



**Universidad Nacional Mayor de San Marcos**

**Universidad del Perú. Decana de América**

Dirección General de Estudios de Posgrado

Facultad de Ingeniería de Sistemas e Informática

Unidad de Posgrado

**Clasificación de lesiones en imágenes dermatoscópicas  
usando redes convolucionales profundas, transfer  
learning, congelamiento de capa óptima y ensamblado**

**TESIS**

Para optar el Grado Académico de Magíster en Ingeniería de  
Sistemas e informática con mención en Ingeniería de Software

**AUTOR**

Henry Miguel HERRERA DEL AGUILA

**ASESOR**

Dr. José Alfredo HERRERA QUISPE

Lima, Perú

2023



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

## Referencia bibliográfica

---

Herrera, H. (2023). *Clasificación de lesiones en imágenes dermatoscópicas usando redes convolucionales profundas, transfer learning, congelamiento de capa óptima y ensamblado*. [Tesis de maestría, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática/Unidad de Posgrado]. Repositorio institucional Cybertesis UNMSM.

---

## Metadatos complementarios

<b>Datos de autor</b>	
Nombres y apellidos	Henry Miguel Herrera Del Aguila
Tipo de documento de identidad	DNI
Número de documento de identidad	41602110
URL de ORCID	<a href="https://orcid.org/0000-0002-5553-3897">https://orcid.org/0000-0002-5553-3897</a>
<b>Datos de asesor</b>	
Nombres y apellidos	José Alfredo Herrera Quispe
Tipo de documento de identidad	DNI
Número de documento de identidad	40362859
URL de ORCID	<a href="https://orcid.org/0000-0002-8207-9714">https://orcid.org/0000-0002-8207-9714</a>
<b>Datos del jurado</b>	
<b>Presidente del jurado</b>	
Nombres y apellidos	David Santos Mauricio Sánchez
Tipo de documento	DNI
Número de documento de identidad	06445495
<b>Miembro del jurado 1</b>	
Nombres y apellidos	Nora Bertha La Serna Palomino
Tipo de documento	DNI
Número de documento de identidad	07665297
<b>Miembro del jurado 2</b>	
Nombres y apellidos	Ciro Rodriguez Rodriguez
Tipo de documento	DNI
Número de documento de identidad	06020241
<b>Datos de investigación</b>	

Línea de investigación	C.0.5.8. Inteligencia artificial
Grupo de investigación	No aplica.
Agencia de financiamiento	Sin financiamiento.
Ubicación geográfica de la investigación	Edificio: Universidad Nacional Mayor de San Marcos País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima Latitud: -12.055926 Longitud: -77.083469
Año o rango de años en que se realizó la investigación	2021 - 2023
URL de disciplinas OCDE	Ingeniería de sistemas y comunicaciones <a href="https://purl.org/pe-repo/ocde/ford#2.02.04">https://purl.org/pe-repo/ocde/ford#2.02.04</a>



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS  
Universidad del Perú. Decana de América  
Facultad de Ingeniería de Sistemas e Informática  
Vicedecanato de Investigación y Posgrado  
Unidad de Posgrado

**ACTA DE SUSTENTACIÓN DE TESIS PARA OPTAR EL GRADO ACADÉMICO  
DE MAGÍSTER EN INGENIERÍA DE SISTEMAS E INFORMÁTICA CON  
MENCIÓN EN INGENIERÍA DE SOFTWARE**

A los veintinueve (29) días del mes de noviembre de 2023, siendo las 4:00 pm., se reunieron en el Auditorio, Profesor: Alfredo Celso Alva Bravo, el Jurado de Tesis conformado por los siguientes docentes:

Dr. David Santos Mauricio Sánchez (Presidente)  
Dra. Nora Bertha La Serna Palomino (Miembro)  
Dr. Ciro Rodriguez Rodriguez (Miembro)  
Dr. José Alfredo Herrera Quispe (Miembro Asesor)

Se inició la Sustentación invitando al candidato a Magíster **HENRY MIGUEL HERRERA DEL AGUILA**, para que realice la exposición oral de la tesis para optar el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software, siendo la Tesis intitulada:

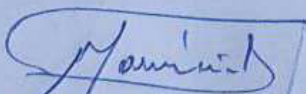
**“CLASIFICACIÓN DE LESIONES EN IMÁGENES DERMATOSCÓPICAS USANDO  
REDES CONVOLUCIONALES PROFUNDAS, TRANSFER LEARNING,  
CONGELAMIENTO DE CAPA ÓPTIMA Y ENSAMBLADO”**

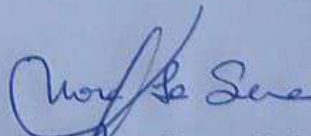
Concluida la exposición, los miembros del Jurado de Tesis procedieron a formular sus preguntas que fueron absueltas por el graduando; acto seguido se procedió a la evaluación correspondiente, habiendo obtenido la siguiente calificación:

..... VEINTINUEVE (19) ..... EXCELENTE .....


Por tanto, el presidente del Jurado, de acuerdo con el Reglamento General de Estudios de Posgrado, otorga al Bachiller **HENRY MIGUEL HERRERA DEL AGUILA** el Grado Académico de Magíster en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software.

Siendo las 17:30 horas, el presidente del Jurado de Tesis, da por concluido el acto académico de Sustentación de Tesis.

  
Dr. David Santos Mauricio Sánchez  
(Presidente)

  
Dra. Nora Bertha La Serna Palomino  
(Miembro)

  
Dr. Ciro Rodriguez Rodriguez  
(Miembro)

  
Dr. José Alfredo Herrera Quispe  
(Miembro Asesor)



## CERTIFICADO DE SIMILITUD

Yo JOSE ALFREDO HERRERA QUISPE en mi condición de asesor acreditado con Dictamen N° 000151-2022-UPG-VDIP-FISI/UNMSM de la tesis, cuyo título es Clasificación de lesiones en imágenes dermatoscópicas usando redes convolucionales profundas, transfer learning, congelamiento de capa óptima y ensamblado, presentado por el bachiller Henry Miguel Herrera del Aguila para optar el grado de Magíster en Ingeniería de Sistemas e Informática con mención en Ingeniería de Software, CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante el software de similitud textual, el documento evaluado cuenta con el porcentaje de 7 % de similitud, nivel **PERMITIDO** para continuar con los trámites correspondientes y para su **publicación en el repositorio institucional.**

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del grado/ título/ especialidad correspondiente.

Firma del Asesor

DNI: 40362859

Asesor: JOSE ALFREDO HERRERA QUISPE



## **DEDICATORIA**

A mi familia por siempre estar allí, a mi esposa por siempre apoyarme y mis compañeros y profesores de maestría por compartir sus conocimientos y camaradería. Papá, mamá esta va para ustedes.



## **AGRADECIMIENTOS**

A mi asesor el Dr. José Herrera por confiar en mi trabajo, muchas gracias por su consejo y comprensión. Al Profesor Gabriel Solari por enseñarme métodos numéricos hace más de 20 años, sus clases sirvieron de mucho. Finalmente, a la Facultad de Ingeniería de Sistemas e Informática de la UNMSM por el excelente trabajo en alentar la divulgación científica.

# Índice General

RESUMEN . . . . .	XI
ABSTRACT . . . . .	XII
<b>I Introducción</b>	<b>1</b>
1.1 Situación problemática . . . . .	1
1.2 Formulación del problema . . . . .	2
1.2.1 Problema general . . . . .	2
1.2.2 Problemas específicos . . . . .	2
1.3 Justificación teórica de la investigación . . . . .	3
1.4 Justificación práctica de la investigación . . . . .	3
1.5 Objetivos de la investigación . . . . .	3
1.5.1 Objetivo general . . . . .	3
1.5.2 Objetivos específicos . . . . .	4
1.6 Hipótesis de la investigación . . . . .	4
1.6.1 Hipótesis general . . . . .	4
1.6.2 Hipótesis específicas . . . . .	4
<b>II Marco Teórico</b>	<b>5</b>
2.1 Antecedentes de investigación . . . . .	5
2.2 Bases Teóricas . . . . .	9
2.2.1 Visión computacional . . . . .	9
2.2.2 Clasificación de imágenes . . . . .	10
2.2.3 Redes neuronales convolucionales (CNN). . . . .	13
2.2.4 Redes neuronales convolucionales profundas (DCNN). . . . .	14
2.2.5 <i>Transfer learning</i> . . . . .	19
2.2.6 Congelamiento de capa. . . . .	19
2.2.7 Búsqueda binaria. . . . .	20
2.2.8 Algoritmos genéticos . . . . .	20
2.2.9 Métodos de ensamblado. . . . .	20
2.2.10 Lesiones de la piel. . . . .	22
2.3 Marco conceptual. . . . .	25
2.3.1 <i>Data augmentation</i> . . . . .	25

2.3.2	Balanceo de clases. . . . .	26
2.3.3	Matriz de confusión. . . . .	27
2.4	Glosario. . . . .	30
<b>III Metodología</b>		<b>31</b>
3.1	Tipos de investigación. . . . .	31
3.1.1	Nivel de investigación. . . . .	31
3.1.2	Tipo de la investigación. . . . .	31
3.1.3	Diseño de la investigación. . . . .	31
3.1.4	Método de la investigación. . . . .	31
3.2	<i>Dataset</i> y partición. . . . .	32
3.2.1	<i>Dataset</i> . . . . .	32
3.2.2	Partición. . . . .	32
3.2.3	Técnica de muestreo de partición. . . . .	32
3.3	Operacionalización de variables. . . . .	32
3.3.1	Identificación de variables. . . . .	32
3.3.2	Matriz de operacionalización de variables. . . . .	33
3.3.3	Descripción de indicadores de variables. . . . .	37
3.4	Técnicas e instrumentos de recolección de datos. . . . .	38
3.4.1	Técnicas. . . . .	38
3.4.2	Instrumentos. . . . .	38
3.4.3	Validación y confiabilidad del instrumento. . . . .	38
3.4.4	Procedimientos . . . . .	38
<b>IV Fase Experimental</b>		<b>43</b>
4.1	Análisis del <i>dataset</i> HAM10000. . . . .	43
4.1.1	Ventajas del uso del <i>dataset</i> HAM10000. . . . .	43
4.1.2	Desventajas del uso del <i>dataset</i> HAM10000. . . . .	44
4.1.3	Aplicación de <i>data augmentation</i> y aplicación de pesos en el <i>dataset</i> HAM10000. . . . .	44
4.2	Modelo DCNN propuesto. . . . .	47
4.3	Arquitecturas DCNN. . . . .	47
4.4	Implementación de modelos. . . . .	47
4.4.1	Búsqueda binaria modificada. . . . .	47
4.4.2	Algoritmos genéticos. . . . .	49
4.4.3	Evaluación . . . . .	50
4.5	Hipótesis específica: <i>data augmentation</i> y aplicación de pesos. . . . .	50
4.5.1	InceptionV3 . . . . .	50
4.5.2	ResNet152V2 . . . . .	51
4.5.3	DenseNet121 . . . . .	52

4.5.4	EfficientNetB6 . . . . .	53
4.6	Hipótesis específica: <i>transfer learning</i> . . . . .	54
4.6.1	InceptionV3 . . . . .	55
4.6.2	ResNet152V2 . . . . .	55
4.6.3	DenseNet121 . . . . .	56
4.6.4	EfficientNetB6 . . . . .	57
4.7	Hipótesis específica: congelamiento de capa. . . . .	58
4.7.1	InceptionV3 . . . . .	59
4.7.2	ResNet152V2 . . . . .	60
4.7.3	DenseNet121 . . . . .	62
4.7.4	EfficientNetB6 . . . . .	64
4.8	Hipótesis específica: ensamblado. . . . .	66
4.8.1	<i>Hard Voting</i> . . . . .	67
4.8.2	<i>Average Soft Voting</i> . . . . .	68
4.8.3	<i>Weighted Soft Voting</i> usando algoritmos genéticos. . . . .	69
4.8.4	Comparativo de los métodos de ensamblaje. . . . .	70
<b>V</b>	<b>Resultados y discusión</b>	<b>72</b>
5.1	Análisis, interpretación y discusión de resultados. . . . .	72
5.1.1	Análisis de resultados experimentales en InceptionV3. . . . .	72
5.1.2	Análisis de resultados experimentales en ResNet152V2. . . . .	74
5.1.3	Análisis de resultados experimentales en DenseNet121. . . . .	76
5.1.4	Análisis de resultados experimentales en EfficientNetB6. . . . .	77
5.1.5	Análisis de resultados experimentales en ensamblado. . . . .	78
5.2	Pruebas de hipótesis. . . . .	79
5.2.1	Hipótesis específica: <i>data augmentation</i> y aplicación de pesos. . . . .	80
5.2.2	Hipótesis específica: <i>transfer learning</i> . . . . .	81
5.2.3	Hipótesis específica: congelamiento de capa. . . . .	82
5.2.4	Hipótesis específica: ensamblado. . . . .	83
5.3	Presentación de resultados. . . . .	84
<b>VI</b>	<b>Conclusiones y Recomendaciones</b>	<b>86</b>
6.1	Conclusiones . . . . .	86
6.1.1	Conclusión general . . . . .	86
6.1.2	Conclusiones específicas . . . . .	86
6.2	Recomendaciones . . . . .	87
	<b>Referencias bibliográficas</b>	<b>88</b>
	<b>Anexos</b>	<b>94</b>

## Lista de Tablas

<b>Tabla 1.</b>	Comparativo de beneficios y limitaciones de técnicas para clasificación de imágenes. . . . .	10
<b>Tabla 2.</b>	Comparativo de técnicas de clasificación de imágenes en función a los parámetros. . . . .	12
<b>Tabla 3.</b>	Matriz de operacionalización de variables. . . . .	34
<b>Tabla 4.</b>	HAM10000: imágenes por tipo de lesión . . . . .	46
<b>Tabla 5.</b>	Arquitecturas DCNN utilizadas . . . . .	47
<b>Tabla 6.</b>	Resultados experimentales para InceptionV3 . . . . .	51
<b>Tabla 7.</b>	Resultados experimentales para ResNet152V2 . . . . .	52
<b>Tabla 8.</b>	Resultados experimentales para DenseNet121 . . . . .	53
<b>Tabla 9.</b>	Resultados experimentales para EfficientNetB6 . . . . .	54
<b>Tabla 10.</b>	Resultados experimentales para InceptionV3 . . . . .	55
<b>Tabla 11.</b>	Resultados experimentales para ResNet152V2 . . . . .	56
<b>Tabla 12.</b>	Resultados experimentales para DenseNet121 . . . . .	57
<b>Tabla 13.</b>	Resultados experimentales para EfficientNetB6 . . . . .	58
<b>Tabla 14.</b>	Resultados experimentales detallados para InceptionV3 . . . . .	59
<b>Tabla 15.</b>	Resultados experimentales resumidos para InceptionV3 . . . . .	59
<b>Tabla 16.</b>	Resultados comparativos para InceptionV3 . . . . .	59
<b>Tabla 17.</b>	Resultados experimentales detallados para ResNet152V2 . . . . .	60
<b>Tabla 18.</b>	Resultados experimentales resumidos para ResNet152V2 . . . . .	61
<b>Tabla 19.</b>	Resultados comparativos para ResNet152V2 . . . . .	61
<b>Tabla 20.</b>	Resultados experimentales detallados para DenseNet121 . . . . .	62
<b>Tabla 21.</b>	Resultados experimentales resumidos para DenseNet121 . . . . .	63
<b>Tabla 22.</b>	Resultados comparativos para DenseNet121 . . . . .	63
<b>Tabla 23.</b>	Resultados experimentales detallados para EfficientNetB6 . . . . .	64
<b>Tabla 24.</b>	Resultados experimentales resumidos para EfficientNetB6 . . . . .	65
<b>Tabla 25.</b>	Resultados comparativos para EfficientNetB6 . . . . .	65
<b>Tabla 26.</b>	Resultados comparativos para el método de ensamblaje <i>Hard Voting</i> . . . . .	67
<b>Tabla 27.</b>	Resultados comparativos para el método de ensamblaje <i>Average Soft Voting</i> . . . . .	68

<b>Tabla 28.</b>	Resultados comparativos para el método de ensamblaje <i>Weighted Soft Voting</i> usando algoritmos genéticos. . . . .	69
<b>Tabla 29.</b>	Pesos obtenidos para el método de ensamblaje <i>Weighted Soft Voting</i> usando algoritmos genéticos . . . . .	70
<b>Tabla 30.</b>	Resultados comparativos de los métodos de ensamblaje. . . . .	71
<b>Tabla 31.</b>	Resultados experimentales para los indicadores de la hipótesis específica 1 en InceptionV3 . . . . .	73
<b>Tabla 32.</b>	Resultados experimentales para los indicadores de la hipótesis específica 2 en InceptionV3 . . . . .	73
<b>Tabla 33.</b>	Resultados experimentales para los indicadores de la hipótesis específica 3 en InceptionV3 . . . . .	74
<b>Tabla 34.</b>	Resultados experimentales para los indicadores de la hipótesis específica 1 en Resnet152V2 . . . . .	74
<b>Tabla 35.</b>	Resultados experimentales para los indicadores de la hipótesis específica 2 en ResNet152V2 . . . . .	75
<b>Tabla 36.</b>	Resultados experimentales para los indicadores de la hipótesis específica 3 (búsqueda binaria) en ResNet152V2 . . . . .	75
<b>Tabla 37.</b>	Resultados experimentales para los indicadores de la hipótesis específica 3 (algoritmos genéticos) en ResNet152V2 . . . . .	75
<b>Tabla 38.</b>	Resultados experimentales para los indicadores de la hipótesis específica 1 en DenseNet121 . . . . .	76
<b>Tabla 39.</b>	Resultados experimentales para los indicadores de la hipótesis específica 2 en DenseNet121 . . . . .	76
<b>Tabla 40.</b>	Resultados experimentales para los indicadores de la hipótesis específica 3 en DenseNet121 . . . . .	77
<b>Tabla 41.</b>	Resultados experimentales para los indicadores de la hipótesis específica 1 en EfficientNetB6 . . . . .	77
<b>Tabla 42.</b>	Resultados experimentales para los indicadores de la hipótesis específica 2 en EfficientNetB6 . . . . .	78
<b>Tabla 43.</b>	Resultados experimentales para los indicadores de la hipótesis específica 3 en EfficientNetB6 . . . . .	78
<b>Tabla 44.</b>	Resultados experimentales para los indicadores de la hipótesis específica 4 usando <i>Hard Voting</i> . . . . .	79
<b>Tabla 45.</b>	Resultados experimentales para los indicadores de la hipótesis específica 4 usando <i>Average Soft Voting</i> . . . . .	79
<b>Tabla 46.</b>	Resultados experimentales para los indicadores de la hipótesis específica 4 usando <i>Weighted Soft Voting</i> con algoritmos genéticos	79
<b>Tabla 47.</b>	Prueba de hipótesis específica 1 para InceptionV3 . . . . .	80

<b>Tabla 48.</b>	Prueba de hipótesis específica 1 para ResNet152V2 . . . . .	80
<b>Tabla 49.</b>	Prueba de hipótesis específica 1 para DenseNet121 . . . . .	80
<b>Tabla 50.</b>	Prueba de hipótesis específica 1 para EfficientNetB6 . . . . .	81
<b>Tabla 51.</b>	Prueba de hipótesis específica 2 para InceptionV3 . . . . .	81
<b>Tabla 52.</b>	Prueba de hipótesis específica 2 para ResNet152V2 . . . . .	81
<b>Tabla 53.</b>	Prueba de hipótesis específica 2 para DenseNet121 . . . . .	81
<b>Tabla 54.</b>	Prueba de hipótesis específica 2 para EfficientNetB6 . . . . .	82
<b>Tabla 55.</b>	Prueba de hipótesis específica 3 para InceptionV3 . . . . .	82
<b>Tabla 56.</b>	Prueba de hipótesis específica 3 para ResNet152V2 aplicando búsqueda binaria . . . . .	82
<b>Tabla 57.</b>	Prueba de hipótesis específica 3 para ResNet152V2 aplicando algoritmos genéticos . . . . .	82
<b>Tabla 58.</b>	Prueba de hipótesis específica 3 para DenseNet121 . . . . .	83
<b>Tabla 59.</b>	Prueba de hipótesis específica 3 para EfficientNetB6 . . . . .	83
<b>Tabla 60.</b>	Prueba de hipótesis específica 4 para <i>Hard Voting</i> . . . . .	83
<b>Tabla 61.</b>	Prueba de hipótesis específica 4 para <i>Average Soft Voting</i> . . . . .	83
<b>Tabla 62.</b>	Prueba de hipótesis específica 4 para <i>Weighted Soft Voting</i> apli- cando algoritmos genéticos . . . . .	84
<b>Tabla 63.</b>	Recursos experimentales: <i>datasets</i> y código fuente para su uso libre . . . . .	95

## Lista de Figuras

<b>Figura 1.</b>	Resultados de exactitud para la clasificación de imágenes. . . .	13
<b>Figura 2.</b>	Funcionamiento de una red neuronal convolucional. . . . .	13
<b>Figura 3.</b>	Arquitectura de la red ResNet. . . . .	15
<b>Figura 4.</b>	Arquitectura de la red DenseNet. . . . .	16
<b>Figura 5.</b>	Arquitectura de la red Inception. . . . .	18
<b>Figura 6.</b>	Nevus melanocítico. . . . .	22
<b>Figura 7.</b>	Melanoma. . . . .	22
<b>Figura 8.</b>	Queratosis seborreica. . . . .	23
<b>Figura 9.</b>	Basalioma. . . . .	23
<b>Figura 10.</b>	Queratosis actínica. . . . .	24
<b>Figura 11.</b>	Lesiones vasculares cutáneas. . . . .	24
<b>Figura 12.</b>	Dermatofibroma. . . . .	25
<b>Figura 13.</b>	<i>Data augmentation</i> aplicado en una imagen de rayos X. . . . .	26
<b>Figura 14.</b>	Balaceo de datos aplicando <i>oversampling</i> y <i>undersampling</i> . . . . .	27
<b>Figura 15.</b>	Matriz de confusión. . . . .	28
<b>Figura 16.</b>	Procedimiento de análisis de imágenes de repositorio HAM10000. . . . .	40
<b>Figura 17.</b>	Procedimiento de la hipótesis específica 1. . . . .	41
<b>Figura 18.</b>	Procedimiento de la hipótesis específica 2. . . . .	41
<b>Figura 19.</b>	Procedimiento de la hipótesis específica 3. . . . .	41
<b>Figura 20.</b>	Procedimiento de la hipótesis específica 4. . . . .	42
<b>Figura 21.</b>	HAM10000: disgregación de imágenes por tipo de Lesión. . . . .	45
<b>Figura 22.</b>	HAM10000: disgregación de imágenes por tipo de Lesión con aumento de imágenes. . . . .	46
<b>Figura 23.</b>	Modelo DCNN. . . . .	47
<b>Figura 24.</b>	Búsqueda binaria modificada. . . . .	48
<b>Figura 25.</b>	Diagrama de flujo de congelación por búsqueda binaria. . . . .	49
<b>Figura 26.</b>	Diagrama de flujo de congelación por algoritmos genéticos. . . . .	49
<b>Figura 27.</b>	Comparativo de evaluación entre <i>datasets</i> para InceptionV3. . . . .	51
<b>Figura 28.</b>	Comparativo de evaluación entre <i>datasets</i> para ResNet152V2. . . . .	52
<b>Figura 29.</b>	Comparativo de evaluación entre <i>datasets</i> para DenseNet121. . . . .	53
<b>Figura 30.</b>	Comparativo de evaluación entre <i>datasets</i> para EfficientNetB6. . . . .	54



<b>Figura 31.</b> Comparativo de evaluación usando <i>transfer learning</i> para InceptionV3. . . . .	55
<b>Figura 32.</b> Comparativo de evaluación usando <i>transfer learning</i> para ResNet152V2. . . . .	56
<b>Figura 33.</b> Comparativo de evaluación usando <i>transfer learning</i> para DenseNet121. . . . .	57
<b>Figura 34.</b> Comparativo de evaluación usando <i>transfer learning</i> para EfficientNetB6. . . . .	58
<b>Figura 35.</b> Comparativo de métodos de optimización para InceptionV3. . .	60
<b>Figura 36.</b> Comparativo de métodos de optimización para ResNet152V2. .	62
<b>Figura 37.</b> Comparativo de métodos de optimización para DenseNet121. .	64
<b>Figura 38.</b> Comparativo de métodos de optimización para EfficientNetB6. .	66
<b>Figura 39.</b> Comparativo para el método de ensamblaje <i>Hard Voting</i> . . . . .	68
<b>Figura 40.</b> Comparativo para el método de ensamblaje <i>Average Soft Voting</i> . .	69
<b>Figura 41.</b> Comparativo para el método de ensamblaje <i>Weighted Soft Voting</i> usando algoritmos genéticos. . . . .	70
<b>Figura 42.</b> Comparativo de los métodos de ensamblaje. . . . .	71
<b>Figura 43.</b> Presentación de resultados por diagnóstico (selección aleatoria 1)	84
<b>Figura 44.</b> Presentación de resultados por diagnóstico (selección aleatoria 2)	85
<b>Figura 45.</b> Presentación de resultados por diagnóstico (selección aleatoria 3)	85
<b>Figura 46.</b> Artículo científico publicado. . . . .	96
<b>Figura 47.</b> Diploma de participación del SIMBIG 2022. . . . .	97
<b>Figura 48.</b> Diploma de participación del SICDAS 2022. . . . .	98

## RESUMEN

La visión computacional y las redes neuronales convolucionales profundas han tenido un impacto significativo en la actividad médica, siendo utilizadas para el diagnóstico de enfermedades mediante imágenes y detección de patrones. En esta investigación, el objetivo es mejorar el desempeño en la clasificación de imágenes dermatoscópicas en 7 tipos de lesiones mediante el uso de redes neuronales convolucionales profundas con *transfer learning*, congelamiento de capa óptima y ensamblado. Los resultados demuestran una mejora en el desempeño de la clasificación del diagnóstico al aplicar un método novedoso de congelamiento mediante búsqueda binaria mejorando su exactitud hasta un 3.18 %. Además, se compara este enfoque con la propuesta de Nagae, que utiliza algoritmos genéticos para el congelamiento de capas, obteniendo en algunos casos una mejora de hasta un 2.65 % en exactitud. Posteriormente, se aplican diversos métodos de ensamblado para mejorar hasta un 4.69 % en exactitud.

**Palabras clave:** visión computacional, DCNN, *transfer learning*, optimización, ensamblado, clasificación de imágenes, imagen dermatoscópica.

## ABSTRACT

Computer vision and deep convolutional neural networks have had a significant impact on medical activity, being used for disease diagnosis through image analysis and pattern detection. In this research, the goal is to enhance performance in the classification of dermatoscopic images into 7 types of lesions using deep convolutional neural networks with transfer learning, optimal layer freezing, and ensemble methods. The results demonstrate an improvement in diagnostic classification performance by applying a novel freezing method using binary search, enhancing its accuracy by up to 3.18 %. Additionally, this approach is compared to Nagae's proposal, which employs genetic algorithms for layer freezing, achieving in some cases an accuracy improvement of up to 2.65 %. Subsequently, various ensemble methods are applied to further improve accuracy by up to 4.69 %.

**Keywords:** computer vision, DCNN, learning transfer, optimization, ensembling, image classification, dermatoscopic image.

# Capítulo I

## Introducción

### 1.1. Situación problemática

La visión computacional es la rama de la ciencia de la computación cuyo ámbito es la clasificación y manipulación de imágenes y videos (Shanmugamani, 2018), en el contexto de la inteligencia artificial y redes neuronales se desarrollan las redes neuronales convolucionales (CNN), si tenemos una cantidad suficiente de imágenes con su clasificación, estas redes pueden detectar patrones e inferir en la clasificación de nuevas imágenes. A partir de las CNN se han construido modelos llamados redes neuronales convolucionales profundas (DCNN), este tipo de redes tienen la capacidad de resolver problemas de clasificación de imágenes con modelos jerárquicos de millones de parámetros y bases de datos de gran tamaño (Picazo Montoya, 2018).

En el ámbito de la telemedicina el uso de la visión computacional y en especial las DCNN ayudan al soporte de la actividad médica ya que puede clasificar por medio de patrones de imágenes los diagnósticos de algunas enfermedades. The International Agency for Research on Cancer (IARC) estimó para el 2020 que el cáncer de piel tendría 324,000 nuevos casos a nivel mundial (*Cancer Today*, s.f.-a), para Latinoamérica y el Caribe estimó 18,811 nuevos casos (*Cancer Today*, s.f.-b) y finalmente para el Perú estimó un total de 1,282 nuevos casos (*Cancer Today*, s.f.-c).

En vista de esta situación problemática la comunidad médica desarrolló métodos de detección para el cáncer de piel como la dermatoscopia y la biopsia, pero por medio de la visión computacional es posible la detección de melanomas u otras lesiones y por tanto servir a la comunidad médica. *Melapp: Aplicativo Móvil que detecta El Cáncer de Piel* (2019) es una aplicación móvil impulsado en conjunto por "Innovate Perú" y la Universidad Cayetano Heredia permite detectar lesiones de piel sospechosas, permitiendo al paciente subir la imagen de la lesión en la piel por medio de su teléfono móvil y enviarlo para su análisis a un dermatólogo para su seguimiento.

Con respecto a la clasificación de lesiones en la piel por medio de imágenes, la revisión

sistemática de literatura en este campo es extensa, especialmente en la forma de tener una predicción más acertada aplicando DCNN, KNN, *random forest*, entre otros, pero aún no existe un estudio que profundice en la forma de como mejorar una DCNN con *transfer learning* por medio del congelamiento de capas en la mejora del desempeño en la clasificación del diagnóstico, uno de los primeros trabajos no relacionados con la telemedicina es el presentado por Nagae et al. (2020), enfocado algoritmos genéticos usando los *datasets* CIFAR-10 y CIFAR-100 presentando una mejora entre el 12 % y 15 % en exactitud y concluyendo que este método puede ser aplicado a otros *datasets*, siendo el trabajo de de Lima Mendes et al. (2021) enfocado en la mejora de la precisión en el diagnóstico de neumonía logrando una mejora del 2 %.

Por tanto, la presente investigación se enfoca en la mejora del desempeño de la clasificación del diagnóstico para 7 tipo de lesiones en la piel, creando un novedoso método de búsqueda binaria que permite mejorar los indicadores de desempeño en la clasificación de diagnóstico y comparándolo con el método presentado por Nagae et al. (2020) en distintas DCNN. Finalmente, para mejorar los resultados obtenidos se aplican distintos métodos de ensamblaje, el cual puede ayudar a posteriores investigaciones en crear marcos de trabajo relacionados a las DCNN en la tarea de clasificación de imágenes.

## 1.2. Formulación del problema

### 1.2.1. Problema general

¿Se puede mejorar el desempeño de las DCNN (*Deep Convolutional Neural Networks*) haciendo uso del *transfer learning*, congelamiento de capa óptima y ensamblado en la tarea de clasificación de diagnóstico de lesiones en imágenes dermatoscópicas?.

### 1.2.2. Problemas específicos

- ¿Cuál será la estrategia adoptada si el conjunto de datos utilizado para entrenar y validar el modelo DCNN no está equilibrado?
- ¿En qué medida el *transfer learning* mejora el desempeño de los modelos DCNN en la clasificación de lesiones en imágenes dermatoscópicas?
- ¿En qué medida el congelamiento de capa mejora el desempeño de los modelos DCNN en la clasificación de lesiones en imágenes dermatoscópicas?
- ¿En qué medida el ensamblado mejora el desempeño en la clasificación de lesiones en imágenes dermatoscópicas?

### **1.3. Justificación teórica de la investigación**

Las redes neuronales convolucionales profundas en la clasificación de imágenes dermatoscópicas representa un avance significativo en la medicina, ya que estas tecnologías han demostrado ser efectivas en el análisis de patrones en imágenes médicas, lo que es esencial en el diagnóstico de enfermedades cutáneas. La implementación de técnicas como *transfer learning* y la identificación de capas a congelar son fundamentales para aprovechar el conocimiento previo y mejorar el desempeño en la clasificación del diagnóstico. Además, la comparación con enfoques previos, como el de Nagae et al. (2020) y compararlo con la búsqueda binaria podría ayudar a posteriores investigaciones relacionadas a mejorar el desempeño en la clasificación de imágenes para diferentes tipos de tarea. Finalmente, explorar diferentes métodos de ensamblado podría abrir posteriores investigaciones para su uso en diferentes tareas que no necesariamente estén relacionados con la tarea de clasificación.

### **1.4. Justificación práctica de la investigación**

La detección temprana del cáncer de piel, como señala Huanca-Huirse et al. (2020), es fundamental para obtener resultados exitosos en el tratamiento y aumentar las tasas de supervivencia de los pacientes. Aunque la dermatoscopia ofrece una prometedora herramienta para la detección temprana de lesiones cutáneas cancerosas, su interpretación por parte de los dermatólogos puede ser subjetiva y consume tiempo. Las redes neuronales convolucionales profundas representan una solución automatizada y precisa para la clasificación de lesiones dermatoscópicas, lo que puede mejorar significativamente la exactitud y precisión del diagnóstico del cáncer de piel. Además, la utilización de estas redes en la clasificación de lesiones dermatoscópicas puede reducir la necesidad de realizar biopsias innecesarias, lo cual es costoso y traumático para los pacientes.

