

***“CLASIFICACIÓN DE PARÁSITOS EN IMÁGENES
DIGITALES DE MUESTRAS DE COPRO MEDIANTE
APRENDIZAJE DE MÁQUINA”***

TESIS

QUE PARA OBTENER EL GRADO DE

***MAESTRO EN CIENCIAS EN INGENIERÍA
ELECTRÓNICA***

PRESENTA:

MIGUEL ÁNGEL GUTIÉRREZ VELÁZQUEZ

**DIRECTOR DE LA TESIS:
DR. MARIO IGNACIO CHACÓN MURGUÍA
CODIRECTOR DE LA TESIS:
DR. CARLOS ARZATE QUINTANA**

CHIHUAHUA, CHIH., SEPTIEMBRE



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®





Chihuahua, Chih. 7 de septiembre de 2021

ROGELIO ENRIQUE BARAY ARANA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
PRESENTE

En cumplimiento con los requerimientos para la obtención del grado de Maestro en Ciencias en Ingeniería Electrónica, le notificamos que el documento de tesis del alumno C. MIGUEL ÁNGEL GUTIÉRREZ VELÁZQUEZ, titulado "*Clasificación de parásitos en imágenes digitales de muestras de copro mediante aprendizaje de máquina*" dirigido por el Dr. Mario Ignacio Chacón Murguía ha sido aprobado y aceptado para su impresión.

Por lo anterior, proponemos le sea concedida la autorización de impresión correspondiente.

Agradeciendo la atención a la presente, quedamos de usted:

ATENTAMENTE

Excelencia en Educación Tecnológica®
"La Técnica por el Engrandecimiento de México"

DR. MARIO IGNACIO CHACÓN MURGUÍA
DIRECTOR DE TESIS

DR. JUAN ALBERTO RAMÍREZ QUINTANA
MIEMBRO DEL COMITÉ TUTORIAL

MTRO. CARLOS ARZATE QUINTANA
MIEMBRO DEL COMITÉ TUTORIAL

MTRA. ALMA DELIA CORRAL SÁENZ
MIEMBRO DEL COMITÉ TUTORIAL





Chihuahua, Chih. 13 de septiembre de 2021

**C. MIGUEL ÁNGEL GUTIÉRREZ VELÁZQUEZ
PRESENTE**

En cumplimiento con los requerimientos para la obtención del grado de Maestro en Ciencias en Ingeniería Electrónica y a propuesta de su Comité Tutorial, la División de Estudios de Posgrado e Investigación le concede la autorización para imprimir la tesis titulada *"Clasificación de parásitos en imágenes digitales de muestras de copro mediante aprendizaje de máquina"* dirigida por el Dr. Mario Ignacio Chacón Murguía con el siguiente contenido de capítulos:

- I Antecedentes
- II Segmentación
- III Aumento de datos
- IV Extracción de características
- V Clasificación
- VI Conclusiones y trabajo futuro

ATENTAMENTE

Excelencia en Educación Tecnológica®
"La Técnica por el Engrandecimiento de México"

ROGELIO ENRIQUE BARAY ARANA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN





CARTA CESIÓN DE DERECHOS

En la ciudad de Chihuahua el día 13 de septiembre de 2021, el que suscribe C. MIGUEL ÁNGEL GUTIÉRREZ VELÁZQUEZ con número de control G19061419, de la Maestría en Ciencias en Ingeniería Electrónica, adscrita a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua, manifiesta que es autor intelectual del presente trabajo de Tesis bajo la dirección del Dr. Mario Ignacio Chacón Murguía y cede los derechos del trabajo titulado "Clasificación de parásitos en imágenes digitales de muestras de copro mediante aprendizaje de máquina", al Tecnológico Nacional de México y/o Instituto Tecnológico de Chihuahua para su difusión, divulgación, transmisión, reproducción, así como su digitalización con fines académicos y de investigación.

C. MIGUEL ÁNGEL GUTIÉRREZ VELÁZQUEZ



DECLARACIÓN DE ORIGINALIDAD

En la ciudad de Chihuahua el día 13 de septiembre de 2021 17 de septiembre de 2021, el que suscribe C. MIGUEL ÁNGEL GUTIÉRREZ VELÁZQUEZ de la Maestría en Ciencias en Ingeniería Electrónica, con número de control G19061419, adscrito a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua, manifiesta que es autor intelectual de la tesis titulada "Clasificación de parásitos en imágenes digitales de muestras de copro mediante aprendizaje de máquina" bajo la dirección el Dr. Mario Ignacio Chacón Murguía; que el contenido es original y que las fuentes de información consultadas para su fundamentación están debidamente citadas y referenciadas.

C. MIGUEL ÁNGEL GUTIÉRREZ VELÁZQUEZ

AGRADECIMIENTOS

Dra. María Elena Álvarez-Buylla Roces

Director de CONCACYT

Presente

Por este conducto aprovecho la ocasión para saludarlo e informarle que a la fecha he obtenido el grado de Maestría en Ciencias en Ingeniería Electrónica en la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua. Motivo por el cual agradezco todo el apoyo brindado por esta Institución que usted representa, el otorgamiento de esta beca permitió dedicarme de tiempo completo a la realización de mis estudios de Posgrado y de esta manera lograr el cumplimiento del objetivo principal del convenio establecido.

Sin otro particular por el momento, me es grato quedar de usted como su seguro servidor, no sin antes reiterar mi agradecimiento. Muchas gracias.

Atentamente



Miguel Ángel Gutiérrez Velázquez

Exbecario CONACYT

AGRADECIMIENTOS

AGRADECIMIENTO

A mi madre Norma, mi padre Héctor, mi hermano Daniel, mi hermana Liz, mis sobrinos Lalo, Isaí y mis sobrinas Dulce y Rebeca por alegrarme cada día de mi vida, por hacerme ser quien soy y por ayudarme y apoyarme en cada etapa de mi vida.

A mi queridísima novia Gina, por siempre haber sido buena conmigo, ofrecerme su cariño, su tiempo y hacerme inmensamente dichoso por su compañía. Porque la vida resulta mucho más bonita a su lado.

A mis amigos Eli, Xavier y Vicente, con los cuales compartí la Ingeniería y la Maestría, y a los cuales les debo muchísimo.

A cada uno de los profesores involucrados durante toda mi etapa escolar, pues cada uno aportó algo y me compartió su conocimiento. Especialmente al Dr. Mario Ignacio Chacón Murguía, que desde el primer momento se interesó en mi formación tanto académica como profesional, que me dio consejos y advertencias necesarias siempre con el objetivo de mejorar mis virtudes y mis defectos.

Al Tecnológico Nacional de México por el financiamiento del proyecto “Clasificación de Parásitos en Muestras de Copro Mediante Aprendizaje de Máquina” con clave 10071.21-P.

NOTACIÓN

SÍMBOLO	DEFINICIÓN
I	Imagen
$I(x,y)$	Intensidad del pixel (x, y) de una imagen
$L(x,y)$	Iluminación del pixel (x, y) de una imagen
$R(x,y)$	Reflectancia del pixel (x, y) de una imagen
I_h	Canal I del espacio de color HSI
r	Ruido de una imagen
h	Función del sistema de adquisición de la imagen
A	Señal donde se detectó tejido
U	Imagen con ruido
v	Imagen sin ruido
Ω	Contorno
H_{PA}	Filtro pasa altas
I_γ	Imagen con corrección gamma
l_o	Contraste menor de la imagen de salida
h_o	Contraste mayor para la imagen de salida
l_i	Contraste menor para la imagen de entrada
h_i	Contraste mayor para la imagen de entrada
C'	Valor de un canal C
C_{max}	Valor mayor del canal C
S'	Valor del canal S del espacio de color HSI
$\bar{S}$	Media del canal S del espacio de color HSI
$\bar{K}$	Media del valor máximo de cada columna y fila del canal S del espacio de color HSI
$p_p(i)$	Probabilidad de encontrar un pixel en el nivel i
f_i	Frecuencia con que se presenta un pixel en el nivel i
N	Cantidad de pixeles en una imagen
P_o	Probabilidad del objeto

NOTACIÓN

H_o	Entropía del objeto
P_f	Probabilidad del fondo
H_f	Entropía del fondo
t	Umbral
n_{pq}	Momentos invariantes
u_{pq}	Momentos centrales de una imagen
$\bar{i}$	Coordenada x del centroide de una imagen binaria
$\bar{j}$	Coordenada y del centroide de una imagen binaria
m_{pq}	Momentos de una imagen
R	Canal R del espacio de color RGB
G	Canal G del espacio de color RGB
B	Canal B del espacio de color RGB
E_{oc}	En otro caso
MAX	Max(R,G,B)
MIN	Min(R,G,B)
H_{SV}	Canal H del espacio de color HSV
S	Canal S del espacio de color HSV
V	Canal V del espacio de color HSV
A	Conjunto de pixeles de interés
B	Elemento estructural
E	Pixeles igual a 1 en el plano de interés
$\mathcal{X}$	Conjunto de datos reales
$\mathbb{R}$	Conjunto de números reales
ϕ	Función de costo en la segmentación por k means
c_i	i-ésimo centro
S_F	Entropía en física
K	Constante de Boltzman
V_{sis}	Estados posibles de un sistema
A_s, B_s	Sistemas con entropía
H	Entropía en teoría de la información
p, q	Distribución de probabilidad

NOTACIÓN

Y	Etiqueta
X	Dato
D_{KL}	Divergencia Kullback-Leibler
D_{JS}	Divergencia de Jensen-Shannon
M	$(p+q)/2$
L_D	Función de costo del discriminador
D	Discriminador de la red GAN
G	Generador de la red GAN
$D(x)$	Distribución de datos reales
Z	Vector de ruido
$G(z)$	Datos generados por el generador G
$D(G(z))$	Distribución de los datos generados
L_G	Función de costo del generador
p_{data}	Distribución de los datos originales
p_z	Distribución del ruido de entrada el generador G
p_g	Distribución de los datos generados
D^*	Valor óptimo del generador D
$V(G,D)$	Valor esperado de la función de costo de la red GAN
$\partial v/\partial D$	Derivada parcial de V con respecto a D
H'	Mini lote normalizado
H_m	Mini lote de las activaciones de una capa
μ	Media de un mini lote
σ	Desviación estándar de un mini lote
ϵ	Valor muy pequeño tal que $\epsilon^2 = 0$
$f(x)$	Función que depende de un valor x
$\max(0,x)$	Función cuyo resultado es el número mayor entre 0 y x
a	Pendiente de la recta para valores no menores o iguales a cero
$Tanh(x)$	Tangente hiperbólica de x
$Area$	Número de pixeles de un blob
x_c	Coordenada en el eje x
Y_c	Coordenada en el eje y

NOTACIÓN

$\overline{cf}$	Distancia del centro al foco de la elipse de un blob
e_M	Longitud del eje mayor de un blob
Ext	Extensión
$Rect$	Píxeles del rectángulo mínimo que cubre un blob
Sol	Solidez
$AreaC$	Área convexa
M_G	Matriz de coocurrencia de niveles de gris
N_G	Niveles de gris de una imagen
$\mathcal{D}$	Dominio
Z	Espacio de características
$P(x)$	Probabilidad marginal
T	Tarea
$\mathcal{Y}$	Espacio de etiquetas
Γ	Función de decisión
$\mathcal{D}_s$	Dominio fuente
τ_s	Tarea fuente
R	Observaciones
$\mathbb{N}^+$	Números enteros positivos
$\mathcal{G}$	Densidad de probabilidad
$GLCM$	Matriz de coocurrencia de niveles de gris
D	Desplazamiento
μ_G	Media de GLCM
σ_G^2	Desviación estándar de GLCM
R	Radio
Pbl	Distribución adjunta
Θ	Textura
C_o	Curva cerrada en el plano complejo
Z	Función compleja
$\mathcal{A}$	Descriptor de Fourier
N_D	Número de elementos en $\mathcal{D}$
L	Número de muestras

NOTACIÓN

k	Número de clases
$\mathcal{M}$	Espacio de ejemplos

CONTENIDO

LISTA DE FIGURAS	xv
LISTA DE TABLAS.....	xix
RESUMEN	xx
INTRODUCCIÓN.....	1
CAPÍTULO 1 ANTECEDENTES	3
1.1 Parásitos.....	3
1.2 Visión por computadora en imágenes médicas.....	3
1.2.1 Mejoramiento de imágenes	4
1.2.2 Segmentación	7
1.2.3 Detección	9
1.2.4 Extracción de características	12
1.2.5 Clasificación sin métodos de aprendizaje profundo.....	13
1.2.6 Clasificación con métodos de aprendizaje profundo.....	15
CAPÍTULO 2 SEGMENTACIÓN.....	18
2.1 Procesamiento de la imagen	18
2.2. Segmentación con k means.....	22
CAPÍTULO 3 AUMENTO DE DATOS.....	25
3.1 Generación de datos	25
3.2 Transformaciones tradicionales	25
3.3 Redes generativas adversarias	26
3.3.1. Entropía en física.....	28
3.3.2. Entropía en teoría de la información	28
3.3.3. Entropía cruzada.....	29
3.3.4. Divergencia de Kullback-Leibler	30
3.3.5. Divergencia de Jensen-Shannon.....	31
3.3.6. Discriminador.....	32
3.3.7 Generador.....	32

3.3.8. Problema min-max	32
3.3.9. Entrenamiento	33
3.3.10 Elementos de la arquitectura de la GAN	35
3.3.11. Implementación	39
CAPÍTULO 4 EXTRACCIÓN DE CARACTERÍSTICAS	48
4.1. Características de forma	48
4.2. Características de textura.....	49
4.2.1 Matriz de coocurrencia de niveles de gris	49
4.2.2 Matriz de diferencia de niveles de gris.....	52
4.2.3 Patrones binarios locales.....	54
4.3. Descriptores de Fourier	55
CAPÍTULO 5 CLASIFICACIÓN.....	58
5.1. Clasificación binaria con Support Vector Machine (SVM)	58
5.2. Clasificación con aprendizaje profundo	62
5.2.1 Percepción visual en mamíferos.....	64
5.2.2 Reconocimiento de características y aprendizaje en modelos profundos.....	64
5.2.3 Transferencia de aprendizaje	65
5.2.4 Red AlexNet.....	67
5.2.5 Clasificación binaria	68
5.2.6 Clasificación multiclase.....	76
CAPÍTULO 6 CONCLUSIONES Y TRABAJO FUTURO	83
REFERENCIAS.....	86

LISTA DE FIGURAS

Figura 2.1. Imágenes de muestra. a) Giardia Lamblia, b) Endolimax Nana, c) Entamoeba Histolytica, d) Chilomastix Mesnili, e) Blastocystis Hominis, f) Entamoeba Coli.....	18
Figura 2.2. Imagen original, fondo azul, parásito en otro color.....	19
Figura 2.3. Espacio de color HSV.....	19
Figura 2.4. Imagen en el espacio de color HSV multiplicada por dos.....	20
Figura 2.5. Canal S.	21
Figura 2.6. Complemento de la binarización del canal S.	21
Figura 2.7. Dilatación.	22
Figura 2.8. Parásito detectado.	22
Figura 2.9. Segmentación mediante K means.....	23
Figura 2.10. Resultado de la segmentación, a) Blastocystis hominis, b) Entamoeba coli, c) Chilomastix mesnili, d) Entamoeba histolytica, e) Giardia lamblia, f) Endolimax nana, g) Enteromonas hominis, h) No parásito.....	24
Figura 3.1. Aumento de datos mediante transformaciones, a) Imagen original, b) Imagen reflejada en el eje y, c) Imagen reflejada en el eje x, d) Aumento de contraste, e) Imagen rotada 90°, f) Imagen rotada 180°, g) Imagen rotada 270°, h) Imagen con ruido gaussiano, i) Imagen con ruido sal y pimienta, j) Imagen con ruido Poisson, k) Aumento de brillo, l) Disminución de brillo, m) Borrado aleatorio, n) Corrección gamma.	26
Figura 3.2. Modelo de una red generativa adversaria	27
Figura 3.3. Tipos de modelos, a) discriminativos, b) generativos.	27
Figura 3.4, Estimación de una distribución observada mediante una distribución uniforme y una distribución binomial.	31
Figura 3.5. Activaciones de la primera capa de convolución de la red Alexnet ante una imagen de parásito.	35
Figura 3.6. Ilustración de la operación convolución traspuesta. La entrada es un mapa de características de 4x4. Cuatro mapas de características intermedios son generados usando cuatro kernels diferentes. Estos se combinan produciendo una salida de 8x8. Los cuatro mapas de características intermedios se basan en la entrada pero no hay relación entre ellos.	36

Figura 3.7. Red antes del dropout y red después del dropout.	37
Figura 3.8. Función de activación ReLU.....	39
Figura 3.9. Función de activación Leaky ReLU.	40
Figura 3.10. Función de activación tangente hiperbólica.	40
Figura 3.11. Imágenes generadas cuando comienza el entrenamiento.....	40
Figura 3.12. Imágenes generadas en la iteración 104.....	41
Figura 3.13. Imágenes generadas al final del entrenamiento en la iteración 19833.	42
Figura 3.14. Imágenes generadas para la clase tulipanes.	43
Figura 3.15. Primera actualización en el entrenamiento de la GAN para generación de parásitos.....	43
Figura 3.16. Iteración 1245 en la generación de parásitos.	44
Figura 3.17. Parásitos generados.....	44
Figura 3.18. Parásitos generados con 1000 épocas y un mini lote de 64.	44
Figura 3.19. Generación de datos sin el aumento de datos clásico en el conjunto de datos de entrenamiento.....	45
Figura 3.20. Parásitos generados con 700 muestras y 96 filtros en la capa de deconvolución del generador.	45
Figura 3.21. Generación de datos con el conjunto de datos repetido.....	46
Figura 3.22. Imágenes generadas con la GAN, a) Blastocystis hominis, b) Entamoeba coli, c) Chilomastix mesnili, d) Entamoeba histolytica, e) Giardia lamblia, f) Endolimax nana, g) Enteromonas hominis.....	46
Figura 4.1. Direcciones de las rotaciones para que la GLCM se invariante a la rotación... 50	50
Figura 4.2. Parásito.....	53
Figura 4.3. Distribuciones de probabilidad acumuladas, P1 es para el vector (0,1), P2 es para el vector (-1,0), P3 es para el vector (1,0) y P4 es para el vector (-1,1).	53
Figura 4.4. Patrones binarios locales para un parásito (izquierda) y un no parásito (derecha).....	56
Figura 4.5. Descriptores de Fourier de un parásito.	57
Figura 4.6. Descriptores de Fourier de un no parásito.	57

Figura 5.1. Función de error del modelo binario SVM 5 con 160 imágenes de entrenamiento, 32 de validación, 32 de prueba y 100 características.....	61
Figura 5.2. Valor de la función del modelo SVM 5.	61
Figura 5.3. ROC de entrenamiento del modelo binario SVM 5 con 160 imágenes de entrenamiento, 32 de validación, 32 de prueba y 100 características.	62
Figura 5.4. Arquitectura de AlexNet [60].....	67
Figura 5.5. Imagen para observar las activaciones de la red AlexNet.	67
Figura 5.6. Activaciones de filtros, a) Conv1, b) Conv2, c) Conv3, d) Conv4, e) Conv5. .	68
Figura 5.7. Desempeño para el Clasificador 1 con 710 muestras de parásitos reales y 710 muestras de no parásitos.....	69
Figura 5.8. Matriz de confusión para del conjunto de prueba (el 15% de 710 muestras de parásitos y 710 muestras de no parásitos) para el Clasificador 1.	69
Figura 5.9. ROC para del conjunto de prueba (el 15% de 710 muestras de parásitos y 710 muestras de no parásitos) para el Clasificador 1.	69
Figura 5.10. Matriz de confusión para del conjunto de prueba formado por 32 parásitos, 32 no parásitos y 32 parásitos generados con la GAN evaluado con el Clasificador 1.....	70
Figura 5.11. ROC para el conjunto de prueba de 32 parásitos, 32 no parásitos y 32 parásitos generados con la GAN evaluado con el Clasificador 1.....	70
Figura 5.12. Desempeño durante el entrenamiento del Clasificador 2.....	71
Figura 5.13. Matriz de confusión del conjunto de prueba del Clasificador 2.....	72
Figura 5.14. ROC del conjunto de prueba del Clasificador 2.....	72
Figura 5.15. Matriz de confusión del conjunto formado por 32 parásitos reales, 32 parásitos generados con la GAN y 32 no parásitos clasificado con el Clasificador 2.....	72
Figura 5.16. ROC del conjunto formado por 32 parásitos reales, 32 parásitos generados con la GAN y 32 no parásitos evaluado con el Clasificador 2.	73
Figura 5.17. Desempeño del Clasificador 3, entrenado con 355 muestras de parásitos reales, 355 muestras de parásitos generados con la GAN y las mismas 710 muestras de no parásitos que el Clasificador 1 y 2.....	74

Figura 5.18. Matriz de confusión del Clasificador 3, entrenado con 355 muestras de parásitos reales, 355 muestras de parásitos generados con la GAN y las mismas 710 muestras de no parásitos que el Clasificador 1 y 2.....	74
Figura 5.19. ROC del Clasificador 2, entrenado con 355 muestras de parásitos reales, 355 muestras de parásitos generados con la GAN y las mismas 710 muestras de no parásitos que el Clasificador 1 y 2.	74
Figura 5.20. ROC del Clasificador 2 clasificando las muestras de 32 parásitos, 32 no parásitos y 32 parásitos generados con la GAN.	75
Figura 5.21.ROC del Clasificador 2 con las muestras de 32 parásitos, 32 no parásitos y 32 parásitos generados con la GAN.....	75
Figura 5.22. Desempeño durante el entrenamiento del clasificador de 8 clases con 7010 muestras reales.....	76
Figura 5.23. Matriz de confusión del conjunto de entrenamiento del Clasificador 4.	77
Figura 5.24. ROC del conjunto de entrenamiento del Clasificador 4.	77
Figura 5.25. Matriz de confusión del conjunto de prueba del Clasificador 4.....	78
Figura 5.26. ROC del conjunto de prueba del Clasificador 4.....	78
Figura 5.27. Matriz de confusión del conjunto de validación del Clasificador 4.	79
Figura 5.28. ROC del conjunto de validación del Clasificador 4.	79
Figura 5.29. Matriz de confusión del conjunto de prueba del Experimento 1 de la Tabla 5.10.....	81
Figura 5.30. Matriz de confusión del conjunto de prueba del Experimento 2 de la Tabla 5.10.....	81

LISTA DE TABLAS

Tabla 2.1. Desempeño en imágenes con parásitos.	23
Tabla 3.1. Arquitectura del generador y del discriminador de la GAN diseñada.	41
Tabla 4.1. Promedio de algunas características de forma de parásitos y no parásitos.	49
Tabla 4.2. Resultados de cuatro características de textura.	52
Tabla 5.1. Resultados con imágenes de prueba del clasificador binario con SVM.	60
Tabla 5.2. Resultados del modelo binario SVM 5 con 160 imágenes de entrenamiento, 32 de validación, 32 de prueba y 100 características.....	62
Tabla 5.3. Clasificador 1. Red con 710 muestras de parásitos reales y 710 muestras de no parásitos, divididas en 70% para entrenamiento, 15% para validación y 15% para prueba. ...	71
Tabla 5.4. Validación cruzada k-fold con $k = 10$ para el Clasificador 1.....	71
Tabla 5.5. Clasificador 2. Red con 710 muestras de parásitos generados con la GAN y 710 muestras de no parásitos, divididas en 70% para entrenamiento, 15% para validación y 15% para prueba.	73
Tabla 5.6. Clasificador 3, entrenado con 355 muestras de parásitos, 355 muestras de parásitos generados con la GAN y 355 muestras de no parásitos, divididas en 70% para entrenamiento, 15% para validación y 15% para prueba.	75
Tabla 5.7. Validación cruzada k-fold con $k = 10$ para el Clasificador 3, entrenado con 355 muestras de parásitos reales, 355 muestras de parásitos generados con la GAN y 710 muestras de no parásitos.	76
Tabla 5.8. Resultados del entrenamiento, validación y prueba del clasificador de 4.....	80
Tabla 5.9. Validación cruzada k-fold con $k = 10$ para el Clasificador 4.....	80
Tabla 5.10. Resultados de los clasificadores entrenados con cierto porcentaje de imágenes reales y un mayor porcentaje de imágenes generadas con la GAN.	80

RESUMEN

CLASIFICACIÓN DE PARÁSITOS EN IMÁGENES DIGITALES DE MUESTRAS DE COPRO MEDIANTE APRENDIZAJE DE MÁQUINA

Miguel Ángel Gutiérrez Velázquez

Maestría en Ciencias en Ingeniería Electrónica

División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Chihuahua
Chihuahua, Chih., agosto 2021

Director: Dr. Mario Ignacio Chacón Murguía

Codirector: Dr. Carlos Arzate Quintana

Esta tesis forma parte del proyecto general “Algoritmos de visión artificial para realizar análisis de imágenes digitales para diagnóstico clínico parasitológico, urológico, hematológico y en muestras histológicas de patología” que realiza el Instituto Tecnológico de Chihuahua en conjunto con la Facultad de Medicina y Ciencias Biomédicas de la Universidad Autónoma de Chihuahua, y del proyecto “Clasificación de parásitos en muestras de copro mediante aprendizaje de máquina”, con clave 10071.21-P del Tecnológico Nacional de México.

Para entrenar una red profunda se requiere una gran cantidad de muestras; sin embargo, cuando se trata de imágenes médicas, las muestras no abundan: ya sea por privacidad, dificultad de la adquisición, costo u otros factores. Para plantear la solución al objetivo (realizar un clasificador multiclase con redes profundas) se tomó en cuenta la escasa cantidad de muestras, con lo cual es menester conocer técnicas y enfoques para un entrenamiento con pocas muestras.

Una vez tenido eso en cuenta, se utilizaron tres métodos: aumento de datos, generación de datos y transferencia de aprendizaje. El aumento de datos consistió en aplicar transformaciones en la imagen original, la generación de datos se efectuó con una red generativa adversaria, una vez se incrementó la cantidad de muestras de entrenamiento, se hizo una transferencia de aprendizaje: se entrenó la red Alexnet para clasificar primero entre dos clases y después entre 8 clases.

Se hicieron diversos experimentos tanto para la generación de datos como para la clasificación, estos experimentos se basaron tanto en la modificación de algunos parámetros de las redes pero, principalmente, en la modificación de los conjuntos de entrenamiento. Las imágenes de parásitos generadas con la GAN fueron utilizadas como conjunto de entrenamiento y prueba para clasificadores SVM y profundos. Un modelo entrenado con imágenes reales para clasificar entre 8 clases tuvo 98.38% de exactitud en el entrenamiento, 98.76% en la validación y 98.4% en la prueba. Siendo, además, capaz de clasificar correctamente las 32 imágenes generadas con la GAN con las que se probó. Otro modelo fue entrenado solo con imágenes generadas con la GAN, teniendo muy buenos resultados en dos clases al clasificar parásitos reales: *Entamoeba histolytica*, donde clasificaron 512 muestras correctamente por 23 incorrectamente y *Entamoeba hominis*, donde se tuvieron 200 clasificaciones correctas y solo 10 incorrectas.

Finalmente, hay cosas mejorables en cada etapa de la investigación: desde el procesamiento de las imágenes, segmentación, extracción de características, generación de imágenes y la clasificación. El alcance de la investigación crecería con un procesamiento y segmentación correcta en imágenes complicadas, con muchos objetos, con superposiciones. En esta tesis se abordó la generación de imágenes con la GAN, aunque sería interesante ver otros métodos y técnicas, pues, en los casos donde las muestras son muy pocas, sería muy útil que las imágenes sintéticas puedan ser utilizadas para entrenar una red profunda con la cual se clasifiquen correctamente toda clase de imágenes reales.

INTRODUCCIÓN

La visión por computadora es un campo científico cuyo objetivo es extraer información de imágenes digitales. Para efectuar tal extracción, se emplean algoritmos que realizan operaciones en las imágenes con el fin de mejorar la información pictórica para la interpretación humana, o para tareas como almacenamiento, transmisión y extracción de información en las imágenes. La visión por computadora aparece en innumerables campos, como: agricultura, automatización, seguridad, medicina, entre otros.

La integración de la visión por computadora en el área de la medicina ayuda a los expertos a realizar los diagnósticos y tratamientos médicos de una manera más eficiente mediante el análisis de imágenes médicas. Las imágenes médicas predominantes son las obtenidas mediante rayos X, tomografía computarizada (CT), imagen de resonancia magnética (MRI), tomografía de emisión de positrones (PET) y ultrasonido. Además de los métodos anteriores, existen imágenes obtenidas a partir de un microscopio.

Las imágenes conseguidas con un microscopio amplifican objetos diminutos, permitiendo observar lo que no es posible de otra manera. Gracias a esto, se puede analizar y examinar la composición, la estructura y las características de células y tejidos de seres vivos, así como la detección y clasificación de parásitos que habitan en el interior de humanos y animales.

Las imágenes de parásitos son utilizadas en esta investigación para identificar el tipo de parásito presente en dichas imágenes. Así pues, esta investigación presenta un sistema de visión que clasifica imágenes de parásitos de manera automática.

El diagnóstico automatizado por medio de la visión por computadora y el aprendizaje de máquina de parásitos, comienza con la adquisición de las muestras; luego, un preprocesamiento para eliminar ruido, normalizar la iluminación y las variaciones inherentes a la adquisición. A continuación, se realiza una segmentación para aislar los parásitos de lo que no es parásito. Después, se extraen características de los objetos segmentados, las cuales, posteriormente, son clasificadas. Sin embargo, con la disponibilidad de miles de muestras y

computadoras más robustas, el diagnóstico de parásitos en imágenes digitales comenzó a ser atendido mediante el aprendizaje profundo.

