



Universidad Autónoma de Querétaro
Facultad de Ingeniería

Clasificación de retinopatía diabética por medio de redes
neuronales convolucionales.

Tesis

Que como parte de los requisitos para el obtener el Grado de
Maestro en Ciencias en Inteligencia Artificial

Presenta:

Rafael Ortiz Feregrino

Dirigido por:

Dr. Saúl Tovar Arriaga

Querétaro, Qro. a 24 de Noviembre de 2019.



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO

Facultad de Ingeniería

Maestría en Ciencias en Inteligencia Artificial

Clasificación de retinopatía diabética por medio de redes neuronales
convolucionales.

Tesis

Que como parte de los requisitos para obtener el Grado de
Maestro en Ciencias en Inteligencia Artificial

Presenta:

Rafael Ortiz Feregrino

Dirigido por:

Dr. Saúl Tovar Arriaga

Dr. Saúl Tovar Arriaga

Presidente

Dr. Jesús Carlos Pedraza

Secretario

Dr. Arturo González Gutiérrez

Vocal

Dr. Efrén Gorrostieta Hurtado

Suplente

Dr. Edgar Alejandro Rivas Araiza

Suplente

Centro Universitario, Querétaro, Qro.

Fecha de Aprobación

México

Índice general

Índice	I
Índice de Figuras	III
Índice de Tablas	V
Índice de Algoritmos	V
1. Agradecimiento	1
2. Abreviaturas y siglas	2
3. Resumen	4
4. Abstract	5
5. Introducción	6
5.1. Planteamiento del problema	9
5.2. Justificación	10
6. Antecedentes	11
6.1. Implementacion de aprendizaje automatizado	11
6.2. Clasificación de RD	14
6.3. Retinopatía diabética	19
6.4. Aprendizaje profundo	24
6.4.1. Arquitecturas de aprendizaje profundo.	26
6.4.2. Redes Neuronales Convolucionales	29

6.4.3. Parámetros de una CNN	35
6.4.4. Funciones de activación.	40
6.4.5. Entrenamiento de una CNN	42
6.5. Procesamiento de los datos	47
6.5.1. Aumento de brillo y contraste	48
6.5.2. Software especializado en aprendizaje profundo	50
7. Hipotesis	52
8. Objetivos	53
9. Metodología	54
9.0.1. Procesamiento de imágenes	57
9.0.2. Entrenamiento	59
10. Resultados	63
11. Conclusiones	67
11.1. Trabajo futuro	68
Referencias	69
Anexos	74

Índice de Figuras

6.1. La primera imagen de retina esta centrada en la mácula. La siguiente imagen esta centrada en el nervio óptico.	19
6.2. Ejemplos de imágenes de retina.	20
6.3. Representación del área de computación inteligente.	25
6.4. Ejemplo de la composición de una Red neuronal tradicional.	25
6.5. Diagrama básico de aprendizaje por refuerzo. (Kulkarni, 2012)	28
6.6. En la CNN de la figura una neurona de entrada tiene repercusión en tres neuronas de la siguiente capa consideradas como salidas en este caso. En cambio en una red profunda tradicional una neurona tiene influencia en todos los nodos de la siguiente capa, como se puede apreciar.	30
6.7. En esta figura se puede observar que las neuronas de entrada de la CNN tan solo a un nodo de salida. En cambio las neuronas de toda la capa de entrada influyen en una sola neurona de salida en la RN típica.	31
6.8. Estructura CNN. (Mathworks, 2017)	32
6.9. Bloque de entrenamiento residual. (Krizhevsky et al., 2017)	33
6.10. Gráfica de los resultados obtenidos utilizando redes planas y redes residuales respectivamente. (Krizhevsky et al., 2017)	34
6.11. Descripción de kernel	35
6.12. Descripción de la función Pooling	36
6.13. Respuestas gráficas de FA descritas en el capítulo.	41

6.14. Resultados de la clasificación de perros y gatos mediante GPU y CPU respectivamente.	46
9.1. La gráfica del lado izquierdo muestra el total de imágenes en sus cinco categorías, la gráfica de la derecha representa el número de imágenes en dos categorías tomando en cuenta el desbalance de los datos.	55
9.2. La primera imagen muestra la arquitectura de la CNN-1. Mientras que la imagen del lado derecho da lugar a la CNN-2.	57
9.3. Metodología de tratamiento de imágenes.	58
9.4. La imagen de la izquierda es el resultado final del procesamiento y se puede observar la diferencia con la original del lado derecho.	59
9.5. Proceso completo del trabajo realizado.	60
10.1. La gráfica muestra la exactitud durante el entrenamiento de ambas redes, este valor se obtuvo con los datos de validación.	65
10.2. Matrices de confusión para la CNN-1 y la CNN-2 respectivamente.	66

Índice de Tablas

5.1. Clasificación por tipo de diabetes mellitus	7
6.1. Estado del arte. Artículos mas relevantes	16
6.2. Diferencias significativas entre aprendizaje automático y profundo	17
6.3. Grados de severidad según la clasificación Escocesa.	22
6.4. Resultados del entrenamiento con GPU y CPU	45
6.5. Software especializado en AP	51
10.1.Descripción de matriz de confusión.	65
10.2.Resultados obtenidos con los datos de prueba.	66

Índice de Algoritmos

1. Pseudocódigo del entrenamiento	61
-----------------------------------	----

Agradecimiento

Deseo expresar mi gratitud a todos los involucrados en este trabajo, que directa o indirectamente estuvieron apoyando de cualquier forma posible.

Quiero agradecer a mi madre Nely Feregrino Vega y mi padre Rafael Ortiz Vega por todo el amor, paciencia; confianza incondicional que siempre me brindaron y por ser mi mayor fuente de inspiración. A mis hermanos Eduardo y Bryan que me ayudan a ser la mejor versión de mí.

A mi asesor Saul Tovar Arriaga por todo el tiempo y trabajo dedicado a este proyecto, así como toda la experiencia que me brindo.

Se agradece al Consejo Nacional de Ciencia y Tecnología (CONACYT) por proveer los fondos realizados para esta investigación.

Abreviaturas y siglas

RD - Retinopatía diabética.

CNN - Redes neuronal convolucionales

IA - Inteligencia artificial

AP - Aprendizaje profundo

RP – Redes profundas

AA – Aprendizaje automático.

RGB – Rojo verde azul.

RNP – Redes neuronales profundas

RN – Redes neuronales.

AR – Aprendizaje por refuerzo.

RNA - Redes de base radial.

RNN – Redes neuronal recurrentes.

RR – Redes residuales.

FA – Funciones de activación.

ICS – Internal covariate shift.

CNN-1 – Red neuronal convolucional lineal.

CNN-2 – Red neuronal convolucional residual.

IMO – Instituto mexicano de oftalmología.

GPU – Unidad de procesamiento gráfico.

CPU – Unidad de procesamiento central.

Resumen

La RD es una enfermedad con diferentes grados de severidad. Los patrones que denotan la severidad de dicha enfermedad son variados; cuando la RD presenta una severidad leve o moderada es difícil identificar los patrones o anomalías. En este trabajo se utilizaron dos arquitecturas diferentes de CNN para la clasificación de RD dependiendo el grado de severidad. Se utilizó una base de datos pública descargada de la página Kaggle. Las arquitecturas presentaron una exactitud del 96.37 % y 97.38 % en la clasificación de RD proliferativa.

Palabras clave-, diagnostico automatizado, aprendizaje profundo, retinopatía diabética, clasificación de imágenes .

Abstract

The disease diabetic retinopathy has different grades of severity. The sequence that this different grade has is really complicate to classification with an inexpert eye. When de RD is low or moderate is so difficult the identification the patterns. This research present two different CNN in order to classification the RD depending the grade of severity. The data base used is public and can be download in the kaggle web page. With the two CNNs achieved an accuracy of 96.37 % and 97.38 % in the classification of proliferative RD.

Key words-, **Automatize detection**, **Deep learning**, **Diabetic retinopathy**, **Image classification**. .

Introducción

A lo largo de la evolución de la IA, se han tratado de resolver problemas más complejos y con mayor impacto social. Uno de los retos más grandes a los cuales nos enfrentamos como sociedad moderna es la prevención de enfermedades las cuales se pueden evitar pero que por falta de recursos siguen teniendo una prevalencia alta en la vida moderna. (Prado-Serrano et al., 2019)

Una de las enfermedades con mayor índice de incidencia es la diabetes, según la organización mundial de la salud. (García Bello et al., 2016) La cual en sus inicios solo se presentaba en adultos mayores y personas con malos hábitos alimenticios, pero que ahora está presente en niños, jóvenes y adultos sin discriminación alguna.