### **1.5. Objetivos de la investigación**

#### **1.5.1. Objetivo general**

Mejorar el desempeño de las DCNN por medio del *transfer learning*, congelamiento de capa óptima y ensamblado en la tarea de clasificación de lesiones en imágenes dermatoscópicas.

## 1.5.2. Objetivos específicos

- Establecer la estrategia a considerarse en caso el conjunto de datos no se encuentre equilibrado.
- Determinar en qué medida el *transfer learning* mejora el desempeño de los modelos DCNN en la clasificación de lesiones en imágenes dermatoscópicas.
- Determinar en qué medida el congelamiento de capa de los modelos DCNN mejora el desempeño en la clasificación de lesiones de imágenes dermatoscópicas.
- Determinar en qué medida los métodos de ensamblado mejoran el desempeño en la clasificación de lesiones de imágenes dermatoscópicas.

## 1.6. Hipótesis de la investigación

### 1.6.1. Hipótesis general

La utilización de modelos DCNN usando el *transfer learning*, el congelamiento de capa óptima por medio de la búsqueda binaria y el ensamblado por medio de algoritmos genéticos tienen un impacto significativo en la mejora del desempeño en la tarea de clasificación de lesiones en imágenes dermatoscópicas.

### 1.6.2. Hipótesis específicas

- El uso de la técnica de *data augmentation* y aplicación de pesos es la mejor estrategia cuando el conjunto de datos no se encuentre equilibrado.
- El *transfer learning* mejora significativamente el desempeño de los modelos DCNN en la clasificación de lesiones en imágenes dermatoscópicas.
- Los métodos de congelamiento por búsqueda binaria mejora significativamente el desempeño en la clasificación de las lesiones en imágenes dermatoscópicas.
- El ensamblado por medio de algoritmos genéticos mejora significativamente el desempeño en la clasificación de las lesiones de imágenes dermatoscópicas.

# Capítulo II

## Marco Teórico

### 2.1. Antecedentes de investigación

H. Li et al. (2021) realizaron una revisión de métodos de *deep learning* y su aplicación en el diagnóstico de enfermedades en la piel. La metodología se basa en el estudio sistemático de métodos de *deep learning* para el diagnóstico de enfermedades en la piel desde el año 1990 hasta el 2017 y los *datasets* con imágenes dermatoscópicas disponibles. El estudio concluye que se han propuesto numerosos trabajos utilizando métodos de *deep learning* para el diagnóstico de enfermedades de la piel con resultados prometedores, sin embargo, existen diversos desafíos que deben abordarse antes de que el *deep learning* pueda ser ampliamente aplicado en escenarios clínicos reales para el diagnóstico de enfermedades de la piel. Estos desafíos incluyen la limitada clasificación de enfermedades en la piel, conjuntos de datos desbalanceados, datos no confiables debido a fuentes heterogéneas de imágenes de piel y la falta de diversidad entre los casos en los conjuntos de datos existentes sobre enfermedades de la piel.

Pratiwi et al. (2021) realizaron un trabajo con respecto a la clasificación de lesiones en la piel utilizando técnicas de *deep ensemble learning*. La metodología utilizada se centra en la experimentación de tres arquitecturas de redes neuronales convolucionales (DCNN): Inception V3, Resnet V2 y DenseNet. Posteriormente, se emplea el ensamblado de modelos mediante un sistema de votación. Los datos utilizados para el entrenamiento consistieron en 10,015 imágenes dermatoscópicas proporcionadas por el conjunto de datos HAM10000. El estudio concluye con los resultados siguientes: 97.23 % de exactitud, 90.12 % de sensibilidad, 97.73 % de especificidad, 82.01 % de precisión, y 85.01 % en F1-Score.

Arkah et al. (2021) realizaron un trabajo con respecto a la clasificación de cáncer en la piel del tipo benigno y maligno utilizando el *transfer learning*. La metodolo-



gía se basa en la experimentación de las siguientes arquitecturas VGG19, GoogleNet y Resnet50 con *transfer learning*. Para el entrenamiento de los modelos se utilizaron 23,906 imágenes dermatoscópicas propuestas por el *ISIC Archive* y un segundo *dataset* compuesto de 33,000 imágenes dermatoscópicas propuesto por la SIIM-ISIC 2020. Concluyen su estudio con los siguientes resultados: el mejor modelo es el generado por la arquitectura Resnet50 con una exactitud de 93.7 %, precisión de 95.7 %, recall de 94.6 % y 95.1 % de F1-score.

Coronado Pérez (2018) realizó un método de procesamiento de imágenes no dermatoscópicas para la detección del cáncer de piel. Este método se basa en el uso de redes neuronales convolucionales y autocodificadores, que permiten realizar un procesamiento eficiente de las imágenes y extraer características relevantes para la detección del cáncer de piel. La metodología empleada se fundamenta en el uso de autocodificadores, que son algoritmos de aprendizaje no supervisado, para la clasificación de 2,360 imágenes en tres clases principales: benigno, premaligno y maligno. El estudio concluye presentando los resultados obtenidos, los cuales se comparan con el ganador del concurso ISBI 2016, a pesar de no superar el primer lugar a nivel de exactitud y ROC AUC score, sus experimentos superan en resultados su nivel de especificidad.

Villa-Pulgarin et al. (2022) realizaron un trabajo con respecto a la optimización de modelos DCNN para clasificaciones de lesiones en la piel. La metodología se centra en el estudio de tres arquitecturas de redes neuronales convolucionales (DCNN): DenseNet201, InceptionResNetV2 e InceptionV3. Estas arquitecturas se investigan utilizando el conjunto de datos HAM10000. Para abordar el desafío del desequilibrio de datos, se utiliza la técnica de aumento de datos (*data augmentation*) y se emplea la técnica de *transfer learning* para mejorar la precisión de las predicciones. Los hallazgos del estudio actual revelaron una precisión del 98 %, 97 % y 96 % respectivamente. Posteriormente, se validó el modelo más sobresaliente (DenseNet201) utilizando el conjunto de datos del ISIC 2019, obteniendo una exactitud del 93 %.

Jojoa Acosta et al. (2021) realizaron un trabajo con respecto a la detección y clasificación de melanomas. La metodología se basa en el uso de *deep learning* basado en redes neuronales convolucionales, específicamente la arquitectura Resnet-152 y el uso de *Mask RCNN* para la detección de melanomas en 2 clases, las benignas y las malignas utilizando como *dataset* los propuestos por el ISIC del año 2017 que consiste en 1,995 imágenes de entrenamiento, 149 de validación y 598 imágenes de prueba. Los hallazgos de este estudio encontraron un nivel de exactitud de 90.4 %, sensibilidad del 82 % y especificidad de 92.5 %.

Moataz et al. (2021) realizaron un trabajo con respecto a la clasificación de lesiones cutáneas. El enfoque metodológico se centra en la utilización de redes neuronales convolucionales profundas (DCNN) con *transfer learning* aplicado al conjunto de datos HAM10000, se emplea la arquitectura Xception para extraer la matriz de características y se utiliza otra arquitectura de red neuronal para clasificar las imágenes en función de dicha matriz. Con el fin de abordar el desequilibrio de datos, los autores aplican la técnica de *data augmentation*. Los hallazgos de este estudio encontraron una precisión, sensibilidad y f1-score del 96 %.

Shawon et al. (2021) realizaron un trabajo con respecto a la detección y clasificación de melanomas benigno o maligno. La metodología empleada se centra en un proceso de preprocesamiento de imágenes y segmentación para identificar la región de interés. Estas características se utilizan para clasificar la imagen. Es importante destacar que en este estudio no se utiliza una red preentrenada, sino que se construye una red convolucional desde cero. Además, se realiza un estudio comparativo entre diferentes técnicas, como KNN o *random forest*. Los datos de entrada utilizados en este estudio son los proporcionados por el *ISIC Challenge 2016*. Los hallazgos de este estudio encontraron que las redes convolucionales son mejores que el KNN y *random forest*, siendo los resultados para las redes convolucionales una precisión de 80.47 %, una especificidad de 52.63 % y una sensibilidad de 81.94 %.

Alagu y Bagan (2021) realizaron una investigación con respecto a la clasificación de melanomas, en este caso lo divide en 3 tipos: basal, escamosa y melanoma. La metodología usada para este trabajo es el uso de la arquitectura DenseNet, el uso de la técnica de *data augmentation* y el uso de optimizadores tales como RMSProp, Adam y SGD. Los hallazgos de este estudio encontraron que el uso del optimizador SGD tiene un mejor resultado con respecto del RMSProp y Adam, con una precisión del SGD del 95 %, 85 % para el RMSProp y 70 % para Adam.

Kumar et al. (2021) realizaron una investigación con respecto a la clasificación de melanomas benigno y maligno. La metodología empleada en este trabajo consiste en desarrollar una red neuronal convolucional desde cero. Además, se realiza un estudio comparativo entre dos optimizadores, Adam y RMSProp. Los datos utilizados como entrada en este estudio son proporcionados por el ISIC. Los resultados de este estudio demostraron que el uso de Adam es mejor que el de RMSProp, con una precisión de entrenamiento del 90 % en Adam con respecto a un 87 % que ofrece el uso de RMSProp, para añadir la precisión de validación de ambos optimizadores son iguales con un 82 %.

Karki et al. (2021) realizaron un trabajo de investigación al respecto a la clasificación de melanomas benigno y maligno. La metodología empleada para este trabajo se toma como datos de entrada los provistos por ISIC y del SIIM-ISIC. Un punto interesante es el uso del ensamblado para mejorar las predicciones siendo el *log ensemble* el que dio un mejor resultado. Asimismo, realizaron su estudio modificando las resoluciones de las imágenes para una mejor comparativa (resoluciones usadas:  $512 \times 512$ ,  $456 \times 456$ ,  $384 \times 384$ ,  $300 \times 300$  y  $256 \times 256$  pixeles). Los resultados del estudio demostraron que el uso de *log ensemble* es mejor que usar el *average ensemble*.

Rahman et al. (2021) realizaron una investigación con respecto a la clasificación de lesiones en la piel, en este caso para 7 tipos de lesiones. La metodología usada para este trabajo es un estudio comparativo entre distintas redes neuronales convolucionales profundas (DCNN) utilizando como datos de entrada el repositorio HAM10000 y el *dataset* provisto por el ISIC 2019, para realizar el ensamblado utilizan el *average ensemble*. Los resultados del estudio demostraron que la arquitectura DenseNet con un 91 % de *recall*, frente a 88 %, 89 %, 88 % y 84 % obtenido por las arquitecturas ResNetXt, SeResNeXt, Xception y ResNet respectivamente. Finalmente, al usar los métodos de ensamblado los resultados mejoraron considerablemente a nivel de *recall* obteniendo un resultado del 93 % para el *average ensembling* y 94 % para el *weighted ensembling*.

Al-Masni et al. (2020), realizaron un trabajo de investigación con respecto a la clasificación de lesiones en la piel, usando para su estudio pruebas en 2, 3 y 7 tipos de lesiones. La metodología usada para este trabajo es un estudio comparativo entre distintas redes neuronales convolucionales profundas (DCNN), para sus datos de entrada utilizaron el datos propuestos por la ISIC para el año 2016 (2 tipos de lesión), 2017 (3 tipos de lesión) y 2018 (7 tipos de lesión), En todos los escenarios abordados en relación al problema del desbalance de clases, se aplicó la técnica del data augmentation. Además, se asignaron pesos por clase para tener en cuenta la desproporción entre ellas durante el entrenamiento del modelo. Asimismo, se empleó un proceso de pre-procesamiento de segmentación de imagen utilizando FrCN para mejorar la calidad y precisión de las segmentaciones realizadas en las imágenes. Los hallazgos del estudio encontraron que el uso de segmentación de imagen si tiene un impacto positivo en todas las arquitecturas DCNN mayor al 4 %.

Mahbod et al. (2020), cuyo objetivo es la clasificación de lesiones en la piel, usando para su estudio pruebas en 7 tipos de lesiones. La metodología usada para este trabajo es un estudio comparativo entre distintas redes neuronales convolucionales profundas (DCNN), teniendo como datos de entrada los provistos por el ISIC 2018 (7 tipos de

lesión) con distintas resoluciones y niveles de ensamblado. Los hallazgos del estudio encontraron que el hecho de haber reducido las imágenes por medio del *downsampling* y la segmentación de imagen además de un ensamblado multinivel de resoluciones y modelos tienen un resultado de precisión promedio de 96.3 % y un ROC AUC score de 98.1 %, superando al primer lugar del concurso ISIC 2018 en términos de precisión (95.8 %), pero no de ROC AUC score (98.3 %).

Grove y Green (2020) realizaron un trabajo de investigación con respecto a la clasificación de lesiones en la piel, usando para su estudio pruebas en 2 tipos de lesiones: melanoma maligno y benigno. La metodología usada para este trabajo es el uso de *transfer learning* con el modelo DCNN ResNet50, para los datos de entrada tomaron los archivos disponibles de la ISIC y para las validaciones usaron los archivos disponibles de Nueva Zelanda Dermnet NZ. Los hallazgos del estudio encontraron que el uso de 2 distintas fuentes de datos (ISIC para entrenamiento y Dermnet NZ para validación) dan unos resultados prometedores para la detección de melanomas malignos y benignos con un 86 % de precisión con la fuente de datos de la ISIC y un 86.7 % con los datos de validación propuestos por Dermnet NZ.

Jain et al. (2021) realizaron un trabajo de investigación con respecto a la clasificación de lesiones en la piel, usando para su estudio pruebas en 7 tipos de lesiones. La metodología usada para este trabajo es un estudio comparativo entre distintas redes neuronales convolucionales profundas (DCNN), como datos de entrada se utiliza la fuente de datos HAM10000, el problema con el desbalance de clases se resolvió por medio de la técnica del *data augmentation*, finalmente optaron para las pruebas reducir la resolución de la imagen a 128x128 píxeles, al utilizar DCNN ya entrenadas están ya cuentan con *transfer learning*. Los resultados del estudio demostraron que el modelo Xception es el que mejor resultados obtuvo en términos de precisión (90.48 %) con respecto a los otros modelos: VGG-19 (66.36 %), ResNet50 (77.6 %), InceptionResNetV2 (85.58 %), InceptionV3 (79.23 %) y MobileNet (88.57 %).

## **2.2. Bases Teóricas**

### **2.2.1. Visión computacional**

Según la Real Academia Española (RAE), la visión se puede interpretar como el acto de utilizar los ojos para percibir y procesar información visual del entorno. Esta capacidad nos permite detectar objetos, clasificarlos y también reconocer comportamientos de interacción entre ellos. Por ejemplo, podemos distinguir entre perros y gatos, detectar fugas de agua en una tubería, apreciar la danza o seguir la trayectoria de una pelota.

En resumen, la visión nos permite obtener información visual del entorno y utilizarla para comprender y interactuar con el mundo que nos rodea.

Según Sucar y Gómez (2011) con respecto a la visión computacional busca emular de alguna manera la capacidad humana de ver y percibir imágenes a través de las computadoras. Su objetivo es interpretar las imágenes adquiridas para poder reconocer los distintos objetos presentes en el entorno y determinar su posición en el espacio. Mediante el procesamiento de estas imágenes, se aplican algoritmos y técnicas para extraer características visuales, como formas, colores y texturas, y utilizar esa información para realizar tareas como detección de objetos, reconocimiento facial, seguimiento de movimiento, entre otras aplicaciones. La visión computacional es un campo de estudio multidisciplinario que combina conocimientos de matemáticas, estadística, inteligencia artificial y procesamiento de imágenes para permitir a las computadoras "ver" y comprender el mundo visualmente, por tanto, la visión computacional es el procedimiento de crear y emular la capacidad visual humana mediante el uso de software y hardware. Se trata de una disciplina que investiga cómo reconstruir, analizar y comprender una escena en función de las propiedades estructurales presentes en dicha escena.

### 2.2.2. Clasificación de imágenes

La clasificación es el proceso de asegurar que las imágenes no clasificadas se incluyan en su clase dentro de ciertas categorías como menciona Lee et al. (2018). Para la clasificación de imágenes existen técnicas cada una con sus ventajas y limitaciones. Gavali y Banu (2019) realizó un trabajo comparativo entre las redes neuronales, SVM (*support vector machine*), lógica difusa y algoritmos genéticos. En la Tabla 1 se observa las ventajas y desventajas de estas 4 técnicas.

**Tabla 1**

*Comparativo de beneficios y limitaciones de técnicas para clasificación de imágenes.*

<b>Técnica de clasificación</b>	<b>Beneficios</b>	<b>Supuestos y/o limitaciones</b>
Redes neuronales	Puede ser usado para clasificación o regresión Capaz de representar funciones booleanas (AND, OR, NOT) Tolerante a entradas distorsionadas.	Estructura de algoritmo de difícil comprensión. Demasiados atributos pueden resultar en un sobreajuste. La estructura de la red óptima solo puede ser obtenida mediante experimentación.

		Las instancias se pueden clasificar por más de una salida.	
Support Vector Machine (SVM)	Modela límites de clase no lineales.	El entrenamiento es lento en comparación con Bayes y árboles de decisión.	
	Es poco probable que se produzca un sobreajuste.	Es difícil determinar los parámetros óptimos cuando los datos de entrenamiento no se pueden separar linealmente.	
	Es posible manejar la complejidad de la regla de decisión y ajustar la frecuencia de error de manera sencilla.	Estructura de algoritmo de difícil comprensión.	
Lógica difusa	Se pueden identificar diferentes relaciones estocásticas para describir propiedades.	El conocimiento previo es muy importante para obtener buenos resultados.	
		No se obtienen soluciones precisas si la dirección de la decisión no es clara.	
Algoritmos genéticos	Se puede utilizar en la clasificación y selección de características.	El cálculo o desarrollo de la función de puntuación no es trivial.	
	Se utiliza principalmente en optimización. Siempre encuentra una "buena" solución (no siempre la mejor solución).	No es el método más eficiente para la solución óptima, en lugar de la global.	
	Puede manejar espacios grandes, complejos, indiferenciables y multimodales.	Complicaciones involucradas en la representación de datos de entrenamiento / salida.	
	Método de búsqueda eficiente para un espacio de problemas complejos.		
	Bueno para refinar características irrelevantes y ruidosas seleccionadas para su clasificación.		

*Fuente.* Tabla traducida y resumida de Gavali y Banu (2019)

A continuación, Gavali y Banu (2019) en la Tabla 2 realiza un análisis comparativo de ellas en función de los parámetros: tipo de enfoque, límites de decisiones no lineales, velocidad de entrenamiento, precisión y rendimiento general.

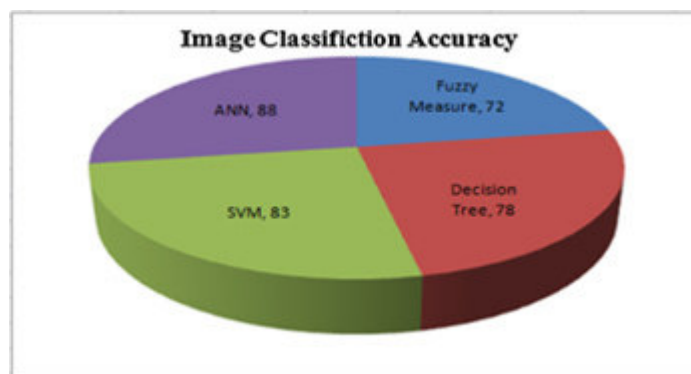
**Tabla 2**

*Comparativo de técnicas de clasificación de imágenes en función a los parámetros.*

<b>Parámetro</b>	<b>Redes neuronales</b>	<b>SVM</b>	<b>Lógica difusa</b>	<b>Algoritmos genéticos</b>
Tipo de enfoque	No paramétrico	No paramétrico con clasificador binario	Estocástico	Grandes datos de series de tiempo
Límites de decisión no lineales	Eficiente cuando los datos tienen solo unas pocas variables de entrada	Eficiente cuando los datos tienen más variables de entrada	Depende del conocimiento previo para los límites de decisión	Depende de la dirección de la decisión
Velocidad de entrenamiento	Estructura de la red, tasa de impulso, tasa de aprendizaje, criterios de convergencia	Tamaño de los datos de entrenamiento, parámetro del kernel, separabilidad de clases	Aplicación iterativa de la integral difusa	Refinando genes irrelevantes y de ruido
Precisión	Depende del número de clases de entrada	Depende de la selección del hiperplano óptimo.	Selección de umbral de corte	Selección de genes
Rendimiento general	Estructura de red	Parámetro de kernel	Integral difusa fusionada	Selección de características

*Fuente.* Tabla traducida y resumida de Gavali y Banu (2019)

Finalmente, presenta la exactitud obtenida por estas 4 técnicas en la Figura 1, donde se observa que las redes neuronales obtienen una mejor exactitud con respecto a las otras 3 técnicas.



**Figura 1**

*Resultados de exactitud para la clasificación de imágenes.*

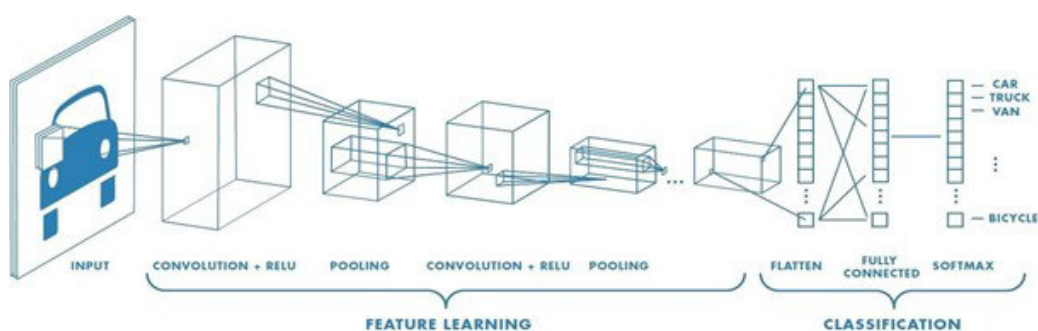
*Fuente.* Tomado del trabajo de Gavali y Banu (2019)

### 2.2.3. Redes neuronales convolucionales (CNN).

Las redes neuronales convolucionales (CNN) son un tipo de arquitectura que ha demostrado ser exitosa en diversas tareas de reconocimiento de objetos. Al aplicar filtros para procesar imágenes, es posible reducir de manera significativa la cantidad de parámetros que la red necesita procesar, como menciona Coronado Pérez (2018).

Uno de los casos de uso más extendidos para las redes neuronales convolucionales es el procesamiento de imágenes, específicamente la clasificación de imágenes (Ferreiras Extremo, 2021).

Las redes neuronales convolucionales se asemejan al proceso visual del ojo humano al identificar características de un objeto. En estas redes, las primeras capas se encargan de detectar patrones básicos, como curvas y formas, y a medida que se avanza en las capas, se especializan en características más complejas. Las capas finales funcionan como una red neuronal convencional para clasificar el objeto (Figura 2).



**Figura 2**

*Funcionamiento de una red neuronal convolucional.*

*Fuente.* Imagen tomada de Saha (2018)

Según LeCun et al. (2015) las redes neuronales convolucionales profundas han lo-



grado avances significativos en el procesamiento de imágenes, videos, voz y audio, permitiendo tareas como reconocimiento de objetos, segmentación de imágenes y generación de música. Por otro lado, las redes neuronales recurrentes han proporcionado avances en la comprensión de datos secuenciales, como texto y voz, permitiendo aplicaciones como el reconocimiento de voz, traducción automática y generación de texto coherente.

#### **2.2.4. Redes neuronales convolucionales profundas (DCNN).**

Como afirma Picazo Montoya (2018) Las redes neuronales convolucionales profundas (DCNN) son un tipo de arquitectura que ha demostrado su eficacia en la resolución de problemas de clasificación de imágenes. Utilizando un enfoque jerárquico, estas redes están compuestas por millones de parámetros y se benefician de grandes bases de datos para lograr un alto rendimiento en la tarea de clasificación de imágenes.

En los últimos años se han realizado esfuerzos con respecto a este tipo de redes neuronales convolucionales profundas, todas estas tienen en común la competencia de clasificación de imágenes propuestas por IMAGENET, esta cuenta con 1.2 millones de imágenes a clasificar en 1000 categorías. El artículo presentado por Krizhevsky et al. (2012) demuestra los retos encontrados para el procesamiento de clasificación de imágenes – tomando en cuenta que su trabajo original fue presentado el 2012 – siendo una de estas las limitaciones de GPU, no contar con un procesador GPU hace la tarea de trabajar con este tipo de redes más pesada, aunque algunos artículos posteriores utilizan el procesador TPU. Otro de los problemas a enfrentar en la clasificación de imágenes es el sobreajuste lo cual puede causar errores al momento de predecir las clasificaciones, para ello Krizhevsky et al. (2012) apuesta por una capa de regularización de *dropout* el cual da una mayor flexibilidad y evita este sobreajuste.

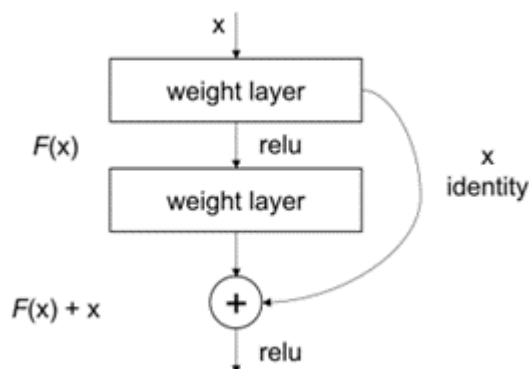
Finalmente, el modelo DCNN propuesta por Krizhevsky et al. (2012) (llamada *Supervision*) fue la ganadora el año 2012 del concurso ImageNet LSVRC-2012 con un error del 15.3 % comparada con la del segundo lugar que fue de un 26.2 %.

#### **ResNet.**

ResNet es una arquitectura CNN presentada por He et al. (2016) cuya particularidad radica en que emplea conexiones residuales para mejorar el proceso de entrenamiento de redes neuronales profundas. Las redes neuronales profundas tienen la capacidad de aprender características complejas a partir de datos, pero a menudo enfrentan el desafío de la desaparición del gradiente durante el entrenamiento. Este problema ocurre cuando los gradientes se vuelven cada vez más pequeños a medida que se propagan a través de las capas de la red, dificultando el aprendizaje de las capas intermedias.

La arquitectura ResNet aborda el problema de la desaparición del gradiente al intro-

ducir conexiones residuales, que permiten que los gradientes fluyan directamente a través de la red sin disminuir su magnitud. Esto se representa visualmente en la Figura 3, donde se muestra cómo las conexiones residuales facilitan el flujo de información a través de las capas de la red.



**Figura 3**

*Arquitectura de la red ResNet.*

*Fuente.* Tomado del libro de Shanmugamani (2018)

Adicionalmente, para incorporar las conexiones residuales en la red, se utilizan bloques residuales que constan de una capa de convolución seguida de una capa de normalización de lote y otra capa de convolución. La salida de estas capas se suma a la entrada de los bloques residuales, formando así una ruta de atajo. Esta ruta de atajo permite que los gradientes fluyan directamente a través de la red, evitando así la desaparición del gradiente y facilitando el entrenamiento de redes neuronales profundas. Los investigadores llevaron a cabo el entrenamiento de la red ResNet utilizando el conjunto de datos de ImageNet. Como resultado, la red ResNet logró un destacado nivel de precisión, con una tasa de error del 3.57 % al clasificar imágenes en el conjunto de datos de validación de ImageNet. Este rendimiento sobrepasó notablemente a las redes neuronales convolucionales más profundas disponibles en ese momento.

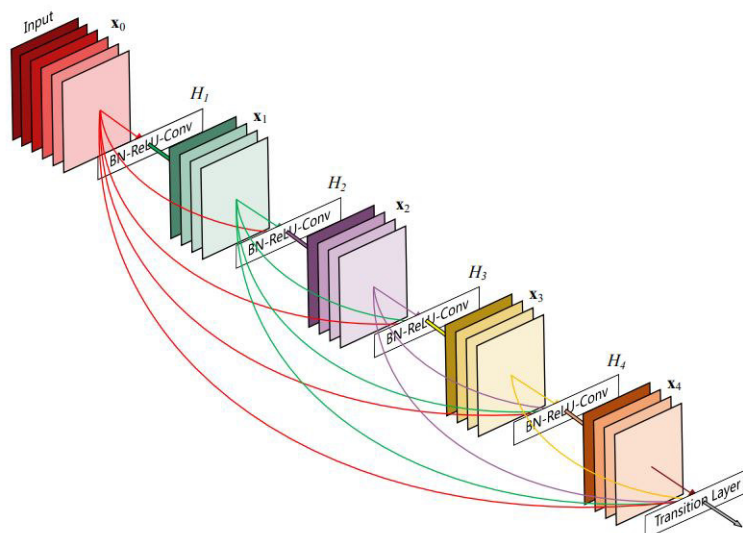
Además, los investigadores realizaron experimentos para evaluar la capacidad de la arquitectura ResNet para transferir el conocimiento adquirido en el conjunto de datos de ImageNet a otras tareas de clasificación de imágenes. Los resultados obtenidos demostraron que la red ResNet preentrenada en ImageNet obtuvo un rendimiento notablemente superior en comparación con las redes entrenadas desde cero en diversos conjuntos de datos de clasificación de imágenes. Esto indica que la transferencia de aprendizaje utilizando la red ResNet preentrenada puede ser altamente efectiva en diferentes dominios y tareas de clasificación de imágenes.

## DenseNet.

Huang et al. (2017) presentan una nueva arquitectura llamada DenseNet. El propósito que impulsa el desarrollo de esta arquitectura es solucionar el desafío de la desaparición de gradientes en las redes neuronales de gran profundidad.

La desaparición de gradientes es un problema común en las redes neuronales profundas, en el que el gradiente se desvanece a medida que se propaga hacia capas más profundas, lo que dificulta el entrenamiento de la red y puede afectar negativamente el rendimiento de la red. Para abordar este problema, DenseNet utiliza conexiones densas entre todas las capas de la red, lo que permite que las capas reciban información de todas las capas previas, mejorando la propagación de gradientes.

En una red neuronal convolucional convencional, las capas están conectadas en secuencia, lo que significa que la salida de cada capa se utiliza como entrada para la siguiente capa. Esta estructura de conexión secuencial se ilustra en la Figura 4.



**Figura 4**

*Arquitectura de la red DenseNet.*

*Fuente.* Tomado del trabajo de Huang et al. (2017)

En DenseNet, todas las capas están conectadas directamente entre sí en una estructura densamente conectada. Esto implica que cada capa recibe información de todas las capas anteriores, lo que facilita el flujo de información a través de la red y mejora la propagación de los gradientes.

DenseNet también introduce el concepto de "bloques densamente conectados", que son unidades de construcción básicas de la red. Cada bloque de densidad contiene múltiples capas densamente conectadas, esto posibilita que la información se transmita de manera más eficiente a lo largo de la red. Además, los bloques de densidad permiten la reutilización de características aprendidas en capas anteriores, lo que re-

duce la cantidad de parámetros necesarios para entrenar la red y mejora la eficiencia computacional.

Además de la comparación con otras arquitecturas de redes neuronales, los autores también realizan un análisis ablativo para investigar la contribución de diferentes componentes de DenseNet al rendimiento de la red. Este análisis muestra que la densa conectividad es la característica clave que contribuye al rendimiento mejorado de DenseNet.

Los resultados experimentales muestran que DenseNet es una arquitectura altamente efectiva para la clasificación de imágenes y que supera a otras arquitecturas populares en términos de precisión y eficiencia computacional.

### **EfficientNet.**

Tan y Le (2019) proponen una arquitectura que utiliza una nueva técnica de escalado compuesto con el objetivo de mejorar la precisión de las redes neuronales convolucionales al mismo tiempo que se reduce el costo computacional y la cantidad de parámetros. Los autores plantean que los métodos anteriores de escalamiento, como incrementar la amplitud, la profundidad o la resolución de entrada de la red, presentan limitaciones debido al aumento en la carga computacional y al riesgo de sobreajuste.

En cambio, los autores proponen una técnica de escalado compuesto que combina tres factores: el ancho de la red (número de canales por capa), la profundidad de la red (número de capas) y la resolución de entrada (tamaño de la imagen). Para encontrar la combinación óptima de estos factores, los autores realizan una búsqueda de hiperparámetros en una red base llamada EfficientNet-B0 utilizando el conjunto de datos ImageNet.

Además, los autores llevaron a cabo un análisis de ablación para examinar el efecto de cada factor en la precisión del modelo. Descubrieron que el ancho de la red era el factor más crucial para mejorar la precisión, seguido por la profundidad de la red y la resolución de entrada. Este análisis les permitió comprender mejor la contribución de cada factor y optimizar la arquitectura EfficientNet en consecuencia.

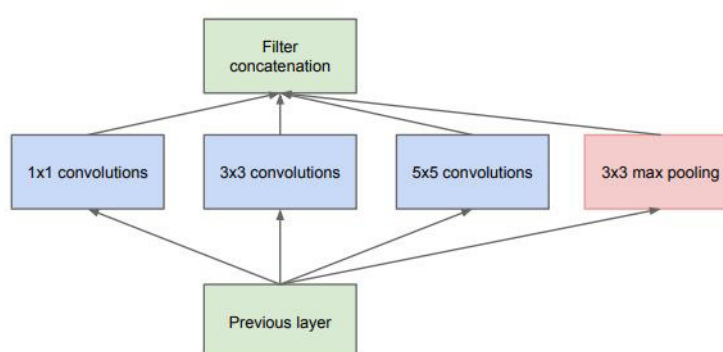
En cuanto a la evaluación del rendimiento, los autores comparan EfficientNet con otras arquitecturas de CNN, como ResNet, DenseNet y MobileNetV2, en términos de precisión y eficiencia computacional. Encuentran que EfficientNet supera a estas arquitecturas en ambos aspectos en el conjunto de datos ImageNet y CIFAR-100.