Para obtener un resultado correcto en el reconocimiento y clasificación de los parásitos con el aprendizaje profundo, se requieren cientos o miles de imágenes de muestra. Cuestión que no es un problema actualmente, debido a la abundancia de bases de datos. Estas imágenes se usan para entrenar el algoritmo de aprendizaje profundo, el cual extrae características abstractas de manera automática para luego clasificarlas.

Los modelos profundos han resultado ser excelentes en la clasificación de imágenes, con lo cual es importante encontrar soluciones para poder clasificar parásitos mediante aprendizaje profundo cuando el conjunto de muestras es escaso.

Este trabajo conjunto entre el Instituto Tecnológico de Chihuahua y la Facultad de Medicina y Ciencias Biomédicas de la Universidad Autónoma de Chihuahua tiene como finalidad el diseño de software (códigos) de clasificación de imágenes de muestras de copro que se utilice en el proyecto “Algoritmos de Visión Artificial Para Realizar Análisis de Imágenes Digitales Para Diagnóstico Clínico Parasitológico, Urológico, Hematológico y en Muestras Histológicas de Patología”. Las muestras tomadas de manera estandarizada son adquiridas por la Facultad de Medicina y Ciencias Biomédicas de la Universidad Autónoma de Chihuahua.

CAPÍTULO 1 ANTECEDENTES

1.1 Parásitos

Un parásito es un organismo que habita dentro de otro organismo de diferente especie, donde el primero se beneficia de los nutrientes del segundo. Así pues, los parásitos intestinales son organismo que viven en el tracto intestinal [1]. El huésped, normalmente, sufre enfermedades, infecciones e incomodidades, en algunas circunstancias, el huésped no muestra señales de alguna infección provocada por los parásitos. El diagnóstico de enfermedades debidas a la presencia de parásitos se consigue por medio de métodos de laboratorio, endoscopías o con procesamiento de imágenes.

1.2 Visión por computadora en imágenes médicas

La visión por computadora es un campo científico cuyo fundamento es la extracción de información en imágenes digitales. Para conseguir la extracción de información, se utilizan algoritmos de procesamiento de imágenes que realizan diversas operaciones en las imágenes con el fin de mejorar la información presente para la consecuente interpretación humana, para el almacenamiento, transmisión o extracción de información.

En las recientes décadas las técnicas de procesamiento de imágenes han sido aplicadas en tomografías, imágenes de resonancia magnética, tomografías de emisión de positrones, mamografías, ultrasonidos, imágenes de rayos X, etcétera, con el objetivo de detectar, diagnosticar y tratar enfermedades en etapas tempranas. La aplicación de aprendizaje de máquina es rol fundamental en muchas tareas en el análisis de imágenes médicas, ya que permite encontrar características que definan correctamente las irregularidades o patrones inherentes en el conjunto de datos. Sin embargo, el diseño de las tareas enfocadas a la extracción de las características era realizado, mayormente, por humanos expertos; los cuales se basaban en su conocimiento empírico [2]. La identificación automática de enfermedades basadas en métodos tradicionales de aprendizaje de máquina requería un esfuerzo tedioso por parte del experto, el cual resultaba en evaluaciones subjetivas, propensas al error humano. Así pues, el desarrollo de nuevas técnicas resultaba necesario.

Es entonces que aparece el aprendizaje profundo. Este no sólo ayuda a seleccionar y extraer características, sino que construye nuevas y, además, realiza una predicción de la muestra. Uno de los primeros trabajos publicados donde se utilizan técnicas profundas en imágenes médicas es el de Shih-Chung *et. al*, en el cual se aplicó una red neuronal convolucional para la detección de nódulos, esto en 1995. Mientras que el análisis por computadora de imágenes médica se remonta a la década de los sesenta. Lodwick *et. al* se enfocaron en el cáncer de pulmón en 1963 y Becker *et. al* en imágenes de rayos X del pecho en 1964. Los casi 30 años desde que se comenzó a utilizar la visión por computadora hasta que se implementó el análisis profundo se debe a que es menester un procesamiento computacional muy extenso, así como de la disponibilidad de un gran conjunto de muestras y el desarrollo de los algoritmos de aprendizaje.

Diversas aplicaciones del aprendizaje profundo se mencionan en [3]. La primera de ellas es la segmentación de partes del cuerpo y sus componentes, como células de sangre, forma de tumores, etcétera. Los algoritmos para la segmentación pueden ser basados en redes neuronales convolucionales (CNN), combinación de mapas de características, entre otros. Otra aplicación es la clasificación de los objetos de la imagen en su clase correspondiente. De nuevo, las CNN presentan gran robustez en esta etapa. La reconstrucción de imágenes y el mejoramiento de imágenes médicas es un área donde el aprendizaje profundo también está presente.

A continuación, se expondrán técnicas y procedimientos realizados en imágenes médicas para el mejoramiento, la segmentación, la detección y la extracción de características de dichas imágenes. Por último, se hace un análisis de metodologías propuestas para la clasificación de elementos de interés presentes en las imágenes médicas.

1.2.1 Mejoramiento de imágenes

Un método para el mejoramiento de imágenes médicas con base en un modelo de red neuronal convolucional (CNN) y ampliación de la banda de frecuencia fue presentando en [4]. Primeramente, se realiza un preprocesamiento para llevar la imagen al dominio de curvelet; en esta etapa se implementa un procesamiento de ciclo de giro, el cual fue usado para corregir

distorsiones en la imagen debido a la eliminación de ruido por el método de curvelet. A continuación, se aplica la transformada de curvelet, aprovechando sus propiedades de multi escala y multi dirección que representan de mejor manera la imagen. Al descomponer la imagen, la información relacionada con detalles y contornos forma parte de la banda de alta frecuencia, al igual que el ruido; con lo cual se aplica un umbral obtenido con el método de validación cruzada en esta banda para remover el ruido. Se prosigue con la CNN, la cual tiene cuatro capas convolucionales. La CNN es empleada para obtener un mapa de iluminación de la imagen original. Las dos primeras capas de la CNN lidian con el brillo alto, la tercera capa se enfoca en las áreas con baja iluminación y la última capa es usada para reconstruir el mapa de brillo. Los pesos de la CNN se obtienen con el algoritmo del gradiente descendiente estocástico. Luego, basándose en

$$I(x, y) = L(x, y)R(x, y) \quad (1.1)$$

donde I es la imagen, L es la iluminación y R es la reflectancia, se estima el valor de L con la CNN. De manera paralela al algoritmo de CNN se sigue el algoritmo de ampliación de banda de frecuencia (FBB), el cual comprime la imagen en el dominio del tiempo para expandir la banda de frecuencia, donde se estima el valor de h en

$$I = hA + r \quad (1.2)$$

para eliminar el ruido r de la imagen I . Donde A es la señal donde se detectó tejido. Por último, se realiza una fusión a nivel pixel de los resultados de ambos acercamientos mencionados. La fusión a nivel pixel se eligió debido a que tiene más información detallada, como bordes y textura. Se presentan 4 criterios de evaluación, siendo los resultados: $SNR = 7.88$, $MSE = 68.15$, $PSNR = 26.152$ y $SSIM = 0.9345$.

La transformada wavelet estacionaria (SWT) se utiliza para el mejoramiento de la iluminación y la forma en [5]. Se aplica la SWT para descomponer la imagen original entre sub-bandas de alta frecuencia y de baja frecuencia. Al inicio, la imagen es procesada para ecualizar el histograma. El resultado de esta operación es transformado con la SWT, la cual divide la imagen en 4 sub-bandas. La información de los bordes está en las frecuencias altas, mientras que las bandas de baja frecuencia poseen información relacionada con la

iluminación. El siguiente paso consiste en aplicar la SWT a la imagen de entrada y a la banda de menor frecuencia, seguidamente se utiliza la descomposición de valores singulares. Después, se promedian ambos resultados para obtener una imagen de salida con un mejoramiento considerable en la iluminación. Sin embargo, por la propia naturaleza del promediado, la información de bordes se pierde. Para resolver esto, se reincorpora la banda de alta frecuencia en la imagen resultante. Finalmente, se aplica la inversa de SWT para obtener la imagen de salida.

W. Rui y W. Guoyu implementaron un filtro homomórfico para mejorar imágenes de rayos X [6]. El filtro homomórfico puede reducir la baja frecuencia e incrementar la alta frecuencia; por lo tanto, reduce los cambios de iluminación y resalta los bordes y detalles. El filtro homomórfico consiste en eliminar el componente L de (1.1). Para realizar esto, se obtiene el logaritmo natural de la imagen, y el producto de L y R se convierte en una suma de logaritmos, luego se aplica la transformada rápida de Fourier (FFT) y se aplica un filtro pasa altas H_{PA} . A continuación, se aplica la IFFT y se eleva a la exponencial el resultado, obteniendo la imagen filtrada. Se prosigue a utilizar el modelo de variación total (TV), el cual es muy usado para restauración de imágenes y para la eliminación de ruido. Si una imagen es

$$u(x, y) = v(x, y) + r(x, y) \quad (1.3)$$

donde u es la imagen con ruido, v es la imagen sin ruido y r es el ruido, la eliminación de ruido por medio de TV consiste en resolver ecuaciones diferenciales parciales. La TV de una imagen es

$$TV(u) = \int_{\Omega} |\nabla u| d\Omega \quad (1.4)$$

donde Ω es el contorno de la imagen u . La eliminación del ruido consiste en minimizar $TV(u)$. El resultado obtenido con el método de TV es utilizado como el filtro H_{PA} pasa altas en el filtro homomórfico. Ya que el filtro TV incrementa $R(x,y)$ y minimiza $L(x,y)$.

1.2.2 Segmentación

Una segmentación de imágenes a color por medio de Fuzzy C-Means (FCM) se presenta en [7]. En este trabajo se cuentan con imágenes microscópicas de la sangre en un espacio de color RGB. El algoritmo propuesto inicia con una normalización RGB, donde cada canal se divide entre la raíz de la suma al cuadrado de los tres canales. Este paso se tuvo que realizar debido a la manera en que las imágenes fueron adquiridas. A continuación, se utilizó la corrección para gamma para mejorar el contraste, esta corrección está dada por

$$I_{\gamma} = \left(l_o + (h_o - l_o) \left(\frac{(I - l_i)}{(h_i - l_i)} \right)^{\gamma} \right) \quad (1.5)$$

donde l_o y h_o son las restricciones del contraste para la imagen de salida, l_i y h_i son las restricciones del contraste para la imagen de entrada e I es la imagen para procesar. Se prosigue a eliminar el ruido mediante el método de Gauss. A la imagen sin ruido se le altera la iluminación mediante

$$C' = 255 - C_{\max} \quad (1.6)$$

es decir, el valor de un canal C será igual a 255 menos el valor máximo del canal C . El siguiente paso implementado fue el mejoramiento de los bordes; se toma el canal verde la imagen y se procesa con una ventana de 3x3. Los pixeles dentro de la ventana son ordenados de manera ascendente dado el valor que tengan y se ponen en un vector, luego se encuentra la media de la lista y se crean dos grupos: uno donde están los pixeles que superan la media y otra donde están los pixeles que no superan la media. De igual manera, se encuentra la media de cada lista y esos valores definen el intervalo en el que los pixeles de la imagen definirán su nuevo valor. Se procede a realizar dos grupos mediante FCM: un grupo donde esté el plasmodium y otro donde estén las células de sangre y el fondo. Finalmente, se realizan procesos morfológicos para definir mejor la estructura de los objetos de interés. El algoritmo propuesto tuvo una precisión del 98.26% y una sensibilidad de 98.61%.

Un algoritmo de segmentación de células de malaria con base en el crecimiento de la región fue desarrollado en [8]. Se comienza cambiando el espacio de color de RGB a HSI y se utiliza solamente el canal I_h para elegir los valores adecuados para ajustar la iluminación y estirar el contraste en la imagen. Luego, se hace uso del canal de saturación para separar las células normales de aquellas que están infectadas con malaria, utilizando la (2.7) para conseguir el objetivo deseado

$$S' = \frac{\bar{S} + \bar{K}}{2} \quad (1.7)$$

donde $\bar{S}$ es la media del canal S y $\bar{K}$ es la media del valor máximo de cada columna y fila del canal S . Se utiliza el canal a^* del espacio de color L^*a^*b para remover el ruido de la imagen con el ajuste de brillo. Finalmente, se utiliza el crecimiento de la región para aislar la malaria de la célula. En un análisis con 52 imágenes, se obtuvo un resultado en la precisión de 98.08%. El crecimiento de la región también se utiliza en [9]. Aquí, primeramente, se filtra cada canal en el espacio de color L^*a^*b , seguido de la obtención del gradiente multi canal. Después, se definen el lugar de las semillas para el crecimiento de la región al encontrar las intensidades mínimas. Una vez establecidas las semillas, el crecimiento de la región realiza lo siguiente: llena el área rodeada por los bordes de los valores altos del gradiente, luego la región continúa creciendo hasta que llenar la siguiente región. El algoritmo se detiene hasta que alcanza un máximo de iteraciones elegido dependiendo del tamaño del objeto.

S. Abidin *et. al* utilizan el método de contornos activos sin borde (ACWE) para segmentar malaria [10]. Primeramente, se obtiene el complemento de la imagen de entrada, el área más brillante es el objeto de interés y el área más oscura es el fondo. A continuación, se aplica el ACWE para realizar la segmentación. Este método permite que el algoritmo sea capaz de segmentar correctamente múltiples objetos con diferente forma e intensidad, así como objetos con contraste bajo, con contornos difuminados, objetos con ruido y objetos que tengan una distancia muy pequeña entre ellos. El ACWE devuelve una máscara binaria donde el blanco es el objeto y el negro es el fondo. El siguiente paso consiste en la aplicación de las operaciones morfológicas de erosión seguida de la dilatación; esto para remover pixeles que

son ruido y para mantener el tamaño de los objetos segmentados. Por último, se aplica la máscara al complemento de la imagen original, se aplica un mejoramiento del contraste y una umbralización. Se obtuvo un resultado del 97.57% de precisión en la segmentación.

Una red neuronal convolucional es utilizada en [11] para la segmentación de imágenes microscópicas. Se cuenta con 465 imágenes con máscaras etiquetadas que describen la segmentación obtenida manualmente. La red tiene cuatro bloques de *pooling* y convolución, importadas del modelo VGG16, el cual transforma la entrada en un mapa de características multidimensional. Después, las capas de sobre muestreo y convolución construyen las máscaras de la segmentación. Dado que el conjunto de muestras es pequeño, se prosigue a hacer un aumento de datos para generar muestras adicionales para el entrenamiento. El generador de datos realiza las siguientes transformaciones: de traslación, de rotación y de inversión. Se entrenó el modelo con el aumento de datos obteniendo resultados que permiten la detección de cualquier especie de parásito.

1.2.3 Detección

En [12], [13] y [14] se implementa una CNN para la detección de la malaria en imágenes digitales. H. Nugroho *et al.* proponen una CNN con dos capas de convolución y un modelo de transferencia de aprendizaje con la red pre entrenada ResNet-50 como extractor de características y comparador. El conjunto de datos consiste en 330 imágenes divididas en dos clases: con plasmodium o sin plasmodium, mientras que el conjunto de prueba consiste en 35 imágenes. Cada imagen de entrenamiento fue ajustada a 100x100 píxeles para la CNN y a 128x128 para la red pre entrenada, todas las imágenes fueron normalizadas. Además, se realizó un aumento de datos para añadir muestras al entrenamiento en la CNN. La arquitectura de la CNN consiste en dos capas convolucionales y dos capas totalmente conectadas, para el proceso de aprendizaje se utilizó el optimizador Adam con un ritmo de aprendizaje de 0.001. Luego, se diseñó una capa de costo, la cual fue implementada en la capa final de la red ResNet-50. Los resultados de la red se comparan con una detección basada en la ubicación del centro de la región de interés, obteniendo una precisión del 75.29% con la CNN y un 98.46% con la transferencia de aprendizaje. Por otro lado, R. Kumar *et al.* implementan una CNN de 12 capas. Las imágenes de entrenamiento son de 50x50x3. La

primera capa convolucional del modelo tiene 32 filtros de 3x3 y la función de activación ReLU, seguida de una capa de pooling de 2x2 y, por último, una capa de normalización para reducir el cambio de covarianza. El resultado obtenido fue del 98.23% de exactitud. Finalmente, en [14] se modifica la red neuronal YOLO para predecir la ubicación de los objetos detectados.

Se utiliza la transformada de Hough y el espacio de color HSV para la detección del parásito *Giardia Lamblia* en [15]. La detección se basa en el cumplimiento de tres condiciones: la densidad de los píxeles debe estar alrededor de los valores indicados por un manual de diagnóstico de parásitos intestinales, que resulta en un rango entre 60 y 71 píxeles; la segunda condición es que los objetos de interés deben presentar cierta estructura y la última condición consiste en que el objeto de interés debe tener ciertos valores en el canal de saturación del espacio de color HSV.

T. Fatima y M. Farid proponen un método basado en la intensidad de los píxeles para la detección de parásitos [16]. El método tiene como primer paso el mejorar la imagen mediante el uso de un filtro bilateral. El resultado del paso anterior se convierte a escala de grises y se hace una detección de contornos. Una vez se tienen los contornos, se realiza una umbralización adaptativa, donde el valor de cada píxel fue determinado mediante métodos estadísticos, los cuales consideran el valor de intensidad de los píxeles vecinos. A continuación, se remueven algunos bordes innecesarios mediante la conectividad N8. Finalmente, se establece un umbral para detectar si una célula está infectada o no, basado en la cantidad de píxeles con un cierto valor de umbral. El algoritmo presentó una precisión de 0.9466 y una exactitud de 0.9180.

En [17] se presenta un método para la detección de malaria basado en la entropía. Inicialmente, se hace un mejoramiento a la imagen mediante la ecualización del histograma seguido de un filtro de mediana para reducir ruido y suavizar la imagen. A la imagen procesada se le aplica la operación morfológica *bottom-hat* para remover objetos no deseados que puedan ser más grandes que los propios parásitos, que son el objeto de interés. Seguidamente, se realiza una umbralización adaptativa considerando la entropía. Este método

utiliza dos distribuciones de probabilidad: para el fondo y para el objeto de interés. Si f_i es la frecuencia que se presenta un nivel de gris en la imagen, se tiene la probabilidad de encontrar un pixel en el nivel i con

$$p_p(i) = \frac{f_i}{N} \quad (1.8)$$

La probabilidad del objeto es una función que depende del umbral t

$$p_o(t) = \sum_{i=t+1}^{L-1} p(i) \quad (1.9)$$

La entropía del objeto es

$$H_o(t) = - \sum_{i=t+1}^{L-1} \frac{p(i)}{p_o(t)} \ln \left(\frac{p(i)}{p_o(t)} \right) \quad (1.10)$$

Por otra parte, la probabilidad del fondo está dada por

$$p_f(t) = \sum_{i=0}^t p(i) \quad (1.11)$$

La entropía del fondo es

$$H_f(t) = - \sum_{i=0}^t \frac{p(i)}{p_f(t)} \ln \left(\frac{p(i)}{p_f(t)} \right) \quad (1.12)$$

Una vez se calculen ambas entropías, se obtiene el valor de umbral t mediante la resta de la entropía del fondo y la entropía del objeto. Después de esto, se realizan las operaciones morfológicas de dilatación y apertura, para aplicar luego el método de ACWE, ya que resuelve el problema para objetos con bordes no claros. Se concluye el algoritmo con la operación de cierre en la imagen con el método de ACWE. Se tuvieron 9.28 falsos positivos por imagen y 0.3 falsos negativos por imagen.

1.2.4 Extracción de características

H. Nugroho *et. al* proponen una clasificación de parásitos mediante la extracción de características de textura [18]. La investigación presentada aquí se divide en cuatro partes: mejoramiento de la imagen, segmentación, extracción de características y clasificación. En el mejoramiento de la imagen, se mejora el contraste, se filtra la imagen con un filtro de mediada de 3x3 y se cambia el espacio de color de RGB a HSV. En la etapa de segmentación se aplica el algoritmo de K-means para agrupar las células del fondo y se prosigue a la aplicación de operaciones morfológicas para mejorar la segmentación. En la extracción de características se obtienen 6 características de textura: la media, que se refiere al promedio de la intensidad de los pixeles, la desviación estándar, que da información sobre la varianza de la imagen, la energía, la cual es una medida de la intensidad de los pixeles, la oblicuidad, que es una medición de la asimetría de los valores de los pixeles, la entropía, que indica la complejidad de la imagen, la suavidad, que indica la suavización presente en la imagen y, finalmente, la curtosis, que es una medida para saber si la intensidad de la imagen es un pico o es plano relativa a la distribución normal. Estas características fueron clasificadas por medio de un perceptrón multicapa, obteniendo una precisión del 85% para los trofozoítos, un 81% de precisión para el parásito schizont y un 96.7% para los gametocitos.

Se presenta en [19] un método para la identificación de parásitos utilizando características de momento invariante. Se tienen 30 imágenes en RGB de 300x300 pixeles obtenidas con la base de datos DPDx. Primero, se realiza un procesamiento en las imágenes que consiste en cambiar el espacio de color a HSV, binarización y la aplicación de las operaciones morfológicas de apertura y cierre. Después, se extraen 7 momentos invariantes basados en

$$n_{pq} = \frac{u_{pq}}{u_{00}^w} \quad (1.13)$$

para

$$\begin{aligned} w &= \lceil (p+q)/2 + 1 \rceil \\ (p+q) &= 2, 3, 4, \dots \end{aligned} \quad (1.14)$$

Donde

$$u_{pq} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (i - \bar{i})^p (j - \bar{j})^q I(i, j) \quad (1.15)$$

$$\begin{aligned} \bar{i} &= \frac{m_{10}}{m_{00}} \\ \bar{j} &= \frac{m_{01}}{m_{00}} \end{aligned} \quad (1.16)$$

$$m_{pq} = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} i^p j^q I(i, j) \quad (1.17)$$

Se reduce la dimensionalidad presentada en la extracción de las características de momentos invariantes mediante el análisis de componentes principales (PCA). Finalmente, las características más significantes son clasificados mediante el método de k-vecinos más cercanos. 0

1.2.5 Clasificación sin métodos de aprendizaje profundo

En [20] se desarrolló una clasificación de parásitos en la sangre basada en Support Vector Machine (SVM). Se realizó un preprocesamiento en la imagen cambiándola de espacio de color. Se utilizó el histograma del canal de saturación para efectuar una segmentación de las bacterias y se prosiguió a extraer características fundamentadas en la textura, forma y color. Estas características fueron clasificadas con el SVM usando 2323 imágenes consiguiendo un 0.966% de precisión. El clasificador SVM también fue usado en [21], donde se siguió un modelo de combinación de DNN y SVM para la clasificación de bacterias en imágenes microscópicas. Como en cada imagen se presentan muchas bacterias, se recortaron dichas regiones para obtener más muestras, las cuales se concatenan y son la entrada de la red Inception V3 para reducir el tiempo de entrenamiento y la complejidad computacional y las características obtenidas de la DNN son clasificadas mediante SVM. Se usaron 800 imágenes para el entrenamiento y el clasificador SVM se apoyó en el kernel de función de base radial obteniendo un 96.07% de exactitud.

Un método basado en la clasificación por Naïve Bayes se utilizó en [22], para la detección de tumores. Se tienen 114 imágenes MRI donde 90 presentan tumores y 24 no. Primeramente, se efectuaron operaciones morfológicas para separar el cráneo de la imagen. Luego, el resultado se restó a la imagen original obteniendo, de esta manera, una imagen sin regiones no deseadas del cráneo. A continuación, se aplicó una umbralización basada en la máxima entropía, dadas las variaciones en intensidad del fondo y del primer plano en las imágenes. Después, se extrajeron 11 características: área, perímetro, excentricidad, diámetro equivalente, solidez, área convexa, longitud del eje mayor, longitud del eje menor, máxima intensidad de pixel, mínima intensidad de pixel y el valor promedio de los pixeles. Finalmente, se utilizó un clasificador de Bayes. Los resultados se evaluaron en 50 objetos; de ellos 13 fueron verdaderos positivos, 34 verdaderos negativos, no hubo falsos positivos y se detectaron 3 falsos negativos.

Clasificadores de k-vecinos más próximos (KNN) se encuentran en [23] y [24]. En el primero se atendió a la correcta detección de cuatro especies de *plasmodium*. Inicialmente, en la etapa del preprocesamiento, se aplicaron los siguientes filtros: de media geométrica, de mediana y la transformada discreta de curvelet. Se prosiguió con un agrupamiento usando k-medias, el cual se destinó al espacio de color L^*a^*b . Para elegir el agrupamiento donde se encuentra la región de interés se observó que dicha región es la que tiene menor número de pixeles, con esto se extrajeron las siguientes características: GLCM, GRLM, características de la forma (perímetro, área), dimensión fractal, características del histograma, patrones binarios locales, momentos de Hu y entropía. Estas características fueron reducidas para obtener las características más discriminantes, esto se realizó al aplicar ANOVA. Un total de 300 imágenes de muestra fueron usadas para clasificar 13 clases usando el clasificador de k-vecinos más próximos. El resultado fue de un 90.17% de precisión. También se utilizó el algoritmo SVM para comparación y este tuvo un desempeño del 84.2%. En el segundo, se realizó una detección de parásitos implementando diferentes clasificadores. Se tienen 130 imágenes obtenidas del sitio online Centre for Disease Control. Al comienzo, se filtraron los objetos que aparecen en las imágenes midiendo la cantidad de pixeles de cada uno y una consecuente umbralización. Luego, se aplicó una segmentación watershed y fue utilizada una

detección de círculos para detectar objetos de interés. Se usó SURF para representar las categorías de las imágenes. Después, se aplicó un agrupamiento de k-medias para cuantizar el espacio de características. Se utilizaron diversos clasificadores, sin embargo, el basado en subespacios KNN tuvo el mejor rendimiento con un 86.3% de precisión general.

1.2.6 Clasificación con métodos de aprendizaje profundo

En [25] se empleó aprendizaje profundo para la clasificación de tumores cerebrales. La base de datos usada se consiguió del sitio web de la escuela de medicina de Harvard. Primeramente, se hizo una segmentación con el método de agrupamiento Fuzzy C-Means. A continuación, se extrajeron características relevantes usando la transformada discreta de wavelet. Finalmente, se utilizó una separación de 3 niveles mediante la onduleta de Haar para extraer 1024 características, reducidas al utilizar análisis de componentes principales. La clasificación se realizó con una técnica de validación cruzada en una red neuronal profunda. El resultado conseguido fue de 0.97% en la métrica de *Measure*.

En [26] se realizó un reconocimiento de cáncer de pulmón usando redes neuronales profundas. Se cuentan con 4480 imágenes las cuales, posteriormente, son procesadas con el fin de obtener más muestras. Las capas de convolución son usadas para segmentar la imagen y para la localización del posible cáncer. El aumento y decremento de la dimensión de la imagen debido a la convolución y a la capa de *max pooling* es aprovechado para buscar la región sin incrementar el tiempo de procesamiento. Se obtuvo un 81% de precisión.

En [27] fue empleado un método para la clasificación de cáncer de seno con redes neuronales convolucionales. Inicialmente, las imágenes se normalizaron al restarle el promedio a cada pixel, luego se dividieron por su varianza. En total se cuenta con 2013 imágenes. La red profunda define la dimensión de las imágenes, luego extrae características de bajo nivel para obtener las de alto nivel. Además, cuenta con dos capas totalmente conectadas y una capa de softmax. Se seleccionó, debido a su eficiencia computacional, el optimizador ADAM. La exactitud fue del 0.9154.

En [28] se diseñó un modelo basado en una red neuronal convolucional codificadora automática. Se añadió a la operación de convolución la imagen reconstruida por el

decodificador. Luego, los valores son reconstruidos con la convolución codificadora. En esta etapa se consiguieron las características. La clasificación empleó, para calcular el gradiente, la función de costo de entropía cruzada. El resultado obtenido es del 95% de precisión.

En [29] se combinó una red neuronal convolucional y una red neuronal recursiva para la clasificación de células. Las muestras son 12,444 imágenes las cuales, mediante una matriz de rotación, son incrementadas para poseer más muestras. Se emplearon parámetros obtenidos con el conjunto de datos ImageNet para que sean los pesos con que se inicializó la CNN. La red RNN utilizó el método de memoria larga de corto plazo para determinar si la información es útil o no. A continuación, una capa unió las características extraídas de la CNN y la RNN mediante la multiplicación punto a punto de las matrices. La clasificación se realizó usando la función de costo de entropía cruzada. El resultado fue de 90.97% de exactitud con el submodelo Xception-LSTM.

Otros métodos que implementan transferencia de aprendizaje se encuentran en [30] y [31]. En [30] se propuso un algoritmo basado en la transferencia de aprendizaje para la detección y clasificación de cáncer de seno. Para ello, se pre entrenaron tres redes neuronales convolucionales: GoogLeNet, VGGNet y ResNet. Las tres arquitecturas fueron usadas para compartir sus propiedades; de esta manera, fue posible que una red aprendiera características genéricas de otra sin que se realizara un entrenamiento adicional. Después, las características de cada CNN se unificaron en una capa totalmente conectada para la clasificación. Finalmente, se comparó el resultado de cada arquitectura individualmente y con el método propuesto, obteniendo, este último, un 97.525% de precisión, un 3% más que el mejor resultado de las CNN individuales. En [31] se aprovechó la transferencia de aprendizaje para la clasificación de células de malaria. El conjunto de datos tiene una cantidad de 27,558 imágenes de muestras, las cuales están disponibles en el sitio web de National Library of Medicine, en las muestras hay imágenes de células infectadas y no infectadas. Primeramente, una imagen en el espacio de color RGB es la entrada, posteriormente los pesos pre entrenados fueron importados a la red ResNet50; la información de entrada es entrenada con dichos pesos. En el modelo la única capa que estaba aprendiendo fue la capa densa. Es importante destacar que fue utilizada la función de activación sigmoid, dado que la clasificación es

puramente binaria. La métrica de la evaluación del entrenamiento y de validación fueron del 95.91% y del 95.4% respectivamente.