Bueno, pero a todo esto ¿Que es la diabetes? es un conjunto de trastornos metabólicos, como puede serlo también la hipertensión arterial, que durante su evolución y en función del grado de control, la edad y la persona puede asociarse o no al desarrollo de complicaciones a lo largo de la vida. (Hsu et al., 2015)

La diabetes mellitus comúnmente denominada solo como diabetes, Se origina como consecuencia de un déficit de la hormona insulina. La diabetes se puede dividir por tipos o grados. Azeze2018,Prado-Serrano2009 En la tabla 5.1 se puede ver los tipos de diabetes y algunas características a resaltar de cada categoría.

Tabla 5.1: Clasificación por tipo de diabetes mellitus

Tipos de diabetes mellitus	Consideraciones
Diabetes tipo 1	No se puede prevenir con los conocimientos actuales-
Diabetes tipo 2	Este tipo de diabetes se puede prevenir, para así evitar complicaciones o la muerte prematura.
Diabetes gestacional	Se caracteriza por el aumento del azúcar en la sangre que aparece durante el embarazo, pero no son tan altos como los establecidos para diagnosticar diabetes.
Otros tipos de diabetes	

La RD es una enfermedad derivada de la diabetes, la cual afecta al ojo, para ser más precisos, ataca la retina, la cual es la parte posterior del ojo y es donde se localizan los vasos sanguíneos y micro venas que alimentan de sangre el órgano ocular.

En la actualidad las técnicas de IA se han vuelto muy populares y con justa razón, ya que han demostrado tener un buen desempeño en la resolución de problemas difíciles de tratar, que en un principio serían imposibles o muy difícil de resolver con algoritmos tradicionales. (Ronneberger et al., 2015)

Un campo en el cual está muy presente la IA es en la medicina, donde se han alcanzado resultados muy prometedores. Una técnica que se está aplicando con frecuencia en esta área es el AP, debido a su versatilidad de adaptarse a problemas donde se requiere una constante mejora. Además de contar con alternativas entre su amplio repertorio de arquitecturas, donde algunas se adaptan mejor a la clasificación de patrones en imágenes (Ronneberger et al., 2015), al-

gunas otras al procesamiento de lenguaje natural, entre otras tareas. (Ingolfsson & Sachs, 1993)

La mayoría de técnicas de IA están inspirados en sistemas biológicos o evolutivos, en modelos que pueden aprender a resolver problemas sin la necesidad de información detallada. Algunas otras técnicas son más rápidas de entrenar, pero requieren de una descripción más afondo del problema, así como un dominio mayor del tema a resolver. (De Oliveira Martins et al., 2009)

5.1. Planteamiento del problema

La RD es la principal causa de pérdida visual no recuperable en los países industrializados, prevaleciendo en pacientes entre los 20 y 64 años, siendo responsable del 10% de nuevos casos de ceguera cada año (Tenorio, 2010). Así mismo, la retinopatía diabética es la principal afección que daña la retina. (Tenorio, 2010; Prado-Serrano et al., 2019)

Se trata de una enfermedad progresiva y asintomática hasta los estadios avanzados (Tenorio, 2010). En México la detección de RD se logra en la mayoría de los casos cuando el grado de severidad de dicha enfermedad está en sus últimas etapas, las cuales pueden ocasionar una pérdida de visión parcial o en el peor de los casos total.

Las causas principales de que la prevalencia de RD sea tan alta en países como México, radica en la poca información que se brinda acerca de la enfermedad, así como un escaso asesoramiento preventivo por parte de las instituciones de salud pública. Aunado a que los centros especializados para dicha enfermedad se encuentran en su mayoría en las capitales, haciendo más difícil el llevar un seguimiento pertinente a pacientes que viven en pueblos o comunidades de difícil acceso.

5.2. Justificación

El desarrollo de nuevo equipo de cómputo y mejoras a las tecnologías computacionales han colaborado al incremento en el uso de AP como técnica para la resolución de problemas altamente no lineales. La clasificación de imágenes médicas e identificaciones de patrones son un campo donde las CNN han demostrado tener un buen desempeño.

El incremento en la recolección de datos y su necesaria clasificación crean un campo donde las técnicas de AP juegan un papel importante. El entrenamiento de las RP no demanda una enseñanza de cada parámetro por sí solo. Si no un aprendizaje a base de ejemplos, recompensas y correcciones, lo cual vuelve a AP una técnica que se puede implementar en la mayoría de los campos de clasificación y optimización incluyendo el análisis de imágenes médicas. (Norouzi et al., 2014)

La creciente prevalencia de la diabetes crea una alarma que se ha desatendido durante mucho tiempo. Y cada paciente debe de tener la certeza de contar con herramientas que lo ayuden a monitorear cualquier aspecto que atente con su integridad física.

Antecedentes

6.1. Implementacion de aprendizaje automatizado

La IA está tomando una fuerza inmensa en todas las áreas de investigación. Ya que el manejo de grandes volúmenes de datos nos ha llevado a la implementación de estas técnicas como AA, AP, entre otras más técnicas en zonas donde tiempo atrás era impensable una aplicación de este estilo. A continuación, se enlistan algunos trabajos realizados con IA en diversas áreas de trabajo.

En economía, comercio y educación. se han realizado diversos trabajos como la selección de segmentos en países con bajo presupuesto para un equipo de manufactura basado en un sistema de soporte de decisión (Jiménez et al, 2014). También se ha utilizado esta misma técnica en la optimización de portafolios para una organización financiera e inversores individuales. (Dong et al., 2004) Lo cual vuelve muy interesante el enfoque que se le puede dar a dichas técnicas. Como por ejemplo también en el área médica, para ser más específicos en odontología, creando un sistema de aprendizaje que ayude a los estudiantes a entender los modelos adaptativos que se utilizan en esta rama de la medicina. (Gascueña, Fernández-caballero & González, Gascueña et al.)

En el área de informática y manejo de datos es más que notable la presen-

cia de algoritmos de IA en la resolución de problemas cotidianos como por ejemplo el spam en los servicios de correo electrónico (Kong et al., 2006), o incluso el análisis de nuevo malware, el entendimiento del funcionamiento y la prevención de ataques que pueden ocurrir a servidores o computadores domésticos. (Chitrakar & Chuanhe, 2012)

En la robótica como en la planeación de trayectorias se han realizado un sinnúmero de investigaciones que prueban el buen funcionamiento de las IA, como es el caso de control de robots utilizando técnicas como lógica difusa, (Hrbáček et al., 2010) o también técnicas como agentes con biorretroalimentación. (Okamura, 2004) Y en la planeación de rutas no necesariamente tiene que ser aplicado al mundo real, se puede implementar en caracterización animada. (Kalisiak & De Van Panne, 2001)

Los problemas que se intentan resolver por medio de técnicas de IA por lo regular son poco habituales que se lleven a una aplicación comercializable ya que por lo general se quedan en solo investigación, pero no quiere decir que no existan empresas que se dediquen a diseñar herramientas de automatización de este estilo. Son pocas y pequeñas, pero están creciendo a pasos agigantados. Al principio se puede dudar de la fiabilidad de estas técnicas ya que la aparición en el mercado empieza con empresas grandes y ya posteriormente se estandariza para el uso de pequeñas y medianas empresas.

En este trabajo se implementó una CNN a la clasificación de imágenes médicas, para ser más específicos a la RD. La detección de patrones en imágenes médicas es un área muy amplia ya que existen diversas enfermedades que se pueden categorizar, ya sean con imágenes RGB para la detección de cáncer (Dabeer et al., 2019) o imágenes de rayos-X para la búsqueda de pulmonía. (Rajpurkar et al., 2017)

Algunas empresas ya comercializan software para la automatización de diagnósticos. También algunas se dedican a la creación de aplicaciones especializadas para resolver problemas específicos que sus clientes requieran, claro si es que esta en las posibilidades de dichas empresas.

Aidence.

Se especializa en el diagnóstico y visualización de resultados en radiografías. Su principal propósito es la automatización y mejora de lectura para todos aquellos que requieran optimizar sus diagnósticos relacionados con radiografías. Pero también brinda ayuda a hospitales y científicos con un software intuitivo en el análisis de imágenes médicas.

MetaOptima.

Dedicada al tratamiento de datos dermatológicos y análisis de la piel. Con la misión de facilitar y mejorar el cuidado de la piel para cualquier persona. El software está dedicado a médicos y a pacientes con dos versiones disponibles. Esta aplicación necesita aditamentos de hardware extras. Con un análisis en la nube.

Diagnostics.ia

Es una empresa que se dedica a crear software automatizado para optimizar el proceso de diagnóstico médico. Presume de tener buenos resultados en sus proyectos realizados, utilizando IA en todos sus proyectos que comenzaron en el año 2014.

