo principal se retuvo, es decir no se re-entrenó esa sección de la red. La sección intermedia se ajustó utilizando una tasa de aprendizaje muy baja (se experimentó con valores entre 0.0001 y 0.01). En la salida de la red, fueron reemplazadas las últimas capas y en su lugar se colocó un grupo compuesto por una nueva capa completamente conectada de 4 neuronas de salida y una capa de clasificación de salida usando la función de pérdida de entropía categórica cruzada (implementada mediante una capa de activación *softmax* seguida de una capa clasificación de salida *cross-entropy*). Este conjunto de salida fue entrenado completamente para ajustarse a la clasificación de las nuevas imágenes. La Tabla muestra la identificación de las capas a reemplazar, reentrenar o mantener para ajustar el modelo de la CNN a la nueva tarea de clasificación de leucocitos.

Tabla

Capas a ser retenidas, ajustadas o completamente reentrenadas en las CNN

Modelo de red	Capas sección inicial (retenidas)	Capas sección central (ajuste fino)	Capas Salida (re-entrenar)
AlexNet	1-5	6-16	17-19
GoogLeNet	1-10	11-141	142-144
ResNet-18	1-6	7-69	70-72
ResNet-50	1-5	6-174	175-177
SqueezeNet	1-5	6-65	66-68

Salida

Entrenar desde cero



Sección Central
Ajuste fino

Talco principal de la red
No re-entrenar

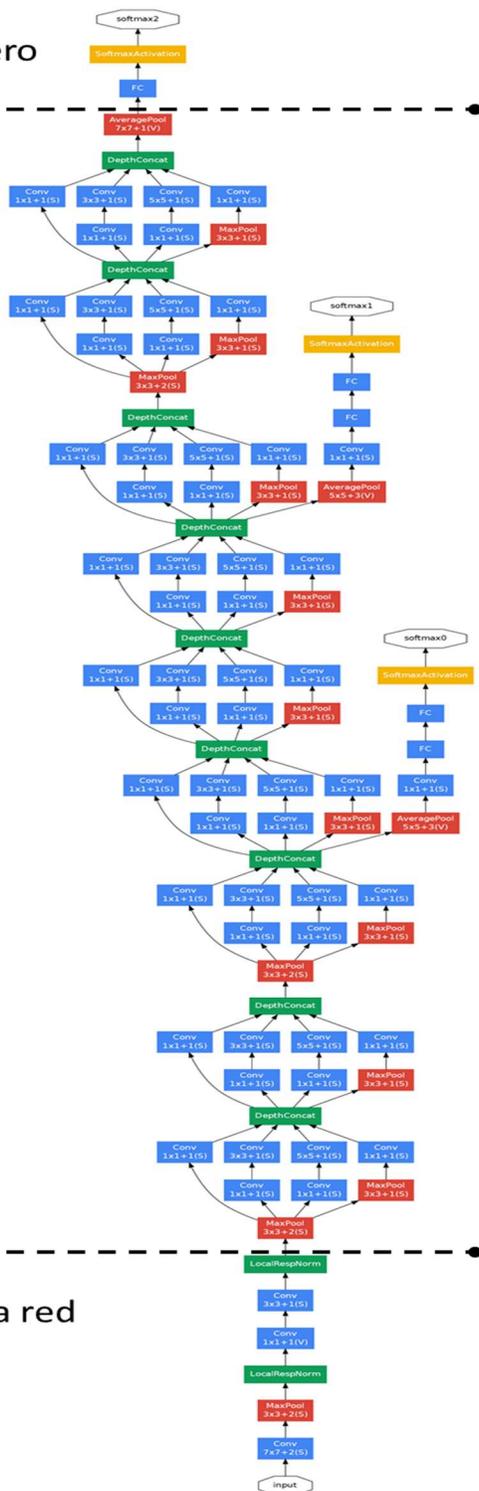


Figura . División de la red GoogleNet para su entrenamiento. Modificado de Szegedy y otros (2015)

El re-entrenamiento de las redes convolucionales se dividió en dos etapas. La primera de ellas (Experimento #6) realizó con las imágenes del conjunto CT1 y estaba destinada a determinar de forma experimental valores adecuados a ser utilizados durante el entrenamiento, es decir, la configuración de los parámetros como: algoritmo de optimización, momento, velocidad de aprendizaje, etc. La segunda etapa (Experimento #7) consistió en el entrenamiento de la red convolucional usando miles de imágenes del conjunto de datos aumentados CTA.

Para el experimento # 6 se utilizó el procedimiento que se describe a continuación:

- Utilizando la red AlexNet y una muestra balanceada de 163 imágenes del conjunto CT1, se entrenó la red usando los algoritmos de optimización *sgdm*, *adam* y *rmsprop* con los parámetros pre-definidos en el paquete MATLAB. Se permitió el ajuste de todos los pesos de la red. La tasa de aprendizaje se estableció inicialmente en 0.0001 al observar que valores elevados de este parámetro evitaban la reducción del error reportado en la función de pérdida. En base a estos resultados se seleccionó el algoritmo a usar.
- Dejando el total de épocas de entrenamiento en 10 y con el algoritmo de optimización de gradiente estocástico descendiente (*sgdm*) se realizaron combinaciones de los siguientes parámetros: tasa de aprendizaje global, factor de aprendizaje nuevas en capas de salida y momento para seleccionar valores a estos.
- Se utilizó la exactitud global de la clasificación como métrica principal para evaluar la efectividad del ajuste de cada parámetro junto con el análisis visual del comportamiento de los gráficos de la exactitud y pérdida (*loss*) obtenidos. Con esto se obtuvieron los parámetros de entrenamiento para proceder al experimento #7.
- Se entrenaron las redes SqueezeNet, GoogleNet, ResNet-50 y ResNet-18 utilizando para validar que los valores de parámetros obtenidos fuesen extrapolables al usar otro modelo de red.

El procedimiento utilizado para el experimento # 7 fue el siguiente:

- Se extrajo un sub-conjunto balanceado de 20860 muestras de cada clase, tomadas al azar del conjunto aumentado de datos CTA.
- El 90% de estos datos fueron empleados para entrenar la red, dejando el restante para validación. De esta forma se utilizaron 18774 imágenes por clase para entrenar el modelo y 2086 para validación.
- Se entrenó cada modelo de red de acuerdo a lo indicado al inicio de esta sección. Todas las redes fueron entrenadas y probadas usando exactamente los mismos conjuntos de datos para evitar que la selección aleatoria de diferentes imágenes pudiera influir en el resultado del entrenamiento y clasificación.
- Con el modelo de red ya entrenada para la nueva tarea se evaluó su predicción, determinando la exactitud y matriz de confusión para los siguientes conjuntos de imágenes:
 - Imágenes aumentadas de validación.
 - Conjunto “Prueba –C” de 210 imágenes no aumentadas.

En todos los casos, tanto los parámetros usados para las nuevas capas de la red neuronal como aquellos empleados para el entrenamiento (algoritmo, velocidad de aprendizaje, regularización, entre otros) fueron mantenidos en valores constantes. Los parámetros tamaño del lote (*mini-batch*) y cantidad de épocas usados para el entrenamiento fueron ajustados en función de los requerimientos de memoria consumida por cada modelo de red neuronal. El código fuente utilizado para el entrenamiento de los clasificadores cuyos resultados se presentan en ésta sección se encuentra disponible en el apéndice A-4 “Scripts ajuste de pesos CNN”.

4.3.2.4. *Resultados obtenidos con el ajuste de pesos de la red neuronal*

En esta sección se presentan mediante tablas y gráficos los resultados logrados al realizar el ajuste de pesos la red convolucional.

del decrecimiento esperado (ver Figura - C y F), ello permitió descartar ese valor. Al comparar el rendimiento obtenido con los valores restantes, se pudo observar que al usar 0.0001 (ver Figura - A y D) se lograba una mejor exactitud promedio. Al utilizar factor de aprendizaje igual a 20 se observaron mayores oscilaciones en la gráfica de exactitud. En función de estas pruebas se seleccionó usar un rango de valores para la tasa global de aprendizaje entre 0.0001 como mínimo y 0.0002 como máximo siendo preferido el primero y un factor de aprendizaje de 10.

Con estos valores establecidos se realizaron variaciones del parámetro momento (*momentum*) como se observa en la Figura . Para valores mayores a 0.95 se muestra gran degradación del resultado. En los valores más bajos de *momentum* ensayados (0.25 y 0.5) se observó una buena exactitud en el conjunto de validación, sin embargo la gráficas de exactitud y pérdida del conjunto de entrenamiento se estancan con respecto a lo observado con valores mayores a 0.8. A partir de las observaciones se decidió continuar con valores de momentum comprendidos en el rango de 0.8 a 0.95. Para la red AlexNet el valor 0.9 brindó resultados satisfactorios.

Finalmente se repitió el ensayo anterior usando variaciones del momento de 0.8 a 0.95 para los modelos de red convolucional seleccionados para esta sección del trabajo: GoogleNet (Figura), ResNet-18 (Figura), ResNet-50 (Figura) y SqueezeNet (Figura). El valor de momento 0.9 resultó ser adecuado para todos los modelos de red ensayados.

En general la exactitud obtenida en el conjunto de entrenamiento difiere en gran medida de la exactitud en el conjunto de validación, ello es indicativo de sobre ajuste en la red neuronal debido a que el conjunto de entrenamiento es pequeño.

Opciones de entrenamiento: Solver = **sgdm** Batch size = **20** WeightLearnRateFactor = **10** BiasLearnRateFactor = **10** Global Learning rate = **0.0001**
 Momentum = **0.25** Momentum = **0.5** Momentum = **0.8**

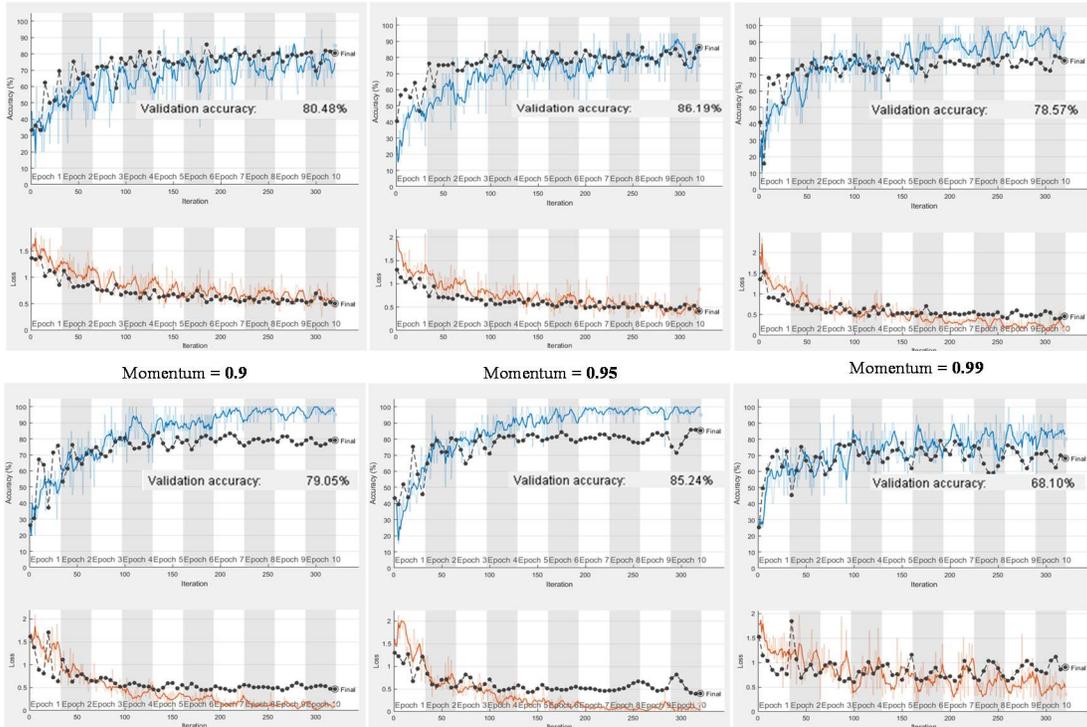


Figura . Ensayos con diferentes valores de momento usando red AlexNet. El factor de aprendizaje de se ajustó a 10 y la tasa de aprendizaje global a 0.0001.