### **Inception.**

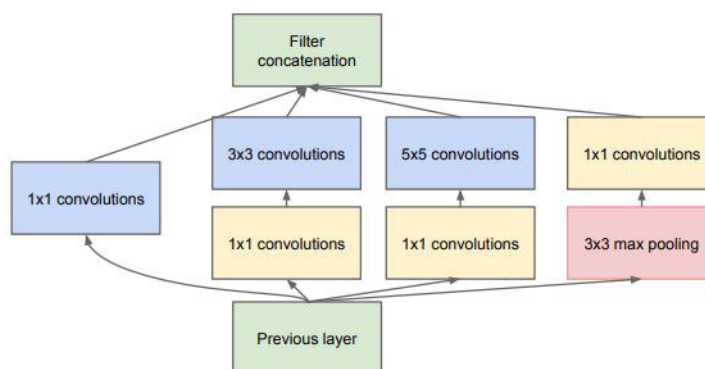
Szegedy et al. (2015) presentan una arquitectura llamada Inception, una red neuronal convolucional diseñada con el objetivo de mejorar tanto la eficiencia como la precisión en tareas de visión por computadora. En su introducción, los autores explican que el

desafío de las redes neuronales convolucionales es encontrar un equilibrio entre la precisión y la eficiencia computacional. Si bien las redes más profundas pueden mejorar la precisión, también requieren más recursos computacionales. Además, las redes más profundas pueden sufrir de problemas de gradiente desvaneciente y sobreajuste.

Para abordar estos desafíos, los autores proponen la arquitectura Inception, que utiliza módulos de convolución de diferentes tamaños para mejorar la eficiencia y la precisión. En lugar de utilizar solo filtros convolucionales de tamaño fijo, la arquitectura Inception utiliza filtros de diferentes tamaños ( $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ ) en paralelo para capturar características a diferentes escalas espaciales. Los autores también utilizan módulos de reducción, que reducen la dimensión de los mapas de características para reducir la carga computacional como se observa en la Figura 5.



(a) Inception module, naïve version



(b) Inception module with dimensionality reduction

### Figura 5

*Arquitectura de la red Inception.*

*Fuente.* Tomado del trabajo de Szegedy et al. (2015)

Los autores presentan una versión simplificada de la arquitectura Inception, que se utiliza para la tarea de clasificación de imágenes en el conjunto de datos ImageNet. La versión simplificada utiliza módulos Inception con filtros  $1 \times 1$  y  $3 \times 3$  y módulos de reducción. La arquitectura resultante es más profunda que las redes convolucionales anteriores y tiene menos parámetros, lo que la hace más eficiente.

Los autores también presentan una versión más profunda de la arquitectura Inception, que utiliza módulos Inception más grandes y más profundos. Esta arquitectura logra una mayor precisión en la tarea de clasificación de imágenes en el conjunto de datos ImageNet que las redes convolucionales anteriores.

Los autores también realizan un análisis detallado de los módulos Inception y cómo contribuyen a la eficiencia y precisión de la red. Encuentran que los módulos con filtros  $1 \times 1$  son efectivos para reducir la dimensión de los mapas de características y para agregar no linealidad a la red, mientras que los módulos con filtros  $3 \times 3$  y  $5 \times 5$  son efectivos para capturar características a diferentes escalas espaciales.

### **2.2.5. *Transfer learning.***

Según B. Li y Rangarajan (2022), *transfer learning* es un concepto mediante el cual se pueden desarrollar modelos basados en datos para tareas con disponibilidad limitada de datos al compartir información de una tarea relacionada.

Esta estrategia ha sido empleada en diversas áreas de investigación, como el procesamiento del lenguaje natural, la visión por computadora, el reconocimiento de voz y más. El propósito fundamental del *transfer learning* es aprovechar el conocimiento adquirido en una tarea para mejorar el desempeño en una tarea relacionada, evitando así la necesidad de entrenar el modelo desde cero. Pan y Yang (2010) define el *transfer learning* como el proceso de utilizar el conocimiento adquirido en una tarea fuente para mejorar el rendimiento en una tarea objetivo. El objetivo principal es aprovechar la información previa para impulsar el desempeño en la nueva tarea. Los autores también explican que el *transfer learning* implica el uso de modelos pre-entrenados, es decir, modelos que han sido entrenados en grandes conjuntos de datos, y que se pueden ajustar o afinar en una tarea específica para mejorar el rendimiento.

Para el ámbito de las DCNN existe un repositorio predefinido para la clasificación de imágenes llamado IMAGENET que contiene la clasificación de 1000 tipos de imágenes, por tanto, para los modelos creados a partir de las arquitecturas DCNN se comparte este conocimiento para poder inferir en nuevas clasificaciones.

### **2.2.6. Congelamiento de capa.**

El congelamiento de capa se podría definir como el hecho de configurar una o más capas específicas de una red neuronal para que tome los valores por defecto provistos por *transfer learning* y no sean modificados en el proceso de entrenamiento, esto puede ser un proceso complicado debido a que los modelos DCNN actuales poseen una cantidad elevada de capas para en sus modelos desde 22 capas para los VGG y más de 500 capas para modelos como ResNet152. Según un estudio de Nagae et al. (2020), la detección de las capas a congelar puede incrementar la precisión entre un 12 % y 15 %.

### **2.2.7. Búsqueda binaria.**

Según la definición de Cormen et al. (2009), la búsqueda binaria es un algoritmo eficiente utilizado para encontrar un elemento en una lista que está ordenada. Funciona dividiendo repetidamente la lista en dos mitades y descartando la mitad que no puede contener al elemento buscado, reduciendo así las ubicaciones posibles a medida que avanza el proceso, hasta que se encuentra el elemento deseado o se determina que no está presente en la lista. La búsqueda binaria tiene una complejidad temporal de  $O(\log n)$ , lo que implica que su tiempo de ejecución aumenta lentamente a medida que crece el tamaño de la lista. En contraste, la búsqueda lineal, que es el método más básico de búsqueda, tiene una complejidad temporal de  $O(n)$ , lo que significa que su tiempo de ejecución aumenta de manera lineal a medida que crece el tamaño de la lista.

Aunque la búsqueda binaria es una técnica muy útil, tiene ciertas limitaciones. Para que la búsqueda binaria funcione correctamente, la lista debe estar ordenada previamente. Además, la búsqueda binaria no se puede utilizar para buscar elementos en una lista que se está actualizando constantemente.

### **2.2.8. Algoritmos genéticos**

Según la definición de Beasley et al. (1993), los algoritmos genéticos son descritos como métodos adaptativos utilizados para resolver problemas de búsqueda y optimización. Estos algoritmos están compuestos por una población de soluciones, una función objetivo para evaluar el rendimiento de los individuos y operadores genéticos, como la selección, la recombinación y la mutación, para generar nuevas soluciones. Además, el artículo presenta algunos ejemplos de aplicaciones de los algoritmos genéticos en diferentes campos, como la planificación de rutas, la optimización de la distribución de recursos y la clasificación de patrones.

### **2.2.9. Métodos de ensamblado.**

Según Dietterich (2000), los métodos de ensamblaje son enfoques de aprendizaje que construyen un conjunto de clasificadores y utilizan la combinación de sus predicciones para clasificar nuevos puntos de datos. Esto se logra mediante el voto ponderado, donde cada clasificador contribuye con su opinión y se asigna un peso a su predicción. Además, afirma que la técnica de ensamblaje es más preciso que usar un solo modelo. A su vez una condición necesaria para la aplicación de los métodos de ensamblado es que los modelos a ensamblar tengan data diversa, es decir, los modelos no pueden tener los mismos datos para su entrenamiento. Entre los métodos de ensamblados se encuentran:

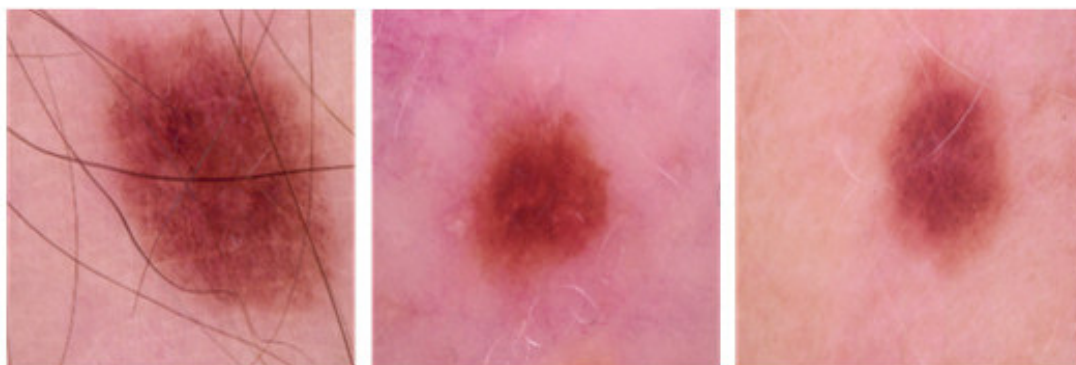
- **Votación bayesiana:** en este tipo de ensamblado se considera un peso por cada modelo y se realiza una suma ponderada para realizar la predicción.
- **Manipulación de los datos de entrenamiento:** en esta forma de ensamblado, se realizan modificaciones en los datos de entrenamiento con el fin de generar varios modelos diferentes, el algoritmo se ejecuta varias veces con distintos datos de entrenamiento y validación. Según Dietterich esta técnica funciona especialmente bien para algoritmos de aprendizaje inestables, algoritmos cuyo clasificador de salida sufre cambios importantes en respuesta a pequeños cambios en los datos de entrenamiento. Por tanto, recomienda que esta técnica puede ser usada para árboles de decisión, redes neuronales y algoritmos de aprendizaje por reglas, los cuales considera inestable. Entre las técnicas que se encuentran en esta definición se encuentran:
  - *Bootstrap aggregation.*
  - *Cross validation committees.*
  - *Ada boost.*
- **Manipulación de las características de entrada:** en este caso, se lleva a cabo una manipulación de las características de entrada, lo que implica la segmentación de las características de los datos de entrada para generar modelos basados en estas segmentaciones. Sin embargo, en posteriores experimentos se detectó que eliminar alguna característica de los datos de entrada impacta negativamente en el modelo generado, por tanto, esta técnica debería aplicarse si contamos con datos altamente redundante.
- **Manipulación de los objetivos de salida:** En este método se crean nuevas categorías de clasificación, por ejemplo, si tenemos un clasificador cuya salida es A o B, se toma cada categoría y se realiza una partición afín a la salida general, es decir si tenemos A, se pueden crear nuevas categorías afines a esta como  $A_1, A_2, \dots, A_n$  y lo mismo para B:  $B_1, B_2, \dots, B_n$ , a partir de estas nuevas clasificaciones se realiza el entrenamiento y cuando se realiza una nueva predicción se toma en cuenta la nueva clasificación que predijo el modelo y se la asigna un voto, y luego se compara las votaciones entre las subcategorías A y B y el que cuente la mayor cantidad de votos será la predicción ganadora.
- **Inyección de aleatoriedad:** este método se basa en los pesos iniciales de una red neuronal, según trabajos citados por Dietterich (2000) el hecho de cambiar aleatoriamente los pesos iniciales de una red neuronal puede dar un resultado distinto.



## 2.2.10. Lesiones de la piel.

### Nevus melanocítico.

Un nevus melanocítico es un término médico utilizado para describir una lesión pigmentada en la piel. Los nevos melanocíticos también se conocen como lunares y pueden aparecer en cualquier parte del cuerpo. A menudo son de color marrón oscuro o negro. La mayoría de los nevos melanocíticos son benignos y no requieren tratamiento. En la Figura 6 se observa un conjunto de imágenes relacionadas a esta lesión.



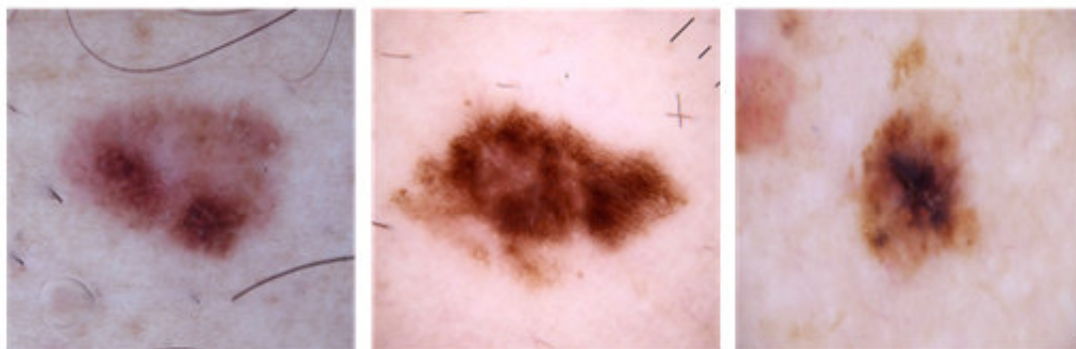
**Figura 6**

*Nevus melanocítico.*

*Fuente.* Imágenes tomadas del trabajo de Tschandl (2018).

### Melanoma.

El melanoma es una forma de cáncer cutáneo que inicia en las células responsables de la producción de melanina. Debido a su habilidad para extenderse velozmente a otras áreas del cuerpo, se cataloga como un tipo de cáncer de piel de naturaleza agresiva. En la Figura 7 se observa un conjunto de imágenes relacionadas a esta lesión.



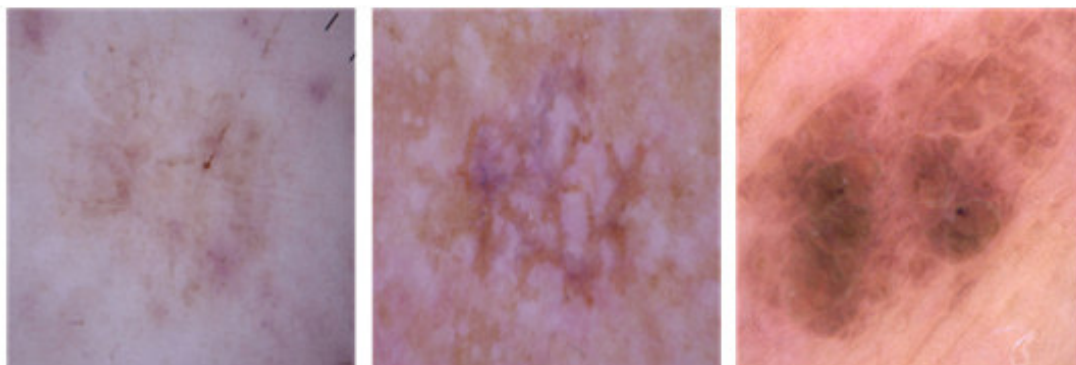
**Figura 7**

*Melanoma.*

*Fuente.* Imágenes tomadas del trabajo de Tschandl (2018).

### **Queratosis seborreica.**

La queratosis seborreica es una condición cutánea benigna caracterizada por la aparición de manchas cerosas de color marrón, negro o moreno. Es una de las neoplasias no cancerosas más frecuentes en los adultos mayores. En la Figura 8 se presenta un conjunto de imágenes relacionadas a esta lesión.



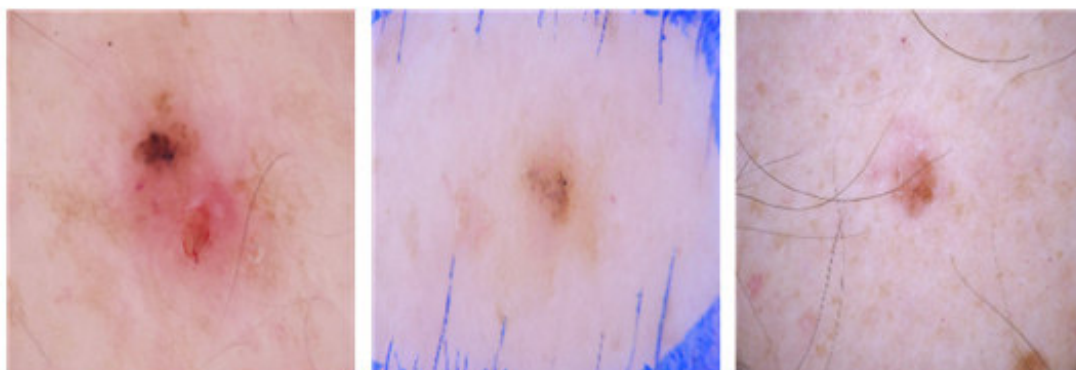
### **Figura 8**

*Queratosis seborreica.*

*Fuente.* Imágenes tomadas del trabajo de Tschandl (2018).

### **Basalioma.**

Se refiere a un tipo de cáncer cutáneo que se origina en las células basales de la piel. En la Figura 9 se observa un conjunto de imágenes relacionadas a esta lesión.



### **Figura 9**

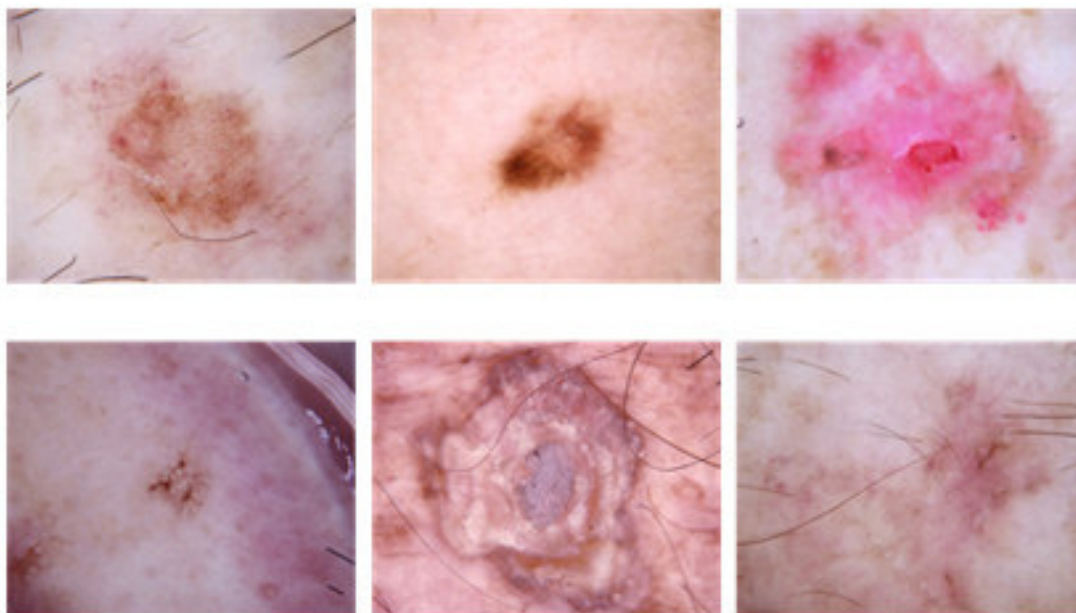
*Basalioma.*

*Fuente.* Imágenes tomadas del trabajo de Tschandl (2018).

### **Queratosis actínica.**

La queratosis actínica es una área de la piel áspera y escamosa que se desarrolla debido a la exposición solar prolongada a lo largo de los años que suele afectar a adultos

mayores. Debido a la posibilidad de que se vuelva cancerosa, generalmente se elimina por precaución. En la Figura 10 se presentan algunas lesiones de este tipo.



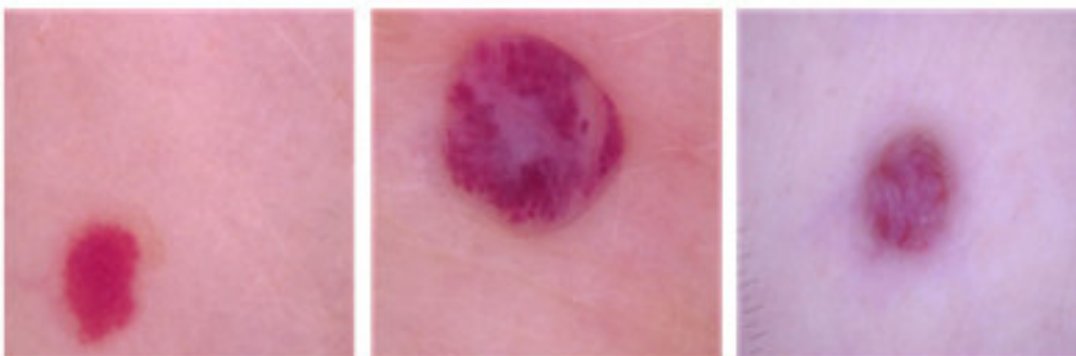
**Figura 10**

*Queratosis actínica.*

*Fuente.* Imágenes tomadas del trabajo de Tschandl (2018).

#### **Lesiones vasculares cutáneas.**

Las lesiones vasculares son irregularidades frecuentes que afectan la piel y los tejidos subyacentes, siendo más conocidas como marcas de nacimiento. En la Figura 11 se observa un conjunto de imágenes relacionadas a esta lesión.



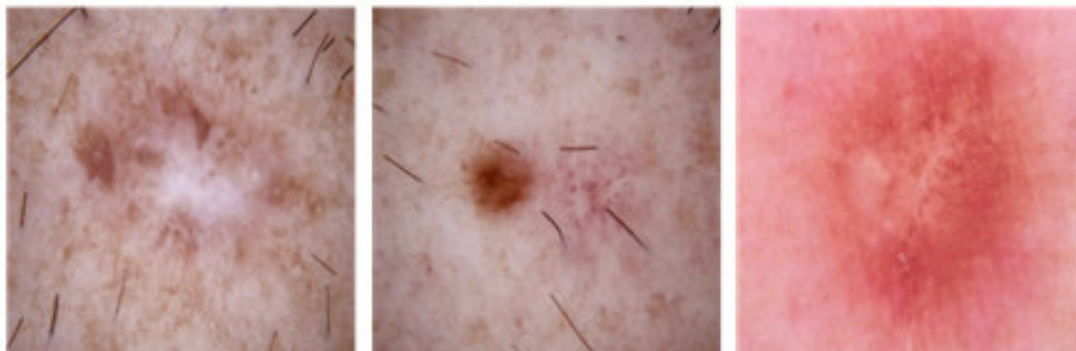
**Figura 11**

*Lesiones vasculares cutáneas.*

*Fuente.* Imágenes tomadas del trabajo de Tschandl (2018).

## **Dermatofibroma.**

Los dermatofibromas son tumores no cancerosos que aparecen con mucha frecuencia en la piel. Suelen observarse más comúnmente en mujeres de entre 30 y 50 años, y tienden a ubicarse principalmente en los brazos y extremidades inferiores. En la Figura 12 se observa un conjunto de imágenes relacionadas a esta lesión.



**Figura 12**

*Dermatofibroma.*

*Fuente.* Imágenes tomadas del trabajo de Tschandl (2018).

## **2.3. Marco conceptual.**

### **2.3.1. *Data augmentation.***

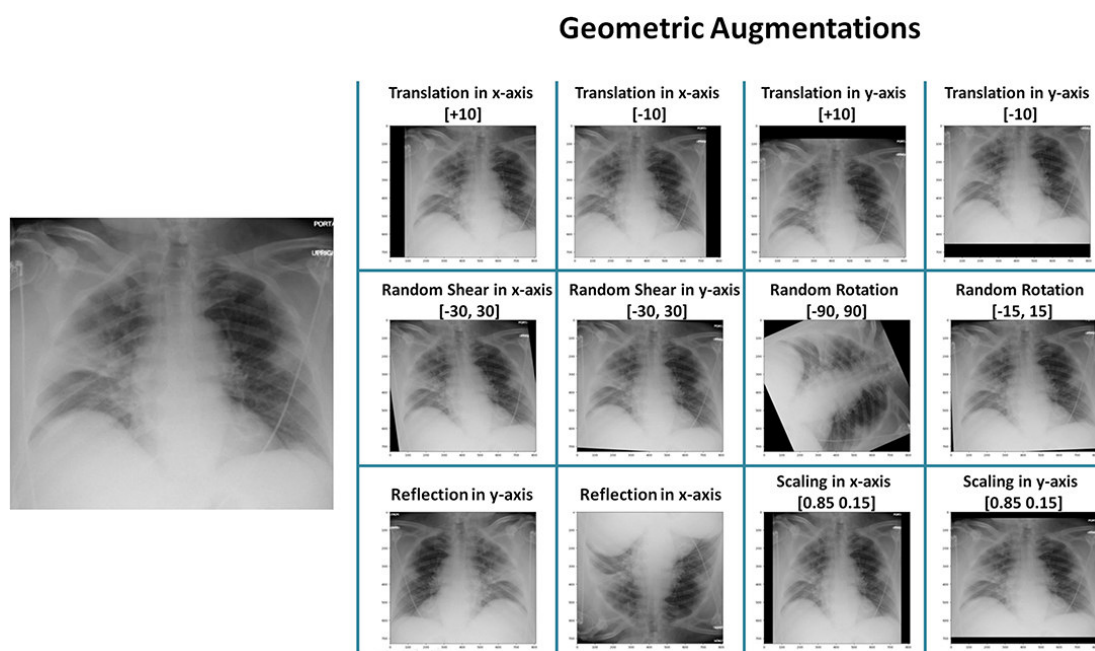
*Data augmentation* es una técnica muy usada en el aprendizaje automático y la visión por computadora, que permite mejorar la capacidad de los modelos para generalizar y mejorar su rendimiento en situaciones del mundo real. Para comprender esta técnica en detalle, es importante entender algunos conceptos clave.

En primer lugar, es fundamental comprender que los modelos de aprendizaje automático se entrenan en el *dataset* de entrenamiento, que está compuesto por una serie de ejemplos de datos de entrada y sus correspondientes etiquetas. El objetivo del modelo es aprender una función que permita predecir las etiquetas correctas para los datos de entrada no vistos durante el entrenamiento. En este proceso, el modelo puede aprender patrones específicos del conjunto de datos de entrenamiento, lo que puede dar lugar a un problema conocido como sobreajuste.

Sin embargo, en muchos casos, el conjunto de datos de entrenamiento puede ser limitado o no lo suficientemente representativo de la realidad. En estos casos, la *data augmentation* puede ser una técnica muy útil para aumentar el tamaño del conjunto de datos de entrenamiento y mejorar la capacidad del modelo para generalizar a situaciones del mundo real.

El aumento de datos consiste en aplicar transformaciones aleatorias a los datos de entrada originales, de tal forma que se generan nuevas muestras que son similares a las originales, pero no idénticas. Por ejemplo, en el caso de imágenes, algunas de las transformaciones más comunes son la rotación, la traslación, el escalado, el recorte, la modificación del brillo y el contraste, entre otras. Al aplicar estas transformaciones, se generan nuevas imágenes que contienen variaciones en la posición, el tamaño y la iluminación, lo que permite que el modelo aprenda patrones más robustos y genere predicciones más precisas y generalizadas.

Es importante destacar que las transformaciones que se aplican en la *data augmentation* deben ser cuidadosamente seleccionadas, de tal forma que sean relevantes para el problema que se está resolviendo y no generen datos sintéticos que sean poco realistas o poco representativos de la realidad. A continuación en la Figura 13 se observa un caso práctico de *data augmentation* aplicada a una imagen de rayos X.



**Figura 13**

*Data augmentation aplicado en una imagen de rayos X.*

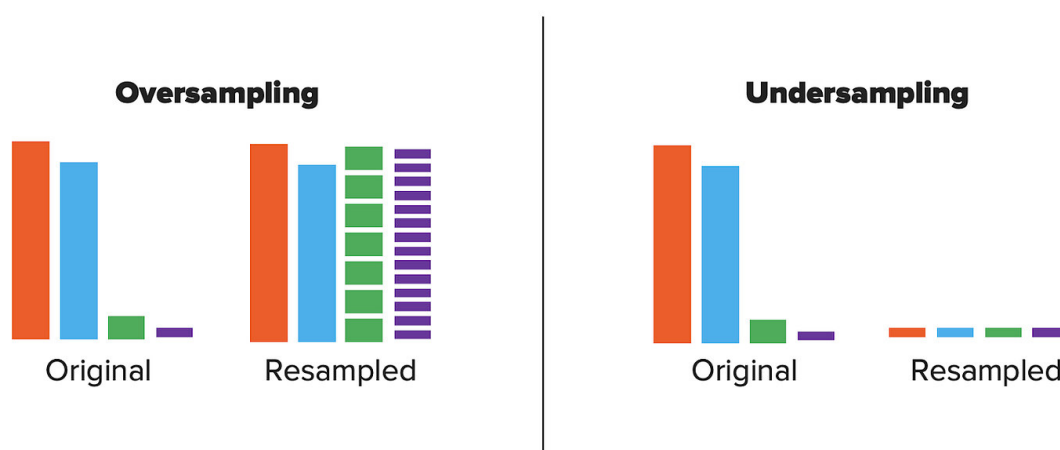
*Fuente.* Imágenes tomadas del trabajo de Shah (2022)

### 2.3.2. Balanceo de clases.

El balanceo o equilibrio de clases es una estrategia empleada en el campo del aprendizaje automático para hacer frente al desequilibrio de clases en un conjunto de datos. El desequilibrio de clases se presenta cuando una o varias clases tienen una proporción considerablemente menor que las demás en el conjunto de datos, lo que puede provocar que el modelo tienda a predecir la clase mayoritaria y no logre capturar patrones en las clases minoritarias. Existen varias técnicas para balancear clases en un conjunto

de datos, y cada una tiene sus propias ventajas y desventajas. Una técnica común es el submuestreo o *undersampling*, que consiste en reducir la cantidad de ejemplos en la clase mayoritaria hasta que se equilibren las clases. Por ejemplo, en el conjunto de datos de diagnóstico médico mencionado anteriormente, se podría reducir la cantidad de ejemplos de la clase sana hasta que tenga la misma cantidad que la clase enferma. Sin embargo, esta técnica puede llevar a la pérdida de información relevante y a la reducción del tamaño del *dataset*, lo que podría afectar el rendimiento del modelo.

Otra técnica es la sobremuestreo u *oversampling*, que consiste en aumentar la cantidad de ejemplos en la clase minoritaria hasta que se equilibren las clases. Por ejemplo, en el conjunto de datos de diagnóstico médico, se podría aumentar la cantidad de ejemplos de la clase enferma mediante la creación de ejemplos sintéticos o mediante la inclusión de nuevos ejemplos de pacientes enfermos. Sin embargo, esta técnica puede llevar a la generación de ejemplos sintéticos poco realistas y a la inclusión de datos redundantes. Finalmente, existe una técnica más avanzada llamada SMOTE que crea ejemplos sintéticos de la clase minoritaria al interpolar entre ejemplos cercanos en el espacio de características, generando datos similares pero con pequeñas variaciones.



**Figura 14**

*Balanceo de datos aplicando oversampling y undersampling.*

*Fuente.* Imágenes tomadas del trabajo de Potyraj (2021)

### 2.3.3. Matriz de confusión.

En el ámbito de la inteligencia artificial y el aprendizaje supervisado, la matriz de confusión es una herramienta que proporciona una representación visual del rendimiento de un algoritmo de aprendizaje supervisado. Cada columna de la matriz muestra el número de predicciones realizadas para cada clase, mientras que cada fila representa las clases reales. De esta manera, la matriz de confusión permite observar los aciertos y errores cometidos por el modelo implementado. A partir de esta matriz se desprenden términos como “falso positivo” (también conocido como “false positive” o FP),

“falso negativo” (también conocido como “false negative” o FN), “verdadero positivo” (también conocido como “true positive”, VP o TP) y “verdadero negativo” (también conocido como “true negative”, VN o TN).

En la Figura 15 se observan estos términos además de la obtención de los indicadores a partir de estos valores.

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
		Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

**Figura 15**

*Matriz de confusión.*

*Fuente.* Imagen tomada de Science (1970)

### **Verdadero positivo.**

Indica el número de predicciones correctas positivas, significa que una clase positiva se identificó correctamente como positiva.

### **Verdadero negativo.**

Indica el número de predicciones correctas negativas, significa que una clase negativa se identificó correctamente como negativa.

### **Falso positivo.**

Indica el número de predicciones incorrectas positivas, significa que una clase negativa se identificó incorrectamente como positiva, también conocido como “error tipo I”.

### **Verdadero negativo.**

Indica el número de predicciones correctas negativas, significa que una clase negativa se identificó correctamente como negativa.

### **Falso negativo.**

Indica el número de predicciones incorrectas negativas, significa que una clase positiva se identificó incorrectamente como negativa, también conocido como “error tipo II”.

**Exactitud.**

Del inglés “Accuracy”, esta métrica indica la proximidad entre el resultado de una medición y el valor verdadero. En el contexto estadístico, la exactitud está vinculada al sesgo de una estimación. Se representa como la proporción de resultados correctos dividida por el número total de casos. La fórmula asociada es la siguiente:

$$Accuracy = \frac{TN + TP}{(TP + TN + FP + FN)}$$

Donde:

TP: true positive (verdadero positivo).

TN: true negative (verdadero negativo).

FP: false positive (falso positivo).

FN: false negative (falso negativo).