CAPÍTULO 2 SEGMENTACIÓN

En este capítulo se presentará el procedimiento seguido para la correcta segmentación de los parásitos en las imágenes de muestra. En la Figura 2.1 se muestran algunos ejemplos de las imágenes de muestra y los círculos rojo indican cuál es el parásito.

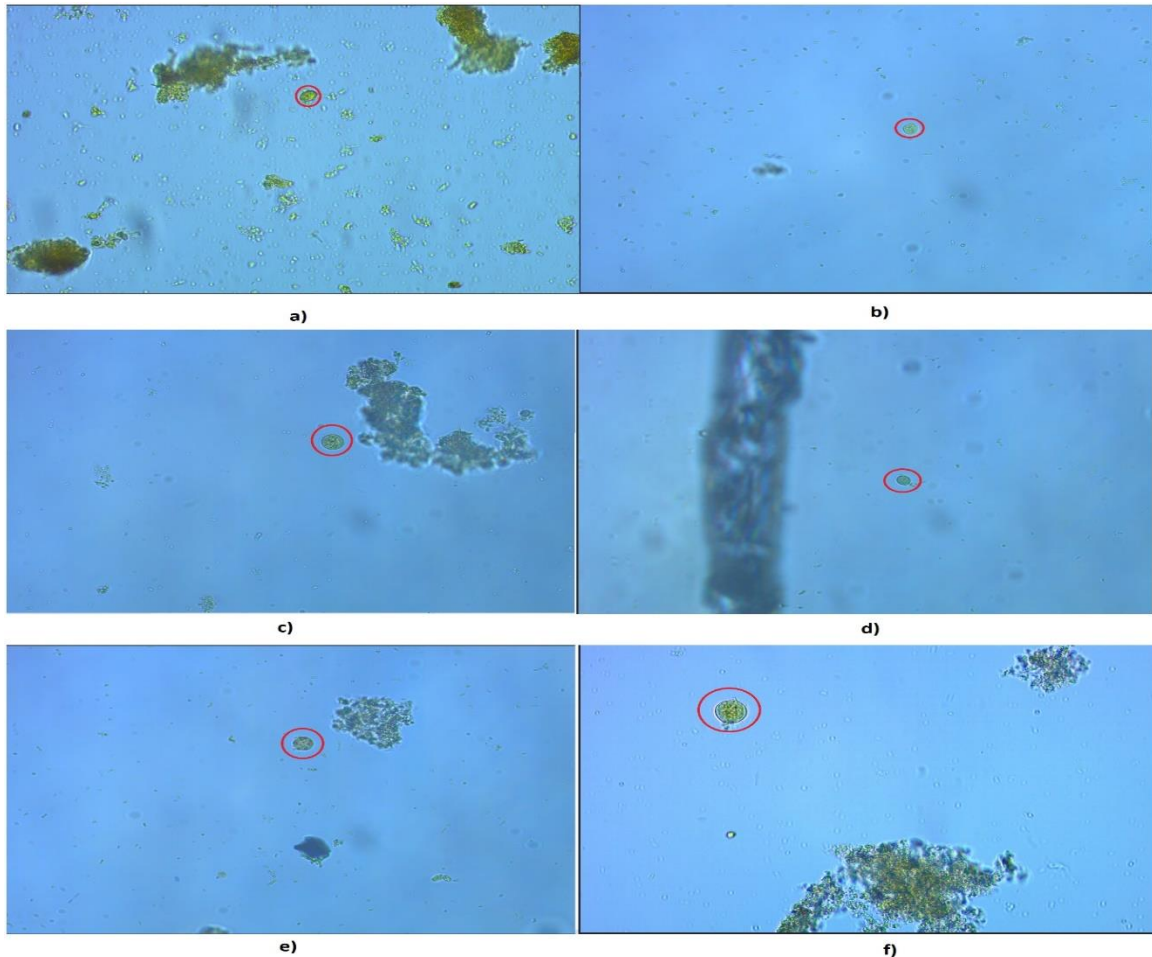


Figura 2.1. Imágenes de muestra. a) Giardia Lamblia, b) Endolimax Nana, c) Entamoeba Histolytica, d) Chilomastix Mesnili, e) Blastocystis Hominis, f) Entamoeba Coli.

2.1 Procesamiento de la imagen

Las imágenes de muestra presentan un fondo de color azul mayormente, mientras que los objetos de interés muestran una combinación de colores claramente distinta (Figura 2.2).

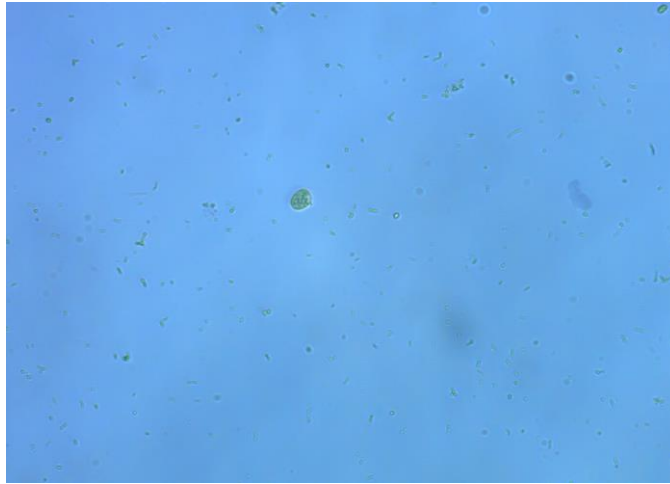


Figura 2.2. Imagen original, fondo azul, parásito en otro color.

Inicialmente se cambia el espacio de color de RGB a HSV. Donde H_{sv} es el matiz, representado como un ángulo que va de 0° a 360° que se divide en los tres matices RGB, S es la saturación, que es la distancia desde el centro a la circunferencia (si la distancia es 0 no hay saturación), V es el valor de la altura del cono. En la Figura 2.3 se muestra este espacio de color.

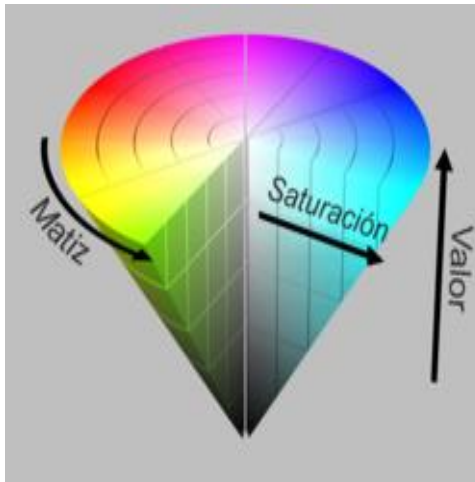


Figura 2.3. Espacio de color HSV.

La transformación de RGB a HSV está dada por (2.1), (2.2) y (2.3). Una vez que se efectúa el cambio en el espacio de color se multiplica la imagen por dos y se obtiene la Figura 2.4. El fondo se debe a que se satura el fondo del espacio HSV y los valores de los píxeles corresponden al blanco. Sin embargo, el objeto de interés no se va hasta el blanco, es por eso

por lo que se ve diferente. Se elige el canal S y se obtiene la Figura 2.. Se aprecia que el fondo mayormente es blanco y los objetos de interés están en tonos de gris.

$$H = \begin{cases} nd & si \quad MAX = MIN \\ 60^\circ \left(\frac{G-B}{MAX-MIN} \right) + 0^\circ & si \quad MAX = R, G \geq B \\ 60^\circ \left(\frac{G-B}{MAX-MIN} \right) + 360^\circ & si \quad MAX = R, G < B \\ 60^\circ \left(\frac{B-R}{MAX-MIN} \right) + 120^\circ & si \quad MAX = G \\ 60^\circ \left(\frac{R-G}{MAX-MIN} \right) + 240^\circ & si \quad MAX = B \end{cases} \quad (2.1)$$

$$S = \begin{cases} 0 & si \quad MAX = 0 \\ 1 - \frac{MIN}{MAX} & eoc \end{cases} \quad (2.2)$$

$$V = MAX \quad (2.3)$$

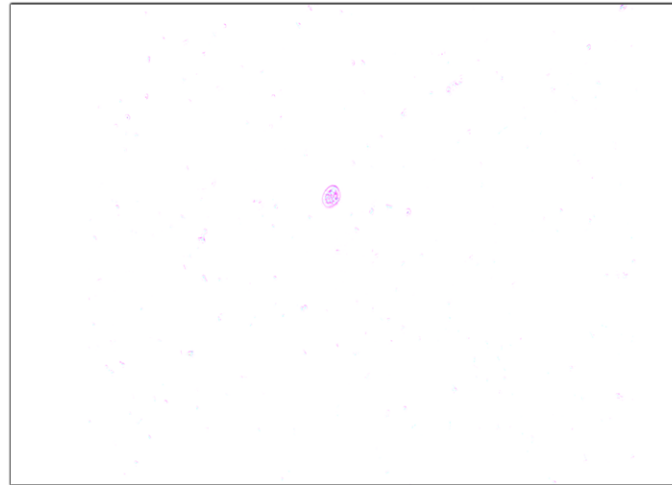


Figura 2.4. Imagen en el espacio de color HSV multiplicada por dos.

Luego se realiza una binarización en el canal S y se obtiene el complemento del canal S binarizado (Figura 2.6), y se prosigue a efectuar la operación morfológica de dilatación

(Figura 2.7). Sea A el conjunto de pixeles de interés y B el elemento estructural y E los pixeles igual a 1 en el plano de interés, la dilatación está definida como

$$A \oplus B = \left\{ E \mid \left[\left(\hat{B} \right)_E \cap A \right] \subseteq A \right\} \quad (2.4)$$

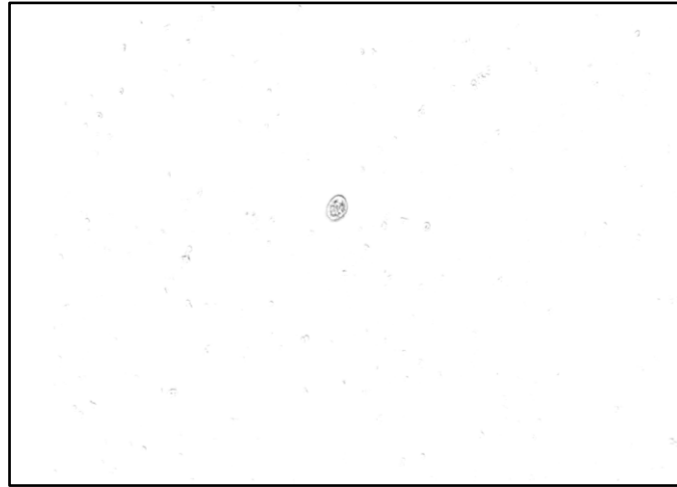


Figura 2.5. Canal S.

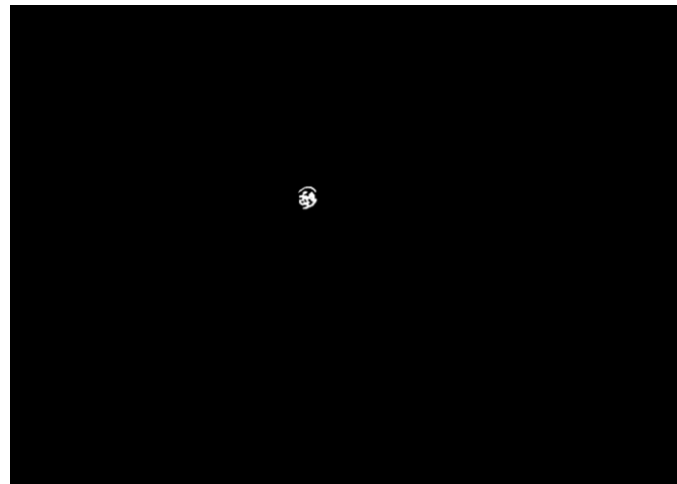


Figura 2.6. Complemento de la binarización del canal S.

Una vez se tiene una máscara donde están ubicados los objetos de interés, se obtienen las coordenadas de la región que contiene a cada objeto. Con esto se corta la imagen para obtener cada objeto individualmente. En la Figura 2.8 se muestra un parásito detectado.

En la Tabla 2.1 se observan los resultados de parásitos detectados para cada clase.

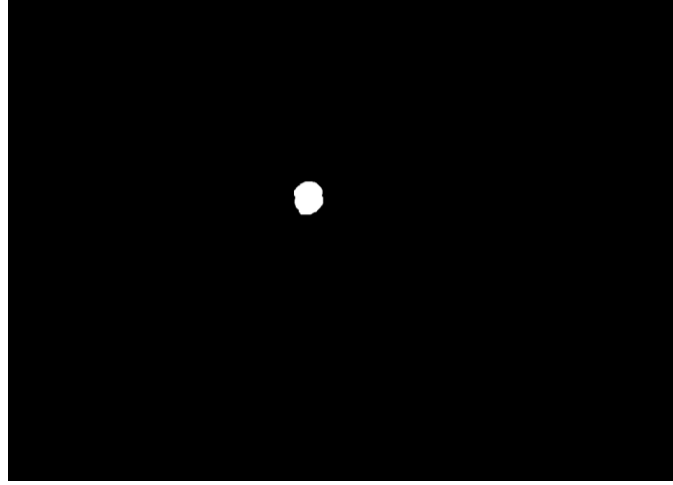


Figura 2.7. Dilatación.

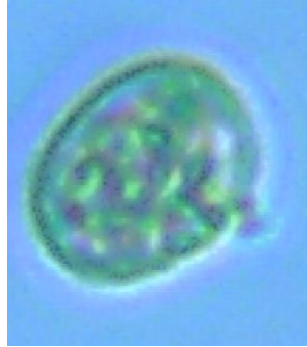


Figura 2.8. Parásito detectado.

2.2. Segmentación con k means

El método de k-means es un algoritmo de agrupamiento que busca minimizar la distancia promedio entre puntos del mismo grupo. Sea $\mathcal{X}$ un conjunto real de n datos en $\mathbb{R}^d$ este algoritmo busca minimizar la función

$$\phi = \sum_{x' \in \mathcal{X}} \min_{c \in C} \|x' - c\|^2 \quad (2.5)$$

Dicho algoritmo consta de 4 pasos. En el primero se seleccionan k centros $C = \{c_1, \dots, c_k\}$. En el segundo, se establece el grupo C_i como el conjunto de puntos en $\mathcal{X}$ que están más cerca de c_i que de c_j para todo $j \neq i$. En el tercero se establece c_i como el centro de masas de todos los puntos en C_i

Tabla 2.1. Desempeño en imágenes con parásitos.

Parásito	Cantidad de imágenes	Cantidad de parásitos	Parásitos detectados	No parásitos detectados
Chilomastix mesnili	33	33	29	135
Entamoeba histolytica	83	86	77	730
Endolimax nana	11	11	11	18
Blastocystis hominis	161	174	153	946
Entamoeba Coli	41	45	36	682
Giardia Lamblia	54	90	70	1200
Enteromonas hominis	27	266	225	844

$$c_i = \frac{1}{|C_i|} \sum_{x' \in C_i} x' \quad (2.6)$$

Finalmente se repite el paso 2 y 3 hasta que C no cambie. En la Figura 2.9 se muestra el resultado de la segmentación para el parásito detectado.

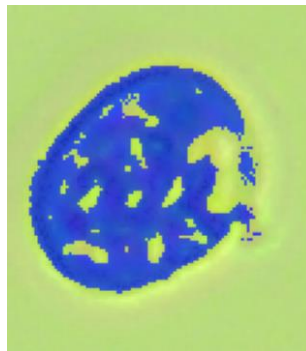


Figura 2.9. Segmentación mediante K means.

Una vez que se tiene la segmentación se realiza una dilatación en la máscara para obtener el parásito segmentado. En la Figura 2.10 se muestran ejemplos de la segmentación para cada tipo de parásito y ejemplos de no parásitos.

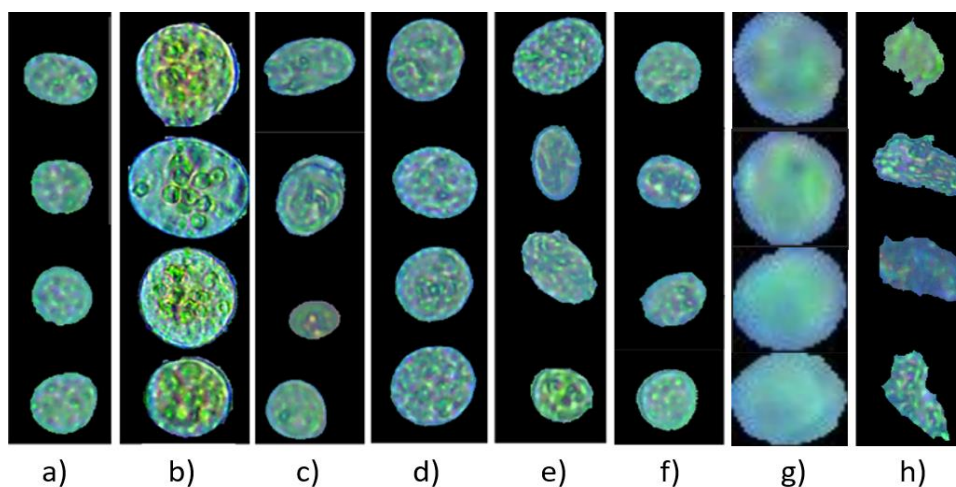


Figura 2.10. Resultado de la segmentación, a) *Blastocystis hominis*, b) *Entamoeba coli*, c) *Chilomastix mesnili*, d) *Entamoeba histolytica*, e) *Giardia lamblia*, f) *Endolimax nana*, g) *Enteromonas hominis*, h) No parásito.

En este capítulo se presentó un método para la detección de objetos de interés y su segmentación. En esta investigación se tiene la premisa de que las imágenes son estandarizadas: un fondo similar en todas las imágenes y los objetos de detección presentan un color similar. Sin embargo, entre las propias muestras existen imágenes que difieren de las demás: con muchos más objetos de interés. Con lo cual, existe un área de investigación para la detección y segmentación de parásitos en imágenes complejas. Además, existen situaciones donde el parásito está unido a objetos que no son fondo, y esto complica también su segmentación. El trabajo aquí realizado funciona para la mayoría de las imágenes de muestra, pues el interés de esta tesis es la clasificación utilizando aprendizaje profundo, no sobre detección en imágenes muy complejas.

CAPÍTULO 3 AUMENTO DE DATOS

En este capítulo se muestran las transformaciones a las imágenes de parásitos, así como la generación de datos mediante las redes generativas adversarias (GAN), con el fin de contar con más muestras. Para entrenar la GAN se hace un aumento de datos mediante técnicas clásicas.

3.1 Generación de datos

Los modelos de aprendizaje de máquina con poca generalización tienden a sobre ajustarse al conjunto de entrenamiento. Esto es, el algoritmo de aprendizaje aprende de memoria las imágenes de entrenamiento, y cuando se le presentan las imágenes de validación tiene un mal desempeño. Se han propuesto diversas técnicas para reducir el sobre ajuste: regularización del dropout, normalización por lotes, transferencia de aprendizaje, preentrenamiento. Sin embargo, el aumento de datos se enfoca en la raíz del problema del sobre ajuste: el conjunto de datos. La premisa consiste en que más información pueda ser extraída del conjunto original de entrenamiento si se incrementa la cantidad de muestras en el conjunto de entrenamiento a partir de realizar transformaciones en dichas muestras. Estas transformaciones incluyen transformaciones geométricas, de color, ruido aleatorio, transferencia de estilo y redes adversarias [32].

3.2 Transformaciones tradicionales

Las transformaciones geométricas consisten en la manipulación de la imagen original al operar sobre ella. Es decir, se parte de una imagen del conjunto de datos y se aplican transformaciones a sus coordenadas, a la intensidad de los píxeles y adición de ruido, principalmente.

Las transformaciones implementadas en esta investigación son: rotación, aumento de contraste, adición de ruido, inversión de ejes y aumento de brillo. En la Figura 3.1 se observan los resultados de las transformaciones elegidas en esta tesis.

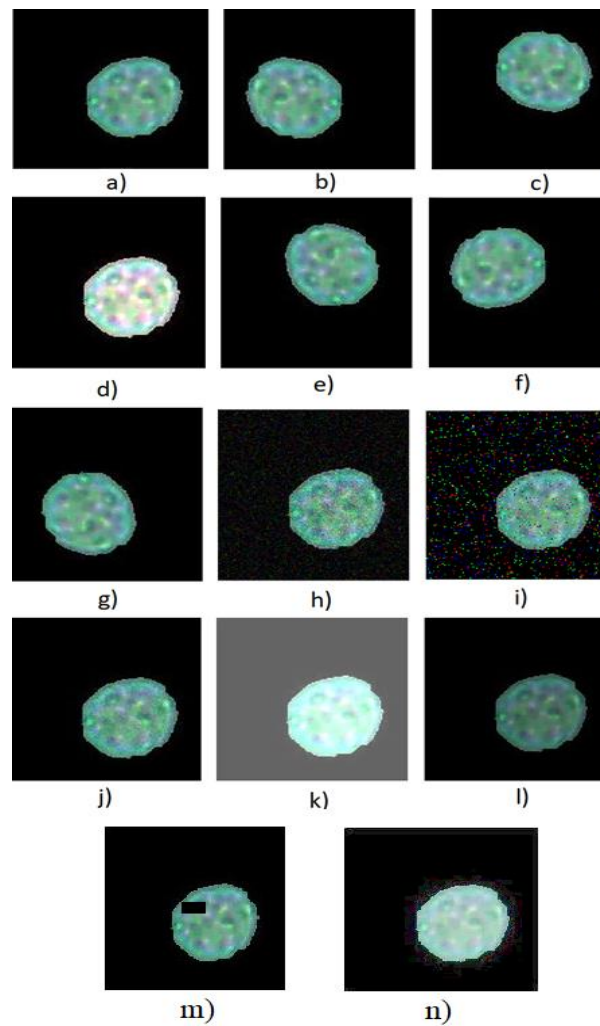


Figura 3.1. Aumento de datos mediante transformaciones, a) Imagen original, b) Imagen reflejada en el eje y, c) Imagen reflejada en el eje x, d) Aumento de contraste, e) Imagen rotada 90°, f) Imagen rotada 180°, g) Imagen rotada 270°, h) Imagen con ruido gaussiano, i) Imagen con ruido sal y pimienta, j) Imagen con ruido Poisson, k) Aumento de brillo, l) Disminución de brillo, m) Borrado aleatorio, n) Corrección gamma.

3.3 Redes generativas adversarias

Las redes generativas adversarias (GAN) es un modelo propuesto por Goodfellow *et. al* en 2014 [33], en el cual se entrenan dos modelos: un modelo generativo G que tiene la distribución de los datos generados y un discriminador D que estima la probabilidad de que una muestra provenga del conjunto de entrenamiento y no de G . En [34], [35], [36], [37], [38] y [39] se han basado en este modelo para el aumento de datos. En la Figura 3.2 se muestra la arquitectura general de una GAN.

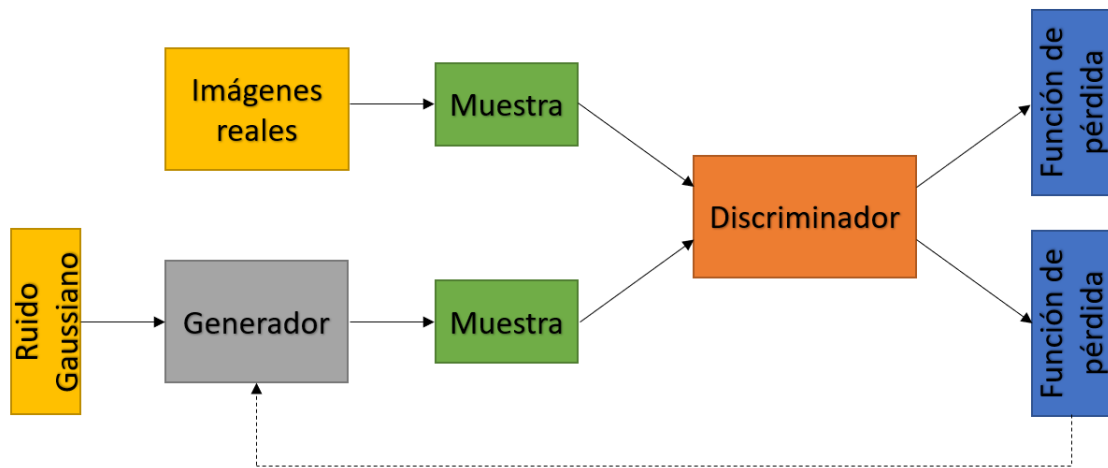


Figura 3.2. Modelo de una red generativa adversaria

Los modelos generativos intentan generar datos que se asemejen a la entrada, mientras que los modelos discriminativos solamente intentan diferenciar las entradas. En la Figura 3.3 se muestra cómo trabaja (a grandes rasgos) cada tipo de modelo. El discriminativo busca separar regiones de un hiperplano, de manera que en cada subregión corresponda un dígito 0 o 1, de tal modo que una entrada la va a asociar a cada subregión y de esta manera saber si dicha entrada es un 0 o 1. Sin embargo, los modelos generativos no realizan esto, sino que buscan generar una distribución que se asemeje a los datos.

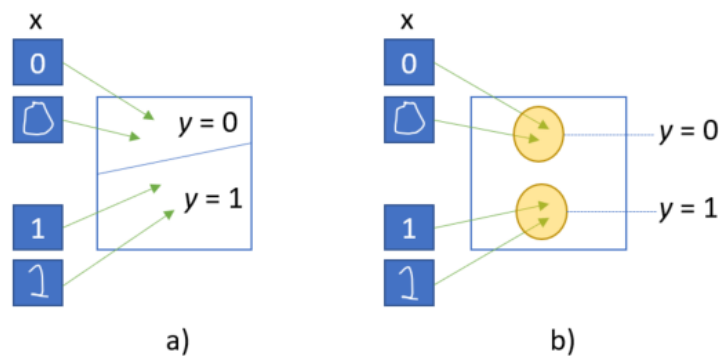


Figura 3.3. Tipos de modelos, a) discriminativos, b) generativos.

Mientras que el modelo discriminativo busca la probabilidad condicional $p(y|x)$ que indica cuál es la probabilidad de que la entrada x pertenezca a la clase y , el modelo generativo busca la probabilidad $p(x,y)$.

3.3.1. Entropía en física

Roger Penrose menciona en su libro Los ciclos del tiempo [40]: “la segunda ley de la termodinámica no es una igualdad, sino una desigualdad que afirma simplemente que cierta magnitud conocida como la entropía de un sistema aislado –que es una medida del desorden, o ‘aleatoriedad’, del sistema- es mayor (o al menos no menor) en instantes posteriores que lo era en instantes anteriores.” Lo importante de esta cita para esta investigación es el hecho de considerar una cantidad que mide cuánto desorden existe en un sistema. Entonces, matemáticamente, la entropía está definida matemáticamente como

$$S_F = k \log(V_{SIS}) \quad (3.1)$$

En este caso no es necesario explicar qué es cada variable para comprender los conceptos posteriores. Únicamente, es de interés mencionar que V_{SIS} representa una cantidad de estados posibles y que es un número enorme. Por esta razón se utiliza el logaritmo, ya que los números que se manejan son grandísimos. Y más importante aún, el logaritmo se emplea para que cada sistema tenga una entropía definida y medible, ya que

$$\log(A_S B_S) = \log(A_S) + \log(B_S) \quad (3.2)$$

Con lo cual, si se tienen dos sistemas independientes entre sí, la entropía de ambos es la suma individual de entropías. Y esto es fundamental para la definición de la entropía en teoría de la información.

3.3.2. Entropía en teoría de la información

En la teoría de la información, el objetivo del transmisor es comunicar un mensaje (a través de un canal) al receptor. El contenido de la información es una medición de cuánta incertidumbre le resuelve al receptor. Es decir, si un mensaje da información sobre un suceso que tiene dos posibles resultados, entonces el contenido de la información podrá ser codificado con un solo bit; con lo cual, el receptor al recibir dicho bit le resuelve una certidumbre del 50%, mientras más posibles resultados, la cantidad para representar dicha información será mayor y la incertidumbre resuelta al receptor es la probabilidad de que dicho

suceso ocurra (mientras más posibles resultados, mayor será la información que obtenga el receptor; en el caso de 3 sucesos, el receptor tiene un 33.3% de acertar al resultado, con lo cual saber cuál resultado ocurrió realmente le resuelve un 66.6% de incertidumbre).

Entonces, la entropía en el paradigma de la teoría de la información es el número esperado de bits de información contenido en cada mensaje; esto es, la cantidad de información útil que se esperaba obtener. Matemáticamente está definida por

$$H(p) = \sum_x p(x) \log_2 \left(\frac{1}{p(x)} \right) \quad (3.3)$$

Donde p es una distribución de probabilidad. Al igual que en la ecuación (3.1), la formalización contiene un logaritmo (en este caso base 2 debido al uso de bits para representar la información) porque la entropía de dos distribuciones independientes debe ser la suma de cada entropía.

