

SEP

TECNM

INSTITUTO TECNOLÓGICO DE TIJUANA
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN



Trabajo de Tesis:

**DETECCIÓN AUTOMÁTICA DE RETINOPATÍA DIABÉTICA POR
MEDIO DE UN NUEVO MODELO DE REDES NEURONALES
CONVOLUCIONALES**

Presentado por el alumno:

RODRIGO CORDERO MARTÍNEZ

Grado al que se aspira:

MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

Directora de tesis:

DRA. DANIELA ADRIANA SÁNCHEZ VIZCARRA

Codirectora de tesis:

DRA. ELBA PATRICIA MELIN OLMEDA

Lugar y fecha:

TIJUANA, BAJA CALIFORNIA. MARZO/2022



Instituto Tecnológico de Tijuana

Tijuana, Baja California,

05/abril/2022

OFICIO No. 041/DEPI/2022

Asunto: **Autorización de Impresión de Tesis**

MARIBEL GUERRERO LUIS
JEFA DEL DEPARTAMENTO DE SERVICIOS ESCOLARES
PRESENTE

En lo referente al trabajo de tesis, "Detección automática de retinopatía diabética por medio de un nuevo modelo de redes neuronales convolucionales". Presentado por C. **Rodrigo Cordero Martínez**, alumno de la Maestría en Ciencias de la Computación con numero de control **M20210001**; informo a usted que a solicitud del comité de tutorial, tengo a bien **Autorizar la impresión de Tesis**, atendiendo las disposiciones de los Lineamientos para la Operación de Estudios de Posgrado del Tecnológico Nacional de México.

Sin más por el momento le envié un cordial saludo.

ATENTAMENTE

Excelencia en Educación Tecnológica



INSTITUTO TECNOLÓGICO DE TIJUANA
DIVISIÓN DE ESTUDIOS DE POSGRADO
E INVESTIGACIÓN

GUADALUPE HERNÁNDEZ ESCOBEDO
JEFE DE DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

ccp. Archivo

GHE/lap



Calzada del Tecnológico S/N Esq. Castillo de Chapultepec y calle Cuauhtemotzin,
Fracc. Tomás Aquino C.P.22414 Tijuana, Baja California. Tel. 01 (664) 6078400
Est. 101

e-mail: dir_tijuana@tecnm.mx | tecnm.mx | Tijuana.tecnm.mx



2022 Flores
Año de **Magón**
PRECURSOR DE LA REVOLUCIÓN MEXICANA



Tijuana, B.C., 01 de Abril de 2022
Asunto: Se autoriza impresión de Trabajo de Tesis

C. Dr. Guadalupe Hernández Escobedo
Jefe de División de Estudios de Posgrado e Inv.
Presente.

En lo referente al trabajo de tesis escrito, con título "**Detección automática de retinopatía diabética por medio de un nuevo modelo de redes neuronales convolucionales**", presentado por el **C. CORDERO MARTINEZ RODRIGO** alumno de la Maestría en Ciencias de la Computación con número de control **M20210001**, informamos a usted que se autoriza el escrito de tesis y se aprueba en todas sus partes, en virtud de reunir los requisitos de un trabajo de grado de maestría y a la vez se autoriza al interesado para que proceda de inmediato a la impresión del mismo y a presentar su examen de grado, ya que cumple con todos los requisitos.

ATENTAMENTE

DRA. DANIELA ADRIANA SANCHEZ VIZCARRA
PRESIDENTE

DRA. ELBA PATRICIA MELIN OLMEDA
SECRETARIO

DR. OSCAR CASTILLO LOPEZ
VOCAL

- c.c.p. Oficina de Titulación
- c.c.p. División de Estudios de Posgrado e Investigación
- c.c.p. Expediente
- c.c.p. Interesado

EPMO/*inf



DECLARACIÓN DE ORIGINALIDAD

Tijuana, BC., 4 de abril de 2022,

Yo, **RODRIGO CORDERO MARTÍNEZ**, estudiante de la Maestría en Ciencias de la Computación, en mi calidad de autor manifiesto que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derechos de publicación, derechos de autor, patente y similaridad. Por lo tanto, la obra realizada es de mi exclusiva autoría y no infringí en copiar el texto o imágenes, de fuentes de información por lo cual soy responsable del escrito que aquí se presenta.

Así mismo, declaro que en las citas textuales que he incluido (las cuales aparecen entre comillas) y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y las publicaciones.

En caso de presentarse cualquier reclamación o acción por parte de terceros en cuanto a los derechos de autor sobre la obra en cuestión, acepto toda la responsabilidad de tal infracción y relevo de esta a mi director de tesis, así como al Tecnológico Nacional de México, al Instituto Tecnológico de Tijuana y a sus respectivas autoridades.

RODRIGO CORDERO MARTÍNEZ



Estudiante de la Maestría en Ciencias de la Computación

CARTA DECLARACIÓN DE PROPIEDAD INTELECTUAL

Tijuana, BC a 4 de abril del 2022

Yo **RODRIGO CORDERO MARTÍNEZ** reconozco que el Trabajo de Tesis de Maestría que realice durante mis estudios en la Maestría en Ciencias de la Computación del Instituto Tecnológico de Tijuana fue parte del Proyecto de Investigación Titulado: DETECCIÓN AUTOMÁTICA DE RETINOPATÍA DIABÉTICA POR MEDIO DE UN NUEVO MODELO DE REDES NEURONALES CONVOLUCIONALES No 10336.21P que desarrolla mi director de tesis la **Dr. DANIELA ADRIANA SÁNCHEZ VIZCARRA** -y del cual es responsable del proyecto de investigación. Por esta razón, los métodos, modelos, algoritmos, y software realizados, así como datos y resultados obtenidos durante el desarrollo de mi tesis de maestría son propiedad intelectual de mi Director de Tesis, del Tecnológico Nacional de México, Instituto Tecnológico de Tijuana y del Conacyt, y No podré utilizarlos por mi cuenta durante, Ni después de terminar mi beca o estudios, excepto a solicitud escrita para poder utilizarlos bajo una colaboración directa con mi director el cual es responsable del proyecto de investigación. Por tanto, estoy de acuerdo en que No podre utilizar ni tomar modelos, ni datos utilizados en este proyecto de investigación y en el desarrollo de tesis para: presentaciones, publicaciones ni desarrollo de mi propia investigación que pudiera desarrollar una vez concluidos mis estudios.

Atentamente

RODRIGO CORDERO MARTÍNEZ



Estudiante de la Maestría en Ciencias de la Computación

Resumen

La retinopatía diabética (RD) es uno de los padecimientos causados por la diabetes mellitus (DM) que afecta a personas jóvenes que no tuvieron suficientes cuidados con su enfermedad; ya sea falta de conocimientos de parte del paciente para cuidar su enfermedad, o una detección tardía de esta. La RD no tiene síntomas en sus primeras etapas, por lo que aumenta el riesgo de que el paciente pueda quedar parcial o completamente ciego. Debido a la reducida cantidad de personas especializadas en la detección de este padecimiento, algunos investigadores han empezado a utilizar las tecnologías del aprendizaje profundo para automatizar la detección de RD. Una de estas tecnologías son las redes neuronales convolucionales (RNC). Existen métodos de preprocesamiento para la base de datos que son importantes para aumentar la precisión de detección de las RNC, y el uso de un algoritmo de optimización puede aumentar aún más tal precisión. En este presente trabajo de Tesis se presentan cuatro métodos de preprocesamiento distintos para utilizarlos en un par de modelos de RNC para compararlos y seleccionar el mejor método. Después, se utiliza un algoritmo genético jerárquico (AGJ) con el mejor método de preprocesamiento con la intención de aumentar la precisión de clasificación de un nuevo modelo de RNC. El uso de AGJ aumentó las precisiones obtenidas por los métodos de preprocesamiento y los resultados obtenidos por otros autores. En el caso de estudio binario (detección de RD) se obtuvo 0.9781 en la precisión más alta, 0.9650 en la precisión media y 0.007665 en la desviación estándar. En el caso de estudio multiclase (clasificación de RD) se obtuvo 0.7762 en la precisión más alta, 0.7596 en la precisión media y 0.009948 en la desviación estándar.

Abstract

Diabetic retinopathy (DR) is one of the conditions caused by diabetes mellitus (DM) that affects young people who did not take sufficient care with their disease; either lack of knowledge on the part of the patient to take care of their disease, or a late detection of it. DR has no symptoms in its early stages, thus increasing the risk that the patient may become partially or completely blind. Due to the small number of people specialized in detecting this condition, some researchers have begun to use deep learning technologies to automate DR detection. One of these technologies is convolutional neural networks (CNN). There are preprocessing methods for the database that are important to increase the accuracy for detection of CNN, and the use of an optimization algorithm can further increase such accuracy. In this work, four different preprocessing methods are presented to be used in a couple of CNN models to compare them and select the best method. Then, a Hierarchical Genetic Algorithm (HGA) with the best preprocessing method is used with the intention of increasing the classification accuracy of a new CNN model. The use of HGA increased the precisions obtained by the preprocessing methods and the results obtained by other authors. In the case of the binary study (DR detection), 0.9781 was obtained for the highest precision, 0.9650 for the average precision, and 0.007665 for the standard deviation. In the case of the multiclass study (DR classification), 0.7762 was obtained for the highest precision, 0.7596 for the average precision, and 0.009948 for the standard deviation.

Dedicatoria

Este trabajo de tesis está dedicada a mi familia y a Dios. A todos ellos agradezco sinceramente por todo su apoyo incondicional. Este trabajo es para ustedes, ya que, nunca podría haberlo terminado si no estuvieran a mi lado. Muchas gracias.

Agradecimientos

Al primero que debo agradecer, es a Dios porque ha estado conmigo incluso cuando yo no tenía algún conocimiento sobre él, ya que, desde mi niñez, estuvo poniendo en mi camino grandes personas que me han instruido por los caminos correctos y me han hecho reflexionar hasta convertirme en la persona que soy actualmente.

Agradecer siempre a mi familia, ya que, aunque son muy pocos los que están físicamente a mi lado, he tenido el apoyo de ellos siempre que lo he necesitado.

A mis padres, Rodrigo y Norma por criarme y educarme para tomar las mejores decisiones para mí y para el resto de las personas que me rodean.

A mis hermanos Eduardo y Ammi Sinaí por estar conmigo en todo momento a pesar de lo diferente de las etapas que pasamos los 3.

Agradezco a CONACYT por el apoyo otorgado a través de su beca a mi número de CVU 1036418 durante el periodo Febrero-2020 a Enero-2022 con el cual pude concluir los estudios de Maestría en Ciencias de la Computación.

A todos, por siempre, se los agradeceré.

Índice General

Resumen.....	I
Abstract.....	II
Dedicatoria.....	III
Agradecimientos.....	IV
Índice General.....	V
Índice de Figuras.....	VII
Índice de Tablas.....	VIII
Índice de Anexos.....	IX
1 Introducción.....	1
1.1 Objetivo general.....	1
1.2 Objetivos específicos.....	1
1.3 Diabetes mellitus.....	2
1.3.1 Padecimientos.....	2
1.3.2 Prevención.....	2
1.4 Retinopatía diabética.....	3
1.4.1 Clasificación.....	3
2 Marco Teórico.....	6
2.1 Red neuronal.....	6
2.1.1 Neurona.....	6
2.1.2 Arquitectura de una red.....	7
2.1.3 Modelo de red Feedforward no lineal.....	8
2.1.4 Tipos de aprendizaje supervisado.....	9
2.2 Algoritmo genético simple.....	9
2.2.1 Reproducción.....	11
2.2.2 Cruce.....	13
2.2.3 Mutación.....	14
2.2.4 Micro algoritmo genético.....	15
2.2.5 Algoritmo genético jerárquico.....	15
2.3 Red neuronal convolucional.....	16
2.3.1 Capa de convolución.....	17
2.3.2 Función de activación ReLU.....	17
2.3.3 Capa de Max-Pooling.....	18
	V

3	Estado del arte	20
3.1	Detección de retinopatía diabética a través de redes neuronales convolucionales con meta plasticidad sináptica.....	20
3.2	Método basado en aprendizaje profundo para el diagnóstico asistido por computadora de la retinopatía diabética	21
3.3	Diseño de una arquitectura de red neuronal convolucional usando algoritmos genéticos.....	21
4	Planteamiento del Problema	25
5	Desarrollo.....	26
5.1	Base de datos APTOS 2019	26
5.2	Preprocesamiento aplicado a la base de datos.....	27
5.2.1	Preprocesamiento 1.....	30
5.2.2	Preprocesamiento 2.....	31
5.2.3	Preprocesamiento 3.....	31
5.2.4	Preprocesamiento 4.....	32
5.3	Algoritmo genético jerárquico.....	33
5.3.1	Genes de los cromosomas	33
6	Resultados	38
6.1	Experimentos realizados sobre los preprocesamientos	38
6.1.1	Caso de estudio binario	39
6.1.2	Caso de estudio multiclase	40
6.1.3	Comparación de resultados	40
6.2	Experimentos realizados sobre el algoritmo genético jerárquico	45
6.2.1	Caso de estudio binario	46
6.2.2	Caso de estudio multiclase	48
6.3	Comparación de resultados	49
6.3.1	Prueba de hipótesis para el caso de estudio binario.	50
6.3.2	Prueba de hipótesis para el caso de estudio multiclase	51
6.4	Interfaz gráfica de la aplicación Web.....	52
7	Conclusiones.....	56
	Referencias	58
	Anexos	64

Índice de Figuras

Figura 1.1. Imágenes de una retina sana a) y otra en etapa avanzada de retinopatía diabética b).....	4
Figura 2.1. Gráfico de una neurona simple.	7
Figura 2.2. Ejemplo de una arquitectura de red neuronal artificial.	8
Figura 2.3. Ejemplo gráfico de selección por ruleta.....	12
Figura 2.4. Seleccionando el individuo más apto.	13
Figura 2.5. Ejemplo de cruce simple.	13
Figura 2.6. Ejemplo de cruce binario.....	14
Figura 2.7. Ejemplo simple de mutación.	15
Figura 2.8. Ejemplo de representación de cromosoma de un AGJ.	16
Figura 2.9. Ejemplo de red neuronal convolucional.....	17
Figura 2.10. Ejemplo de una convolución.	17
Figura 2.11. Gráfico de la función ReLU.	18
Figura 2.12. Ejemplo de la operación Max-Pooling.	19
Figura 3.1. Ejemplos extraídos de la base de datos MNIST.....	22
Figura 3.2. Ejemplo de la fracción del cromosoma utilizado para obtener las épocas.....	23
Figura 4.1. Propuesta del método.	25
Figura 5.1. Ejemplos de imágenes extraídas de APTOS 2019.	27
Figura 5.2. Estructura del modelo 1 de RNC.....	28
Figura 5.3. Estructura del modelo 2 de RNC.....	29
Figura 5.4. Ejemplos utilizando el preprocesamiento 1.	30
Figura 5.5. Ejemplos utilizando el preprocesamiento 2.	31
Figura 5.6. Ejemplos utilizando el preprocesamiento 3.	32
Figura 5.7. Ejemplos utilizando el preprocesamiento 4.	33
Figura 5.8. Genes de control.	34
Figura 5.9. Genes paramétricos para capas convolucionales.	35
Figura 5.10. Genes paramétricos para las capas completamente conectadas.	36
Figura 6.1. Comparación de resultados entre los modelos de RNC con 3 y 10 capas convolucionales.	41
Figura 6.2. Prueba de hipótesis para el caso de estudio binario.	43
Figura 6.3. Prueba de hipótesis para el caso de estudio multiclase.....	44
Figura 6.4. Modelo de RNC que obtuvo la mejor precisión para el caso de estudio binario.	47
Figura 6.5. Modelo de RNC que obtuvo la mejor precisión para el caso de estudio multiclase.	49
Figura 6.6. Prueba de hipótesis entre los dos experimentos del caso binario.....	51
Figura 6.7. Prueba de hipótesis entre los dos experimentos del caso multiclase..	52
Figura 6.8. Interfaz inicial de la aplicación Web.	53
Figura 6.9. Seleccionando una imagen en Windows 11.....	54
Figura 6.10. Realizando una clasificación binaria.	54
Figura 6.11. Realizando una clasificación multiclase.	55

Índice de Tablas

Tabla 3.1. Distintos hiperparámetros de la RNC con sus rangos.....	23
Tabla 5.1. Número de imágenes por clase de la base de datos APTOS 2019.....	26
Tabla 5.2. Número de imágenes divididas entre sanas y con RD de la base de datos APTOS 2019	26
Tabla 6.1. distribución de las imágenes de la base de datos APTOS 2019.....	38
Tabla 6.2. Resultados del caso de estudio binario.....	39
Tabla 6.3. Resultados del caso de estudio multiclase.....	40
Tabla 6.4. Resultados obtenidos por la prueba de rangos con signo de Wilcoxon	41
Tabla 6.5. Resultados obtenidos por el AGJ para el caso de estudio binario.	46
Tabla 6.6. Resultados obtenidos por el AGJ para el caso de estudio multiclase. .	48
Tabla 6.7. Comparación de resultados para el caso de estudio binario.....	49
Tabla 6.8. Comparación de resultados para el caso de estudio multiclase.....	50

Índice de Anexos

Anexo 1. Resultados experimento del caso binario usando una RNC de profundidad 3 con el método de preprocesamiento 1	64
Anexo 2. Resultados experimento del caso binario usando una RNC de profundidad 3 con el método de preprocesamiento 2	65
Anexo 3. Resultados experimento del caso binario usando una RNC de profundidad 3 con el método de preprocesamiento 3	66
Anexo 4. Resultados experimento del caso binario usando una RNC de profundidad 3 con el método de preprocesamiento 4	67
Anexo 5. Resultados experimento del caso binario usando una RNC de profundidad 10 con el método de preprocesamiento 1	68
Anexo 6. Resultados experimento del caso binario usando una RNC de profundidad 10 con el método de preprocesamiento 2	69
Anexo 7. Resultados experimento del caso binario usando una RNC de profundidad 10 con el método de preprocesamiento 3	70
Anexo 8. Resultados experimento del caso binario usando una RNC de profundidad 10 con el método de preprocesamiento 4	71
Anexo 9. Resultados experimento del caso multiclase usando una RNC de profundidad 3 con el método de preprocesamiento 1	72
Anexo 10. Resultados experimento del caso multiclase usando una RNC de profundidad 3 con el método de preprocesamiento 2	73
Anexo 11. Resultados experimento del caso multiclase usando una RNC de profundidad 3 con el método de preprocesamiento 3	74
Anexo 12. Resultados experimento del caso multiclase usando una RNC de profundidad 3 con el método de preprocesamiento 4	75
Anexo 13. Resultados experimento del caso multiclase usando una RNC de profundidad 10 con el método de preprocesamiento 1	76
Anexo 14. Resultados experimento del caso multiclase usando una RNC de profundidad 10 con el método de preprocesamiento 2	77
Anexo 15. Resultados experimento del caso multiclase usando una RNC de profundidad 10 con el método de preprocesamiento 3	78
Anexo 16. Resultados experimento del caso multiclase usando una RNC de profundidad 10 con el método de preprocesamiento 4	79
Anexo 17. Características de la computadora utilizada para elaborar el presente trabajo de Tesis.....	80
Anexo 18. Tiempos de ejecución de los experimentos del AGJ para el caso de estudio binario.....	81
Anexo 19. Tiempos de ejecución de los experimentos del AGJ para el caso de estudio multiclase.....	82
Anexo 20. Código de Matlab sobre el método de preprocesamiento 1 (solo imágenes de retina sanas).....	82
Anexo 21. Código de Matlab sobre el método de preprocesamiento 2 (solo imágenes de retina sanas).....	83

Anexo 22. Código de Matlab sobre el método de preprocesamiento 3 (solo imágenes de retina sanas).....	84
Anexo 23. Código de Matlab sobre el método de preprocesamiento 4 (solo imágenes de retina sanas).....	85

1 Introducción

La diabetes mellitus (DM) es un problema de salud pública mundial de alta prioridad [1]. Existen complicaciones sistémicas, una de ellas es la principal causa de ceguera y debilidad visual irreversible: la retinopatía diabética (RD) [2]. La DM puede afectar a más de una parte del cuerpo porque aumenta el riesgo de enfermedades cardíacas y accidentes cerebrovasculares, que a largo plazo pueden derivar en diferentes afecciones, como: insuficiencia renal, ceguera, amputaciones, impotencia sexual, etc. [3].

El mayor problema actual con la RD es que se vuelve incurable en etapas avanzadas, por lo que la detección temprana es importante [4]. Sin embargo, hay un gran número de pacientes potenciales y un número reducido de técnicos expertos. Debido a esta situación, se han desarrollado diversos sistemas con capacidad de detección y diagnóstico automático para agilizar el proceso de tratamiento, los cuales se basan en técnicas actuales y métodos de avances tecnológicos que se han logrado en inteligencia artificial y aprendizaje profundo [5].

En el mundo de la medicina, el uso de métodos de aprendizaje profundo se ha visto principalmente en la rama de la oftalmología con el análisis de imágenes de fondo de ojo y en la tomografía de coherencia óptica. Estos métodos han logrado un rendimiento de alto impacto en la detección de enfermedades oculares [6]. Las redes neuronales han sido una de las tecnologías más utilizadas en los últimos años para la detección de RD [7].

1.1 Objetivo general

Diseñar un nuevo modelo de red neuronal convolucional para la detección y clasificación automática de retinopatía diabética.

1.2 Objetivos específicos

- Realizar investigación de preprocesamientos aplicados a imágenes de retinopatía diabética.
- Desarrollar redes neuronales convolucionales para entrenar imágenes para la clasificación de RD.

- Realizar experimentos con el método propuesto y realizar pruebas estadísticas para la comparación de resultados.

1.3 Diabetes mellitus

La diabetes mellitus (DM) es una alteración metabólica caracterizada por la presencia de hiperglucemia crónica [8] que se acompaña, en mayor o menor medida, de alteraciones en el metabolismo de los hidratos de carbono, de las proteínas y de los lípidos [9]. El origen y la etiología de la DM pueden ser muy diversos, pero conllevan inexorablemente la existencia de alteraciones en la secreción de insulina, de la sensibilidad a la acción de la hormona, o de ambas en algún momento de su historia natural [10].

1.3.1 Padecimientos

La DM puede afectar a más de una parte del cuerpo [11], por lo que hay que tener cuidado e ir constantemente a que un médico realice un debido chequeo médico [12].

La DM puede tener distintos padecimientos [13], tales como:

- Complicaciones agudas.
- Complicaciones crónicas.
- Retinopatía diabética.
- Neuropatía.
- Nefropatía.
- Morbilidad y mortalidad cardiovascular.
- Hipertensión.

1.3.2 Prevención

El origen de la DM en personas se origina debido a la combinación de varios factores, como la edad, el sedentarismo y/o una adecuada alimentación [14]. Existen distintas estrategias que se pueden seguir de manera individual para prevenir la aparición de la DM [15], entre los cuales se encuentran:

- Consumo de frutas y verduras.
- Restringir el consumo de comidas rápidas con alto contenido energético.