Opciones de entrenamiento: Solver = **sgdm** Batch size = **20** WeightLearnRateFactor = **10** BiasLearnRateFactor = **10** Global Learning rate = **0.0001**
 Momentum = **0.8** Momentum = **0.85**

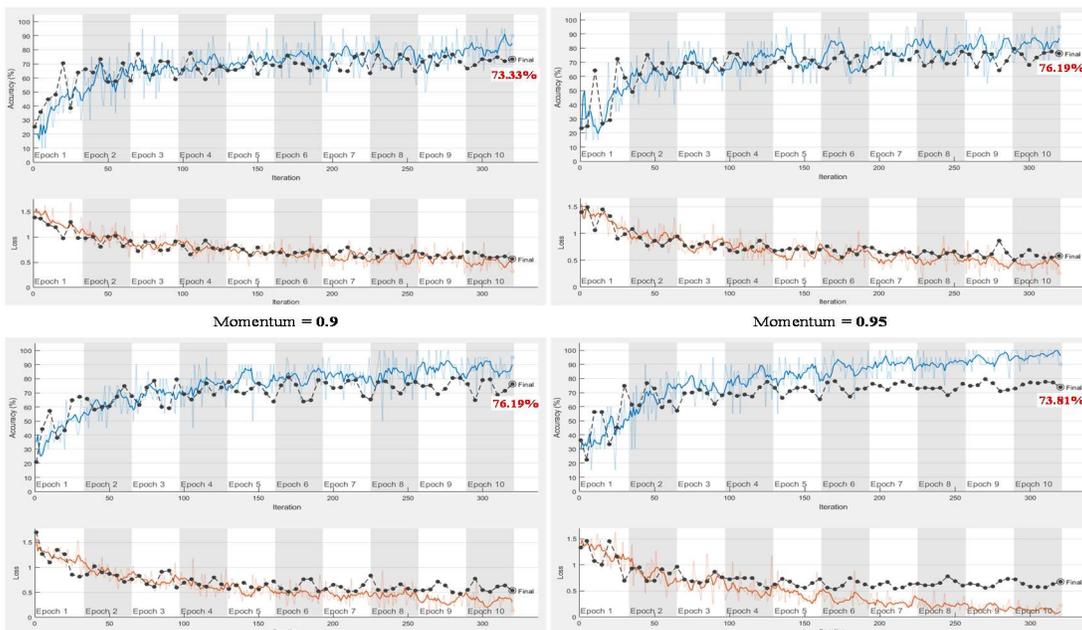


Figura . Ensayo entrenamiento usando red GoogleNet

Opciones de entrenamiento: Solver = **sgdm** Batch size = **20** WeightLearnRateFactor = **10** BiasLearnRateFactor = **10** Global Learning rate = **0.0001**
 Momentum = **0.8** Momentum = **0.85**

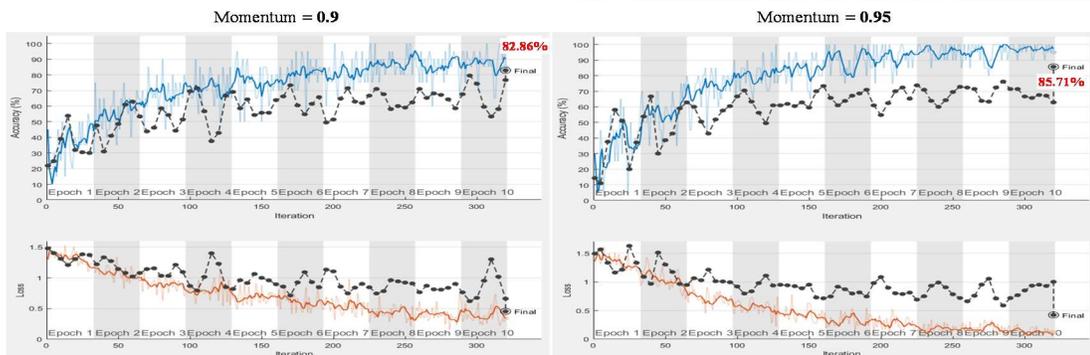
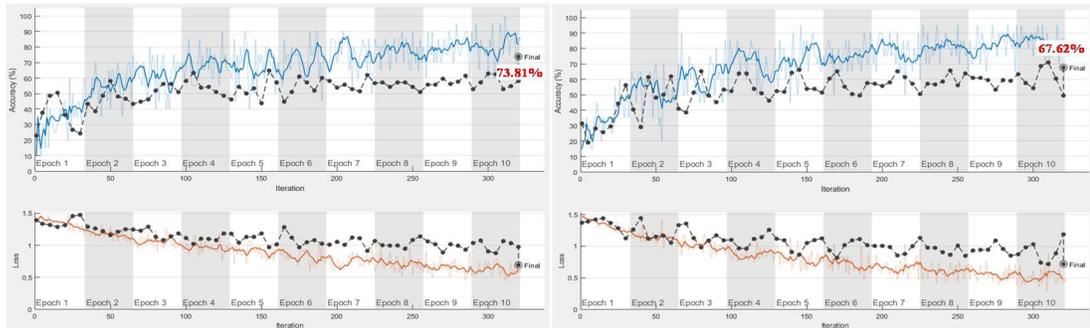


Figura . Ensayo entrenamiento usando red ResNet-18

Opciones de entrenamiento: Solver = **sgdm** Batch size = **20** WeightLearnRateFactor = **10** BiasLearnRateFactor = **10** Global Learning rate = **0.0001**
 Momentum = **0.8** Momentum = **0.85**

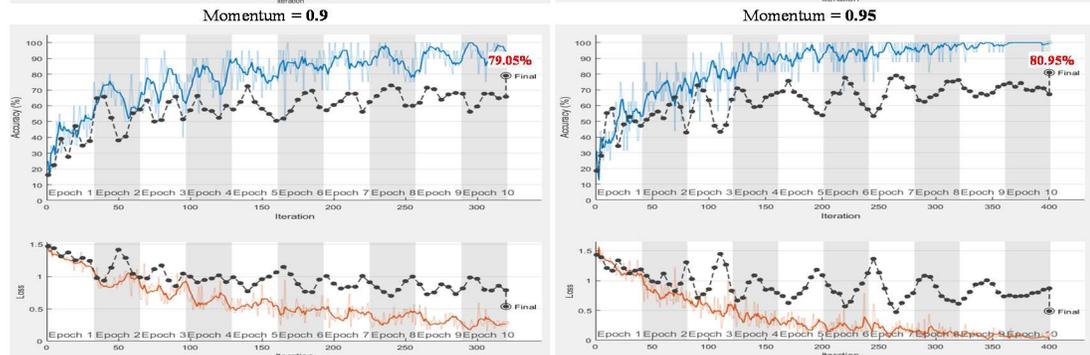
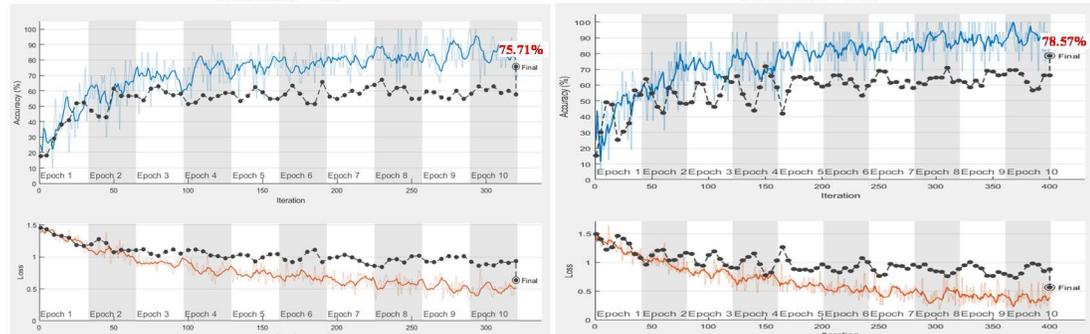


Figura . Ensayo entrenamiento usando red ResNet-50

Opciones de entrenamiento: Solver = **sgdm** Batch size = **20** WeightLearnRateFactor = **10** BiasLearnRateFactor = **10** Global Learning rate = **0.0001**
 Momentum = **0.8** Momentum = **0.85**

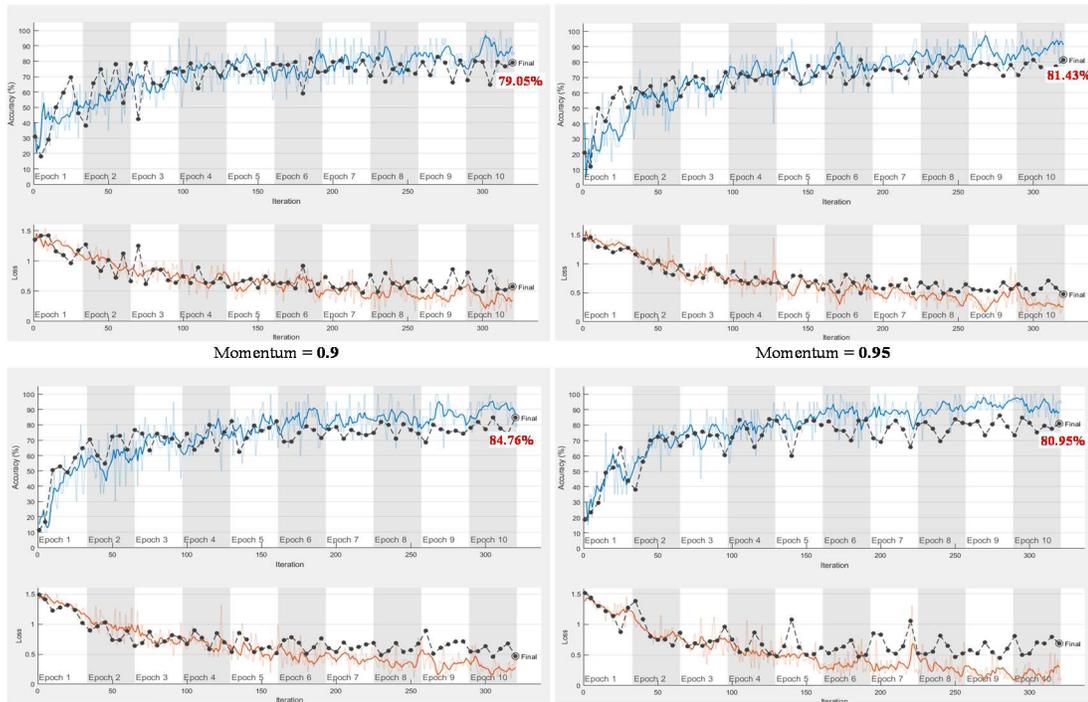
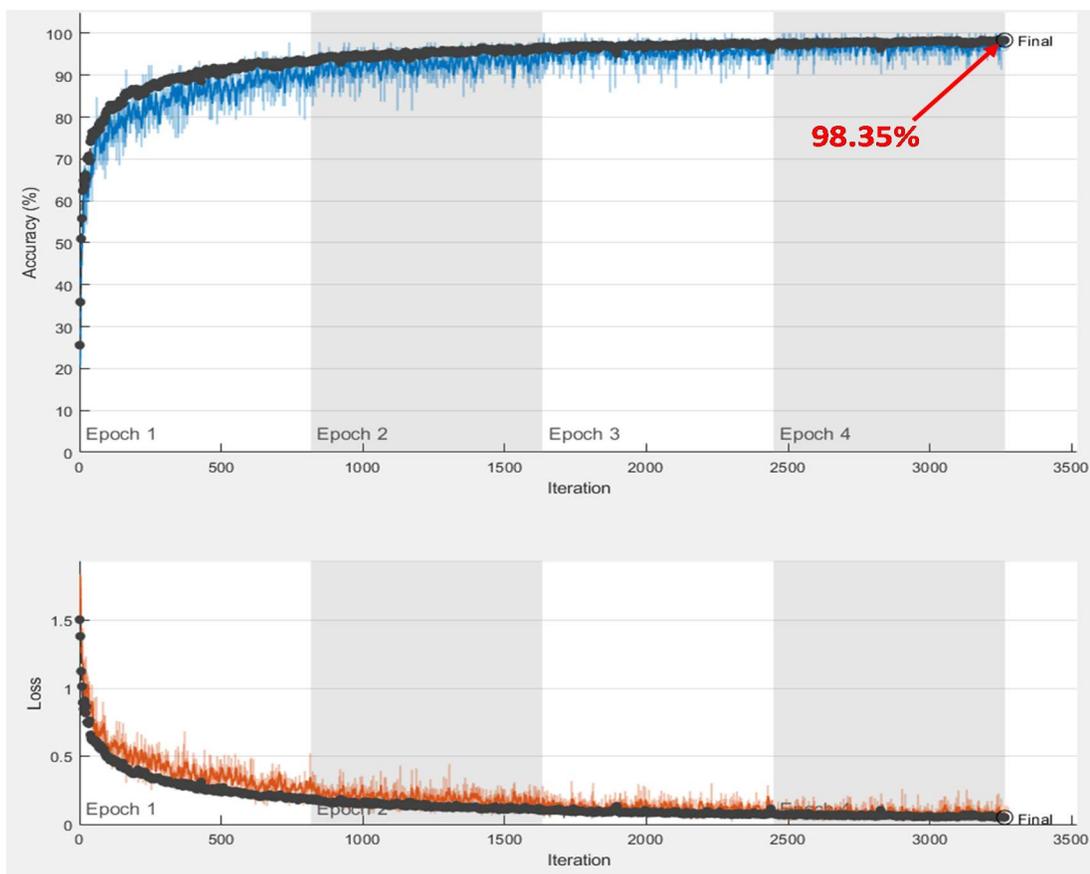