**Precisión.**

Del inglés “Precision”, Esta métrica indica la proporción de casos positivos detectados. La fórmula asociada es la siguiente.

$$Precision = \frac{TP}{(TP + FP)}$$

**Sensibilidad.**

Del inglés “Recall” o “Sensitivity”, esta métrica representa la proporción de casos positivos que fueron correctamente identificados por el modelo. La fórmula asociada es la siguiente:

$$Recall = \frac{TP}{(TP + FN)}$$

**Especificidad.**

Del inglés “Specificity”, esta métrica indica los casos negativos que el modelo ha clasificado correctamente. La fórmula asociada es la siguiente:

$$Recall = \frac{TN}{(TN + FP)}$$

**F1-score.**

Se trata de una métrica que combina la precisión y la sensibilidad en una sola medida de manera armónica. La fórmula asociada es la siguiente:

$$F1 - score = 2 * \frac{Recall * Precision}{(Recall + Precision)}$$



**ROC AUC score.**

El ROC AUC score indica la relación entre la tasa de detección correcta de casos positivos (TPR) y la tasa de identificación errónea de casos negativos como positivos (FPR) del modelo. La fórmula asociada es la siguiente:

$$AUC = \frac{\sum Rangos(CP) - CP * \frac{CP+1}{2}}{CP + CN}$$

Donde:

$\sum Rangos(CP)$ : es la suma de los rangos de todos los ejemplos clasificados positivamente.

CP: Número de ejemplos positivos en el *dataset*.

CN: Número de ejemplos negativos en el *dataset*.

## 2.4. Glosario.

- **nv**: nevus melanocítico.
- **ml**: melanoma.
- **bkl**: queratosis seborreica.
- **bcc**: basalioma.
- **akiec**: queratosis actínica.
- **vasc**: lesiones vasculares cutáneas.
- **df**: dermatofibroma.
- **CNN**: red neuronal convolucional.
- **DCNN**: red neuronal convolucional profunda.
- **G.A. o A.G.**: *genetic algorithm* o algoritmo genético.
- **B.S.**: *binary search* o búsqueda binaria.
- **HAM10000**: *dataset* público que contiene 10,015 imágenes de lesiones dermatoscópicas, propuesta por Tschandl (2018).
- **CIFAR-10 y CIFAR-100**: *datasets* públicos usados para validación de modelos de visión computacional, 10 y 100 significan el número de clases que contienen.

# Capítulo III

## Metodología

### 3.1. Tipos de investigación.

#### 3.1.1. Nivel de investigación.

Se considera que esta investigación se ubica en un nivel de investigación aplicada, ya que se busca aplicar técnicas de inteligencia artificial (específicamente DCNN) para solucionar un problema práctico y concreto, que es la clasificación de lesiones en la piel.

#### 3.1.2. Tipo de la investigación.

Se considera el enfoque cuantitativo como el adecuado para esta investigación, se aplica la recopilación y análisis de información cuantitativa y estadística (Creswell y Creswell, 2017). En este caso, se utilizan técnicas cuantitativas para medir el desempeño en la clasificación del diagnóstico en 5 dimensiones: exactitud, precisión, sensibilidad, F1-score y ROC AUC score.

#### 3.1.3. Diseño de la investigación.

El diseño aplicado será el experimental, se aplican pruebas de hipótesis comparando un antes y después de la aplicación de las técnicas de optimización (Creswell y Creswell, 2017). Además, debido a la naturaleza del repositorio de datos sobre el que se trabaja (HAM10000), es menester manipular de forma intencional el conjunto de datos inicial, por ende el subtipo de diseño experimental será el diseño experimental puro.

#### 3.1.4. Método de la investigación.

El método de la investigación será del tipo deductivo, esto se justifica en que se utilizarán métodos generales ya existentes y aceptados como las técnicas de inteligencia

artificial y optimización para probar las hipótesis por medio de la experimentación.

## **3.2. Dataset y partición.**

### **3.2.1. Dataset.**

El *dataset* o conjunto de datos que se aplicó en el estudio son las 10,015 imágenes dermatoscópicas contenidas en el repositorio HAM10000 de dimensiones originales de  $450 \times 600$  píxeles, propuesto por Tschandl (2018), este *dataset* fue recolectado a lo largo de 20 años del departamento de dermatología de la Universidad Médica de Viena y la clínica de dermatología oncológica del Dr. Cliff Rosendahl en Queensland.

### **3.2.2. Partición.**

Acorde a Gillis (2022), se recomienda en la construcción de modelos de machine learning que la muestra para realizar el entrenamiento sea el 70 % u 80 % del conjunto de datos original y el 30 % ó 20 % para la validación, en este caso vamos a aplicar la validación cruzada con 4 pliegues o *folds*, por tanto, tendremos un 75 % para pruebas y 25 % para validación por cada pliegue.

### **3.2.3. Técnica de muestreo de partición.**

Acorde a Gillis (2022), la selección de la muestra será realizada de forma aleatoria, debido a que estamos realizando una tarea de clasificación y los datos no se encuentran balanceado su selección será estratificada.

## **3.3. Operacionalización de variables.**

### **3.3.1. Identificación de variables.**

#### **Variable independiente**

**Nombre:** imagen dermatoscópica.

**Definición conceptual:** es una imagen generada por dermatoscopia, según Palacios-Martínez y Díaz-Alonso (2017), la dermatoscopia es una técnica de diagnóstico no invasiva realizada en tiempo real, diseñada para examinar las lesiones en la piel. Esta técnica mejora la precisión del diagnóstico de las lesiones con exceso de pigmentación y facilita la detección temprana de lesiones potencialmente malignas como el melanoma. Además, su implementación no implica un aumento significativo en el tiempo dedicado a la exploración física.

**Definición de operacionalidad:** es una imagen etiquetada con un tipo de lesión.

**Dimensión:** tipo de lesión.

### **Variable dependiente**

**Nombre:** desempeño de clasificación de diagnóstico.

**Definición conceptual:** se refiere al rendimiento al realizar la tarea de clasificación del diagnóstico. Según el Instituto Nacional del Cáncer (2011), el diagnóstico refiere al proceso que identifica una afección por medio de sus síntomas o lesiones.

**Definición de operacionalidad:** se refiere al proceso de cuantificar el desempeño de clasificación del diagnóstico.

**Dimensiones:** las dimensiones evaluadas serán las siguientes:

- Exactitud.
- Precisión.
- Sensibilidad.
- F1-score.
- ROC AUC score.

### **3.3.2. Matriz de operacionalización de variables.**

En la Tabla 3 se presenta la matriz de operacionalización de variables.

**Tabla 3***Matriz de operacionalización de variables.*

<b>Variable</b>	<b>Tipo de Variable</b>	<b>Definición conceptual</b>	<b>Definición de operacionalidad</b>	<b>Dimensiones</b>	<b>Indicadores</b>
Imagen dermatoscópica.	Independiente.	Es una imagen generada por dermatoscopia, según Palacios-Martínez y Díaz-Alonso (2017), la dermatoscopia es una técnica de diagnóstico no invasiva realizada en tiempo real, diseñada para examinar las lesiones en la piel. Esta técnica mejora la precisión del diagnóstico de las lesiones con exceso de pigmentación y facilita la detección temprana de lesiones potencialmente malignas, como el melanoma. Además, su implementación no implica un aumento significativo en el tiempo dedicado a la exploración física..	Es una imagen etiquetada con un tipo de lesión.	Tipo de lesión.	El repositorio HAM1000 contiene 7 tipos de lesiones contenidas en imágenes dermatoscópicas, entre las cuales se encuentra: nevus melanocítico, melanoma, queratosis seborreica, basalioma, queratosis actínica, lesiones vasculares cutáneas y dermatofibroma.

Desempeño de clasificación de diagnóstico.	Dependiente.	Se refiere al rendimiento al realizar la tarea de clasificación del diagnóstico. Según el Instituto Nacional del Cáncer (2011), el diagnóstico refiere al proceso que identifica una afección por medio de sus síntomas o lesiones.	Se refiere al proceso de cuantificar el desempeño de clasificación del diagnóstico.	Exactitud.	Del inglés “Accuracy”, esta métrica indica la proximidad entre el resultado de una medición y el valor verdadero. En el contexto estadístico, la exactitud está vinculada al sesgo de una estimación. Se representa como la proporción de resultados correctos dividida por el número total de casos.
				Precisión.	Del inglés “Precision”, Esta métrica indica la proporción de casos positivos detectados.
				Sensibilidad.	Del inglés “Recall” o “Sensitivity”, esta métrica representa la proporción de casos positivos que fueron correctamente identificados por el modelo.
				F1-score.	Se trata de una métrica que combina la precisión y la sensibilidad en una sola medida de manera armónica.

---

ROC AUC score. El ROC AUC score indica la relación entre la tasa de detección correcta de casos positivos (TPR) y la tasa de identificación errónea de casos negativos como positivos (FPR) del modelo.

---

*Fuente.* Elaboración propia.

### 3.3.3. Descripción de indicadores de variables.

**Dimensión:** tipo de lesión.

**Indicadores:** los indicadores asociados a esta dimensión está asociado con el repositorio HAM1000 contiene 7 tipos de lesiones contenidas en imágenes dermatoscópicas, entre los cuales se encuentra:

- Nevus melanocítico.
- Melanoma.
- Queratosis seborreica.
- Basalioma.
- Queratosis actínica.
- Lesiones vasculares cutáneas.
- Dermatofibroma.

La definición de cada tipo de lesión se encuentra en el punto 2.2 (Bases Teóricas), sección **Lesiones de la piel**.

**Dimensión:** exactitud.

**Indicadores:** su definición está descrita en el punto 2.3 (Marco Conceptual), sección **Exactitud**.

**Dimensión:** precisión.

**Indicadores:** su definición está descrita en el punto 2.3 (Marco Conceptual), sección **Precisión**.

**Dimensión:** sensibilidad.

**Indicadores:** su definición está descrita en el punto 2.3 (Marco Conceptual), sección **Sensibilidad**.

**Dimensión:** F1-score.

**Indicadores:** su definición está descrita en el punto 2.3 (Marco Conceptual), sección **F1-score**.

**Dimensión:** ROC AUC score.

**Indicadores:** su definición está descrita en el punto 2.3 (Marco Conceptual), sección **ROC AUC score**.



### **3.4. Técnicas e instrumentos de recolección de datos.**

#### **3.4.1. Técnicas.**

Se decidió descargar uno de los repositorios de datos públicos utilizado por la ISIC, llamado HAM10000 y presentado por Tschandl (2018).

#### **3.4.2. Instrumentos.**

Al usar un *dataset* público presentado por otro autor, vamos a citar los instrumentos utilizados por Tschandl (2018) para la elaboración del repositorio HAM10000.

- Extracción de imágenes y metadata de archivos powerpoint.
- Digitalización de diapositivas.
- Extracción de datos de un sistema de dermatoscopia digital.
- Filtrado de imágenes dermatoscópicas.
- Unificación de diagnósticos patológicos.
- Revisión de calidad manual.

#### **3.4.3. Validación y confiabilidad del instrumento.**

En el caso del repositorio público HAM10000 presentado por Tschandl (2018), la validación se realizó por medio de un juicio de expertos en los siguientes puntos:

- Validación histopatología.
- Validación confocal de la imagen.
- Seguimiento.
- Consenso.

#### **3.4.4. Procedimientos**

Para validar la hipótesis de investigación se realizarán los siguientes procedimientos.

1. Análisis del repositorio HAM10000 (Figura 16).
  - Análisis de imágenes.
  - Redimensionamiento a  $224 \times 224$  pixeles.
  - Generación de 4 folds de data sin aumentar.

- Aplicación de *data augmentation* y aplicación de pesos.
- Generación de 4 folds con la data aumentada del item anterior.

## 2. Definición de Modelo DCNN.

- Creación de modelo DCNN.
- Selección de modelos preentrenados: InceptionV3, ResNet152V2, DenseNet121 y EfficientNetB6.

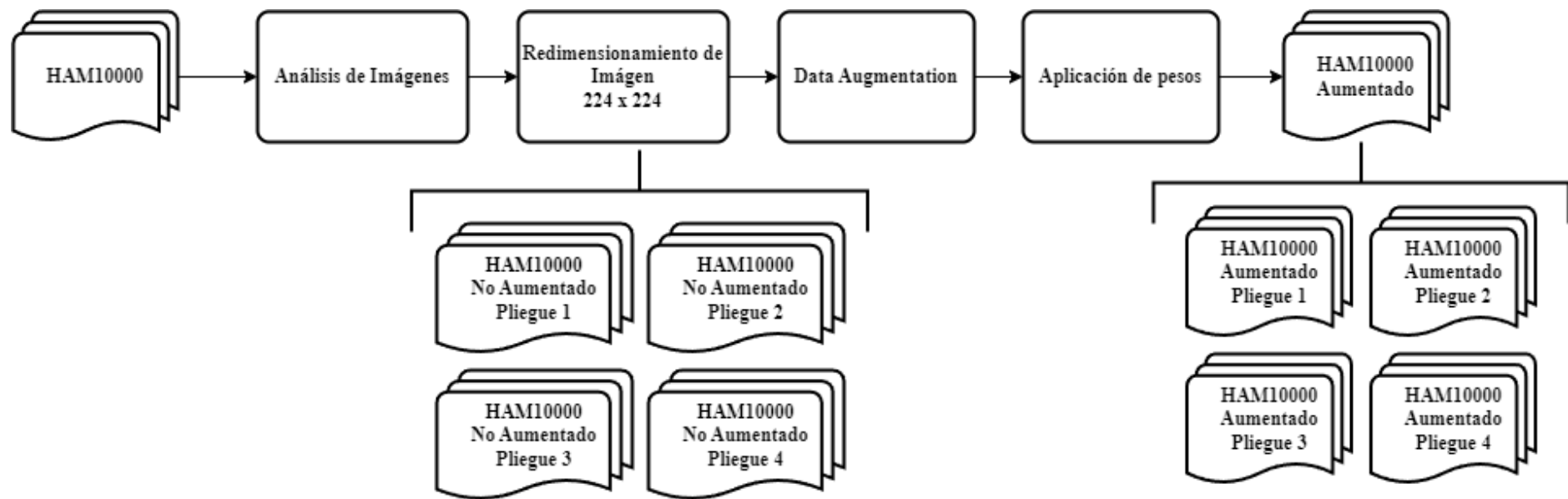
## 3. Experimentos de validación por cada hipótesis específica.

- Hipótesis específica 1 (Figura 17): Se tomará los *datasets* generados con la data por defecto y la aumentada con aplicación de pesos. Se realizarán los experimentos por cada modelo preentrenado.
- Hipótesis específica 2 (Figura 18): Se tomará el *dataset* con la data aumentada y se realizarán los experimentos por cada modelo sin *transfer learning* y con *transfer learning*.
- Hipótesis específica 3 (Figura 19): Se tomará el *dataset* con la data aumentada y se realizarán los experimentos por cada modelo con *transfer learning* aplicando congelamiento por medio de algoritmos genéticos y búsqueda binaria modificada y comparándolo al no aplicar congelamiento.
- Hipótesis específica 4 (Figura 20): Se tomará el *dataset* con la data aumentada y se realizarán los experimentos por cada modelo aplicando métodos de ensamblaje y se realizará la comparación con el modelo que presente mejor resultado de la hipótesis específica 3.

## 4. Validación por hipótesis específicas.

- Hipótesis específica 1: se tomarán los indicadores propuestos en la metodología y se realizarán las pruebas de hipótesis con un valor p de 0.05.
- Hipótesis específica 2: se tomarán los indicadores propuestos en la metodología y se realizarán las pruebas de hipótesis con un valor p de 0.05.
- Hipótesis específica 3: se tomarán los indicadores propuestos en la metodología y se realizarán las pruebas de hipótesis con un valor p de 0.05.
- Hipótesis específica 4: se tomarán los indicadores propuestos en la metodología y se realizarán las pruebas de hipótesis con un valor p de 0.05.

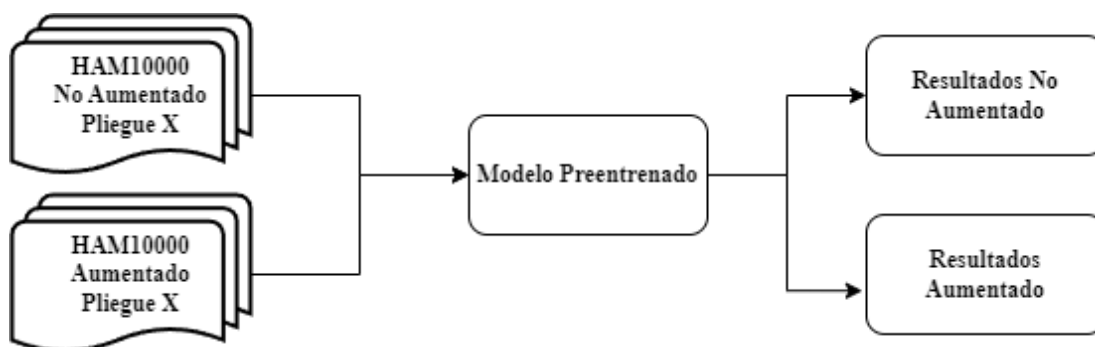
## 5. Finalmente, una vez validadas estas 4 hipótesis específicas se puede dar por comprobada la hipótesis general.



**Figura 16**

*Procedimiento de análisis de imágenes de repositorio HAM10000.*

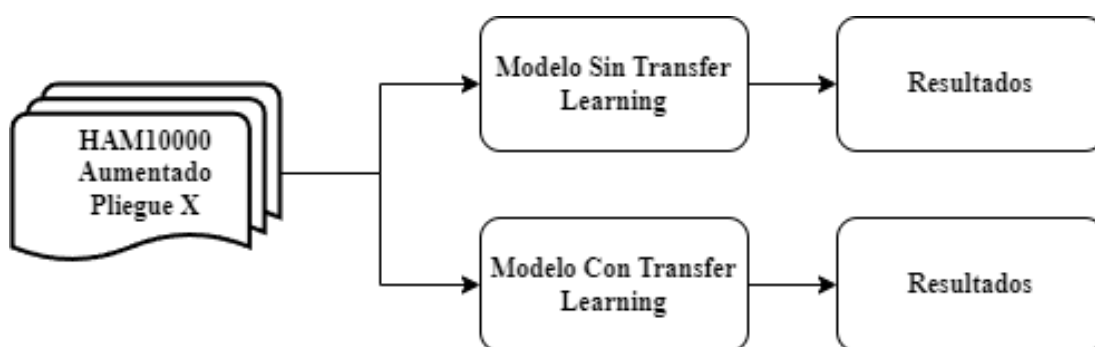
*Fuente. Elaboración Propia.*



**Figura 17**

*Procedimiento de la hipótesis específica 1.*

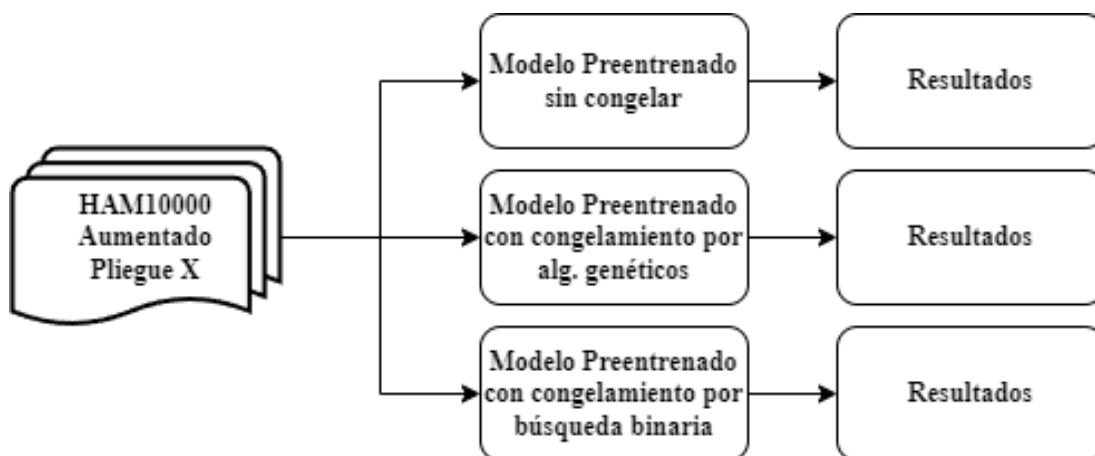
*Fuente. Elaboración Propia.*



**Figura 18**

*Procedimiento de la hipótesis específica 2.*

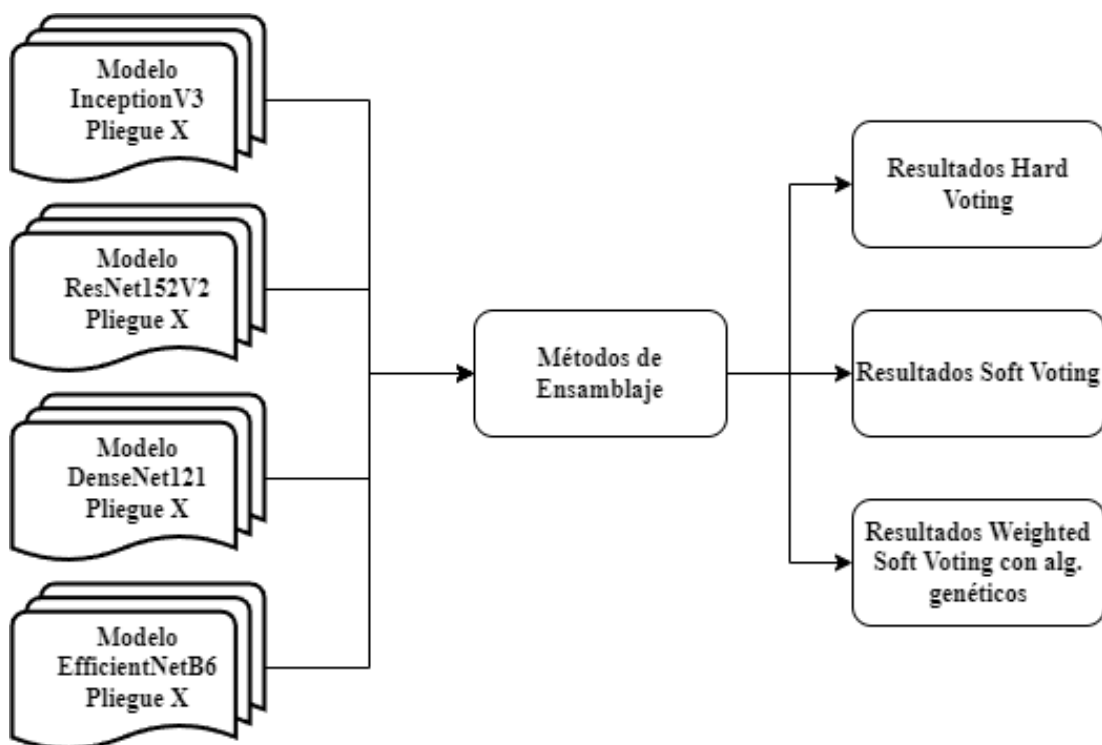
*Fuente. Elaboración Propia.*



**Figura 19**

*Procedimiento de la hipótesis específica 3.*

*Fuente. Elaboración Propia.*



**Figura 20**

*Procedimiento de la hipótesis específica 4.*

*Fuente. Elaboración Propia.*

# Capítulo IV

## Fase Experimental

Todos los recursos como los *datasets* y modelos generados (con sus resultados) junto a su código fuente para su uso libre se encuentran disponibles en la Tabla 63 en los anexos.

### 4.1. Análisis del *dataset* HAM10000.

Previo a la fase experimental, se realizó un análisis de las imágenes provistas por el HAM10000, de forma que podamos tener un resultado más fiable al momento de realizar los experimentos, pruebas y presentación de los resultados al ser las imágenes la variable independiente en la matriz de operacionalización de variables (Tabla 3).

#### 4.1.1. Ventajas del uso del *dataset* HAM10000.

1. Conjunto de datos grande y etiquetado: el repositorio HAM10000 es un conjunto de datos grande que contiene 10,015 imágenes de lesiones de piel etiquetadas por expertos en dermatología. Este conjunto de datos es uno de los más grandes y completos disponibles para el estudio de las lesiones cutáneas y puede utilizarse para el entrenamiento y evaluación de algoritmos de aprendizaje automático y análisis de imágenes.
2. Variedad de categorías de lesiones de piel: el conjunto de datos HAM10000 incluye siete categorías diferentes de lesiones de piel, lo que permite el estudio de diferentes tipos de lesiones y patologías de la piel. Esto hace que el conjunto de datos sea más representativo de la diversidad de lesiones de piel que se encuentran en la práctica clínica diaria.
3. Imágenes de alta calidad: todas las imágenes del *dataset* HAM10000 son imágenes de alta resolución y calidad, lo que permite una evaluación detallada de las características de las lesiones de piel.

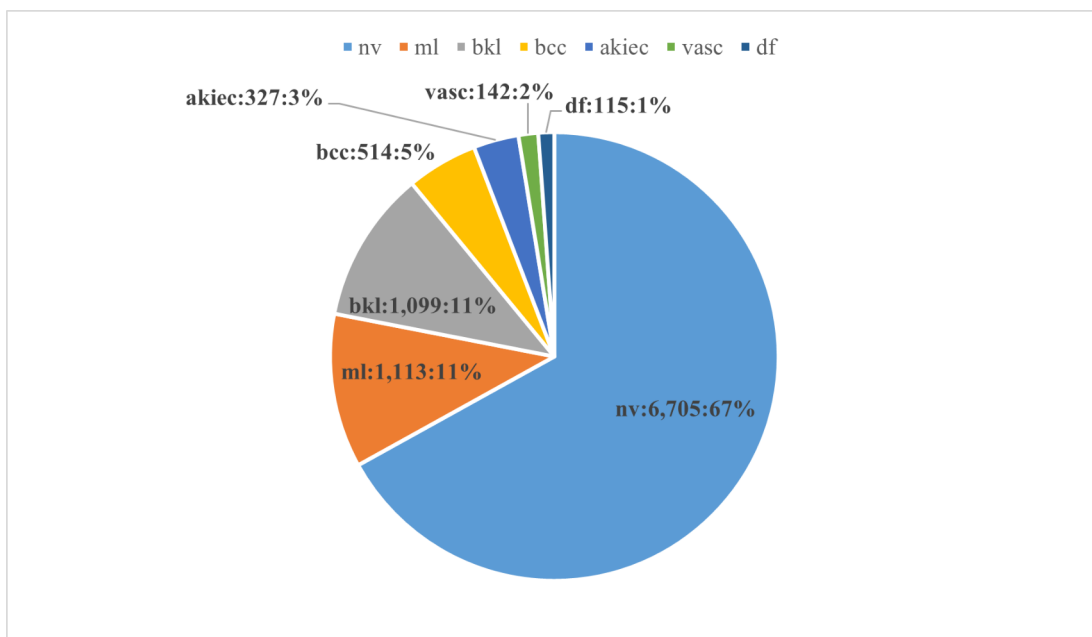
4. Disponibilidad gratuita: el *dataset* HAM10000 está disponible de forma pública y gratuita para su descarga y es utilizada por investigadores y profesionales de la salud interesados en su análisis y aplicación en el análisis de imágenes de lesiones cutáneas.
5. Utilidad en la práctica clínica: los algoritmos de análisis de imágenes entrenados con el *dataset* HAM10000 pueden ser útiles para el diagnóstico y la clasificación de las lesiones cutáneas, lo que podría mejorar la precisión y eficiencia del proceso de diagnóstico en la práctica clínica diaria.

#### **4.1.2. Desventajas del uso del *dataset* HAM10000.**

1. Tamaño del conjunto de datos: aunque el conjunto de datos contiene 10,015 imágenes etiquetadas, esto puede ser insuficiente para algunos tipos de análisis y aprendizaje automático. Además, el conjunto de datos solo incluye imágenes de lesiones cutáneas de ciertas categorías, lo que limita su utilidad para el estudio de otras enfermedades de la piel.
2. Sesgo de selección: las imágenes en el *dataset* se recopilaron en una clínica en Austria y Queensland, lo que puede limitar su representatividad de la diversidad étnica y geográfica de las lesiones de piel. Además, las imágenes se recopilaron de pacientes que recibieron tratamiento en una clínica de dermatología, lo que puede sesgar la selección de las lesiones de piel incluidas en el *dataset*.
3. Falta de información adicional: el *dataset* HAM10000 no incluye información adicional sobre los pacientes o las lesiones de piel, lo que podría limitar su uso en la identificación de factores de riesgo o características clínicas asociadas con las diferentes categorías de lesiones cutáneas.
4. Etiquetado subjetivo: aunque las imágenes en el conjunto de datos están etiquetadas por expertos en dermatología, el etiquetado sigue siendo subjetivo y puede haber cierta variabilidad entre los diferentes etiquetadores en la identificación y clasificación de las lesiones de piel.

#### **4.1.3. Aplicación de *data augmentation* y aplicación de pesos en el *dataset* HAM10000.**

Tal como se observa en la Figura 21 existe un evidente desbalance de datos, debido a que el 67 % de estas pertenecen a las lesiones del tipo Nevus melanocítica (nv), para tanto es necesario aumentar de forma artificial la cantidad de imágenes para tratar de balancearlas.



**Figura 21**

*HAM10000: disgregación de imágenes por tipo de Lesión.*

*Fuente. Elaboración Propia.*

Por tanto, se procede a tomar todas las imágenes que no sean del tipo nv y crear imágenes nuevas a partir de las existentes. Las nuevas imágenes serán creadas a partir de esta lista:

- Rotación de imagen en 90° en sentido horario.
- Rotación de imagen en 90° en sentido antihorario.
- Rotación de imagen en 180°.
- Reflexión de la imagen en el eje X.
- Reflexión de la imagen en el eje Y.

A continuación, en la Tabla 4 se observa la nueva cantidad de imágenes generadas, a pesar de esto, el conjunto de datos sigue siendo desbalanceado, por tanto, se optó por asignar un peso por cada tipo de lesión de forma que cuando el modelo de clasificación sea ejecutado, se determine que un error en la clasificación de un tipo de lesión con menos imágenes será más "castigado" que un tipo de lesión con más imágenes. Para el cálculo del peso se aplicó la siguiente fórmula:

$$Peso_i = \frac{TotalImágenes}{NumeroClases \times ImágenesClase_i}$$

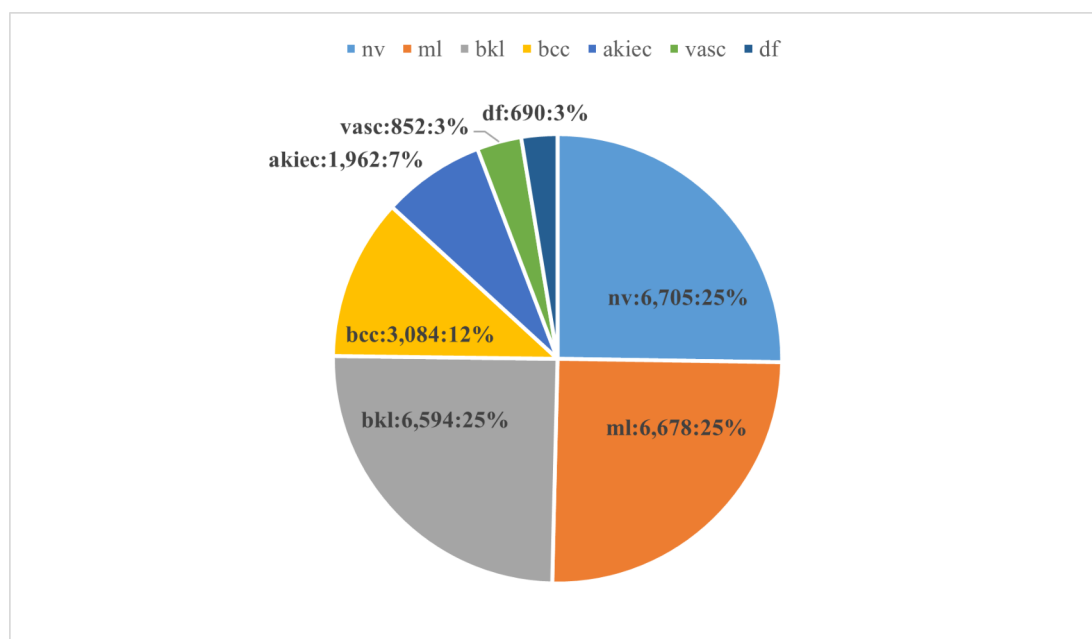


**Tabla 4**  
*HAM10000: imágenes por tipo de lesión*

Tipo de Lesión	Acrónimo	Cantidad Inicial	Cantidad Aumentada	Cantidad Total	Unidad	Peso
Nevus Melanocítico	nv	6,705	0	6,705	Imagen	0.566
Melanoma	ml	1,113	5,565	6,678	Imagen	0.568
Queratosis seborreica	bkl	1,099	5,495	6,594	Imagen	0.576
Basalioma	bcc	514	2,570	3,084	Imagen	1.231
Queratosis actínica	akiec	327	1,635	1,962	Imagen	1.934
Lesiones vasculares cutáneas	vasc	142	710	852	Imagen	4.454
Dermatofibroma	df	115	575	690	Imagen	5.500
<b>Total</b>		<b>10,015</b>	<b>16,550</b>	<b>26,565</b>	<b>Imagen</b>	

*Fuente.* Elaboración Propia.