3.3.3. Entropía cruzada

En la expresión (3.3) se tiene de antemano la distribución de probabilidad, si se conociera no haría falta tener un clasificador. Como los modelos de clasificación buscan generar una distribución a partir de los datos de entrada, se requiere otra expresión aparte de (3.3) para cuantificar la entropía. Para esto se utiliza la entropía cruzada, la cual es una medición de similitud entre dos distribuciones de probabilidad, definida como

$$H(p, q) = - \sum_x p(x) \log(q(x)) \quad (3.4)$$

Donde q es una distribución de probabilidad. Si p y q fuesen las distribuciones del generador y discriminador de la GAN (como el discriminador y el generador de la GAN clasifican de manera binaria), se llega a la entropía cruzada binaria

$$H = y \log(p(y)) + (1 - y) \log(1 - p(y)) \quad (3.5)$$

Donde y es la etiqueta asociada al dato x . Que en el caso de las redes GAN será 0 o 1, dependiendo si el dato viene de la distribución de los datos o del generador.

3.3.4. Divergencia de Kullback-Leibler

Como ya se mencionó, la entropía estima cuantitativamente la cantidad de información que hay en los datos. Ahora, se pretende cuantificar qué tanta información se pierde cuando se sustituye la distribución observada por una distribución aproximada. Para esto se utiliza la divergencia de Kullback-Leibler

$$D_{KL}(p \parallel q) = \sum_x p(x) [\log(p(x)) - \log(q(x))] \quad (3.6)$$

La divergencia D_{KL} es el valor esperado de la diferencia de dos distribuciones. Tomando el ejemplo de la Figura 3.4, se observan tres distribuciones: en color rojo está la distribución observada, la distribución real de los datos; en color azul una estimación mediante una distribución uniforme y en naranja una estimación con una distribución binomial. Si se quisiera saber cuál de estas dos estimaciones pierden menor cantidad de información para representar la distribución real, se calcula D_{KL} .

$$D_{KL}(\text{Observada} \parallel \text{Uniforme}) = 0.338$$

$$D_{KL}(\text{Observada} \parallel \text{Binomial}) = 0.477$$

Estos resultados indican que si se estima la distribución real mediante la distribución uniforme, se pierde menos cantidad de información.

Sin embargo, D_{KL} no puede ser usada como una métrica de distancia, ya que no es simétrica $D_{KL}(p \parallel q) \neq D_{KL}(q \parallel p)$ si p y q son distintas. Otro problema de D_{KL} es que no es suave (tiene una discontinuidad). Reescribiendo la ecuación (3.6) utilizando propiedades de los logaritmos tenemos

$$D_{KL}(p \parallel q) = \sum_x p(x) \left[\log \left(\frac{p(x)}{q(x)} \right) \right] \quad (3.7)$$

Se observa que si $q(x)$ es 0, entonces hay una indeterminación para D_{KL} . La única manera de que sea continua cuando q toma el valor de 0 es que p también sea cero, ya que

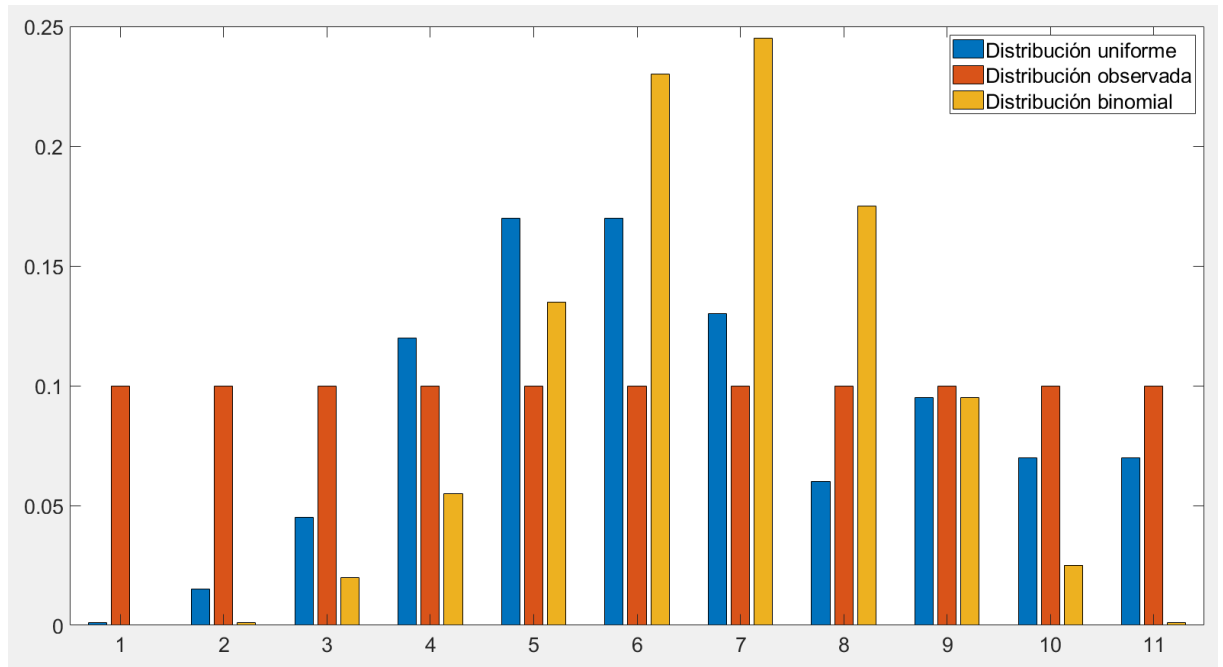


Figura 3.4, Estimación de una distribución observada mediante una distribución uniforme y una distribución binomial.

$$\lim_{p \rightarrow 0} p(x) \left[\log \left(\frac{p(x)}{q(x)} \right) \right] = 0 \quad (3.8)$$

3.3.5. Divergencia de Jensen-Shannon

Para lidiar con los dos inconvenientes de D_{KL} : no ser simétrica y no ser suave (tiene una indeterminación cuando q es cero), se tiene la divergencia de Jensen-Shannon (D_{JS}), la cual sí puede ser utilizada como una métrica de distancia entre distribuciones. D_{JS} está definida como

$$D_{JS}(p \parallel q) = \frac{1}{2} D_{KL}(p \parallel M) + \frac{1}{2} D_{KL}(q \parallel M) \quad (3.9)$$

Donde

$$M = \frac{1}{2}(p + q) \quad (3.10)$$

A partir de (3.9) es fácil observar que es simétrica y que es suave porque ni p ni q son divisores en el logaritmo al obtener D_{KL} .

3.3.6. Discriminador

Una vez definidos los conceptos necesarios, se abordará la matemática del discriminador. Como se mencionó, el objetivo del discriminador es etiquetar correctamente las imágenes generadas como *falso* y los datos verdaderos como *cierto*. Se define una función de costo genérica

$$L_D = \text{Error}(D(x), 1) + \text{Error}(D(G(z)), 0) \quad (3.11)$$

Donde $D(x)$ es la distribución de los datos reales, $D(G(z))$ es la distribución de los datos generados por el generador, z es un vector latente (ruido). Se utiliza como función de costo la entropía cruzada (3.6). Para saber cuál es la función de costo del discriminador se parte de (3.5), cuando se entrene el discriminador la etiqueta que provenga de los datos reales $D(x)$ será igual a 1, y $p(y)$ será igual a $D(x)$, con lo cual, al sustituir esos valores en (3.5) queda

$$L_D = \log(D(x)) \quad (3.12)$$

3.3.7 Generador

El objetivo del generador es lograr que el discriminador no logre distinguir entre los datos generados y los datos verdaderos. Definiendo una ecuación de costo genérica

$$L_G = \text{Error}(D(G(z)), 0) \quad (3.13)$$

De igual manera se parte de la entropía cruzada. Cuando se entrene el generador, las etiquetas y serán igual a 0 y $p(y)$ será igual a la distribución de los datos generados. Sustituyendo esos valores en (3.5)

$$L_G = \log(1 - D(G(z))) \quad (3.14)$$

3.3.8. Problema min-max

Una vez se tienen las dos funciones de costo (3.12) y (3.14) se juntan en una misma expresión

$$\log(D(x)) + \log(1 - D(G(z))) \quad (3.15)$$

El discriminador debe clasificar correctamente y el generador compite para que el discriminador se equivoque, con lo cual se tiene un problema de max-min. Con lo cual (3.15) se reescribe como

$$\min_G \max_D \left\{ \log(D(x)) + \log(1 - D(G(z))) \right\} \quad (3.16)$$

Sin embargo, la expresión (3.15) no considera todo el conjunto de datos. Para hacerlo se debe obtener el valor esperado de (3.15).

$$E_{x \sim p_{data}} [\log(D(x))] + E_{z \sim p_z} [\log(1 - D(G(z)))] \quad (3.17)$$

Donde p_{data} es la distribución de los datos originales, p_z es la distribución del ruido de entrada. Con lo cual, la expresión (3.17) indica que se obtiene el valor esperado de los datos originales de la función de costo del discriminador y el valor esperado de los datos generados de la función de costo del generador.

3.3.9. Entrenamiento

Se puede definir (3.17) como una función de G y D (generador y discriminador)

$$V(G, D) = E_{x \sim p_{data}} [\log(D(x))] + E_{z \sim p_z} [\log(1 - D(G(z)))] \quad (3.18)$$

Sin embargo, no es de interés la distribución del ruido de entrada, sino la distribución de los datos generados. Con lo cual el segundo término en (3.18) se cambia para obtener el valor esperado del conjunto de datos generados.

$$V(G, D) = E_{x \sim p_{data}} [\log(D(x))] + E_{x \sim p_g} [\log(1 - D(x))] \quad (3.19)$$

Donde p_g es la distribución de los datos generados.

La etapa del entrenamiento comienza con el discriminador, con lo cual es necesario encontrar el valor óptimo de D , D^* . Para esto se deriva parcialmente (3.19) con respecto a D

$$\frac{\partial}{\partial D} V(G, D) = \frac{\partial}{\partial D} \left\{ E_{x \sim p_{data}} [\log(D(x))] + E_{x \sim p_g} [\log(1-D(x))] \right\} \quad (3.20)$$

$$\frac{\partial}{\partial D} V(G, D) = \frac{p_{data}(x)}{D(x)} - \frac{p_g(x)}{1-D(x)} \quad (3.21)$$

El valor óptimo de D , D^* , es el que hace que la ecuación (3.21) sea igual a 0. Al imponer esta condición y despejando D se obtiene que el valor óptimo D^* es

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \quad (3.22)$$

A continuación, se sigue con el generador, para lo cual se toma el discriminador fijo. Primeramente, se sustituye el valor óptimo D^* en (3.19)

$$V(G, D^*) = E_{x \sim p_{data}} \left[\log \left(\frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right) \right] + E_{x \sim p_g} \left[\log \left(\frac{p_g(x)}{p_{data}(x) + p_g(x)} \right) \right] \quad (3.23)$$

Desarrollando (3.23) mediante propiedades de los logaritmos

$$\begin{aligned} V(G, D^*) &= E_{x \sim p_{data}} \left[\log(p_{data}(x)) - \log(p_{data}(x) + p_g(x)) + \log(2) \right] \\ &+ E_{x \sim p_g} \left[\log(p_g(x)) - \log(p_{data}(x) + p_g(x)) + \log(2) \right] - \log(4) \end{aligned} \quad (3.24)$$

Se agrupan los términos del logaritmo de 2

$$\begin{aligned} V(G, D^*) &= E_{x \sim p_{data}} \left[\log(p_{data}(x)) - \log \left(\frac{p_{data}(x) + p_g(x)}{2} \right) \right] \\ &+ E_{x \sim p_g} \left[\log(p_g(x)) - \log \left(\frac{p_{data}(x) + p_g(x)}{2} \right) \right] - \log(4) \end{aligned} \quad (3.25)$$

La ecuación (3.25) se asemeja a la ecuación (3.6), ya que se tiene el valor esperado de una diferencia de distribuciones de probabilidad. Con lo cual se reescribe (3.25) en función de D_{KL}

$$V(G, D^*) = D_{KL} \left(p_{data} \parallel \frac{p_{data} + p_g}{2} \right) + D_{KL} \left(p_g \parallel \frac{p_{data} + p_g}{2} \right) - \log(4) \quad (3.26)$$

Ahora, la ecuación (3.26) es la (3.9) más una constante, reescribiendo (3.26) en términos de D_{JS} .

$$V(G, D^*) = 2D_{JS}(p_{data} \parallel p_g) - \log(4) \quad (3.27)$$

Se observa en (3.27) que existe un único mínimo global en $-\log(4)$, este mínimo se presenta si y solo si las distribuciones p_{data} y p_g son iguales. Ya que si son iguales D_{KL} es cero porque el cociente entre las distribuciones sería 1 y el logaritmo de 1 es 0. Con lo cual el aporte de D_{JS} en (3.27) es nulo y queda $-\log(4)$. Así pues, las redes GAN buscan igualar lo más posible la distribución generada a la distribución de los datos reales.

3.3.10 Elementos de la arquitectura de la GAN

Para comprender mejor cómo funciona la GAN, se explicarán algunos conceptos importantes que la componen.

3.3.10.1 Capa convolucional

Esta capa se enfoca en el uso de filtros que ajustan sus valores y realizan una convolución sobre la entrada en toda su dimensionalidad espacial [41]. Estos filtros de dimensión pequeña producen un mapa de activación al realizar la convolución sobre la entrada. Como se observa en la Figura 3.5.

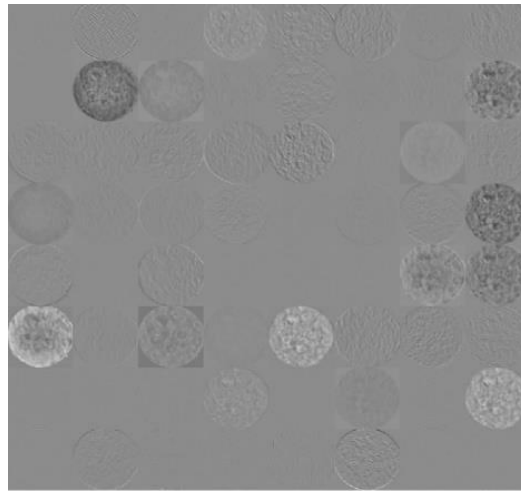


Figura 3.5. Activaciones de la primera capa de convolución de la red Alexnet ante una imagen de parásito.

La red aprenderá de los filtros que se disparen cuando encuentren una característica en una posición dada de la entrada. Además de esto, la capa de convolución reduce significativamente la complejidad de la entrada.

3.3.10.2 Capa convolucional traspuesta

La convolución traspuesta surge de la necesidad de una transformación que vaya en dirección opuesta a la operación de convolución. Esta capa funciona como un sobre muestreo, donde se aumenta la dimensionalidad de la entrada. La convolución traspuesta se utiliza como una transformación la cual funciona como una capa de decodificación para proyectar mapas de características a una dimensión espacial mayor. En la Figura 3.6 se muestra una interpretación de la convolución traspuesta en dos dimensiones [42].

3.3.10.3 Dropout

Un problema en el aprendizaje de redes profundas es la codependencia, donde todos los pesos se entrenan juntos y algunas conexiones tendrán una mayor capacidad de predecir correctamente que otras [43]. Si toda la red es entrenada de manera iterativa, las conexiones más fuertes se aprenden más y las más débiles son ignoradas. Después de muchas iteraciones, solamente un poco de neuronas se entrena, las demás dejan de participar en el entrenamiento.

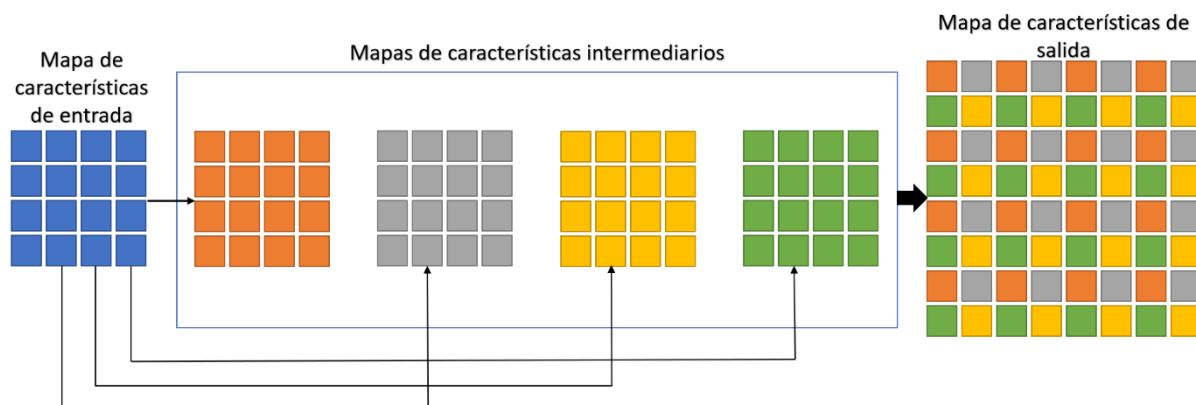


Figura 3.6. Ilustración de la operación convolución traspuesta. La entrada es un mapa de características de 4x4. Cuatro mapas de características intermedios son generados usando cuatro kernels diferentes. Estos se combinan produciendo una salida de 8x8. Los cuatro mapas de características intermedios se basan en la entrada pero no hay relación entre ellos.

Así pues, el dropout es una técnica para prevenir el sobre ajuste [44]. Esto mediante remover temporalmente algunas neuronas en la red, junto con sus entradas y salidas. La manera de elegir qué neurona remover es totalmente aleatoria. En el caso más sencillo, a cada neurona se le asigna una probabilidad P independiente de otras neuronas. En la Figura 3.7 se muestra el resultado de aplicar dropout a una red. Para entender la matemática detrás del dropout se recomienda revisar [45].

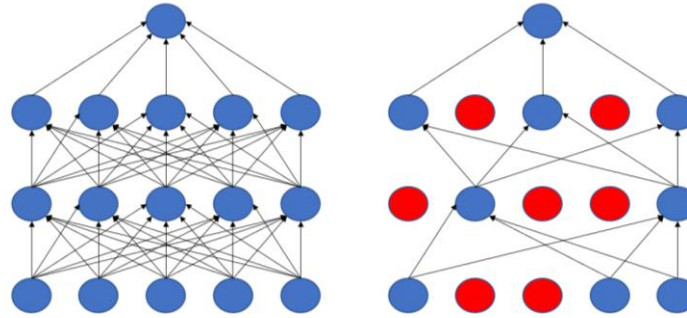


Figura 3.7. Red antes del dropout y red después del dropout.

3.3.10.4 Normalización por lotes

Es un algoritmo para mejorar la optimización en un modelo de aprendizaje al reducir el cambio de la covarianza del modelo [46]. Sin embargo, en [47] se menciona que esto no es así, sino que la normalización por lotes lo que hace es que ayuda a predecir los gradientes de las funciones de costo, con lo cual permite un rango mayor de factores de aprendizaje y una convergencia más rápida. Sea H_m un mini lote de activaciones de la capa a normalizar

$$H' = \frac{H_m - \mu}{\sigma} \quad (3.28)$$

Donde H' es el mini lote normalizado, μ y σ son vectores de la media y desviación estándar, de cada lote del conjunto de muestras de entrenamiento, respectivamente. Durante el entrenamiento

$$\mu = \frac{1}{m} \sum_i H_i \quad (3.29)$$

$$\sigma = \sqrt{\varepsilon + \frac{1}{m} \sum_i (H_m - \mu)_i^2} \quad (3.30)$$

Donde ε es un valor pequeño para evitar que (3.30) pueda ser cero. La normalización por lotes parametriza las redes profundas, reduciendo el problema de la coordinación de las actualizaciones a través de muchas capas. Este problema se presenta debido a que el gradiente informa cómo se debe actualizar cada parámetro sin tomar en cuenta el cambio existente en otras capas, pero en la práctica todas las capas se actualizan simultáneamente [48].

3.3.10.5 ReLU

Es una función de activación que actúa como un rectificador de media onda. La ReLU converge, típicamente, mucho más rápido en redes con muchas capas [49]. Con otras funciones de activación, las neuronas pueden tomar valores en lo cual es más probable que las clases sean linealmente separables o más fácilmente separables mediante métodos no lineales. La ReLU le permite a la red obtener fácilmente una representación dispersa, siendo esta su mayor ventaja y contribución [50]. Otras funciones dan valores pequeños pero no nulos, con la ReLU sí se tienen cero reales dando una representación dispersa verdadera

Matemáticamente está definida mediante. En la Figura 3.8 se muestra gráficamente.

$$f(x) = \max(0, x) \quad (3.31)$$

3.3.10.6 Leaky ReLU

Las ReLU tienen la desventaja de que durante la optimización el gradiente es 0 cuando la neurona no está activada. Esto podría llevar a casos donde la neurona nunca se activa y la optimización no ajusta los pesos de las neuronas que no iniciaron activas. La Leaky ReLU permite un valor pequeño para los valores menores de 0, sacrificando una representación dispersa por un valor de gradiente distinto de cero, que puede ser más robusto y útil durante la optimización [51].

Matemáticamente

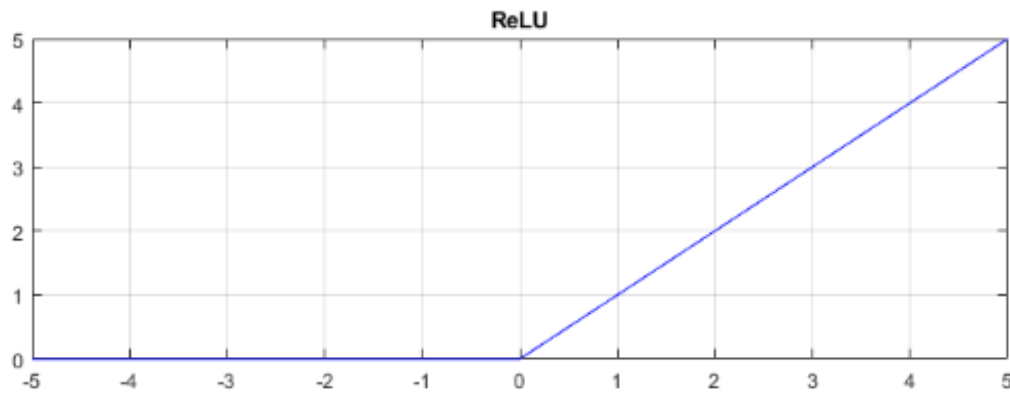


Figura 3.8. Función de activación ReLU.

$$f(x) = \begin{cases} x & x > 0 \\ ax & x \leq 0 \end{cases} \quad (3.32)$$

Donde a es la pendiente de la recta de la función para valores no positivos. En la Figura 3.9 se muestra gráficamente.

3.3.10.7 Tangente hiperbólica

Esta función de activación es una función centrada en el origen cuyo rango es $[-1, 1]$. Esta función está definida por

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.33)$$

En la Figura 3.10 se observa la gráfica de esta función.

3.3.11. Implementación

Se implementó una red tipo GAN cuyas capas y valores se muestran en la Tabla 3.1. Primeramente, se utilizó la base de datos que se encuentra en http://download.tensorflow.org/example_images/flower_photos.tgz. En esa base de datos se tienen 5 clases de flores: margaritas, diente de león, rosas, girasoles y tulipanes. Esas clases tienen 633, 898, 641, 699 y 799 muestras, respectivamente. Al comenzar el entrenamiento, las muestras generadas son ruido aleatorio, ya que es la entrada al generador y aún no ha habido tiempo para aprender alguna característica de las imágenes de muestra.

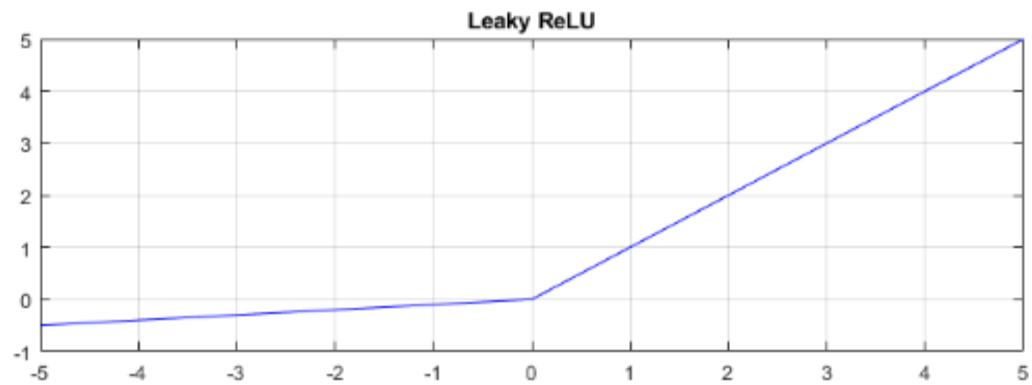


Figura 3.9. Función de activación Leaky ReLU.

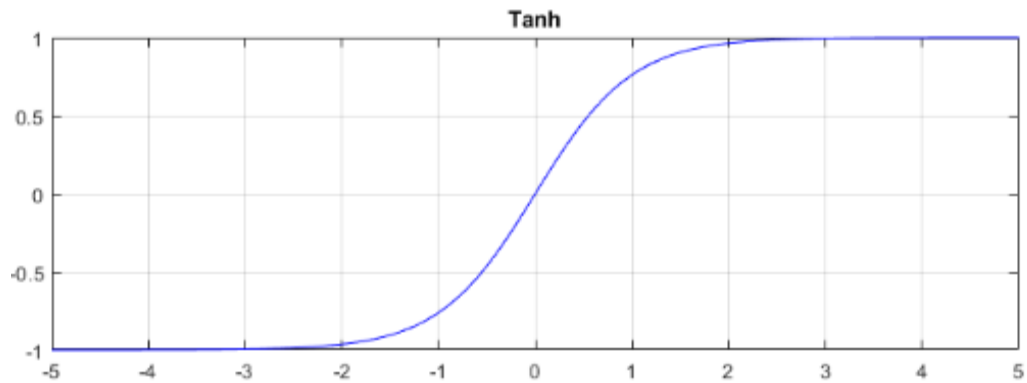


Figura 3.10. Función de activación tangente hiperbólica.

Conforme la red se sigue entrenando el discriminador obtiene características más representativas y esta información la utiliza el generador para hacer una distribución que se asemeje a la obtenida por el generador (Figura 3.11).

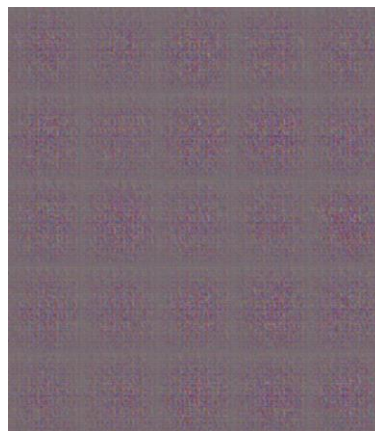


Figura 3.11. Imágenes generadas cuando comienza el entrenamiento.

Tabla 3.1. Arquitectura del generador y del discriminador de la GAN diseñada.

Generador			Discriminador		
Capa	Tipo de capa	Dimensión de la activación	Capa	Tipo de capa	Dimensión de la activación
1	Entrada de ruido gaussiano	1x1x100	1	Muestras del conjunto de entrenamiento	64x64x3
2	Proyección y cambio de tamaño	4x4x512	2	Dropout	64x64x3
3	Convolución traspuesta	8x8x384	3	Convolución	32x32x64
4	Normalización por lotes	8x8x384	4	Leaky ReLU	32x32x64
5	ReLU	8x8x384	5	Convolución	16x16x128
6	Convolución traspuesta	16x16x192	6	Normalización por lotes	16x16x128
7	Normalización por lotes	16x16x192	7	Leaky ReLU	16x16x128
8	ReLU	16x16x192	8	Convolución	8x8x256
9	Convolución traspuesta	32x32x96	9	Normalización por lotes	8x8x256
10	Normalización por lotes	32x32x96	10	Leaky ReLU	8x8x256
11	ReLU	32x32x96	11	Convolución	4x4x512
12	Convolución traspuesta	64x64x3	12	Normalización por lotes	4x4x512
13	Tanh	64x64x3	13	Leaky ReLU	4x4x512
			14	Convolución	1x1x1

En la iteración 104 el generador ya comienza a recibir información sobre cómo es la distribución de los datos reales; la salida ya no es ruido, sino imágenes que intentan representar la apariencia general de una flor, como se observa en la Figura 3.12.

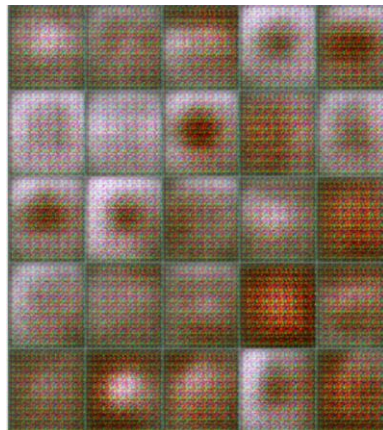


Figura 3.12. Imágenes generadas en la iteración 104.

En la Figura 3.13 se muestra el resultado final del entrenamiento



Figura 3.13. Imágenes generadas al final del entrenamiento en la iteración 19833.

Para generar solo un tipo de flores, ahora la red se entrenó con solo las imágenes de la clase tulipanes, que cuenta con 799 muestras. La red se entrenó durante 500 épocas, dando como resultado la Fig. 3.14. En definitiva, en este caso se observa que se requieren más muestras para una generación de muestras similar a flores, ya que con las 799 bajó mucho la calidad visual de las imágenes generadas en comparación con las 3670 (cuyo resultado se ve en la Figura 3.13). En un análisis inicial se tiene que la generación de muestras por tipo de flor requiere muchas más muestras que 799, en el uso de un conjunto con muchas clases no se sabe cuál imagen generada pertenecerá a algún tipo de flor, sin embargo, es el conocimiento de todas las imágenes de flores las que le enseñan al generador cómo se ve una flor.