Figura . Ensayo entrenamiento usando red SqueezeNet

Ajuste de pesos entrenando con datos aumentados (Experimento #7)

Las figuras 94 a 98 presentan las gráficas de exactitud y pérdida obtenidas durante el entrenamiento de cada uno de los modelos de redes convolucionales utilizados. Al evaluar el conjunto de prueba de 210 imágenes de leucocitos se logró una exactitud global elevada, la cual va desde 87.62% hasta 94.76%, siendo las redes ResNet-18 y ResNet-50 las que exhiben los valores más elevados. La tasa de reconocimiento exitoso de cada tipo de leucocitos varió de la siguiente manera:

- Eosinófilos: desde 59.09 % (GoogleNet) hasta 86.36% (ResNet-18).
- Linfocitos: desde 90.57% (GoogleNet) hasta 98.11% (ResNet-18 y SqueezeNet).
- Monocitos: desde 76.19% (GoogleNet) hasta 100% (AlexNet).
- Neutrófilos segmentados: desde 91.23% (AlexNet) hasta 97.37 % (ResNet-18).



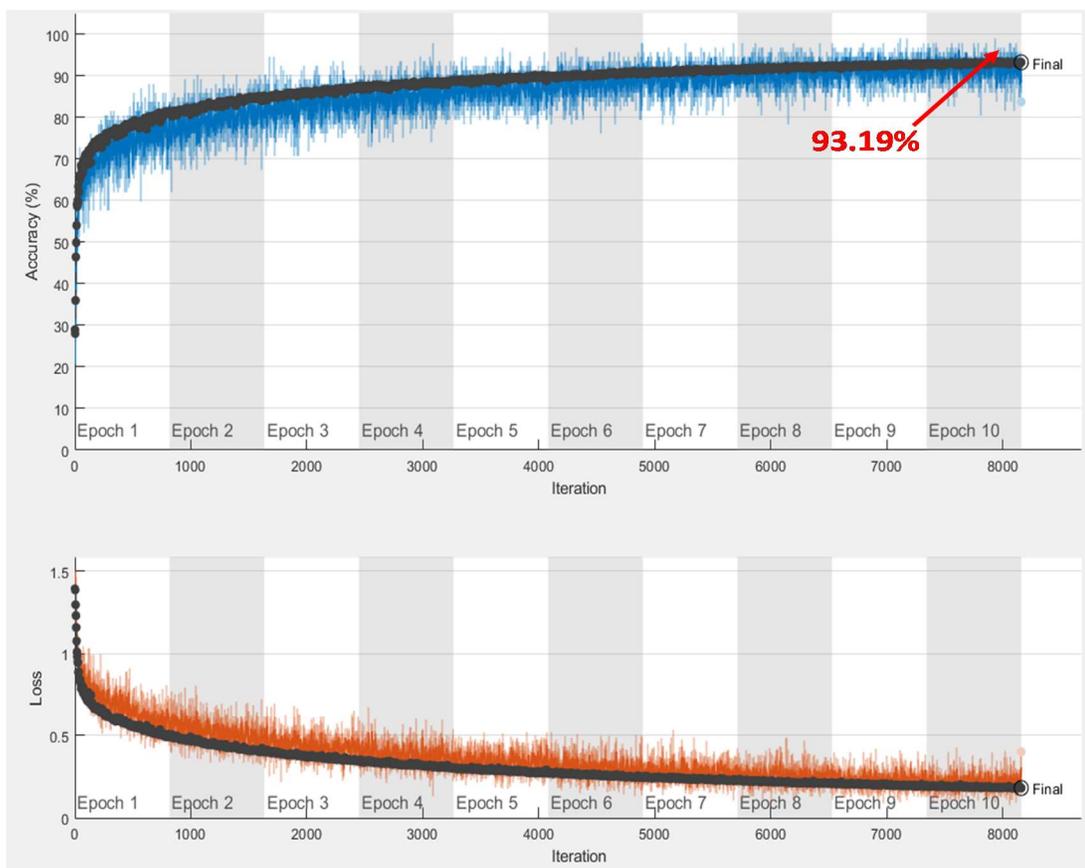
Conjunto de validación (10% de CT2)

Conjunto de Prueba - C

Exact. Global = 98.35%					Exact. Global = 90.95%					Prom.Ver.Pos = 90.24%						
	E	L	M	N	E	L	M	N	E	L	M	N	E	L	M	N
E	2069	3	9	5	99.19%	0.14%	0.43%	0.24%	17	0	2	3	77.27%	0.00%	9.09%	13.64%
L	1	2039	32	14	0.05%	97.75%	1.53%	0.67%	0	49	4	0	0.00%	92.45%	7.55%	0.00%
M	0	8	2068	10	0.00%	0.38%	99.14%	0.48%	0	0	21	0	0.00%	0.00%	100.00%	0.00%
N	17	19	20	2030	0.81%	0.91%	0.96%	97.32%	4	5	1	104	3.51%	4.39%	0.88%	91.23%

Figura . Entrenamiento de clasificador red AlexNet usando datos aumentados. **Arriba:** gráficos de exactitud y pérdida. **Abajo:** Matriz de confusión para los datos de validación y prueba.

Para la red AlexNet la exactitud global fue de 90.95%, logrando su mejor desempeño al clasificar exitosamente los 21 monocitos (100%) del conjunto de prueba y su nivel más bajo al identificar 17 de 22 eosinófilos (77.27%). Un 13.64% de los eosinófilos fue clasificado como neutrófilo segmentado siendo ésta la mayor confusión (en términos porcentuales) entre clases observada. Un total de 5 neutrófilos segmentados (4.39%) fueron clasificados como linfocitos. En el comportamiento de este clasificador se observó que 2 (9.09%) de los eosinófilos y 4 (7.55%) de los linfocitos fueron clasificados incorrectamente como monocitos.



Conjunto de validación (10% de CT2)								Conjunto de Prueba - C									
Exact. Global = 93.19%								Exact. Global = 87.62%				Prom.Ver.Pos = 79.93%					
	E	L	M	N	E	L	M	N	E	L	M	N	E	L	M	N	
E	2000	9	32	45	95.88%	0.43%	1.53%	2.16%	E	13	0	1	8	59.09%	0.00%	4.55%	36.36%
L	8	1926	54	98	0.38%	92.33%	2.59%	4.70%	L	2	48	1	2	3.77%	90.57%	1.89%	3.77%
M	29	50	1942	65	1.39%	2.40%	93.10%	3.12%	M	1	3	16	1	4.76%	14.29%	76.19%	4.76%
N	68	60	50	1908	3.26%	2.88%	2.40%	91.47%	N	4	3	0	107	3.51%	2.63%	0.00%	93.86%

Figura . Entrenamiento de clasificador red GoogleNet usando datos aumentados. **Arriba:** gráficos de exactitud y pérdida. **Abajo:** Matriz de confusión para los datos de validación y prueba.

Para la red GoogleNet la exactitud global fue de 87.62%, siendo este el promedio más bajo de todos los modelos ensayados. Además mostró la menor tasa de aciertos para tres las cuatro clases. El mejor desempeño fue logrado al clasificar correctamente 107 neutrófilos segmentados (93.86% del total) mientras que el mayor error se presentó en la clasificación de eosinófilos logrando identificar solamente 13 de 22 de estas células de forma correcta (59.09% de acierto). La mayor confusión ocurrió nuevamente en la diferenciación entre eosinófilos y neutrófilos segmentados, donde un 36.36% de los primeros fueron asignados erradamente al segundo tipo.

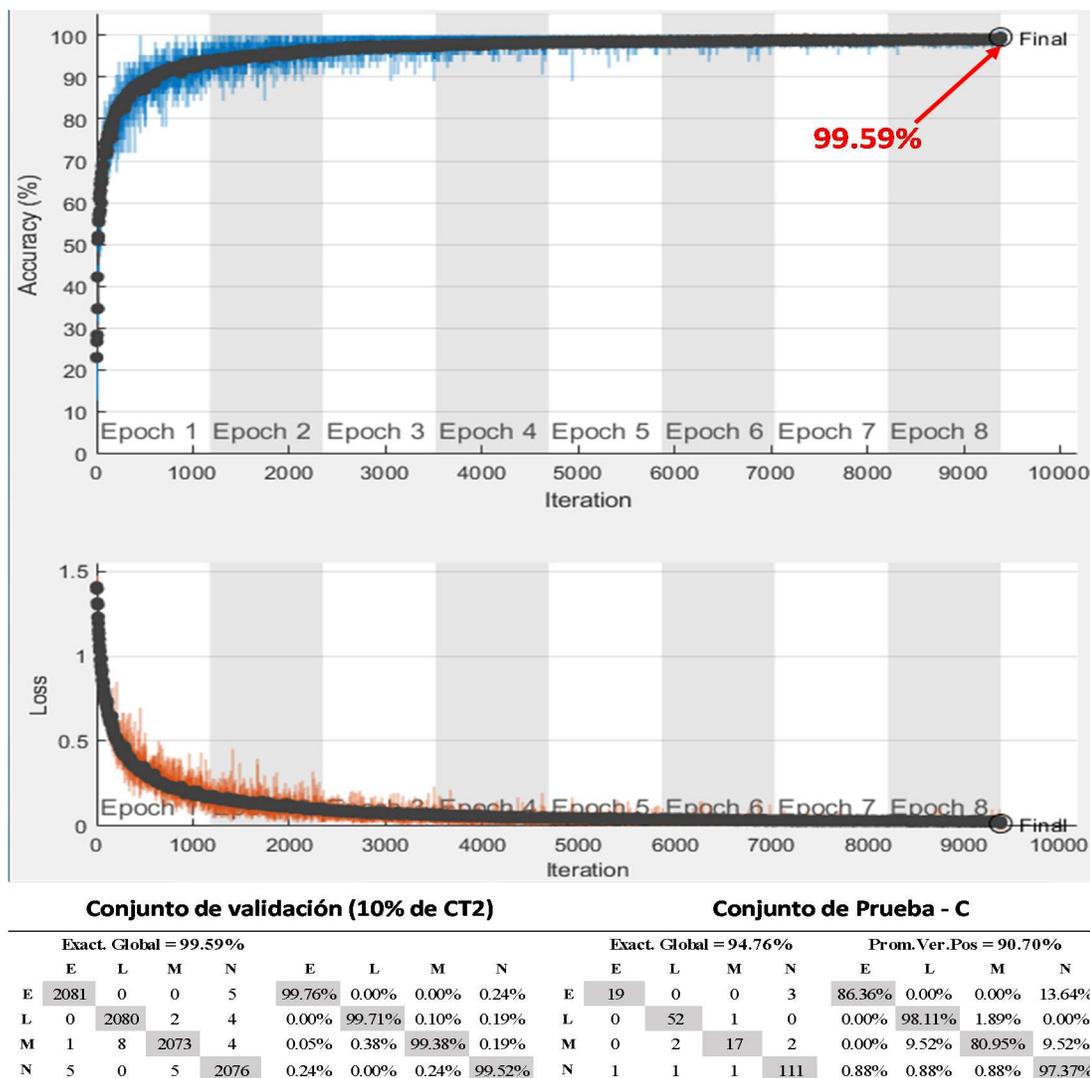


Figura . Entrenamiento de clasificador red ResNet-18 usando datos aumentados.