- Limitar el consumo de bebidas azucaradas.
- Si se tiene obesidad, reducir el peso corporal.
- Consumo de pescado ricos en omega 3.
- Consumo de arroz blanco.
- Realizar ejercicio periódicamente.

1.4 Retinopatía diabética

La retinopatía diabética (RD) es una complicación microvascular común y específica de la diabetes y sigue siendo la principal causa de ceguera prevenible en las personas en edad laboral. Se identifica en personas con diabetes mellitus y se asocia con un mayor riesgo de complicaciones vasculares sistémicas potencialmente mortales, que incluyen accidente cerebrovascular, enfermedad coronaria e insuficiencia cardíaca [16]. La RD genera una disminución progresiva de la agudeza visual y puede producir ceguera. Esta pérdida de la visión disminuye la calidad de vida del paciente y de sus familiares, y genera altos gastos económicos a la familia y al país [17].

De acuerdo con la Encuesta Nacional de Salud y Nutrición de Medio Camino 2016 (ENSANUT), 54.5% de las personas con DM obtuvo una visión disminuida, mientras que el 11.2% sufrió daño en la retina y 9.95% perdió la vista [18].

La probabilidad de que una persona en el mundo en el año 2002 se quedara ciega, a causa de la RD era solo de 0.75% [19]. Debido al crecimiento de la cantidad de pacientes con DM, mundialmente en el año 2013 existían 382 millones de personas con un aproximado de 35% de estos pacientes tuvieron algún nivel de RD, y se estima que para el año 2035 sean 559 millones de personas con DM, teniendo así más de 4 millones de personas ciegas a causa de RD [20].

1.4.1 Clasificación

La RD se puede clasificar en 5 etapas [21]:

- La etapa inicial es simplemente una retina sana donde no hay indicios del padecimiento.
- No proliferativa leve se encuentran solo microaneurismas.

- No proliferativa moderada se localizan algunas microaneurismas y otras lesiones microvasculares.
- No proliferativa severa hay más de 20 hemorragias intrarretinianas en cuatro cuadrantes o abalorios venosos en dos o más cuadrantes, o anomalías microvasculares intrarretinianas en uno o más cuadrantes.
- Proliferativa se encuentran neovascularización del disco óptico (NDO), hemorragia prerretiniana o hemorragia vítrea; las características de alto riesgo son NDO leve con hemorragia vítrea, NDO moderada a grave con o sin hemorragia vítrea.

En la Figura 1.1 se puede observar una imagen de a) una imagen de retina completamente sana (primera etapa en la clasificación) y al lado derecho b) una imagen de retina en el estado más avanzado de retinopatía diabética.

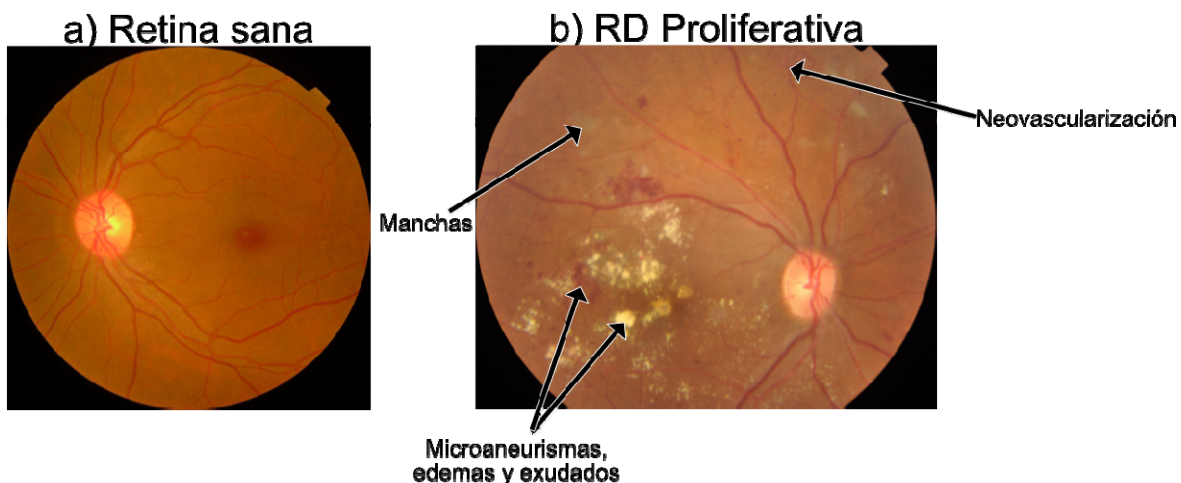


Figura 1.1. Imágenes de una retina sana a) y otra en etapa avanzada de retinopatía diabética b).

La aportación generada al realizar este presente trabajo de Tesis fue un artículo para un congreso internacional donde se utilizaron distintos métodos de preprocesamiento aplicados a una base de datos con imágenes de retina, con el objetivo de detectar y clasificar retinopatía diabética de la manera más efectiva, el cual fue presentado en el 21st International Conference on Hybrid Intelligent Systems (HIS 2021) [22].

El presente trabajo de Tesis se encuentra estructurado por capítulos, los cuales se describen brevemente a continuación:

Introducción

En el capítulo 2 se definen los conceptos básicos y su proveniencia para un mayor entendimiento del tema de investigación y el problema al que se le busca solución. Los temas que se profundizan en este capítulo son: red neuronal artificial y convolucional; diabetes mellitus y retinopatía diabética.

Una vez explicadas las definiciones básicas para el abordado del planteamiento del problema, sigue el capítulo 3 donde se muestran los trabajos hechos por investigadores y expertos anteriores que abordan parte de este tema de investigación. Desde la idea de implementar el uso de las redes neuronales convolucionales para la detección de retinopatía diabética (el paciente tiene o no tiene el padecimiento), hasta la clasificación por grado de seriedad de este padecimiento.

En el capítulo 4 se muestra brevemente el planteamiento del problema donde se describe la función objetivo utilizada a lo largo de todo el presente trabajo de Tesis.

En el capítulo 5 se describe el desarrollo de los métodos diseñados y sus propiedades para todos los experimentos del presente trabajo.

Luego, en el capítulo 6 se observa la creación y experimentación del modelo de red neuronal convolucional aplicado en la detección de retinopatía diabética y su clasificación, obteniendo buenos resultados para llegar a una conclusión satisfactoria.

Después, en el capítulo 7 se presentan las conclusiones donde se resumirán los resultados obtenidos por el desarrollo después de haber realizado el presente proyecto de Tesis.

Finalmente, al final del trabajo se muestra la bibliografía estudiada para la realización de este proyecto de Tesis y los anexos.

2 Marco Teórico

Es importante tener un conocimiento general sobre los temas y tecnologías que se quieren utilizar para dar con la solución. Hay una brecha entre las áreas de investigación de la medicina con las del área tecnológico, por el cual muchas personas, incluyendo a los mismos médicos, no ven como una solución viable el uso de estas tecnologías para funcionar en instalaciones de mucha seriedad como lo es el área médica.

2.1 Red neuronal

También llamadas redes neuronales artificiales, son una herramienta muy utilizada para la resolución de problemas de clasificación como el reconocimiento de caracteres, reconocimiento facial, y el diagnóstico de diferentes enfermedades [23].

El uso de la inteligencia artificial en el área médica es un recurso muy utilizado, siendo las redes neuronales un recurso que aún hoy se utiliza para la detección de enfermedades, reconocimiento de patrones y predicción [24].

2.1.1 Neurona

Es el componente mínimo de una red neuronal. Es un dispositivo que transforma las entradas en una única salida. Esta salida se puede transmitir a otras neuronas [25].

Las señales de entrada se modulan con un factor llamado peso que se encarga de graduar la importancia de la conexión entre dos neuronas. Entonces se puede decir que una neurona es en realidad un procesador muy simple con una limitada capacidad de cómputo que está restringida a un conjunto de sumas y productos; y una memoria pequeña para almacenar los pesos y activaciones.

La función de activación es una función que se aplica al valor de entrada. Este paso consiste en convertir dicho valor en un nivel de activación para la neurona, el cual es el equivalente al nivel de excitación de una neurona biológica. En la Figura 2.1 se puede observar una representación de una neurona artificial.

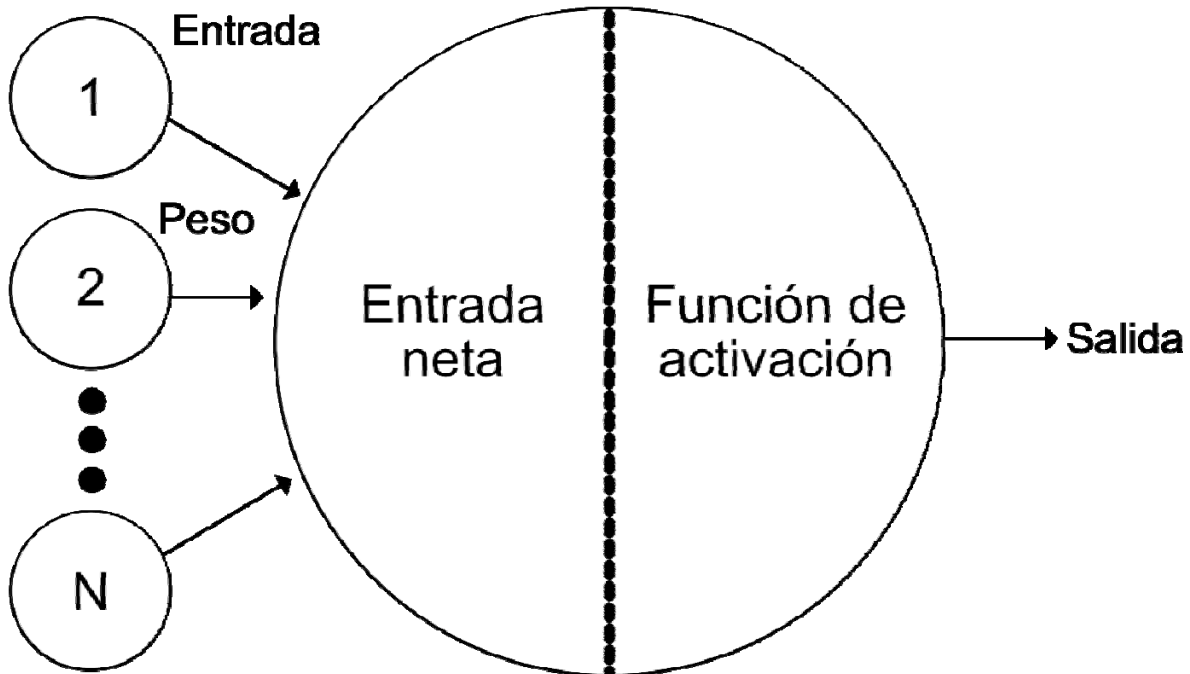


Figura 2.1. Gráfico de una neurona simple.

2.1.2 Arquitectura de una red

Para obtener una red neuronal, se deben definir las conexiones entre distintas neuronas. Estas se agrupan en capas, cada una de ellas con un conjunto de neuronas variables y de comportamiento similar, que juntándolas se obtiene la red neuronal [26].

Cada capa está organizada de manera consecutiva, estando cada capa conectada con la siguiente, con excepción de la última capa que equivale a la capa de salida. Por lo que tenemos tres tipos de capas.

- Capa de entrada: En esta capa se encuentran neuronas que dependen completamente de los datos de entrada al problema.
- Capas intermedias: También son conocidas como capas ocultas. A diferencia de la capa de entrada y salida, las capas ocultas no están limitadas a una capa, por lo que puedes tener las que necesites dependiendo del tipo y complejidad del problema que se desee resolver. Estas capas consiguen las propiedades de generalización extracción de características, adaptabilidad, etc.

- Capa de salida: La cantidad de neuronas que posea esta capa, dependerá del formato esperado de salida de la red.

Estos tipos de capas se pueden apreciar de mejor manera en la Figura 2.2.

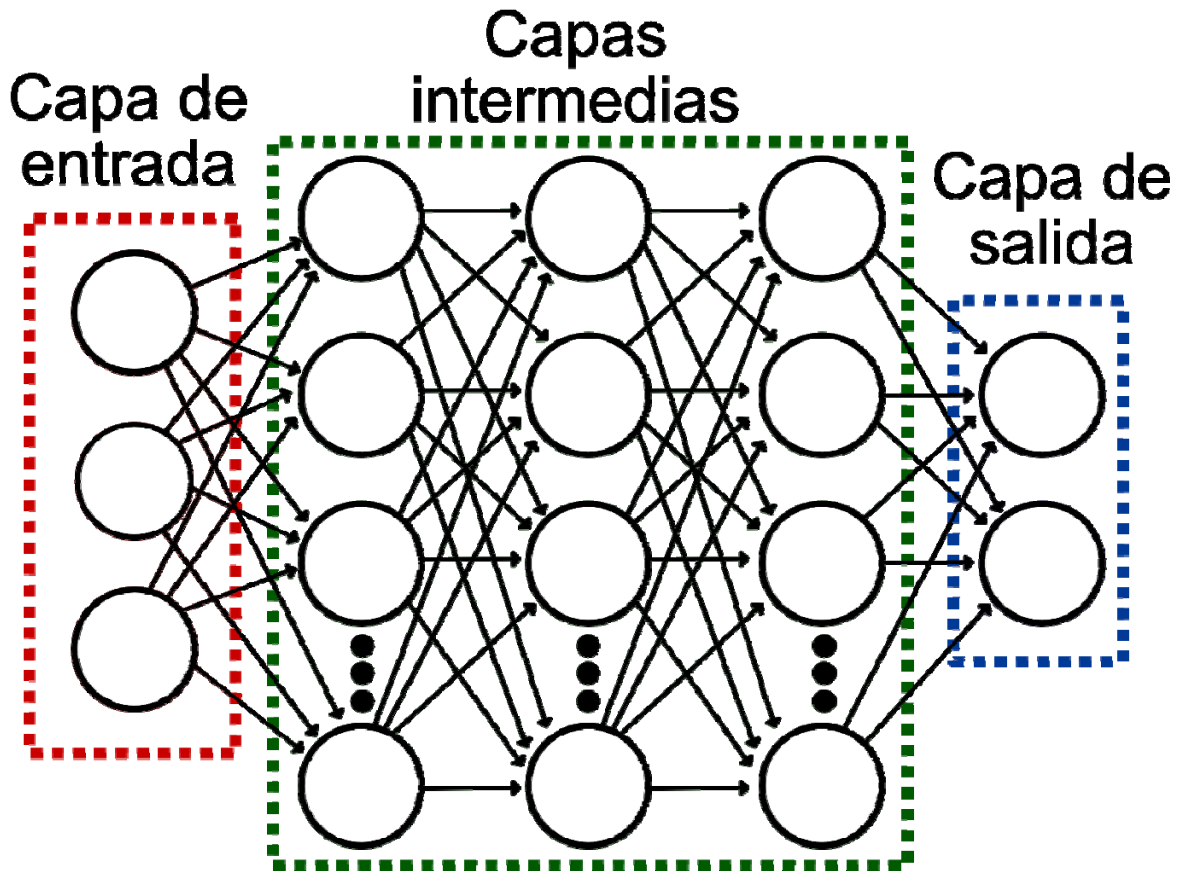


Figura 2.2. Ejemplo de una arquitectura de red neuronal artificial.

2.1.3 Modelo de red Feedforward no lineal

El modelo de red Feedforward se caracteriza porque la salida de una neurona está conectada a las entradas de otras neuronas, pero no a su propia entrada; esto quiere decir que los impulsos se propagan en una única dirección. Este modelo de red es bastante eficiente en costo computacional y les es más sencillos llegar a una solución cuando ya estén entrenadas [27].

En los modelos no lineales, la salida de una neurona es una función no lineal de la suma de sus entradas.

2.1.4 Tipos de aprendizaje supervisado

El modelo de red Feedforward no lineal se divide en dos distintos tipos de red neuronal, siendo uno de ellos el supervisado, y el otro el no supervisado; siendo el primero más utilizado cuando se tiene conocimiento de los valores en la capa de entrada y salida [28].

En los modelos con aprendizaje supervisado, los pesos de las neuronas se pueden ajustar ya que se conocen los datos de entrada y salida deseada; por lo que, usando el error obtenido, se permite hacer un ajuste a los pesos tratando de minimizar ese error.

Para crear un modelo de red neuronal que pueda usarse para detectar enfermedades o anomalías en fotografías, es mejor si la red tiene aprendizaje supervisado [29].

2.2 Algoritmo genético simple

La evolución es un tema en biología que se ha debatido ampliamente durante muchos años [30], y este proceso sirvió de inspiración para el desarrollo de un algoritmo de optimización llamado algoritmo genético simple (llamado a partir de aquí como algoritmo genético o AG). Los algoritmos genéticos se basan en la teoría de la evolución de Darwin [31].

Los AG son algoritmos de búsqueda basados en las mecánicas de selección y genética naturales. Combinan la supervivencia del más apto entre el resto de los individuos con un intercambio de información estructurado pero aleatorio para formar un algoritmo de búsqueda humana. En cada generación, se crea un conjunto de criaturas artificiales (individuos) usando trozos y piezas de dos de estos individuos, buscando que el individuo resultante se convierta en el más apto. Aunque el proceso sea aleatorio, los AG no se consideran un método simple que se base en la aleatoriedad, sino que contiene atributos que especifican áreas donde son más eficientes y permite almacenar la línea evolutiva de los individuos para crear nuevos espacios de búsqueda que sean más efectivos [32].

Los AG fueron creados por John Holland, sus colegas y estudiantes de la Universidad de Michigan [33]. En su investigación tenían un par de metas:

1. Resumir y explicar de manera rigurosa los procesos adaptivos de los sistemas naturales.
2. Diseñar sistemas artificiales que retuvieran los mecanismos de los sistemas naturales.

Este enfoque ha llevado a importantes descubrimientos en la ciencia de los sistemas naturales y artificiales.

El tema central de la investigación sobre AG ha sido robusto. Si los sistemas artificiales pueden ser robustos, significa que los rediseños pueden reducirse o incluso ser eliminados. Esto también significa que los sistemas existentes pueden realizar sus funciones de mejor forma y por más tiempo. Sin embargo, los sistemas biológicos aún tenían capacidades superiores, tales como auto repararse, mejorarse a sí mismo e incluso reproducirse, mientras que apenas podían existir en los sistemas artificiales más sofisticados.

Esto llegaba a una interesante conclusión: cuando los sistemas artificiales alcancen la robustez deseada, los sistemas biológicos lo seguirán haciendo mejor; los secretos de adaptarse y sobrevivir son mejor aprendidos desde una muestra biológica. Siendo aquí donde los AG obtuvieron su inspiración y comenzaron a separarse del funcionamiento de otros métodos de optimización más tradicionales.

Existen cuatro formas en las que los AG son diferentes de otros algoritmos de optimización o de búsqueda:

1. Los AG trabajan con una codificación del conjunto de parámetros, no de los parámetros en sí.
2. Buscan desde una población de puntos, no desde un solo punto.
3. Tienen una función objetivo, no obtienen información de una fuente externa.
4. Utilizan reglas de transición probabilísticas, no reglas determinadas.

Las mecánicas de los AG son bastante simples, resumiéndose en sus funciones más complejas a copiar e intercambiar de forma parcial la información de los individuos. Las operaciones simples y el gran efecto que estas tienen a la

hora de optimizar son dos de los principales atractivos de enfocarse en los AG. La forma de medir a los individuos es por su aptitud, es decir, cada individuo de la población generada tendrá la misma función objetivo pero distintas aptitudes. El individuo que tenga su aptitud más cerca de la función objetivo, se le considera el individuo más apto de la población y mejor solución para la optimización.

Una AG simple que produce buenos resultados en una gran cantidad de problemas prácticos está compuesto de tres operaciones:

1. Reproducción.
2. Cruce.
3. Mutación.

2.2.1 Reproducción

Empezando con la reproducción, también conocida como selección, es un proceso en el cual los individuos son copiados de acuerdo con la función objetivo. Esto significa que copiando individuos tomando en cuenta el objetivo, es más probable que la siguiente descendencia sea mejor en la siguiente generación. Por supuesto, no hay que olvidar que esta operación de reproducción es una versión artificial inspirada en la teoría de Darwin.

La operación de reproducción puede implementarse de distintas formas en un algoritmo. Una de ellas es la de distribuir en una especie de ruleta a los individuos dándoles más probabilidad de ser seleccionado a los individuos con mejor aptitud. Después de ser seleccionados, la operación de reproducción está concluida. En la Figura 2.3 se puede apreciar de manera gráfica un ejemplo de selección por ruleta utilizando 4 individuos distintos.

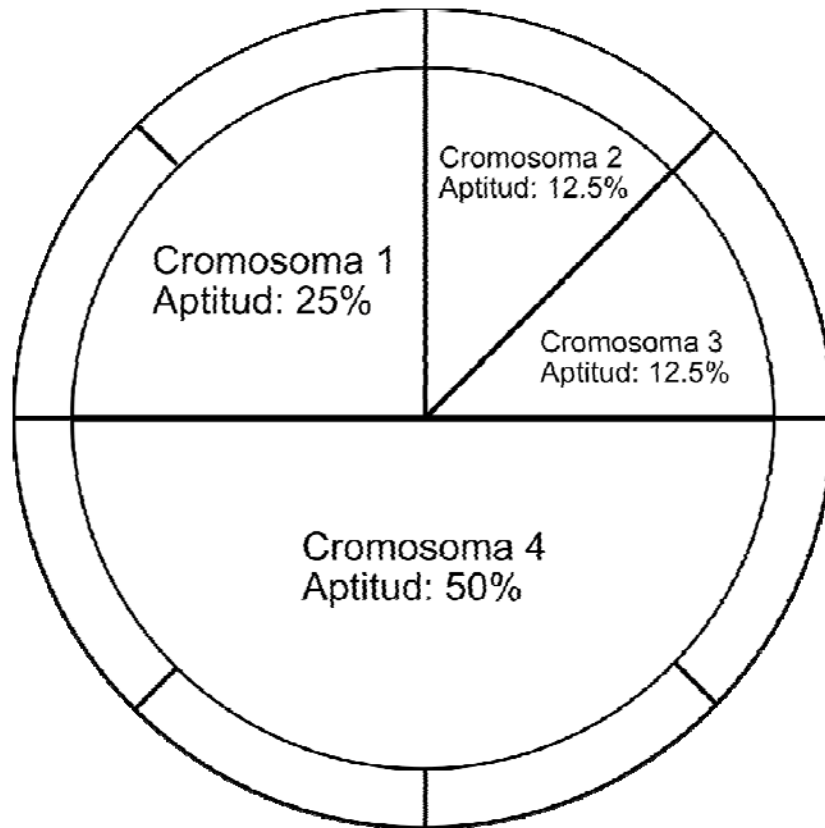


Figura 2.3. Ejemplo gráfico de selección por ruleta.

Existe otro método de selección llamada selección por torneo. La idea principal de este método consiste en realizar la selección en base a comparaciones directas entre individuos. Se selecciona al azar un número p de individuos (generalmente se escoge $p = 2$ pero el valor depende del desarrollador en base al problema a resolver o en el tamaño de la población). De entre los individuos seleccionados se selecciona el más apto para pasarlo a la siguiente generación. Un ejemplo de este método de selección se puede observar en la Figura 2.4 donde se selecciona el individuo con la aptitud más alta.