De esta manera, se intuye que en la generación de parásitos será similar: si se pretende obtener un tipo específico de parásito es necesario entrenar con muchas imágenes de ese mismo parásito, y si solo se quieren obtener parásitos sin importar la clase, entonces es el conjunto de muchas muestras de varios tipos de parásitos los que le permitirán a la GAN generar una muestra nueva con las características generales que tiene un parásito.

Una vez generadas imágenes de flores, se prosigue a hacer lo propio con parásitos. Para esto se tienen 89 imágenes de parásitos obtenidos de las imágenes de muestra y a alguno de estos se les aplicó las transformaciones de la Figura 3.1 para tener un total de 470 imágenes de muestras con las cuales se entrenará la GAN.

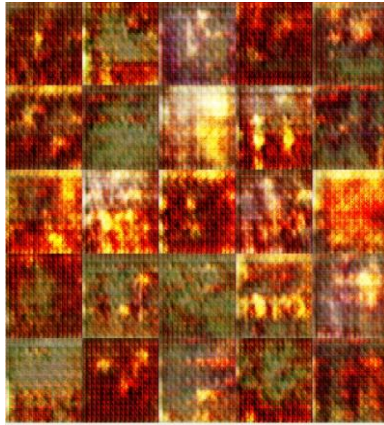


Figura 3.14. Imágenes generadas para la clase tulipanes.

Al inicio del entrenamiento las imágenes son ruido aleatorio, en la iteración 133 se realiza una actualización en las imágenes generadas, dando la Figura 3.15. En esta imagen se observa que lo primero que aprende la red sobre el conjunto de muestras son tres cosas: una forma regular similar a un círculo, el color verde y que el fondo es negro.

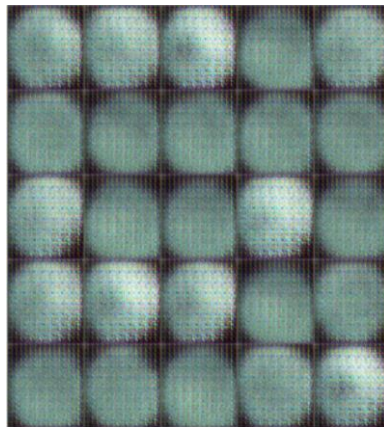


Figura 3.15. Primera actualización en el entrenamiento de la GAN para generación de parásitos.

En la iteración 1245 ya está más definida la forma del parásito, los tonos de color y se intenta asemejar la estructura interna del parásito, como se observa en la Figura 3.16. Finalmente, en la Figura 3.17 se muestran los parásitos generados al final del entrenamiento.

A continuación, se modificaron dos parámetros de la red: las épocas, pasando de 500 a 1000 y el mini lote, de 128 a 64. Los resultados de este cambio se observan en la Figura 3.18.

Se aprecia mucho más detalle en los parásitos, así como una forma más variada y colores más parecidos.

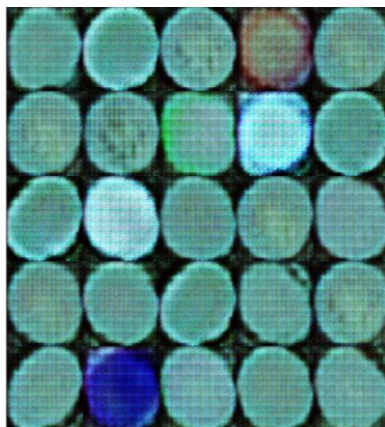


Figura 3.16. Iteración 1245 en la generación de parásitos.

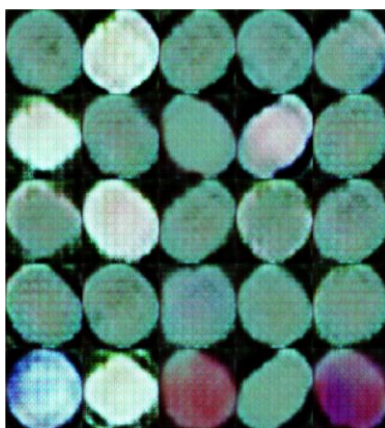


Figura 3.17. Parásitos generados.

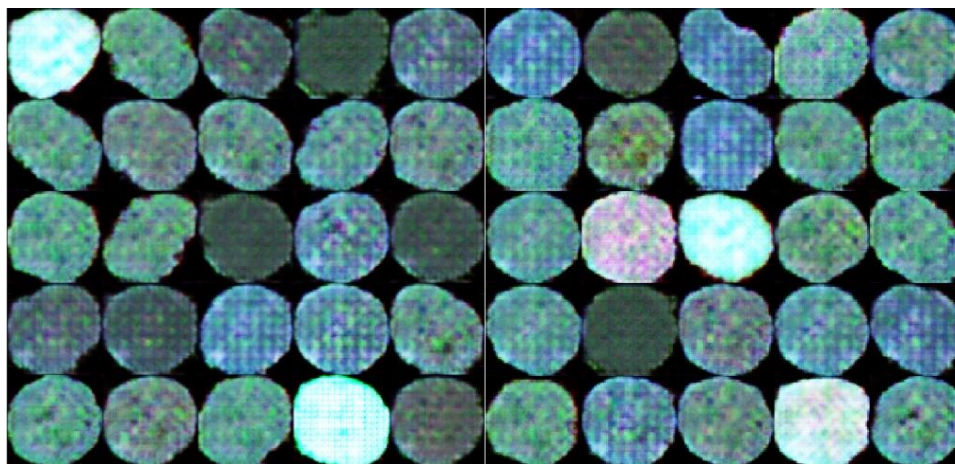


Figura 3.18. Parásitos generados con 1000 épocas y un mini lote de 64.

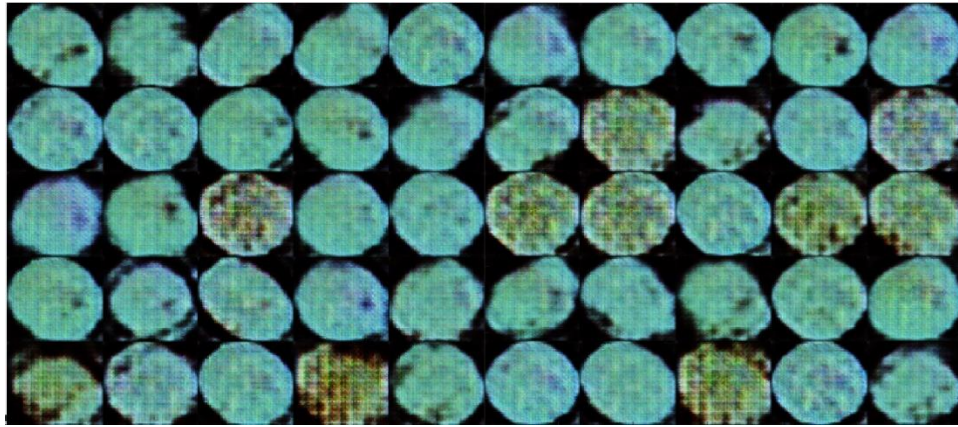


Figura 3.19. Generación de datos sin el aumento de datos clásico en el conjunto de datos de entrenamiento.

Con los mismos hiperparámetros con los que se obtuvo la Figura 3.19 se efectúa un entrenamiento utilizando solamente las imágenes originales, es decir, sin el aumento de datos con técnicas clásicas. Se obtiene, de esta manera, la Figura 3.20.

En la Figura 3.20 se observa que la red es capaz de aprender las características básicas de los parásitos, sin embargo los detalles de textura no aparecen en las imágenes generadas. Con lo cual, se muestra que al aumentar las muestras de entrenamiento mediante técnicas clásicas la red logra generar imágenes mucho más similares a las deseadas.

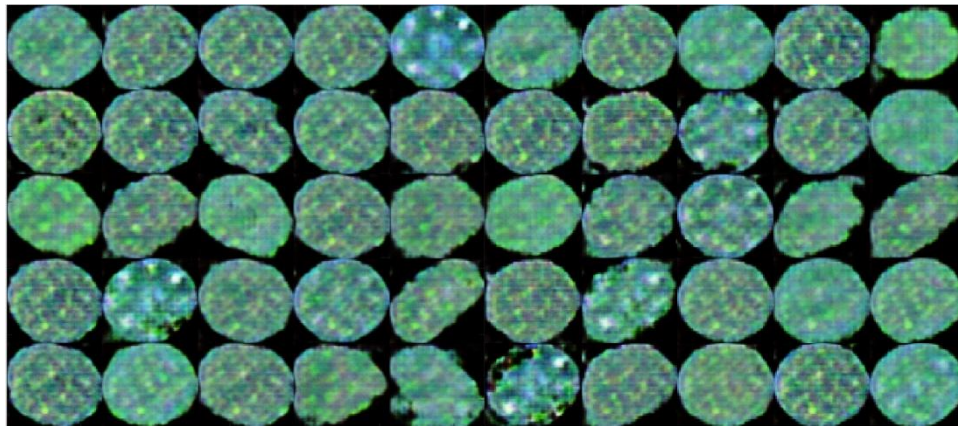


Figura 3.20. Parásitos generados con 700 muestras y 96 filtros en la capa de deconvolución del generador.

Se realiza otro experimento con más muestras y modificando la cantidad de filtros en la capa de deconvolución del generador. Las muestras son 700 y la cantidad de filtros son 96. En la Figura 3.21 se muestran los resultados. En la Figura 3.21 se tienen los mismos

hiperparámetros que los usados para obtener la Figura 3.20, pero se modificó el conjunto de entrenamiento; ahora se tiene el mismo conjunto utilizado en la Figura 3.19 pero replicado, es decir, todas esas imágenes se copian en el mismo conjunto de entrenamiento para tener un total de 534 imágenes. Se observa que no hay mucha diferencia entre replicar el conjunto de entrenamiento y aumentarlo mediante técnicas clásicas.

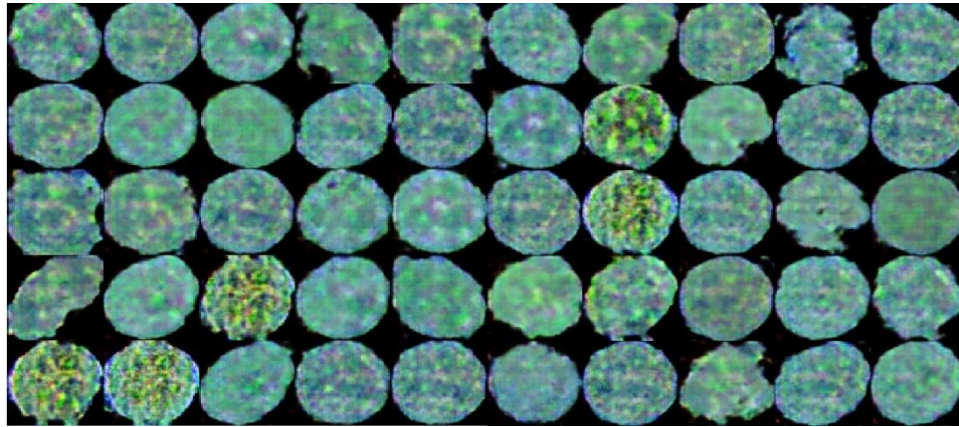


Figura 3.21. Generación de datos con el conjunto de datos repetido.

En la Figura 3.22 se muestran algunos ejemplos de los resultados de la generación de imágenes con la GAN para cada clase de parásito.

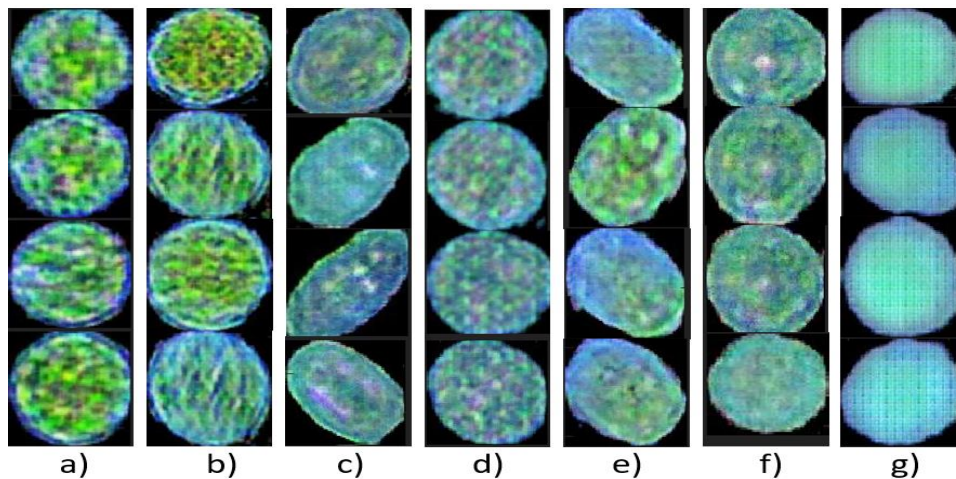


Figura 3.22. Imágenes generadas con la GAN, a) *Blastocystis hominis*, b) *Entamoeba coli*, c) *Chilomastix mesnili*, d) *Entamoeba histolytica*, e) *Giardia lamblia*, f) *Endolimax nana*, g) *Enteromonas hominis*.

Existen muchas áreas en las cuales la cantidad de muestras disponibles es muy escasa, en el área de la medicina esto es evidente, ya sea por privacidad, por la dificultad de obtener las muestras u otros factores. Se está en la necesidad de aumentar el conjunto de muestras. En este capítulo se presentaron técnicas clásicas de transformación y la generación de imágenes sintéticas con la GAN, además, de cómo el conjunto de entrenamiento influye en la calidad de las imágenes generadas.

Visualmente las imágenes obtenidas son buenas, se aprecia la forma y textura de los parásitos. A pesar de que se estudió la matemática de la GAN, aún quedan interrogantes que quedaron sin respuesta en esta investigación, la principal es cómo esta red hace que la probabilidad del generador G se asemeje a la del discriminador D . Y la otra es cuál es la mejor manera de entrenar una red GAN para obtener las mejores imágenes posibles.

CAPÍTULO 4 EXTRACCIÓN DE CARACTERÍSTICAS

En el presente capítulo se expondrán diversas características encontradas en la literatura. Las características aquí presentes de los objetos detectados (parásitos o no parásitos) son de textura y forma ya que es lo que más distingue visualmente a un parásito de un no parásito. Estas características se utilizarán para entrenar un clasificador binario no profundo.

4.1. Características de forma

Primeramente, se obtienen características de la forma, las cuales consisten en la medición de características en una imagen de tal manera que atienda a una descripción de la forma y estructura de un objeto [52]. Para obtener las características de la forma se utiliza la función de Matlab *regionprops* [53]. Se seleccionan las siguientes características de esta función:

- Área. Es el número de píxeles en la región, y se calcula con

$$Area = \frac{\sum_i (x_{c_i} + x_{c_{i-1}})(y_{c_i} + y_{c_{i-1}})}{2} \quad (4.1)$$

Donde x_c e y_c son las coordenadas de la región.

- Excentricidad. Es la relación de la distancia entre el foco de la elipse y la longitud de su eje mayor.

$$exc = \frac{\overline{cf}}{e_M} \quad (4.2)$$

Donde $\overline{cf}$ es la distancia del centro al foco de la elipse y e_M es la distancia del foco al vértice.

- Extensión. Es la relación de los píxeles en la región con los píxeles en el rectángulo que delimita la región. Se obtiene con

$$Ext = \frac{Area}{Rect} \quad (4.3)$$

- Solidez. Es la proporción de los pixeles en un área convexa que también está en la región.

$$Sol = \frac{Area}{AreaC} \quad (4.4)$$

- Longitud del eje mayor. Longitud en pixeles del eje mayor de la elipse que tiene el mismo segundo momento central normalizado que la región. Donde el momento central es un momento de la distribución de probabilidad de una variable aleatoria en la media de dicha variable.
- Longitud del eje menor. Longitud en pixeles del eje menor de la elipse que tiene el mismo segundo momento central normalizado que la región. Donde el momento central es un momento de la distribución de probabilidad de una variable aleatoria en la media de dicha variable.

En la Tabla 4.1 se tiene el promedio de características extraídas para 10 parásitos y 10 no parásitos.

Tabla 4.1. Promedio de algunas características de forma de parásitos y no parásitos.

Objeto	Área	Excentricidad	Extensión	Solidez	Eje Mayor	Eje Menor
Parásito	10553	0.5145	0.7243	0.9442	132.1149	108.2641
No parásito	5920.625	0.69203	0.76196	0.90251	116.0039	69.1397409893267

4.2. Características de textura

4.2.1 Matriz de coocurrencia de niveles de gris

La matriz de coocurrencia de niveles de gris es un enfoque para describir la textura en una imagen. Ya que el histograma sólo brinda información sobre la distribución de las

intensidades pero no de la posición relativa entre cada valor del pixel, es necesario utilizar la matriz de coocurrencia para obtener información de más valor sobre la posición relativa de los valores de los pixeles en la imagen [54].

Sea I_{gris} una imagen en escala de gris de $n \times n$ y sea N_g el número total de niveles de gris en la imagen, la matriz de coocurrencia es una matriz $GLCM$ de orden N_g , donde el (i,j) -ésimo elemento de $GLCM$ representa el número de veces que un pixel con intensidad i es adyacente a un pixel con intensidad j .

La $GLCM$ de I_{gris} es

$$GLCM(i, j) = \sum_{x=1}^n \sum_{y=1}^n \begin{cases} 1 & \text{si } I_{gris}(x, y) = i \text{ e } I_{gris}(x + \Delta x, y + \Delta y) = j \\ 0 & \text{en otro caso} \end{cases} \quad (4.5)$$

Donde Δx y Δy es la distancia del pixel de interés y su vecindario. Esta distancia $(\Delta x, \Delta y)$ hace que la matriz $GLCM$ sea variante a la rotación. Es decir, la misma imagen rotada tiene diferente $GLMC$. Esto se soluciona al usar un conjunto de *offset* que recorran 180° el mismo parámetro de distancia $([0 \ \Delta]$ para 0° , $[-\Delta, \Delta]$ para 45° , $[-\Delta, 0]$ para 90° y $[-\Delta, -\Delta]$ para 135°). Estas rotaciones son las que se muestran en la Figura 4.1 [55]. Una vez se hace esto, se promedian estas 4 rotaciones para obtener una $GLCM$ invariante a las rotaciones.

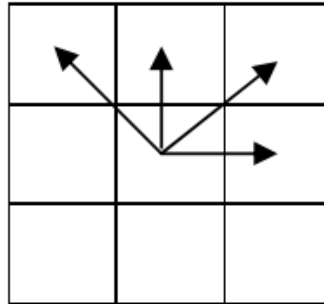


Figura 4.1. Direcciones de las rotaciones para que la $GLCM$ se invariante a la rotación.

4.2.2.1 Contraste

El contraste es una medición las variaciones de la intensidad entre un pixel y su vecino en toda la imagen. Para una imagen constante el contraste es 0, puesto que $i = j$ para todo i y j , haciendo la expresión (4.5) cero.

$$G_{MContraste} = \sum_{i=1}^{N_g-1} GLCM_{i,j} (i-j)^2 \quad (4.6)$$

4.2.2.2 Homogeneidad

Es una medición de la proximidad de la distribución de elementos en $GLCM$ a la diagonal de $GLCM$.

$$G_{M Homogeneidad} = \sum_{i=1}^{N_g-1} \sum_{j=1}^{N_g-1} \frac{GLCM_{ij}}{1+(i-j)^2} \quad (4.7)$$

4.2.2.3 Energía

Es la energía de cada elemento de la matriz de coocurrencia. El cuadrado de la energía es el segundo momento angular, que da valores grandes cuando en $GLCM$ hay pocos valores altos y es baja cuando sus elementos son similares.

$$G_{MEnergía} = \left(\sum_{i=1}^{N_g-1} \sum_{j=1}^{N_g-1} GLCM_{ij} \right)^2 \quad (4.8)$$

4.2.2.4 Correlación

Es una medición de qué tanta dependencia lineal hay en los tonos de gris de la imagen. Si la correlación es 0, no existe correlación lineal.

$$G_{M Correlación} = \sum_{i=1}^{N_g-1} \sum_{j=1}^{N_g-1} GLCM_{ij} \frac{(i-\mu_G)(j-\mu_G)}{\sigma_G^2} \quad (4.9)$$

Donde

$$\mu_G = \sum_{i=1}^{N_g-1} \sum_{j=1}^{N_g-1} i GLCM_{ij} \quad (4.10)$$

$$\sigma_G^2 = \sum_{i=1}^{N_g-1} \sum_{j=1}^{N_g-1} GLCM_{ij} (i-\mu_G)^2 \quad (4.11)$$

En la Tabla 4.2 se muestran la energía, homogeneidad, correlación y contraste de 5 parásitos y 5 no parásitos. La energía de los parásitos varió desde 0.1698 hasta 0.3404 y de los no parásitos desde 0.21 hasta 0.52, esta tendencia indica que la energía de los parásitos es más baja. La homogeneidad es muy similar para ambos, al igual que la correlación. El contraste tiene valores muy distintos entre la misma clase, hubo un no parásito que tuvo 0.07 y otro 0.27.

Tabla 4.2. Resultados de cuatro características de textura.

Parásitos				No parásitos			
Energía	Homogeneidad	Correlación	Contraste	Energía	Homogeneidad	Correlación	Contraste
0.1698	0.9314	0.9635	0.2797	0.2247	0.9693	0.9785	0.1861
0.1429	0.9218	0.9683	0.2358	0.2132	0.9596	0.9887	0.2736
0.3006	0.9659	0.9577	0.1072	0.3839	0.9823	0.9732	0.1459
0.3404	0.9726	0.9782	0.2073	0.5282	0.9874	0.9823	0.0776
0.2407	0.9464	0.9624	0.2273	0.3973	0.9756	0.9669	0.2170

4.2.2 Matriz de diferencia de niveles de gris

La matriz de diferencia de niveles de gris se basa en la incidencia de dos pixeles con una distinción absoluta de los niveles de gris y divididos por un desplazamiento $\delta = (\Delta x, \Delta y)$.

Sea I_δ la diferencia entre un pixel y otro dado por el desplazamiento δ y $\mathcal{I}(i|\delta)$ la densidad de probabilidad.

$$I_\delta(x, y) = |I(x, y) - I(x + \Delta x, y + \Delta y)|$$

$$\mathcal{I}(i|\delta) = \text{prob}(I_\delta(x, y) = i) \quad (4.12)$$

Se eligieron cuatro posibles vectores de desplazamiento [49]: $(0, d)$, $(-d, d)$, $(d, 0)$, $(-d, -d)$, donde d es la distancia del desplazamiento, que se ajustó en 1.

Si se la aplica la matriz de diferencia de niveles de gris a la Figura 4.2 se obtienen las distribuciones de probabilidad de la Figura 4.3.

Donde P1 es el resultado para el vector $(0,d)$, P2 lo es para $(-d,d)$, P3 es para $(d,0)$ y P4 es para $(-d,-d)$.

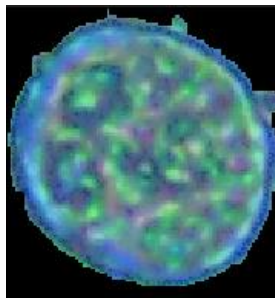


Figura 4.2. Parásito.

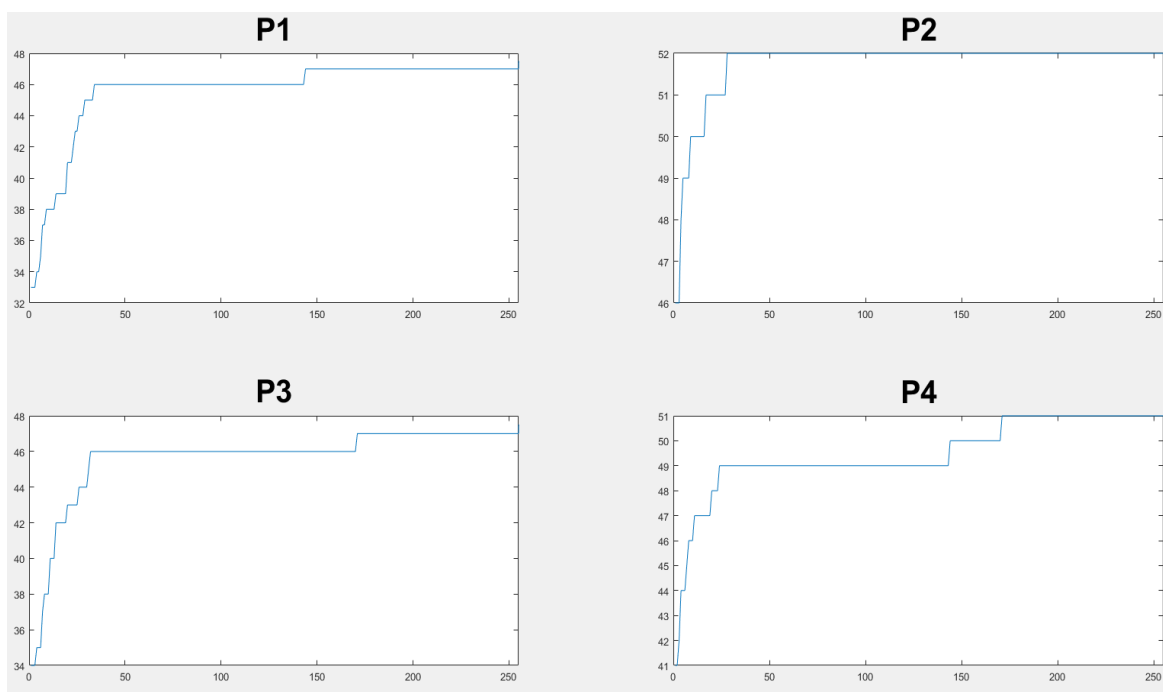


Figura 4.3. Distribuciones de probabilidad acumuladas, P1 es para el vector $(0,1)$, P2 es para el vector $(-1,0)$, P3 es para el vector $(1,0)$ y P4 es para el vector $(-1,1)$.

4.2.2.1 Contraste

Es el momento de inercia sobre el origen

$$GLD_{Contraste} = \sum i^2 g(i) \quad (4.13)$$

4.2.2.2 Entropía

Es la medición de comparación de la distribución.

$$GLD_{Entropía} = -\sum \mathcal{G}(i) \log \mathcal{G}(i) \quad (4.14)$$

4.2.2.3 Media

Es el valor esperado de la distribución

$$GLD_{Media} = \frac{1}{N_D} \sum i \mathcal{G}(i) \quad (4.15)$$

Donde N_D es el número de elementos en $\mathcal{G}$.

4.2.2.4 Segundo momento angular

Representa la uniformidad de la distribución de los niveles de gris de la imagen.

$$GLD_{SMA} = \sum \mathcal{G}(i)^2 \quad (4.16)$$

4.2.3 Patrones binarios locales

Los patrones binarios locales es un descriptor local de una imagen basado en el vecindario para cualquier pixel dado. El vecindario del pixel está dado por VE número de vecinos dentro de un radio R [56].

Definiendo una textura Θ en un vecindario local en una imagen en escala de grises como la distribución adjunta pbl de los niveles de gris de $VE + 1$ ($VE > 0$) pixeles

$$\Theta = pbl(g_c, g_0, \dots, g_{VE-1}) \quad (4.17)$$

Donde g_c corresponde al valor del pixel central del vecindario, $g_{VE}(VE=0, \dots, VE-1)$ corresponden a los pixeles equidistantes del radio R que forma la simetría circular del conjunto de pixeles vecinos.

Si el valor del pixel central se resta de los valores del vecindario, la textura local es

$$\Theta = pbl(g_c, g_0 - g_c, \dots, g_{VE-1} - g_c) \quad (4.18)$$

Si se asume que las diferencias son independientes de g_c

$$\Theta \approx pbl(g_c) pbl(g_0 - g_c, \dots, g_{VE-1} - g_c) \quad (4.19)$$

Como $pbl(g_c)$ describe la iluminación de la imagen, no hay relación entre esta cantidad y la textura de la imagen. Con lo cual puede omitirse de (4.19)

$$\Theta \approx pbl(g_0 - g_c, \dots, g_{VE-1} - g_c) \quad (4.20)$$

Para lograr invarianza con respecto a cualquier transformación en la escala de grises

$$\Theta \approx pbl(s(g_0 - g_c), \dots, s(g_{VE-1} - g_c)) \quad (4.21)$$

Donde

$$s(g) = \begin{cases} 1 & g \geq 0 \\ 0 & g < 0 \end{cases} \quad (4.22)$$

Ahora, un peso binomial 2^{VE} asigna a cada valor $s(g_P - g_c)$ transformando las diferencias en el vecindario en un código LBP que caracteriza la textura local de la imagen alrededor de $(\bar{i}, \bar{j})$. LBP es un operador: $LBP_{VE,R}$,

$$LBP_{VE,R}(\bar{i}, \bar{j}) = \sum_{i=0}^{VE-1} s(g_i - g_c) 2^i \quad (4.23)$$

En la Figura 4.4 se muestra el histograma mediante LBP de un parásito y un no parásito.

4.3. Descriptores de Fourier

Los descriptores de Fourier son un método para modelar las formas 2D descritas por un contorno cerrado.

Sea C_o una curva cerrada en el plano complejo. Si se recorre en dirección contraria a las manecillas del reloj de manera constante, se obtiene una función compleja $Z(t)$.