El clasificador basado en la red ResNet-18 mostró (junto a la red ResNet-50) la mayor exactitud global, alcanzando un 94.76% al identificar correctamente 199 de 210 leucocitos y la mejor tasa de aciertos en tres de los cuatro tipos de células. Su menor rendimiento se observó al clasificar los monocitos, acertando 17 de 21 (80.95%); mientras que, el mayor éxito fue alcanzado para los linfocitos con un 98.11% (52 de 53). La mayor confusión entre clases ocurrió al identificar 3 eosinófilos (13.64%) como segmentados. Esta red identificó correctamente 111 de 114 neutrófilos segmentados de forma correcta.

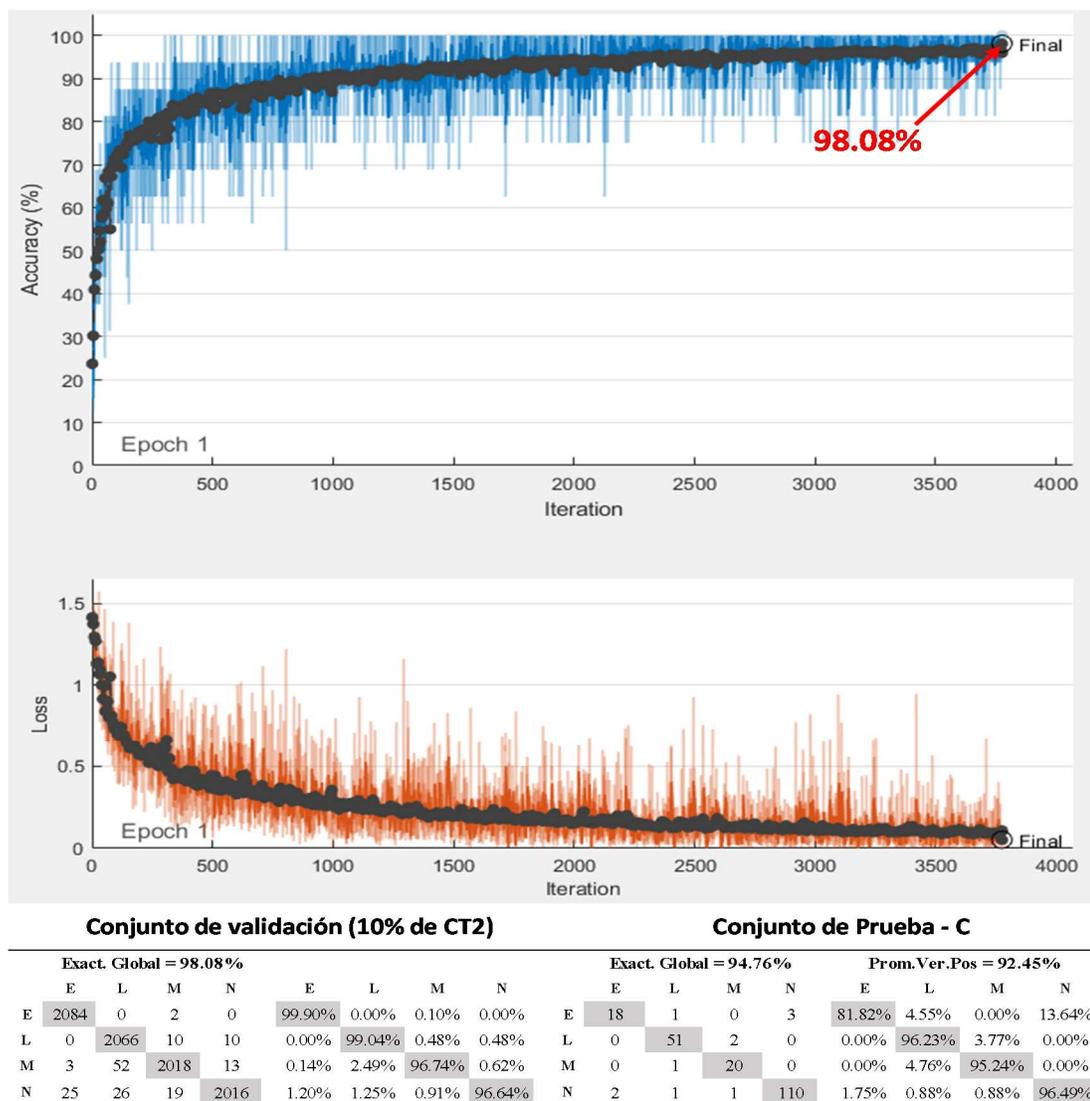
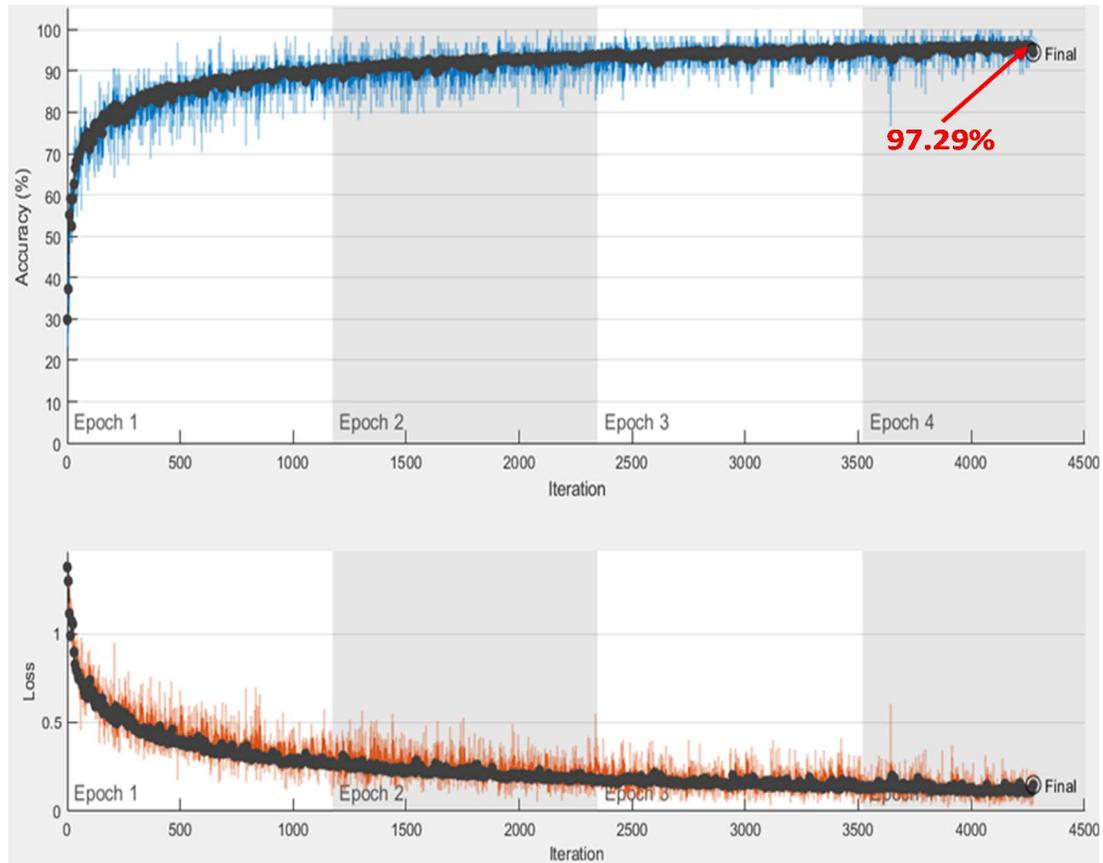


Figura . Entrenamiento de clasificador red ResNet-50 usando datos aumentados

Utilizando la red ResNet-50 se logró clasificar correctamente el 94.76%, siendo neutrófilos segmentados la categoría de mayor éxito (96.49%) y eosinófilos la de menor acierto (81.82%). En las métricas obtenidas por los diferentes clasificadores (ver Tabla en página 206), ResNet-50 logró los mejores valores en 10 de 11 indicadores, siendo superada únicamente en el valor de precisión en un 1.49% por la red ResNet-18. Por otro lado, el tiempo de clasificación por imagen fue el mayor de todos, superando en un factor de 3X a la red AlexNet y 2X a ResNet-18.



Conjunto de validación (10% de CT2)

Conjunto de Prueba - C

	Exact. Global = 97.29%								Exact. Global = 91.43%				Prom.Ver.Pos = 85.39%					
	E	L	M	N	E	L	M	N	E	L	M	N	E	L	M	N		
SqueezeNet	E	2061	4	4	17	98.80%	0.19%	0.19%	0.81%	E	17	0	0	5	77.27%	0.00%	0.00%	22.73%
	L	3	2004	40	39	0.14%	96.07%	1.92%	1.87%	L	0	52	1	0	0.00%	98.11%	1.89%	0.00%
	M	5	13	2026	42	0.24%	0.62%	97.12%	2.01%	M	0	3	15	3	0.00%	14.29%	71.43%	14.29%
	N	29	20	10	2027	1.39%	0.96%	0.48%	97.17%	N	4	1	1	108	3.51%	0.88%	0.88%	94.74%

Figura . Entrenamiento de clasificador red SqueezeNet usando datos aumentados.

Con la red SqueezeNet se consiguió resultados ligeramente inferiores a AlexNet pero significativamente mejores que con GoogleNet, incluso comparte con ResNet-18 la mayor tasa de aciertos de linfocitos (98.11%). Sin embargo, el 22.73% de los eosinófilos, y 14.29% de los monocitos fueron clasificados erradamente como neutrófilos segmentados. La tasa de falsos negativos resultó ser la segunda más desfavorable con un promedio de 14.61%. Para las clases linfocitos y neutrófilos segmentados la exactitud es elevada superando el 94%.

La Tabla presenta un resumen de la exactitud global alcanzada para los conjuntos de validación del entrenamiento (compuesto por 2086 imágenes) y de prueba (210 leucocitos), junto al tiempo promedio requerido para realizar la clasificación y el tiempo utilizado para entrenar la red. La Figura representa ésta información de forma gráfica, en ella se evidencia la superioridad de las redes ResNet-50 y ResNet-18 para realizar la clasificación de los leucocitos y la similitud en el rendimiento entre AlexNet y SqueezeNet. Inicialmente se esperaba que la red GoogleNet exhibiera una de las tasas de acierto más elevadas cosa que no ocurrió. Esto pudiera entenderse

Tabla

Exactitud de los clasificadores entrenados con imágenes aumentadas

Modelo de red	Tiempo entrenamiento (min)	Exactitud validación (%)	Exactitud prueba (%)	Tiempo clasificación (ms)
AlexNet	824	98.35	90.95	5.155
GoogleNet	4606	93.19	87.62	11.12
ResNet-18	3691	99.59	94.76	7.46
Resnet-50	1813	98.08	94.76	15.64
SqueezeNet	762	97.29	91.43	6.25

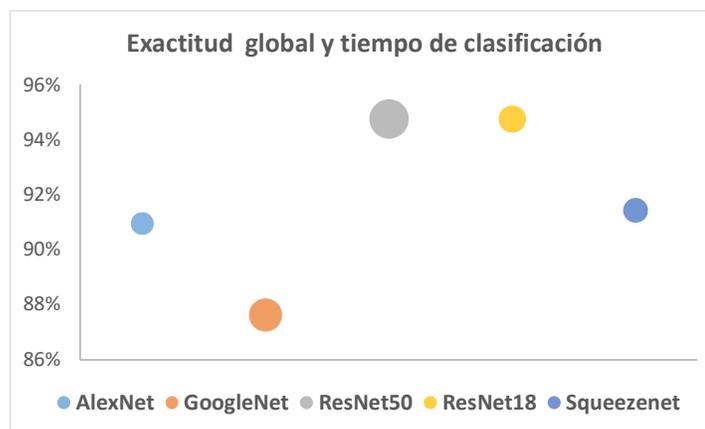


Figura . Comparación exactitud global en experimento #7. El radio de los círculos representa el tiempo relativo requerido para clasificar una imagen.

Al comparar los resultados obtenidos mediante la técnica de extracción de rasgos (Tabla y 21, página 177) y ajuste de pesos (Tabla y 34) se observa con facilidad una mejora significativa en la mayoría de los indicadores. Algunos valores resaltantes son los siguientes:

- Exactitud: +13.81% (ResNet-18), +10% (SqueezeNet) y +6.63% (ResNet-50).
- Precisión: +33.13% (ResNet-18), +23.31% (SqueezeNet) y +18.64% (ResNet-50).
- Sensitividad: +22.06 (ResNet-18), +10.81% (ResNet-50) y 9.26% (AlexNet).
- Especificidad: 18.49 (GoogleNet), +8.6% ResNet-18 y +4.99% (SqueezeNet).