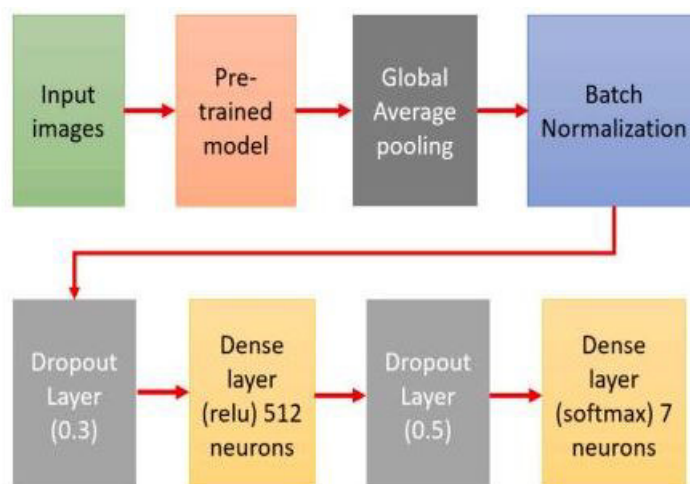
Finalmente, con el aumento de imágenes el porcentaje de Nevus Melanocítico se decrecienta de 67 % a 25 % y la distribución de tipos de lesiones se homogeniza de forma parcial como se observa en la Figura 22.



**Figura 22**  
*HAM10000: disgregación de imágenes por tipo de Lesión con aumento de imágenes.*  
*Fuente.* Elaboración Propia.

## 4.2. Modelo DCNN propuesto.

Para la realización de todos los experimentos se basó en el modelo DCNN propuesto por Rahman et al. (2021) (Figura 23).



**Figura 23**

*Modelo DCNN.*

*Fuente.* Tomado del trabajo de Rahman et al. (2021).

## 4.3. Arquitecturas DCNN.

Los modelos pre-entrenados utilizados son los presentados en la Tabla 5 indicando el número de capas, los pesos predefinidos por *transfer learning* y los autores en los que se basaron la implementación.

**Tabla 5**

*Arquitecturas DCNN utilizadas*

Arquitectura DCNN	Número de capas	Basado en	Pesos	Implementación
InceptionV3	311	Szegedy et al. (2015)	ImageNet	Python
ResNet152V2	564	He et al. (2016)	ImageNet	Python
DenseNet121	427	Huang et al. (2017)	ImageNet	Python
EfficientNetB6	666	Tan y Le (2019)	ImageNet	Python

*Fuente.* Elaboración Propia.

## 4.4. Implementación de modelos.

### 4.4.1. Búsqueda binaria modificada.

En la Figura 24 se indica la implementación de la búsqueda binaria de Herrera y Herrera (2022). Una de las ventajas es su ejecución en sesiones paralelas. Otra de las

ventajas es que el número de iteraciones máximo para encontrar esta capa al cual aplicar el congelamiento hacia atrás es  $\log_2(\text{capas}) + 1$ .

---

**Algorithm 1** Modified binary search for layer selection

---

```

procedure BINARY_SEARCH(model, layers, folds, delta, data)
  LB  $\leftarrow$  0
  UB  $\leftarrow$  layers ▷ DCNN layers embebbed in the model.
  IB  $\leftarrow$   $\lfloor \frac{LB+UB}{2} \rfloor$  ▷ First intermediate bound.
  LB_score  $\leftarrow$  train_and_evaluate(model, data, backward = LB, folds)
  IB_score  $\leftarrow$  train_and_evaluate(model, data, backward = IB, folds)
  UB_score  $\leftarrow$  train_and_evaluate(model, data, backward = UB, folds)
  iterations  $\leftarrow$   $\lfloor \log_2(\text{layers}) \rfloor$ 
  for iteration in iterations do
    min_score  $\leftarrow$  min(LB_Score, UB_Score)
    if IB_score + delta  $\geq$  min_score then
      if min_score = LB_Score then
        LB  $\leftarrow$  IB ▷ New lower bound.
        LB  $\leftarrow$  IB_score ▷ New lower score.
      else
        UB  $\leftarrow$  IB ▷ New upper bound.
        UB  $\leftarrow$  IB_score ▷ New upper score.
      end if
    else
      break ▷ Stopping iterations.
    end if
    IB  $\leftarrow$   $\lfloor \frac{LB+UB}{2} \rfloor$  ▷ New intermediate bound.
    if IB = LB or IB = UB then
      break ▷ Stopping iterations.
    end if
    IB_score  $\leftarrow$  train_and_evaluate(model, data, backward = IB, folds)
  end for
  best_layer  $\leftarrow$  0
  best_score  $\leftarrow$  max(LB_score, IB_score, UB_score)
  if best_score = LB_score then
    best_layer  $\leftarrow$  LB
  else if best_score = IB_score then
    best_layer  $\leftarrow$  IB
  else
    best_layer  $\leftarrow$  UB
  end if
  return best_layer
end procedure

```

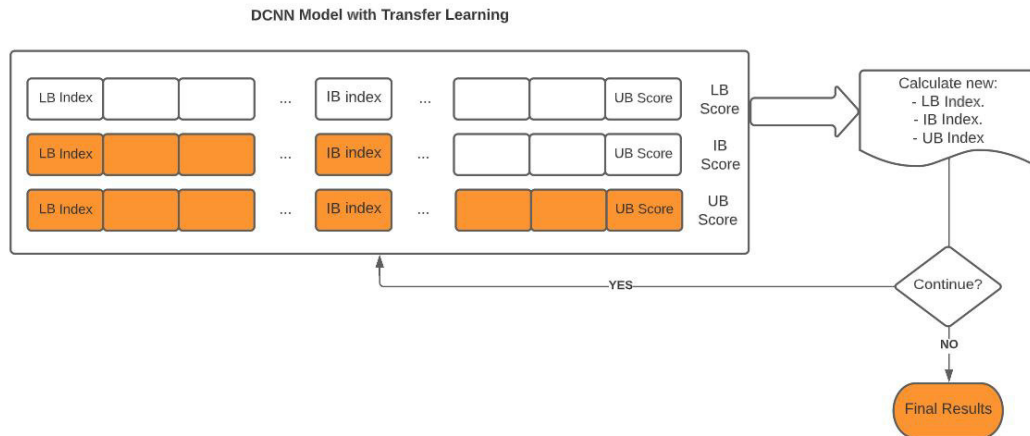
---

**Figura 24**

*Búsqueda binaria modificada.*

*Fuente.* Tomado del trabajo de Herrera y Herrera (2022).

En la Figura 25 se observa el diagrama de flujo para este algoritmo.



**Figura 25**

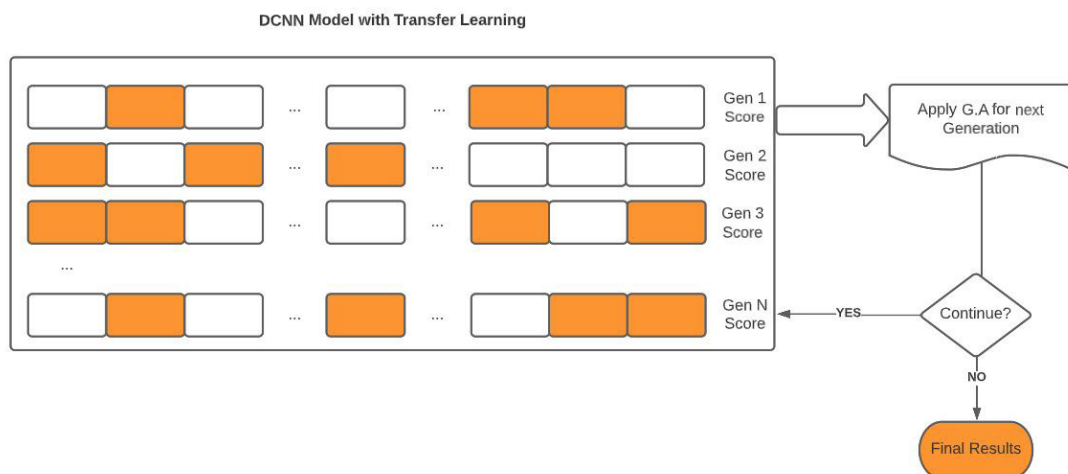
*Diagrama de flujo de congelación por búsqueda binaria.*

*Fuente.* Tomado de la presentación de Herrera y Herrera (2022).

#### 4.4.2. Algoritmos genéticos.

Para la implementación de algoritmos genéticos se utilizó la librería Python Genetic Algorithm Library (2020), que contiene la implementación del método "Elite" y su uso es muy sencillo ya que solo se define las entradas, su tipo de dato y la función objetivo para hallar el mejor valor.

Una de las limitaciones de este método es el alto costo computacional, por tanto, el número de población inicial e iteraciones se limitó de forma que sean soportados por Kaggle, contando con una población inicial de 4 genotipos con un máximo de 4 iteraciones. A continuación, como se observa en la Figura 26 presentamos un diagrama de flujo de este algoritmo.



**Figura 26**

*Diagrama de flujo de congelación por algoritmos genéticos.*

*Fuente.* Tomado de la presentación de Herrera y Herrera (2022).

### 4.4.3. Evaluación

En todos los modelos se ha utilizado la validación cruzada con 4 pliegues o *folders*, según la documentación relacionada para pruebas de hipótesis desarrollada por Kukar y Kononenko (2002) se sugiere 10 pliegues, pero se ha realizado en 4 pliegues teniendo en cuenta el alto costo computacional para la ejecución de redes neuronales convolucionales y que otros trabajos tales como Kawasaki et al. (2015); Wada et al. (2019) toman en cuenta el mismo número de pliegues.

En el caso del método de la búsqueda binaria modificada toma un delta arbitrario de 0.005, en el caso de los algoritmos genéticos, tal como se explicó en el punto anterior se detendrá al finalizar sus iteraciones. En todos los experimentos se usará el indicador *accuracy* ponderado para seleccionar el mejor resultado, los otros indicadores ya mencionados en la Tabla 3 se mostrarán en el siguiente capítulo.

Otro punto que se debe considerar es el número de épocas usadas para el modelo de DCNN son 10, pero para *efficientNetB6* se redujo a 5 debido a las limitaciones de hardware de Kaggle.

Finalmente, todos los resultados serán asociados con su promedio (Avg.), mediana (Mdn.), varianza (Var.) y desviación estándar (S.D.).

**Nota:** Cada pliegue o *fold* contiene 19,924 imágenes para entrenamiento y 6,641 imágenes para validación dividida en 7 clases de lesiones.

## 4.5. Hipótesis específica: *data augmentation* y aplicación de pesos.

La premisa de esta hipótesis específica es demostrar que el uso de las técnicas de *data augmentation* y aplicación de pesos se obtendrá un mejor resultado. Para esto se compara el mismo modelo DCNN variando sus datos de entrenamiento y validación, por tanto, se tienen los resultados con el conjunto de datos original (10,015 imágenes) y el aumentado (26,565 imágenes) sin aplicar congelamiento de capas en ambos conjuntos.

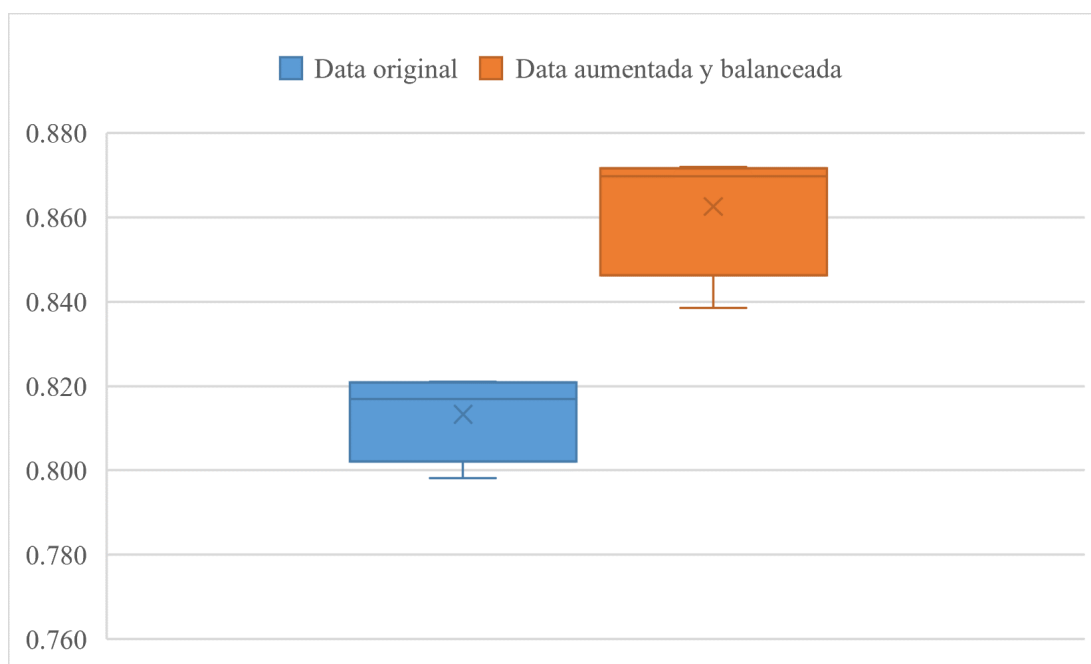
### 4.5.1. InceptionV3

Se observa en la Tabla 6 y la Figura 27 la diferencia notoria con respecto al uso del *data augmentation* y la aplicación de pesos, aumentando su *accuracy* en un 4.9 % en promedio.

**Tabla 6**  
Resultados experimentales para InceptionV3

	Data original	Data aumentada y balanceada
Fold 1	0.8199	0.8704
Fold 2	0.8211	0.8693
Fold 3	0.8139	0.8386
Fold 4	0.7982	0.8720
Avg.	0.81328	0.86257
Mdn.	0.81690	0.869835
Var.	0.00011	0.000257
S.D.	0.00033	0.000771

Fuente. Elaboración Propia.



**Figura 27**  
Comparativo de evaluación entre datasets para InceptionV3.  
Fuente. Elaboración Propia.

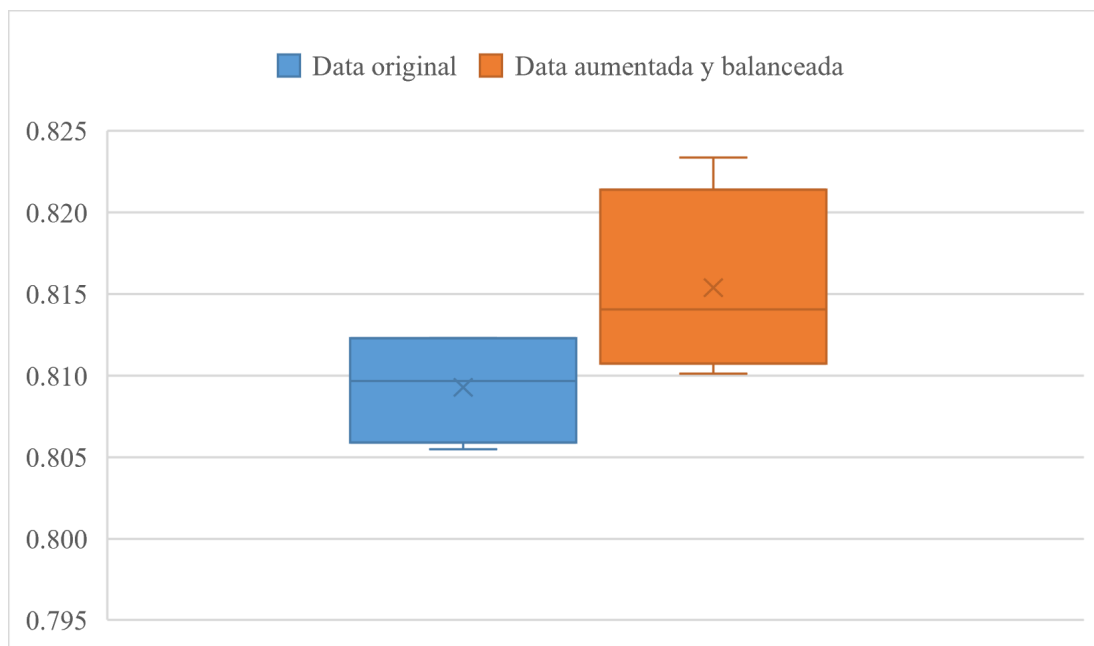
#### 4.5.2. ResNet152V2

Se observa en la Tabla 7 y la Figura 28 una ligera diferencia con respecto al uso del *data augmentation* y aplicación de pesos, aumentando su *accuracy* en un 0.6 % en promedio.

**Tabla 7**  
Resultados experimentales para ResNet152V2

	Data original	Data aumentada y balanceada
Fold 1	0.8055	0.8126
Fold 2	0.8071	0.8155
Fold 3	0.8123	0.8234
Fold 4	0.8122	0.8101
Avg.	0.80928	0.81540
Mdn.	0.80965	0.81405
Var.	0.00001	0.00003
S.D.	0.00004	0.00010

Fuente. Elaboración Propia.



**Figura 28**  
Comparativo de evaluación entre datasets para ResNet152V2.  
Fuente. Elaboración Propia.

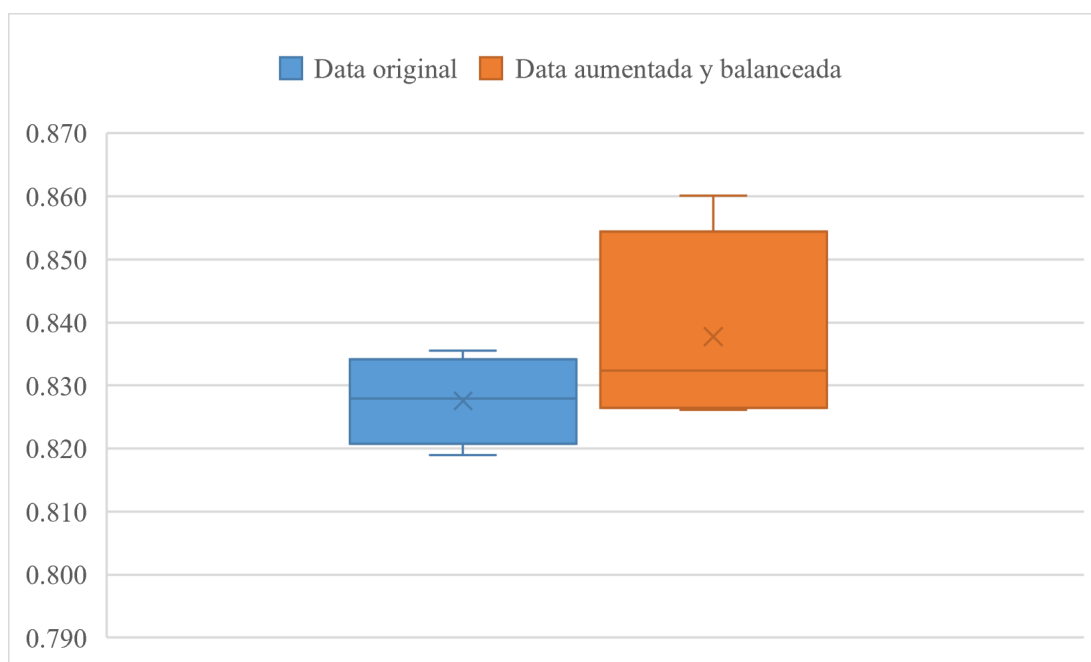
### 4.5.3. DenseNet121

Se observa en la Tabla 8 y la Figura 29 una ligera diferencia con respecto al uso del *data augmentation* y la aplicación de pesos, aumentando su *accuracy* en un 1% en promedio.

**Tabla 8**  
Resultados experimentales para DenseNet121

	Data original	Data aumentada y balanceada
Fold 1	0.8355	0.8261
Fold 2	0.8299	0.8276
Fold 3	0.8259	0.8601
Fold 4	0.8190	0.8372
Avg.	0.82758	0.83776
Mdn.	0.82790	0.83241
Var.	0.00005	0.00025
S.D.	0.00014	0.00074

Fuente. Elaboración Propia.



**Figura 29**  
Comparativo de evaluación entre datasets para DenseNet121.  
Fuente. Elaboración Propia.

#### 4.5.4. EfficientNetB6

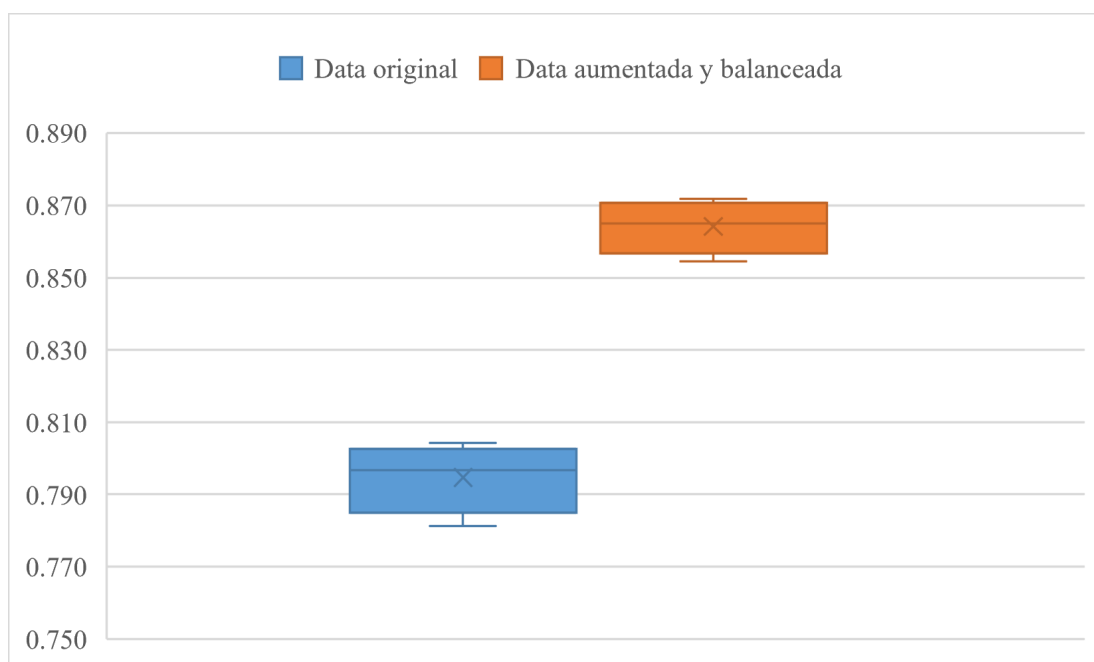
Se observa en la Tabla 9 y la Figura 30 la diferencia notoria con respecto al uso del *data augmentation* y la aplicación de pesos, aumentando su *accuracy* en un 7% en promedio.



**Tabla 9**  
Resultados experimentales para *EfficientNetB6*

	Data original	Data aumentada y balanceada
Fold 1	0.7812	0.8546
Fold 2	0.8043	0.8719
Fold 3	0.7975	0.8628
Fold 4	0.7958	0.8673
Avg.	0.79470	0.86415
Mdn.	0.79665	0.86505
Var.	0.00009	0.00005
S.D.	0.00028	0.00016

Fuente. Elaboración Propia.



**Figura 30**  
Comparativo de evaluación entre datasets para *EfficientNetB6*.  
Fuente. Elaboración Propia.

#### 4.6. Hipótesis específica: *transfer learning*.

La premisa de esta hipótesis específica es demostrar que el uso de *transfer learning* tiene un mejor resultado. Para esto se compara el modelo DCNN con el conjunto de datos aumentado (26,565 imágenes) y con aplicación de pesos, sin y con *transfer learning*.

### 4.6.1. InceptionV3

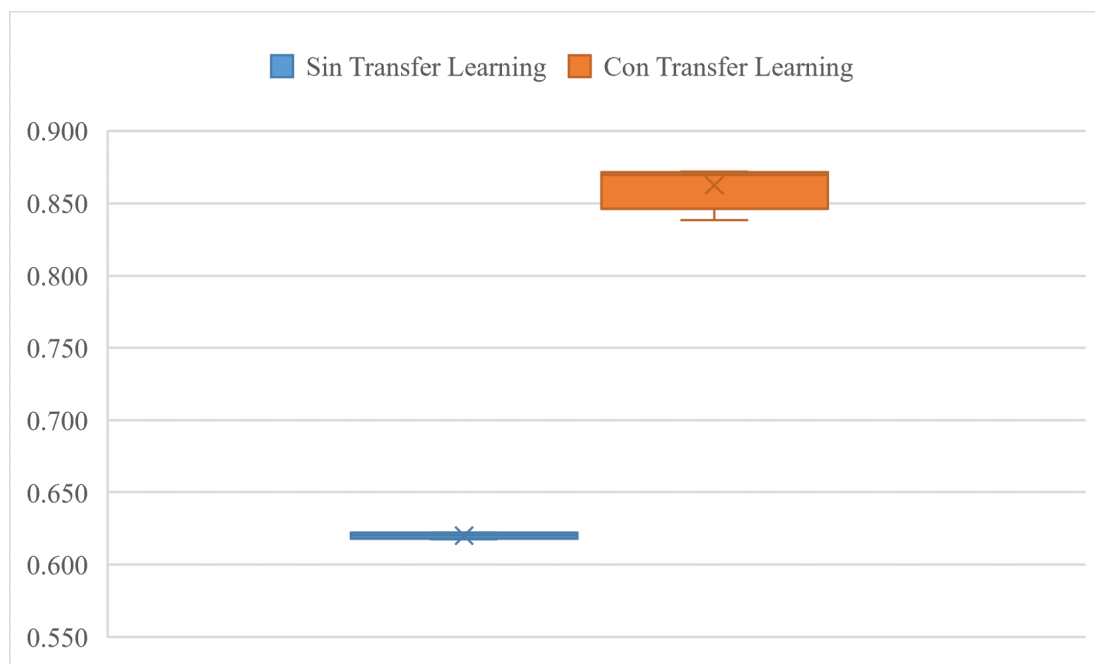
Se observa en la Tabla 10 y la Figura 31 la diferencia notoria con respecto al uso de *transfer learning* aumentando su *accuracy* en un 24 % en promedio.

**Tabla 10**

*Resultados experimentales para InceptionV3*

	<i>Sin transfer learning</i>	<i>Con transfer learning</i>
Fold 1	0.6177	0.8704
Fold 2	0.6192	0.8693
Fold 3	0.6222	0.8386
Fold 4	0.6211	0.8720
Avg.	0.62005	0.86257
Mdn.	0.62015	0.86984
Var.	0.000004	0.00026
S.D.	0.00001	0.00077

*Fuente.* Elaboración Propia.



**Figura 31**

*Comparativo de evaluación usando transfer learning para InceptionV3.*

*Fuente.* Elaboración Propia.

### 4.6.2. ResNet152V2

Se observa en la Tabla 11 y la Figura 32 la diferencia notoria con respecto al uso de *transfer learning* aumentando su *accuracy* en un 26 % en promedio.

**Tabla 11**  
Resultados experimentales para ResNet152V2

	<i>Sin transfer learning</i>	<i>Con transfer learning</i>
Fold 1	0.5629	0.8126
Fold 2	0.5376	0.8155
Fold 3	0.5440	0.8234
Fold 4	0.6144	0.8101
Avg.	0.56473	0.81540
Mdn.	0.55345	0.81405
Var.	0.00121	0.00003
S.D.	0.00364	0.00010

Fuente. Elaboración Propia.



**Figura 32**  
Comparativo de evaluación usando transfer learning para ResNet152V2.  
Fuente. Elaboración Propia.

### 4.6.3. DenseNet121

Se observa en la Tabla 12 y la Figura 33 la diferencia notoria con respecto al uso de *transfer learning* aumentando su *accuracy* en un 29 % aprox. en promedio.

**Tabla 12**  
Resultados experimentales para DenseNet121

	<i>Sin transfer learning</i>	<i>Con transfer learning</i>
Fold 1	0.5178	0.8261
Fold 2	0.5770	0.8276
Fold 3	0.5671	0.8601
Fold 4	0.5079	0.8372
Avg.	0.54245	0.83776
Mdn.	0.54245	0.83241
Var.	0.00120	0.00025
S.D.	0.00360	0.00074

Fuente. Elaboración Propia.



**Figura 33**  
Comparativo de evaluación usando transfer learning para DenseNet121.  
Fuente. Elaboración Propia.

#### 4.6.4. EfficientNetB6

Se observa en la Tabla 13 y la Figura 34 la diferencia notoria con respecto al uso de *transfer learning* aumentando su *accuracy* en un 44 % en promedio.

**Tabla 13**  
Resultados experimentales para *EfficientNetB6*

	<i>Sin transfer learning</i>	<i>Con transfer learning</i>
Fold 1	0.3842	0.8546
Fold 2	0.4301	0.8719
Fold 3	0.3906	0.8628
Fold 4	0.4665	0.8673
Avg.	0.41785	0.86415
Mdn.	0.41035	0.86505
Var.	0.00146	0.00005
S.D.	0.00439	0.00016

Fuente. Elaboración Propia.



**Figura 34**  
Comparativo de evaluación usando transfer learning para *EfficientNetB6*.  
Fuente. Elaboración Propia.

#### 4.7. Hipótesis específica: congelamiento de capa.

La premisa de la hipótesis específica es demostrar que el congelamiento de capa mejoran los indicadores de diagnóstico. Para esto se usan los métodos de búsqueda binaria modificada y algoritmos genéticos.

### 4.7.1. InceptionV3

En la Tabla 14 se observa que el congelamiento hasta la capa 4 es mejor con respecto a no congelar. En la Tabla 15 se encuentra el detalle sumariado.

**Tabla 14**

*Resultados experimentales detallados para InceptionV3*

Capa	0	1	2	4	9	19	38	77	155	311
Fold 1	0.87037	0.86826	0.85577	0.84779	0.69874	0.71981	0.64905	0.57919	0.51009	0.46010
Fold 2	0.86930	0.85364	0.85484	0.86117	0.74040	0.72640	0.63183	0.56151	0.58304	0.46815
Fold 3	0.83858	0.85123	0.87020	0.84671	0.72594	0.71028	0.65487	0.57220	0.52763	0.45370
Fold 4	0.87201	0.86523	0.88104	0.85605	0.73362	0.72128	0.61060	0.58365	0.57130	0.46409
Avg.	0.86257	0.85959	0.86546	0.85293	0.72468	0.71944	0.63659	0.57414	0.54802	0.46151
Mdn.	0.869835	0.859435	0.862985	0.851920	0.729780	0.720545	0.640440	0.575695	0.549465	0.462095
Var.	0.000257	0.000071	0.000157	0.000048	0.000334	0.000045	0.000396	0.000093	0.001208	0.000038
S.D.	0.000771	0.000212	0.000472	0.000143	0.001002	0.000136	0.001188	0.000279	0.003623	0.000114

*Fuente.* Elaboración Propia.

**Tabla 15**

*Resultados experimentales sumariados para InceptionV3*

Capa/Iteración	1	2	3	4	5	6	7	8
0	<b>0.862565</b>	<b>0.862565</b>	<b>0.862565</b>	<b>0.862565</b>	<b>0.862565</b>	<b>0.862565</b>	0.862565	0.862565
1								<b>0.85959</b>
2							<b>0.8654625</b>	<b>0.8654625</b>
4						0.852930	0.852930	
9					0.724675	0.724675		
19				0.7194425	0.7194425			
38			0.6365875	0.6365875				
77		0.5741375	0.5741375					
155	0.548015	0.548015						
311	0.461510							

*Fuente.* Elaboración Propia.

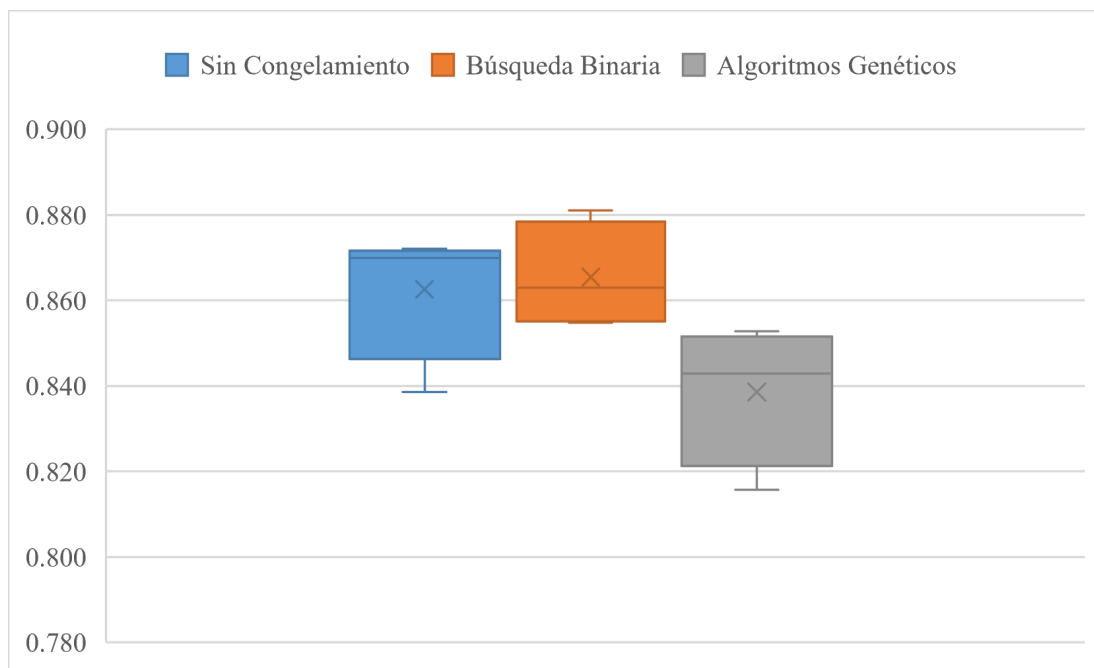
En la Tabla 16 y en la Figura 35 se muestra el comparativo con los resultados obtenidos con los algoritmos genéticos y el no aplicar congelamiento.