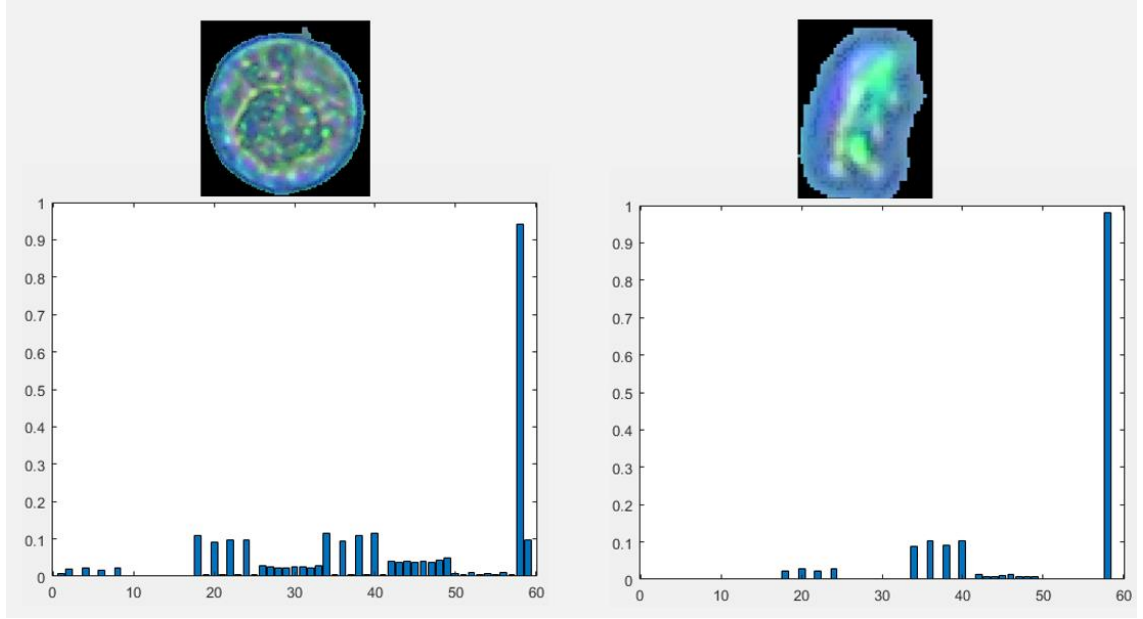


Figura 4.4. Patrones binarios locales para un parásito (izquierda) y un no parásito (derecha).

$$z(S) = \sum_n \Lambda_n e^{nSi} \quad (4.24)$$

Los coeficientes Λ_n son los descriptores de Fourier de la curva C_o

$$\Lambda_n = \frac{1}{L} \sum_{t=0}^{L-1} z(t) e^{-i(2\pi/L)nt} \quad (4.25)$$

Donde L es en número de muestras equitativamente espaciadas. En la Figura 4.5 se muestran los descriptores de Fourier para un parásito y en la Figura 4.6 para un no parásito.

Al observar las imágenes de parásitos y no parásitos, resulta evidente que hay una característica que los diferencian principalmente: la forma. Mientras que los parásitos tienen una forma suave, los no parásitos son bastante irregulares. Por eso razón se decidió utilizar una característica de forma: los descriptores de Fourier. Sin embargo, existen no parásitos que pueden tener una forma semejante. Por esa razón también se eligieron características de textura. Con la llegada de las redes profundas la extracción de características por métodos manuales ya quedó a segundo término, puesto que las redes profundas extraen características por sí mismas. La razón por la cual se realizó la extracción de características manuales fue para comprobar si las imágenes generadas con la red GAN podían tener una forma y textura

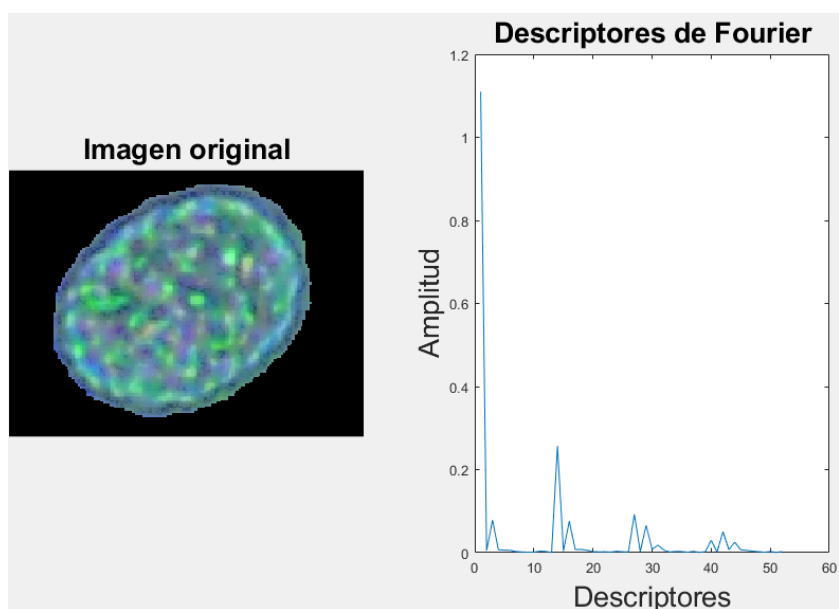


Figura 4.5. Descriptores de Fourier de un parásito.

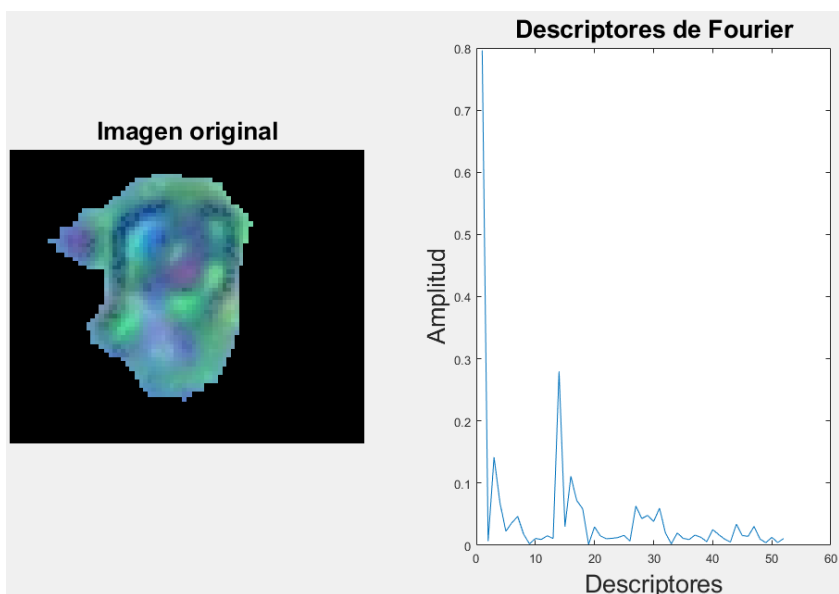


Figura 4.6. Descriptores de Fourier de un no parásito.

a las imágenes reales. Esta cuestión puede ser verificada visualmente, pero se prefirió entrenar un clasificador, pues esta es una investigación donde la intervención humana en la clasificación no tiene cabida.

CAPÍTULO 5 CLASIFICACIÓN

En este capítulo se mostrarán los modelos analizados e implementados para la tarea de clasificación. Primeramente, se tiene una clasificación entre dos clases: parásito y no parásito, utilizando el enfoque de máquina de vectores de soporte; luego una clasificación binaria y multiclase empleando modelos de aprendizaje profundo.

5.1. Clasificación binaria con Support Vector Machine (SVM)

Se entrenó un modelo de SVM con las siguientes características: descriptores de Fourier y la matriz de coocurrencia de niveles de gris. Se tienen 4 características de la matriz de coocurrencia: contraste, homogeneidad, energía y correlación; y 52 características de los descriptores de Fourier. El vector de características tiene esta forma [Contraste, Homogeneidad, Energía, Correlación, Descriptores de Fourier], de esas 56 características 20 fueron utilizadas para el entrenamiento del SVM: 1 2 3 4 5 6 8 9 10 13 14 15 18 22 23 24 36 37 54 56. Ese número indica la posición de la característica en el vector de características. En las Figuras 4.5 se observa que la cantidad de picos en los descriptores de Fourier no son 56, sino menos, entonces, por eso se eligieron 20 características. Esas 20 características son elegidas de la siguiente manera: se inicializa el algoritmo de selección con un conjunto vacío (que representa las características elegidas al inicio), luego se añade una característica a dicho conjunto de tal modo que esta maximice la predicción, esto se hace sucesivamente hasta tener el número de características deseado (20). Se entrenó con 100 imágenes: 50 parásitos y 50 no parásitos. Se evaluó el modelo con 79 imágenes de prueba: 44 no parásitos y 35 parásitos. El clasificador asignó correctamente 76 muestras a su correspondiente clase en la prueba. Este modelo es SVM 1.

En la Tabla 5.1 se muestran las métricas de evaluación de SVM 1. Se observa que la proporción de resultados ciertos es aceptable en la clasificación. A continuación, se entrenó el modelo SVM 2, con 162 muestras: 81 parásitos y 81 no parásitos. Tal elección se debe a que se tienen 111 muestras de no parásitos y se eligieron 81 para el entrenamiento y 30 para la prueba. Además, se quiso tener clases balanceadas. Las características en este clasificador son

las mismas que en el anterior. De las 60 muestras de prueba, 29 muestras de parásitos y 27 muestras de no parásitos fueron clasificadas correctamente. En la Tabla 5.1 se muestran los resultados como SVM 2.

El hecho de entrenar el clasificador con clases balanceadas es porque los clasificadores tienden a sesgar el modelo de aprendizaje. En el modelo de máquina de vectores de soporte se asume una distribución equitativa y los costos de una mala clasificación son iguales, el aprendizaje que predice la clase con menor muestras puede ignorarse. Esto causa que las muestras de prueba de la clase minoritaria sean clasificadas incorrectamente más a menudo que las de la clase mayoritaria.

Se prosiguió con el modelo SVM 3, que utiliza como imágenes de prueba a 30 parásitos generados con la GAN diseñada (una combinación entre los parásitos generados de la Figura 3.20 y 3.21) y los mismos 30 no parásitos de prueba que en el experimento anterior. De las 60 muestras 57 fueron clasificadas correctamente: 28 muestras de parásitos y 29 de no parásitos.

Como último experimento, se entrenó el clasificador SMV 4 quitando 30 imágenes de parásitos y sustituyéndolas por 30 imágenes de parásitos generadas. Así, el conjunto de entrenamiento está compuesto por 51 imágenes reales y 30 generadas. El conjunto de prueba fue el mismo que en el SVM 2. De las 60 imágenes del conjunto de prueba hubo 24 mal clasificadas, 18 parásitos y 6 no parásitos.

En resumen: SVM 1 representa el clasificador entrenado con 100 imágenes: 50 de parásitos y 50 de no parásitos, y el conjunto de prueba consiste en 79 imágenes: 44 no parásitos y 35 parásitos; SVM 2 es el clasificador entrenado con 162 imágenes: 82 de parásitos y 82 no parásitos, y el conjunto de prueba es de 60 imágenes: 30 parásitos y 30 no parásitos; SVM 3 es el clasificador idéntico al SVM 2 pero el conjunto de prueba difiere en que los 30 parásitos son los generados mediante la GAN; y finalmente, SVM 4 es el clasificador con 162 imágenes: 51 parásitos reales, 30 parásitos generados y los mismos 81 no parásitos, el conjunto de prueba es el mismo que en SVM 2.

De la Tabla 5.1 se observa que las imágenes generadas mediante la GAN diseñada se pueden utilizar como imágenes en el conjunto de prueba, pero utilizarlas en el conjunto de

entrenamiento hace que baje muchísimo el desempeño. Esto no quiere decir que las imágenes generadas con la GAN no sirvan. Aquí hay dos puntos: puede ser que las características con las cuales se entrenó el clasificador SVM 4 no sean suficientemente buenas, y la segunda es que las imágenes generadas no sean lo suficientemente parecidas a las imágenes originales. Visualmente las imágenes generadas son similares pero la resolución es menor, de unos 500x500 los parásitos reales (el tamaño varía) a 64x64 de las imágenes generadas. Este hecho impacta en la textura, no en la forma.

Tabla 5.1. Resultados con imágenes de prueba del clasificador binario con SVM.

Clasificador	Exactitud	Precisión	Sensibilidad	F1
SVM 1 Entrenamiento: 50 parásitos, 50 no parásitos. Prueba: 44 no parásitos, 35 parásitos	0.962	0.9211	1	0.9589
SVM 2 Entrenamiento: 82 parásitos, 82 no parásitos. Prueba: 30 no parásitos, 30 parásitos	0.9667	0.9375	1	0.9677
SVM 3 Entrenamiento: 81 parásitos, 81 no parásitos. Prueba: 30 no parásitos, 30 parásitos generados con la GAN	0.9831	0.9677	1	0.9836
SVM 4 Entrenamiento: 51 parásitos y 30 parásitos generados con la GAN, 81 no parásitos. Prueba: 30 no parásitos, 30 parásitos	0.7966	0.7813	0.8333	0.8065

Luego se realizó otro clasificador (SVM 5) en el que se tienen 224 muestras: 160 para el entrenamiento, 32 para la validación y 32 para la prueba (los tres conjuntos están perfectamente balanceados). Se utilizaron 124 características de la matriz de coocurrencia de niveles de gris, descriptores de Fourier y patrones binarios locales. Se seleccionaron 100 características de la misma manera que en los modelos SVM anteriores. En la Figura 5.1 se tiene la superficie del error medio durante el entrenamiento del modelo SVM 5, los puntos azules son los valores observados, el punto negro el siguiente punto, el asterisco es el mínimo

alcanzado y la media del modelo corresponde a la superficie de error. El modelo va en busca de los hiperparámetros adecuados para que la función de pérdida sea minimizada. En la Figura 5.2, se tiene una gráfica bidimensional del mínimo observado contra el valor estimado de la función de pérdida durante 30 iteraciones. Se aprecia que el error medido y el predicho convergen en un valor muy pequeño. Luego se tiene la Figura 5.3, donde se muestra la ROC del clasificador. Se observa un ajuste perfecto en la esquina superior izquierda, lo que indica que hubo una clasificación en el conjunto de entrenamiento exacta.

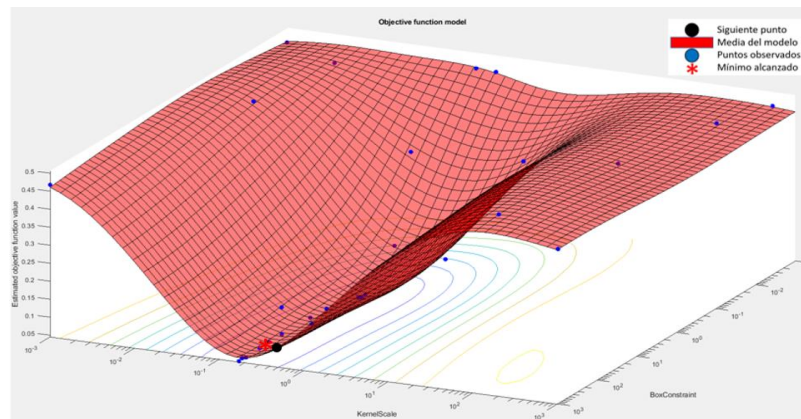


Figura 5.1. Función de error del modelo binario SVM 5 con 160 imágenes de entrenamiento, 32 de validación, 32 de prueba y 100 características.

Se calculó la predicción del modelo SVM 5 utilizando validación cruzada. Dando una predicción de 0.9. En la Tabla 5.2 se tienen las métricas de evaluación para el entrenamiento, validación, prueba 1 y prueba 2. La prueba 1 consiste en un conjunto de prueba con 16 parásitos y 16 no parásitos; por otro lado, prueba 2 es un conjunto de prueba con 16 no parásitos y 16 parásitos generados mediante la GAN.

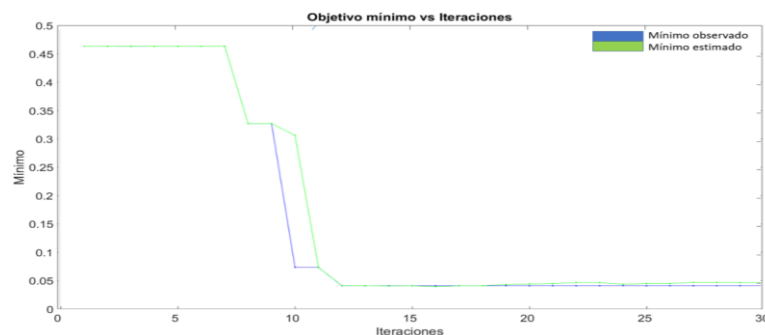


Figura 5.2. Valor de la función del modelo SVM 5.

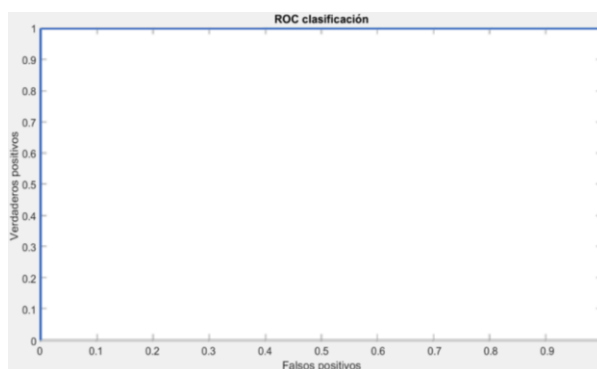


Figura 5.3. ROC de entrenamiento del modelo binario SVM 5 con 160 imágenes de entrenamiento, 32 de validación, 32 de prueba y 100 características.

Tabla 5.2. Resultados del modelo binario SVM 5 con 160 imágenes de entrenamiento, 32 de validación, 32 de prueba y 100 características.

	Exactitud	Precisión	Sensibilidad	F1
Entrenamiento	1	1	1	1
Validación	0.8947	0.9	1	0.9
Prueba 1 Prueba:16 parásitos y 16 no parásitos	0.8750	0.8333	0.9375	0.8824
Prueba 2 Prueba:16 parásitos generados con la GAN y 16 no parásitos	0.5	1	0.0323	0.0625

5.2. Clasificación con aprendizaje profundo

Los modelos de redes neuronales convolucionales profundas (DCNN) calculan una función polinómica factorizada mientras que las redes poco profundas calculan una función expandida, con lo cual, aquellas funciones que permiten una factorización pueden ser representadas y calculadas exponencialmente más eficiente en arquitecturas profundas.

Las DCNN calculan funciones como la convolución, rectificación y el max pooling, las cuales están ubicadas en capas. Estas capas están dispuestas en serie, donde la salida capa de convolución pasa a través de la rectificación y finalmente al max pooling.

En temas de imágenes, las redes profundas funcionan de la siguiente manera: los píxeles son vectores (en imágenes RGB, el vector es $I = (R, G, B)$), la aplicación de la convolución en la imagen crea una matriz de vectores RGB cuyos valores de los elementos se incrementan en presencia de ciertas características en las capas iniciales, como sombra, bordes, color. La salida suele pasar por la unidad de rectificación, que se activa si el resultado de la convolución supera un cierto umbral. Esta etapa es llamada detector, ya que la activación solo pasa si la activación del filtro indica que se halló una característica en esa posición. Ahí recae un gran poder de las DCNNs: el encontrar características en diferentes locaciones y orientaciones. Esto se consigue al pasar la salida de la unidad de rectificación a una capa para realizar un submuestreo de la actividad de muchos filtros con traslape espacial. Típicamente, esta capa es de max pooling.

Estas características abstractas se vuelven disponibles para el procesamiento de las siguientes capas, las cuales realizar una serie de operaciones similares para detectar características más complejas.

Otra característica de las DCNN es una conectividad escasa. Los nodos están conectados, usualmente, localmente con nodos con un traslape espacial o temporal. Esto reduce la cantidad de parámetros que necesitan ser aprendidos.

En este punto, surge la pregunta: ¿cómo aprenden tan bien las DCNN? Una respuesta es la composición jerárquica de características. Esta explicación más tradicional le brinda a la efectividad de las DCNN el paradigma de los mamíferos para aprender: la composición jerárquica de características más complejas a partir de características menos abstractas. Mediante esta explicación, es posible compartir el conocimiento, puesto que puede haber problemas donde las características menos abstractas sean comunes, y las características específicas de la clase sean las que están en las capas más ocultas.

5.2.1 Percepción visual en mamíferos

La percepción visual involucra la evaluación de estímulos ante ciertas características y la atención [57]. Los caminos y mecanismos en el cerebro que realizan esas funciones en mamíferos han evolucionado de manera independiente por 300 millones de años. Ante distintos retos visuales: caminar en cierto ecosistema o volar, esas redes se han moldeado para resolver de mejor manera la percepción del entorno. El procesamiento de la información que es común a todas las especies vertebradas recae en la información de la retina. Para una ubicación dada, distintos circuitos de la retina extraen diferentes tipos de información: contraste, frecuencia espacial y temporal, dirección del movimiento, luminiscencia, etcétera. Esta información se transmite al cerebro en varias clases de axones de células ganglionares (RGC). Los distintos tipos de clases de RGC exhiben ciertas morfologías, conexiones de entrada en la retina y de salida hacia el cerebro. La retina en todos los vertebrados contiene entre 10 y 20 clases funcionales de RGC, y el procesamiento paralelo de las propiedades visuales independientes por la retina le permiten al cerebro analizar la escena visual rápida y eficientemente.

En 1962 Wiesel y Hubel publicaron un descubrimiento en la corteza visual del gato. Utilizando celdas de grabaciones en las áreas V1 y V2 identificaron dos tipos de celdas: simples y complejas, dependiendo de su disparo ante patrones. Las celdas simples detectan características de bajo nivel (bordes, orientación, etcétera) y las celdas complejas tienen como entrada muchas celdas de bajo nivel y se disparan a las mismas características, pero con mayor grado de invarianza espacial. A partir de esto se pensó que muchas de esas capas simples y complejas podían encontrarse intercalados en el procesamiento visual y explicar la capacidad de reconocimiento de características abstractas.

5.2.2 Reconocimiento de características y aprendizaje en modelos profundos

Las redes neuronales artificiales, durante décadas, han sido consideradas como un ejemplo de caja negra. Se mencionó que era imposible entender cómo transforman la entrada en la salida; luego se encontró que su funcionalidad estaba distribuida a través de las unidades de la red, así que era imposible distinguir cuál parte de la red era responsable de una categoría

aprendida. Así pues, se ha vuelto un tema de investigación muy importante, pues como comenta Vincenzo Innocente, físico del CERN: “Como científico, no estoy satisfecho con solo distinguir entre perros y gatos. Como científico quiero ser capaz de decir: ‘la diferencia es esta y esta’.” Los modelos de aprendizaje supervisado reciben una salida deseada y los parámetros de la red se ajustan para conseguir el objetivo de clasificar una entrada a la salida deseada. Las CNN contienen neuronas que aprenden a distinguir categorías visuales de alto nivel de objetos como árboles, gatos, perros, caras, etcétera. Sin embargo, dichas categorías son aprendidas sin ayuda humana. Una salida deseada puede ser un cuarto; pero no se da información sobre nada más, no se indica ningún objeto como cama, televisión, peinador. La aparición de estas categorías en los modelos profundos se debe a que son pasos útiles para cumplir el objetivo establecido por el humano (el indicar la salida específica). Esto quiere decir que esas categorías están presentes en el conjunto de entrenamiento. Así pues, esas categorías son intrínsecas al conjunto de datos: no son arbitrarios. La adquisición de esas características de alto nivel se realiza sin ninguna estructura lingüística; esto es, las categorías visuales no requieren ser parte del lenguaje. Las unidades de capas separadas tienen la misma estructura, mientras que aprenden categorías de diferente nivel dependiendo en su ubicación dentro la arquitectura de la red; no hay nada en las unidades que las lleve a aprender alguna categoría en específica; con lo cual, la interacción entre las unidades es la que provoca que las categorías visuales aparezcan [58].

5.2.3 Transferencia de aprendizaje

El escenario ideal en el aprendizaje profundo es aquel donde hay muchísimas imágenes de entrenamiento etiquetadas, que tiene la misma distribución que las imágenes de prueba. Sin embargo, esto suele ser difícil de conseguir. Ahí entra la transferencia de aprendizaje. El aprender a transferir es el resultado de la generalización de la experiencia. Es decir, es posible realizar una transferencia de conocimiento de una situación a otra, siempre y cuando la persona generalice su experiencia. El requisito de la transferencia es que exista una conexión entre las dos actividades de aprendizaje. Inspirado en la habilidad de los seres humanos de transferir conocimientos a través de dominios, la transferencia de aprendizaje tiene como

objetivo aprovechar el conocimiento de un dominio relacionado para mejorar el aprendizaje o minimizar el número de ejemplos etiquetados requeridos en un dominio de prueba.

Para entenderla transferencia de aprendizaje se requieren varias definiciones.

- Dominio. Un dominio $\mathcal{D}$ está formado de dos partes: el espacio de características ζ y la distribución marginal $P(\mathcal{M})$.

$$D = \{\zeta, P(\mathcal{M})\} \quad (5.1)$$

El símbolo $\mathcal{M}$ representa el conjunto de ejemplos

$$\mathcal{M} = \{x \mid x_i \in \zeta, i = 1, \dots, n\} \quad (5.2)$$

- Tarea. Una tarea τ consiste en un espacio de etiqueta Υ y una función de decisión Γ que se espera sea aprendida desde los datos de muestra

$$\tau = \{\Upsilon \mid \Gamma\} \quad (5.3)$$

Algunos modelos de aprendizaje dan como salida la distribución condicional predicha de los ejemplos

$$\Gamma(x_j) = \{P(y_k \mid x_j) \mid y_j \in \zeta, k = 1, \dots, |\zeta|\} \quad (5.4)$$

En la práctica, el dominio se observa por un número de ejemplos con o sin la información etiquetada. El dominio fuente $\mathcal{D}_S$ correspondiente a la tarea fuente τ_S se observa a través del par ejemplo-etiqueta

$$D_S = \{(x, y) \mid x_i \in \zeta^S, y_i \in \Upsilon^S, i = 1, \dots, n^S\} \quad (5.5)$$

- Transferencia de aprendizaje. Dada una o más observaciones correspondientes a $r^S \in \mathbb{N}^+$ dominios y tareas $\{(\mathcal{D}_{S_i} \mid \tau_{S_i}) \mid i = 1, \dots, r^S\}$ y alguna o algunas observaciones $r^T \in \mathbb{N}^+$ sobre el dominio y tarea de prueba $\{(\mathcal{D}_{T_j} \mid \tau_{T_j}) \mid j = 1, \dots, r^T\}$, la transferencia de aprendizaje utiliza el conocimiento obtenido del dominio fuente al mejorar el desempeño de las funciones de decisión aprendidas en el dominio de prueba.

5.2.4 Red AlexNet

Es una red neuronal convolucional profunda utilizada en el concurso de ImageNet LSVRC en 2012 para clasificar imágenes entre 1000 clases. La red tiene 60 millones de parámetros y 650000 neuronas; está compuesta por 5 capas convolucionales, algunas de ellas seguidas por capas de max pooling y tres capas totalmente conectadas [59]. AlexNet se entrenó con 1.2 millones de imágenes, con 50000 imágenes de validación y 150000 imágenes de prueba. La arquitectura se muestra en la Figura 5.4.

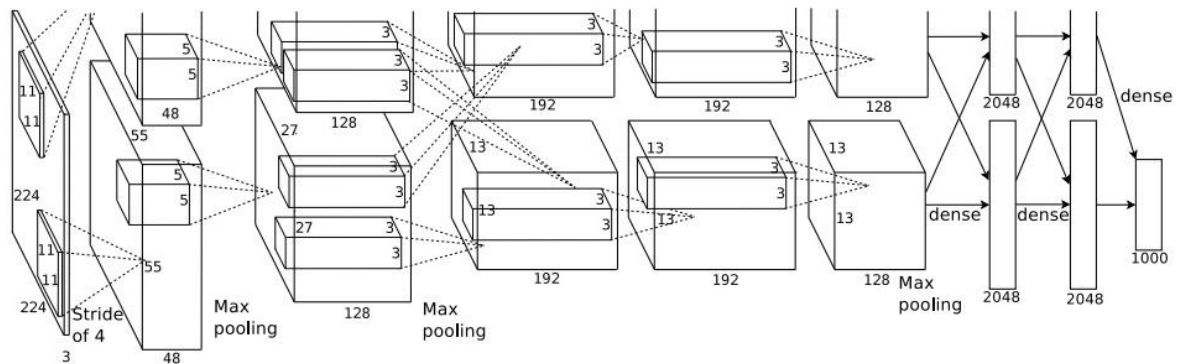


Figura 5.4. Arquitectura de AlexNet [59].

El conocimiento de la red son los valores de los filtros. Esto es, cada filtro, al actuar en la imagen de entrada, activará ciertas regiones de la imagen generando características extraídas por dichos filtros. Como se hará una transferencia de aprendizaje se observan algunas activaciones ante una imagen de entrada genérica de un gato (Figura 5.5) y en la Figura 5.6 se observa cómo se activan las capas de convolución de la red Alexnet.



Figura 5.5. Imagen para observar las activaciones de la red AlexNet.