6.2. Clasificación de RD

Existen varias técnicas basadas en visión por computadora que utilizan imágenes de fondo de ojo para clasificar el grado de severidad en la RD. (Welikala et al., 2014) Los métodos más utilizados para la detección de patrones en imágenes son técnicas de IA (Ronneberger et al., 2015; Krizhevsky et al., 2017; Szegedy et al., 2015; Lecun et al., 2015). El objetivo de estos sistemas, así como el propuesto en este trabajo, es la creación de nuevas herramientas que faciliten y mejoren la calidad de vida de las personas.

La clasificación de los factores que denotan la RD es un tema que se ha intentado resolver con AA. Algunas clasificaciones o búsquedas de patrones se han realizado con AA como en 2014 utilizando operadores lineales con los cuales buscaban clasificar la RD proliferativa con buenos resultados de sensibilidad del 86.2 y especificidad del 94.4 %. 2016 Srivastava et . que buscaba microaneurismas y hemorragias, a lo cual alcanzo una exactitud del 97 % y 92 % respectivamente, utilizando 143 imágenes las cuales no indica el grado de RD. En 2016, Pratt et al. alcanzó una exactitud del 75 % clasificando RD en sus 5 grados de severidad según la clasificación escocesa utilizando redes neuronales convolucionales y una sensibilidad del 95 % y especificidad del 35 % debido a la naturaleza de la base de datos. Otro trabajo donde ahora se utilizó AP en 2017 Quellec et al. alcanzó una exactitud de 95 % para imágenes con RD proliferativa y 90,000 imágenes que se utilizaron tanto para entrenamiento y validación. También en 2017 Litjens et al. demostró una exactitud de 90 % clasificando con CNN y categorías no descritas al igual que las imágenes utilizadas. En 2018 Ramachandran et al. utilizando RNP alcanzo un intervalo de confianza de 80.7 % - 99.5 % en la detección de factores relacionados con la RD y un conjunto de

imágenes reducido de 1,685. Utilizando múltiples RNP, en 2018 Wang et al. alcanzó 95.68 % de exactitud. Uno de los artículos más recientes en 2018 por De Fauw et al. detectó múltiples enfermedades oculares entre ellas la RD. La exactitud promedio de la arquitectura es de 96 % esto incluye todas las enfermedades oculares que se intentan diagnosticar en el documento. La tabla 6.1 muestra un resumen de los trabajos mencionados anteriormente en la detección de RD. Cabe mencionar que cada trabajo crea categorías diferentes para la clasificación, con excepción del trabajo realizado por Pratt et al. Que se basa en la clasificación escocesa.

Tabla 6.1: Estado del arte. Artículos mas relevantes

Autor	Técnica	Resultados	Búsqueda	No. de imágenes
Sinthanayothin (2014)	Operador lineal modificado y clasificación dual	86.2 % de sensibilidad y 94.4 % de especificidad.	Retinopatía diabética proliferativa.	60
(Srivastava et al., 2017)	Múltiples kernels de aprendizaje	97 % para microaneurismas y 92 % para hemorragias	Microaneurismas y hemorragias.	143
(Pratt et al., 2016)	Redes neuronales convolucionales.	75 % de exactitud.	Clasificación de RD, 5 categorías.	80,000
(Catania & Nicolitz, 2018)	Redes neuronales convolucionales.	90 % de exactitud.	Clasificación de RD.	No reporta
(Quellec et al., 2017)	Aprendizaje profundo, ConvNets.	95.40 % de exactitud.	Clasificación de RD, 2 categorías.	90,000
De Fauw et al.(2018)	Aprendizaje profundo.	96 % de exactitud.	Detección de enfermedades en la retina.	14,884
(Ramachandrar et al., 2018)	Redes neuronales profundas	80.7 % de exactitud. 99.5 % de exactitud.	- de Clasificación de RD.	1,685

Como se muestra en la tabla 6.1 se han implementado métodos de inteligencia artificial para la clasificación de diferentes características que denotan la RD, algunos trabajos se realizaron con AA y otros con AP, La tabla 6.2, denota las características más significativas entre estos dos métodos.

Tabla 6.2: Diferencias significativas entre aprendizaje automático y profundo

Aprendizaje automático	Aprendizaje profundo
+ Buenos resultados con pocos datos	- Requieren grandes cantidades de datos
+ Rápido entrenamiento	- Carga computacional intensiva
- Se necesitan diferentes característica y clasificadores para alcanzar buenos resultados	+ Las características y clasificadores se aprenden automáticamente
- La exactitud no mejora después de un punto	+ La exactitud no tiene un límite de mejora

Tiempo atrás era imposible pensar en crear RN con gran cantidad de parámetros entrenables para la detección de patrones en imágenes, o incluso en cualquier otra tarea que se pretendiera utilizar aprendizaje profundo debido al alto costo computacional de estas topologías. Gracias a los grandes progresos en términos de poder computacional y las grandes cantidades de información que son posibles de recolectar y clasificar en la actualidad, las redes neuronales han tenido un gran auge, pero no solo estos factores han favorecido la continua mejora del aprendizaje profundo. Las nuevas ideas, algoritmos y las diversas arquitecturas de las redes neuronales son así mismo un factor importante que de igual manera mejora el desempeño de las redes neuronales. (Lecun et al., 2015)

En las variadas arquitecturas de CNN destacan algunas por encima de otras. No solo por su exactitud si no también debido a que algunas cuentan con gran cantidad de parámetros a entrenar, lo que no es un factor favorable, ya que ocasiona que el costo computacional sea más alto. Es necesario recalcar que algunas arquitecturas funcionan bien para algunos problemas y otras no, esto indica que se deben de probar diferentes topologías y dictaminar cual es la más indicada para el problema.

6.3. Retinopatía diabética

La retina es la zona que se encuentra en la parte posterior del ojo y contiene una red de vasos sanguíneos. En este sector se pueden manifestar algunas enfermedades tales como glaucoma, problemas degenerativos maculares, RD, y más enfermedades. (Mustafa et al., 2018) La retinopatía diabética es la principal causa de pérdida visual no recuperable en los países industrializados, prevaleciendo en pacientes entre los 20 y 64 años de edad, siendo responsable del 10 de nuevos casos de ceguera cada año. (Tenorio, 2010) Así mismo, la retinopatía diabética es la principal afección que daña la retina. (Prado-Serrano et al., 2019)

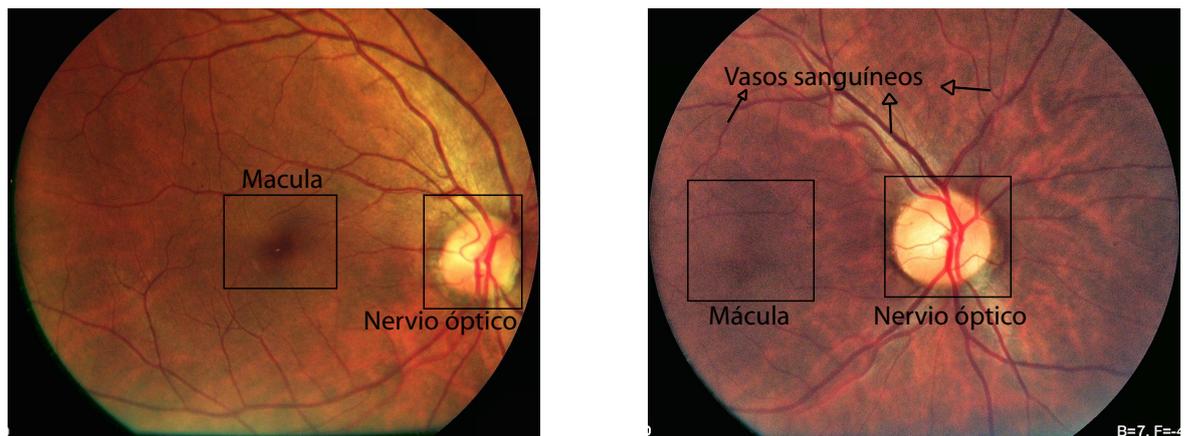


Figura 6.1: La primera imagen de retina esta centrada en la mácula. La siguiente imagen esta centrada en el nervio óptico.

La RD es una enfermedad que afecta la parte posterior del ojo, en la cual se encuentran la mayor cantidad de vasos sanguíneos, así como otras partes importantes del ojo. Podemos encontrar la macula que es la zona de mejor visión y el nervio óptico el cual es el epicentro de todas las arterias que recorren la retina. Por ende cualquier objeto que impida el paso de luz en estos dos puntos

importantes es un riesgo para la visión de la persona. En la fig. 6.2 se muestran las partes principales de la retina.