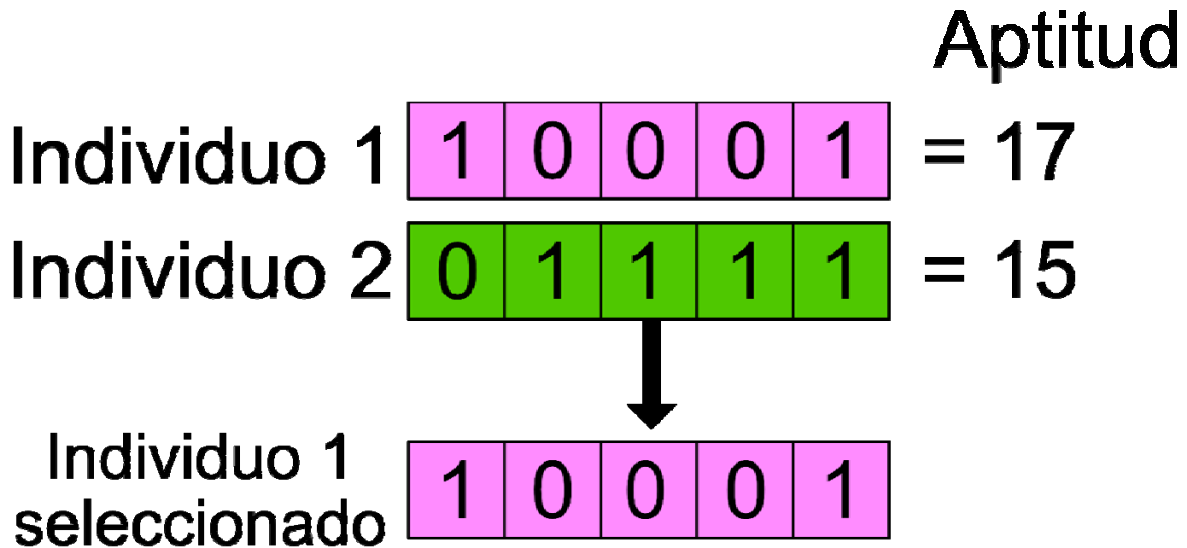


Figura 2.4. Seleccionando el individuo más apto.

2.2.2 Cruce

Una vez que los individuos se han seleccionado, se puede aplicar la operación de cruce simple. Esta operación consiste en escoger un punto de manera aleatorio entre los cromosomas, e intercambiar el área seleccionada. En la Figura 2.5 se puede observar un ejemplo de este cruce.

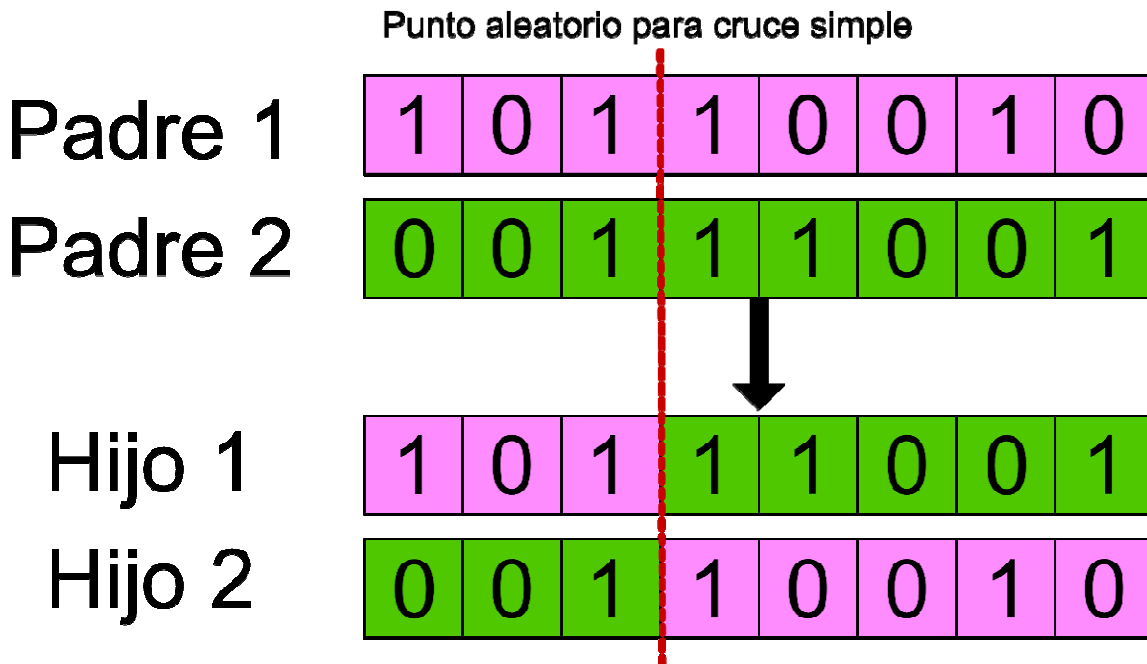


Figura 2.5. Ejemplo de cruce simple.

Otro método de cruce muy utilizado es el cruce binario. Este método consiste en crear una máscara con la misma longitud de un cromosoma con valores en sus genes de 0 o 1. Acto seguido se obtiene el primer hijo al utilizar la máscara con los valores del primer padre cuando sea 1 y los valores del segundo padre cuando sea 0. Cuando el primer hijo esté listo, se invierten el orden de los padres y se genera el segundo hijo. En la Figura 2.6 se puede observar más en detalle el cruce binario.

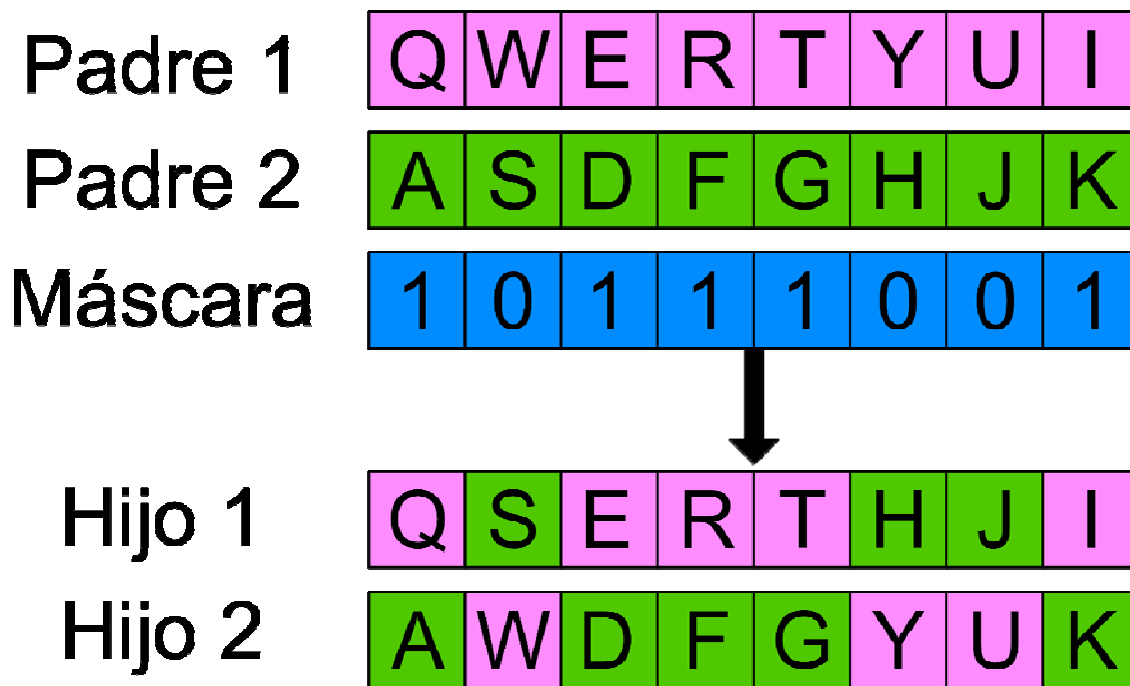


Figura 2.6. Ejemplo de cruce binario.

2.2.3 Mutación

Originalmente, la mutación se diseñó solo para el cromosoma representado binariamente. Para adoptar el concepto de introducir variaciones en el cromosoma, se ha diseñado una mutación aleatoria para el cromosoma del número real, como se observa en la Ecuación 2.1:

$$g = g + \psi(\mu, \sigma) \quad (2.1)$$

Donde g es el valor real del gen; ψ es una función aleatoria que puede ser gaussiana o normalmente distribuida; μ , σ son la media y la varianza relacionadas

con la función aleatoria, respectivamente. En la Figura 2.7 se puede observar un ejemplo de mutación sobre un cromosoma.

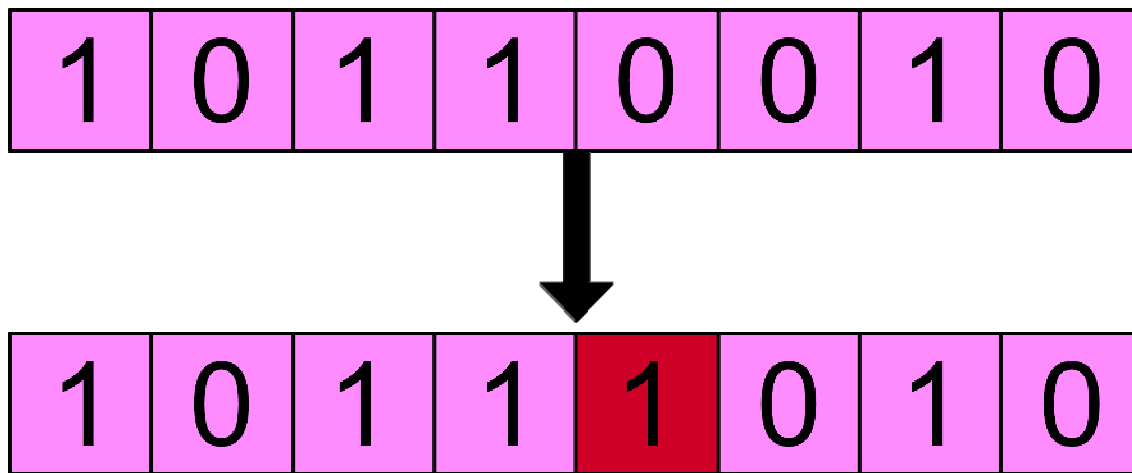


Figura 2.7. Ejemplo simple de mutación.

2.2.4 Micro algoritmo genético

Una variación del algoritmo genético es la denominada micro algoritmo genético, también expresada como μ AG [34]. La principal diferencia de este algoritmo se encuentra en la cantidad máxima de individuos que se encuentra en la población, siendo de 5 como máximo [35]. La intención de tener una población tan pequeña es la de aumentar las probabilidades de cruzar a los mejores individuos [36]. Al ser una población pequeña, es más fácil sustituir a los padres por mejores hijos en menos tiempo. Sin embargo, esto puede resultar contraproducente si el algoritmo genético requiere optimizar muchas variables ya que limita la diversidad de los individuos.

2.2.5 Algoritmo genético jerárquico

Una de las extensiones de los algoritmos genéticos simples son los algoritmos genéticos jerárquicos (AGJ). La principal diferencia es que en los algoritmos genéticos jerárquicos hay dos tipos de genes: control y paramétrico [37]. Los AGJ fueron desarrollados para resolver una clase particular de problemas jerárquicos. Se puede representar como un árbol donde los genes de los pisos superiores (genes de control) se encargan de controlar los genes de los pisos inferiores (genes paramétricos) [38]. Un ejemplo de la representación de un cromosoma de un AGJ se puede observar en la Figura 2.8.

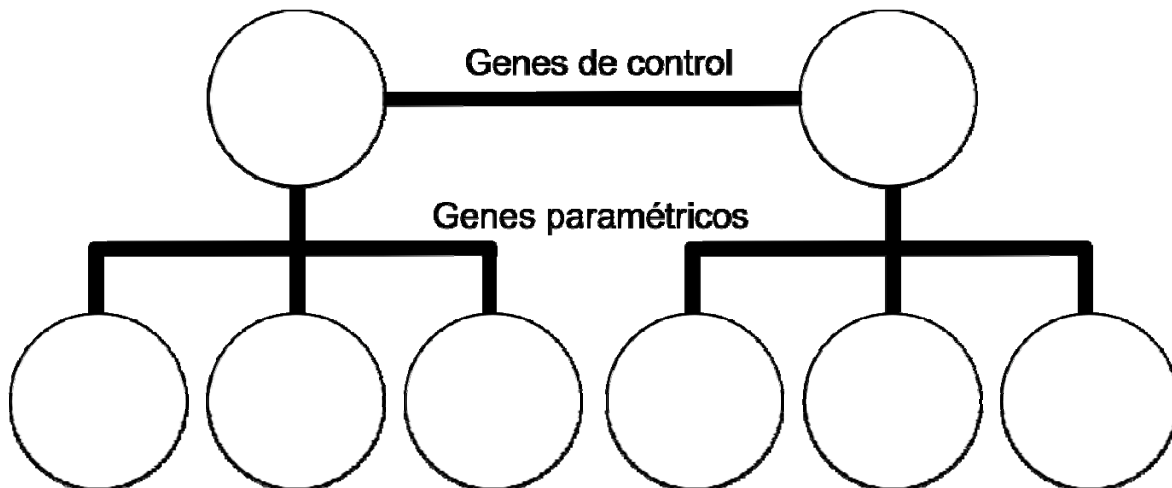


Figura 2.8. Ejemplo de representación de cromosoma de un AGJ.

2.3 Red neuronal convolucional

El científico Yann André LeCun es reconocido por escribir un artículo en el cual muestra las bases para utilizar un nuevo tipo red neuronal utilizada para el reconocimiento de dígitos escritos a mano [39].

Para que una red neuronal aprenda tanto como sea posible, a menudo involucra a técnicos calificados que extraen características importantes de las imágenes [40]. Sin embargo, en los últimos años se ha utilizado una especie de red neuronal que realiza esta extracción de características sin necesidad de un técnico experto. Estas se denominan: redes neuronales convolucionales (RNC) [41].

Una RNC es un tipo de Red Neuronal Artificial con aprendizaje supervisado que puede identificar objetos ya que procesa sus capas tomando como inspiración la corteza visual para decodificar la percepción y convertirla en visión [42]. Para lograr esto, la RNC contiene varias capas ocultas especializadas y con un orden específico: esto quiere decir que las primeras capas pueden detectar bordes sencillos como líneas y/o curvas para reconocer formas más complejas como una huella digital o un rostro en las capas más profundas [43]. En la Figura 2.9 se puede observar un gráfico de una red neuronal convolucional.

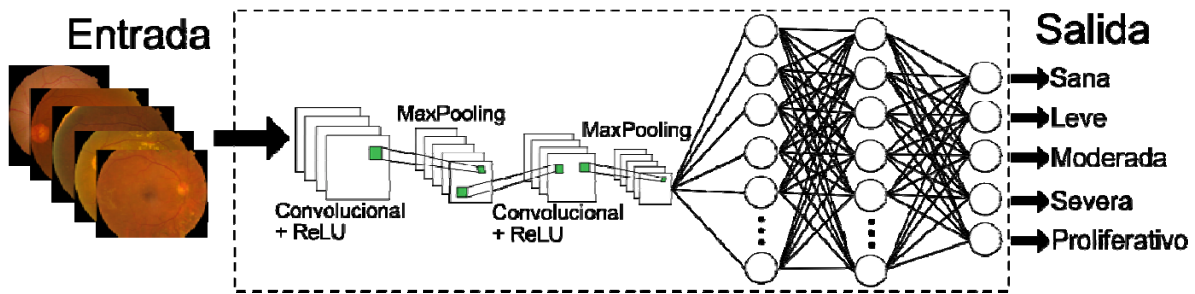


Figura 2.9. Ejemplo de red neuronal convolucional.

2.3.1 Capa de convolución

Esta capa se considera la segunda en la jerarquía de la RNC estando después de la capa de donde se inserta la imagen. El objetivo de esta capa es usar un kernel (esto se considera como el filtro que se aplica a una imagen para extraer ciertas características o patrones importantes) para hacer un producto escalar después de tomar un grupo de píxeles de la imagen de entrada [44]. No existe una regla para determinar el tamaño del kernel, podría ser 2x2, 4x4 o más grande, pero el tamaño más común es 3x3. El kernel pasará por todas las neuronas de entrada y obtendremos una nueva matriz [45]. Un ejemplo visual usando un kernel de 3x3 se puede observar en la Figura 2.10.

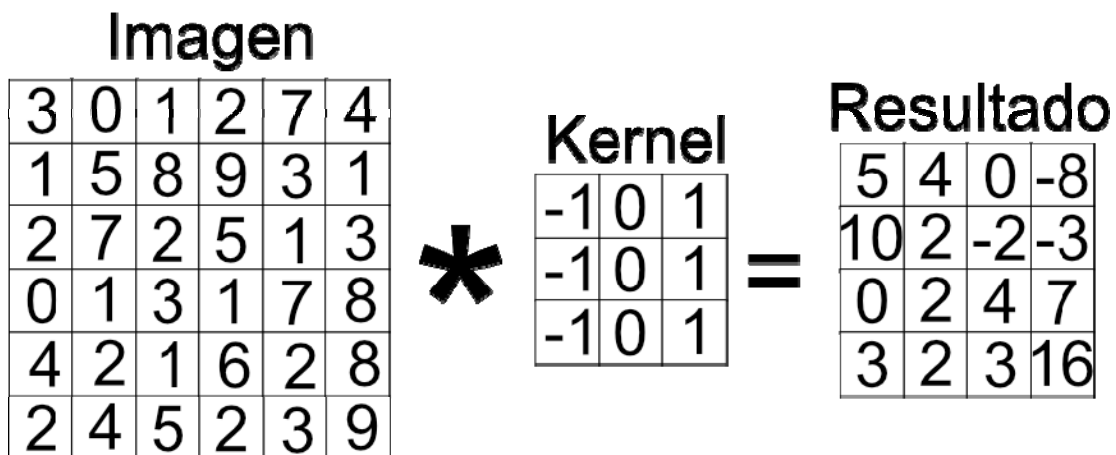


Figura 2.10. Ejemplo de una convolución.

2.3.2 Función de activación ReLU

Hay muchas funciones de activación que se pueden usar en redes neuronales, pero una de las más comunes en el uso de RNC es la función de activación de la unidad lineal rectificadora (ReLU, por sus siglas en inglés) que inserta todos los valores positivos a medida que ingresan, pero todos los valores

negativos se transforman a 0. Esto le permite ser más eficiente computacionalmente debido a su velocidad de convergencia. Sin embargo, puede afectar el entrenamiento si los valores ingresados tienden a cero [46]. Esto se puede apreciar en la Ecuación 2.2 y en la Figura 2.11.

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (2.2)$$



Figura 2.11. Gráfico de la función ReLU.

2.3.3 Capa de Max-Pooling

En esta capa, la operación encuentra el valor máximo entre una ventana de muestra más pequeña y pasa el valor como un resumen de características sobre esa área [47]. Esto reduce el tamaño de los datos, obteniendo el mismo tamaño que la ventana de muestra en la que se opera [48]. Con esto se obtiene una imagen más pequeña con las mejores características de la imagen anterior, y a su vez reduciendo la carga computacional [49]. Al igual que el kernel utilizado en capas convolucionales, las ventanas de las capas Max-Pooling pueden tener diferentes tamaños, pero el tamaño 2x2 es el más común. Un ejemplo de esto se puede apreciar en la Figura 2.12.

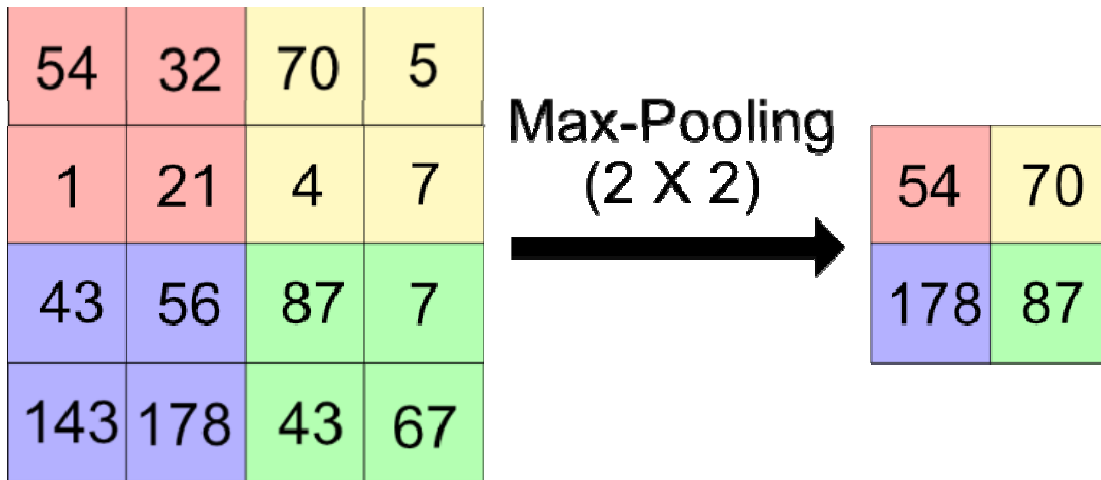


Figura 2.12. Ejemplo de la operación Max-Pooling.

3 Estado del arte

Distintos trabajos han tenido el mismo enfoque de la detección y clasificación de retinopatía diabética. Sin embargo, también es importante mencionar el trabajo de los autores en lo que respecta a la optimización de redes neuronales convolucionales. En este trabajo se utilizaron estas dos técnicas inteligentes para cumplir nuestro objetivo, y es importante señalar cuáles trabajos anteriores nos sirvieron de inspiración.

3.1 Detección de retinopatía diabética a través de redes neuronales convolucionales con meta plasticidad sináptica

El trabajo realizado por Víctor Vives-Boix y Daniel Ruiz-Fernández consiste en realizar experimentos para clasificar de forma binaria y multiclase la retinopatía diabética [50]. Los autores decidieron tomar arquitecturas de redes neuronales convolucionales existentes y entrenadas previamente para enfocarlas en la clasificación de retinopatía diabética, pero no se limitaron a solo utilizarlos, sino también a modificar dichos modelos para mejorar los resultados obtenidos.

De acuerdo con sus autores, se propuso un enfoque inspirado biológicamente sobre la meta plasticidad sináptica en redes neuronales convolucionales. Se sabe que este fenómeno biológico interfiere directamente tanto en el aprendizaje como en la memoria al reforzar sucesos menos comunes durante el proceso de aprendizaje. La meta plasticidad sináptica se ha incluido en la etapa de retro propagación de una operación de convolución para cada capa convolucional.

El método propuesto se evaluó mediante el uso de la base de datos APTOS 2019 con cuatro arquitecturas de redes neuronales convolucionales existentes. Los resultados muestran que las arquitecturas de redes neuronales convolucionales, incluida la meta plasticidad sináptica, mejoran la precisión obtenida. Los mejores resultados se han obtenido para la arquitectura InceptionV3 con meta plasticidad sináptica con una precisión del 95.56% para una clasificación binaria utilizando 3662 imágenes para el entrenamiento.