**Tabla 16**

*Resultados comparativos para InceptionV3*

	Sin congelamiento	Búsqueda binaria	Algoritmos genéticos
Fold 1	0.87037	0.85577	0.85280
Fold 2	0.86930	0.85484	0.83780
Fold 3	0.83858	0.87020	0.81570
Fold 4	0.87201	0.88104	0.84790
Avg.	0.862565	0.865463	0.838550
Mdn.	0.869835	0.862985	0.842850
Var.	0.000257	0.000157	0.000271
S.D.	0.000771	0.000472	0.000813

*Fuente.* Elaboración Propia.



**Figura 35**

*Comparativo de métodos de optimización para InceptionV3.*

*Fuente. Elaboración Propia.*

#### 4.7.2. ResNet152V2

En la Tabla 17 se observa que el congelamiento hasta la capa 26 es mejor con respecto a no congelar. En la Tabla 18 se encuentra el detalle sumariado.

**Tabla 17**

*Resultados experimentales detallados para ResNet152V2*

Capa	0	17	26	35	70	141	282	564
Fold 1	0.81256	0.82535	0.81722	0.82325	0.76920	0.69121	0.63369	0.50783
Fold 2	0.81554	0.82744	0.82578	0.84551	0.76058	0.70456	0.62852	0.51348
Fold 3	0.82337	0.83406	0.82397	0.83391	0.77157	0.70155	0.62551	0.49375
Fold 4	0.81012	0.82756	0.82111	0.81629	0.80214	0.68996	0.62957	0.50866
Avg.	0.81540	0.82860	0.82202	0.82974	0.77587	0.69682	0.62932	0.50593
Mdn.	0.81405	0.82750	0.82254	0.82858	0.77039	0.69638	0.62905	0.50825
Var.	0.00003	0.00001	0.00001	0.00016	0.00033	0.00005	0.00001	0.00007
S.D.	0.00010	0.00004	0.00004	0.00049	0.00099	0.00016	0.00003	0.00022

*Fuente. Elaboración Propia.*

**Tabla 18***Resultados experimentales resumidos para ResNet152V2*

Capa/Iteración	1	2	3	4	5	6
0	<b>0.8153975</b>	<b>0.8153975</b>	<b>0.8153975</b>	0.8153975	0.8153975	
17					0.8286025	0.8286025
26						<b>0.8220200</b>
35				<b>0.8297400</b>	<b>0.8297400</b>	<b>0.8297400</b>
70			0.7758725	0.7758725		
141		0.6968200	0.6968200			
282	0.6293225	0.6293225				
564	0.5059300					

*Fuente.* Elaboración Propia.

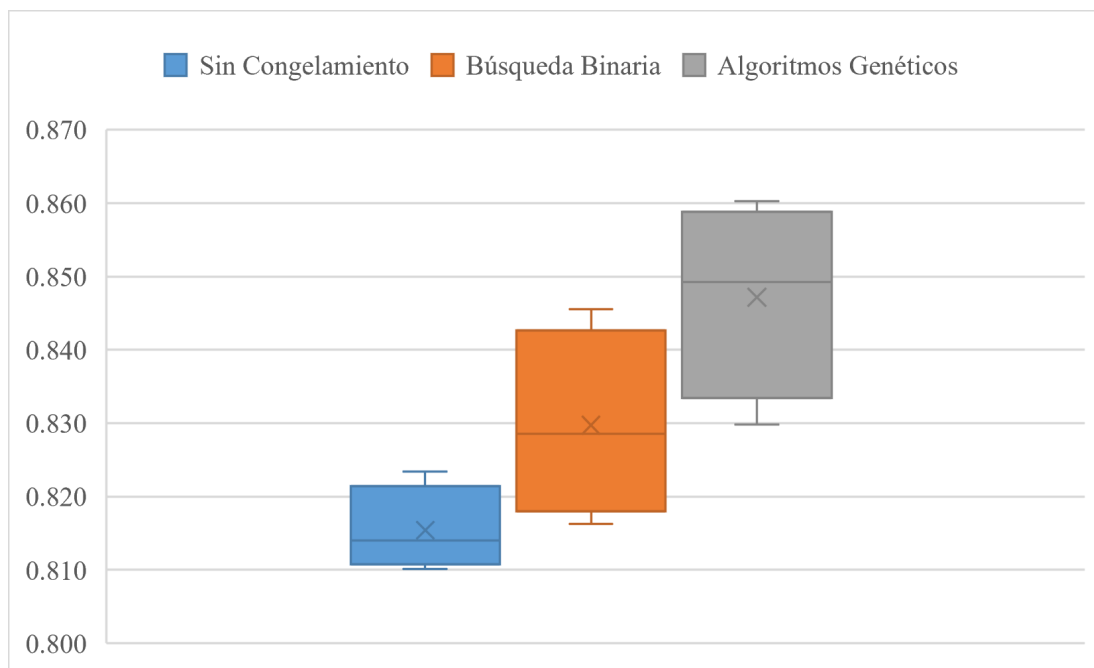
En la Tabla 19 y en la Figura 36 se muestra el comparativo con los resultados obtenidos con los algoritmos genéticos y el no aplicar congelamiento.

**Tabla 19***Resultados comparativos para ResNet152V2*

	Sin congelamiento	Búsqueda binaria	Algoritmos genéticos
Fold 1	0.81256	0.82325	0.86030
Fold 2	0.81554	0.84551	0.85420
Fold 3	0.82337	0.83391	0.84430
Fold 4	0.81012	0.81629	0.82980
Avg.	0.81540	0.82974	0.84715
Mdn.	0.81405	0.82858	0.84925
Var.	0.00003	0.00016	0.00018
S.D.	0.00010	0.00049	0.00053

*Fuente.* Elaboración Propia.





**Figura 36**

*Comparativo de métodos de optimización para ResNet152V2.*

*Fuente. Elaboración Propia.*

### 4.7.3. DenseNet121

En la Tabla 20 se observa que el congelamiento hasta la capa 53 es mejor con respecto a no congelar.

**Tabla 20**

*Resultados experimentales detallados para DenseNet121*

Capa	0	53	66	79	106	213	427
Fold 1	0.82611	0.86073	0.86706	0.86224	0.84974	0.75248	0.57678
Fold 2	0.82759	0.88737	0.84054	0.86644	0.86327	0.76841	0.58259
Fold 3	0.86011	0.87035	0.86132	0.86493	0.83436	0.78301	0.57687
Fold 4	0.83722	0.87637	0.86583	0.86764	0.85047	0.77940	0.58169
Avg.	0.83776	0.87371	0.85869	0.86531	0.84946	0.77083	0.57948
Mdn.	0.83241	0.87336	0.86358	0.86569	0.85011	0.77391	0.57928
Var.	0.00025	0.00012	0.00015	0.00001	0.00014	0.00019	0.00001
S.D.	0.00074	0.00037	0.00046	0.00002	0.00042	0.00056	0.00003

*Fuente. Elaboración Propia.*

A continuación, en la Tabla 21 podemos observar el detalle sumariado.

**Tabla 21***Resultados experimentales resumidos para DenseNet121*

<b>Capa/Iteración</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
0	<b>0.837758</b>	0.837758	0.837758		
53			<b>0.873705</b>	<b>0.873705</b>	<b>0.873705</b>
66					<b>0.858688</b>
79				0.865313	0.865313
106		<b>0.84946</b>	0.84946	0.84946	
213	0.770825	0.770825			
427	0.579483				

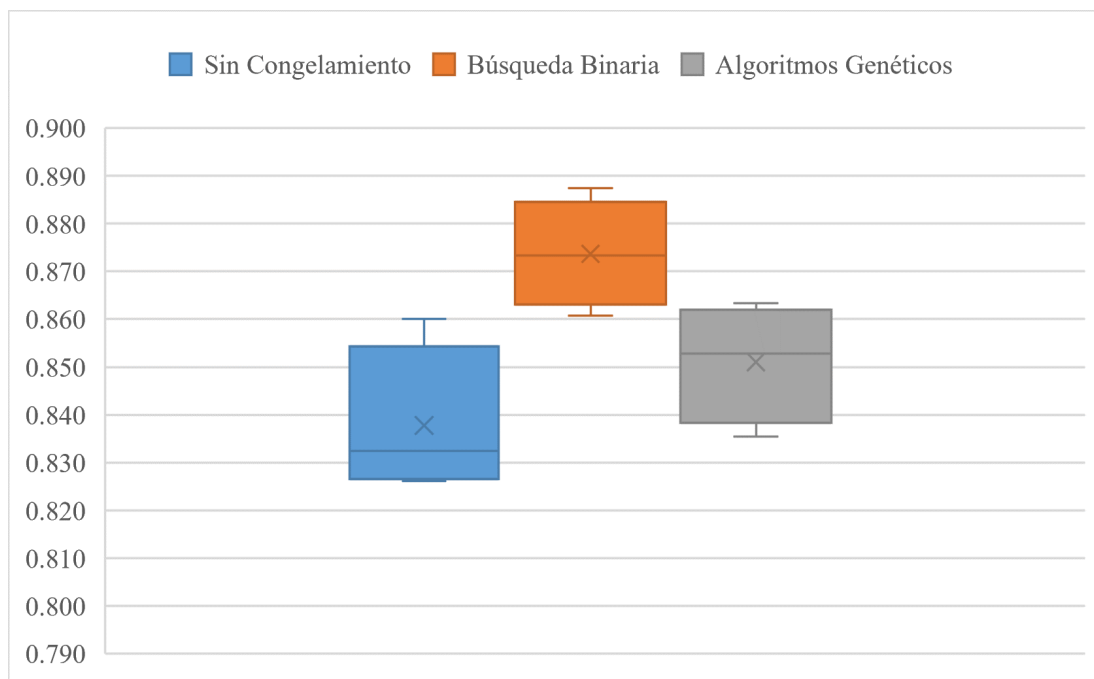
*Fuente.* Elaboración Propia.

En la Tabla 22 y en la Figura 37 se muestra el comparativo con los resultados obtenidos con los algoritmos genéticos y el no aplicar congelamiento.

**Tabla 22***Resultados comparativos para DenseNet121*

	<b>Sin congelamiento</b>	<b>Búsqueda binaria</b>	<b>Algoritmos genéticos</b>
Fold 1	0.82611	0.86073	0.83540
Fold 2	0.82759	0.88737	0.84720
Fold 3	0.86011	0.87035	0.86330
Fold 4	0.83722	0.87637	0.85830
Avg.	0.83776	0.87371	0.85105
Mdn.	0.83241	0.87336	0.85275
Var.	0.00025	0.00012	0.00015
S.D.	0.00074	0.00037	0.00046

*Fuente.* Elaboración Propia.



**Figura 37**

*Comparativo de métodos de optimización para DenseNet121.*

*Fuente. Elaboración Propia.*

#### 4.7.4. EfficientNetB6

En la Tabla 23 se observa que el congelamiento hasta la capa 330 es mejor con respecto a no congelar. En la Tabla 24 podemos observar el detalle sumariado.

**Tabla 23**

*Resultados experimentales detallados para EfficientNetB6*

Capa	0	166	249	291	312	322	327	330	331	333	666
Fold 1	0.8546	0.8654	0.8746	0.8844	0.8865	0.8850	0.8865	0.8892	0.8820	0.8726	0.6090
Fold 2	0.8719	0.8850	0.8806	0.8819	0.8813	0.8845	0.8872	0.8922	0.8860	0.8856	0.6095
Fold 3	0.8628	0.8628	0.8675	0.8789	0.8764	0.8806	0.8761	0.8863	0.8819	0.8947	0.6225
Fold 4	0.8673	0.8707	0.8694	0.8777	0.8704	0.8872	0.8827	0.8765	0.8806	0.8868	0.6108
Avg.	0.8642	0.8710	0.8730	0.8807	0.8787	0.8843	0.8831	0.8861	0.8826	0.8849	0.6130
Mdn.	0.8651	0.8680	0.8720	0.8804	0.8789	0.8848	0.8846	0.8878	0.8820	0.8862	0.6102
Var.	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0000	0.0001	0.0001
S.D.	0.0002	0.0003	0.0001	0.0001	0.0001	0.0000	0.0001	0.0001	0.0000	0.0003	0.0001

*Fuente. Elaboración Propia.*

**Tabla 24***Resultados experimentales resumidos para EfficientNetB6*

Capa/Iteración	1	2	3	4	5	6	7	8	9
0	0.8642	0.8642							
166		0.8710	0.8710						
249			0.8730	0.8730					
291				0.8807	0.8807				
312					0.8787	0.8787			
322						0.8843	0.8843		
327							0.8831	0.8831	
330								0.8861	0.8861
331									0.8826
333	0.8849	0.8849	0.8849	0.8849	0.8849	0.8849	0.8849	0.8849	0.8849
666	0.6130								

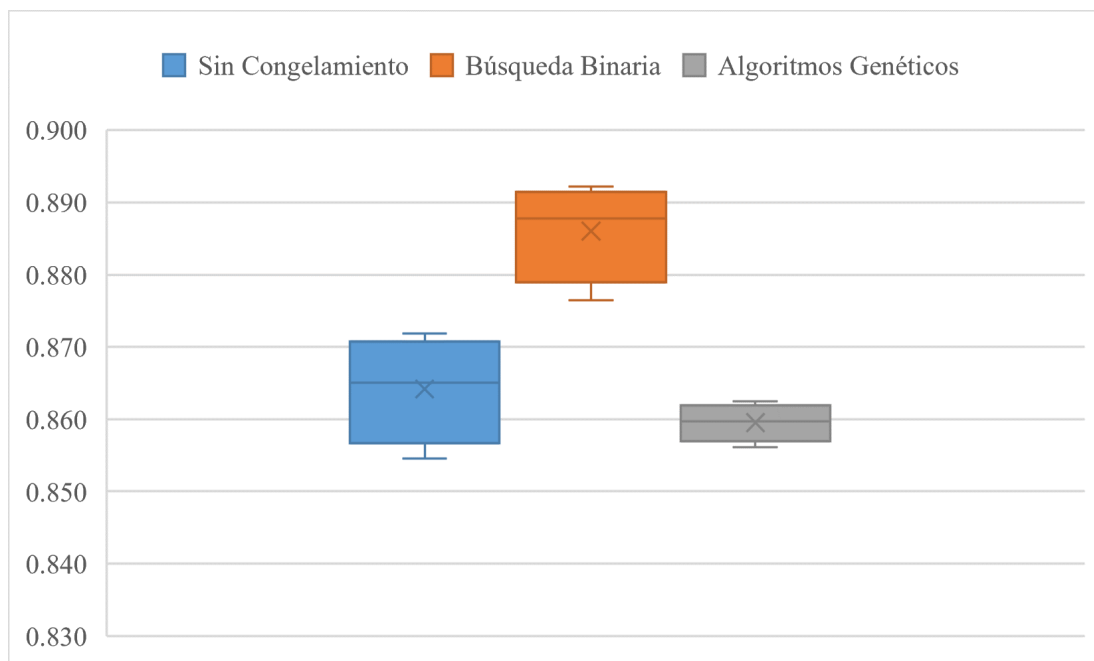
*Fuente.* Elaboración Propia.

En la Tabla 25 y en la Figura 38 se muestra el comparativo con los resultados obtenidos con los algoritmos genéticos y el no aplicar congelamiento.

**Tabla 25***Resultados comparativos para EfficientNetB6*

	Sin congelamiento	Búsqueda binaria	Algoritmos genéticos
Fold 1	0.8546	0.8892	0.8561
Fold 2	0.8719	0.8922	0.8625
Fold 3	0.8628	0.8863	0.8601
Fold 4	0.8673	0.8765	0.8594
Avg.	0.864150	0.886050	0.859525
Mdn.	0.865050	0.887750	0.859750
Var.	0.000054	0.000046	0.000007
S.D.	0.000163	0.000139	0.000021

*Fuente.* Elaboración Propia.



**Figura 38**

*Comparativo de métodos de optimización para EfficientNetB6.*

*Fuente. Elaboración Propia.*

## 4.8. Hipótesis específica: ensamblado.

Según Dietterich (2000), si se escogen los pesos indicados para cada modelo se llega a un mejor resultado, el ensamblaje combina las fortalezas de diferentes algoritmos para crear un modelo más robusto y preciso. Al combinar varios modelos, se reducen los errores y aumentar la estabilidad del modelo final, especialmente en casos de datos no balanceados o con ruido en los datos. Además, el ensamblaje puede mejorar la capacidad de generalización del modelo, lo que significa que es más probable que se desempeñe de forma correcta con datos nuevos y desconocidos.

Finalmente, para la fase experimental se aplicaron 3 técnicas de ensembling y se comparan con los mejores resultados obtenidos por modelo en la sección anterior (congelamiento de capa) usando la búsqueda binaria. Los métodos de ensamblaje también son mencionados por Raschka (2021):

- *Hard Voting*: se tomará la moda de los resultados de cada modelo.

$$\hat{y} = \text{moda}\{C_1(x), C_2(x), \dots, C_m(x)\}$$

Una de las desventajas del uso de este modelo se debe a que no toma en cuenta la distribución probabilística de cada modelo (solo toma la predicción final), además no contempla si un modelo tiene mejor precisión que otro.

- *Soft Voting*: a diferencia del método anterior, aquí se toma en cuenta la distribución probabilística de cada modelo y junto con un peso por cada modelo.

$$\hat{y} = \operatorname{argmax}_i \sum_{j=1}^m w_j p_{ij}$$

En este caso si el peso asignado a cada modelo es el mismo, se tomaría la media aritmética de cada probabilidad por modelo y tomamos aquella de mayor valor y se asigna la predicción.

- *Weighted Soft Voting* aplicando algoritmos genéticos: Es una extensión de la definición de *Soft Voting*, en este caso los pesos seleccionados para cada modelo será determinado por un algoritmo genético.

#### 4.8.1. *Hard Voting*.

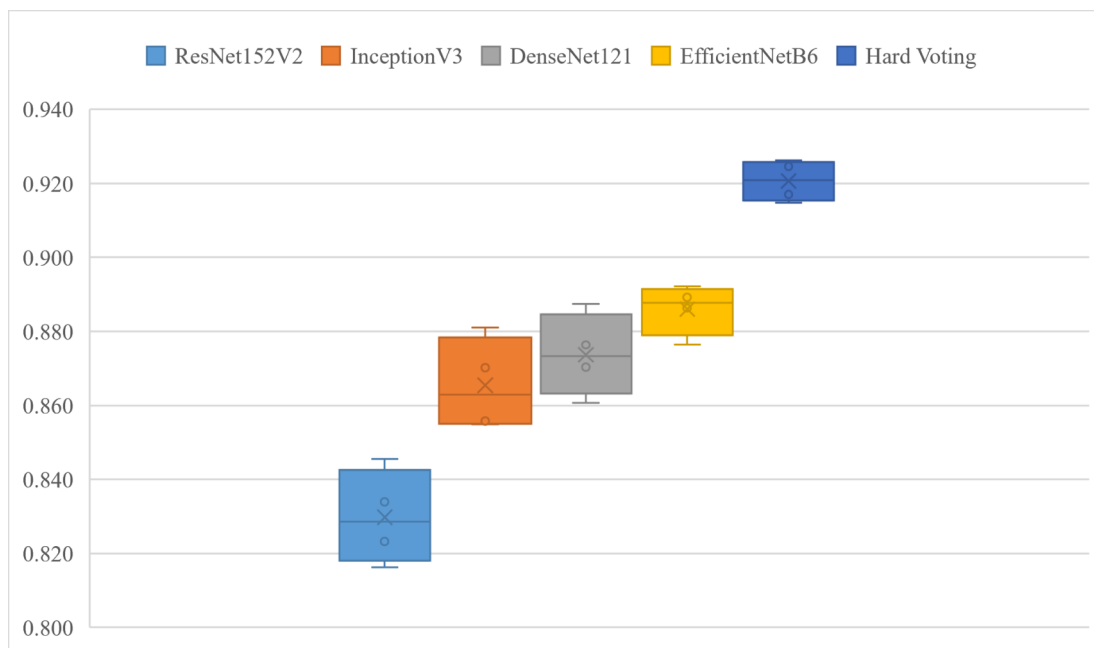
Se observa en la Tabla 26 y la Figura 39 la mejora al aplicar el ensamblaje por *Hard Voting* comparado con cada modelo.

**Tabla 26**

*Resultados comparativos para el método de ensamblaje Hard Voting.*

	<b>ResNet152V2</b>	<b>InceptionV3</b>	<b>DenseNet121</b>	<b>EfficientNetB6</b>	<b>Hard Voting</b>
Fold 1	0.8233	0.8558	0.8607	0.8892	0.9170
Fold 2	0.8455	0.8548	0.8874	0.8922	0.9262
Fold 3	0.8339	0.8702	0.8704	0.8863	0.9246
Fold 4	0.8163	0.8810	0.8764	0.8765	0.9148
Avg.	0.82974	0.86546	0.87371	0.88605	0.92065
Mdn.	0.82858	0.86299	0.87336	0.88775	0.92080
Var.	0.00016	0.00016	0.00012	0.00005	0.00003
S.D.	0.00049	0.00047	0.00037	0.00014	0.00009

*Fuente.* Elaboración Propia.



**Figura 39**

*Comparativo para el método de ensamblaje Hard Voting.*

*Fuente. Elaboración Propia.*

#### 4.8.2. Average Soft Voting.

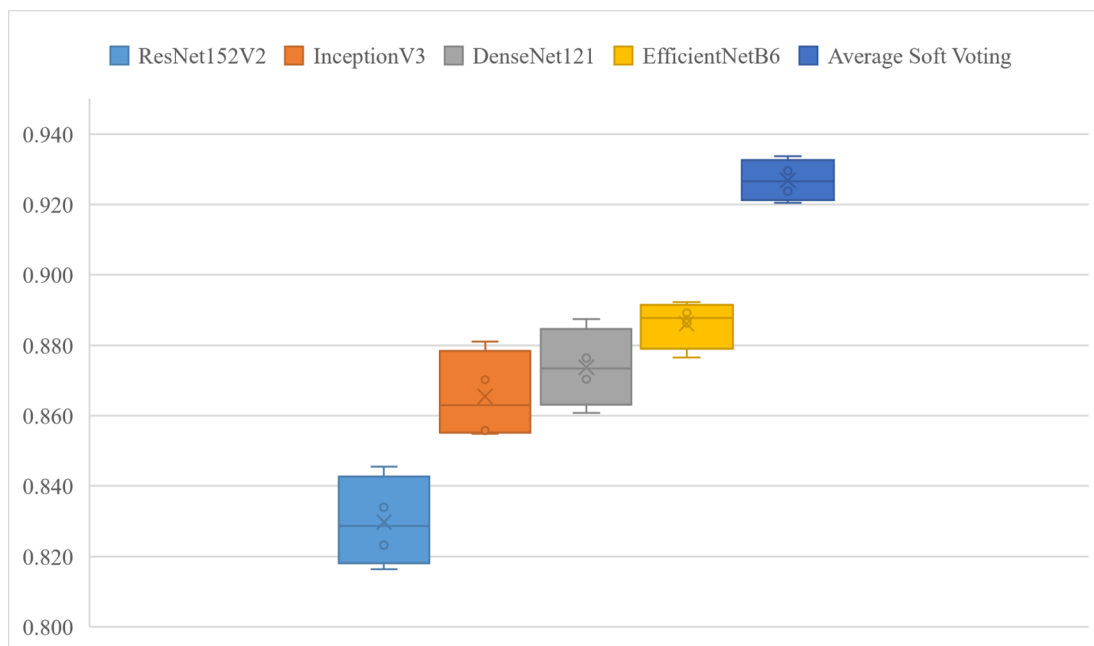
Se observa en la Tabla 27 y la Figura 40 la mejora al aplicar el ensamblaje por *Average Soft Voting* con cada modelo.

**Tabla 27**

*Resultados comparativos para el método de ensamblaje Average Soft Voting.*

	ResNet152V2	InceptionV3	DenseNet121	EfficientNetB6	Avg. Soft Voting
Fold 1	0.8233	0.8558	0.8607	0.8892	0.9237
Fold 2	0.8455	0.8548	0.8874	0.8922	0.9337
Fold 3	0.8339	0.8702	0.8704	0.8863	0.9295
Fold 4	0.8163	0.8810	0.8764	0.8765	0.9205
Avg.	0.82974	0.86546	0.87371	0.88605	0.92685
Mdn.	0.82858	0.86299	0.87336	0.88775	0.92660
Var.	0.00016	0.00016	0.00012	0.00005	0.00003
S.D.	0.00049	0.00047	0.00037	0.00014	0.00010

*Fuente. Elaboración Propia.*



**Figura 40**

*Comparativo para el método de ensamblaje Average Soft Voting.*

*Fuente. Elaboración Propia.*

### 4.8.3. *Weighted Soft Voting* usando algoritmos genéticos.

Se observa en la Tabla 28 y la Figura 41 la mejora al aplicar el ensamblaje por *Weighted Soft Voting* usando algoritmos genéticos con cada modelo.

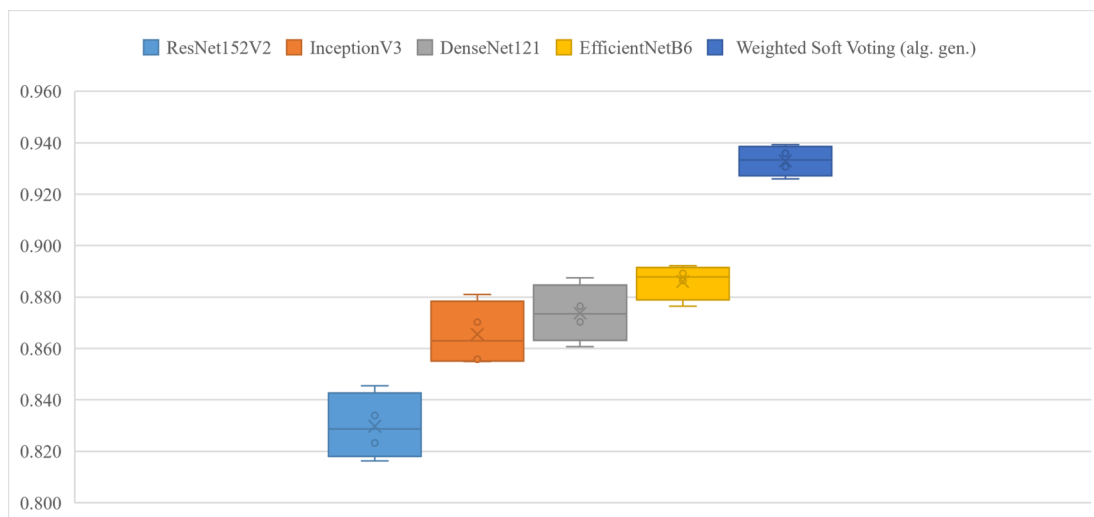
**Tabla 28**

*Resultados comparativos para el método de ensamblaje Weighted Soft Voting usando algoritmos genéticos.*

	ResNet152V2	InceptionV3	DenseNet121	EfficientNetB6	W. Soft Voting (A.G.)
Fold 1	0.8233	0.8558	0.8607	0.8892	0.9306
Fold 2	0.8455	0.8548	0.8874	0.8922	0.9393
Fold 3	0.8339	0.8702	0.8704	0.8863	0.9359
Fold 4	0.8163	0.8810	0.8764	0.8765	0.9259
Avg.	0.82974	0.86546	0.87371	0.88605	0.93293
Mdn.	0.82858	0.86299	0.87336	0.88775	0.93325
Var.	0.00016	0.00016	0.00012	0.00005	0.00003
S.D.	0.00049	0.00047	0.00037	0.00014	0.00010

*Fuente. Elaboración Propia.*





**Figura 41**

Comparativo para el método de ensamblaje *Weighted Soft Voting* usando algoritmos genéticos.

Fuente. Elaboración Propia.

Finalmente, en la Tabla 29 tenemos los pesos obtenidos mediante algoritmos genéticos, se observa que el modelo *EfficientNetB6* tiene un mayor peso que los demás y *ResNet152V2* el menor.

**Tabla 29**

Pesos obtenidos para el método de ensamblaje *Weighted Soft Voting* usando algoritmos genéticos

	<b>EfficientNetB6</b>	<b>DenseNet121</b>	<b>InceptionV3</b>	<b>ResNet152V2</b>
Fold 1	0.3936	0.3123	0.1886	0.1054
Fold 2	0.3952	0.2958	0.1921	0.1168
Fold 3	0.3692	0.2110	0.2319	0.1879
Fold 4	0.3944	0.1919	0.2981	0.1156
Avg.	0.38811	0.25278	0.22767	0.13145
Mdn.	0.39399	0.25342	0.21199	0.11621
Var.	0.00016	0.00362	0.00259	0.00144
S.D.	0.00048	0.01085	0.00777	0.00433

Fuente. Elaboración Propia.

#### 4.8.4. Comparativo de los métodos de ensamblaje.

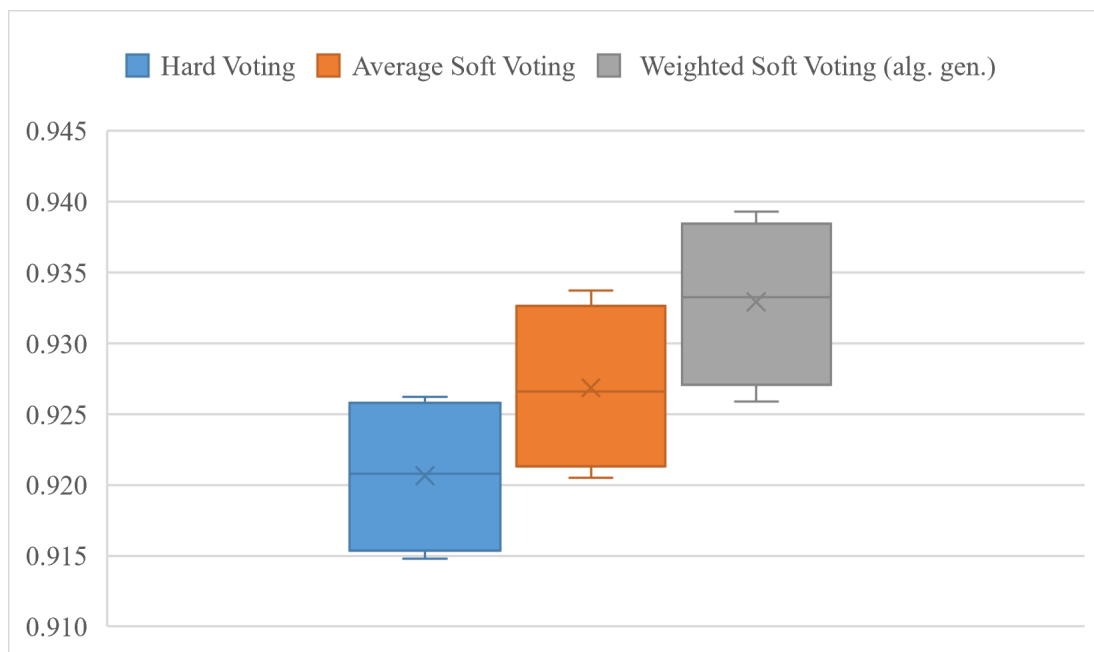
Se observa en la Tabla 30 y la Figura 42 que el ensamblaje por *Weighted Soft Voting* usando algoritmos genéticos produce mejores resultados que los otros 2 métodos de ensamblado.

**Tabla 30**

*Resultados comparativos de los métodos de ensamblaje.*

	<i>Hard Voting</i>	<i>Avg.Soft Voting</i>	<i>W.Soft Voting (A.G.)</i>
Fold 1	0.9170	0.9237	0.9306
Fold 2	0.9262	0.9337	0.9393
Fold 3	0.9246	0.9295	0.9359
Fold 4	0.9148	0.9205	0.9259
Avg.	0.92065	0.92685	0.93293
Mdn.	0.92080	0.92660	0.93325
Var.	0.00003	0.00003	0.00003
S.D.	0.00009	0.00010	0.00010

*Fuente.* Elaboración Propia.

**Figura 42**

*Comparativo de los métodos de ensamblaje.*

*Fuente.* Elaboración Propia.

# Capítulo V

## Resultados y discusión

### 5.1. Análisis, interpretación y discusión de resultados.

El análisis, interpretación y discusión de los resultados se basarán en los indicadores de la variable dependiente de la matriz de operacionalización de variables llamado "desempeño de clasificación de diagnóstico" (Tabla 3), los cuales son:

- Exactitud.
- Precisión.
- Sensibilidad.
- F1-score.
- ROC AUC score.

Finalmente, tomar en cuenta que los valores mostrados para el análisis, interpretación y discusión de resultados así como las pruebas de hipótesis se comparan los valores ponderados de cada indicador extraídos del reporte de clasificación.

#### 5.1.1. Análisis de resultados experimentales en InceptionV3.

Se observa en la Tabla 31 una mejora cercana al 5 % al aplicar el aumento de datos y la posterior aplicación de pesos, la diferencia es cercana al 10 % a nivel del indicador ROC AUC Score, esto debido a que al momento de entrenar el modelo se tomó en cuenta como indicador su exactitud, por ende al enfocarse en un solo indicador hace que los demás indicadores se vean afectados.