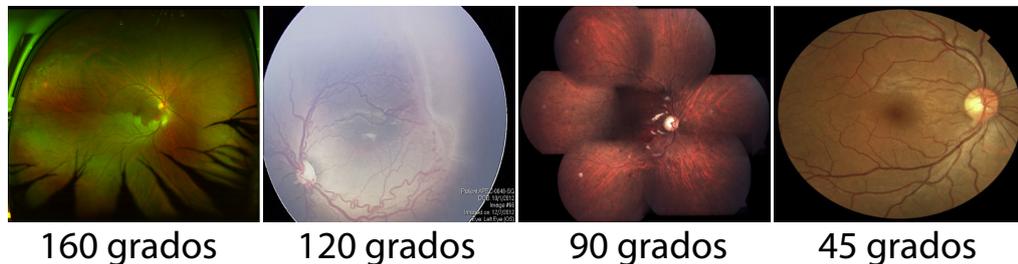


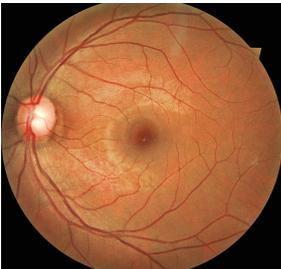
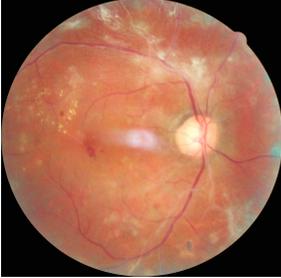
Figura 6.2: Ejemplos de imágenes de retina.

La RD se caracteriza por la aparición de diferentes factores como exudados, microaneurismas, hemorragias y neovasos; que amenazan la integridad de la parte posterior del ojo, el paso de luz en las zonas de mejor visión y el desprendimiento de la retina. En la RD la aparición de ciertas características son mas importantes que otras. Así como la zona en donde las anomalías prevalecen. (Zachariah et al., 2015) Los factores se describen en la tabla 6.3. La clasificación mas utilizada es la Escocesa, la cual sirve como base para la separación de las imágenes.

Existen diferentes tipos de métodos para la toma de imágenes de la retina. La mas utilizada al menos en México, consiste en dilatar la pupila por medio de gotas que se colocan directamente en el ojo. Después de un determinado tiempo la pupila se dilata lo suficiente para que una cámara especial tome una foto de la parte posterior del ojo. Dicha cámara puede tomar cierto diámetro de la retina y se clasifican por grados. (García-Aguirre et al., 2017) Un ejemplo de los diferentes grados de captura se pueden observar en la fig. 6.2. Como se puede observar en la tabla 6.3. los grados de RD leve y moderada presentan características difíciles de ver a simple vista, se requiere de un entrenamiento

para poder distinguir y clasificar los factores. Para las ultimas dos categorías se pueden observar factores perceptibles a simple vista, pero es difícil separar las categorías con un ojo no entrenado.

Tabla 6.3: Grados de severidad según la clasificación Escocesa.

Grado	Características	Ejemplo
R0 - RD Ausente	Ausente.	
R1 - RD Leve	Microaneurismas, exudados, hemorragias en uno o ambos hemicampos	
R2 - RD Moderada	Las mismas características de la RD leve en mas cantidades.	
R3 - RD Severa no proliferativa	Anomalías microvasculares intrarretinianas, cuentas venosas, características de R1 y R2.	
R4 - RD Proliferativa	Hemorragia vítrea, desprendimiento de retina.	

En la RD proliferativa se pueden encontrar manchas de color negro por diversas zonas de la retina, pero esto no indica un factor negativo del todo, ya que algunas de estas coloraciones se deben a intervenciones con láser para cauterizar hemorragias que se presentaron con anterioridad.

6.4. Aprendizaje profundo

Las técnicas de computación inteligente se crearon para resolver problemas altamente no lineales, se aplican para resolver conflictos en los cuales los algoritmos convencionales no pueden lograr buenos resultados. (Gollapudi & Gollapudi, 2019) La mayoría de los algoritmos de IA están inspirados en comportamientos biológicos y crean un campo que cada día se expande mas y mas, gracias al desarrollo de nuevas técnicas y modificaciones a las ya existentes. En este proyecto se utilizo aprendizaje profundo el cual pertenece a la rama de AP. En la fig. 6.3 podemos observar como se agrupan las técnicas antes mencionadas.

Aprendizaje profundo ha cambiado mucho su significado en los últimos tiempos, pero una definición aceptada seria que AP trabaja con redes neuronales que contienen un gran número de parámetros y capas en sus diferentes arquitecturas. (Yamashita et al., 2009) Algunos trabajos también definen a las RP como aquellas redes que cuentan con mas de dos capas ocultas. Las arquitecturas fundamentales de AP pueden dividirse en 4:

- Redes no supervisadas pre entrenadas.
- Redes neuronales convolucionales.
- Redes neuronales recurrentes.
- Redes neuronales recursivas.

A partir de las cuatro arquitecturas ya mencionadas se han derivado variantes dentro de las mismas. También han surgido nuevos modelos como AR que en recientes experimentos han demostrado buenos resultados. (Kaelbling

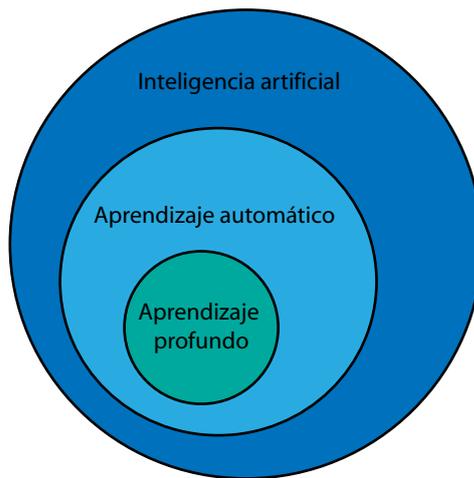


Figura 6.3: Representación del área de computación inteligente.

et al., 1996) La fig 6.4 muestra una red neuronal profunda.

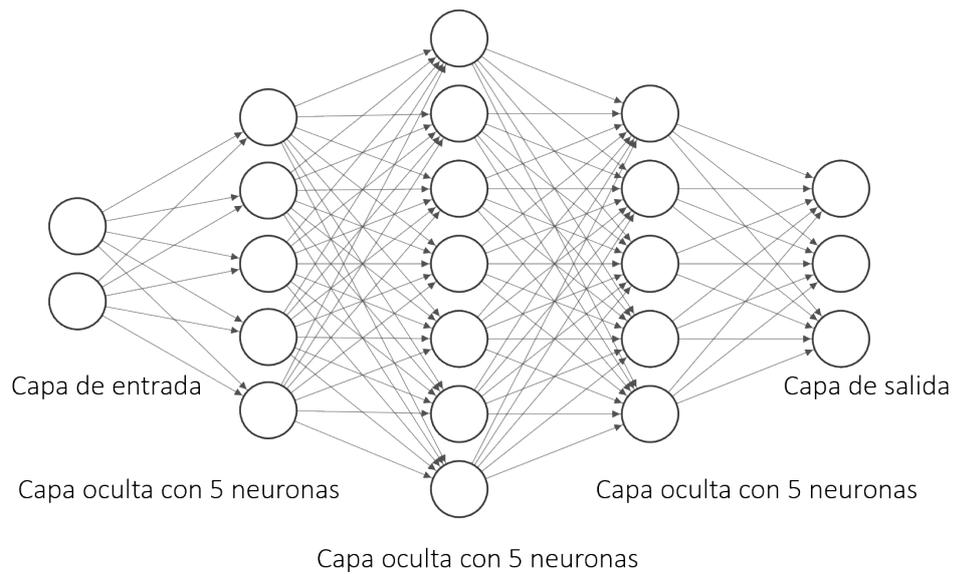


Figura 6.4: Ejemplo de la composición de una Red neuronal tradicional.

6.4.1. Arquitecturas de aprendizaje profundo.

En los siguientes párrafos se presentarán algunas arquitecturas de AP. Podríamos decir que se presentan las redes principales o comunes dentro del área de AP, ya que algunas son variantes de otras, lo cual crea un mar de arquitecturas con pequeñas variantes.

- Redes de base radial (RNA): Las RNA están siendo olvidadas y reemplazadas por sus compañeras más modernas. Una de las limitantes de esta red es la poca profundidad, la cual restringe a dos capas ocultas por lo general. Pero aun con esta debilidad se puede implementar a resolver algunos problemas específicos. El mayor problema es la poca exploración de su potencial en la literatura.
- Redes neuronales recurrentes (RNN): Toca mencionar a las RNN las cuales se caracterizan por el procesamiento de datos secuenciales. Debido a lo antes mencionado las RNN se implementan en problemas como la comprensión de textos, secuencias de datos discretos, secuencias de datos biológicos, entre otros más. En estos casos la entrada se encuentra de la forma $\hat{x}_1 \dots \hat{x}_n$ donde \hat{x}_t es un punto de d-dimensiones recibido en el tiempo t .
- Redes neuronales convolucionales (CNN): Las CNN están inspiradas en las redes biológicas y comúnmente son usadas en visión computacional para la clasificación de imágenes y detección de objetos. La principal inspiración para la creación de esta arquitectura surgió a partir del estudio del córtex visual de los gatos por los científicos Hubel y Wisel's , En el cual se explica cómo algunas neuronas en particular se excitan a partir de una porción

específica del campo visual. La primera red creada usando el concepto antes mencionado fue la llamada “neocognitron”. Que después paso a ser la ya conocida arquitectura “LeNet-5”.