3.2 Método basado en aprendizaje profundo para el diagnóstico asistido por computadora de la retinopatía diabética

En el trabajo realizado por Omar Dekhil y otros, vieron un incremento importante de pacientes que perdían la visión a causa de la retinopatía diabética. En respuesta a este problema, presentaron una herramienta de diagnóstico asistido por computadora basada en redes neuronales convolucionales para clasificar imágenes de fondo de ojo en una de las cinco etapas de RD [51]. El método propuesto consistió en primero aplicar un preprocesamiento a la base de datos APTOD 2019. La RNC propuesta consta de cinco capas convolucionales, cada una con la función de activación ReLU. Finalmente, se agregaron tres capas completamente conectadas.

Se adoptó el aprendizaje de transferencia para minimizar el sobreajuste (popularmente conocido como *overfitting*) entrenando el modelo en un conjunto de datos más grande de 3,2 millones de imágenes (es decir, la base de datos ImageNet) antes del uso del modelo en la base de datos APTOS 2019. El modelo de red neuronal utilizado para realizar la transferencia de aprendizaje fue la VGG 2016.

El enfoque propuesto ha logrado una precisión de prueba del 77% (0.7700), lo que ofrece, de acuerdo con sus autores, una solución prometedora para un diagnóstico temprano exitoso y una estadificación de la RD de manera automatizada. No se realizaron pruebas para clasificación binaria.

3.3 Diseño de una arquitectura de red neuronal convolucional usando algoritmos genéticos.

El trabajo elaborado por Ashray Bhandare y Devinder Kaur consiste en el desarrollo de un algoritmo genético con el objetivo de descubrir arquitecturas de redes neuronales convolucionales que puedan funcionar sin la intervención de un técnico experto [52]. En otras palabras, se buscaba automatizar el proceso de la selección de una red neuronal convolucional utilizando un algoritmo genético que se encargara de optimizar los hiperparámetros de la red.

La base de datos utilizada en este trabajo fue la de MNIST, una base de datos de imágenes de los números del 0 al 9 escrito a mano. Es una extensión de

la base de datos “NIST” (National Institute of Standards and Technology), siglas de los autores originales, sin embargo, las imágenes fueron preprocesadas para facilitar el trabajo a los algoritmos de aprendizaje. Debido a esta modificación, se agregó la letra “M” al nombre de la base de datos.

La base de datos MNIST contiene 50000 imágenes etiquetadas que sirven para entrenar y validar la red neuronal convolucional, y otras 10000 imágenes que no están etiquetadas y normalmente son usadas para realizar pruebas. Es una de las bases de datos más populares, ya que permite a los investigadores estudiar sus métodos propuestos en un entorno controlado. En la Figura 3.1 se pueden apreciar algunos ejemplos extraídos de la base de datos MNIST. En este trabajo, el algoritmo genético se encarga de optimizar distintos hiperparámetros de la red neuronal convolucional. En la Tabla 3.1 se pueden apreciar los rangos utilizados de los hiperparámetros optimizados.



Figura 3.1. Ejemplos extraídos de la base de datos MNIST.

Tabla 3.1. Distintos hiperparámetros de la RNC con sus rangos.

Hiperparámetros	Rango
Número de épocas.	(0 - 127)
Tamaño de Batch.	(0 - 256)
Número de capas convolucionales.	(0 - 8)
Número de filtros para cada capa convolucional.	(0 - 64)
Tamaño del filtro para cada capa convolucional.	(0 - 8)
Funciones de activación para cada capa convolucional.	(Sigmoid, tanh, relu, linear)
Capa Max-Pooling después de cada capa convolucional.	(true, false)
Tamaño de kernel de cada capa Max-Pooling.	(0 - 8)
Número de capas ocultas	(0 - 8)
Número de neuronas en cada capa oculta.	(0 - 64)
Funciones de activación de cada capa oculta.	(Sigmoid, tanh, softmax, relu)
Optimizador	(Adagrad, Adadelta, RMS, SGD)

Para insertar todos los valores de la tabla en un cromosoma, los autores de este trabajo tomaron la decisión de crear un cromosoma binario, es decir, cada gen tendrá solo valores de 0 o 1. Para obtener los valores decimales de los rangos mostrados en la Tabla 3.1, se toman varios genes de acuerdo con el valor máximo que puede obtener cada hiperparámetro. Esto quiere decir que, para obtener los valores del número de épocas, son necesarios 7 genes mientras que para el tamaño de los filtros solo requieren 3 genes. Esta representación se puede observar en la Figura 3.2.

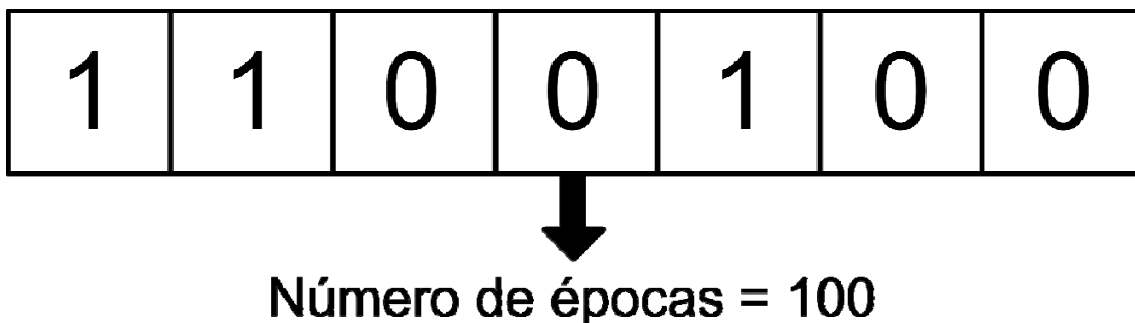


Figura 3.2. Ejemplo de la fracción del cromosoma utilizado para obtener las épocas.

El tamaño de la población del algoritmo genético es de 10 individuos, y la forma de evaluar el fitness de cada uno es mediante la precisión obtenida (valor entre 0 y 1) de la red neuronal convolucional (creada tomando los hiperparámetros del cromosoma) con las pruebas de la base de datos MNIST.

Como propiedades del algoritmo genético, para la selección se realiza la operación de torneo por selección tomando dos individuos, el cruce se realiza en un solo punto con una probabilidad de 50%, y para la mutación la operación de multipunto con probabilidad del 80% mutando solo el 10% de los genes.

El algoritmo genético se ejecutó 10 veces obteniendo como resultado más alto una precisión de 0.9919. Solo se utilizó la base de datos MNIST, pero los autores concluyeron que los algoritmos genéticos tienen el potencial de generar arquitecturas de red neuronal convolucional exitosas de forma automática.

4 Planteamiento del Problema

En el presente trabajo de tesis se pretende desarrollar un nuevo modelo de red neuronal convolucional para la detección y clasificación de retinopatía diabética utilizando las técnicas inteligentes mencionadas en el marco teórico y tomando como referencia los trabajos expuestos en el estado del arte. Esto por medio de una base de datos llamada APTOS 2019 que será de utilidad al momento de realizar el entrenamiento y validación de la red neuronal y de finalmente probarla. Esto se aprecia mejor en la Figura 4.1.

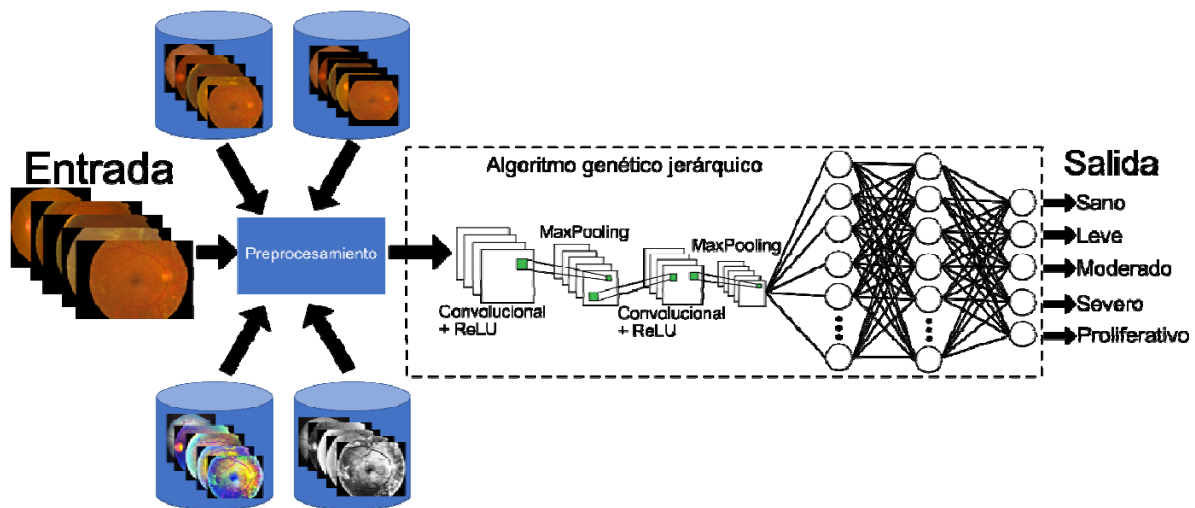


Figura 4.1. Propuesta del método.

Como se puede observar en la Figura 4.1, se utilizaron 4 métodos de preprocesamiento distintos para la entrada de datos de la red neuronal. Estos métodos se describen a detalle en el siguiente capítulo. Los hiperparámetros de esta red neuronal se optimizan con el uso de un algoritmo genético jerárquico. Los métodos de preprocesamiento, de la misma forma que los hiperparámetros y rangos de sus valores del algoritmo genético jerárquico se describen a detalle en el siguiente capítulo.

5 Desarrollo

Este capítulo se divide en tres secciones, siendo la primera la información de la base de datos APTOS 2019 mencionada en el capítulo anterior. En el segundo, el trabajo realizado en los preprocesamientos aplicados a la base de datos, y el tercero en la obtención de un modelo de redes neuronales convolucionales utilizando un algoritmo genético jerárquico.

5.1 Base de datos APTOS 2019

La base de datos APTOS 2019 está compuesta por 5590 imágenes donde 3662 están etiquetadas y se utilizan para entrenar y validar la RNC [53]. Las imágenes están divididas en 5 clases, tal como se observa en la Tabla 5.1. En la Tabla 5.2 se puede observar la distribución de las imágenes entre imágenes de la retina sanas e imágenes con retinopatía diabética para la realización de una clasificación binaria.

Tabla 5.1. Número de imágenes por clase de la base de datos APTOS 2019.

Clases de retinopatía diabética	Número de imágenes
Retina sana	1805
No Proliferativo (NP) Leve	370
NP Moderado	999
NP Severo	295
Proliferativo	193

Tabla 5.2. Número de imágenes divididas entre sanas y con RD de la base de datos APTOS 2019

Tipo de imagen	Número de imágenes
Retina sana	1805
Retina con RD	1857

Las imágenes contienen mucho ruido y diferentes tamaños, como se puede observar en la Figura 5.1. Es una de las bases de datos con mayor cantidad de imágenes que contempla las 5 clasificaciones de la RD, estando después de la base de datos EyePACS usado en el concurso de Kaggle en 2015 [54].

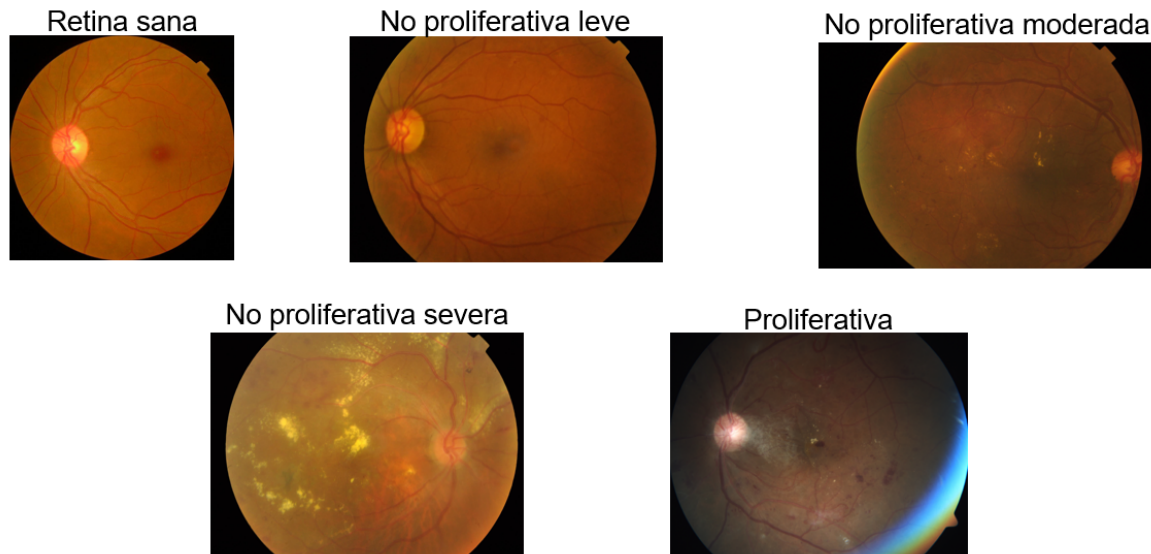


Figura 5.1. Ejemplos de imágenes extraídas de APTOS 2019.

5.2 Preprocesamiento aplicado a la base de datos

Antes de aplicar el preprocesamiento a las imágenes, se desarrollaron dos modelos de RNC para poder comparar los preprocesamientos y saber cuál ofrece mejores resultados para la detección de RD. El primer modelo de RNC requiere que las imágenes tengan 200 píxeles de ancho y 200 píxeles de alto. Tiene una profundidad de tres convoluciones, es decir, una capa convolucional seguida de una función de activación ReLU y una capa Max-Pooling. Esto se repite dos veces más con la diferencia de que la primer y segunda convolucional tienen 32 filtros mientras que el tercero tiene 64 filtros. Se agregan dos capas completamente conectadas con 512 neuronas cada una. Finalmente, está la capa de salida, que tiene 2 o 5 neuronas según el caso de estudio. En la Figura 5.2 se puede ver más a detalle la estructura de este modelo.

Tamaño de imagen de 200 x 200

Capa convolucional con tamaño de filtro 3x3, 32 filtros

ReLU, MaxPooling

Capa convolucional con tamaño de filtro 3x3, 32 filtros

ReLU, MaxPooling

Capa convolucional con tamaño de filtro 3x3, 64 filtros

ReLU, MaxPooling, Dropout 50%

Capa completamente conectada con 512 neuronas

ReLU, MaxPooling, Dropout 50%

Capa completamente conectada con 512 neuronas

ReLU

Capa completamente conectada con 2 o 5 neuronas

Figura 5.2. Estructura del modelo 1 de RNC.

El segundo modelo tiene una profundidad de 10 convolucionales, con la diferencia de que los filtros se duplican cada dos convolucionales, es decir, las dos primeras capas convolucionales tienen 32 filtros, la tercera y la cuarta tienen filtros 64, etc. Se agregan dos capas completamente conectadas con 1024 neuronas cada una. Finalmente, está la capa de salida, que tiene 2 o 5 neuronas según el caso de estudio. En la Figura 5.3 se puede ver más a detalle la estructura de este modelo.

Tamaño de imagen de 512 x 512

Capa convolucional con tamaño de filtro 3x3, 32 filtros

ReLU, MaxPooling

Capa convolucional con tamaño de filtro 3x3, 32 filtros

ReLU, MaxPooling

Capa convolucional con tamaño de filtro 3x3, 64 filtros

ReLU, MaxPooling

Capa convolucional con tamaño de filtro 3x3, 64 filtros

ReLU, MaxPooling

Capa convolucional con tamaño de filtro 3x3, 128 filtros

ReLU, MaxPooling

Capa convolucional con tamaño de filtro 3x3, 128 filtros

ReLU, MaxPooling

Capa convolucional con tamaño de filtro 3x3, 256 filtros

ReLU, MaxPooling

Capa convolucional con tamaño de filtro 3x3, 256 filtros

ReLU, MaxPooling

Capa convolucional con tamaño de filtro 3x3, 512 filtros

ReLU, MaxPooling

Capa convolucional con tamaño de filtro 3x3, 512 filtros

ReLU, MaxPooling, Dropout 50%

Capa completamente conectada con 1024 neuronas

ReLU, MaxPooling, Dropout 50%

Capa completamente conectada con 1024 neuronas

ReLU

Capa completamente conectada con 2 o 5 neuronas

Figura 5.3. Estructura del modelo 2 de RNC.

5.2.1 Preprocesamiento 1

El primer preprocesamiento es el más importante de los cuatro porque también es utilizado como base por los tres preprocesamientos restantes.

Este preprocesamiento es muy utilizado [55] y consiste en tomar la retinografía y tomar cada valor de los píxeles que la conforman (siendo cada uno valores enteros entre 0 y 255). Si el píxel seleccionado tenía un valor menor a 20, entonces equivalía a 0, valiendo 1 en caso contrario, de la misma forma que ese observa en la Ecuación 5.1.

$$f(x) = \begin{cases} 0, & x < 20 \\ 1, & x \geq 20 \end{cases} \quad (5.1)$$

Estos nuevos valores se almacenan en una nueva imagen en la misma ubicación del píxel original. Como resultado, tenemos una imagen binaria de la retina. Ahora debemos buscar la posición del objeto más grande para obtener el píxel central de la retina con su alto y ancho. Ya con estos datos, los utilizamos para extraer la retina en la imagen original, dando como resultado, la retina sin el ruido que contenían los píxeles oscuros fuera del área de la retina. Se pueden apreciar ejemplos del resultado en la Figura 5.4.

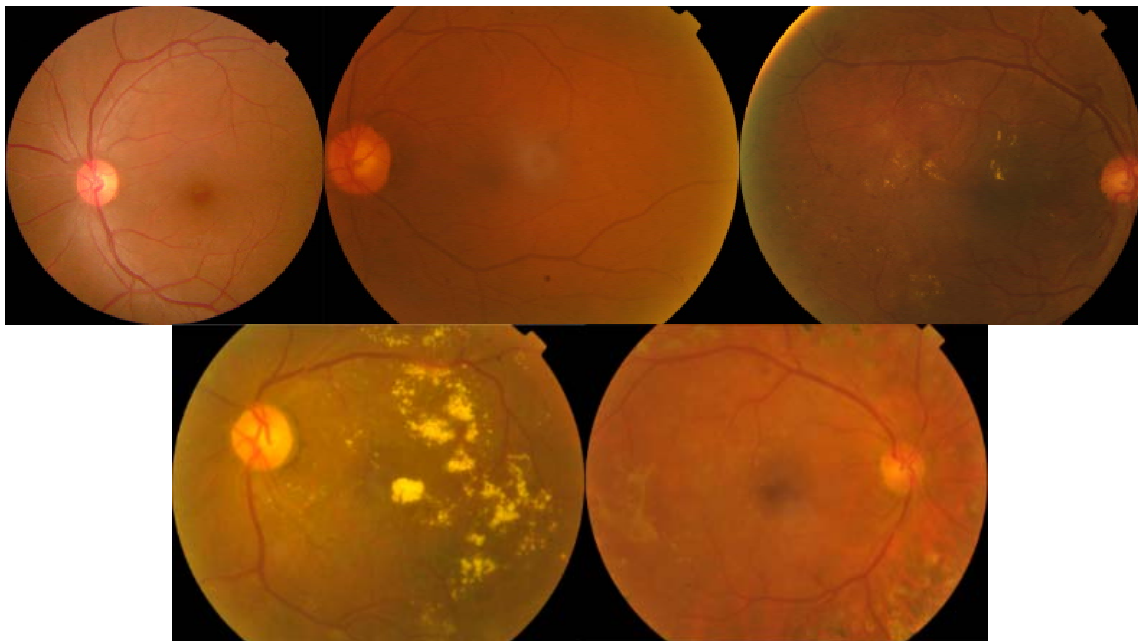


Figura 5.4. Ejemplos utilizando el preprocesamiento 1.

5.2.2 Preprocesamiento 2

Este preprocesamiento es parecido al anterior, con la diferencia de que se busca mantener las imágenes con la misma cantidad de píxeles en el ancho y alto agregando píxeles negros ya sea en la parte inferior y superior o los costados. Esto con la intención de no deformar la imagen antes de insertarla a la RNC. Utilizado por [56].

Primero se debe realizar el preprocesamiento 1 que se detalló anteriormente. Ya con esta imagen, se extrae el alto y ancho de la imagen. Comparamos ambos valores buscando el mayor. Se pueden apreciar ejemplos del resultado en la Figura 5.5.

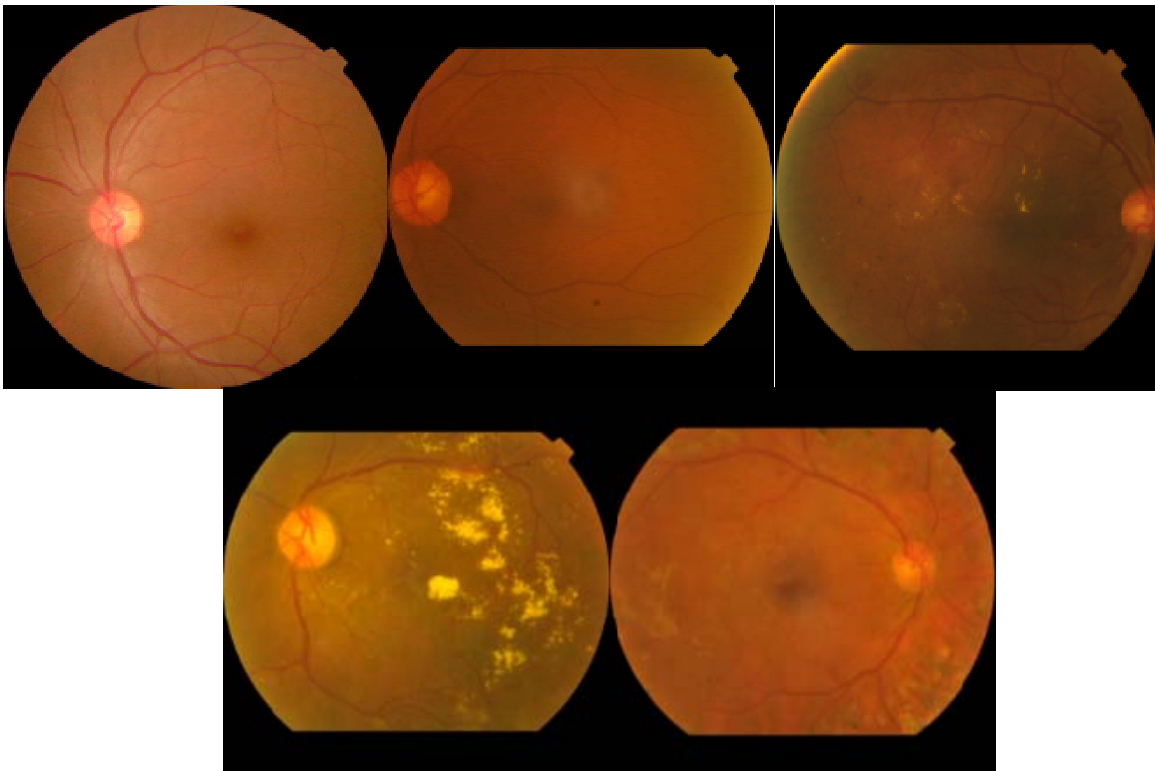


Figura 5.5. Ejemplos utilizando el preprocesamiento 2.