Tabla
Métricas de evaluación para clasificadores del Experimento #7

Red-cap		Exactitud	Precisión	Sensibilidad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error	AUC-ROC
ALEXNET	E	0.9571	0.8095	0.7727	0.9787	0.0213	0.2273	0.0429	0.8696
	L	0.9571	0.9074	0.9245	0.9682	0.0318	0.0755	0.0429	0.9461
	M	0.9667	0.7500	1.0000	0.9630	0.0370	0.0000	0.0333	0.9813
	N	0.9381	0.9720	0.9123	0.9688	0.0313	0.0877	0.0619	0.9401
	Prom	0.9548	0.8597	0.9024	0.9696	0.0304	0.0976	0.0452	0.9343
	Desv.Std	0.0120	0.0990	0.0947	0.0066	0.0066	0.0947	0.0120	0.0468
GOOGLNET	E	0.9238	0.6500	0.5909	0.9628	0.0372	0.4091	0.0762	0.7543
	L	0.9476	0.8889	0.9057	0.9618	0.0382	0.0943	0.0524	0.9333
	M	0.9667	0.8889	0.7619	0.9894	0.0106	0.2381	0.0333	0.8682
	N	0.9143	0.9068	0.9386	0.8854	0.1146	0.0614	0.0857	0.9116
	Prom	0.9381	0.8336	0.7993	0.9498	0.0502	0.2007	0.0619	0.8669
	Desv.Std	0.0237	0.1227	0.1587	0.0448	0.0448	0.1587	0.0237	0.0798
RESNET18	E	0.9810	0.9500	0.8636	0.9947	0.0053	0.1364	0.0190	0.9268
	L	0.9810	0.9455	0.9811	0.9809	0.0191	0.0189	0.0190	0.9810
	M	0.9714	0.8947	0.8095	0.9894	0.0106	0.1905	0.0286	0.8949
	N	0.9619	0.9569	0.9737	0.9479	0.0521	0.0263	0.0381	0.9607
	Prom	0.9738	0.9368	0.9070	0.9782	0.0218	0.0930	0.0262	0.9409
	Desv.Std	0.0091	0.0284	0.0843	0.0210	0.0210	0.0843	0.0091	0.0379
RESNET50	E	0.9714	0.9000	0.8182	0.9894	0.0106	0.1818	0.0286	0.8997
	L	0.9762	0.9444	0.9623	0.9809	0.0191	0.0377	0.0238	0.9716
	M	0.9810	0.8696	0.9524	0.9841	0.0159	0.0476	0.0190	0.9681
	N	0.9667	0.9735	0.9649	0.9688	0.0313	0.0351	0.0333	0.9668
	Prom	0.9738	0.9219	0.9244	0.9808	0.0192	0.0756	0.0262	0.9516
	Desv.Std	0.0061	0.0461	0.0710	0.0087	0.0087	0.0710	0.0061	0.0346
SQUEEZENET	E	0.9571	0.8095	0.7727	0.9787	0.0213	0.2273	0.0429	0.8696
	L	0.9762	0.9286	0.9811	0.9745	0.0255	0.0189	0.0238	0.9778
	M	0.9619	0.8824	0.7143	0.9894	0.0106	0.2857	0.0381	0.8407
	N	0.9333	0.9310	0.9474	0.9167	0.0833	0.0526	0.0667	0.9319
	Prom	0.9571	0.8879	0.8539	0.9648	0.0352	0.1461	0.0429	0.9050
	Desv.Std	0.0178	0.0568	0.1304	0.0327	0.0327	0.1304	0.0178	0.0617

Tabla
Comparación de indicadores del Experimento #7

	Exactitud	Precisión	Sensitividad	Especificidad	Tasa Falsos Positivos	Tasa Falsos Negativos	Error	AUC-ROC	TP-A	G-Mean
ALEXNET	95.48%	85.97%	90.24%	96.96%	3.04%	9.76%	4.52%	93.60%	90.24%	93.54%
GOOGLNET	93.81%	83.36%	79.93%	94.98%	5.02%	20.07%	6.19%	87.46%	79.93%	87.13%
RESNET18	97.38%	93.68%	90.70%	97.82%	2.18%	9.30%	2.62%	94.26%	90.70%	94.19%
RESNET50	97.38%	92.19%	92.44%	98.08%	1.92%	7.56%	2.62%	95.26%	92.45%	95.22%
SQUEEZENET	95.71%	88.79%	85.39%	96.48%	3.52%	14.61%	4.29%	90.94%	85.39%	90.77%

4.4. INTERFAZ DE IDENTIFICACIÓN Y CONTEO DE LEUCOCITOS

Para facilitar la realización de pruebas combinando los algoritmos de segmentación y clasificación codificados, se procedió a crear una interfaz gráfica de fácil uso que presenta los elementos mostrados en la Figura . En su utilización deben seguirse en secuencia los siguientes pasos:

1. Hacer clic sobre el tipo de imagen que asemeje mejor al conjunto de datos que se desea evaluar.
2. Seleccionar la ubicación de las imágenes mediante el botón “Buscar Directorio de Imágenes”.
3. Elegir el tipo de clasificador a ser utilizado (ResNet-18 o ResNet-50, en función de los experimentos realizados).
4. Presionar el botón “Analizar” y esperar por los resultados.

En la parte inferior izquierda de la pantalla se presentan el conteo diferencial de leucocitos realizado por el sistema, indicando cuantas células de cada tipo fueron encontradas. Si las imágenes de origen se encuentran separadas de acuerdo al tipo de leucocitos (previamente clasificadas por un experto médico), el sistema calcula y muestra en la parte derecha de la pantalla la matriz de confusión y algunas métricas de rendimiento de interés como la sensibilidad, especificidad y exactitud entre otras.

Los archivos necesarios para la visualización de la interfaz gráfica se encuentran disponibles en el apéndice A-5 “Código Sistema Clasificación Leucocitos”. Para su ejecución se requiere copiar en el directorio de MATLAB todos los archivos contenidos en la carpeta “MAT_Scripts” de CD que acompaña a este trabajo.

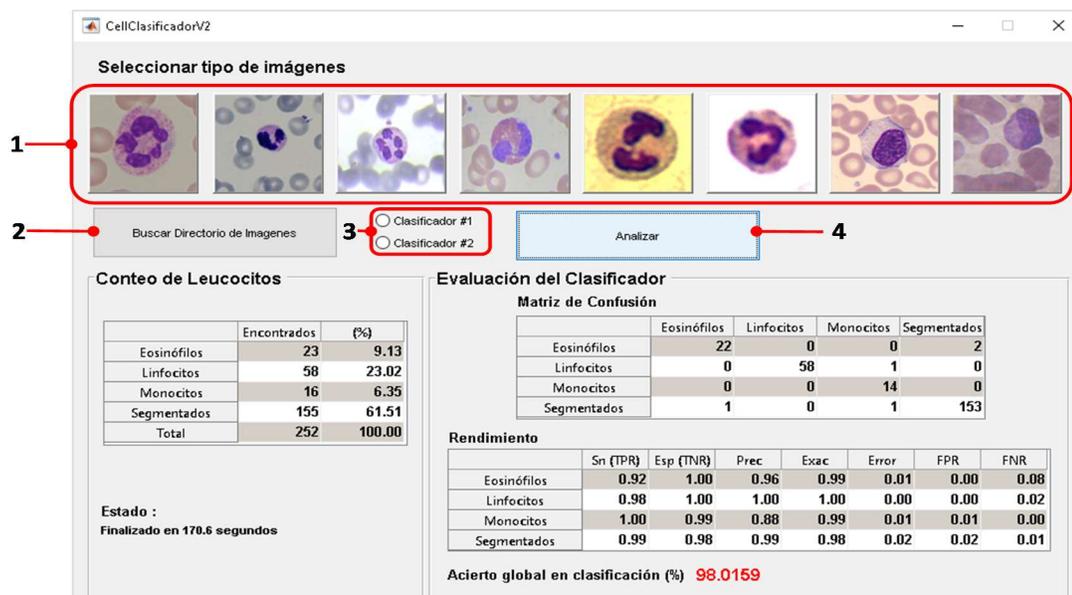


Figura . Interfaz gráfica del sistema de análisis automático para la segmentación, clasificación y conteo de leucocitos en imágenes digitales de frotis de sangre. En la parte superior se encuentran los botones de acción mientras que en la parte inferior se presentan los resultados obtenidos por el sistema.

4.5. EVALUACIÓN DEL SISTEMA Y PRUEBAS FINALES

Como pruebas finales de operación del sistema propuesto se realizaron dos ensayos destinados a medir: (a) El funcionamiento general del sistema mediante la clasificación de las imágenes originales de todas las bases de datos y (b) la clasificación del banco de imágenes de múltiples células. Los resultados de ambos se presentan y discuten en la presente sección.

Estos ensayos fueron realizados utilizando el clasificador basado en la red convolucional ResNet-18 (Clasificador #1), ya que el mismo tiene un costo computacional menor debido al número de parámetros del modelo (utilizando menos memoria) y permite evaluar las imágenes más rápidamente. De manera adicional en el anexo “C” se muestran los resultados obtenidos al usar el Clasificador #2, basado en la red ResNet-50 operando bajo las mismas condiciones. Durante la ejecución estas pruebas se empleó un computador con sistema operativo Windows 10®, equipado con procesador Intel® Core™ i5 3230M-2.60 GHz, 4GB de memoria RAM DD3 y tarjeta de video integrada Intel® HD Graphics 4000.

4.5.1. Funcionamiento general del sistema

Este ensayo consistió en hacer funcionar en conjunto los algoritmos de extracción y clasificación implementados, usando las imágenes previamente pre-procesadas manualmente de todas las bases de datos disponibles, con el fin de observar el rendimiento obtenido y estimar el tiempo necesario para realizar el conteo diferencial de leucocitos.

4.5.1.1. Clasificación del conjunto ALL-IDB1

Como se observa en la Figura , un 98.08% de las imágenes de este conjunto de datos fue clasificado de forma correcta. El mayor error se presentó al clasificar dos neutrófilos segmentados como eosinófilos. La exactitud lograda sobre esta base de datos es superior tanto al 88% alcanzado por Nagabhushana y Rajeshwari (2014) como al 90% reportado por Bhavnani y otros (2016). El tiempo requerido para realizar el análisis de las 104 imágenes fue de 126.32 segundos, promediando 1.21 segundos por célula.



Figura . Resultado clasificación imágenes ALL-IDB1

4.5.1.2. Clasificación del conjunto ALL-IDB2

Todas las células de este grupo de imágenes fueron clasificadas correctamente según puede observarse en la Figura . La clasificación de las 97 células presentes se realizó en 43.63 segundos, con un promedio de 0.45 segundos por célula.



Figura . Resultado clasificación imágenes ALL-IDB2

4.5.1.3. Clasificación del conjunto IUMS-IDB

Hasta un 98.02% de las 252 células presentes en este conjunto de imágenes fueron clasificadas de manera acertada en un tiempo de 151.15 segundos (0.6 segundos por célula) como se muestra en la Figura . En total, solamente 5 células de 252 fueron identificadas de forma incorrecta



Figura . Resultado clasificación imágenes IUMS-IDB

El mayor error se registró en la clasificación errada de dos eosinófilos los cuales fueron identificados como segmentados. Es de hacer notar que incluso el experto humano presentó dudas al momento de catalogar algunas imágenes dentro del conjunto, manifestando que la calidad de la coloración en varias fotografías no permitía observar con claridad la cromatina del citoplasma. Pese a esto, al comparar con los resultados presentados por Sarrafzadeh y otros (2014) se ha logrado alcanzar una mayor exactitud (98.02% vs 97.73%) y especificidad promedio (99.25% vs 95.95%) aunque una sensibilidad ligeramente inferior (97.25% vs 98.25%).

4.5.1.4. Clasificación del conjunto SMC-IDB

La Figura presenta los resultados obtenidos al clasificar las 375 imágenes de este conjunto. Un 99.47% de los leucocitos fueron reconocidos de manera correcta al fallar la identificación de solo dos de ellos. Al igual que ocurrió con las imágenes del conjunto ALL-IDB2, estas son de gran calidad y exhiben una buena tinción que permite apreciar con claridad las características de las células. Ello explica que el 100% de los eosinófilos y monocitos presentes fuesen clasificados de forma exitosa y que solamente errara la identificación de uno entre 262 neutrófilos segmentados. El tiempo de procesamiento total fue de 366.08 segundos, promediando 0.97 segundos por célula.