**Tabla 31**

*Resultados experimentales para los indicadores de la hipótesis específica 1 en InceptionV3*

	Data no aumentada					Data aumentada y balanceada				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.8199	0.8080	0.8199	0.8100	0.8112	0.8704	0.8725	0.8704	0.8701	0.9184
Fold 2	0.8211	0.8110	0.8211	0.8115	0.8261	0.8693	0.8736	0.8693	0.8694	0.9184
Fold 3	0.8139	0.8216	0.8139	0.8052	0.8402	0.8386	0.8449	0.8386	0.8383	0.8983
Fold 4	0.7982	0.7794	0.7982	0.7771	0.7844	0.8720	0.8737	0.8720	0.8718	0.9193
Avg.	0.81328	0.80500	0.81328	0.80095	0.81548	0.86258	0.86618	0.86258	0.86240	0.91360
Mdn.	0.81690	0.80950	0.81690	0.80760	0.81865	0.86985	0.87305	0.86985	0.86975	0.91840
Var.	0.00011	0.00033	0.00011	0.00026	0.00057	0.00026	0.00020	0.00026	0.00026	0.00010
S.D.	0.00033	0.00098	0.00033	0.00078	0.00171	0.00077	0.00060	0.00077	0.00078	0.00031

*Fuente.* Elaboración Propia.

En la Tabla 32 se observa un evidente incremento sustancial en el uso de *transfer learning* en todos sus indicadores cercano mayor al 20 %. Se debe tener en cuenta que al no hacer uso del *transfer learning* los pesos son iniciados de forma aleatoria, por lo tanto, se requeriría un número mayor de épocas para lograr el resultado que se obtiene por medio de esta.

**Tabla 32**

*Resultados experimentales para los indicadores de la hipótesis específica 2 en InceptionV3*

	Sin transfer learning					Con transfer learning				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.6177	0.6671	0.6177	0.6282	0.7719	0.8704	0.8725	0.8704	0.8701	0.9184
Fold 2	0.6192	0.6665	0.6192	0.6197	0.7708	0.8693	0.8736	0.8693	0.8694	0.9184
Fold 3	0.6222	0.6557	0.6222	0.6159	0.7707	0.8386	0.8449	0.8386	0.8383	0.8983
Fold 4	0.6211	0.6807	0.6211	0.6280	0.7713	0.8720	0.8737	0.8720	0.8718	0.9193
Avg.	0.62005	0.66750	0.62005	0.62295	0.77118	0.86258	0.86618	0.86258	0.86240	0.91360
Mdn.	0.62015	0.66680	0.62015	0.62385	0.77104	0.86985	0.87305	0.86985	0.86975	0.91840
Var.	0.000004	0.00010	0.00000	0.00004	0.00000	0.00026	0.00020	0.00026	0.00026	0.00010
S.D.	0.00001	0.00031	0.00001	0.00011	0.00000	0.00077	0.00060	0.00077	0.00078	0.00031

*Fuente.* Elaboración Propia.

En la Tabla 33 se observa una mejora al aplicar el congelamiento de capa por búsqueda binaria siendo levemente superior en todos sus indicadores, esto es debido a la naturaleza de la arquitectura Inception es del tipo "paralelo", tal como se mostró en Figura 5, por tanto es posible que necesite una mayor cantidad de épocas de entrenamiento.

**Tabla 33**

Resultados experimentales para los indicadores de la hipótesis específica 3 en InceptionV3

	Sin congelamiento					Congelamiento con búsqueda binaria				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.8704	0.8725	0.8704	0.8701	0.9184	0.8558	0.8567	0.8558	0.8554	0.9071
Fold 2	0.8693	0.8736	0.8693	0.8694	0.9184	0.8548	0.8596	0.8548	0.8545	0.9074
Fold 3	0.8386	0.8449	0.8386	0.8383	0.8983	0.8702	0.8782	0.8702	0.8705	0.9193
Fold 4	0.8720	0.8737	0.8720	0.8718	0.9193	0.8810	0.8818	0.8810	0.8807	0.9249
Avg.	0.86258	0.86618	0.86258	0.86240	0.91360	0.86545	0.86908	0.86545	0.86528	0.91466
Mdn.	0.86985	0.87305	0.86985	0.86975	0.91840	0.86300	0.86890	0.86300	0.86295	0.91332
Var.	0.00026	0.00020	0.00026	0.00026	0.00010	0.00016	0.00016	0.00016	0.00016	0.00008
S.D.	0.00077	0.00060	0.00077	0.00078	0.00031	0.00047	0.00049	0.00047	0.00048	0.00024

Fuente. Elaboración Propia.

### 5.1.2. Análisis de resultados experimentales en ResNet152V2.

Se observa en la Tabla 34 una mejora cercana al 1.5 % al aplicar el aumento de datos y la posterior aplicación de pesos, la diferencia es cercana al 10 % a nivel del indicador ROC AUC Score, esto debido a que al momento de entrenar el modelo se tomó en cuenta como indicador su exactitud, por ende al enfocarse en un solo indicador hace que los demás indicadores se vean afectados.

**Tabla 34**

Resultados experimentales para los indicadores de la hipótesis específica 1 en Resnet152V2

	Data no aumentada					Data aumentada y balanceada				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.8055	0.7899	0.8055	0.7835	0.7771	0.8126	0.8182	0.8126	0.8096	0.8807
Fold 2	0.8071	0.7961	0.8071	0.7866	0.7834	0.8155	0.8256	0.8155	0.8160	0.8840
Fold 3	0.8123	0.8001	0.8123	0.8021	0.8109	0.8234	0.8265	0.8234	0.8224	0.8883
Fold 4	0.8122	0.8096	0.8122	0.8043	0.8249	0.8101	0.8184	0.8101	0.8112	0.8825
Avg.	0.80928	0.79893	0.80928	0.79413	0.79908	0.81540	0.82218	0.81540	0.81480	0.88388
Mdn.	0.80965	0.79810	0.80965	0.79435	0.79715	0.81405	0.82200	0.81405	0.81360	0.88325
Var.	0.00001	0.00007	0.00001	0.00011	0.00051	0.00003	0.00002	0.00003	0.00003	0.00001
S.D.	0.00004	0.00020	0.00004	0.00034	0.00154	0.00010	0.00006	0.00010	0.00010	0.00003

Fuente. Elaboración Propia.

En la Tabla 35 se observa un evidente incremento sustancial en el uso de *transfer learning* en todos sus indicadores mayor al 25 %. Se debe tener en cuenta que al no hacer uso del *transfer learning* los pesos son iniciados de forma aleatoria, por lo tanto, se requeriría un número mayor de épocas para lograr el resultado que se obtiene por medio de esta.

**Tabla 35**

*Resultados experimentales para los indicadores de la hipótesis específica 2 en ResNet152V2*

	Sin transfer learning					Con transfer learning				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.5629	0.6309	0.5629	0.5741	0.7393	0.8126	0.8182	0.8126	0.8096	0.8807
Fold 2	0.5376	0.6083	0.5376	0.5378	0.7140	0.8155	0.8256	0.8155	0.8160	0.8840
Fold 3	0.5440	0.6040	0.5440	0.5564	0.7249	0.8234	0.8265	0.8234	0.8224	0.8883
Fold 4	0.6144	0.6176	0.6144	0.6142	0.7570	0.8101	0.8184	0.8101	0.8112	0.8825
Avg.	0.56473	0.61520	0.56473	0.57063	0.73381	0.81540	0.82218	0.81540	0.81480	0.88388
Mdn.	0.55345	0.61295	0.55345	0.56525	0.73212	0.81405	0.82200	0.81405	0.81360	0.88325
Var.	0.00121	0.00014	0.00121	0.00106	0.00035	0.00003	0.00002	0.00003	0.00003	0.00001
S.D.	0.00364	0.00043	0.00364	0.00319	0.00104	0.00010	0.00006	0.00010	0.00010	0.00003

*Fuente.* Elaboración Propia.

En la Tabla 36 se observa una leve mejora en todos sus indicadores al aplicar el congelamiento de capa por búsqueda binaria, por tanto es posible que se necesite una mayor cantidad de épocas de entrenamiento para obtener un mejor resultado. Un caso distinto se presenta al momento de usar algoritmos genéticos como se muestra en la Tabla 37 donde se observa una mejora en todos sus indicadores en un 3 %.

**Tabla 36**

*Resultados experimentales para los indicadores de la hipótesis específica 3 (búsqueda binaria) en ResNet152V2*

	Sin congelamiento					Congelamiento con búsqueda binaria				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.8126	0.8182	0.8126	0.8096	0.8807	0.8232	0.8253	0.8232	0.8234	0.8876
Fold 2	0.8155	0.8256	0.8155	0.8160	0.8840	0.8455	0.8468	0.8455	0.8455	0.9002
Fold 3	0.8234	0.8265	0.8234	0.8224	0.8883	0.8339	0.8385	0.8339	0.8334	0.8937
Fold 4	0.8101	0.8184	0.8101	0.8112	0.8825	0.8163	0.8163	0.8163	0.8154	0.8844
Avg.	0.81540	0.82218	0.81540	0.81480	0.88388	0.82973	0.83173	0.82973	0.82943	0.89148
Mdn.	0.81405	0.82200	0.81405	0.81360	0.88325	0.82855	0.83190	0.82855	0.82840	0.89065
Var.	0.00003	0.00002	0.00003	0.00003	0.00001	0.00016	0.00018	0.00016	0.00017	0.00005
S.D.	0.00010	0.00006	0.00010	0.00010	0.00003	0.00049	0.00055	0.00049	0.00051	0.00014

*Fuente.* Elaboración Propia.

**Tabla 37**

*Resultados experimentales para los indicadores de la hipótesis específica 3 (algoritmos genéticos) en ResNet152V2*

	Sin congelamiento					Congelamiento con algoritmos genéticos				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.8126	0.8182	0.8126	0.8096	0.8807	0.8603	0.8703	0.8603	0.8602	0.9109
Fold 2	0.8155	0.8256	0.8155	0.8160	0.8840	0.8542	0.8608	0.8542	0.8550	0.9097
Fold 3	0.8234	0.8265	0.8234	0.8224	0.8883	0.8443	0.8472	0.8443	0.8441	0.9017
Fold 4	0.8101	0.8184	0.8101	0.8112	0.8825	0.8298	0.8401	0.8298	0.8307	0.8950
Avg.	0.81540	0.82218	0.81540	0.81480	0.88388	0.84715	0.85460	0.84715	0.84750	0.90431
Mdn.	0.81405	0.82200	0.81405	0.81360	0.88325	0.84925	0.85400	0.84925	0.84955	0.90568
Var.	0.00003	0.00002	0.00003	0.00003	0.00001	0.00018	0.00018	0.00018	0.00017	0.00006
S.D.	0.00010	0.00006	0.00010	0.00010	0.00003	0.00053	0.00055	0.00053	0.00051	0.00017

*Fuente.* Elaboración Propia.

### 5.1.3. Análisis de resultados experimentales en DenseNet121.

Se observa en la Tabla 38 una mejora cercana al 1 % al aplicar el aumento de datos y la posterior aplicación de pesos, la diferencia es cercana al 6 % a nivel del indicador ROC AUC Score, esto debido a que al momento de entrenar el modelo se tomó en cuenta como indicador su exactitud, por ende al enfocarse en un solo indicador hace que los demás indicadores se vean afectados.

**Tabla 38**

*Resultados experimentales para los indicadores de la hipótesis específica 1 en DenseNet121*

	Data no aumentada					Data aumentada y balanceada				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.8355	0.8299	0.8355	0.8226	0.8292	0.8261	0.8355	0.8261	0.8256	0.8915
Fold 2	0.8299	0.8319	0.8299	0.8263	0.8517	0.8276	0.8433	0.8276	0.8272	0.8909
Fold 3	0.8259	0.8146	0.8259	0.8176	0.8308	0.8601	0.8652	0.8601	0.8597	0.9124
Fold 4	0.8190	0.8109	0.8190	0.8070	0.8095	0.8372	0.8412	0.8372	0.8363	0.8971
Avg.	0.82758	0.82183	0.82758	0.81838	0.83030	0.83775	0.84630	0.83775	0.83720	0.89798
Mdn.	0.82790	0.82225	0.82790	0.82010	0.83000	0.83240	0.84225	0.83240	0.83175	0.89430
Var.	0.00005	0.00011	0.00005	0.00007	0.00030	0.00025	0.00017	0.00025	0.00025	0.00010
S.D.	0.00014	0.00034	0.00014	0.00021	0.00089	0.00074	0.00051	0.00074	0.00074	0.00030

*Fuente.* Elaboración Propia.

En la Tabla 39 se observa un evidente incremento sustancial en el uso de *transfer learning* en todos sus indicadores mayor al 29 % a nivel de exactitud y un 17 % a nivel de ROC AUC Score. Se debe tener en cuenta que al no hacer uso del *transfer learning* los pesos son iniciados de forma aleatoria, por lo tanto, se requeriría un número mayor de épocas para lograr el resultado que se obtiene por medio de esta.

**Tabla 39**

*Resultados experimentales para los indicadores de la hipótesis específica 2 en DenseNet121*

	Sin transfer learning					Con transfer learning				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.5178	0.6044	0.5178	0.5405	0.7126	0.8261	0.8355	0.8261	0.8256	0.8915
Fold 2	0.5770	0.6360	0.5770	0.5912	0.7480	0.8276	0.8433	0.8276	0.8272	0.8909
Fold 3	0.5671	0.5919	0.5671	0.5497	0.7231	0.8601	0.8652	0.8601	0.8597	0.9124
Fold 4	0.5079	0.6535	0.5079	0.5405	0.7169	0.8372	0.8412	0.8372	0.8363	0.8971
Avg.	0.54245	0.62145	0.54245	0.55548	0.72515	0.83775	0.84630	0.83775	0.83720	0.89798
Mdn.	0.54245	0.62020	0.54245	0.54510	0.72002	0.83240	0.84225	0.83240	0.83175	0.89430
Var.	0.00120	0.00080	0.00120	0.00059	0.00025	0.00025	0.00017	0.00025	0.00025	0.00010
S.D.	0.00360	0.00240	0.00360	0.00176	0.00075	0.00074	0.00051	0.00074	0.00074	0.00030

*Fuente.* Elaboración Propia.

En la Tabla 40 la mejora al aplicar el congelamiento de capa por búsqueda binaria en todos sus indicadores.

**Tabla 40**

Resultados experimentales para los indicadores de la hipótesis específica 3 en DenseNet121

	Sin congelamiento					Congelamiento con búsqueda binaria				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.8261	0.8355	0.8261	0.8256	0.8915	0.8607	0.8707	0.8607	0.8621	0.9147
Fold 2	0.8276	0.8433	0.8276	0.8272	0.8909	0.8874	0.8898	0.8874	0.8876	0.9291
Fold 3	0.8601	0.8652	0.8601	0.8597	0.9124	0.8704	0.8727	0.8704	0.8694	0.9176
Fold 4	0.8372	0.8412	0.8372	0.8363	0.8971	0.8764	0.8791	0.8764	0.8761	0.9224
Avg.	0.83775	0.84630	0.83775	0.83720	0.89798	0.87373	0.87808	0.87373	0.87380	0.92094
Mdn.	0.83240	0.84225	0.83240	0.83175	0.89430	0.87340	0.87590	0.87340	0.87275	0.91999
Var.	0.00025	0.00017	0.00025	0.00025	0.00010	0.00012	0.00007	0.00012	0.00012	0.00004
S.D.	0.00074	0.00051	0.00074	0.00074	0.00030	0.00037	0.00022	0.00037	0.00035	0.00012

Fuente. Elaboración Propia.

#### 5.1.4. Análisis de resultados experimentales en EfficientNetB6.

Se observa en la Tabla 41 una mejora cercana al 6 % al aplicar el aumento de datos y la posterior aplicación de pesos, la diferencia es cercana al 9 % a nivel del indicador ROC AUC Score, esto debido a que al momento de entrenar el modelo se tomó en cuenta como indicador su exactitud, por ende al enfocarse en un solo indicador hace que los demás indicadores se vean afectados.

**Tabla 41**

Resultados experimentales para los indicadores de la hipótesis específica 1 en EfficientNetB6

	Data no aumentada					Data aumentada y balanceada				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.7812	0.7836	0.7812	0.7804	0.8155	0.8546	0.8563	0.8546	0.8534	0.9090
Fold 2	0.8043	0.7996	0.8043	0.7989	0.8299	0.8719	0.8729	0.8719	0.8711	0.9188
Fold 3	0.7975	0.7804	0.7975	0.7841	0.7968	0.8628	0.8653	0.8628	0.8623	0.9131
Fold 4	0.7958	0.7946	0.7958	0.7904	0.8193	0.8673	0.8690	0.8673	0.8671	0.9163
Avg.	0.79470	0.78955	0.79470	0.78845	0.81537	0.86415	0.86588	0.86415	0.86348	0.91430
Mdn.	0.79665	0.78910	0.79665	0.78725	0.81739	0.86505	0.86715	0.86505	0.86470	0.91471
Var.	0.00009	0.00008	0.00009	0.00007	0.00019	0.00005	0.00005	0.00005	0.00006	0.00002
S.D.	0.00028	0.00025	0.00028	0.00020	0.00057	0.00016	0.00015	0.00016	0.00017	0.00005

Fuente. Elaboración Propia.

En la Tabla 42 se observa un evidente incremento sustancial en el uso de *transfer learning* en todos sus indicadores mayor al 44 % a nivel de exactitud y de ROC AUC Score. Se debe tener en cuenta que al no hacer uso del *transfer learning* los pesos son iniciados de forma aleatoria, por lo tanto, se requeriría un número mayor de épocas para lograr el resultado que se obtiene por medio de esta.



**Tabla 42**

*Resultados experimentales para los indicadores de la hipótesis específica 2 en EfficientNetB6*

	Sin transfer learning					Con transfer learning				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.3842	0.4629	0.3842	0.3944	0.6367	0.8546	0.8563	0.8546	0.8534	0.9090
Fold 2	0.4301	0.5013	0.4301	0.4395	0.6619	0.8719	0.8729	0.8719	0.8711	0.9188
Fold 3	0.3906	0.4551	0.3906	0.3843	0.6325	0.8628	0.8653	0.8628	0.8623	0.9131
Fold 4	0.4665	0.5225	0.4665	0.4782	0.6771	0.8673	0.8690	0.8673	0.8671	0.9163
Avg.	0.41785	0.48545	0.41785	0.42410	0.65207	0.86415	0.86588	0.86415	0.86348	0.91430
Mdn.	0.41035	0.48210	0.41035	0.41695	0.64930	0.86505	0.86715	0.86505	0.86470	0.91471
Var.	0.00146	0.00102	0.00146	0.00188	0.00045	0.00005	0.00005	0.00005	0.00006	0.00002
S.D.	0.00439	0.00305	0.00439	0.00563	0.00134	0.00016	0.00015	0.00016	0.00017	0.00005

*Fuente.* Elaboración Propia.

En la Tabla 43 se observa la mejora al aplicar el congelamiento de capa por búsqueda binaria en todos sus indicadores.

**Tabla 43**

*Resultados experimentales para los indicadores de la hipótesis específica 3 en EfficientNetB6*

	Sin congelamiento					Congelamiento con búsqueda binaria				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.8546	0.8563	0.8546	0.8534	0.9090	0.8892	0.8921	0.8892	0.8889	0.9303
Fold 2	0.8719	0.8729	0.8719	0.8711	0.9188	0.8922	0.8932	0.8922	0.8916	0.9319
Fold 3	0.8628	0.8653	0.8628	0.8623	0.9131	0.8863	0.8891	0.8863	0.8860	0.9281
Fold 4	0.8673	0.8690	0.8673	0.8671	0.9163	0.8765	0.8779	0.8765	0.8763	0.9219
Avg.	0.86415	0.86588	0.86415	0.86348	0.91430	0.88605	0.88808	0.88605	0.88570	0.92805
Mdn.	0.86505	0.86715	0.86505	0.86470	0.91471	0.88775	0.89060	0.88775	0.88745	0.92920
Var.	0.00005	0.00005	0.00005	0.00006	0.00002	0.00005	0.00005	0.00005	0.00004	0.00002
S.D.	0.00016	0.00015	0.00016	0.00017	0.00005	0.00014	0.00015	0.00014	0.00013	0.00006

*Fuente.* Elaboración Propia.

### 5.1.5. Análisis de resultados experimentales en ensamblado.

En este punto se realizó comparaciones entre el modelo EfficientNetB6 (debido a que obtuvo los mejores indicadores después de la aplicación de congelamiento de capa por búsqueda binaria) y comparándolo con los 3 modelos ensamblados: *Hard Voting*, *Average Soft Voting* y *Weighted Soft Voting* aplicando algoritmos genéticos presentados en las Tablas 44, 46 y 45. En todos los casos se observan mejoras que van desde el 3.46 % hasta el 4.69 % a nivel de exactitud hasta mejoras desde el 2.16 % hasta 2.94 % a nivel de ROC AUC Score. Debemos tomar en cuenta que a nivel de costo de ejecución, los experimentos indican que implementar *Hard Voting* y *Average Soft Voting* son fáciles de implementar, pero el uso de algoritmos genéticos (con los que se obtuvieron los mejores indicadores) requieren de un mayor tiempo de ejecución para hallar los mejores pesos.

**Tabla 44**

*Resultados experimentales para los indicadores de la hipótesis específica 4 usando Hard Voting*

	Congelamiento con búsqueda binaria (EfficientNetB6)					Ensamblaje por <i>Hard Voting</i>				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.8892	0.8921	0.8892	0.8889	0.9303	0.9170	0.9184	0.9170	0.9167	0.9474
Fold 2	0.8922	0.8932	0.8922	0.8916	0.9319	0.9262	0.9268	0.9262	0.9261	0.9531
Fold 3	0.8863	0.8891	0.8863	0.8860	0.9281	0.9246	0.9253	0.9246	0.9243	0.9521
Fold 4	0.8765	0.8779	0.8765	0.8763	0.9219	0.9148	0.9151	0.9148	0.9145	0.9460
Avg.	0.88605	0.88808	0.88605	0.88570	0.92805	0.92065	0.92140	0.92065	0.92040	0.94962
Mdn.	0.88775	0.89060	0.88775	0.88745	0.92920	0.92080	0.92185	0.92080	0.92050	0.94973
Var.	0.00005	0.00005	0.00005	0.00004	0.00002	0.00003	0.00003	0.00003	0.00003	0.00001
S.D.	0.00014	0.00015	0.00014	0.00013	0.00006	0.00009	0.00009	0.00009	0.00010	0.00004

*Fuente.* Elaboración Propia.

**Tabla 45**

*Resultados experimentales para los indicadores de la hipótesis específica 4 usando Average Soft Voting*

	Congelamiento con búsqueda binaria (EfficientNetB6)					Ensamblaje por <i>Average Soft Voting</i>				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.8892	0.8921	0.8892	0.8889	0.9303	0.9237	0.9246	0.9237	0.9234	0.9514
Fold 2	0.8922	0.8932	0.8922	0.8916	0.9319	0.9337	0.9342	0.9337	0.9337	0.9577
Fold 3	0.8863	0.8891	0.8863	0.8860	0.9281	0.9295	0.9303	0.9295	0.9293	0.9551
Fold 4	0.8765	0.8779	0.8765	0.8763	0.9219	0.9205	0.9210	0.9205	0.9203	0.9499
Avg.	0.88605	0.88808	0.88605	0.88570	0.92805	0.92685	0.92753	0.92685	0.92668	0.95352
Mdn.	0.88775	0.89060	0.88775	0.88745	0.92920	0.92660	0.92745	0.92660	0.92635	0.95328
Var.	0.00005	0.00005	0.00005	0.00004	0.00002	0.00003	0.00003	0.00003	0.00004	0.00001
S.D.	0.00014	0.00015	0.00014	0.00013	0.00006	0.00010	0.00010	0.00010	0.00011	0.00004

*Fuente.* Elaboración Propia.

**Tabla 46**

*Resultados experimentales para los indicadores de la hipótesis específica 4 usando Weighted Soft Voting con algoritmos genéticos*

	Congelamiento con búsqueda binaria (EfficientNetB6)					Ensamblaje por <i>Weighted Soft Voting</i>				
	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score	Exactitud	Precisión	Sensibilidad	F1-score	ROC AUC Score
Fold 1	0.8892	0.8921	0.8892	0.8889	0.9303	0.9306	0.9319	0.9306	0.9303	0.9561
Fold 2	0.8922	0.8932	0.8922	0.8916	0.9319	0.9393	0.9399	0.9393	0.9392	0.9615
Fold 3	0.8863	0.8891	0.8863	0.8860	0.9281	0.9359	0.9365	0.9359	0.9357	0.9593
Fold 4	0.8765	0.8779	0.8765	0.8763	0.9219	0.9259	0.9262	0.9259	0.9257	0.9530
Avg.	0.88605	0.88808	0.88605	0.88570	0.92805	0.93293	0.93363	0.93293	0.93273	0.95747
Mdn.	0.88775	0.89060	0.88775	0.88745	0.92920	0.93325	0.93420	0.93325	0.93300	0.95770
Var.	0.00005	0.00005	0.00005	0.00004	0.00002	0.00003	0.00004	0.00003	0.00004	0.00001
S.D.	0.00014	0.00015	0.00014	0.00013	0.00006	0.00010	0.00011	0.00010	0.00011	0.00004

*Fuente.* Elaboración Propia.

## 5.2. Pruebas de hipótesis.

Como se indicó en los procedimientos para validar la hipótesis general se deberá validar sus 4 hipótesis específicas. Se realizaron las pruebas de validación indicando el valor p por cada indicador aplicando la prueba de T-Student pareada y la mejora de

cada indicador en función a su promedio, finalmente, se considera que para descartar la hipótesis nula el valor p deberá ser menor de 0.05.

### 5.2.1. Hipótesis específica: *data augmentation* y aplicación de pesos.

Al aplicar la prueba de hipótesis en InceptionV3 (Tabla 47) y EfficientNetB6 (Tabla 50), el valor p es menor que 0.05, por tanto podemos descartar la hipótesis nula y validar la hipótesis alternativa.

Al aplicar la prueba de hipótesis en ResNet152V2 (Tabla 48) y DenseNet121 (Tabla 49) algunos de sus indicadores tienen un valor mayor de 0.05, para este caso se referencia el trabajo de Mandrekar (2010) que indica el ROC AUC Score como la medida más confiable al evaluar un clasificador de diagnóstico médico, precisamente los valores que cumplen con descartar la hipótesis nula en estos casos.

**Tabla 47**

*Prueba de hipótesis específica 1 para InceptionV3*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.0493	0.0612	0.0493	0.0614	0.0981
Valor p	0.0161	0.0246	0.0161	0.0167	0.0087

*Fuente.* Elaboración Propia.

**Tabla 48**

*Prueba de hipótesis específica 1 para ResNet152V2*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.0061	0.0233	0.0061	0.0207	0.0848
Valor p	<b>0.1221</b>	0.0174	<b>0.1221</b>	0.0252	0.0043

*Fuente.* Elaboración Propia.

**Tabla 49**

*Prueba de hipótesis específica 1 para DenseNet121*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.0102	0.0245	0.0102	0.0188	0.0677
Valor p	<b>0.3804</b>	<b>0.0955</b>	<b>0.3804</b>	<b>0.1591</b>	0.0085

*Fuente.* Elaboración Propia.

**Tabla 50***Prueba de hipótesis específica 1 para EfficientNetB6*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.0695	0.0763	0.0695	0.0750	0.0989
Valor p	0.00004	0.0001	0.00004	0.00002	0.0005

*Fuente.* Elaboración Propia.**5.2.2. Hipótesis específica: *transfer learning* .**

Al aplicar *transfer learning* todos los indicadores aumentan de forma significativa con un valor p inferior a 0.05 (Tablas 51, 52, 53 y 54), por tanto se descarta la hipótesis nula y se valida la hipótesis alternativa. Algo a agregar es que el uso de *transfer learning* es perfecta cuando se tiene una menor cantidad de recursos en hardware, recordar que a excepción de EfficientNetB6 que tuvo 5 épocas de entrenamiento (y por tanto la mejora a nivel de indicadores es mayor), en 10 épocas no se pudo superar los indicadores al aplicar estos pesos preentrenados.

**Tabla 51***Prueba de hipótesis específica 2 para InceptionV3*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.2425	0.1987	0.2425	0.2395	0.1424
Valor p	0.00010	0.00002	0.00010	0.00003	0.00009

*Fuente.* Elaboración Propia.**Tabla 52***Prueba de hipótesis específica 2 para ResNet152V2*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.2507	0.2070	0.2507	0.2442	0.1501
Valor p	0.00102	0.00013	0.00102	0.00088	0.00068

*Fuente.* Elaboración Propia.**Tabla 53***Prueba de hipótesis específica 2 para DenseNet121*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.2953	0.2249	0.2953	0.2817	0.1728
Valor p	0.00039	0.00119	0.00039	0.00040	0.00045

*Fuente.* Elaboración Propia.

**Tabla 54***Prueba de hipótesis específica 2 para EfficientNetB6*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.4463	0.3804	0.4463	0.4394	0.2622
Valor p	0.00011	0.00010	0.00011	0.00019	0.00009

*Fuente.* Elaboración Propia.**5.2.3. Hipótesis específica: congelamiento de capa.**

Al aplicar el congelamiento de capa se consiguen resultados que descartan la hipótesis nula para ResNet152V2 (Tabla 57), DenseNet121 (Tabla 58) y EfficientNetB6 (Tabla 59) debido a que los valores p de cada indicador es menor de 0.05.

Algo a considerarse es que no se puede descartar la hipótesis nula para los casos de InceptionV3 (Tabla 55) y ResNet152V2 aplicando búsqueda binaria (Tabla 56), esto posiblemente se deba a la cantidad de épocas de entrenamiento (10) y es posible que se deba incrementar este número para obtener mejores resultados.

**Tabla 55***Prueba de hipótesis específica 3 para InceptionV3*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.0029	0.0029	0.0029	0.0029	0.0011
Valor p	<b>0.8119</b>	<b>0.8172</b>	<b>0.8119</b>	<b>0.8150</b>	<b>0.8996</b>

*Fuente.* Elaboración Propia.**Tabla 56***Prueba de hipótesis específica 3 para ResNet152V2 aplicando búsqueda binaria*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.0143	0.0096	0.0143	0.0146	0.0076
Valor p	<b>0.0744</b>	<b>0.1442</b>	<b>0.0744</b>	<b>0.0718</b>	<b>0.0874</b>

*Fuente.* Elaboración Propia.**Tabla 57***Prueba de hipótesis específica 3 para ResNet152V2 aplicando algoritmos genéticos*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.0318	0.0324	0.0318	0.0327	0.0204
Valor p	0.0190	0.0216	0.0190	0.0214	0.0193

*Fuente.* Elaboración Propia.

**Tabla 58***Prueba de hipótesis específica 3 para DenseNet121*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.0360	0.0318	0.0360	0.0366	0.0230
Valor p	0.0384	0.0328	0.0384	0.0390	0.0432

*Fuente.* Elaboración Propia.**Tabla 59***Prueba de hipótesis específica 3 para EfficientNetB6*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.0219	0.0222	0.0219	0.0222	0.0138
Valor p	0.0248	0.0278	0.0248	0.0261	0.0237

*Fuente.* Elaboración Propia.

#### 5.2.4. Hipótesis específica: ensamblado.

En todos los tipos de ensamblado se demuestra que todos los indicadores aumentan de forma significativa con un valor p menor de 0.05, por tanto se descarta la hipótesis nula y se valida la hipótesis alternativa (Tablas 60, 61 y 62). Tener en cuenta que la comparación se realizó con el modelo EfficientNetB6 que tenía indicadores más elevados que los otros 3 modelos.

**Tabla 60***Prueba de hipótesis específica 4 para Hard Voting*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.0346	0.0333	0.0346	0.0347	0.0216
Valor p	0.0008	0.0009	0.0008	0.0008	0.0010

*Fuente.* Elaboración Propia.**Tabla 61***Prueba de hipótesis específica 4 para Average Soft Voting*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.0408	0.0395	0.0408	0.0410	0.0255
Valor p	0.0003	0.0005	0.0003	0.0003	0.0005

*Fuente.* Elaboración Propia.

**Tabla 62**

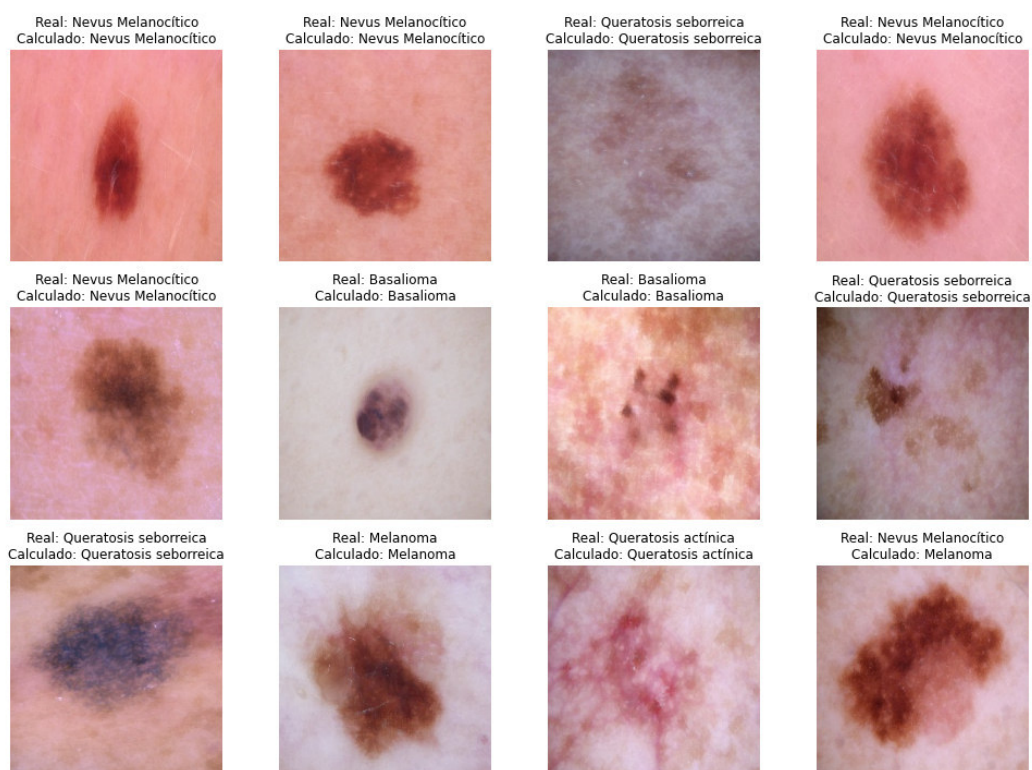
*Prueba de hipótesis específica 4 para Weighted Soft Voting aplicando algoritmos genéticos*

	<b>Exactitud</b>	<b>Precisión</b>	<b>Sensibilidad</b>	<b>F1-score</b>	<b>ROC AUC Score</b>
Mejora	0.0469	0.0456	0.0469	0.0470	0.0294
Valor p	0.0001	0.0002	0.0001	0.0002	0.0002

*Fuente.* Elaboración Propia.