5.2.3 Preprocesamiento 3

Para realizar este preprocesamiento, se aplica todo lo explicado en el preprocesamiento 1. Luego se extraen los canales rojo, verde y azul y se aplica una ecualización de histograma a cada uno. Después se unieron los tres canales para formar una sola imagen. Se pueden apreciar ejemplos del resultado en la Figura 5.6.

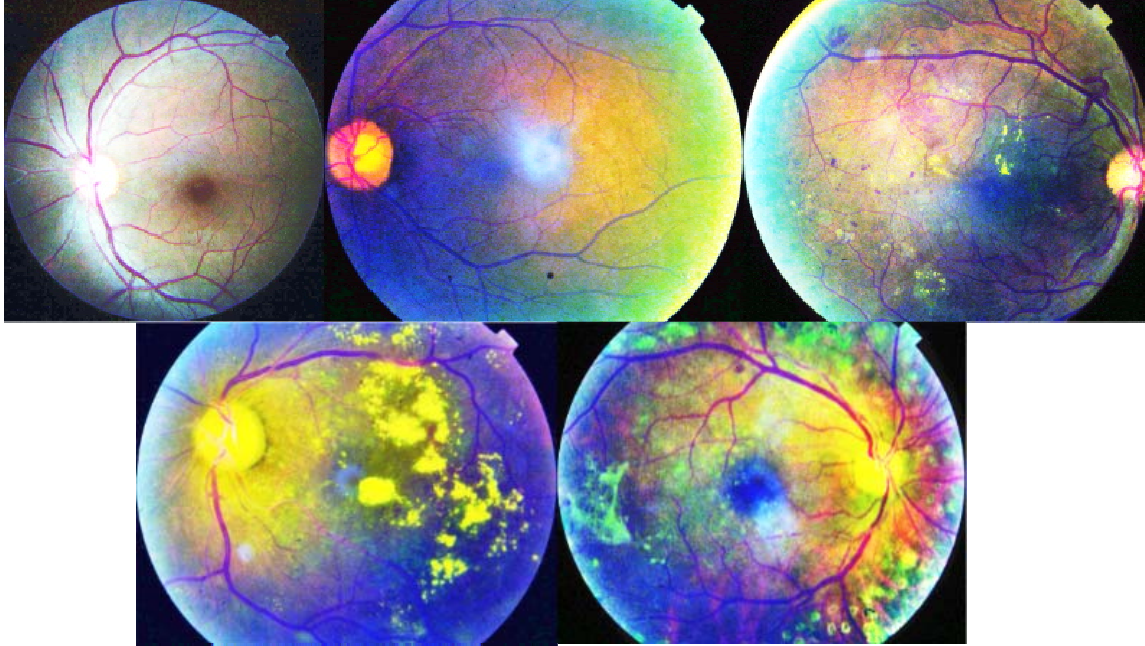


Figura 5.6. Ejemplos utilizando el preprocesamiento 3.

5.2.4 Preprocesamiento 4

Algunos autores dicen que el canal verde de las retinografías contenía la información más importante para la detección de RD [57]. Para realizar este preprocesamiento, se extrajo dicho canal de la imagen original y se aplicó una ecualización de histograma. Esto nos dio como resultado una imagen a escala de grises siendo el único preprocesamiento que tiene una capa de profundidad en vez de tres. Se pueden apreciar ejemplos del resultado en la Figura 5.7.

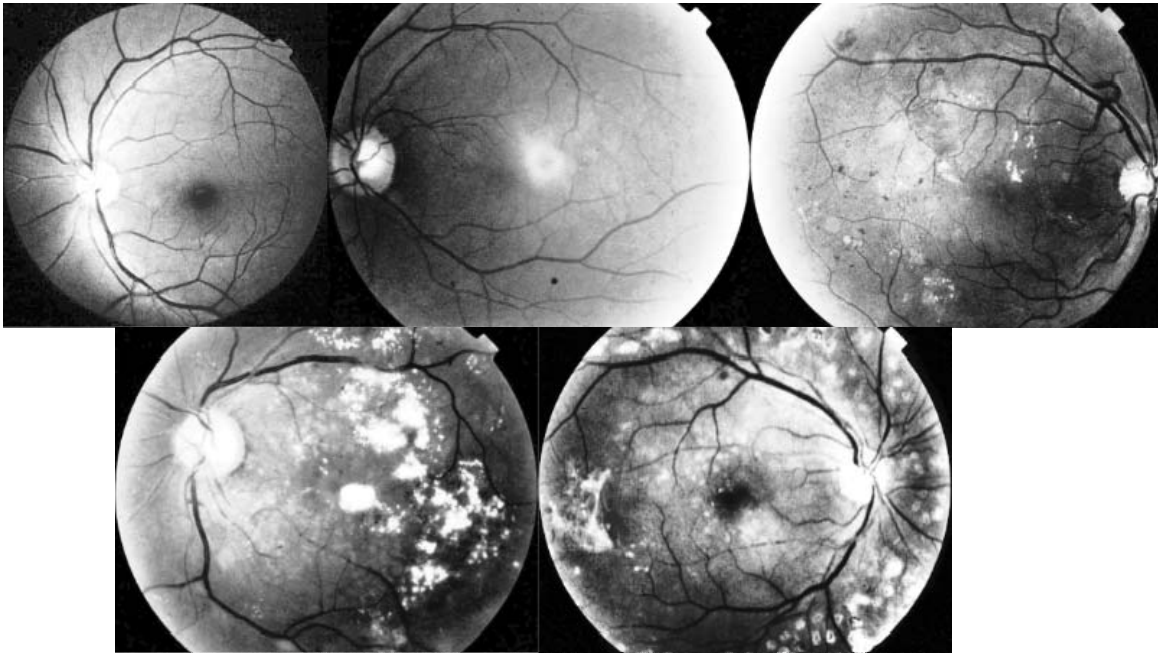


Figura 5.7. Ejemplos utilizando el preprocesamiento 4.

5.3 Algoritmo genético jerárquico

Como se comentó previamente, para optimizar la red neuronal convolucional se utilizó un algoritmo genético jerárquico. En esta sección se va a explicar en detalle la función de cada uno de los genes de los cromosomas del algoritmo genético.

5.3.1 Genes de los cromosomas

Como se mencionó anteriormente, los cromosomas que son utilizados en los algoritmos genéticos jerárquicos tienen genes de control y genes paramétricos. Los datos dentro de cada generación tienen un propósito diferente. Primero, es necesario conocer las funciones de los genes de control para poder comprender qué es lo que hacen los genes paramétricos.

5.3.1.1 Genes de control

El primer gen de control tiene un valor entero que decide cuántas capas convolucionales se usarán en el modelo de RNC, con un valor mínimo de 3 y un máximo de 5. El segundo gen también tiene un valor entero y especifica cuántas capas completamente conectadas tendrá la red, con un valor mínimo de 3 y un máximo de 6. Las capas estarán después de las capas convolucionales y antes de la capa de clasificación. Finalmente, el tercer gen puede tener valores enteros de

10, 20, 30, 40, 50, 60, 70, 80, 90 o 100, y representa el número de épocas. En la Figura 5.8 se puede observar una representación gráfica del cromosoma con genes de control.

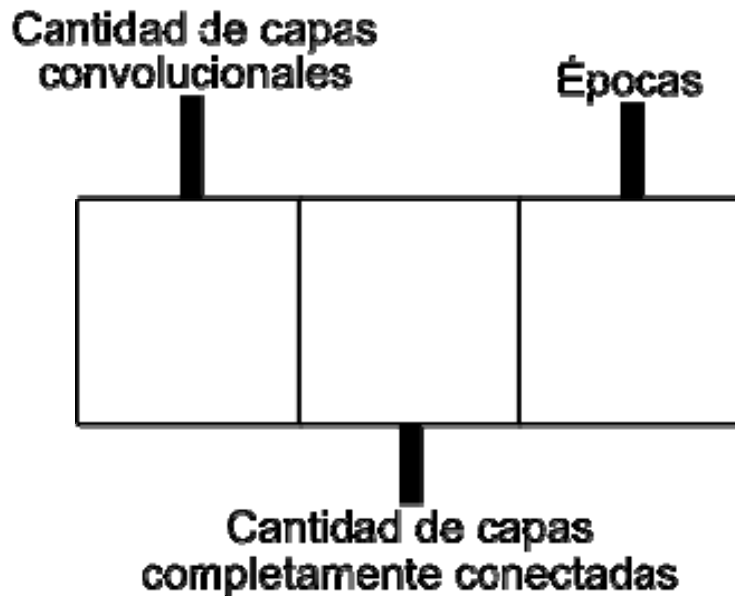


Figura 5.8. Genes de control.

5.3.1.2 Genes paramétricos para capas convolucionales

Como se mencionó anteriormente, el primer gen de los genes de control contiene el valor sobre cuántas capas convolucionales tendrá la red neuronal convolucional. El primer gen de los genes paramétricos para capas convolucionales tiene un valor entero de 1, 2 o 3. El valor 1 significa que será solo una capa convolucional con su función de activación ReLU. El valor 2 significa que tendrá lo mencionado en el valor 1 y una capa Max-Pooling con un kernel de 2x2 y un valor de stride de 2. Finalmente, si el valor es 3 serán las capas mencionadas con el valor 2 y una capa con probabilidad de Dropout.

El segundo gen paramétrico contiene un valor entero que equivaldrá al alto y ancho del tamaño del filtro para la capa convolucional. Los valores pueden ser 3, 4 o 5. El tercer gen paramétrico contiene un valor entero que representa el número de filtros de la capa convolucional. La cantidad mínima y máxima de filtros dependerá de la posición en la que se encuentre la capa convolucional. Si es la capa convolucional 1 o 2, entonces la cantidad de filtros mínima y máxima será de

16 y 32, respectivamente. Si es la capa convolucional 3 o 4, entonces la cantidad de filtros mínima y máxima será de 64 y 128, respectivamente. Finalmente, si es la capa convolucional 5, entonces la cantidad de filtros mínima y máxima será de 256 y 512, respectivamente.

En el cuarto y último gen paramétrico se encuentra el valor de probabilidad de Dropout que solo se utilizará en caso de que el primer gen paramétrico tenga un valor de 3. El valor de Dropout es un número entero y puede ser 10, 20 30, 40 o 50. Antes de insertar este valor a la RNC, el valor del Dropout se divide entre 100. Una representación gráfica de los genes paramétricos para las capas convolucionales se puede observar en la Figura 5.9.

Cantidad de capas convolucionales

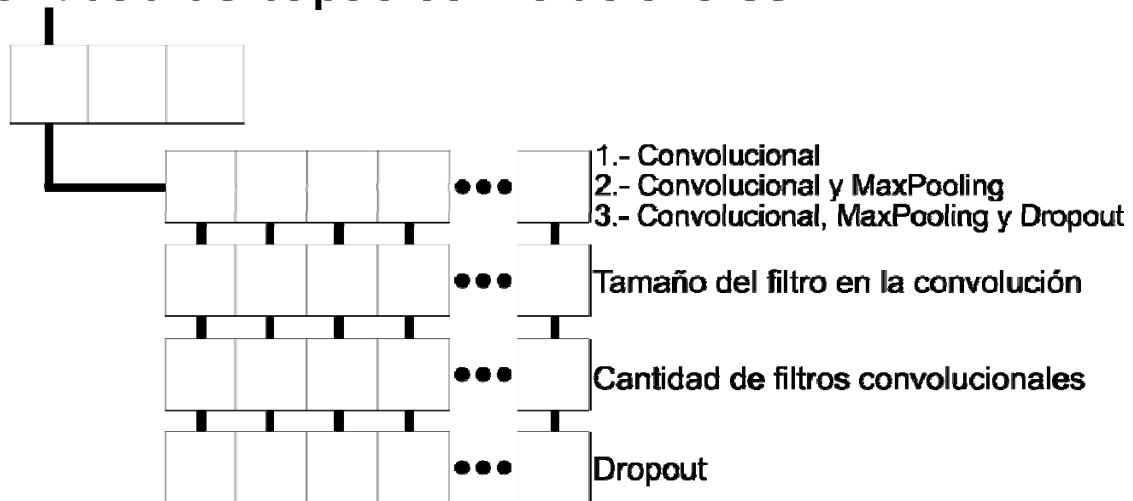


Figura 5.9. Genes paramétricos para capas convolucionales.

5.3.1.3 Genes paramétricos para capas completamente conectadas

Como se mencionó anteriormente, el segundo gen de los genes de control contiene el valor sobre cuántas capas completamente conectadas tendrá la red neuronal convolucional. El primer gen de los genes paramétricos para capas completamente conectadas tiene un valor entero de 1 o 2. El valor 1 significa que será solo una capa completamente conectada con su función de activación ReLU. El valor 2 significa que tendrá lo mencionado en el valor 1 y una capa con probabilidad de Dropout.

El segundo gen paramétrico contiene un valor entero que representa el número de neuronas de la capa completamente conectada. La cantidad mínima y máxima de neuronas dependerá de la posición en la que se encuentre la capa completamente conectada. Si es la capa completamente conectada 1 o 2, entonces la cantidad de neuronas mínima y máxima será de 64 y 128, respectivamente. Si es la capa completamente conectada 3 o 4, entonces la cantidad de neuronas mínima y máxima será de 128 y 256, respectivamente.

En el tercer y último gen paramétrico se encuentra el valor de probabilidad de Dropout que solo se utilizará en caso de que el primer gen paramétrico tenga un valor de 2. El valor de Dropout es un número entero y puede ser 10, 20 30, 40 o 50. Antes de insertar este valor a la RNC, el valor del Dropout se divide entre 100. Una representación gráfica de los genes paramétricos para las capas convolucionales se puede observar en la Figura 5.10.

Cantidad de capas completamente conectadas

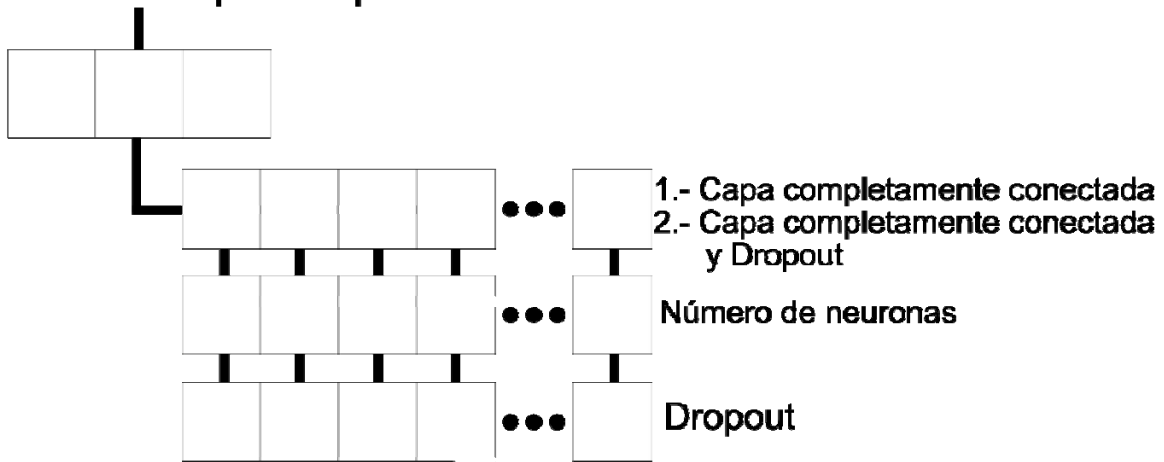


Figura 5.10. Genes paramétricos para las capas completamente conectadas.

La función objetivo en todos los experimentos de este trabajo de Tesis se puede observar en la Ecuación 5.2.

$$Precisión = \frac{VP + VN}{VP + FP + VN + FN} \tag{5.2}$$

Donde *VP* representa a los Verdaderos Positivos, *VN* a los Verdaderos Negativos, *FP* a los Falsos Positivos y *FN* a los Falsos Negativos. El objetivo de la

experimentación del algoritmo genético jerárquico es la de maximizar el valor de la precisión.

Los métodos de preprocesamiento presentados fueron elaborados utilizando Matlab 2020b y las herramientas “Image Processing Toolbox”. Para el desarrollo del algoritmo genético jerárquico, se tomó la decisión de cambiar el lenguaje de programación a Python debido a las compatibilidades con la tarjeta gráfica a la hora de entrenar los modelos de redes neuronales artificiales [58].

6 Resultados

En este capítulo se mostrarán los resultados obtenidos por los experimentos realizados sobre el preprocesamiento y el algoritmo genético jerárquico con sus respectivas pruebas estadísticas. Cada experimento en esta sección fue ejecutado 30 veces con la intención de obtener los valores de la media y la desviación estándar. Estos datos se utilizan a lo largo de la sección para hacer pruebas de hipótesis comparando distintos experimentos.

En algunos experimentos también se tomaron en cuenta los valores máximos y mínimos conseguidos. Cada una de las ejecuciones se realizaron con 10 épocas por limitaciones de tiempo y utilizando el Optimizador Adaptive Moment Estimation (Adam). Las imágenes del conjunto de datos APTOS 2019 se distribuyeron como se observa en la Tabla 6.1 para cada uno de los experimentos de este trabajo de tesis.

Tabla 6.1. distribución de las imágenes de la base de datos APTOS 2019.

Tipos de imágenes	Porcentaje de imágenes	Número de imágenes
Entrenamiento	72%	2637
Validación	8%	293
Pruebas	20%	732

6.1 Experimentos realizados sobre los preprocesamientos

Se tomaron en cuenta dos casos de estudio, siendo el primero un caso binario y el segundo, un multiclase. En ambos se usaron dos modelos de redes neuronales convolucionales mencionados anteriormente para realizar los experimentos. Se realizaron 16 experimentos (8 para el caso de estudio binario y los otros 8 para el caso de estudio multiclase) y cada uno fue ejecutado 30 veces, con el fin de obtener la precisión media y la desviación estándar. También se agregaron los valores máximos y mínimos obtenidos en cada experimento. Los 8 experimentos de cada caso de estudio son en realidad los dos modelos de RNC mencionados anteriormente, aplicando los 4 preprocesamientos a cada uno.

6.1.1 Caso de estudio binario

Aunque la base de datos APTOS 2019 está dividido en 5 clases, también sirve para realizar una red neuronal convolucional con una salida binaria, es decir, la retina tiene o no retinopatía diabética, justo como se explicó en la sección anterior. Los resultados obtenidos para este caso de estudio se pueden apreciar en la Tabla 6.2.

Tabla 6.2. Resultados del caso de estudio binario.

#	Profundidad de la RNC	Método	Precisión media	Desviación estándar	Valor máximo	Valor mínimo
#1	3	1	0.9461	0.00739364	0.9461	0.9154
#2	3	2	0.9401	0.00824493	0.9523	0.9154
#3	3	3	0.9127	0.00849006	0.9270	0.8881
#4	3	4	0.9205	0.00828794	0.9311	0.8984
#5	10	1	0.9012	0.07387703	0.9536	0.5375
#6	10	2	0.8893	0.08135433	0.9591	0.5293
#7	10	3	0.9011	0.08574371	0.9659	0.5539
#8	10	4	0.8954	0.07383870	0.9523	0.6508

De acuerdo con los resultados obtenidos, la precisión media más alta se encuentra en el experimento #2 donde se utiliza una RNC de profundidad 3 y el segundo preprocesamiento. El experimento #1 tiene la desviación estándar más baja. El valor máximo más alto fue obtenido por el experimento #7 y el mínimo fue obtenido por el experimento #6.

6.1.2 Caso de estudio multiclase

Para este caso de estudio, se tomaron en cuenta las 5 clasificaciones en la que está etiquetada la base de datos originalmente, es decir, 5 clasificaciones. Los resultados obtenidos se pueden apreciar en la Tabla 6.3.

Tabla 6.3. Resultados del caso de estudio multiclase

#	Profundidad de la RNC	Método	Precisión media	Desviación estándar	Valor máximo	Valor mínimo
#1	3	1	0.7218	0.00724198	0.7381	0.7053
#2	3	2	0.7401	0.01051900	0.7585	0.7176
#3	3	3	0.7214	0.01474537	0.7490	0.6917
#4	3	4	0.7193	0.01299899	0.7408	0.6726
#5	10	1	0.7091	0.03506277	0.7394	0.5866
#6	10	2	0.7204	0.02397598	0.7585	0.6712
#7	10	3	0.6879	0.10015835	0.7449	0.3042
#8	10	4	0.6907	0.04782971	0.7449	0.5512

De acuerdo con los resultados obtenidos, la precisión media más alta se encuentra en el experimento #2 donde se utiliza una RNC de profundidad 3 y el segundo preprocesamiento. El experimento #1 tiene la desviación estándar más baja. El valor máximo más alto fue obtenido por el experimento #6 y el mínimo más bajo fue obtenido por el experimento #7.

6.1.3 Comparación de resultados

Para tener un mayor entendimiento de los datos, se realizaron distintas pruebas estadísticas para determinar cuál preprocesamiento ofrece mejores resultados.

6.1.3.1 Prueba de rangos con signo de Wilcoxon

La primera prueba realizada fue la Prueba de Rangos con Signo de Wilcoxon ya que se puede observar en las Tablas 6.2 y 6.3 que hay diferencia en los resultados de las RNC usando profundidad 3 y 10. Para ver si el modelo con profundidad 10 ofrecía peores resultados, se compararon los resultados de ambos modelos. La diferencia de estos resultados se puede observar con mayor claridad en la Figura 6.1.

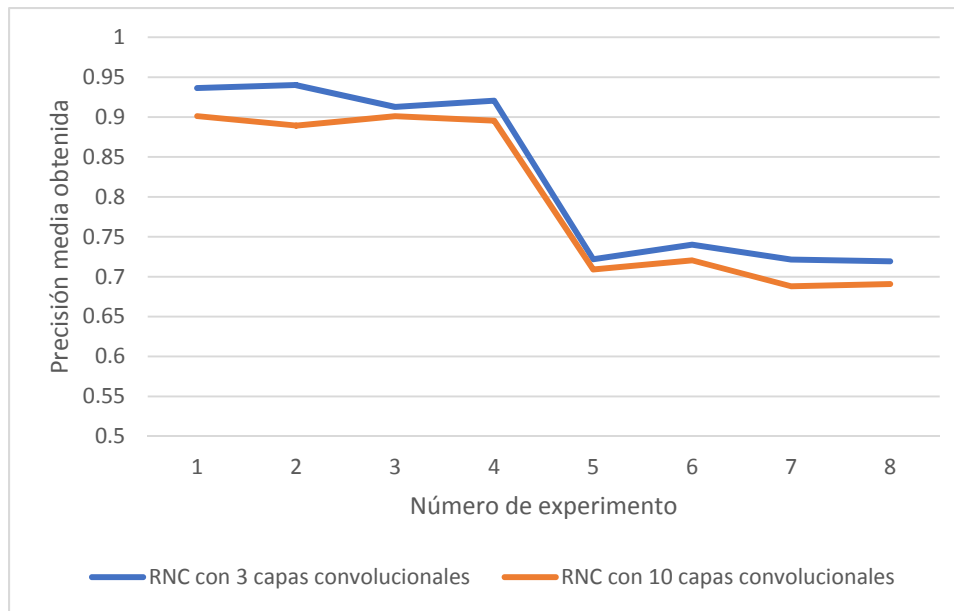


Figura 6.1. Comparación de resultados entre los modelos de RNC con 3 y 10 capas convolucionales.