Figura . Resultado clasificación imágenes SMC-IDB

4.5.1.5. Clasificación del conjunto LISC-Database

Este conjunto de imágenes presenta gran calidad al igual que el anterior, lo cual permitió alcanzar un 99.52% de exactitud global. Únicamente un eosinófilo fue clasificado equivocadamente como monocito (ver Figura). Este valor de exactitud fue mayor a los reportados tanto por Rezatofighi & Soltanian-Zadeh (2011) (usando máquinas de soporte vectorial) como por Shahin y otros (2019) (basado en redes neuronales convolucionales) al trabajar sobre el mismo conjunto de datos. El sistema demoró en promedio 0.76 segundos para identificar cada una de las 207 células presentes.



Figura . Resultado clasificación imágenes LISC

4.5.1.6. Clasificación de DataSet1 base de datos WBC-ID

Ambos subconjuntos del DataSet1 (JTSC-I y JTSC-II) presentan imágenes bastante diferentes a las del resto de las bases de datos debido al tipo de coloración empleado. Sin embargo, el sistema ha aprendido a clasificar las células presentes exitosamente alcanzando respectivamente un 99.49% y 100% de exactitud, como se observa en la Figura . En total se han clasificado correctamente 296 de 297 leucocitos. El menor tiempo de procesamiento en estas imágenes (0.3 y 0.27 segundos para cada subconjunto) se debe a su menor resolución.

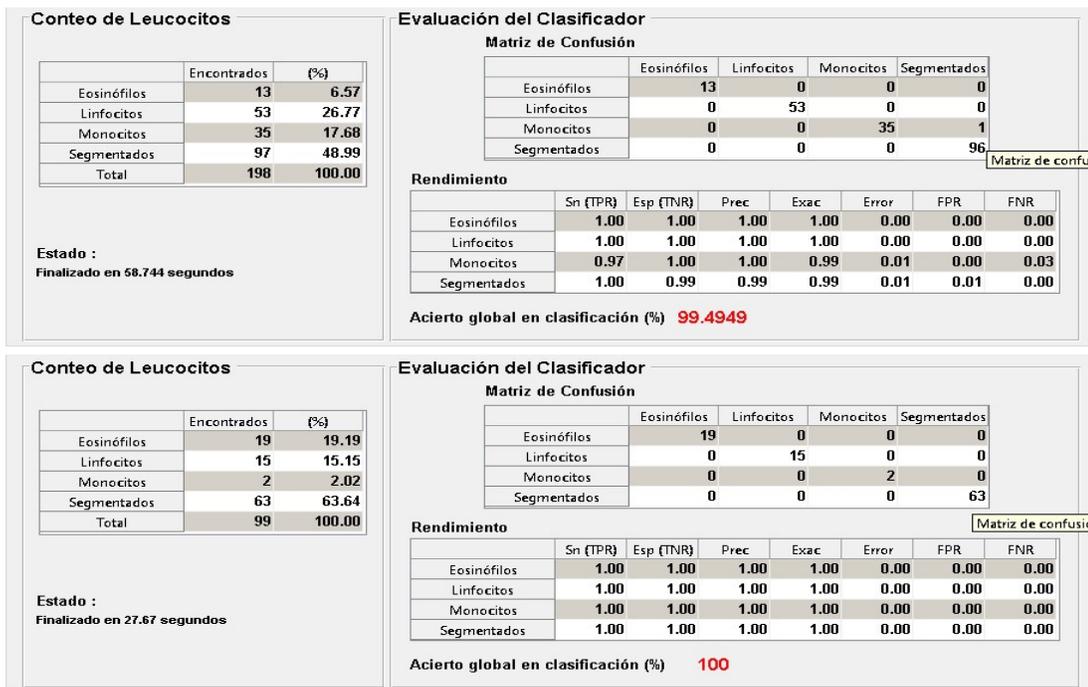


Figura . Resultado clasificación imágenes WBC-ID. Arriba: JTSC-I. Abajo: JTSC-II

4.5.1.7. Clasificación de DataSet2 base de datos WBC-ID

Por su parte con DataSet2 se logró un 97.92% de exactitud al acertar en 94 de 96 leucocitos (ver Figura). El sistema falló únicamente al asignar 2 monocitos como linfocitos. El tiempo de procesamiento total en este caso fue de 80.64 segundos, demorando 0.84 segundos en cada imagen.



Figura . Resultado clasificación imágenes WBC-ID (DataSet2)

4.5.1.8. Clasificación del conjunto BDL

Al evaluar este conjunto de imágenes, se observó que el sistema detectó 545 posibles células cuando en realidad existían 527, como se muestra en la Figura . Ello indica que el algoritmo de extracción identificó áreas como leucocitos cuando en realidad no lo eran, incurriendo en un 3.42% de falsa detección. En algunos de estos casos se logró identificar una relación con la predicción de pertenencia arrojada por el clasificador, permitiendo definir criterios de clasificación a ser utilizados como valores umbrales y rechazar las muestras que no los satisfagan. De manera empírica se establecieron los criterios mostrados en la Tabla .



Figura . Resultado clasificación imágenes BDL

Tabla

Criterios empíricos de clasificación para aceptar o rechazar muestras del conjunto BDL

	Eosinófilos	Linfocitos	Monocitos	N. Segmentados
Valor umbral	0.7	0.92	0.98	0.9
Imágenes rechazadas	1	5	8	4

Luego de aplicar estos criterios se obtuvo el resultado mostrado en la Figura . Se logró reducir la falsa detección de leucocitos, aumentar la exactitud global, la cual pasó de 96.88% a 98.87% y mejorar tanto la especificidad como sensibilidad en la respuesta emitida por el sistema. Este resultado indica que aun pese a tratarse de imágenes de baja calidad, con ruido y desenfocadas, el sistema ha logrado localizar e identificar de manera correcta la mayoría de los leucocitos presentes, demostrando gran robustez al realizar dicha labor.



Figura . Resultado clasificación imágenes BDL usando criterios de rechazo de la Tabla .

4.5.2. Resultados con imágenes crudas de múltiples células

En esta prueba se utilizaron imágenes sin pre-procesamiento alguno, es decir justo como fueron capturadas o publicadas por sus autores. Buscó comparar la exactitud obtenida por el sistema automático de clasificación de leucocitos en función del criterio de su contraparte humana. El experto humano clasificó las células presentes banco de imágenes según se indica en la Tabla . Posteriormente, los mismos datos fueron evaluados por el sistema propuesto obteniendo los resultados que se muestran en la Figura . En todos los casos, el sistema computacional arrojó resultados similares los

indicados por el experto médico mostrando diferencias máximas comprendidas entre - 2.84% y 3.08%.

Tabla

Clasificación de las imágenes y conteo diferencial realizado por experto humano

Conjunto de imágenes	Clasificación imágenes				Total	Conteo diferencial de leucocitos			
	Eos.	Linf.	Mono.	N. Seg.		Eos.	Linf.	Mono.	N. Seg.
M-ALL-IDB	3	39	7	23	72	4.17%	54.17%	9.72%	31.94%
M-LISC	8	66	17	82	173	4.62%	38.15%	9.83%	47.40%
M-IUMS	13	23	6	67	109	11.93%	21.10%	5.50%	61.47%
M-BLD	0	23	0	34	57	0.0%	40.35%	0.0%	59.65%

M-ALL-IDB			M-LISC		
	Encontrados	(%)		Encontrados	(%)
Eosinófilos	2	2.78	Eosinófilos	9	5.49
Linfocitos	38	52.78	Linfocitos	65	39.63
Monocitos	7	9.72	Monocitos	16	9.76
Segmentados	25	34.72	Segmentados	74	45.12
Total	72	100.00	Total	164	100.00

M-IUMS			M-BDL		
	Encontrados	(%)		Encontrados	(%)
Eosinófilos	10	9.09	Eosinófilos	0	0.00
Linfocitos	24	21.82	Linfocitos	19	39.58
Monocitos	5	4.55	Monocitos	1	2.08
Segmentados	71	64.55	Segmentados	28	58.33
Total	110	100.00	Total	48	100.00

Figura . Conteo de leucocitos realizado por el sistema automático propuesto

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

En el presente trabajo de investigación se han combinado principios, métodos y tecnologías existentes para el desarrollo de un sistema automático capaz de realizar la identificación, clasificación y conteo de leucocitos en imágenes digitales de frotis de sangre periférica. Este sistema constituye una herramienta capaz de promover la automatización de los laboratorios clínicos, al permitir ejecutar un examen de rutina en forma rápida y eficiente, sin la adquisición de equipos especializados. Para lograr esto se incorporaron, en la mejor medida posible, algoritmos y modelos de bajo costo computacional, los cuales, permiten su ejecución en equipos de prestaciones limitadas y sin requerimientos específicos de hardware, maximizando así los usuarios potenciales.

Mediante la revisión bibliográfica se pudo comprobar que numerosas técnicas de segmentación de imágenes han sido aplicadas en la identificación de las células presentes en la sangre humana y aquellas que se basan de métodos de reconocimiento de patrones han producidos mejores resultados. De esta forma, la incorporación del aprendizaje automático como parte de los algoritmos de segmentación ha permitido alcanzar resultados y superar limitaciones encontradas con las técnicas tradicionales, siendo dominante el uso del paradigma de entrenamiento supervisado debido a su menor costo computacional de implementación.

Por este motivo, la segmentación y posterior extracción de los leucocitos se realizó por medio de una máquina de soporte vectorial (SVM) capaz de identificar diferentes elementos que pueden encontrarse en la imagen del frotis de sangre periférica, tales como el núcleo y citoplasma de los leucocitos, plasma u otras células. La selección y uso de una SVM como bloque constructivo del algoritmo de segmentación permitió, luego de un breve entrenamiento, lograr elevadas tasas de clasificación de los píxeles

en la identificación de las regiones en la imagen, mostrando robustez y gran capacidad de adaptación del método a bancos de datos de características no homogéneas.

La utilización de un algoritmo eficiente de generación de superpíxeles permitió reducir notablemente el tiempo requerido para la identificación y extracción de las células presentes en las imágenes. Siendo la reducción del tiempo necesario para la realización del conteo diferencial de leucocitos una de las motivaciones que impulsaron la ejecución del presente trabajo, esto tuvo un profundo impacto sobre la evaluación del funcionamiento del sistema propuesto. El tiempo promedio requerido para extraer y clasificar un leucocito fue de 0.6662 segundos, en comparación con métodos como los usados por Rezaatoughi y Soltanian-Zadeh (2011) o Di Ruberto y otros (2016) han requerido, respectivamente, tiempos de procesamiento de 9.6 y 8.6 segundos por célula. Este resultado permitió contar 100 leucocitos en aproximadamente 66.63 segundos, lo cual es comparable en rendimiento a las tecnologías de automatización comerciales dominantes permitiendo concluir que la propuesta desarrollada puede competir con dichas alternativas.

La creación de un banco de datos utilizando imágenes adquiridas bajo condiciones muy diferentes con grandes variantes de iluminación, contraste, coloración y resolución, permitió disponer del volumen de información suficiente para plantear el uso de técnicas de procesamiento basadas en el aprendizaje profundo como las redes neuronales convolucionales. Estos datos experimentales incluyeron imágenes publicadas por distintos grupos de investigación a nivel mundial así como otras adquiridas durante el desarrollo del proyecto adaptando un equipo de fotografía a un microscopio estándar. Para incrementar la cantidad de datos de entrenamiento se utilizaron técnicas de aumento artificial, mediante operaciones de ajuste de intensidad y transformaciones geométricas, hasta disponer de más de 120.000 imágenes. Esto permitió disponer de suficientes ejemplos para entrenar los modelos de las redes convolucionales sin sobreajuste.

Los algoritmos modernos de inteligencia artificial fundamentados en el aprendizaje profundo pueden caracterizar las imágenes a partir de los datos sin procesar y obtener una clasificación similar o superior a otras técnicas que utilizan vectores de rasgos cuidadosamente seleccionados por personal especializado, ello permite que puedan ser utilizados para encontrar soluciones efectivas a problemas complejos en corto tiempo. Por ello, se observó que al usar una red convolucional pre-entrenada como extractor de rasgos junto a un clasificador implementado mediante una máquina de vectores de soporte fue posible alcanzar, en algunos casos, valores de exactitud cercanos al 90% de acierto.