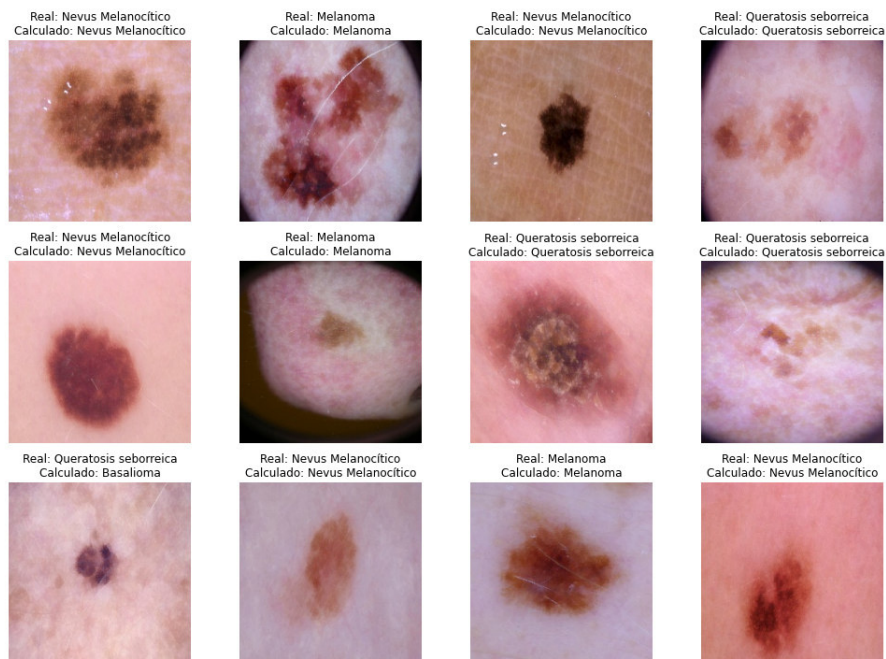
### 5.3. Presentación de resultados.

A continuación, se presentan los resultados del uso del modelo ensamblado, las imágenes fueron seleccionadas de forma aleatoria y cada una de ellas está asociado a su diagnóstico real y calculado. Como se puede observar en las Figura 43, 44 y 45 el cálculo es correcto en la mayoría de casos.

**Figura 43**

*Presentación de resultados por diagnóstico (selección aleatoria 1)*

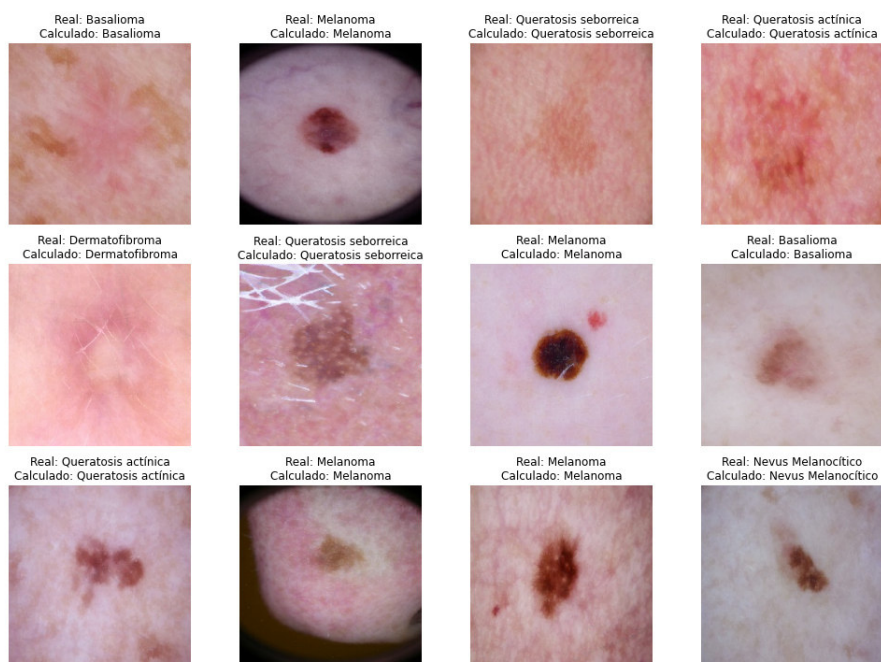
*Fuente.* Elaboración Propia.



**Figura 44**

*Presentación de resultados por diagnóstico (selección aleatoria 2)*

*Fuente. Elaboración Propia.*



**Figura 45**

*Presentación de resultados por diagnóstico (selección aleatoria 3)*

*Fuente. Elaboración Propia.*



# Capítulo VI

## Conclusiones y Recomendaciones

### 6.1. Conclusiones

#### 6.1.1. Conclusión general

El uso de modelos DCNN mediante el *transfer learning*, congelamiento de capa óptima por medio de la búsqueda binaria y ensamblado por medio de algoritmos genéticos mejora significativamente el desempeño en la tarea de clasificación de lesiones en imágenes dermatoscópicas.

La mejora del desempeño solo en términos de exactitud llega hasta un 3.18 % sin la aplicación del ensamblado y un 4.69 % más con el ensamblaje.

#### 6.1.2. Conclusiones específicas

- El aumento de datos y aplicación de pesos garantiza una mejor desempeño en la clasificación del diagnóstico, el aumento de datos proporciona una mayor riqueza de datos cuando se cuentan con pocos datos para una clase específica y la aplicación de pesos ayuda en el ahorro de recursos de hardware.
  - En el caso de ResNet152V2 y DenseNet121 es posible que se pueda obviar este procedimiento, se recomienda realizar pruebas con otros *datasets*.
- El uso de *transfer learning* mejora significativamente el desempeño de los modelos DCNN en la clasificación de lesiones en imágenes dermatoscópicas.
- El congelamiento de capa por medio de la búsqueda binaria mejora significativamente el desempeño en la clasificación del diagnóstico de lesiones en imágenes dermatoscópicas.
  - En el caso de InceptionV3 se observa que el uso de congelamiento por búsqueda binaria y algoritmos genéticos no aumenta de forma significativa

este desempeño, se recomienda realizar más épocas de entrenamiento y usar una población inicial mayor en caso de los algoritmos genéticos para tener mejores indicadores de diagnóstico, la contraparte será un uso mayor de recursos de hardware para obtenerlos.

- En el caso de ResNet152V2 se demuestra que los algoritmos genéticos tienen un mejor resultado que la búsqueda binaria, se recomienda realizar más épocas de entrenamiento y probar con otros *datasets*.
- El ensamblaje por medio de algoritmos genéticos mejora significativamente el desempeño en la clasificación de lesiones en imágenes dermatoscópicas, el hecho de contar varios modelos ayuda a cubrir los defectos que tengan los otros.

## 6.2. Recomendaciones

- Para futuros trabajos se recomienda proponer métodos más potentes para el congelamiento en capa óptima usado en la búsqueda binaria, es posible verificando los resultados experimentales que algunas iteraciones pueden ser omitidas usando un nuevo método con complejidad algorítmica menor que  $O(\log n)$ .
- Se recomienda ampliar este estudio de congelamiento de capas a otros *datasets* y no solo enfocarlo en el uso de clasificación de imágenes, sino en otros campos como el procesamiento de lenguaje natural.
- Se recomienda continuar con el estudio de métodos de congelamiento de capa en la clasificación de imágenes cuando la resolución cambia. En el presente trabajo de investigación se usó una resolución de  $224 \times 224$  píxeles, es muy probable que los resultados cambien si se usa otra resolución.
- Es posible que exista un método híbrido entre la búsqueda binaria y los algoritmos genéticos, reduciendo y ahorrando recursos de hardware, uno de los problemas de los algoritmos genéticos y las DCNN es la gran cantidad de capas que maneja y hallar la combinación correcta donde aplicar el congelamiento se incrementa en función al número de estas, pero si se aplicase una congelación por búsqueda binaria y luego algoritmos genéticos esto podría reducirse.
- El uso de algoritmos genéticos en el ensamblaje para hallar los pesos óptimos por cada modelo consume demasiados recursos de hardware, se recomienda investigar si existe una solución más económica y si es posible que este problema de optimización pueda ser solucionado por medio de la programación lineal.

## Referencias bibliográficas

- Alagu, S., y Bagan, K. B. (2021). Skin cancer classification in dermoscopy images using convolutional neural network. En *Aip conference proceedings* (Vol. 2336, p. 040013). doi: <https://doi.org/10.1063/5.0045757>
- Al-Masni, M. A., Kim, D.-H., y Kim, T.-S. (2020). Multiple skin lesions diagnostics via integrated deep convolutional networks for segmentation and classification. *Computer methods and programs in biomedicine*, 190, 105351. doi: <https://doi.org/10.1016/j.cmpb.2020.105351>
- Arkah, Z. M., Al-Dulaimi, D. S., y Khekan, A. R. (2021). Big transfer learning for automated skin cancer classification. *Indonesian Journal of Electrical Engineering and Computer Science*, 23(3), 1611–1619. doi: <https://doi.org/10.11591/ijeecs.v23.i3.pp1611-1619>
- Beasley, D., Bull, D. R., y Martin, R. R. (1993). An overview of genetic algorithms: Part 1, fundamentals. *University computing*, 15(2), 56–69.
- Cancer today*. (s.f.-a). Descargado 16/07/2021, de <https://t.ly/pzWY>
- Cancer today*. (s.f.-b). Descargado 16/07/2021, de <https://t.ly/Jbqa>
- Cancer today*. (s.f.-c). Descargado 16/07/2021, de <https://t.ly/woMs>
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., y Stein, C. (2009). *Introduction to algorithms*. MIT press.
- Coronado Pérez, R. R. (2018). *Reconocimiento de patrones en imágenes no dermatoscópicas para la detección de enfermedades malignas en la piel, utilizando redes neuronales convolutivas y autocodificadores*. Universidad Nacional de San Agustín de Arequipa.
- Creswell, J. W., y Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.
- de Lima Mendes, R., da Silva Alves, A. H., de Souza Gomes, M., Bertarini, P. L. L., y do Amaral, L. R. (2021). Many layer transfer learning genetic algorithm (mltlda):

- a new evolutionary transfer learning approach applied to pneumonia classification. En *2021 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2476–2482).
- Dietterich, T. G. (2000). Ensemble methods in machine learning. En *International workshop on multiple classifier systems* (pp. 1–15). doi: [https://doi.org/10.1007/3-540-45014-9\\_1](https://doi.org/10.1007/3-540-45014-9_1)
- Ferreras Extremo, A. (2021). *Estudio de algoritmos de redes neuronales convolucionales en dataset de imágenes médicas*. Universidad de Valladolid. Escuela Técnica Superior de Ingenieros de Telecomunicación.
- Gavali, P., y Banu, J. S. (2019). Deep convolutional neural network for image classification on cuda platform. En *Deep learning and parallel computing environment for bioengineering systems* (pp. 99–122). Elsevier. doi: <https://doi.org/10.1016/b978-0-12-816718-2.00013-0>
- Gillis, A. S. (2022, Abril). *What is data splitting and why is it important?* TechTarget. Descargado de <https://t.ly/cQm4>
- Grove, R., y Green, R. (2020). Melanoma and nevi classification using convolution neural networks. En *2020 35th international conference on image and vision computing new zealand (ivcnz)* (pp. 1–6). doi: <https://doi.org/10.1109/ivcnz51579.2020.9290736>
- He, K., Zhang, X., Ren, S., y Sun, J. (2016). Deep residual learning for image recognition. En *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).
- Herrera, H., y Herrera, J. (2022). Optimal layer selection on deep convolutional neural networks using backward freezing and binary search. En *Annual international conference on information management and big data* (pp. 175–190). Springer Nature Switzerland. doi: [https://doi.org/10.1007/978-3-031-35445-8\\_13](https://doi.org/10.1007/978-3-031-35445-8_13)
- Huanca-Huirse, N. L., Roque-Roque, J. S., Laurel-Vargas, V. N., y Quispe-Sancho, A. W. (2020). Percepción del riesgo de cáncer de piel en un hospital de altura. *Revista de la Facultad de Medicina Humana*, 20(2), 222–226. doi: <https://doi.org/10.25176/rfmh.v20i2.2895>
- Huang, G., Liu, Z., Van Der Maaten, L., y Weinberger, K. Q. (2017). Densely connected convolutional networks. En *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).
- Instituto Nacional del Cáncer. (2011, Feb). *Diccionario de cáncer del nci*. Descargado de <https://www.cancer.gov/espanol/publicaciones/>

diccionarios/diccionario-cancer/def/diagnostico (Accedido el 07 de Julio, 2023)

- Jain, S., Singhanian, U., Tripathy, B., Nasr, E. A., Aboudaif, M. K., y Kamrani, A. K. (2021). Deep learning-based transfer learning for classification of skin cancer. *Sensors*, 21(23), 8142. doi: <https://doi.org/10.3390/s21238142>
- Jojoa Acosta, M. F., Caballero Tovar, L. Y., Garcia-Zapirain, M. B., y Percybrooks, W. S. (2021). Melanoma diagnosis using deep learning techniques on dermatoscopic images. *BMC Medical Imaging*, 21(1), 1–11. doi: <https://doi.org/10.1186/s12880-020-00534-8>
- Karki, S., Kulkarni, P., y Stranieri, A. (2021). Melanoma classification using efficientnets and ensemble of models with different input resolution. En *2021 australian computer science week multiconference* (pp. 1–5). doi: <https://doi.org/10.1145/3437378.3437396>
- Kawasaki, Y., Uga, H., Kagiwada, S., y Iyatomi, H. (2015). Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. En *International symposium on visual computing* (pp. 638–645).
- Krizhevsky, A., Sutskever, I., y Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25. doi: <https://doi.org/10.1145/3065386>
- Kukar, M., y Kononenko, I. (2002). Reliable classifications with machine learning. En *European conference on machine learning* (pp. 219–231).
- Kumar, V. A., Mishra, V., y Arora, M. (2021). Convolutional neural networks for malignant and benign cell classification using dermatoscopic images. En *2021 third international conference on intelligent communication technologies and virtual mobile networks (icicv)* (pp. 1040–1044). doi: <https://doi.org/10.1109/icicv50876.2021.9388605>
- LeCun, Y., Bengio, Y., y Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436–444. doi: <https://doi.org/10.1038/nature14539>
- Lee, S.-J., Chen, T., Yu, L., y Lai, C.-H. (2018). Image classification based on the boost convolutional neural network. *IEEE Access*, 6, 12755–12768. doi: <https://doi.org/10.1109/access.2018.2796722>
- Li, B., y Rangarajan, S. (2022). A conceptual study of transfer learning with linear models for data-driven property prediction. *Computers & Chemical Engineering*, 157, 107599. doi: <https://doi.org/10.1016/j.compchemeng.2021.107599>

- Li, H., Pan, Y., Zhao, J., y Zhang, L. (2021). Skin disease diagnosis with deep learning: a review. *Neurocomputing*, 464, 364–393. doi: <https://doi.org/10.1016/j.neucom.2021.08.096>
- Mahbod, A., Schaefer, G., Wang, C., Dorffner, G., Ecker, R., y Ellinger, I. (2020). Transfer learning using a multi-scale and multi-network ensemble for skin lesion classification. *Computer methods and programs in biomedicine*, 193, 105475. doi: <https://doi.org/10.1016/j.cmpb.2020.105475>
- Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, 5(9), 1315–1316. doi: <https://doi.org/10.1097/jto.0b013e3181ec173d>
- Melapp: Aplicativo móvil que detecta el cáncer de piel.* (2019, Abril). Descargado 16/07/2021, de <https://t.ly/pDGJ>
- Moataz, L., Salama, G. I., y Abd Elazeem, M. H. (2021). Skin cancer diseases classification using deep convolutional neural network with transfer learning model. En *Journal of physics: Conference series* (Vol. 2128, p. 012013). doi: <https://doi.org/10.1088/1742-6596/2128/1/012013>
- Nagae, S., Kawai, S., y Nobuhara, H. (2020). Transfer learning layer selection using genetic algorithm. En *2020 IEEE congress on evolutionary computation (cec)* (pp. 1–6). doi: <https://doi.org/10.1109/cec48606.2020.9185501>
- Palacios-Martínez, D., y Díaz-Alonso, R. (2017). Dermatoscopia para principiantes (i): características generales. *SEMERGEN-Medicina de Familia*, 43(3), 216–221. doi: <https://doi.org/10.1016/j.semerg.2015.11.009>
- Pan, S. J., y Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10), 1345–1359.
- Picazo Montoya, O. (2018). *Redes neuronales convolucionales profundas para el reconocimiento de emociones en imágenes.*
- Potyraj, E. (2021). *4 ways to improve class imbalance for image data.* Descargado 09/07/2023, de <https://towardsdatascience.com/4-ways-to-improve-class-imbalance-for-image-data-9adec8f390f1>
- Pratiwi, R. A., Nurmaini, S., Rini, D. P., Rachmatullah, M. N., y Darmawahyuni, A. (2021). Deep ensemble learning for skin lesions classification with convolutional neural network. *IAES International Journal of Artificial Intelligence*, 10(3), 563. doi: <https://doi.org/10.11591/ijai.v10.i3.pp563-570>

- Python Genetic Algorithm Library. (2020). *geneticalgorithm*. Python Package Index. Descargado de <https://pypi.org/project/geneticalgorithm/> (Accedido el 26 de Julio, 2022)
- Rahman, Z., Hossain, M. S., Islam, M. R., Hasan, M. M., y Hridhee, R. A. (2021). An approach for multiclass skin lesion classification based on ensemble learning. *Informatics in Medicine Unlocked*, 25, 100659. doi: <https://doi.org/10.1016/j.imu.2021.100659>
- Raschka, S. (2021). *Ensemblevoteclassifier - mlxtend*. Descargado de <https://t.ly/lXIii>
- Saha, S. (2018, Dic). *A comprehensive guide to convolutional neural networks - the eli5 way*. Towards Data Science. Descargado 16/07/2021, de <https://t.ly/zM6y>
- Science, D. (1970, Ene). *Confusion matrix*. Descargado 13/12/2021, de <https://t.ly/CVXG>
- Shah, D. (2022). *Data augmentation guide*. Descargado 09/07/2023, de <https://www.v7labs.com/blog/data-augmentation-guide>
- Shanmugamani, R. (2018). *Deep learning for computer vision: Expert techniques to train advanced neural networks using tensorflow and keras*. Packt Publishing Ltd.
- Shawon, M., Abedin, K. F., Majumder, A., Mahmud, A., y Mishu, M. M. C. (2021). Identification of risk of occurring skin cancer (melanoma) using convolutional neural network (cnn). *AIUB Journal of Science and Engineering (AJSE)*, 20(2), 47–51. doi: <https://doi.org/10.53799/ajse.v20i2.140>
- Sucar, L. E., y Gómez, G. (2011). *Visión computacional. Instituto Nacional de Astrofísica, Óptica y Electrónica. Puebla, México*.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2015). Going deeper with convolutions. En *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).
- Tan, M., y Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. En *International conference on machine learning* (pp. 6105–6114).
- The International Agency for Research on Cancer (IARC). (s.f.). *Global cancer observatory*. Descargado 16/07/2021, de <https://gco.iarc.fr/>

- Tschandl, P. (2018). *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions*. Harvard Dataverse. doi: <https://doi.org/10.7910/DVN/DBW86T>
- Villa-Pulgarin, J. P., Ruales-Torres, A. A., y y otros, A.-G. (2022). Optimized convolutional neural network models for skin lesion classification. *Comput. Mater. Contin.*, 70(2), 2131–2148. doi: <https://doi.org/10.32604/cmc.2022.019529>
- Wada, A., Tsuruta, K., Irie, R., Kamagata, K., Maekawa, T., Fujita, S., ... others (2019). Differentiating alzheimer's disease from dementia with lewy bodies using a deep learning technique based on structural brain connectivity. *Magnetic Resonance in Medical Sciences*, 18(3), 219.



## **Anexos**

**Tabla 63**

*Recursos experimentales: datasets y código fuente para su uso libre*

URL	Descripción
<i>Datasets</i>	
<a href="https://www.kaggle.com/datasets/henryherrera007/ham10k4folds">https://www.kaggle.com/datasets/henryherrera007/ham10k4folds</a>	Dataset generado para 4 folds con data aumentada.
<a href="https://www.kaggle.com/datasets/henryherrera007/ham10k-4folds-nda">https://www.kaggle.com/datasets/henryherrera007/ham10k-4folds-nda</a>	Dataset generado para 4 folds sin data aumentada.
<a href="https://www.kaggle.com/datasets/henryherrera007/resnetv152-cv-bl">https://www.kaggle.com/datasets/henryherrera007/resnetv152-cv-bl</a>	Modelos Generados usando el método de búsqueda binaria para InceptionV3 (mejor resultado).
<a href="https://www.kaggle.com/datasets/henryherrera007/dn121-cv-bl">https://www.kaggle.com/datasets/henryherrera007/dn121-cv-bl</a>	Modelos Generados usando el método de búsqueda binaria para ResNet152V2 (mejor resultado).
<a href="https://www.kaggle.com/datasets/henryherrera007/iv3-cv-bl">https://www.kaggle.com/datasets/henryherrera007/iv3-cv-bl</a>	Modelos Generados usando el método de búsqueda binaria para DenseNet121 (mejor resultado).
<a href="https://www.kaggle.com/datasets/henryherrera007/e6-cv-bl">https://www.kaggle.com/datasets/henryherrera007/e6-cv-bl</a>	Modelos Generados usando el método de búsqueda binaria para EfficientNetB6 (mejor resultado).
<b>Código Fuente</b>	
<a href="https://www.kaggle.com/code/henryherrera007/ham10000-dataset-generator-n-folds">https://www.kaggle.com/code/henryherrera007/ham10000-dataset-generator-n-folds</a>	Generador de datasets para HAM10000 para N folds con data aumentada, el número de folds y la dimensión de imagen es configurable.
<a href="https://www.kaggle.com/code/henryherrera007/ham10000-dataset-generator-n-folds-nda">https://www.kaggle.com/code/henryherrera007/ham10000-dataset-generator-n-folds-nda</a>	Generador de datasets para HAM10000 para N folds sin data aumentada, el número de folds y la dimensión de imagen es configurable.
<a href="https://www.kaggle.com/code/henryherrera007/ham10000-iv3-cv-nda">https://www.kaggle.com/code/henryherrera007/ham10000-iv3-cv-nda</a>	Clasificación de lesiones sin usar data aumentada para InceptionV3.
<a href="https://www.kaggle.com/code/henryherrera007/ham10000-resnetv152-cv-nda">https://www.kaggle.com/code/henryherrera007/ham10000-resnetv152-cv-nda</a>	Clasificación de lesiones sin usar data aumentada para ResNet152V2.
<a href="https://www.kaggle.com/code/henryherrera007/ham10000-iv3-dn121-nda">https://www.kaggle.com/code/henryherrera007/ham10000-iv3-dn121-nda</a>	Clasificación de lesiones sin usar data aumentada para DenseNet121.
<a href="https://www.kaggle.com/code/henryherrera007/ham10000-e6-cv-nda">https://www.kaggle.com/code/henryherrera007/ham10000-e6-cv-nda</a>	Clasificación de lesiones sin usar data aumentada para EfficientNetB6.
<a href="https://www.kaggle.com/code/henryherrera007/ham10000-iv3-cv-no-tl">https://www.kaggle.com/code/henryherrera007/ham10000-iv3-cv-no-tl</a>	Clasificación de lesiones sin usar transfer learning para InceptionV3.
<a href="https://www.kaggle.com/code/henryherrera007/ham10000-resnetv152-cv-no-tl">https://www.kaggle.com/code/henryherrera007/ham10000-resnetv152-cv-no-tl</a>	Clasificación de lesiones sin usar transfer learning para ResNet152V2.
<a href="https://www.kaggle.com/code/henryherrera007/ham10000-dn121-cv-no-tl">https://www.kaggle.com/code/henryherrera007/ham10000-dn121-cv-no-tl</a>	Clasificación de lesiones sin usar transfer learning para DenseNet121.
<a href="https://www.kaggle.com/code/henryherrera007/ham10000-e6-cv-no-tl">https://www.kaggle.com/code/henryherrera007/ham10000-e6-cv-no-tl</a>	Clasificación de lesiones sin usar transfer learning para EfficientNetB6.
<a href="https://www.kaggle.com/code/hh Herrera007/binary-search">https://www.kaggle.com/code/hh Herrera007/binary-search</a>	Congelamiento por búsqueda binaria, versión simplificada.
<a href="https://www.kaggle.com/code/henryherrera007/ham10000-iv3-cv">https://www.kaggle.com/code/henryherrera007/ham10000-iv3-cv</a>	Congelamiento por búsqueda binaria para InceptionV3 (resultados por capa en sus versiones).
<a href="https://www.kaggle.com/code/hh Herrera007/ham10000-resnetv152-cv">https://www.kaggle.com/code/hh Herrera007/ham10000-resnetv152-cv</a>	Congelamiento por búsqueda binaria para ResNet152V2 (resultados por capa en sus versiones).
<a href="https://www.kaggle.com/code/henryherrera007/ham10000-dn121-cv">https://www.kaggle.com/code/henryherrera007/ham10000-dn121-cv</a>	Congelamiento por búsqueda binaria para DenseNet121 (resultados por capa en sus versiones).
<a href="https://www.kaggle.com/code/hh Herrera007/ham10000-e6-cv">https://www.kaggle.com/code/hh Herrera007/ham10000-e6-cv</a>	Congelamiento por búsqueda binaria para EfficientNetB6 (resultados por capa en sus versiones).
<a href="https://www.kaggle.com/code/hh Herrera007/ham10000-iv3-cv-ga">https://www.kaggle.com/code/hh Herrera007/ham10000-iv3-cv-ga</a>	Congelamiento por algoritmos genéticos para InceptionV3.
<a href="https://www.kaggle.com/code/hh Herrera007/ham10000-r152v2-cv-ga">https://www.kaggle.com/code/hh Herrera007/ham10000-r152v2-cv-ga</a>	Congelamiento por algoritmos genéticos para ResNet152V2 - sin presentación final de resultados.
<a href="https://www.kaggle.com/code/hh Herrera007/ham10000-r152v2-cv-ga-with-previous-results/">https://www.kaggle.com/code/hh Herrera007/ham10000-r152v2-cv-ga-with-previous-results/</a>	Congelamiento por algoritmos genéticos para ResNet152V2 - con presentación final de resultados (dataset generado a partir de los resultados del código anterior).
<a href="https://www.kaggle.com/code/hh Herrera007/ham10000-dn121-cv-ga">https://www.kaggle.com/code/hh Herrera007/ham10000-dn121-cv-ga</a>	Congelamiento por algoritmos genéticos para DenseNet121.
<a href="https://www.kaggle.com/code/hh Herrera007/ham10000-e6-cv-ga">https://www.kaggle.com/code/hh Herrera007/ham10000-e6-cv-ga</a>	Congelamiento por algoritmos genéticos para EfficientNetB6 - sin presentación final de resultados.
<a href="https://www.kaggle.com/code/hh Herrera007/ham10000-e6-cv-ga-with-previous-results">https://www.kaggle.com/code/hh Herrera007/ham10000-e6-cv-ga-with-previous-results</a>	Congelamiento por algoritmos genéticos para EfficientNetB6 - con presentación final de resultados (dataset generado a partir de los resultados del código anterior).
<a href="https://www.kaggle.com/code/hh Herrera007/model-ensembling-cv-fold-1">https://www.kaggle.com/code/hh Herrera007/model-ensembling-cv-fold-1</a>	Métodos de ensamblaje para el Fold 1.
<a href="https://www.kaggle.com/code/hh Herrera007/model-ensembling-cv-fold-2">https://www.kaggle.com/code/hh Herrera007/model-ensembling-cv-fold-2</a>	Métodos de ensamblaje para el Fold 2.
<a href="https://www.kaggle.com/code/henryherrera007/model-ensembling-cv-fold-3">https://www.kaggle.com/code/henryherrera007/model-ensembling-cv-fold-3</a>	Métodos de ensamblaje para el Fold 3.
<a href="https://www.kaggle.com/code/henryherrera007/model-ensembling-cv-fold-4">https://www.kaggle.com/code/henryherrera007/model-ensembling-cv-fold-4</a>	Métodos de ensamblaje para el Fold 4.

*Fuente. Elaboración Propia.*



## Optimal Layer Selection on Deep Convolutional Neural Networks Using Backward Freezing and Binary Search

Henry Miguel Herrera Del Aguila<sup>(✉)</sup>  and José Alfredo Herrera Quispe 

Universidad Nacional Mayor de San Marcos, Cercado de Lima 15081, Peru  
[henry.herrera2@unmsm.edu.pe](mailto:henry.herrera2@unmsm.edu.pe)

**Abstract.** Transfer Learning in Deep Convolutional Neural Networks is a highly used method for image classification, the performance depends on the selection of the layers to be frozen; the search for the optimal layer where the backward freezing will be applied becomes increasingly complicated as the number of layers increases. We found an approximation of the optimal layer to apply backward freezing using a modified Binary Search method. We experimented with InceptionV3, Xception, DenseNet121, NasNetMobile, and ResNet50; the case of study is the classification of melanomas of benign and malignant class from Kaggle. We use partitions with four-folds and measure the performance with the ROC AUC score. Our results show an improvement in the ROC AUC score for ResNet50 and no significant change for InceptionV3. Code: <https://www.kaggle.com/code/hherrera007/binary-search>.

[AQ1](#)

[AQ2](#)

### Figura 46

*Artículo científico publicado.*

*Fuente.* Tomado del trabajo de Herrera y Herrera (2022).



**Universidad Nacional Mayor de  
SAN MARCOS**  
Universidad del Perú. Decana de América

**SIMBig22**

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
VICEDECANATO DE INVESTIGACIÓN Y POSGRADO

*Certificado*

Nº027-2022-VICEIP-FISI/UNMSM

Otorgado a:

**Henry Miguel Herrera del Águila**

En mérito a su participación como:

**Autor**

En la:

**9th International Conference on Information Management and Big Data**

Evento académico internacional, desarrollado del 16 al 18 de noviembre 2022, registrado con COD: XI2R-22110224.

Auditorio Rosa Alarco Larrabure – Biblioteca Central de la Universidad Nacional Mayor de San Marcos.

Lima-Perú

Lima, diciembre 2022



**Dra. Nora Bertha La Serna Palomino**  
Vicedecana de Investigación y Posgrado

**Figura 47**

*Diploma de participación del SIMBIG 2022.*

*Fuente. SIMBIG 2022.*



Universidad Nacional Mayor de  
**SAN MARCOS**  
Universidad del Perú. Decana de América



Firmado digitalmente por:  
UNMSM  
Motivo: Servidor de  
Agente Automatizado  
Fecha: 23/01/2023 14:53:05-0600

N.º 0000040-2023-D-FISI/UNMSM

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**III SEMINARIO INTERNACIONAL DE CIENCIA DE DATOS APLICADA A LA  
SOCIEDAD**

**CERTIFICADO**

Otorgado a:

**HENRY MIGUEL HERRERA DEL AGUILA**

En mérito a su participación como:

**PONENTE**

Evento organizado por el Decanato, Instituto de Investigación, Grupo de Investigación iTData y Vicedecanato Académico, realizado el 29 de diciembre de 2022 bajo la modalidad virtual, con una duración de 10 horas académicas, según N.º 36 Consejo de Facultad Virtual (Sesión Ordinaria Nº 22 - Miércoles 30 de noviembre de 2022).

Ciudad Universitaria, 23 de enero de 2023.

Dr. CARLOS EDMUNDO NAVARRO DEPAZ  
DECANO



Esta es una copia auténtica imprimible de un documento electrónico archivado por la Universidad Nacional Mayor de San Marcos, aplicando lo dispuesto por el Art. 25 de D.S. 070-2013-PCM y la Tercera Disposición Complementaria Final del D.S. 026-2016-PCM. Su autenticidad e integridad pueden ser contrastadas a través de la siguiente dirección web <https://tramiteonline.unmsm.edu.pe/sgdfd/mcc/verifica/unmsm>, ingresando el número 0000040-2023-D-FISI/UNMSM y código de verificación 2308CVG.

**Figura 48**

*Diploma de participación del SICDAS 2022.*

*Fuente. SICDAS 2022.*