Las capas convolucionales, realizan operaciones convolucionales, la cual utilizan filtros que pasan la información de activación de una capa a otras. Las operaciones convolucionales usan filtros de tres dimensiones con una profundidad de igual al número de neuronas.

En los últimos años se han implementado nuevas técnicas de aprendizaje, entre algunos modelos se encuentra el AR, el cual ha demostrado un gran avance en el área de IA, esto se puede observar en el documental Alpha GO.

- Aprendizaje por refuerzo: Dentro de todas las técnicas de IA los modelos basados en AR son considerados como los más avanzados. (Kulkarni, 2012) AR interactúa con el medio que lo rodea. El medio por si solo puede mostrar múltiples estados. Estas pequeñas definiciones pueden resultar un poco ingenua, pero esto permite al desarrollador crear modelos avanzados de IA para resolver tareas complejas.

Para definir con mayor claridad el AP interactúa con el medio para aprender a tomar la mejor posible acción denominada ac sobre el estado et en el que se encuentra el medio en el paso t . La acción del agente actual cambia el estado del medio de St a $St + 1$ y genera una recompensa rt para el agente. De esta manera el agente tomara la mejor posible acción para el nuevo estado ($St + 1$), obteniendo así una recompensa $rt + 1$ y así sucesivamente. Durante algunas iteraciones el agente tratará de mejorar su decisión de cuál será la mejor acción a tomar dado un estado del medio usando recompensas que recibe durante el entrenamiento. En la fig. 6.5 podemos ver lo

antes mencionado.

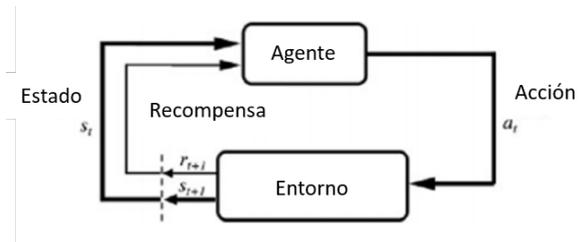


Figura 6.5: Diagrama básico de aprendizaje por refuerzo. (Kulkarni, 2012)

Para el desarrollo de este artículo se trabajaron con la arquitectura CNN ya que resultan desempeñarse bien cuando se refiere a imágenes como datos de entrada, y por lo general se aplican en esta área. En el siguiente subcapítulo se describe con mayor profundidad la técnica mencionada en este párrafo.

6.4.2. Redes Neuronales Convolucionales

Las redes convolucionales están inspiradas en la organización animal del cortex visual (Du & Cheng, 2014). Son similares a las redes profundas convencionales, la diferencia más grande entre estas dos redes es la conexión entre capas. Las CNNs usan convoluciones en lugar de una multiplicación general de matrices en al menos una de sus capas (Liu et al., 2017). Si una RP normal se utiliza en trabajos con imágenes, se necesitaría un número enorme de neuronas, por ende un número aún más grande de conexiones, lo cual puede resultar en un sobre entrenamiento y en una carga computacional más alta (Gollapudi & Gollapudi, 2019). En cambio las CNN aprovechan el volumen de entrada para transformarlo en un volumen de salida, esto se refiere a que una neurona se encarga de una región de la imagen o matriz. Esto significa que almacenar menos parámetros, lo que reduce tanto los requerimientos de memoria del modelo y mejora su eficiencia estadística. Lo que indica que calcular la salida requiere menos operaciones.

Si hay M entradas y N salidas, entonces la matriz de multiplicación requiere $(M \times N)$ parámetros, esto implica que el tiempo de ejecución del algoritmo es $O(m \times n)$. Si limitamos el número de conexiones, cada salida puede tener K . Entonces el enfoque tiene $(k \times n)$ parámetros y $O(k \times n)$ tiempo de ejecución. En la mayoría de los casos por no decir en todos k suele ser menor que m . Y m suele a ser del mismo tamaño que n , k es prácticamente insignificante comparado con $m \times n$. La fig. 6.6 y la fig. 6.7 representan gráficamente lo antes mencionado. La primera muestra una perspectiva en dirección de la entrada hacia la salida y la segunda en dirección opuesta.

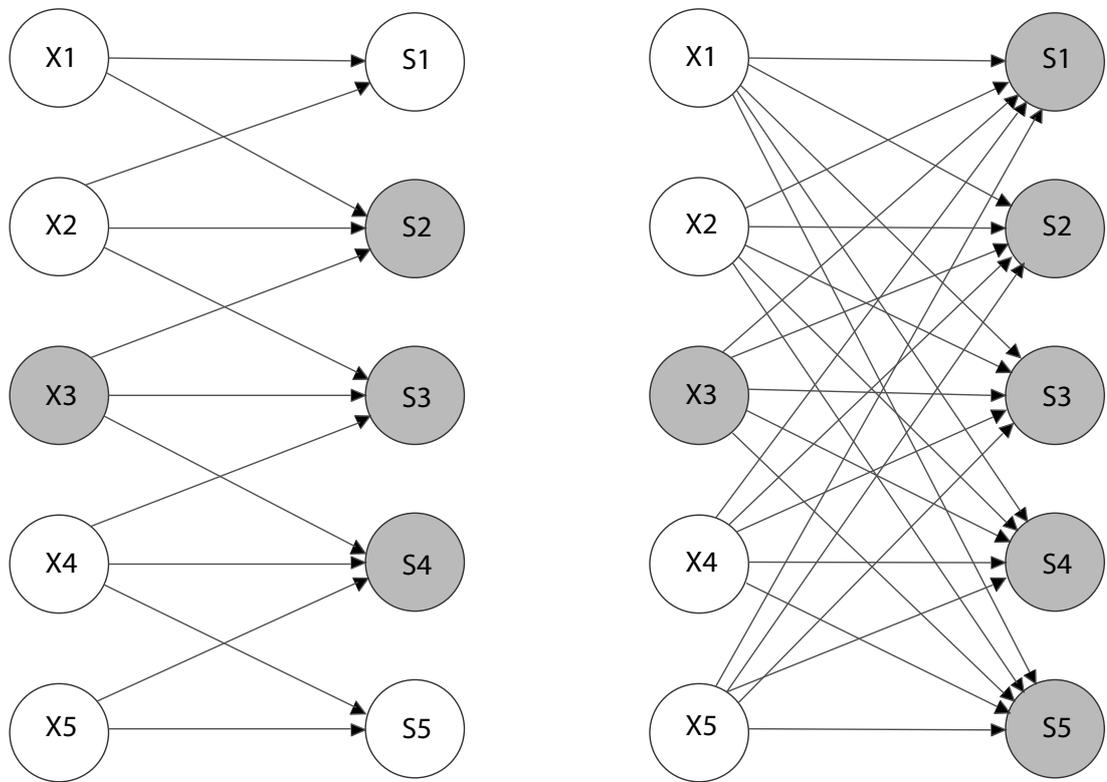


Figura 6.6: En la CNN de la figura una neurona de entrada tiene repercusión en tres neuronas de la siguiente capa consideradas como salidas en este caso. En cambio en una red profunda tradicional una neurona tiene influencia en todos los nodos de la siguiente capa, como se puede apreciar.