Al experimentar utilizando este enfoque, se comprobó que las capas más internas de la red neuronal permitieron alcanzar mejores valores en métricas como la exactitud global, sensibilidad, sensibilidad, precisión y especificidad, entre otros. Esto es atribuible debido a las marcadas diferencias entre los datos en entrenamiento del modelo pre-entrenado (imágenes naturales) y la nueva tarea de clasificación que implica usar imágenes de microscopio.

Al clasificar los leucocitos usando estos rasgos genéricos (ya que corresponden al modelo pre-entrenado que no ha sido ajustado con los propios datos que se espera clasificar) se observó que los mejores resultados se obtienen al usar conjuntos experimentales balanceados en la cantidad de muestras pertenecientes a cada clase. La estrategia de composición del clasificador en este caso tuvo poca influencia, siendo los resultados ligeramente superiores al usar el esquema 1 vs todos (OVA). También se comprobó que al usar la red convolucional como extractor de características junto al clasificador SVM, la incorporación de un mayor conjunto de datos de entrenamiento con datos artificialmente aumentados no aporta beneficios, toda vez que es posible ajustar bien al clasificador usando un conjunto de entrenamiento pequeño.

Se determinó que la transferencia de aprendizaje mediante el método de ajuste fino en los pesos sinápticos de la red convolucional, era más efectivo, en términos de exactitud, para adaptar modelos pre-entrenados a nuevos problemas de clasificación, que el método de simple de extracción de características. Como evidencia de ello para los cinco modelos de red en los cuales se realizó el ajuste fino la exactitud de clasificación promedio en los conjuntos de validación y prueba alcanzó un 97.3% y 91.90%, respectivamente.

La mayor exactitud de clasificación fue lograda utilizando redes neuronales residuales, cuyos valores máximos de aciertos se ubicaron en 99.54% y 94.76% para los datos de validación y prueba. Esto comprueba la superioridad de éstas frente a las otras arquitecturas de red evaluadas.

Por medio de la creación de una interfaz gráfica de usuario se unificó la operación de los algoritmos de extracción y clasificación de imágenes para finalmente integrar el sistema de conteo de leucocitos. Esta permitió utilizar el sistema y realizar las pruebas operativas finales así como la medición del tiempo requerido para el procesamiento.

La solución planteada demostró una elevada capacidad para clasificar correctamente imágenes provenientes de diferentes bases de datos al lograr una exactitud promedio de 99.03% en la identificación de las células, lo cual supera el estado del arte actual del área. El sistema también mostró buena capacidad de generalización como quedó evidenciado al lograr extraer un 95.86% de las células presentes en un banco adicional de imágenes de validación y clasificarlas con un margen de error en torno al 3% respecto al criterio de un observador humano experto.

En el mejor conocimiento del autor de este trabajo no existen a la fecha trabajos que involucren la combinación de las técnicas presentadas en esta investigación aplicados específicamente a la identificación y conteo de leucocitos.

RECOMENDACIONES

Una vez culminada la presente investigación, se plantea un conjunto de recomendaciones específicas derivadas de los resultados obtenidos y orientadas a enmarcar futuros trabajos:

- El error en la extracción de glóbulos blancos en imágenes con múltiples células presentes fue del 4.14% e incluyó aquellos casos en los cuales los leucocitos se encontraban unidos, situación no considerada en el algoritmo de segmentación implementado. Se recomienda incorporar las modificaciones pertinentes en el algoritmo de extracción de células a fin de manejar dicho caso como medio para incrementar la exactitud.
- Desarrollar a partir de los modelos de red entrenados durante el presente trabajo, otros sistemas que permitan la identificación de basófilos y/o clasificación de leucocitos inmaduros que puedan ser empleados en el diagnóstico de ciertas patologías.
- Ampliar el uso de las redes convolucionales profundas dentro del sistema al incorporar algoritmos de reconocimiento y localización de objetos tales como Faster R-CNN o YOLO, esto permitiría hacer más rápido al sistema.
- Exportar el algoritmo de la solución propuesta a una plataforma de código abierto para permitir su utilización en diferentes arquitecturas computacionales.

REFERENCIAS BIBLIOGRÁFICAS

1. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Susstrunk, S. (2010). *SLIC Superpixels*. EPFL Technical Report 149300.
2. Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Susstrunk, S. (2012). SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11), 2274-2281. doi:10.1109/TPAMI.2012.120
3. Arias, F. G. (2006). *El Proyecto de Investigación: Introducción a la metodología científica* (Sexta ed.). Caracas: EPISTEME.
4. Bäckström, N. (2017). *Designing a Lightweight Convolutional Neural Network for Onion and Weed Classification*. Trabajo de grado, Uppsala Universit.
5. Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Transactions on Software Engineering*, 35(8), 1798-1828.
6. Bhavnani, L., Udesang, J., & Mahasweta, J. (2016). Segmentation and Counting of WBCs and RBCs from Microscopic Blood Sample Images. *International Journal of Image, Graphics and Signal Processing*, 8(11), 32-40. doi:10.5815/ijigsp.2016.11.05
7. Byun, H., & Lee, S.-W. (2002). A Survey on Pattern Recognition Applications of Support Vector Machines. *Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines* (págs. 213-236). Londres: Springer-Verlag.
8. Canziani, A., Paszke, A., & Culurciello, E. (2017). An Analysis of Deep Neural Network Models for Practical Applications. *CoRR*, abs/1605.07678.
9. Carmona, S. E. (2016). *Tutorial sobre Máquinas de Vectores Soporte (SVM)*. Reporte Técnico, Universidad Nacional de Educación a Distancia (UNED), Dpto. de Inteligencia Artificial, Madrid.
10. Ciesla, B. (2014). *Hematología en la Práctica* (Segunda ed.). Philadelphia: AMOLCA.
11. Comaniciu, D., & Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 603-619. doi:10.1109/34.1000236
12. Di Ruberto, C., Loddo, A., Putzu, L. (2016). A leucocytes count system from blood smear images: Segmentation and counting of white blood cells based on learning by sampling. *Machine Vision and Applications*, 27(8), 1151-1160.
13. Egmont-Petersen, M., de Ridder, D., & Handels, H. (2002). Image processing with neural networks - a review. *Pattern Recognition*, 35(10), 2279-2301.

14. Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2), 167-181.
15. Forsyth, D. A., & Ponce, J. (2011). *Computer Vision: A Modern Approach* (Segunda ed.). New Jersey, USA: Pearson.
16. Frenkel, E. P. (2003). Biología de la sangre. En M. H. Beers, A. J. Fletcher, T. V. Jones, R. Porter, M. Berkwits, & J. L. Kaplan (Edits.), *Manual Merck de información médica general* (Vol. 2, págs. 1170-1172). Barcelona, España: Oceano.
17. Fukushima, K. (1980). Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*, 193-202.
18. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., & Herrera, F. (2011). An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. *Pattern Recognition*, 1761-1776.
19. Gao, Y., & Mosalam, K. (2018). Deep Transfer Learning for Image-Based Structural Damage Recognition. *Computer-Aided Civil and Infrastructure Engineering*, 33(9), 1-21.
20. García-Ordás, M. T., Alaiz-Rodríguez, R., & Alegre, E. (2016). Clasificación y reconocimiento de patrones. En E. Alegre, G. Pajares, & A. de la Escalera (Edits.), *Conceptos y métodos en visión por computador* (págs. 159-179). Madrid, España: Grupo de Visión del Comité Español de Automática.
21. Gonzalez, R. C., & Woods, R. E. (2002). *Digital Image Processing* (Segunda ed.). New Jersey, USA: Prentice Hall.
22. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. Obtenido de <http://www.deeplearningbook.org>
23. Grundland, M., & Dodgson, N. A. (2004). Color Histogram Specification by Histogram Warping. *Proceedings of SPIE Vol. 5667*, (págs. 610-624).
24. Gurcan, M. N., Boucheron, L. E., Madabhushi, A. C., Rajpoot, N. M., & Yener, B. (2009). Histopathological Image Analysis: A Review. *IEEE Reviews in Biomedical Engineering*, 2, 147 - 171. doi:10.1109/RBME.2009.2034865
25. Habibzadeh, M., Krzyzak, A., & Fevens, T. (2012). Analysis of white blood cell differential counts using dual-tree complex wavelet transform and support vector machine classifier. En R. T. L. Bolc (Ed.), *Proceedings of the International Conference on Computer Vision and Graphics ICCVG 2012*, (págs. 414-422). Varsovia, Polonia.
26. Habibzadeh, M., Krzyzak, A., & Fevens, T. (2013). White Blood Cell Differential Counts Using Convolutional Neural Networks for Low Resolution Images. En L.

- Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, & J. M. Zurada (Ed.), *Artificial Intelligence and Soft Computing. ICAISC 2013. Lecture Notes in Computer Science*. 7895, págs. 263-274. Berlin: Springer. doi:10.1007/978-3-642-38610-7_25
27. Hadhrami, E. A., Mufti, M. A., Taha, B., & Werghi, N. (2018). Transfer learning with convolutional neural networks for moving target classification with micro-Doppler radar spectrograms. *Proceedings of the 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, (págs. 148-154). Chengdu - China. doi:10.1109/ICAIBD.2018.8396184
 28. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (págs. 770-778). Las Vegas - USA: IEEE. doi:10.1109/CVPR.2016.90
 29. Hernández, R., Fernández, C., & Baptista, M. d. (2014). *Metodología de la investigación* (Sexta ed.). Ciudad de México: McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V.
 30. Herrera, P. J., Guijarro, M., & Guerrero, J. M. (2016). Operaciones de transformación de imágenes. En E. Alegre, G. Pajares, & A. de la Escalera (Edits.), *Conceptos y métodos en visión por computador* (págs. 61-76). Madrid, España: Grupo de Visión del Comité Español de Automática.
 31. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*.
 32. Hussain, M., Bird, J. J., & Faria, D. R. (2018). A Study on CNN Transfer Learning for Image Classification. En A. Lotfi, H. Bouchachia, A. Gegov, C. Langensiepen, & M. McGinnity (Ed.), *UKCI 2018. Advances in Intelligent Systems and Computing*. 840, págs. 191-202. Springer, Cham. doi:10.1007/978-3-319-97982-3_16
 33. Iandola, F., Han, S., W. Moskewicz, M., Ashraf, K., Dally, W., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. *CoRR, abs/1602.07360*.
 34. Jiang, M., Cheng, L., Qin, F., Du, L., & Zhang, M. (2018). White Blood Cells Classification with Deep Convolutional Neural Networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 32(9). doi:10.1142/S0218001418570069
 35. Kim, A. H., Lee, W., Kim, M., Kim, Y., & Han, K. (2014). White blood cell differential counts in severely leukopenic samples: a comparative analysis of different solutions available in modern laboratory hematology. *Blood research*, 49(2), 120-126. doi:doi:10.5045/br.2014.49.2.120

36. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems*, (págs. 1097–1105). Lake Tahoe - USA.
37. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 436-444. doi:doi:10.1038/nature14539
38. LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, E., Hubbard, W., & Jackel, L. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*(1), 541-551.
39. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278 - 2324. doi:10.1109/5.726791
40. Li, F.-F., Johnson, J., & Yeung, S. (2017). *Training Neural Networks, Part 2*. Obtenido de CS231n: Convolutional Neural Networks for Visual Recognition: http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture7.pdf
41. Lin, M., Chen, Q., & Yan, S. (2013). Network In Network. doi:<https://arxiv.org/abs/1312.4400v3>
42. Lisicki, M. (2014). *A deep learning computer vision system for image based cytometry*. Tesis de Maestría en Computer Science, Halifax, Nova Scotia.
43. Liu, D., Soran, B., Petrie, G., & Shapiro, L. (2011). *A Review of Computer Vision Segmentation Algorithms*. University of Washington.
44. Liu, Z., Liu, J., Xiao, X., Yuan, H., Li, X., Chang, J., & Zheng, C. (2015). Segmentation of White Blood Cells through Nucleus Mark Watershed Operations and Mean Shift Clustering. *Sensors*, 22561-86. doi:10.3390/s150922561
45. Martín del Brío, B., & Sanz, M. A. (2002). *Redes Neuronales y Sistemas Difusos* (Segunda ed.). México: Alfaomega.
46. Martín, D., García, F., & Armingol, J. (2016). El color: Modelos y transformaciones de los espacios de color. En E. Alegre, G. Pajares, & A. de la Escalera (Edits.), *Conceptos y métodos en visión por computador* (págs. 47-60). Madrid, España: Grupo de Visión del Comité Español de Automática.
47. Martínez-Castro, J., Reyes-Cadena, S., & Felipe-Riverón, E. (2014). Leukocytes Detection, Classification and Counting in Smears of Peripheral Blood. *Revista Mexicana de Ingeniería Biomédica*, 35(1), 41-51.
48. Mathur, A., Tripathi, A. S., & Kuse, M. (2013). Scalable system for classification of white blood cells from Leishman stained blood stain images. *J Pathol Inform*, 4(15).
49. McDonald, G. A., Paul, J., & Cruickshank, B. (2001). *Atlas de hematología* (Quinta ed.). Madrid: Editorial médica panamericana.