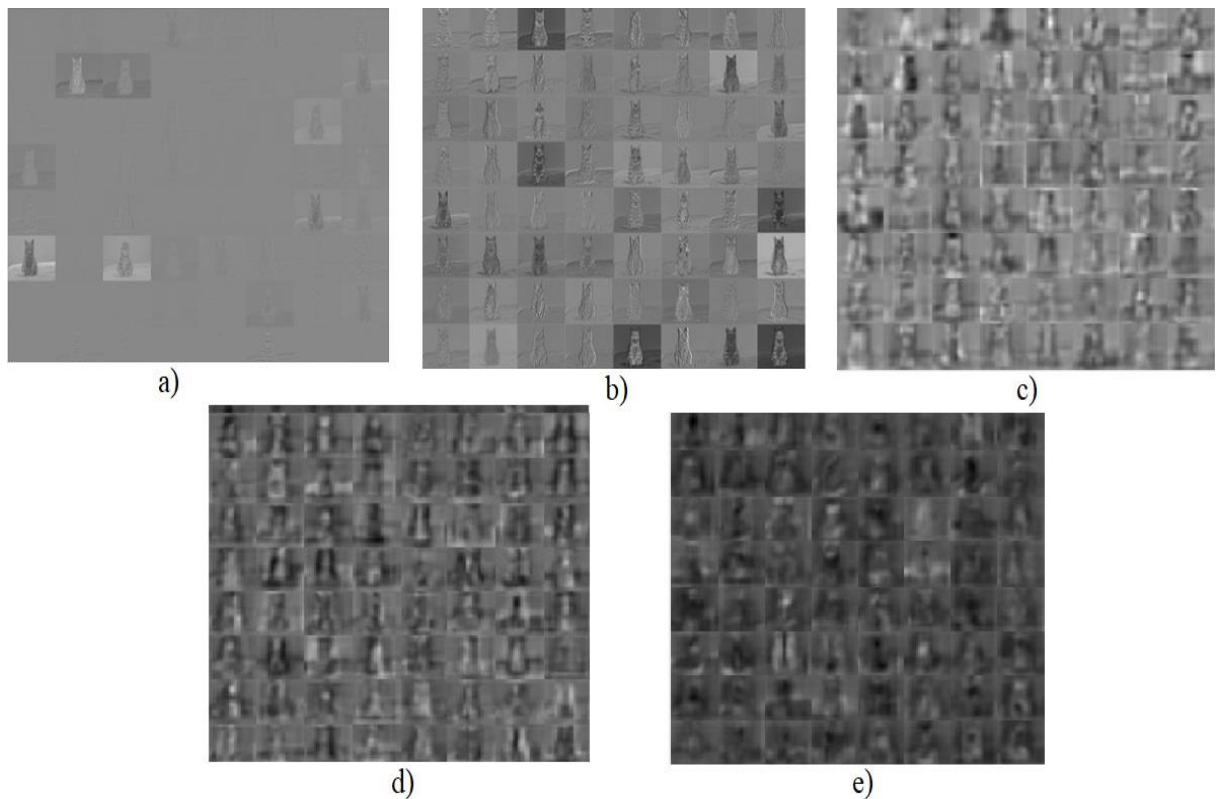


Figura 5.6. Activaciones de filtros, a) Conv1, b) Conv2, c) Conv3, d) Conv4, e) Conv5.

5.2.5 Clasificación binaria

En esta sección se presentará una clasificación binaria al transferir aprendizaje de una red profunda ya entrenada (AlexNet). Se modificó la capa 23 de la red AlexNet, correspondiente a una capa totalmente conectada, para que fuera de tamaño 2, pues ahora solo clasificará dos clases y no mil, para la cual fue diseñada originalmente. La red se entrenó con el gradiente estocástico descendente con momentum, con una tasa de aprendizaje inicial de 0.001, este valor disminuye 0.5 cada 5 épocas. El total de épocas es de 10 con un mini lote de 64. Primeramente, se tienen 710 muestras de parásitos y 710 muestras de no parásitos. Se realizó una división de las muestras de la siguiente manera: 70% para entrenamiento, 15% para validación y 15% para prueba. En la Figura 5.7 se muestra la gráfica de desempeño del entrenamiento y validación (Clasificador 1). La precisión en el entrenamiento fue del 100% y de la validación 99.53%. En la Figura 5.8 se muestra la matriz de confusión en la clasificación de las muestras de prueba y en la Figura 5.9 se observa la ROC.

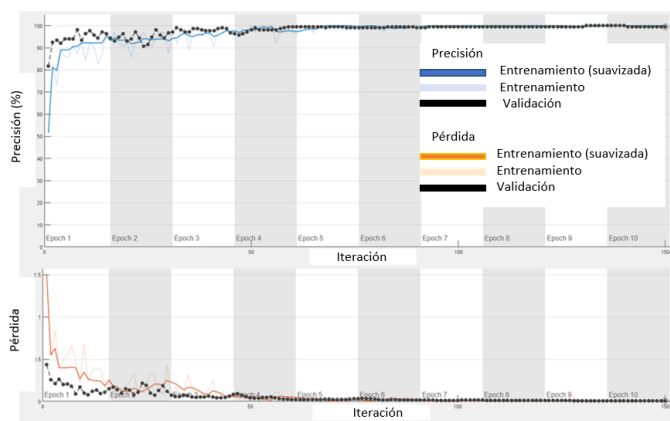


Figura 5.7. Desempeño para el Clasificador 1 con 710 muestras de parásitos reales y 710 muestras de no parásitos.

Salida	No parásitos	105 49.3%	1 0.5%	99.1 %	0.9%
	Parásitos	2 0.9%	105 49.3%	98.1 %	1.9%
		98.1 %	99.1 %	98.6 %	1.4%
		No parásito	Parásitos		
		Target			

Figura 5.8. Matriz de confusión para del conjunto de prueba (el 15% de 710 muestras de parásitos y 710 muestras de no parásitos) para el Clasificador 1.

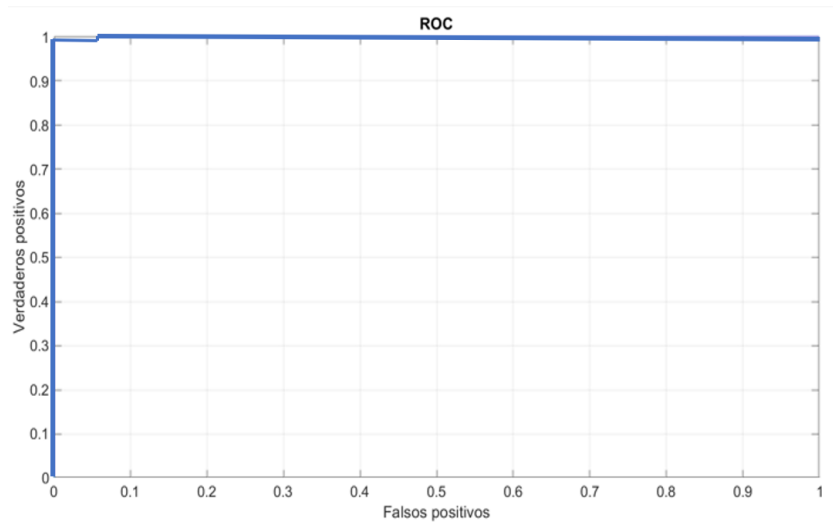


Figura 5.9. ROC para del conjunto de prueba (el 15% de 710 muestras de parásitos y 710 muestras de no parásitos) para el Clasificador 1.

A continuación, se evaluó el Clasificador 1 con un conjunto de prueba formado por 32 parásitos, 32 parásitos generados con la GAN y 32 no parásitos. En la Figura 5.10 se muestra la matriz de confusión de este experimento y en la Figura 5.11 la ROC.

Salida	No parásitos	32 33.3%	0 0.0%	100 %
	Parásitos	0 0.0%	64 66.7%	100 %
		100 %	100 %	100 %
		0.0%	0.0%	0.0%
		No parásito	Parásitos	
		Target		

Figura 5.10. Matriz de confusión para del conjunto de prueba formado por 32 parásitos, 32 no parásitos y 32 parásitos generados con la GAN evaluado con el Clasificador 1.

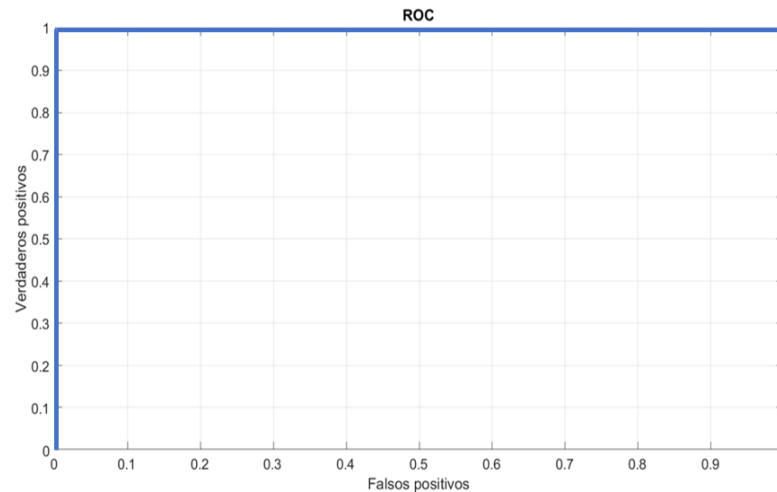


Figura 5.11. ROC para el conjunto de prueba de 32 parásitos, 32 no parásitos y 32 parásitos generados con la GAN evaluado con el Clasificador 1.

En la Tabla 5.3 se resumen los resultados de este entrenamiento. La validación cruzada k-fold con $k = 10$ se muestra en la Tabla 5.4.

Se efectuó otro experimento (Clasificador 2) modificando las muestras. Ahora se tienen 710 muestras de parásitos generados con la GAN y las mismas 710 muestras de no parásitos del experimento anterior. Todos los parámetros se mantuvieron igual. En la Figura 6.19 se muestra el desempeño para el entrenamiento y la validación.

CLASIFICACIÓN

Tabla 5.3. Clasificador 1. Red con 710 muestras de parásitos reales y 710 muestras de no parásitos, divididas en 70% para entrenamiento, 15% para validación y 15% para prueba.

	Exactitud	Precisión	Sensibilidad	F1
Entrenamiento	100%	100%	100%	100%
Validación	99.59%	100%	100%	99.9%
Prueba 1 (0.15 del total de todas las muestras)	98.59%	98.13%	99.06%	98.59%
Prueba 2 (32 parásitos reales, 32 parásitos generados con la GAN y 32 no parásitos)	100%	100%	100%	100%

Tabla 5.4. Validación cruzada k-fold con k = 10 para el Clasificador 1.

Precisión	0.9929	1	0.9929	1	1	0.9930	1	0.9859	1	1	0.9965
K	1	2	3	4	5	6	7	8	9	10	Promedio

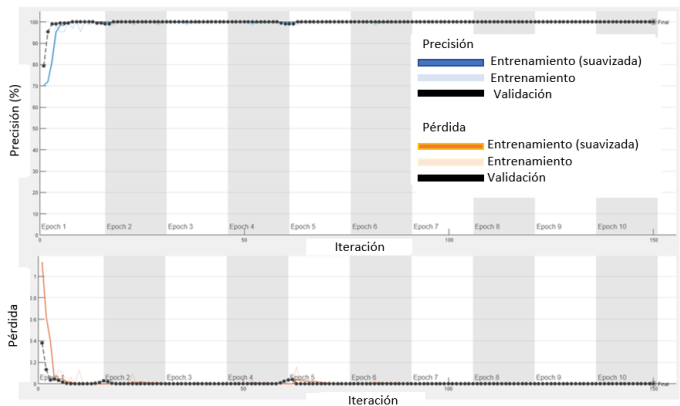


Figura 5.12. Desempeño durante el entrenamiento del Clasificador 2.

En la Figura 5.13 se muestra la matriz de confusión para el conjunto de prueba y en la Figura 5.14 se muestra la ROC. Se clasificó otro conjunto de prueba con el Clasificador 2 (formado por 32 parásitos reales, 32 parásitos generados con la GAN y 32 no parásitos). Los

resultados se muestran en la Figura 5.15 y la ROC en la Figura 5.16. En la Tabla 5.5 se muestran los resultados.

Salida	No parásitos	106 % 50%	0 0.0%	100 % 0.0%
	Parásitos	0 0.0%	106 0.0%	100 % 0.0%
		100 % 0.0%	100 % 0.0%	100 % 0.0%
		No parásito	Parásitos	
		Target		

Figura 5.13. Matriz de confusión del conjunto de prueba del Clasificador 2.

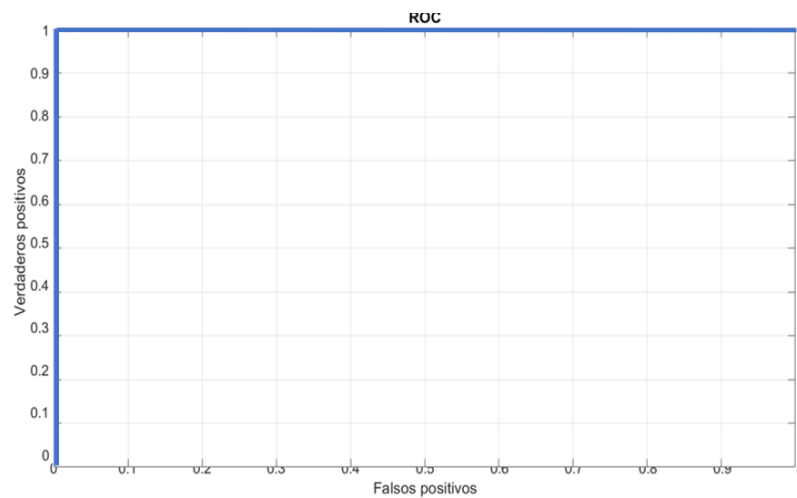


Figura 5.14. ROC del conjunto de prueba del Clasificador 2.

Salida	No parásitos	32% 33.3%	0 0.0%	100 % 0.0%
	Parásitos	23 24.0%	41 42.7%	64.1 % 35.9%
		58.2 % 41.8%	100 % 0.0%	76.0 % 24.0%
		No parásito	Parásitos	
		Target		

Figura 5.15. Matriz de confusión del conjunto formado por 32 parásitos reales, 32 parásitos generados con la GAN y 32 no parásitos clasificado con el Clasificador 2.

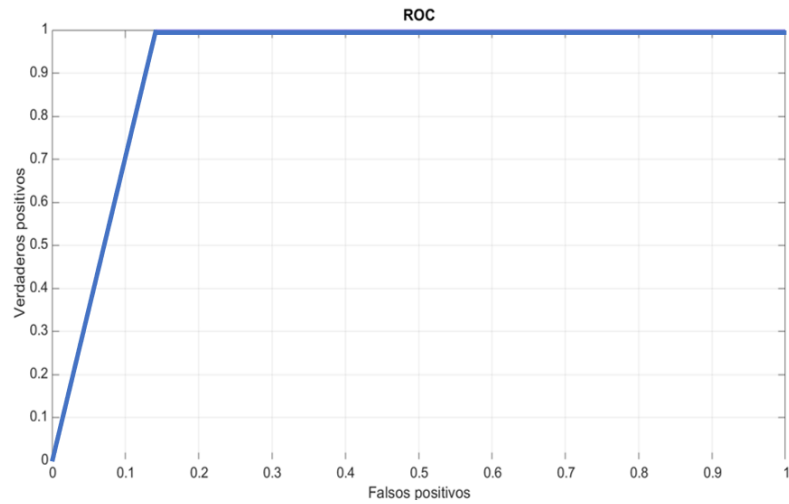


Figura 5.16. ROC del conjunto formado por 32 parásitos reales, 32 parásitos generados con la GAN y 32 no parásitos evaluado con el Clasificador 2.

Tabla 5.5. Clasificador 2. Red con 710 muestras de parásitos generados con la GAN y 710 muestras de no parásitos, divididas en 70% para entrenamiento, 15% para validación y 15% para prueba.

	Exactitud	Precisión	Sensibilidad	F1
Entrenamiento	100%	100%	100%	100%
Validación	100%	100%	100%	100%
Prueba 1 (0.15 del total de todas las muestras)	100%	100%	100%	100%
Prueba 2 (32 parásitos reales, 32 parásitos generados con la GAN y 32 no parásitos)	76.04%	64.06%	100%	0.7809%

Finalmente, se hizo otro experimento (Clasificador 3) manteniendo todos igual excepto las muestras. Se tienen 355 muestras de parásitos reales, 355 muestras de parásitos generados con la GAN y las mismas 710 muestras de no parásitos. En la Figura 5.17 se muestran las gráficas de desempeño. En la Figura 5.18 se muestra la matriz de confusión para el conjunto de entrenamiento y en la Figura 5.19 está la ROC. Se hizo de nuevo el experimento con las muestras de 32 parásitos, 32 no parásitos y 32 parásitos generados con la GAN.

En la Figura 5.20 está la matriz de confusión y en la Figura 5.21 la ROC. En la Tabla 6.7 se muestran los resultados del conjunto de prueba.

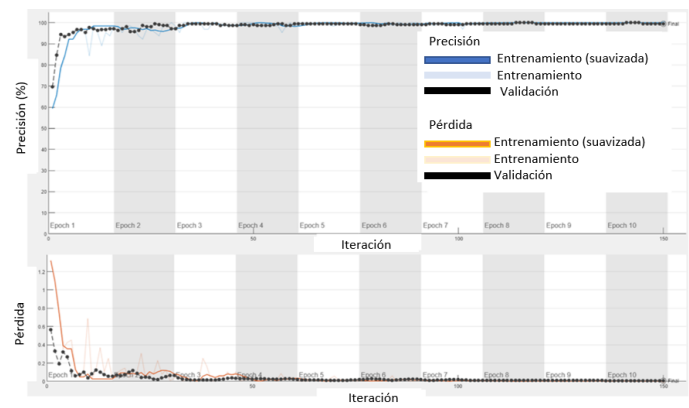


Figura 5.17. Desempeño del Clasificador 3, entrenado con 355 muestras de parásitos reales, 355 muestras de parásitos generados con la GAN y las mismas 710 muestras de no parásitos que el Clasificador 1 y 2.

Salida	No parásitos	105 49.5%	1 0.5%	99.1 0.9%
	Parásitos	4 1.9%	102 48.1%	96.2 3.8%
Target	No parásito	96.3 3.7%	99% 1.0%	97.6 2.4%
	Parásitos			

Figura 5.18. Matriz de confusión del Clasificador 3, entrenado con 355 muestras de parásitos reales, 355 muestras de parásitos generados con la GAN y las mismas 710 muestras de no parásitos que el Clasificador 1 y 2.

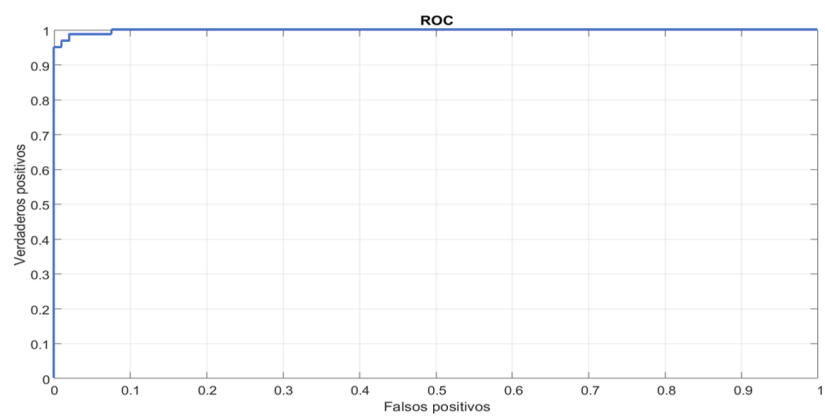


Figura 5.19. ROC del Clasificador 2, entrenado con 355 muestras de parásitos reales, 355 muestras de parásitos generados con la GAN y las mismas 710 muestras de no parásitos que el Clasificador 1 y 2.

CLASIFICACIÓN

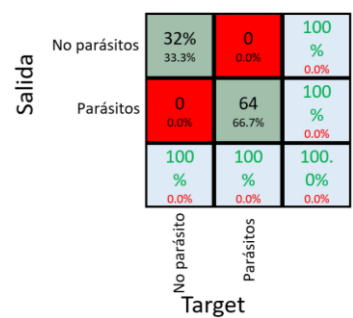


Figura 5.20. ROC del Clasificador 2 clasificando las muestras de 32 parásitos, 32 no parásitos y 32 parásitos generados con la GAN.

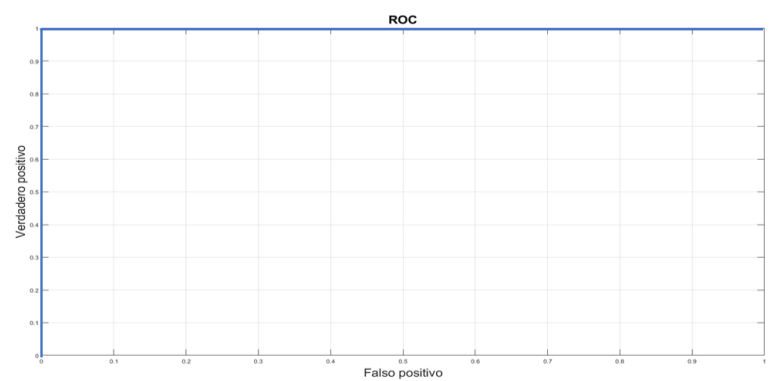


Figura 5.21. ROC del Clasificador 2 con las muestras de 32 parásitos, 32 no parásitos y 32 parásitos generados con la GAN.

Tabla 5.6. Clasificador 3, entrenado con 355 muestras de parásitos, 355 muestras de parásitos generados con la GAN y 355 muestras de no parásitos, divididas en 70% para entrenamiento, 15% para validación y 15% para prueba.

	Exactitud	Precisión	Sensibilidad	F1
Entrenamiento	99.88%			
Validación	99.53%			
Prueba 1 (0.15 del total de todas las muestras)	97.64%	93.57%	99.02%	96.21%
Prueba 2 (32 parásitos reales, 32 parásitos generados con la GAN y 32 no parásitos)	100%	100%	100%	100%

En la Tabla 5.7 se muestra la validación cruzada para cada iteración del Clasificador 3.

Tabla 5.7. Validación cruzada k-fold con $k = 10$ para el Clasificador 3, entrenado con 355 muestras de parásitos reales, 355 muestras de parásitos generados con la GAN y 710 muestras de no parásitos.

Precisión	1	0.9929	1	1	1	1	0.9929	0.9788	0.9859	0.9859	0.9937
K	1	2	3	4	5	6	7	8	9	10	Promedio

5.2.6 Clasificación multiclase

Ahora, se hace algo similar a la sección anterior, pero a la red AlexNet se le cambia la capa totalmente conectada para clasificar 8 clases: 7 de parásitos y una de no parásitos. En total se tienen 7010 imágenes divididas de la siguiente manera: 1008 de *Blastocystis hominis*, 720 de *Chilomastix meslini*, 360 de *Endolimax nana*, 828 de *Entamoeba coli*, 1512 de *Entamoeba histolytica*, 1170 de *Enteromonas hominis*, 648 de *Giardia lamblia* y 764 de no parásitos. Todas estas imágenes consisten en parásitos reales. A partir de una sola imagen se obtuvieron 36 mediante la transformación de la imagen original. Este experimento es el Clasificador 4.

En la Figura 5.22 se muestra el desempeño durante el entrenamiento. Las muestras fueron divididas, nuevamente, en 70% para el entrenamiento, 15% para la validación y 15% para la prueba. La red se entrenó durante 10 épocas, con 76 iteraciones por cada una. El entrenamiento duró 48 minutos y 53 segundos.

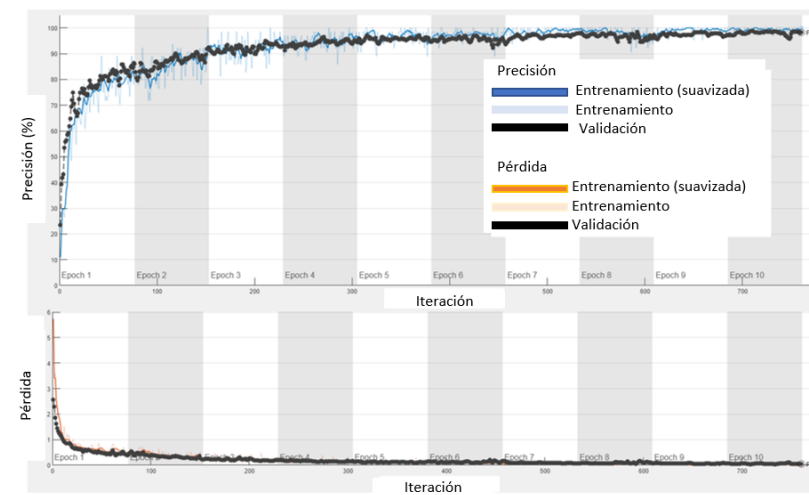


Figura 5.22. Desempeño durante el entrenamiento del clasificador de 8 clases con 7010 muestras reales.

CLASIFICACIÓN

En la Figura 5.23 se muestra la matriz de confusión para los datos de entrenamiento y en la Figura 5.24 se encuentra la ROC de dichos resultados.

Salida	Blastocystis Hominis	706 14.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 % 0.0%
	Chilomastix Mesnili	0 0.0%	504 10.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 % 0.0%
	Endolimax Nana	0 0.0%	0 0.0%	252 5.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 % 0.0%
	Entamoeba Coli	0 0.0%	0 0.0%	0 0.0%	578 11.8%	2 0.1%	0 0.0%	0 0.0%	99.7 % 0.3%
	Entamoeba Hystolitica	0 0.0%	0 0.0%	0 0.0%	0 0.0%	105 8 21.6%	0 0.0%	0 0.0%	100 % 0.0%
	Enteromonas Hominis	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	819 16.7%	0 0.0%	100 % 0.0%
	Giardia Lamblia	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	454 9.3%	100 % 0.0%
	No parásito	2 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	553 10.9% 99.6% 0.4%
		99.7 % 0.3%	100 % 0.0%	100 % 0.0%	100 % 0.0%	99.8 % 0.2%	100 % 0.0%	100 % 0.0%	100 % 0.1%
Target		Blastocystis Hominis	Chilomastix Mesnili	Endolimax Nana	Entamoeba Coli	Entamoeba Hystolitica	Enteromonas Hominis	Giardia Lamblia	No parásito

Figura 5.23. Matriz de confusión del conjunto de entrenamiento del Clasificador 4.

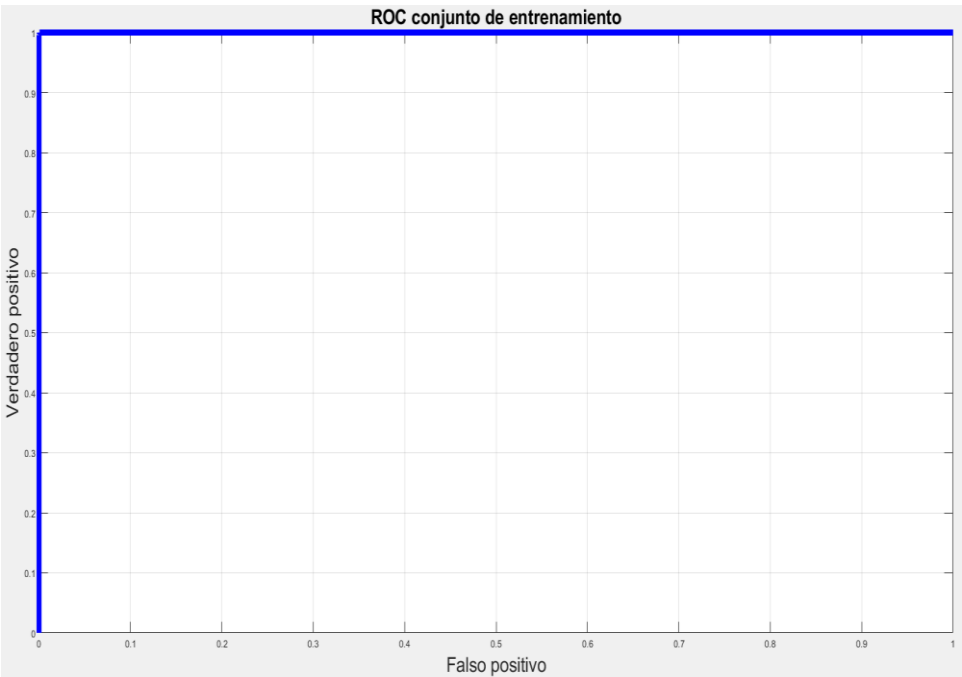


Figura 5.24. ROC del conjunto de entrenamiento del Clasificador 4.

En la Figura 5.25 y 5.26 está los resultados para el conjunto de prueba.

Salida	Blastocystis Hominis	150 14.3%	0 0.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.3 % 0.7%
	Chilomastix Mesnili	0 0.0%	106 10.1%	2 0.2%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	98.1 % 1.9%
	Endolimax Nana	0 0.0%	0 0.0%	53 5.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	98.1 % 1.9%
	Entamoeba Coli	0 0.0%	0 0.0%	0 0.0%	123 11.7%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	99.2 % 0.8%
	Entamoeba Hystolitica	0 0.0%	0 0.0%	0 0.0%	0 0.0%	227 21.6%	0 0.0%	0 0.0%	0 0.0%	100 % 0.0%
	Enteromonas Hominis	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	175 16.7%	0 0.0%	0 0.0%	100 % 0.0%
	Giardia Lamblia	0 0.0%	2 0.2%	0 0.0%	0 0.0%	1 0.1%	0 0.0%	94 8.9%	0 0.0%	96.9 % 3.1%
	No parásito	2 0.1%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	1 0.1%	1 0.1%	112 10.7%	97.4 % 2.6%
		100 % 0.0%	98.1 % 1.9%	96.4 % 3.6%	99.2 % 0.8%	98.3 % 1.7%	99.4 % 0.6%	98.9 % 1.1%	100 % 0.0%	99.0 % 1.0%
		Blastocystis Hominis	Chilomastix Mesnili	Endolimax Nana	Entamoeba Coli	Entamoeba Hystolitica	Enteromonas Hominis	Giardia Lamblia	No parásito	
		Target								

Figura 5.25. Matriz de confusión del conjunto de prueba del Clasificador 4.