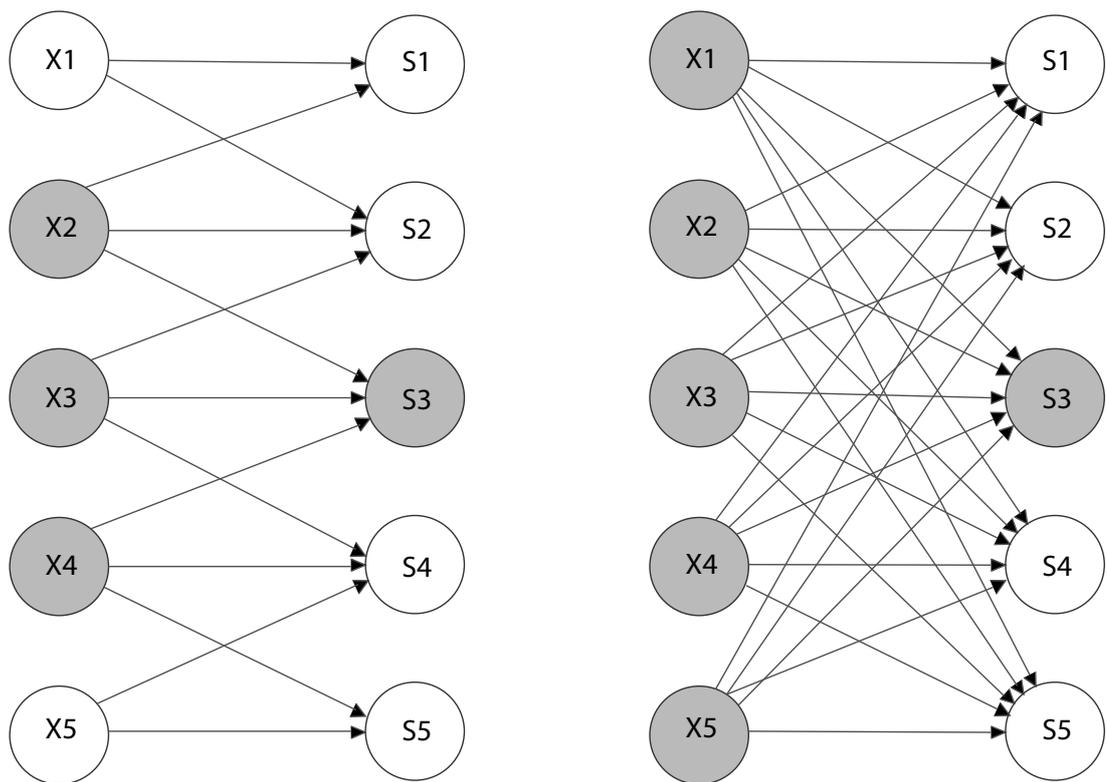


Figura 6.7: En esta figura se puede observar que las neuronas de entrada de la CNN tan solo a un nodo de salida. En cambio las neuronas de toda la capa de entrada influyen en una sola neurona de salida en la RN típica.

Cuando se utilizan imágenes de entrada en una CNN se aplican las operaciones convolucionales en dos ejes. Lo cual indica que se utilizara un Kernel de dos dimensiones:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n) \quad (6.1)$$

La convolución es conmutativa, así que se puede escribir de la siguiente

manera:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n) K(m, n) \quad (6.2)$$

Usualmente la ecuación 6.2 es mas sencilla de implementar en librerías de aprendizaje automático, "la razón es que hay menos variación en el rango de valores válidos de M y N ".

Los filtros aplicados en las transformaciones de las CNNs pueden detectar bordes, formas y patrones. Entre mas profunda sea la capa mas abstracto es el patrón. En la fig. 6.8 podemos observar la configuración básica de una CNN. La primera etapa de una CNN es la aplicación de convoluciones para producir un conjunto de activaciones lineales. En la segunda etapa se aplican funciones de activación no lineales, esta etapa se suele denominar etapa de detección. Y por ultimo se utiliza una función de agrupación para modificar la salida de la capa. (Yamashita et al., 2009)

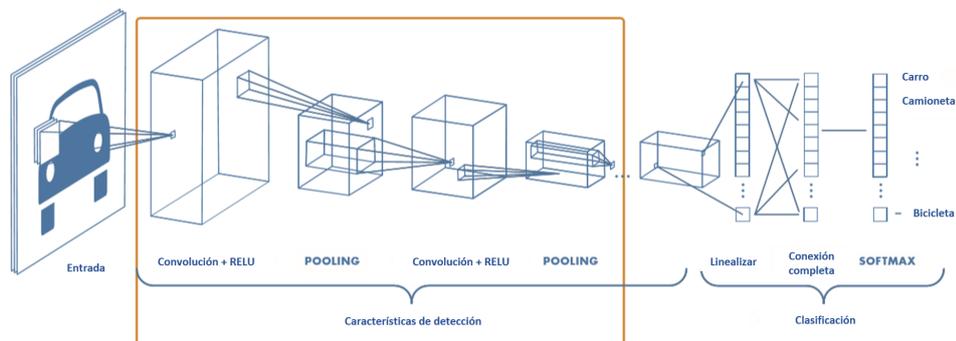


Figura 6.8: Estructura CNN. (Mathworks, 2017)

Redes residuales.

Una de las vertientes más interesantes en las CNN son las redes residuales, ya que la mayoría de las arquitecturas de CNN son denominadas lineales o planas, debido a que sus capas están una tras otra, esto quiere decir que a lo largo del entrenamiento los datos de entrada en una capa son totalmente diferentes a la salida, esto implica que no se toma en cuenta si hay información importante en la entrada, olvidando por completo esos parámetros a la salida de la capa. Este problema se denomina degradación e indica que no todos los sistemas son fáciles de optimizar. (Krizhevsky et al., 2017) Lo anteriormente mencionado se puede ver en la figura siguiente.

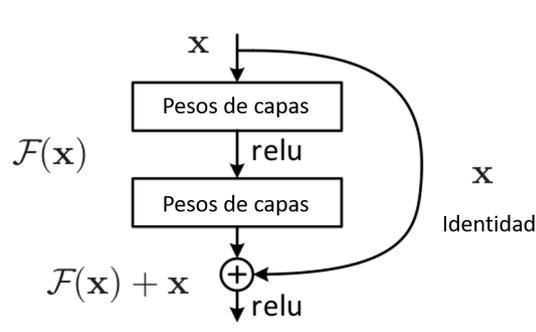


Figura 6.9: Bloque de entrenamiento residual. (Krizhevsky et al., 2017)

La implementación de RR mejora el tiempo de entrenamiento, lo que ocasiona que se llega a una convergencia en un número menor de épocas. (Krizhevsky et al., 2017) Esto es mencionado y provado en el artículo Deep Residual Learning, y la fig. 6.9 muestra uno de los resultados obtenidos en dicho artículo. Demostrando que se puede encontrar la convergencia con mayor rapidez.

Esta nueva forma de implementar CNN empezó a resonar por todos lados, pero el punto más alto de popularidad que alcanzaron estas redes denominadas de tipo residual, se dio cuando un equipo de investigadores de Google desarrollaron la llamada CNN Inception, que posteriormente evolucionó a Inception

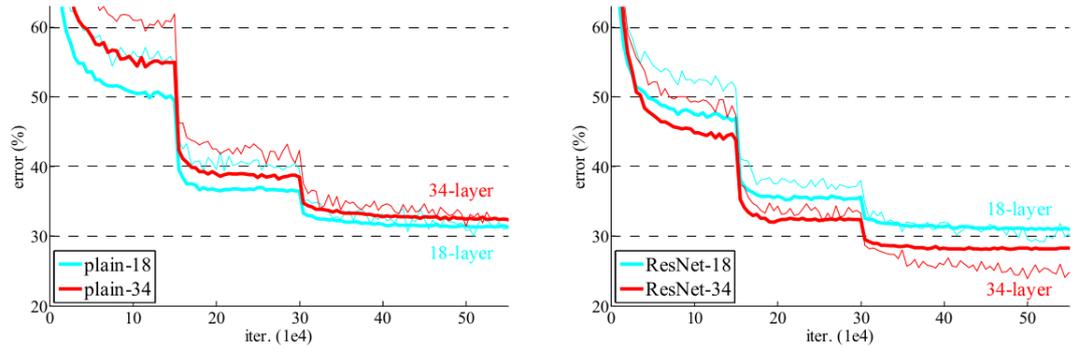


Figura 6.10: Gráfica de los resultados obtenidos utilizando redes planas y redes residuales respectivamente. (Krizhevsky et al., 2017)

V4, (Krizhevsky et al., 2017) la cual gana premios en competiciones de clasificación, venciendo a las actuales campeonas basadas en redes planas.

6.4.3. Parámetros de una CNN

Las funciones principales que conforman una CNN están descritas a continuación. Es necesario recalcar que solo se mencionan las funciones principales ya que existen variantes de estas mismas funciones.

Kernel.

Kernel es el parámetro que usa la capa de convolución para convolucionar los datos de entrada. El kernel contiene dos parámetros. El llamado paso que si en su caso es igual a uno, la salida será de igual tamaño que la de los datos de entrada, si el paso es de dos producirá una salida de la mitad del tamaño que la entrada; y el tamaño del Kernel el cual puede ser cualquier dimensión de un tamaño rectangular.

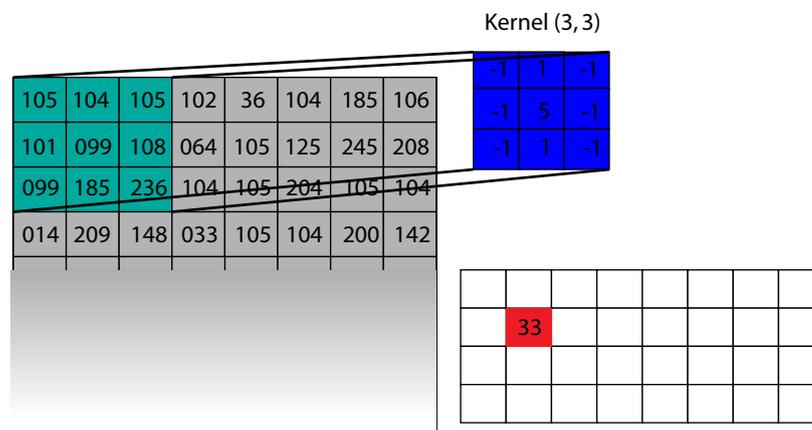


Figura 6.11: Descripción de kernel

Pooling.