La hipótesis nula fue “El modelo de RNC con profundidad 10 no reduce la precisión media” y nuestra afirmación se encontraba en la hipótesis alternativa, siendo “El modelo de RNC con profundidad 10 reduce la precisión media”. La Tabla 6.4 muestra los rangos obtenidos por la prueba.

Tabla 6.4. Resultados obtenidos por la prueba de rangos con signo de Wilcoxon

#	RNC Profundidad 3	RNC Profundidad 10	Diferencia	Rangos	Rangos con signos
1	0.9363	0.9012	0.0351	7	7
2	0.9401	0.8893	0.0508	8	8
3	0.9127	0.9011	0.0116	1	1
4	0.9205	0.8954	0.0251	4	4
5	0.7218	0.7091	0.0127	2	2
6	0.7401	0.7204	0.0197	3	3
7	0.7214	0.6879	0.0335	6	6
8	0.7193	0.6907	0.0286	5	5

Sumando los rangos positivas y negativas, el más bajo de ellos obtuvo un valor de 0. De acuerdo con los valores críticos de la Prueba de Rangos con Signo de Wilcoxon, con un valor de alfa de 0.05, para rechazar esta hipótesis el valor debe ser menor que 6. Como se puede observar, 0 es menor que 6, por lo que se

rechaza la hipótesis nula, lo que significa que, con un nivel de significancia del 5%, tenemos suficiente evidencia para respaldar la afirmación de que los modelos de RNC para este trabajo pueden reducir sus precisiones medias mediante el uso de profundidad 10.

6.1.3.2 Prueba de hipótesis para el caso de estudio binario

De acuerdo con la prueba anterior, los modelos de RNC que utilizan 3 capas convolucionales ofrecieron mejores resultados, por lo que se realizó una prueba de hipótesis comparando los dos mejores resultados del caso de estudio binario: el preprocesamiento 2 y 1. Nuestra afirmación es “La precisión media del preprocesamiento 2 es mayor que la precisión media del Preprocesamiento 1 para el caso de estudio binario” siendo nuestra hipótesis alternativa y buscamos rechazar la hipótesis nula.

Para lograr esto, el valor crítico debe ser menor que -1.96 porque usamos un valor Alfa de 0.05. Nuestra estadística de prueba estandarizada fue -1,8564, lo que significa que se falla al rechazar la hipótesis nula. Por lo tanto, no hay evidencia suficiente para respaldar la afirmación, con un nivel de significancia del 5%, de que la precisión media del Preprocesamiento 2 es mayor que la precisión media del Preprocesamiento 1 para el caso de estudio binario. En la Figura 6.2 se puede observar una gráfica de esta prueba de hipótesis.

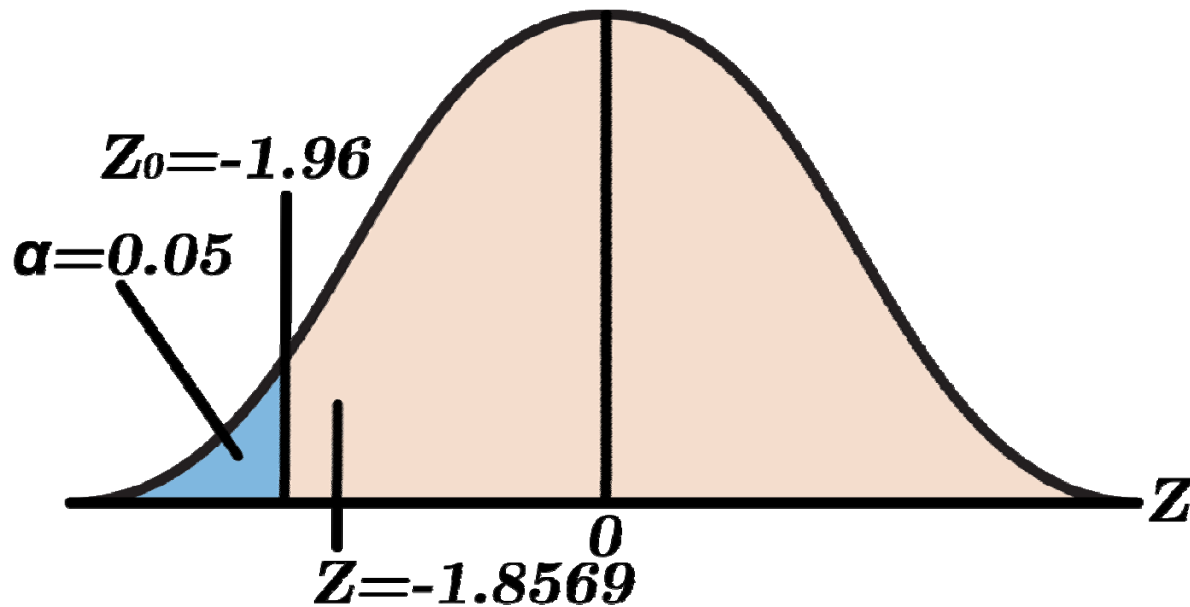


Figura 6.2. Prueba de hipótesis para el caso de estudio binario.

6.1.3.3 Prueba de hipótesis para el caso de estudio multiclase

Al igual que en el caso de estudio binario, los mejores resultados se obtuvieron utilizando el preprocesamiento 2 y 1. Por tanto, nuestra afirmación es muy similar a la utilizada en el caso de estudio binario: “La precisión media del preprocesamiento 2 es mayor que la precisión media del preprocesamiento 1 para el caso de estudio multiclase”.

Para lograr esto, el valor crítico debe ser menor que -1.96 porque usamos un valor α de 0.05. Nuestro estadístico de prueba estandarizado obtuvo $-7,8358$, lo que significa que la hipótesis nula fue rechazada, por lo que hay suficiente evidencia con un 5% de nivel de significancia para respaldar la afirmación de que la precisión media del Preprocesamiento 2 es mayor que la precisión media del preprocesamiento 1 para el caso de estudio multiclase. En la Figura 6.3 se puede observar una gráfica de esta prueba de hipótesis.

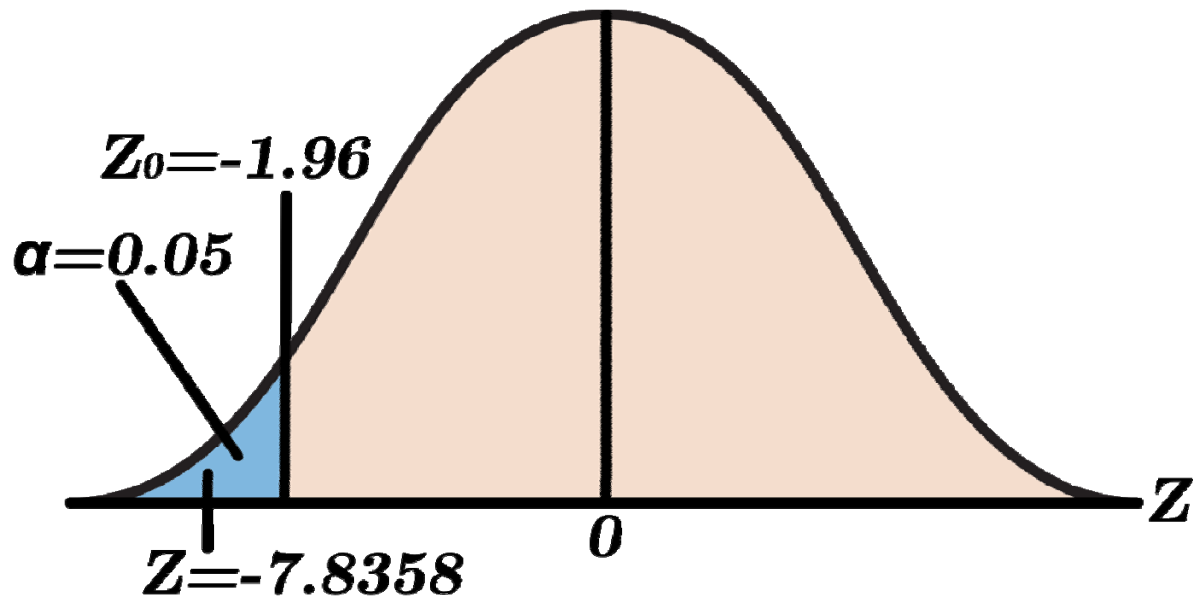


Figura 6.3. Prueba de hipótesis para el caso de estudio multiclase.

6.2 Experimentos realizados sobre el algoritmo genético jerárquico

Al igual que en la experimentación del preprocesamiento, se realizaron dos casos de estudio para la experimentación sobre el algoritmo genético, siendo: caso de estudio binario y caso de estudio multiclase. En ambos casos de estudio se utilizaron los mismos cromosomas y genes detallados en la sección anterior. Cada experimento se ejecutó 30 veces. En ambos casos de estudio se utilizó el preprocesamiento 2 y se dividió de la misma manera que en la experimentación anterior.

Al igual que en los experimentos realizados en el preprocesamiento, se utilizó el optimizador Adam, sin embargo, en estos experimentos no está fijo el valor de 10 épocas porque va a depender del valor del gen de control.

Se usaron las mismas propiedades para algoritmo genético jerárquico en cada experimento, es decir, la población inicial fue de 10 individuos con una probabilidad de selección de 80%, 50% de probabilidad para cruzar y un 80% de probabilidad de mutación, estos valores se tomaron de acuerdo con los resultados de los trabajos de otros autores [58]. Los mejores hijos reemplazan a los peores padres. Para el cruce se utiliza el cruce binario, y para la mutación se toma un renglón de manera aleatoria de cada gen paramétrico y se altera de la misma manera que se detalla en la sección anterior. El valor *fitness* de cada cromosoma se calcula con la precisión obtenida por el correspondiente modelo de RNC usando las imágenes de prueba.

6.2.1 Caso de estudio binario

En la Tabla 6.5 se pueden observar los resultados obtenidos para este caso de estudio tomando en cuenta la información detallada anteriormente.

Tabla 6.5. Resultados obtenidos por el AGJ para el caso de estudio binario.

Número de ejecución	Mejor fitness obtenido	Número de ejecución	Mejor fitness obtenido	Número de ejecución	Mejor fitness obtenido
1	0.9658	11	0.9713	21	0.9658
2	0.9399	12	0.9563	22	0.9686
3	0.9727	13	0.9590	23	0.9658
4	0.9563	14	0.9693	24	0.9549
5	0.9720	15	0.9618	25	0.9563
6	0.9597	16	0.9699	26	0.9631
7	0.9645	17	0.9713	27	0.9706
8	0.9781	18	0.9727	28	0.9686
9	0.9652	19	0.9631	29	0.9699
10	0.9604	20	0.9631	30	0.9747

Como se puede observar en la ejecución #8, el mejor modelo de RNC obtenido por el algoritmo genético jerárquico logró una precisión de 0.9781 utilizando 100 épocas. Se obtuvo una precisión media de 0.9650. Con la información de la Tabla 6.5 también se obtuvo una desviación estándar de 0.007665. En la Figura 6.4 se pueden observar las capas e hiperparámetros del modelo de RNC.

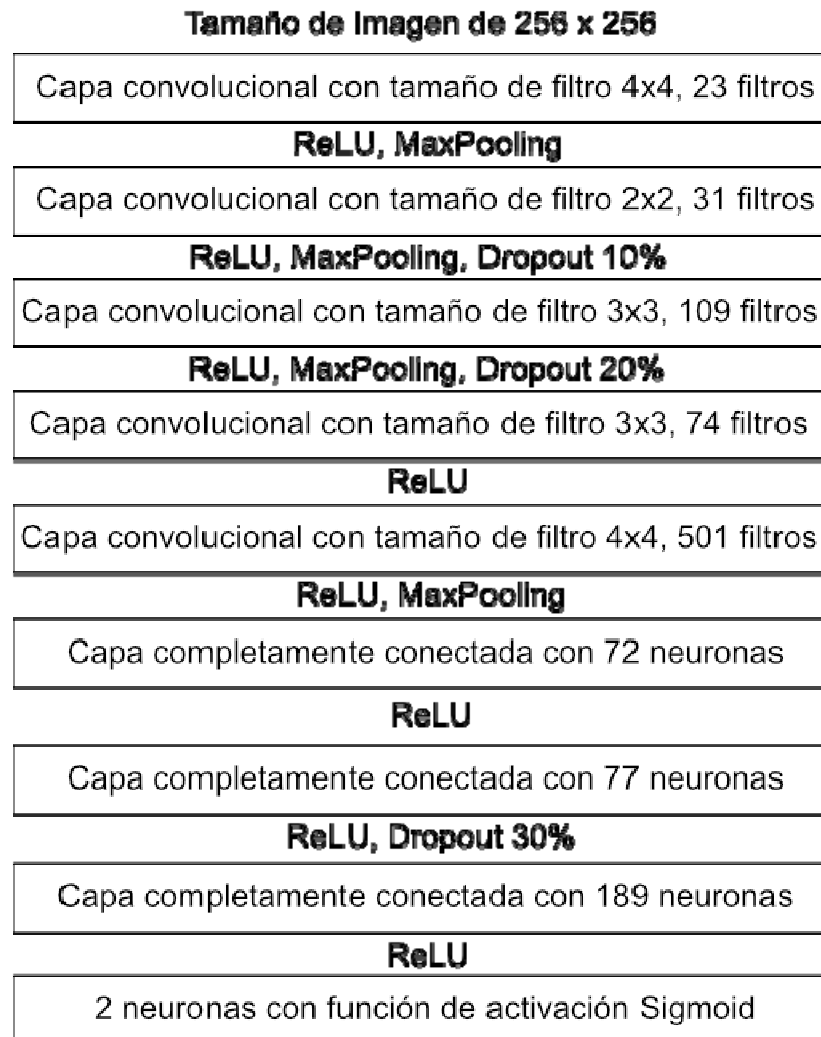


Figura 6.4. Modelo de RNC que obtuvo la mejor precisión para el caso de estudio binario.

6.2.2 Caso de estudio multiclase

En la Tabla 6.6 se pueden observar los resultados obtenidos para este caso de estudio tomando en cuenta la información detallada anteriormente.

Tabla 6.6. Resultados obtenidos por el AGJ para el caso de estudio multiclase.

Número de ejecución	Mejor fitness obtenido	Número de ejecución	Mejor fitness obtenido	Número de ejecución	Mejor fitness obtenido
1	0.7558	11	0.7639	21	0.7544
2	0.7748	12	0.7762	22	0.7557
3	0.7571	13	0.7612	23	0.7680
4	0.7694	14	0.7503	24	0.7366
5	0.7598	15	0.7612	25	0.7748
6	0.7667	16	0.7476	26	0.7571
7	0.7571	17	0.7571	27	0.7735
8	0.7557	18	0.7530	28	0.7353
9	0.7680	19	0.7598	29	0.7639
10	0.7653	20	0.7503	30	0.7598

Como se puede observar en la ejecución #12, el mejor modelo de RNC obtenido por el algoritmo genético jerárquico logró una precisión de 0.7762 utilizando 90 épocas. Se obtuvo una precisión media de 0.7596. Con la información de la Tabla 6.6 también se obtuvo una desviación estándar de 0.009948132. En la Figura 6.5 se pueden observar las capas e hiperparámetros del modelo de RNC.

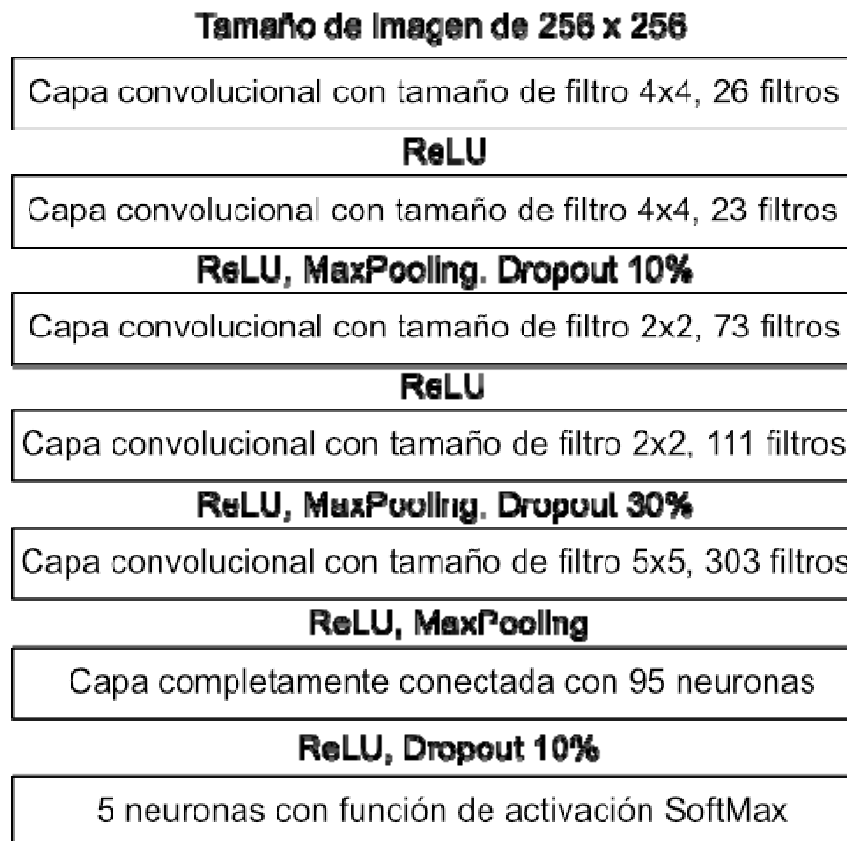


Figura 6.5. Modelo de RNC que obtuvo la mejor precisión para el caso de estudio multiclase.

6.3 Comparación de resultados

Antes de presentar las conclusiones, las mejores precisiones para el caso de estudio binario se pueden observar en la Tabla 6.7. En la Tabla 6.8 se pueden observar las precisiones para el caso de estudio multiclase. Se presentan las precisiones en el trabajo relacionado, la experimentación de métodos de preprocesamiento y la experimentación de algoritmos genéticos jerárquicos.

Tabla 6.7. Comparación de resultados para el caso de estudio binario.

Trabajo	Mejor precisión obtenida	Precisión media	Desviación estándar
Experimento de preprocesamiento binario #2	0.9523	0.9401	0.008245
Experimentación binaria de AGJ	0.9781	0.9650	0.007665
Modificación de InceptionV3 [50]	0.9557	—	—

Tabla 6.8. Comparación de resultados para el caso de estudio multiclase.

Trabajo	Mejor precisión obtenida	Precisión media	Desviación estándar
Experimento de preprocesamiento multiclase #2	0.7585	0.7401	0.010519
Experimentación multiclase de AGJ	0.7762	0.7596	0.009948
Modificación de InceptionV3 [50]	0.7667	—	—
VGG 2016 entrenado previamente [51]	0.7700	—	—

De acuerdo con las Tablas 6.7 y 6.8, no existe una forma posible de hacer una prueba estadística porque los otros autores no proporcionaron la precisión media ni la desviación estándar. Por lo tanto, las pruebas de hipótesis a continuación son solo entre la experimentación de los métodos de preprocesamiento y la experimentación del AGJ.

6.3.1 Prueba de hipótesis para el caso de estudio binario.

De acuerdo con los resultados de la Tabla 6.7, la prueba de hipótesis estará entre el trabajo “Experimento de preprocesamiento binario #2” y “Experimentación binaria de AGJ”. La experimentación binaria de AGJ obtuvo una precisión media más alta, por lo que nuestra afirmación es que la experimentación binaria de AGJ ofrece una precisión media mayor que la ofrecida por el experimento de preprocesamiento binario #2.

Para rechazar la hipótesis nula, el valor crítico debe ser inferior a -1.96 porque el valor de Alfa es 0.05. La puntuación de la prueba estadística es -1.7691. -1.7691 es mayor a -1.96, por lo que se falla al rechazar la hipótesis nula. Eso significa que no hay suficiente evidencia para respaldar la afirmación. La prueba de hipótesis se puede observar en la Fig. 6.6.

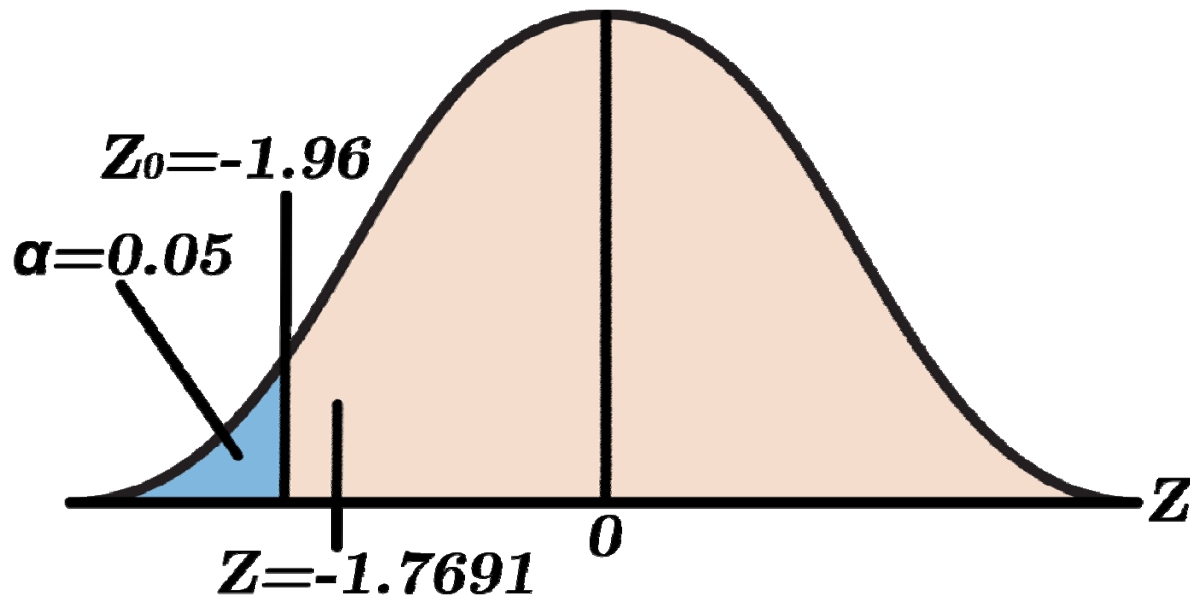


Figura 6.6. Prueba de hipótesis entre los dos experimentos del caso binario.