50. Mohamed, M., Far, B., & Guaily, A. (2012). An efficient technique for white blood cells nuclei automatic segmentation. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, (págs. 220-225). Seul-Corea del Sur. doi:10.1109/ICSMC.2012.6377703
51. Nagabhushana, R., & Rajeshwari, P. (Junio de 2014). Geometrical Analysis of Leukocyte Nucleus to Detect Lymphoblast from Microscopic Blood Images. *International Journal of Technology and Engineering Science*, 2(6), 1947-1951.
52. Nagpal, S., Singh, M., Vatsa, M., Singh, & Richa. (2018). Deep Learning: Fundamentals and Beyond. En M. Vatsa, R. Singh, & A. Majumdar (Edits.), *Deep Learning in Biometrics* (págs. 1-31). Boca Raton, Florida USA: CRC Press.
53. Nwankpa, C. E., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation Functions: Comparison of Trends in Practice and Research for Deep Learning. *arXiv:1811.03378v1*.
54. Oliveira, M., Chatbri, H., Little, S., O'Connor, N. E., & Sutherland, A. (2017). A comparison between end-to-end approaches and feature extraction based approaches for Sign Language recognition. *2017 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, (págs. 1-6). Christchurch - Nueva Zelanda. doi:10.1109/IVCNZ.2017.8402478
55. Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62-66. doi:10.1109/TSMC.1979.4310076
56. Özkaya, U., & Seyfi, L. (2018). Fine-Tuning Models Comparisons on Garbage Classification for Recyclability. *ISAS2018-Winter. 2nd International Symposium on Innovative Approaches in Scientific Studies*, (págs. 514-517). Sansun -Turquía.
57. Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 2(10), 1345-1359.
58. Patterson, J., & Gibson, A. (2017). *Deep learning: A practitioner's approach*. Sebastopol, USA: O'Reilly Media.
59. Peng, B., Zhang, L., & Zhang, D. (2013). A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, 1020–1038.
60. Pratt, W. K. (2001). *Digital Image Processing*. Los Altos: JOHN WILEY & SONS, INC.
61. Prinyakupt, J., & Pluempitiwiriyawej, C. (2015). Segmentation of white blood cells and comparison of cell morphology by linear and naïve Bayes classifiers. *Biomedical Engineer Online*, 14(63). doi:DOI 10.1186/s12938-015-0037-1
62. Putzu, L., & Di Ruberto, C. (2013). White Blood Cells Identification and Counting from Microscopic Blood Image. *International Journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering*, 7(1), 20-27.

63. Quintero, C., Merchán, F., Cornejo, A., & Sánchez Galán, J. (2018). Uso de Redes Neuronales Convolucionales para el Reconocimiento Automático de Imágenes de Macroinvertebrados para el Biomonitorio Participativo. *6th Engineering, Science and Technology Conference - Panama 2017 (ESTEC 2017)* (págs. 585-596). Ciudad de Panamá - Panamá: KnE Engineering. doi:10.18502/keg.v3i1.1462
64. Rajeshwar, D., Priyanka, & Swapna, D. (2012). Image Segmentation Techniques. *International Journal of Electronics & Communication Technology*, 66-70.
65. Ramirez-Cortes, J., Gomez-Gil, P., Alarcon-Aquino, V., Gonzalez-Bernal, J., & Garcia-Pedrero, A. (2010). Neural Networks and SVM-Based Classification of Leukocytes Using the Morphological Pattern Spectrum. (P. Melin, J. Kacprzyk, & W. Pedrycz, Edits.) *Soft Computing for Recognition Based on Biometrics. Studies in Computational Intelligence*, 312, 19-35. doi:10.1007/978-3-642-15111-8_2
66. Ramoser, H., Laurain, V., Bischof, H., & Ecker, R. (2005). Leukocyte segmentation and classification in blood-smear images. *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, (págs. 3371-3374). Shanghai.
67. Rawat, W., & Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, 29(9), 2352-2449. doi:10.1162/neco_a_00990
68. Reinhard, E., Ashikhmin, M., Gooch, B., & Shirley, P. (2001). Color Transfer between Images. *Applied Perception*, 34-41.
69. Ren, X., & Malik, J. (2003). Learning a Classification Model for Segmentation. *International Conference on Computer Vision*, (págs. 10-17).
70. Rezatofghi, S. H., & Soltanian-Zadeh, H. (2011). Automatic Recognition of Five Types of White Blood Cells in Peripheral Blood. *Computerized Medical Imaging and Graphics*, 35(4), 333-343. doi:10.1016/j.compmedimag.2011.01.003
71. Rodak, B. F., & Carr, J. H. (2014). Introducción al exámen de sangre de frotis periférica. En *Atlas de Hematología Clínica* (Cuarta ed., págs. 1-10). Editorial Médica Panamericana.
72. Rodrigues, P., Ferreira, M., & Monteiro, J. (2008). Segmentation and classification of leukocytes using neural networks: a generalization direction. En *Speech, Audio, Image and Biomedical Signal Processing using Neural Networks* (págs. 373-396). Berlin: Springer.
73. Ruggero, D. L., Piuri, V., & Scotti, F. (2011). All-IDB: The acute lymphoblastic leukemia image database for image processing. *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP)*, (págs. 2045-2048). Bruselas-Bélgica. doi:10.1109/ICIP.2011.6115881

74. Ruiz, F. (2016). *Algoritmos de visión computacional para la detección y clasificación de leucocitos en imágenes de frotis sanguíneo*. Tesis de Maestría, Aguascalientes.
75. Sánchez-Salmerón, A. j., & Ricolfe-Viala, C. (2016). Sistema de captura de imágenes. En E. Alegre, G. Pajares, & A. de la Escalera (Edits.), *Conceptos y métodos en visión por computador* (págs. 11-29). Madrid, España: Grupo de Visión del Comité Español de Automática.
76. Saraswat, M., & Arya, K. V. (2014). Automated Microscopic Image Analysis for Leukocytes Identification: A Survey. *Micron*, 65, 20-33. doi:10.1016/j.micron.2014.04.001
77. Sarrafzadeh, O., Rabbani, H., Talebi, A., & Usefi Banaem, H. (2014). Selection of the best features for leukocytes classification in blood smear microscopic images. *Proceedings of SPIE. Medical Imaging 2014: Digital Pathology, 9041*. San Diego - USA. doi:10.1117/12.2043605
78. Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques. *Journal of Electronic Imaging*, 13(1), 146-168. doi:10.1117/1.1631315
79. Shahin, A., Guo, Y., Amin, K., & Sharawi, A. A. (2019). White blood cells identification system based on convolutional deep neural learning networks. *Computer Methods and Programs in Biomedicine*, 168, 69–80.
80. Shallu, & Mehra, R. (2018). Breast cancer histology images classification: Training from scratch or transfer learning. *ICT Express*, 4(4), 247–254. doi:10.1016/j.icte.2018.10.007
81. Sharma, N., Jain, V., & Mishra, A. (2018). An Analysis Of Convolutional Neural Networks For Image Classification. (Elsevier, Ed.) *Procedia Computer Science*, 132, 377-384. doi:10.1016/j.procs.2018.05.198
82. Shi, J., & Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 888-905.
83. Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *Proceedings of the International Conference on Learning Representations 2015*. Obtenido de <https://arxiv.org/abs/1409.1556>
84. Smith, A. (1978). Color Gamut Transform Pairs. *ACM Siggraph Computer Graphics*, (págs. 12-19). doi:10.1145/800248.807361
85. Stutz, D., Hermans, A., & Leibe, B. (2016). Superpixels: An Evaluation of the State-of-the-Art. *Computer Vision and Image Understanding*. doi:10.1016/j.cviu.2017.03.007
86. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference*

- on Computer Vision and Pattern Recognition (CVPR)* (págs. 1-9). Boston - USA: IEEE. doi:10.1109/CVPR.2015.7298594
87. Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., & Liang, J. (2016). Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Transactions on Medical Imaging*, 35(5), 1299 - 1312. doi:10.1109/TMI.2016.2535302
 88. The MathWorks, Inc. (2019). Deep Learning Toolbox™ Reference. Natick, Massachusetts, USA. Obtenido de https://la.mathworks.com/help/pdf_doc/deeplearning/nnet_ref.pdf
 89. Torrey, L., & Shavlik, J. (2009). Transfer Learning. En E. Soria, J. Martin, R. Magdalena, M. Martinez, & A. Serrano (Edits.), *Handbook of Research on Machine Learning Applications*. IGI Global.
 90. Universidad Pedagógica Experimental Libertador. (2003). *Manual de Trabajos de Grado de Especialización y Maestría y Tesis Doctorales* (Tercera ed.). Caracas: FEDUPEL.
 91. Van de Wolfshaar, J., Karaaba, M. F., & Wiering, M. A. (2015). Deep Convolutional Neural Networks and Support Vector Machines for Gender Recognition. *IEEE Symposium Series on Computational Intelligence*, (págs. 188-195). Cape Town. doi:0.1109/SSCI.2015.37
 92. Vapnik, V., & Cortes, C. (1995). Support-Vector Networks. *Machine Learning*, 273-297.
 93. Vélez, S. J., Moreno, D. A., Sánchez, C. Á., & Sánchez, M. J. (2003). *Visión por computador*. Madrid, España: Dykinson.
 94. Wu, Z., & Leahy, R. (1993). An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 5, 1101-1113.
 95. Xu, Y., Jia, Z., Wang, L., Ai, Y., Zhang, F., Lai, M., & Chang, E. (2017). Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features. *BMC Bioinformatics*.
 96. Xue, D.-X., Zhang, R., Feng, H., & Wang, Y.-L. (2016). CNN-SVM for Microvascular Morphological Type Recognition with Data Augmentation. *Journal of Medical and Biological Engineering*, 36(6), 755-764. doi:10.1007/s40846-016-0182-4
 97. Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Proceedings of the 27th International Conference on Neural Information Processing Systems*. 2, págs. 3320-3328. Montreal - Canada: MIT Press. Obtenido de <https://arxiv.org/abs/1411.1792>

98. Zack, G., Rogers, W., & Latt, S. (1977). Automatic measurement of sister chromatid exchange frequency. *The Journal of histochemistry and cytochemistry*, 5(7), 741–753.
99. Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. *Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science* (págs. 818-833). Springer, Cham.
100. Zhang, C., Xiao, X., Li, X., Chen, Y., Zhen, W., Chang, J., . . . Liu, Z. (2014). White Blood Cell Segmentation by Color-Space-Based K-Means Clustering. *Sensors*, 14, 16128-16147.
101. Zheng, X., Wang, Y., Wang, G., & Liu, J. (2018). Fast and robust segmentation of white blood cell images by self-supervised learning. *Micron*, 107, 55-71.
102. Zhu, P., Isaacs, J., Fu, B., & Ferrari, S. (2018). Deep learning feature extraction for target recognition and classification in underwater sonar images. *2017 IEEE 56th Annual Conference on Decision and Control (CDC)* (págs. 2724-2731). Melbourne - Australia: IEEE. doi:10.1109/CDC.2017.8264055

APÉNDICES Y ANEXOS EN CD

APÉNDICES

Código o scripts de MATLAB:

A-1: Pruebas iniciales.

A-2: Segmentación.

A-3: Clasificador SVM

A-4: Ajuste de pesos CNN

A-5: Código sistema clasificación leucocitos

ANEXOS

ANEXO A: Pruebas segmentación mediante umbral global

ANEXO B: Pruebas de transferencia de color

ANEXO C: Resultados obtenidos con clasificador #2 (ResNet-50)