La función de agrupación reemplaza la salida de la capa con un resumen estadístico de las salidas cercanas. (Yamashita et al., 2009) Además la función

agrupación ayuda a reducir la variación de entrada. Esto nos ayuda a determinar si una característica buscada en los datos de entrada esta presente, en vez de enfocarnos en donde esta exactamente. Un ejemplo sencillo seria la detección de rostros, a nosotros nos interesa el saber si en la imagen de entrada se encuentra una nariz, una boca, dos ojos, etc. Es indispensable el saber que estos factores existen, posteriormente enfocarse en si los ojos estan en la posición correcta, uno de lado izquierdo y otro de lado derecho, esto nos lleva al querer guardar la característica de la posición. Debido a lo mencionado en este párrafo, la función agrupación por lo regular contiene menos parámetros que una capa de convoluciones.

La función pooling como se mencionó anteriormente trabaja en pequeñas regiones de tamaño $P_q \times P_q$ en cada capa, y produce otra capa con la misma profundidad. Por lo regular se utiliza un stride $S_q > 1$ para pooling. En tal caso, el largo de la nueva capa seria $(L_q - P_q) / S_q + 1$ y la amplitud debe ser $(L_q - P_q) / S_q + 1$ por lo tanto la función pooling reduce drásticamente las dimensiones de cada mapa de activación.

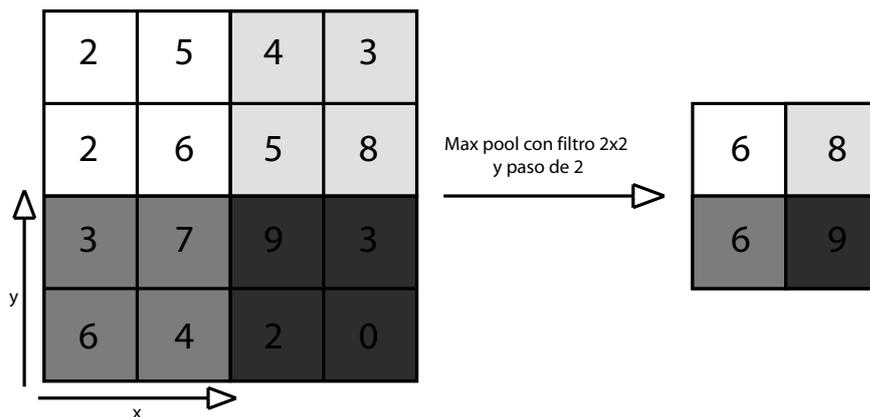


Figura 6.12: Descripción de la función Pooling

Strides.

Hay otras maneras de reducir la salida de una CNN, por ejemplo los strides son una forma en la que se controla donde se aplica la función de convolución, ya que no es necesario en cada posición espacial.

Cuando un stride de S_q es usado en la q th capa, la convolución es desarrollada para ser usada en $1, S_q + 1, 2 S_q + 1$, y así durante todas las dimensiones de las capas. El tamaño espacial de la salida desarrolla la convolución tiene un ancho de $(L_q - F_q)/S_q + 1$ y un ancho de $(B_q - F_q)/S_q + 1$. Como resultado el uso de strides termina siendo una reducción de cada dimensión espacial por un factor de aproximadamente S_q .

Dropout

Es una técnica muy eficiente para evitar el sobre-entrenamiento. La acción consiste en invalidar algunas neuronas, esto ayuda a que cada neurona de la capa influya en la decisión de salida, para tener un balance de importancia entre las neuronas de la capa. Se determina cuales nodos serán desactivados en base a un porcentaje de uso, donde 1 es el uso de todas las neuronas y 0.5 la mitad. Esto nos indica que entre mas bajo sea el valor de Dropout mas neuronas serán deshabilitadas en cada paso de entrenamiento. Se debe tener cuidado de no colocar un valor muy bajo ya que esto puede ocasionar que la CNN tarde o incluso no llegue a una convergencia. En la siguiente figura se puede observar la diferencia entre una capa sin y con dropout.

El proceso de entrenamiento con dropout sigue los siguientes pasos:

- Los nodos de entrada son muestreados con una probabilidad p_i , y los nodos intermedios son muestreados con una probabilidad p_h . Además, todos los ejemplos son independientes de uno a otro. Cuando un nodo es removido desde la red, todas las aristas son removidas también.

- Muestra una simple instancia de entrenamiento o un pequeño paquete de instancias de entrenamiento.
- Se actualizan los pesos de las aristas retenidas en la red usando back-propagation sobre la instancia de entrenamiento muestreada o sobre un conjunto pequeño de datos.

Tipos de clasificaciones.

Los diferentes problemas a resolver demandan una clasificación variada. En ocasiones se desea realizar solo una separación binaria, en casos específicos clasificaciones con mas de dos categorías. Para estos trabajos variados existen diferentes métodos de clasificación.

- One-hot encoding: Este método convierte las etiquetas de salida en vectores o nodos. Si se realiza una clasificación de gatos y perros se puede representar como una separación binaria representada por un vector de salida $[0, 1]$ o $[1, 0]$ respectivamente. Pero si la clasificación consta de 1000 clases, se tendrá un vector con tamaño de 1000 enteros. En estos casos es mejor utilizar el método de Softmax.
- Softmax: Es una manera de forzar la salida de la RN representada como la suma de 1. Lo cual puede ser considerado como parte de una distribución de probabilidad. Lo cual lo hace un metodo mas que viable para una clasificación con múltiples clases.
- Cross-entropy: Compara la distancia entre las salidas de softmax y one-hot encoding. El objetivo es la reducción del valor de la función perdida. La probabilidad tiene que ser maximizada para la etiqueta correcta. El valor

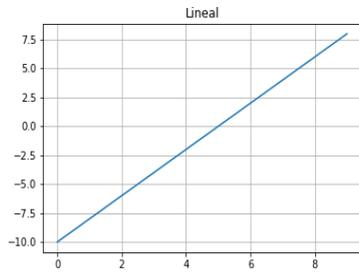
logarítmico es utilizado para una estabilidad numérica, ya que maximizar una función es equivalente a minimizar el negativo de la misma función.

6.4.4. Funciones de activación.

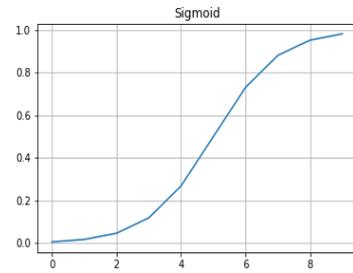
Las FA sirven para propagar la salida de cada nodo en una capa a la siguiente capa. Estas FA son funciones escalares. Se usan las FA para que las neuronas de las capas ocultas en una red neuronal introduzcan la no linealidad dentro de sus posibilidades de modelado. La mayoría de FA pertenecen a la clase logística de transformación. (Patterson & Gibson, 2008)

- Lineal: La transformación lineal básicamente una función de identidad $f(x) = Wx$ donde la variable dependiente tiene una relación proporcional con la variable independiente. La salida de esta FA se puede observar en la figura 1.6 a.
- Sigmoid: Esta FA, puede reducir los valores extremos o los valores atípicos en los datos, sin la necesidad de revolverlos. En su parte central se encuentra el límite de decisión. Como se puede observar en la figura 1.6 b. Básicamente esta FA puede convertir variables que se encuentran cerca del rango infinito en simples probabilidades entre 0 y 1.
- Tanh: La FA tanh es una función trigonométrica hiperbólica, que representa una proporción entre los lados opuesto y el adyacente de un triángulo rectángulo. A diferencia de la función Sigmoid, el rango de normalización está entre -1 a 1 . Lo cual le da la ventaja de lidiar con números negativos más fácilmente. La representación gráfica se encuentra en la fig 1.6 c.
- ReLU: La rectificación lineal activa un nodo solo si la entrada supera una cierta cantidad. Cuando la entrada es menor a cero la salida es cero, pero cuando la entrada alcanza un umbral, esta tiene una relación lineal con la variable dependiente. Como se puede apreciar en la figura 1.6 d.

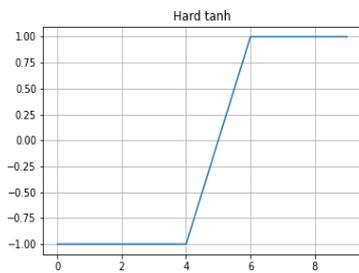
- Softplus: Esta FA es una versión suave de ReLU. Como se puede ver en la figura 1.6. e.