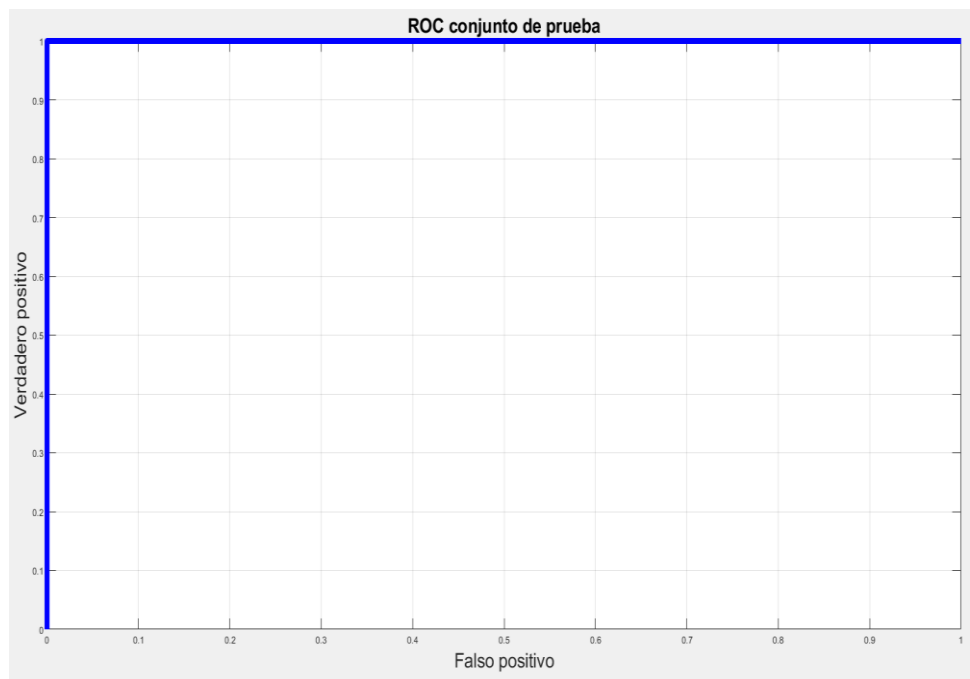


Figura 5.26. ROC del conjunto de prueba del Clasificador 4.

Por otra parte, en la Figura 5.27 y 5.28 para el conjunto de validación.

Salida	Blastocystis Hominis	145 13.8%	0 0.0%	0 0.0%	4 0.4%	1 0.1%	0 0.0%	0 0.0%	1 0.1%	96.0 % 4.0%
	Chilomastix Mesnili	0 0.0%	103 9.8%	5 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	95.4 % 4.6%
	Endolimax Nana	0 0.0%	0 0.0%	52 4.9%	0 0.0%	2 0.2%	0 0.0%	0 0.0%	0 0.0%	96.3 % 3.7%
	Entamoeba Coli	0 0.0%	0 0.0%	0 0.0%	123 11.7%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	99.2 % 0.8%
	Entamoeba Hystolitica	0 0.0%	0 0.0%	0 0.0%	0 0.0%	227 21.6%	0 0.0%	0 0.0%	0 0.0%	100 % 0.0%
	Enteromonas Hominis	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	176 16.7%	0 0.0%	0 0.0%	100 % 0.0%
	Giardia Lamblia	0 0.0%	0 0.0%	2 0.2%	0 0.0%	0 0.0%	0 0.0%	95 9.0%	0 0.0%	97.9 % 2.1%
	No parásito	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	1 0.1%	111 10.6%	97.4 % 2.6%
		100 %	99.0 %	88.1 %	96.9 %	98.3 %	99.4 %	99.0 %	99.1 %	98.2 %
		Blastocystis Hominis	Chilomastix Mesnili	Endolimax Nana	Entamoeba Coli	Entamoeba Hystolitica	Enteromonas Hominis	Giardia Lamblia	No parásito	
		Target								

Figura 5.27. Matriz de confusión del conjunto de validación del Clasificador 4.

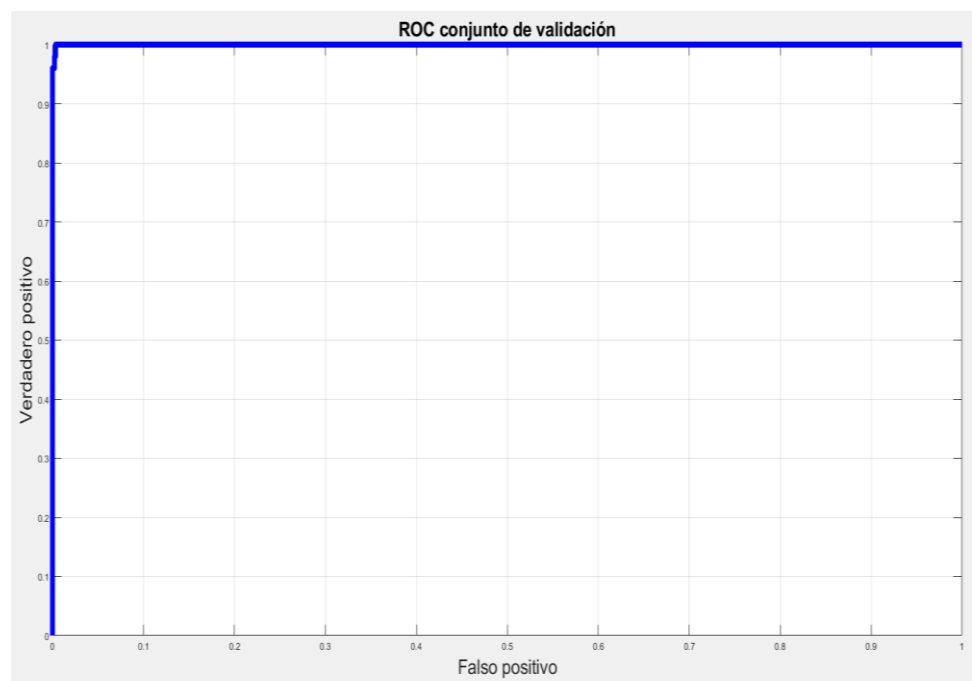


Figura 5.28. ROC del conjunto de validación del Clasificador 4.

CLASIFICACIÓN

En la Tabla 5.8 se observan los resultados del entrenamiento, validación y prueba. En la Tabla 5.9 se muestra la validación cruzada de la red con $k = 10$.

Tabla 5.8. Resultados del entrenamiento, validación y prueba del clasificador de 4.

	Entrenamiento	Validación	Prueba
Exactitud	98.38%	98.76%	98.4%

Tabla 5.9. Validación cruzada k-fold con $k = 10$ para el Clasificador 4.

K	1	2	3	4	5	6	7	8	9	10	Promedio
Exactitud	1	0.988	0.975	0.99	1	0.994	0.992	0.998	0.998	0.993	0.998

Después se realizaron experimentos de prueba en los que se utilizaron muestras generadas con la GAN: 253 de *Blastocystis hominis*, 225 de *Chilomastix mesnili*, 609 *Endolimax nana*, 231 de *Entamoeba histolytica*, 1014 de *Enteromonas hominis*, 197 *Giardia lamblia* y 9979 muestras de imágenes de no parásitos reales. En la Tabla 5.10 se muestran los resultados.

Tabla 5.10. Resultados de los clasificadores entrenados con cierto porcentaje de imágenes reales y un mayor porcentaje de imágenes generadas con la GAN.

	Entrenamiento	Validación	Prueba	Prueba con parásitos reales (5230 muestras)
Experimento 1 3155 muestras generadas con la GAN y 9979 imágenes de no parásitos.	99.9%	99.5%	99.3%	76.1%
Experimento 2 3155 muestras generadas con la GAN + 100 imágenes reales para	99.8%	98.6%	98.7%	91.3%

CLASIFICACIÓN

cada clase de parásito y 4384 imágenes de no parásitos.				
--	--	--	--	--

Salida	Blastocystis Hominis	11 0.1%	0 0.0%	3 0.0%	694 4.0%	0 0.0%	0 0.0%	22 0.1%	86 0.5%	1.3 %
	Chilomastix Mesnili	0 0.0%	235 1.4%	35 0.2%	9 0.1%	0 0.0%	2 0.0%	9 0.1%	190 1.1%	49.0 %
	Endolimax Nana	0 0.0%	416 2.4%	172 1.0%	12 0.1%	0 0.0%	0 0.0%	40 0.2%	320 1.8%	17.9 %
	Entamoeba Coli	0 0.0%	0 0.0%	7 0.0%	174 10.1%	9 0.0%	0 0.0%	5 0.0%	63 0.4%	95.9 %
	Entamoeba Hystolitica	21 0.1%	162 0.9%	102 0.6%	129 0.7%	512 3.0%	0 0.0%	0 0.0%	78 0.5%	51.9 %
	Enteromonas Hominis	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	200 1.2%	0 0.0%	970 5.6%	17.1 %
	Giardia Lamblia	0 0.0%	57 0.3%	68 0.4%	102 0.6%	23 0.1%	7 0.0%	323 1.9%	500 2.8%	29.9 %
	No parásito	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	1 0.1%	997 57.6%	100 %
		34.4 %	27.0 %	44.4 %	64.9 %	95.7 %	95.7 %	81.0 %	81.9 %	76.1 %
		65.6%	73.0%	55.6%	35.1%	4.3%	4.3%	19.0%	18.1%	23.9%
		Blastocystis Hominis	Chilomastix Mesnili	Endolimax Nana	Entamoeba Coli	Entamoeba Hystolitica	Enteromonas Hominis	Giardia Lamblia	No parásito	
		Target								

Figura 5.29. Matriz de confusión del conjunto de prueba del Experimento 1 de la Tabla 5.10.

Salida	Blastocystis Hominis	339 2.0%	0 0.0%	1 0.0%	359 2.1%	2 0.0%	0 0.0%	12 0.1%	3 0.0%	47.3 %
	Chilomastix Mesnili	0 0.0%	266 1.6%	137 0.8%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	65.7 %
	Endolimax Nana	0 0.0%	28 0.2%	841 5.0%	0 0.0%	16 0.1%	0 0.0%	0 0.0%	0 0.0%	95.9 %
	Entamoeba Coli	112 0.7%	2 0.0%	1 0.0%	162 9.7%	6 0.0%	0 0.0%	8 0.0%	0 0.0%	93.0 %
	Entamoeba Hystolitica	0 0.0%	1 0.0%	290 1.7%	29 0.2%	604 3.6%	0 0.0%	0 0.0%	5 0.0%	65.0 %
	Enteromonas Hominis	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	106 6.3%	0 0.0%	35 0.2%	96.8 %
	Giardia Lamblia	0 0.0%	45 0.3%	102 0.6%	39 0.2%	15 0.1%	9 0.1%	791 4.7%	4 0.0%	78.7 %
	No parásito	1 0.0%	14 0.1%	8 0.0%	6 0.0%	7 0.0%	102 0.6%	61 0.4%	978 58.3%	98.0 %
		75.0 %	74.7 %	60.9 %	79.0 %	93.5 %	90.5 %	90.7 %	99.5 %	91.3 %
		25.0%	25.3%	39.1%	21.0%	6.5%	9.5%	9.3%	0.5%	8.7%
		Blastocystis Hominis	Chilomastix Mesnili	Endolimax Nana	Entamoeba Coli	Entamoeba Hystolitica	Enteromonas Hominis	Giardia Lamblia	No parásito	
		Target								

Figura 5.30. Matriz de confusión del conjunto de prueba del Experimento 2 de la Tabla 5.10.

En este capítulo se presentaron diversos clasificadores, tanto profundos como no profundos, binarios como multiclase, y no solo fue de interés el desempeño de los modelos, sino la importancia de los datos. Como las imágenes médicas son escasas, resulta apropiado el experimentar con tales datos. Se hizo un aumento en el conjunto de muestras, se combinaron muestras de la GAN con reales, se dejaron solo generadas con la GAN, y se observó que los resultados fueron buenos. Además, se utilizó la transferencia de aprendizaje, ya que es un enfoque muy ad hoc con conjuntos de pocas muestras. Contrario a la inmensa cantidad de publicaciones utilizando redes profundas, aquí no se buscó modificar los parámetros de la red, sino el conjunto de datos.

CAPÍTULO 6 CONCLUSIONES Y TRABAJO FUTURO

En este capítulo se realizará una discusión destacando los resultados obtenidos, cómo mejorarlos, los problemas que se tuvieron durante la investigación, cómo poder resolverlos y, además, qué es, en términos generales, lo mejorable de este trabajo.

Para explicar el contenido antes mencionado, se seguirá el siguiente orden: se discutirá, primeramente, lo relacionado al procesamiento y segmentación de los parásitos, luego el aumento de datos, seguido de la extracción de características y, por último, lo relacionado a la clasificación. Este trabajo tiene la premisa de utilizar imágenes estandarizadas, con lo cual se espera que las imágenes sean muy similares entre sí: una cantidad de objetos parecida, el fondo de un color igual. Sin embargo, es menester el desarrollo de una técnica de procesamiento mucho más general, porque, al tratarse de imágenes médicas, es natural que aparezca alguna imagen con la suficiente contaminación (es decir, no parecida a las demás de su tipo) que requiera saberse si tiene o no parásitos. En este trabajo se observó que el método elegido funciona bien, es decir, los parásitos son detectados, pero tiene un límite muy claro y notorio: cuando las imágenes son visualmente más complejas, es decir, con muchos elementos que no corresponden al fondo, y cuando el parásito no está aislado, sino que forma parte de un objeto más grande, en este caso el parásito no es detectado o es detectado como parte de un objeto más grande. Debido a esto, algoritmo para detectar los parásitos que no sucumba ante imágenes complejas es necesario, además de útil: un algoritmo de detección y segmentación que tenga en cuenta la complejidad de las imágenes puede brindar trabajos muy buenos y resolver grandes dificultades, permitiendo expandir de muestras estandarizadas a cualquier tipo de imágenes adquiridas.

Por otra parte, el aumento de datos es un área prometedora. Lo que se pretendía saber al comienzo era si imágenes generadas artificialmente podían ser útiles para luego clasificar imágenes reales, pues esto quitaría el problema de la poca cantidad de muestras que se tienen en el área médica. Y no solo eso, sino que la cantidad de imágenes necesarias para generar dichas imágenes fuera pequeña. Como se observó en la Figura 5.29, el Experimento 1

clasificó dos tipos de parásitos reales a partir de imágenes generadas con la GAN: *Entamoeba histolytica* y *Enteromonas hominis*. Esto deja claro que un estudio a una gran profundidad para mejorar las imágenes generadas ayudaría enormemente a todos aquellos problemas de clasificación que se ven imposibilitados de tener éxito por la carencia de imágenes de muestra. A pesar de haber hecho varios experimentos durante la generación de imágenes, no quedó tan claro cómo asegurar que un clasificador entrenado con imágenes de parásitos generados con la GAN pueda clasificar imágenes reales.

Con el auge de las redes profundas, la extracción de características ya no es un tema tan de gran relevancia. Sin embargo, hay que tener en cuenta que, si con modelos de aprendizaje de máquina es suficiente, entonces no hace falta el aprendizaje profundo, no se debe usar en todo. Durante esta investigación se buscaron características obtenidas manualmente para diferenciar entre clases de parásitos, pero solo se llegó a una clasificación binaria utilizando un modelo no profundo.

La transferencia de aprendizaje resultó muy eficiente. La red ya pre entrenada con millones de imágenes fue capaz de utilizar conocimiento previo para aprender sobre los tipos de parásitos y clasificarlos correctamente. Sin embargo, esta área sigue siendo un terreno desconocido: ¿por qué aprenden las redes profundas?, ¿cómo saber que el conocimiento adquirido funcionará en otro desconocido?, ¿por qué surgen conceptos visuales durante el entrenamiento?, ¿por qué es necesario reconocer bordes, texturas, colores para que la red profunda clasifique correctamente? Y así infinidad de preguntas. La red ya aprendió y puede ser utilizada para aprender sobre otras imágenes sin utilizar muchas para el entrenamiento, pero el meollo del asunto sigue siendo misterioso. El mejor modelo fue el Clasificador 4, donde se utilizó la red Alexnet y se reentrenó con imágenes de 7 tipos de parásitos y de no parásitos.

Para finalizar, se hará una especie de recomendación para mejorar el trabajo hecho. Lo más importante de todo: un algoritmo que funcione con imágenes muy complejas es vital. Otro punto fundamental es mejorar las imágenes generadas artificialmente, ya que ayudaría mucho en cualquier problema de clasificación donde las muestras sean escasas. Utilizar redes tipo

GAN más modernas y eficaces u otros modelos de inteligencia artificial para entrenar una red profunda con imágenes sintéticas y que pueda clasificar imágenes reales sería muy útil. En esta investigación no se pudo clasificar imágenes reales a partir de las imágenes generadas para todas las clases disponibles. Y no necesariamente para diseñar una red profunda desde cero, sino, también, utilizar una red ya entrenada, que, como se vio en esta investigación, funcionó muy bien. Sin embargo, sería muy útil entrenar una red profunda con imágenes generadas y poder asegurar la correcta clasificación de imágenes reales. Las redes profundas funcionan, pero saber qué hacen, por qué hacen eso, cómo lo hacen, podría ser más importante que utilizar una. Cualquier aporte en este tema sería grandioso e, inevitablemente, ayudaría a mejorar su aplicación.

En retrospectiva, existieron cosas durante el trabajo que hoy no se habrían hecho. Algunos algoritmos que son útiles se encontraron tarde. La elección de experimentos pudo ser mejor, faltó experiencia en el uso de la red profunda. Pero como dijo Einstein: “Si supiese qué es lo que estoy haciendo, no lo llamaría investigación, ¿verdad?”

REFERENCIAS

- [1] R. Haque, «Human intestinal parasites,» *Journal of health, population and nutrition*, vol. 24(4), pp. 387-391, 2007.
- [2] D. Shen, G. Wu y H. Suk, «Deep Learning in Medical Image Analysis,» *Annual Review of Biomedical Engineering*, vol. 19, pp. 221-248, 2017.
- [3] A. Hafiz y G.Bhat, «A Survey of Deep Learning Techniques for Medical Diagnosis,» *Information and Communication Technology for Sustainable Development*, vol. 933, pp. 161-170, 2019.
- [4] T. Qiu, C. Wen, K. Xie, F. Wen, G. Sheng y X. Tang, «Efficient medical image enhancement based on CNN-FBB Model,» *IET Image Processing*, vol. 13, nº 10, pp. 1736-1744, 2019.
- [5] P. Rasti, M. Daneshmand, F. Alisinanoglu, C. Ozcinar y G. Anbarjafari, «Medical Image Illumination enhancement and sharpening by using stationary wavelet transform,» de *Signal Processing and Communication Application Conference (SIU)*, Zonguldak, 2016.
- [6] W. Rui y W.Guoyo, «Medical X-ray image enhancement method based on TV-homomorphic filter,» de *International Conference on Image, Vision and Computing (ICIVC)*, Chengdu, 2017.
- [7] Y. Hendrawan, C. Angkoso y R. Wahyuningrum, «Colour image segmentation for malaria parasites detection using cascading method,» de *International Conference on Sustainable Information Engineering and Technology*, Malang, 2017.
- [8] M. Wattana y T. Boonsri, «Improvement of Complete Malaria Cell Image Segmentation,» de *International Conference on Digital Information Management*,

Fukuoka, 2017.

- [9] L. Chen, M. Strauch, M. Daub, M. jansen, H. Luigis y D. Merhof, «Instant Segmentation of Nematode Cysts in Microscopic Images of Soil Sample,» de *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2019.
- [10] S. Abidin, U. Salamah y A. Nugroho, «Segmentation of malaria parasite candidates from thick blood smear microphotographs image using active contour without edge,» de *International Conference on Biomedical Engineering (IBIOMED)*, Yogyakarta, 2016.
- [11] P. Najgebauer, R. Grycuk, L. Rutkowski, R. Sherer y A. Siwocha, «Microscopic Sample Segmentation by Fully Convolutional Network for Parasite Detection,» *Artificial Intelligence and Soft Computin (ICAISAC)*, vol. 11508, pp. 164-171, 2019.
- [12] A. Nugruho, et al, «Performance of Convolutional Neural Network in Detecting Plasmodium Parasites,» de *International Conference on System Engineering and Technology (ICSET)*, Shah Alam, 2019.
- [13] R. Kumar, S. Singh y A. Khamparia, «Malaria Detection Using Custom Convolutional Neural Network Model on Blood Smear Slide Images,» *Advanced Informatics for Computing Research*, vol. 1075, pp. 20-28, 2019.
- [14] A. Chowdhury, J. Roberson, A. Hukko, S. Bodapati y D. Cappelleri, «Automated Complete Blood Cell Count and Malaria Pathogen Detection Using Convolutional Neural Network,» *IEEE Robotics and Automation Letters*, vol. 5, nº 2, pp. 1047-1054, 2020.
- [15] V. Sánchez, E. iturrizaga, J. León y C. Del Carpio, «An Algorithm for Giardia Lamblia Detection in Digital Images Acquired Through an Optical Microscope,» de *Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI)*, Bogotá, 2018.

- [16] M.Farid y T. Fatima., «Automatic detection of Plasmodium parasites from microscopic blood images,» *Journal of Parasite Disease*, vol. 44, pp. 69-78, 2019.
- [17] F. Azif, A. Nugroho y S. Wibirama, «Adaptive Threshold Determination Based on Entropy in Active Contour without Edge Method for Malaria Parasite Candidate Detection,» de *International Conference on Science and Technology (ICST)*, Yogyakarta, 2018.
- [18] A. Nugroho, S. Akbar y E. Murhandarwati, «Feature extraction and classification for detection malaria parasites in thin blood smear,» de *International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, Semarang, 2015.
- [19] A. Ravendran, T. de Silva y R. Senanayake, «Moment invariant features for automatic identification of critical malaria parasites,» de *International Conference on Industrial and Information Systems (ICIIS)*, Peradeniya, 2015.
- [20] I. Dave, «Image Analysis for Malaria Parasite Detection from Microscopic Images of Thick Blood Smear,» de *2017 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, Chennai, India, 2017.
- [21] T. Ahmed, M. Wahid y M. Hasan, «Combining Deep Convolutional Neural Network with Support Vector Machine to Classify Microscopic Bacteria Images,» de *International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Bangladesh, 2019.
- [22] H. Zaw, N. Maneerat y K. Win, «Brain Tumor detection based on Naïve Bayes Classification,» de *International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, Laos, 2019.
- [23] A. Nanoti, S. Jain, C. Gupta y G. Vyas, «Detection of malaria parasite species and life cycles stages using microscopic images of thin blood smear,» de *International*

- Conference on Inventive Computation Technologies*, India, 2016.
- [24] A. Olugboja y Z. Wang, «Malarie Parasite detection using different machine learning classifier,» de *International Conference on Machine Learning and Cybernetics (ICMLC)*, Ningbo, China, 2017.
- [25] H. Mohsen, E. El-Dahshan, E. El-Horbaty y A. Salem, «Classification using deep learning neural networks for brain tumors,» *Future Computing and Informatics Journal*, vol. 3, nº 1, pp. 68-71, 2018.
- [26] D. Davcev y G. Jakimovsk, «Lung cancer medical image recognition using Deep Neural Networks,» de *International Conference on Digital Information Management*, Alemania, 2018.
- [27] F. Spanhol, L. Oliveira, C. Petitjean y L. Heutte, «Breast cancer histopathological image classification using Convolutional Neural Networks,» de *International Joint Conference on Neural Networks (IJCNN)*, Vancouver, 2016.
- [28] M. Chen, X. Shi, Y. Zhang, D. Wu y M. Guizani, «Deep Features Learning for Medical Image Analysis with Convolutional Autoencoder Neural Network,» *IEEE Transactions on Big Data*, vol. 7, nº 4, pp. 750-758, 2017.
- [29] G. Liang, h. Hong, W. Xie y L. Zheng, «Combining Convolutional Neural Network With Recursive Neural Network for Blood Cell Image Classification,» *IEEE Access*, vol. 6, pp. 36188-36197, 2018.
- [30] J. Rodrigues, S. Khan, N. Islam, Z. Jan e I. Din, «A novel Deep Learning based framework for the detection and classification of breast cancer using transfer learning,» *Pattern Recognition Letters*, vol. 125, pp. 1-6, 2019.
- [31] A. Reddy y D. Juliet, , «Transfer Learning with ResNet-50 for Malaria Cell-Image Classification,» de *International Conference on Communication and Signal Processing*

(ICCSP), India, 2019.

- [32] T. Khoshgoftaar y C. Shorten, «A survey on Image Data Augmentation for Deep Learning,» *Journal of Big Data*, vol. 6, nº 60, 2019.
- [33] I. Goodfellow, J. Abadie, M. Mirza, B. Xu, D. Farley, S. Ozair, A. Courville y Y. Bengio, «Generative Adversarial Network,» *Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 2, pp. 2672-1680, 2014.
- [34] A. Antoniu, A. Storkey y H. Edwards, «Augmenting Image Classifiers Using Data Augmentation Generative Adversarial Networks,» de *International Conference on Artificial Neural Networks (ICANN)*, 2018.
- [35] J. Yang, Z. Zhao, H. Zhang e Y. Shi, «Data Augmentation for X-Ray Prohibited Item Images Using Generative Adversarial Networks,» *IEEE Access*, vol. 7, pp. 28894-28902, 2019.
- [36] V. Sandfort, K. Yan, P. Pickhardt y R. Summers, «Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks,» *Scientific reports*, vol. 9, 2019.
- [37] H. Shi, L. Wang G. Ding, F. Yang y X. Li, «Data Augmentation with Improved Generative Adversarial Networks,» de *International Conference on Pattern Recognition*, Beijing, 2018.
- [38] F. Konidaris, T. Tagaris, M. Sdraka y A. Stafylopatis, «Generative Adversarial Networks as an Advanced Data Augmentation Technique for MRI Data,» *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, vol. 5, pp. 48-59, 2019.
- [39] S. Shao, P. Wang y R. Yan, «Generative adversarial networks for data augmentation in machine fault diagnosis,» *Computers in Industry*, vol. 106, pp. 85-93, 2019.

- [40] R. Penrose, «La Segunda Ley y su misterio subyacente,» de *Ciclos del tiempo: una extraordinaria nueva visión del universo*, México, Debate, 2014, pp. 9-32.
- [41] K. O'shea y R. Nash, «An Introduction to Convolutional Neural Networks,» *ArXiv e-prints*, 2015.
- [42] Gao H, Yuan H, Wang Z. y Ji S., «Pixel Transposed Convolutional Networks,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, n° 5, pp. 1218-1227, 2020.
- [43] C. Ranjan, «Towards Data Science,» 2019. [En línea]. Available: <https://towardsdatascience.com/simplified-math-behind-dropout-in-deep-learning-6d50f3f47275>.
- [44] N. Srivastava, G. Hinton, A. Krizhevsky , I. Sutskever y R. Salakhutdinov, «Dropout: A Simple Way to Prevent Neural Networks from Overfitting,» *Journal of Machine Learning*, vol. 15, n° 56, pp. 1929-1958, 2014.
- [45] P. Baldi y P. Sadowski, «Understanding Dropout,» *Advances in Neural Information Processing Systems*, vol. 26, 2013.
- [46] S. Ioffe y C. Szegedy, «Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,» de *International Conference on Machine Learning*, 2015.
- [47] S. Santurkar, D. Tsipras, A. Ilyas y A. Madry, «How Does Batch Normalization Help Optimization?,» *NuerIPS*, 2018.
- [48] I. Goodfellow, Y. Bengio y A. Courville , «Batch Normalization,» de *Deep Learning*, MIT Press, 2016, pp. 317-321.
- [49] Y. LeCun, Y. Bengio y G. Hinton, «Deep Learning,» *Nature*, vol. 521, pp. 436-444, 2015.

- [50] X. Glorot, A. Bordes y Y. Bengio, «Deep Sparse Rectifier Neural Networks,» *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, vol. 15, pp. 15-323, 2011.
- [51] A. Maas, «Rectifier Nonlinearities Improve Neural Network Acoustic Models,» de *ICML Workshop on Deep Learning for Audio, Speech and Language Processing (WDLASL)*, 2013.
- [52] M. Yang, K. Kpalma y J. Ronsin, «A Survey of Shape Feature Extraction Techniques,» *Pattern Recognition*, pp. 43-90, 2008.
- [53] Mathworks.com, «Measure properties of image regions,» MATLAB, [En línea]. Available: <https://www.mathworks.com/help/images/ref/regionprops.html>.
- [54] A. Eleyan y H. Demirel, «Co-occurrence matrix and its statistical features as a new approach for face recognition,» *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 19, nº 1, pp. 97-107, 2011.
- [55] B. Sebastian, A. Unnikrishnan y K. Balakrishnan, «Gray Level Co-Occurrence Matrices: Generalisation and some new features,» *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, vol. 2, nº 2, 2012.
- [56] P. Bhagat, P. Choudhary y K. Manglem, «Sensors for Health Monitoring,» *Elsevier*, p. 278, 2019.
- [57] E. I. Knudsen, «Evolution of neural processing for visual perception in vertebrates,» *Journal of Comparative Neurology*, vol. 528, nº 17, pp. 2888-2901, 2020.
- [58] E. Rubio, «Throwing light on black boxes: emergence of visual categories from deep learning.,» *Synthese*, 2020.
- [59] A. Krizhevsky, I. Sutskever y G. Hinton, «ImageNet classification with deep convolutional neural networks,» *Advances in neural information processing systems*, vol.

REFERENCIAS

25, pp. 1097-1105, 2012.