6.3.2 Prueba de hipótesis para el caso de estudio multiclase

De acuerdo con los resultados de la Tabla 6.8, la prueba de hipótesis será entre el trabajo “Experimento de preprocesamiento multiclase #2” y “Experimentación multiclase de AGJ”. La experimentación multiclase de AGJ obtuvo una precisión media más alta, por lo que nuestra afirmación es que la experimentación multiclase de AGJ ofrece una precisión media mayor que la ofrecida por el experimento de preprocesamiento multiclase #2.

Para rechazar la hipótesis nula, el valor crítico debe ser inferior a -1.96 porque el valor de Alfa es 0,05. La puntuación de la prueba estadística es -7.3771. -7.3771 es inferior a -1.96, por lo que se rechaza la hipótesis nula. Eso significa que hay suficiente evidencia para respaldar la afirmación. La prueba de hipótesis se puede observar en la Fig. 6.7.

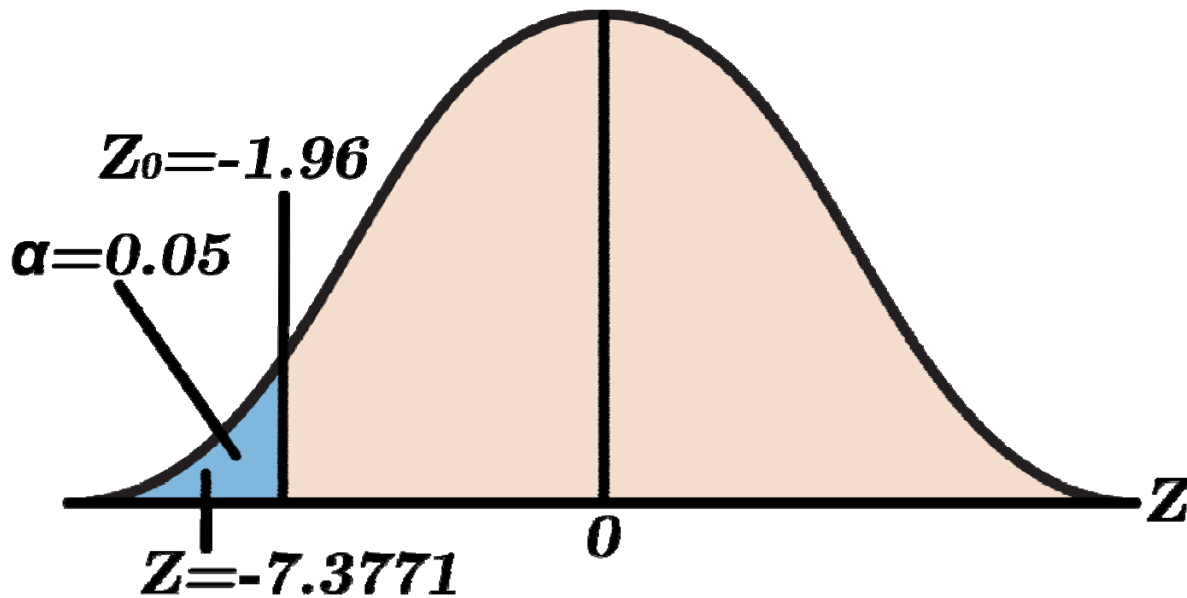


Figura 6.7. Prueba de hipótesis entre los dos experimentos del caso multiclase.

6.4 Interfaz gráfica de la aplicación Web

Tras finalizar todos los experimentos relacionados a este presente trabajo de Tesis, se empezó a desarrollar una aplicación Web con la intención de que pueda ser utilizado por estudiantes, investigadores y técnicos expertos del tema para el uso de la detección y clasificación de retinopatía diabética.

Se tomó la decisión de hacer una aplicación Web por las facilidades de acceso que ofrecía a diferencias de otras soluciones, como las aplicaciones móviles que requieren más trabajo debido a los diferentes sistemas operativos que se utilizan [59].

La aplicación Web consiste en una sola página que permite subir una imagen de retina. En esta se muestra la información principal del nombre de la página e imágenes de los logos de la institución y de CONACYT. Se encuentra un texto de bienvenida explicando la función de la página.

Las opciones para hacer funcionar la página se encuentran debajo del último texto. Se conforman por un cuadro gris que sirve para subir la imagen de la retina. As la derecha, dos *RadioButton* que sirven para cargar los modelos de RNC ya entrenados. Siendo los mejores modelos obtenidos en la experimentación del AGJ los que se utilizan para esta página Web. Hasta la derecha se encuentra el botón

para iniciar la clasificación de la retina. Como se pudo observar a lo largo de este trabajo de Tesis, la opción binaria permite detectar la enfermedad de RD, y el multiclase la clasificación de la gravedad de la enfermedad. La interfaz completa al momento de iniciar la aplicación se puede observar en la Figura 6.8.



Figura 6.8. Interfaz inicial de la aplicación Web.

Al momento de iniciar la página, no hay ninguna imagen de retina cargada, sino una imagen con un signo de interrogación que representa que no se ha cargado ninguna imagen. Para cargar una imagen, solo se da *click* en el cuadro mencionado anteriormente, y dependiendo desde dónde se esté ejecutando la aplicación, saldrá una ventana parecida a la que se puede observar en la Figura 6.9. Solo se puede cargar una imagen a la vez.

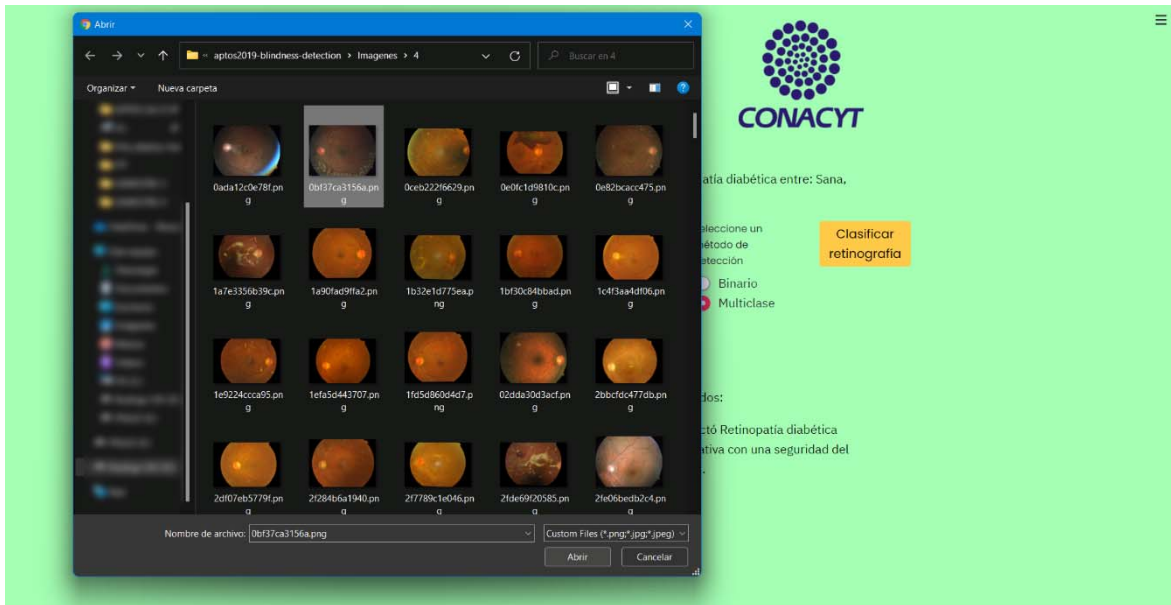


Figura 6.9. Seleccionando una imagen en Windows 11.

Dicha imagen se mostrará en la aplicación y solo resta ejecutar el botón de clasificación. Al iniciar la aplicación, el modelo binario estará cargado, pero esto se puede cambiar antes de iniciar la clasificación. En la Figura 6.10 se puede observar la clasificación binaria, mientras que en la Figura 6.11 una clasificación multiclase.

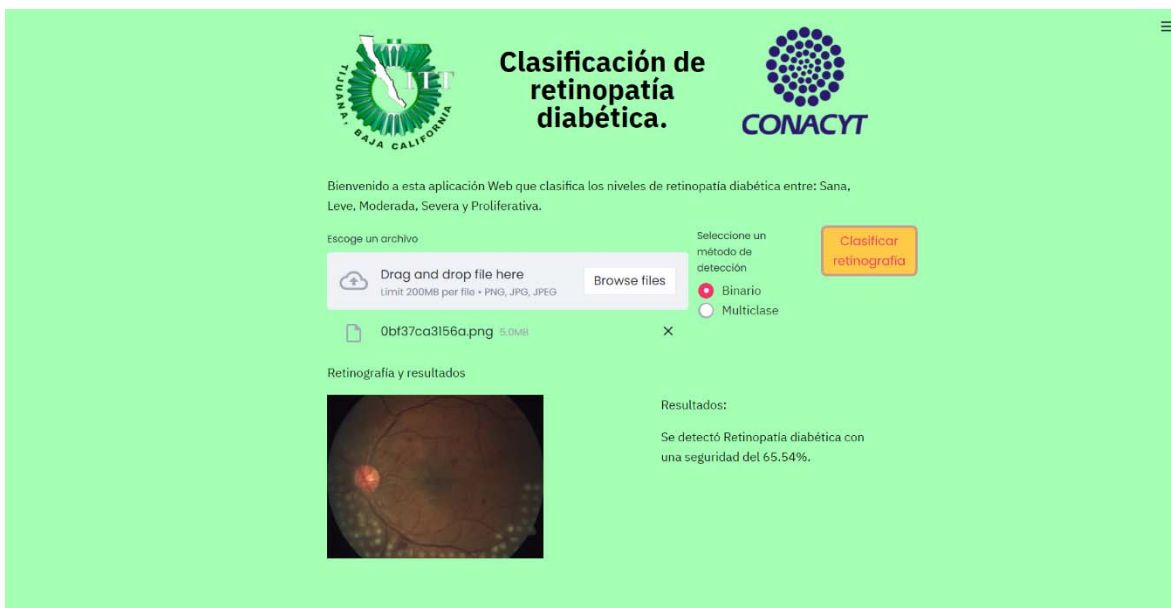


Figura 6.10. Realizando una clasificación binaria.

TIJUANA BAJA CALIFORNIA

Clasificación de retinopatía diabética.

CONACYT

Bienvenido a esta aplicación Web que clasifica los niveles de retinopatía diabética entre: Sana, Leve, Moderada, Severa y Proliferativa.

Escoge un archivo

Drag and drop file here
Limit 200MB per file • PNG, JPG, JPEG

Browse files

0bf37ca3156a.png 5.0MB

Selecciona un método de detección

Binario

Multiclase

Clasificar retinografía

Retinografía y resultados



Resultados:

Se detectó Retinopatía diabética Proliferativa con una seguridad del 40.44%.

Figura 6.11. Realizando una clasificación multiclase.

7 Conclusiones

En este trabajo de tesis se presentó el uso de 4 preprocesamientos diferentes aplicados a imágenes de retina. Además, se diseñaron dos modelos diferentes de RNC y se hicieron entrenamientos con cada uno con los 4 preprocesamientos. Para analizar diferentes situaciones, se prepararon dos casos de estudio para comparar sus resultados y ver qué modelo de preprocesamiento y RNC era más eficiente para la detección y clasificación de RD. En el primer caso de estudio se obtuvieron resultados similares para los dos mejores preprocesamientos. Pero en el segundo caso de estudio, el mejor preprocesamiento fue el cuadrado usando una RNC con profundidad 3. Esto no significa que solo se deban usar modelos RNC con esta profundidad, pero no es necesario usar muchas capas de convolución ya que estas imágenes pueden proporcionar peores resultados.

Después de la experimentación de los preprocesamientos, se diseñó un AGJ para la creación de modelos RNC con enfoque en la detección de RD. Antes de eso, se utilizó el preprocesamiento 2 para ayudar en la extracción de las funciones de RD. En el experimento se obtuvieron dos modelos de RNC en base a los casos de estudio anteriores con buenos resultados para la detección de RD y mejores que los obtenidos en la experimentación previa. Trabajos recientes prefirieron el uso de modelos de RNC existentes, pero gracias a los resultados obtenidos en este trabajo de tesis, creemos que el uso de algoritmos para la optimización podría crear mejores modelos de RNC.

Como trabajo futuro, el número de métodos de preprocesamiento podría aumentar o cambiar. Por ejemplo, el quinto preprocesamiento podría ser una combinación de los métodos de preprocesamiento #2 y #4. El método de preprocesamiento #2 se utilizó a lo largo de toda la experimentación AGJ, y una mejora de ese método podría aumentar aún más la precisión obtenida.

Además, el AGJ podría mejorarse aumentando el número de hiperparámetros. Hay muchos hiperparámetros para usar, como el tamaño del kernel para la capa Max-Pooling o una función de activación distinta después de la

capa convolucional. Los AGJs tienen sus límites, por lo que cambiar el algoritmo del optimizador podría mejorar la arquitectura de la RNC por completo.

Finalmente, APTOS 2019 es una buena base de datos para la detección de DR, pero no es la más grande. Por lo tanto, el uso de una base de datos más grande podría ayudar a la detección y clasificación de esta enfermedad. Con los resultados obtenidos se comprobó que el uso de modelos RNC es un método robusto de aprendizaje profundo que puede detectar enfermedades complicadas que requieren de un técnico experto.

A lo largo de este presente trabajo de Tesis se realizaron experimentos tanto para métodos de preprocesamiento de imágenes como de optimización de redes neuronales convolucionales. Trabajos realizados por otros autores inspiraron al desarrollo de los métodos propuestos, pero, con los resultados obtenidos después de la experimentación, se mejoraron los resultados de otros autores y se propusieron nuevas soluciones usando tecnologías que aún no se habían utilizado para la creación de un modelo de RNC con la intención de detectar y clasificar la RD. Los métodos de preprocesamiento utilizados en este trabajo pueden servir de inspiración a otros autores sin que requieran hacer los experimentos por su cuenta. El algoritmo genético jerárquico descrito en este trabajo señala los mejores hiperparámetros a optimizar para aumentar la precisión de los modelos de RNC al terminar su entrenamiento. Todo el trabajo realizado fue con el objetivo de obtener un modelo de RNC enfocado a detectar y clasificar el padecimiento de RD, objetivo el cual fue cumplido.

Referencias

- [1] J. Gu, H. Lu, C. Chen, Z. Gu, M. Hu, L. Liu, J. Yu, G. Wei y J. Huo, «Diabetes mellitus as a risk factor for chemotherapy-induced peripheral neuropathy: A meta-analysis,» *Supportive Care in Cancer*, vol. 29, pp. 7461-7469, 2021.
- [2] S. S. Rahim, V. Palade, J. Shuttleworth y C. Jayne, «Automatic screening and classification of diabetic retinopathy and maculopathy using fuzzy image processing,» *Brain Informatics*, vol. 3, pp. 249-267, 2016.
- [3] F. Jeyafzam, B. Vaziri, M. Y. Suraki, A. A. Hosseinabadi y A. Slowik, «Improvement of grey wolf optimizer with adaptive middle filter to adjust support vector machine parameters to predict diabetes complications,» *Neural Computing and Applications*, vol. 33, pp. 15205-15228, 2021.
- [4] R. Sarki, K. Ahmed, H. Wang, Y. Zhang, J. Ma y K. Wang, «Image preprocessing in classification and identification of diabetic eye diseases,» *Data Science and Engineering*, vol. 6, pp. 455-471, 2021.
- [5] P. Nage y S. Shitole, «A survey on automatic diabetic retinopathy screening,» *SN Computer Science*, vol. 2, p. 439, 2021.
- [6] P. S. Mangipudi, H. M. Pandey y A. Choudhary, «Improved optic disc and cup segmentation in glaucomatic images using Deep Learning Architecture,» *Multimedia Tools and Applications*, vol. 80, pp. 30143-30163, 2021.
- [7] Y. Sun, «The neural network of one-dimensional convolution-an example of the diagnosis of diabetic retinopathy,» *IEEE Access*, vol. 7, pp. 69657-69666, 2019.
- [8] J. Rodriguez-Saldana, *The Diabetes Textbook. Clinical Principles, Patient Management and Public Health Issues*, Cham: Springer Nature Switzerland AG, 2019.
- [9] E. Bonora y R. A. DeFronzo, *Diabetes. Epidemiology, Genetics, Pathogenesis, Diagnosis, Prevention, and Treatment*, Cham: Springer International Publishing AG, 2018.
- [10] I. Conget, «Diagnóstico, clasificación y patogenia de la diabetes mellitus,» *Revista Española de Cardiología*, vol. 55, nº 5, pp. 528-535, 2002.
- [11] C. J. Stocker, *Type 2 Diabetes. Methods and Protocols.*, Hatfield: Humana Press, 2020.

- [12] B. Kumar Tripathi y A. Kumar Srivastava, «Diabetes mellitus: Complications and therapeutics,» *Med Sci Monit*, vol. 12, nº 4, pp. 130-147, 2006.
- [13] L. Poretsky, *Principles of Diabetes Mellitus*, Cham: Springer International Publishing AG, 2004.
- [14] R. Walker, *The Diabetes Handbook*, London: Dorling Kindersley Limited, 2020.
- [15] A. E. Pinilla-Roa y M. d. P. Barrera-Perdomo, «Prevención en diabetes mellitus y riesgo cardiovascular: enfoque médico y nutricional,» *Revista de la Facultad de Medicina*, vol. 66, nº 3, pp. 459-468, 2018.
- [16] N. Cheung, P. Mitchell y T. Y. Wong, «Diabetic retinopathy,» *Seminar*, vol. 376, nº 9735, pp. 124-136, 2010.
- [17] C. Pineda Sarabia, X. Zarco Vite y M. Ruiz Morales, «Retinopatía diabética, una complicación descuidada,» *Revistas UNAM*, vol. 25, nº 2, pp. 83-85, 2018.
- [18] T. Shamah-Levy, C. Ruiz-Matus, J. Rivera-Dommarco, P. Kuri-Morales, L. Cuevas-Nasu, M. Jiménez-Corona y M. Romero-Martínez, *Encuesta Nacional de Salud y Nutrición de Medio Camino 2016*, Cuernavaca, México: Instituto Nacional de Salud Pública, 2017.
- [19] A. Das, S. Stroud, A. Mehta y S. Rangasamy, «New treatments for diabetic retinopathy,» *Diabetes, Obesity & Metabolism*, vol. 17, nº 3, pp. 219-230, 2015.
- [20] B. Rodríguez Rodríguez, «Prevención de ceguera por retinopatía diabética: ¿dónde estamos?,» *Revista Cubana de Oftalmología*, vol. 28, nº 1, pp. 119-128, 2015.
- [21] S. Majumder y N. Kehtarnavaz, «Multitasking deep learning model for detection of five stages of diabetic retinopathy,» *IEEE Access*, vol. 9, pp. 123220-123230, 2021.
- [22] «Comparison of Image Pre-processing for Classifying Diabetic Retinopathy Using Convolutional Neural Networks,» *Hybrid Intelligent Systems*, vol. 420, pp. 194-204, 2022.
- [23] S. Haykin, *Neural Networks. A Comprehensive Foundation*, Hamilton: Pearson Education, 1999.
- [24] D. A. Gomez-Cravioto, R. E. Diaz-Ramos, F. J. Cantu-Ortiz y H. G. Ceballos, «Data Analysis and forecasting of the COVID-19 spread: A comparison of recurrent neural networks and time series models,» *Cognitive Computation*,

2021.

- [25] F. Azam, *Biologically Inspired Modular Neural Networks*, Blacksburg: Virginia Polytechnic Institute and State University, 2000.
- [26] C. C. Aggarwal, *Neural Networks and Deep Learning. A Textbook.*, Cham: Springer International Publishing AG, 2018.
- [27] M. Flasiński, *Introduction to Artificial Intelligence*, Cham: Springer International Publishing AG, 2016.
- [28] S. Haykin, *Neural Networks and Learning Machines*, Hamilton: Pearson Education, 2009.
- [29] G. Yang, C. Wang, J. Yang, Y. Chen, L. Tang, P. Shao, J.-L. Dillenseger, H. Shu y L. Luo, «Weakly-supervised convolutional neural networks of renal tumor segmentation in abdominal CTA images,» *BMC Medical Imaging*, vol. 20, p. 37, 2020.
- [30] J. Rosenau, «Evolution and biogeography: Leading students in Darwin's and Wallace's footsteps,» *Evolution: Education and Outreach*, vol. 5, pp. 582-584, 2012.
- [31] J. I. Serrano y M. D. del Castillo, «On the origin of the evolutionary computation species influences of Darwin's theories on Computer Science,» *Artificial Intelligence Review*, vol. 38, pp. 41-54, 2012.
- [32] K. F. Man, K. S. Tang y S. Kwong, *Genetic Algorithms. Concepts and Designs*, Kowloon: Springer-Verlag London Berlin Heidelberg, 1999.
- [33] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Michigan: Addison-Wesley Publishing Company, Inc., 1989.
- [34] E. Rosado-Tamariz, M. A. Zuniga-Garcia y R. Batres, «Optimization of a drum boiler startup using dynamic simulation and a micro-genetic algorithm,» *Energy Reports*, vol. 6, n° 2, pp. 410-416, 2020.
- [35] K. Krishnakumar, «Micro-Genetic Algorithms For Stationary And Non-Stationary Function Optimization,» de *1989 Symposium on Visual Communications, Image Processing, and Intelligent Robotics Systems*, Philadelphia, PA, United States, 1989.
- [36] A. Santiago, B. Dorronsoro, H. J. Fraire y P. Ruiz, «Micro-Genetic algorithm with fuzzy selection of operators for multi-Objective optimization: FAME,» *Swarm and Evolutionary Computation*, vol. 61, p. 100818, 2021.

- [37] Z.-y. Xing, X.-m. Pang, H.-y. Ji, Y. Qin y L.-m. Jia, «Hierarchical genetic algorithm based RBF Neural Networks and application for modelling of the automatic depth control electrohydraulic system,» *International Journal of Control, Automation and Systems*, vol. 9, pp. 759-767, 2011.
- [38] C. H. Garcia-Capulin, F. J. Cuevas, G. Trejo-Caballero y H. Rostro-Gonzalez, «A hierarchical genetic algorithm approach for curve fitting with B-splines,» *Genetic Programming and Evolvable Machines*, vol. 16, pp. 151-166, 2015.
- [39] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard y L. D. Jackel, «Backpropagation Applied to Handwritten Zip Code Recognition,» *Neural Computation*, vol. 1, n° 4, pp. 541-551, 1989.
- [40] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems*, Sebastopol: O'Reilly, 2019.
- [41] Y. Xu, Y. Chi y Y. Tian, «Deep convolutional neural networks for feature extraction of images generated from complex networks topologies,» *Wireless Personal Communications*, vol. 103, pp. 327-338, 2018.
- [42] A. Bhandari, J. Koppen y M. Agzarian, «Convolutional neural networks for Brain Tumour segmentation,» *Insights into Imaging*, vol. 11, p. 77, 2020.
- [43] R. Xin, J. Zhang y Y. Shao, «Complex network classification with Convolutional Neural Network,» *Tsinghua Science and Technology*, vol. 25, n° 4, pp. 447-457, 2020.
- [44] L. Qiao, Y. Zhu y H. Zhou, «Diabetic retinopathy detection using prognosis of Microaneurysm and early diagnosis system for non-proliferative diabetic retinopathy based on Deep Learning Algorithms,» *IEEE Access*, vol. 104302, p. 104292, 2020.
- [45] R. Rodriguez, C. I. Gonzales, G. E. Martínez y P. Melin, «An improved convolutional neural network based on a parameter modification of the convolution layer,» *Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms: Theory and Applications*, vol. 940, pp. 125-147, 2021.
- [46] Y. Ying, J. Su, P. Shan, L. Miao, X. Wang y S. Peng, «Rectified exponential units for convolutional neural networks,» *IEEE Access*, vol. 7, pp. 101633-101640, 2019.
- [47] H. Pratt, F. Coenen, D. M. Broadbent, S. P. Harding y Y. Zheng, «Convolutional Neural Networks for diabetic retinopathy,» *Procedia Computer Science*, vol. 90, pp. 200-205, 2016.

- [48] E. Irmak, «Multi-classification of brain tumor MRI images using deep convolutional neural network with fully optimized framework,» *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 45, n° 3, pp. 1015-1036, 2021.
- [49] D. Danon, M. Arar, D. Cohen-Or y A. Shamir, «Image resizing by reconstruction from deep features,» *Computational Visual Media*, vol. 7, pp. 453-466, 2021.
- [50] V. Vives-Boix y D. Ruiz-Fernández, «Diabetic retinopathy detection through convolutional neural networks with synaptic metaplasticity,» *Computer Methods and Programs in Biomedicine*, vol. 206, pp. 106094-106094, 2021.
- [51] O. Dekhil, A. Naglah, M. Shaban, M. Ghazal, F. Taher y A. Elbaz, «Deep Learning based method for computer aided diagnosis of diabetic retinopathy,» *2019 IEEE International Conference on Imaging Systems and Techniques (IST)*, pp. 1-4, 2019.
- [52] A. Bhandare y D. Kaur, «Designing convolutional neural network architecture using genetic algorithms,» *International Journal of Advanced Network, Monitoring and Controls*, vol. 6, n° 3, pp. 26-35, 2021.
- [53] Kaggle, «Kaggle,» Agosto 2019. [En línea]. Available: <https://www.kaggle.com/c/aptos2019-blindness-detection>. [Último acceso: 28 Febrero 2021].
- [54] Kaggle, «Kaggle,» Julio 2015. [En línea]. Available: <https://www.kaggle.com/c/diabetic-retinopathy-detection/>. [Último acceso: 22 Febrero 2021].
- [55] A. K. Gangwar y V. Ravi, «Diabetic retinopathy detection using transfer learning and Deep Learning,» *Evolution in Computational Intelligence*, vol. 1176, pp. 679-689, 2020.
- [56] H. N. Pham, R. J. Tan, Y. T. Cai, S. Mustafa, N. C. Yeo, H. J. Lim, T. T. Do, B. P. Nguyen y M. C. Chua, «Automated grading in diabetic retinopathy using image processing and modified EfficientNet,» *Computational Collective Intelligence*, pp. 505-515, 2020.
- [57] D. Doshi, A. Shenoy, D. Sidhpura y P. Gharpure, «Diabetic retinopathy detection using deep convolutional neural networks,» *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, pp. 261-266, 2016.
- [58] G. Nguyen, S. Dlugolinsky, M. Bobák, V. Tran, Á. López García, I. Heredia, P. Malík y L. Hluchý, «Machine Learning and Deep Learning frameworks and

libraries for large-scale data mining: a survey,» *Artificial Intelligence Review*, vol. 52, pp. 77-124, 2019.

- [59] P. Melin y D. Sánchez, «Multi-objective optimization for modular granular neural networks applied to pattern recognition,» *Information Sciences*, Vols. %1 de %2460-461, pp. 594-610, 2017.
- [60] R. Harrison, D. Flood y D. Duce, «Usability of mobile applications: literature review and rationale for a new usability model,» *Journal of Interaction Science*, vol. 1, nº 1, 2013.

Anexos

Anexo 1. Resultados experimento del caso binario usando una RNC de profundidad 3 con el método de preprocesamiento 1

Número de experimento	Precisión obtenida
1	0.9304
2	0.9441
3	0.9154
4	0.9243
5	0.9354
6	0.9304
7	0.9461
8	0.9195
9	0.9400
10	0.9372
11	0.9345
12	0.9338
13	0.9359
14	0.9393
15	0.9441
16	0.9366
17	0.9386
18	0.9325
19	0.9297
20	0.9454
21	0.9400
22	0.9393
23	0.9400
24	0.9427
25	0.9420
26	0.9400
27	0.9318
28	0.9345
29	0.9454
30	0.9413
Media	0.93634
Desviación Estándar	0.00739364
Máximo	0.9461
Mínimo	0.9154

Anexo 2. Resultados experimento del caso binario usando una RNC de profundidad 3 con el método de preprocesamiento 2

Número de experimento	Precisión obtenida
1	0.9386
2	0.9359
3	0.9468
4	0.9454
5	0.9332
6	0.9482
7	0.9345
8	0.9393
9	0.9523
10	0.9509
11	0.9250
12	0.9400
13	0.9338
14	0.9386
15	0.9495
16	0.9461
17	0.9413
18	0.9393
19	0.9338
20	0.9427
21	0.9325
22	0.9400
23	0.9447
24	0.9400
25	0.9154
26	0.9393
27	0.9297
28	0.9482
29	0.9468
30	0.9509
Media	0.94009
Desviación Estándar	0.00824493
Máximo	0.9523
Mínimo	0.9154

Anexo 3. Resultados experimento del caso binario usando una RNC de profundidad 3 con el método de preprocesamiento 3

Número de experimento	Precisión obtenida
1	0.9202
2	0.9161
3	0.9216
4	0.9222
5	0.9072
6	0.9045
7	0.9106
8	0.9216
9	0.9195
10	0.9229
11	0.9181
12	0.9188
13	0.8881
14	0.9202
15	0.9065
16	0.9141
17	0.9113
18	0.9045
19	0.9079
20	0.9154
21	0.9065
22	0.9168
23	0.9072
24	0.9045
25	0.9093
26	0.9188
27	0.9270
28	0.9127
29	0.9079
30	0.8977
Media	0.91265667
Desviación Estándar	0.00849006
Máximo	0.927
Mínimo	0.8881

Anexo 4. Resultados experimento del caso binario usando una RNC de profundidad 3 con el método de preprocesamiento 4

Número de experimento	Precisión obtenida
1	0.9236
2	0.9168
3	0.9277
4	0.9263
5	0.9311
6	0.9236
7	0.9168
8	0.9277
9	0.9243
10	0.9263
11	0.9161
12	0.9284
13	0.9134
14	0.9216
15	0.9284
16	0.9229
17	0.9284
18	0.8984
19	0.9304
20	0.9181
21	0.9181
22	0.9250
23	0.9127
24	0.9188
25	0.9263
26	0.9188
27	0.9229
28	0.8984
29	0.9127
30	0.9100
Media	0.92046667
Desviación Estándar	0.00828794
Máximo	0.9311
Mínimo	0.8984

Anexo 5. Resultados experimento del caso binario usando una RNC de profundidad 10 con el método de preprocesamiento 1

Número de experimento	Precisión obtenida
1	0.9154
2	0.8950
3	0.8854
4	0.9291
5	0.9332
6	0.9250
7	0.9263
8	0.9195
9	0.9304
10	0.9004
11	0.9263
12	0.9018
13	0.8731
14	0.9359
15	0.9291
16	0.8881
17	0.9386
18	0.9427
19	0.9154
20	0.9181
21	0.5375
22	0.9100
23	0.9536
24	0.9482
25	0.8895
26	0.8226
27	0.9004
28	0.8786
29	0.9222
30	0.9441
Media	0.90118333
Desviación Estándar	0.07387703
Máximo	0.9536
Mínimo	0.5375

Anexo 6. Resultados experimento del caso binario usando una RNC de profundidad 10 con el método de preprocesamiento 2

Número de experimento	Precisión obtenida
1	0.8977
2	0.8950
3	0.8936
4	0.9591
5	0.9236
6	0.9031
7	0.9536
8	0.8936
9	0.8977
10	0.8977
11	0.9400
12	0.5293
13	0.9113
14	0.9318
15	0.9236
16	0.9236
17	0.9536
18	0.9245
19	0.9127
20	0.9127
21	0.8895
22	0.7353
23	0.8786
24	0.8240
25	0.9441
26	0.8827
27	0.9482
28	0.8690
29	0.8349
30	0.8936
Media	0.88925667
Desviación Estándar	0.08135433
Máximo	0.9591
Mínimo	0.5293

Anexo 7. Resultados experimento del caso binario usando una RNC de profundidad 10 con el método de preprocesamiento 3

Número de experimento	Precisión obtenida
1	0.7613
2	0.8718
3	0.9536
4	0.9100
5	0.5539
6	0.9563
7	0.9263
8	0.9400
9	0.9263
10	0.9413
11	0.8267
12	0.9345
13	0.9591
14	0.9495
15	0.8963
16	0.9495
17	0.9372
18	0.9659
19	0.9563
20	0.8281
21	0.9550
22	0.9454
23	0.9209
24	0.9441
25	0.9413
26	0.8240
27	0.9400
28	0.7763
29	0.9004
30	0.9413
Media	0.90108667
Desviación Estándar	0.08574371
Máximo	0.9659
Mínimo	0.5539

Anexo 8. Resultados experimento del caso binario usando una RNC de profundidad 10 con el método de preprocesamiento 4

Número de experimento	Precisión obtenida
1	0.8199
2	0.9277
3	0.9386
4	0.9291
5	0.7954
6	0.6508
7	0.9072
8	0.7967
9	0.9345
10	0.9495
11	0.8649
12	0.9181
13	0.9386
14	0.8936
15	0.9454
16	0.9250
17	0.8295
18	0.9454
19	0.7121
20	0.8922
21	0.9277
22	0.9359
23	0.9427
24	0.9468
25	0.9345
26	0.9523
27	0.9195
28	0.9222
29	0.9141
30	0.9509
Media	0.89536
Desviación Estándar	0.0738387
Máximo	0.9523
Mínimo	0.6508

Anexo 9. Resultados experimento del caso multiclase usando una RNC de profundidad 3 con el método de preprocesamiento 1

Número de experimento	Precisión obtenida
1	0.7258
2	0.7231
3	0.7271
4	0.7108
5	0.7231
6	0.7299
7	0.7244
8	0.7244
9	0.7340
10	0.7176
11	0.7381
12	0.7244
13	0.7258
14	0.7231
15	0.7176
16	0.7053
17	0.7312
18	0.7299
19	0.7149
20	0.7149
21	0.7203
22	0.7217
23	0.7190
24	0.7176
25	0.7271
26	0.7108
27	0.7162
28	0.7162
29	0.7162
30	0.7244
Media	0.72183
Desviación Estándar	0.00724198
Máximo	0.7381
Mínimo	0.7053

Anexo 10. Resultados experimento del caso multiclase usando una RNC de profundidad 3 con el método de preprocesamiento 2

Número de experimento	Precisión obtenida
1	0.7462
2	0.7381
3	0.7367
4	0.7435
5	0.7585
6	0.7476
7	0.7422
8	0.7340
9	0.7544
10	0.7503
11	0.7422
12	0.7190
13	0.7476
14	0.7381
15	0.7271
16	0.7422
17	0.7312
18	0.7422
19	0.7367
20	0.7422
21	0.7517
22	0.7503
23	0.7517
24	0.7476
25	0.7203
26	0.7299
27	0.7435
28	0.7271
29	0.7422
30	0.7176
Media	0.74006333
Desviación Estándar	0.010519
Máximo	0.7585
Mínimo	0.7176

Anexo 11. Resultados experimento del caso multiclase usando una RNC de profundidad 3 con el método de preprocesamiento 3

Número de experimento	Precisión obtenida
1	0.7312
2	0.7203
3	0.7353
4	0.7271
5	0.7490
6	0.7381
7	0.7080
8	0.6917
9	0.7285
10	0.7217
11	0.7135
12	0.7203
13	0.7340
14	0.7162
15	0.7326
16	0.7108
17	0.7026
18	0.7131
19	0.7108
20	0.7408
21	0.6930
22	0.7408
23	0.7231
24	0.7367
25	0.7244
26	0.7326
27	0.7149
28	0.6930
29	0.7162
30	0.7203
Media	0.72135333
Desviación Estándar	0.01474537
Máximo	0.749
Mínimo	0.6917

Anexo 12. Resultados experimento del caso multiclase usando una RNC de profundidad 3 con el método de preprocesamiento 4

Número de experimento	Precisión obtenida
1	0.7040
2	0.7285
3	0.6726
4	0.7258
5	0.7299
6	0.7176
7	0.7094
8	0.7162
9	0.7080
10	0.7312
11	0.7244
12	0.7217
13	0.7203
14	0.7408
15	0.7203
16	0.7162
17	0.7203
18	0.7285
19	0.7394
20	0.7176
21	0.7217
22	0.7149
23	0.7040
24	0.7080
25	0.7135
26	0.7271
27	0.7312
28	0.7340
29	0.7149
30	0.7162
Media	0.71927333
Desviación Estándar	0.01299899
Máximo	0.7408
Mínimo	0.6726

Anexo 13. Resultados experimento del caso multiclase usando una RNC de profundidad 10 con el método de preprocesamiento 1

Número de experimento	Precisión obtenida
1	0.7271
2	0.7381
3	0.5866
4	0.7176
5	0.7176
6	0.7271
7	0.7162
8	0.7217
9	0.6985
10	0.7285
11	0.6385
12	0.6767
13	0.7108
14	0.7312
15	0.7299
16	0.7094
17	0.7353
18	0.7299
19	0.7094
20	0.7312
21	0.7353
22	0.7353
23	0.7094
24	0.7394
25	0.7231
26	0.7012
27	0.6317
28	0.7271
29	0.7108
30	0.6794
Media	0.70913333
Desviación Estándar	0.03506277
Máximo	0.7271
Mínimo	0.7381

Anexo 14. Resultados experimento del caso multiclase usando una RNC de profundidad 10 con el método de preprocesamiento 2

Número de experimento	Precisión obtenida
1	0.6971
2	0.6930
3	0.7067
4	0.6971
5	0.7190
6	0.6944
7	0.6821
8	0.7258
9	0.6917
10	0.6930
11	0.6985
12	0.7285
13	0.7476
14	0.7040
15	0.7203
16	0.7585
17	0.6712
18	0.7503
19	0.7230
20	0.7489
21	0.7503
22	0.7285
23	0.7407
24	0.7367
25	0.7162
26	0.7544
27	0.7244
28	0.7285
29	0.7367
30	0.7449
Media	0.7204
Desviación Estándar	0.02397598
Máximo	0.7585
Mínimo	0.6712

Anexo 15. Resultados experimento del caso multiclase usando una RNC de profundidad 10 con el método de preprocesamiento 3

Número de experimento	Precisión obtenida
1	0.7408
2	0.5484
3	0.7217
4	0.7381
5	0.7367
6	0.6562
7	0.7449
8	0.7285
9	0.7299
10	0.7367
11	0.6630
12	0.7135
13	0.7271
14	0.7135
15	0.7135
16	0.6930
17	0.6930
18	0.7299
19	0.3984
20	0.6753
21	0.7217
22	0.7353
23	0.7176
24	0.7408
25	0.3042
26	0.7135
27	0.6889
28	0.7449
29	0.7249
30	0.7435
Media	0.68791333
Desviación Estándar	0.10015835
Máximo	0.7449
Mínimo	0.3042

Anexo 16. Resultados experimento del caso multiclase usando una RNC de profundidad 10 con el método de preprocesamiento 4

Número de experimento	Precisión obtenida
1	0.7271
2	0.6944
3	0.7121
4	0.7449
5	0.7340
6	0.6016
7	0.7217
8	0.7067
9	0.6876
10	0.7326
11	0.7026
12	0.6849
13	0.6603
14	0.6276
15	0.7271
16	0.6057
17	0.7285
18	0.5512
19	0.7203
20	0.7162
21	0.7271
22	0.7012
23	0.7285
24	0.6930
25	0.6439
26	0.6303
27	0.6521
28	0.7135
29	0.7271
30	0.7176
Media	0.69071333
Desviación Estándar	0.04782971
Máximo	0.7449
Mínimo	0.5512

Todos los experimentos en este presente trabajo de Tesis fueron realizados con la misma computadora. A continuación, en el Anexo 17 se puede observar las características de dicha computadora, esto con el objetivo de ayudar a otros autores al momento de querer escoger las piezas adecuadas para algún experimento similar al presentado a lo largo de este documento.

Anexo 17. Características de la computadora utilizada para elaborar el presente trabajo de Tesis.

Nombre de la pieza o tecnología	Detalles
Modelo	Laptop Dell G3 15 (3590)
Procesador	Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz
Sistema operativo	Windows 11 Pro
Memoria RAM	HyperX Impact RAM 16x2 (32) GB 2666MHz
Disco interno	512 GB SSD
Disco externo	2 TB HDD
Gráfica dedicada	NVIDIA GeForce GTX 1660 ti Max-Q
Memoria de la gráfica	GDDR6 6 GB
Software utilizado para los métodos de preprocesamiento	MATLAB R2020b: Image Processing Toolbox
Software utilizado para el entrenamiento de redes neuronales	Lenguaje de programación Python con los Frameworks Tensorflow y Keras
Software utilizado para la creación de la página Web	Lenguaje de programación Python con el Framework Streamlit

Anexo 18. Tiempos de ejecución de los experimentos del AGJ para el caso de estudio binario.

Número de experimento	Tiempo transcurrido (en hh:mm:ss)
1	09:34:44
2	10:11:54
3	07:27:46
4	08:13:41
5	08:04:49
6	10:26:18
7	06:46:18
8	09:16:31
9	09:04:37
10	09:48:50
11	08:36:11
12	08:15:21
13	07:14:38
14	07:26:58
15	07:46:34
16	09:31:22
17	06:24:01
18	10:21:27
19	15:24:08
20	09:09:56
21	10:53:32
22	10:25:18
23	06:57:37
24	07:56:09
25	07:04:05
26	12:03:50
27	12:49:32
28	11:32:12
29	09:10:44
30	10:37:58

Anexo 19. Tiempos de ejecución de los experimentos del AGJ para el caso de estudio multiclase.

Número de experimento	Tiempo transcurrido (en hh:mm:ss)
1	26:19:25
2	11:13:56
3	14:35:35
4	14:50:54
5	32:29:06
6	28:54:25
7	20:52:54
8	10:26:50
9	10:01:41
10	10:12:37
11	10:22:50
12	09:29:11
13	11:51:56
14	11:51:31
15	11:49:58
16	12:10:14
17	12:07:27
18	14:17:06
19	14:06:42
20	14:11:29
21	14:10:11
22	14:20:27
23	13:29:27
24	16:17:47
25	14:47:27
26	15:01:48
27	14:12:53
28	18:55:29
29	19:06:07
30	21:59:27

Anexo 20. Código de Matlab sobre el método de preprocesamiento 1 (solo imágenes de retina sanas).

```
clear all

digitDatabasePath = 'D:\APTOS CNN Testing\aptos2019-
blindness-detection\Imagenes\0\';
digitImages =
imageDatastore(digitDatabasePath, 'IncludeSubfolders', true,
'LabelSource', 'foldernames');

croppedPath = 'D:\diabetic-retinopathy-
detection\Cropped\0\';

for i=1:1805
```

```

path = string(digitImages.Files(i));
data = imread(path);

imgOriginal = data;
dm = max(data,[],3);
img = image(data, 'AlphaData', double(dm<20));
newimg = img.AlphaData;
newimg = ~newimg;

stats = regionprops(newimg, 'BoundingBox');

biggerWidth = 0;
bestIndex = 0;
for k = 1 : length(stats)
    actualWidth = stats(k).BoundingBox(3);
    if(actualWidth > biggerWidth)
        biggerWidth = actualWidth;
        bestIndex = k;
    end
end
xMin = ceil(stats(bestIndex).BoundingBox(1));
xMax = xMin + stats(bestIndex).BoundingBox(3) - 1;
yMin = ceil(stats(bestIndex).BoundingBox(2));
yMax = yMin + stats(bestIndex).BoundingBox(4) - 1;
img2 = imgOriginal(yMin:yMax,xMin:xMax,:);
% imwrite(img2, strcat(string(i), '.png'));
newimg = bwareaopen(newimg,10);
imshow(newimg);
clear imgOriginal newimg stats img2
i
end

```

Anexo 21. Código de Matlab sobre el método de preprocesamiento 2 (solo imágenes de retina sanas).

```

clear all

digitDatabasePath = 'D:\APTOS CNN Testing\aptos2019-
blindness-detection\Cropped\0\';
digitImages =
imageDatastore(digitDatabasePath, 'IncludeSubfolders', true,
e, 'LabelSource', 'foldernames');

for i=1:1805

```

```

path = string(digitImages.Files(i));
img = imread(path);
[m, n] = size(rgb2gray(img));
if m==n
    iwant=img;
else
    if m>n
        iwant = uint8(zeros(m,m,3));
        p1 = (m-n)/2;
        iwant(:,p1:m-p1-1,:) = img;
    else
        iwant = uint8(zeros(n,n,3));
        p1 = (n-m)/2;
        iwant(p1:n-p1-1,,:,) = img;
    end
end

imwrite(iwant, strcat(string(i), '.png'));
clear img m n iwant p1
i
end

```

Anexo 22. Código de Matlab sobre el método de preprocesamiento 3 (solo imágenes de retina sanas).

```

clear all

digitDatabasePath = 'D:\APTOS CNN Testing\aptos2019-
blindness-detection\Cropped\0\';
digitImages =
imageDatastore(digitDatabasePath, 'IncludeSubfolders', true,
'LabelSource', 'foldernames');

for i=1:1805
    path = string(digitImages.Files(i));
    img = imread(path);

    r = histeq(img(:,:,1));
    g = histeq(img(:,:,2));
    b = histeq(img(:,:,3));

    newImg(:,:,1) = r;
    newImg(:,:,2) = g;
    newImg(:,:,3) = b;

    imwrite(newImg, strcat('0_', string(i), '.png'));

```

```

        clear img newImg
        i
end
Anexo 23. Código de Matlab sobre el método de preprocesamiento 4 (solo imágenes de retina sanas).

clear all

digitDatabasePath = 'D:\APTOS CNN Testing\aptos2019-
blindness-detection\Cropped\0\';
digitImages =
imageDatastore(digitDatabasePath, 'IncludeSubfolders', true,
'LabelSource', 'foldernames');

for i=1:1805
    path = string(digitImages.Files(i));
    img = imread(path);

    g = histeq(img(:,:,2));

    newImg(:,:,) = g;

    imwrite(uint8(newImg),strcat('0_',string(i),'.png'));
    clear img newImg g
    i
end

```