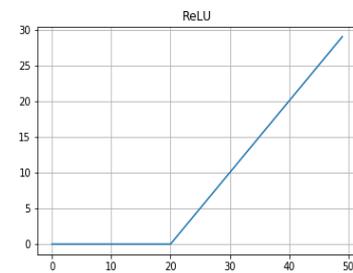
(a)



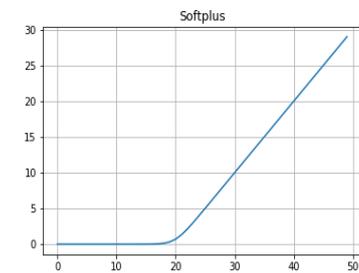
(b)



(c)



(d)



(e)

Figura 6.13: Respuestas gráficas de FA descritas en el capítulo.

6.4.5. Entrenamiento de una CNN

Backpropagation

Toda RN debe encontrar el óptimo valor para cada nodo en cada capa, por ende, se pueden aplicar diferentes métodos para encontrar dicho óptimo. El algoritmo principal se denomina Backpropagation, el cual aplica la regla de la cadena de cálculo diferencial, en donde el error gradiente estando en términos de sumas de productos, encontrando el gradiente local a lo largo de los diversos caminos desde el nodo de salida. Esto implica que un número exponencial de caminos, que se pueden calcular eficientemente utilizando programación dinámica. Este método consta de dos fases:

- Fase delantera: Las entradas para un tiempo de entrenamiento alimentan la RN. Esto resulta en una acción de cálculo en cascada a través de las capas.
- Fase hacia atrás: El objetivo de esta fase es aprender sobre el gradiente de la función de pérdida respecto a los diferentes pesos usando la regla de la cadena de cálculo diferencial. Esto actualiza el gradiente de los pesos.

Sobre entrenamiento

Es un problema común dentro de AP ya que se suele dar cuando un RN se ajusta de modo que se desempeña bien con los datos de entrenamiento, pero si se alimenta con datos externos no presenta buenas predicciones. Esto implica que se puede llegar a encontrar buenos resultados durante el entrenamiento, pero no se sabrá el desempeño real hasta que no se comprueben los resultados con datos reales externos a los datos de validación.

Con pocos datos suele suceder el sobre entrenamiento, ya que no se cuentan con suficientes ejemplos como para llegar a ajustar los pesos de una RN correctamente. El problema de las pocas cantidades de información se tenía con frecuencia en tiempo atrás, pero hoy en día el manejo de grandes cantidades de datos en todos los ámbitos, abrió paso al uso de AP con más regularidad, ya que con más recurrencia se cuenta con lo necesario para entrenar cualquier RN que se diseñe. Además, que se han desarrollado técnicas que ayudan a evitar el sobre entrenamiento. Las cuales se mencionarán a continuación.

Paro de emergencia.

Es una forma de regularización, en la cual cuando el gradiente no mejora después de determinadas épocas o corridas, se puede tomar la decisión de detener el proceso de entrenamiento en ese punto, pero se mantiene guardado los pesos y parámetros de la red neuronal. El paro de emergencia actúa como un regularizador debido por que restringe el espacio de los parámetros.

Batch Normalization.

Otra técnica habitual mente ocupada para la reducción del riesgo de sobre entrenamiento es la normalización de los datos. Uno de los problemas que suele suceder cuando se tiene una RN muy grande en las capas intermedias es la distribución de la función de activación que cambia con bastante frecuencia durante el proceso de entrenamiento. Esto hace más lento el proceso de entrenamiento debido a que cada capa debe aprender a adaptarse por sí sola a la nueva distribución en cada paso de entrenamiento. Este problema es conocido como internal covariate shift (ICS).

El proceso de batch normalization sigue el proceso mostrado a continuación:

1. Calcular la media y varianza de las capas de entrada que sus ecuaciones que describen son la 6.3 y 6.4 respectivamente.

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (6.3)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (6.4)$$

2. Normalizar las capas de entrada usando la estadística previamente calculada representada en la ecuación 6.5.

$$\hat{x}_B = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + E}} \quad (6.5)$$

3. Escalar y combinar el orden para obtener la salida de la capa, utilizando la ecuación 6.6.

$$y_i = \gamma \hat{x}_i + \beta \quad (6.6)$$

Nótese que γ y β aprenden durante el entrenamiento a través de los parámetros originales de la red.

Entrenamiento GPU vs CPU.

Uno de los principales impedimentos para la implementación de RP como técnica para la resolución de problemas era el poder de computo. Era casi imposible pensar que alguna máquina soportara toda la carga de cálculos que requiere una RP o cualquier otra técnica de IA. Durante mucho tiempo no se miraban las técnicas de IA debido al pobre poder de computo que se tenía, hasta que poco a poco se fue introduciendo el cálculo en paralelo y la creación de GPU's. (Shi et al., 2017)

Tabla 6.4: Resultados del entrenamiento con GPU y CPU

	No. De épocas	Tiempo promedio de entrenamiento por época	Tiempo de entrenamiento total	Exactitud	Perdida
GPU	15	42.6 segundos	0.177 Hrs	80 %	46 %
CPU	15	1188.7 segundos	4.9 Hrs	78 %	52 %

La tabla 6.4 muestra el entrenamiento de una CNN mediante GPU y CPU para la clasificación de perros y gatos, con el fin de poder observar las diferencias en los tiempos de entrenamiento y si esto afecta los resultados finales. La CNN consta de dos capas convolutivas con 256 filtros y una tercera capa para linealizar la salida con 128 nodos que posteriormente se conectan a una capa de una sola neurona para la clasificación binaria. Cada capa cuenta con una función de pooling. El número de imágenes para el entrenamiento fue de 22500 con un tamaño de 80 x 80 píxeles en blanco y negro, para la validación que se realiza cada fin de época se utilizó el 10 % de las imágenes.

La figura muestra el valor de la exactitud y perdida durante el entrenamiento de la RN entrenada por GPU y CPU respectivamente. Solo queda mencionar que la mayoría de los API's dedicados a AP brindan la opción de un entrenamiento por GPU si este cumple los requisitos del sistema.

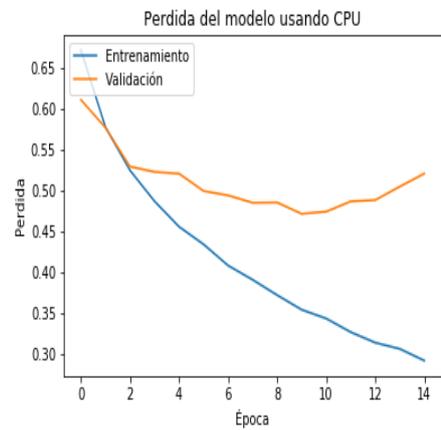
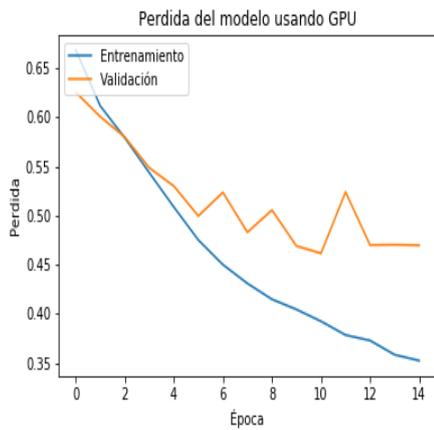
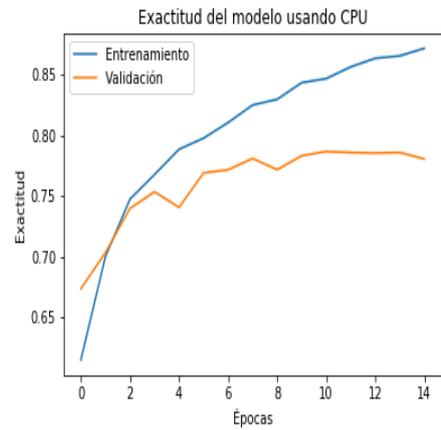
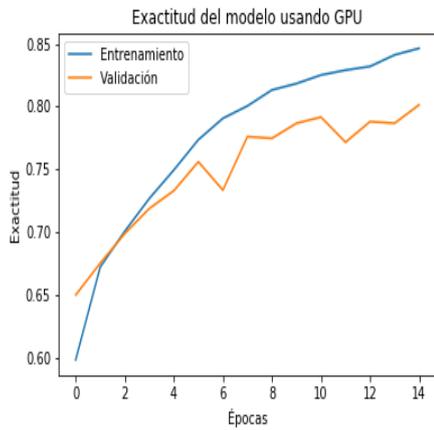


Figura 6.14: Resultados de la clasificación de perros y gatos mediante GPU y CPU respectivamente.

Lo antes mencionado son las características principales que se sugiere tener en cuenta para el entrenamiento. Existen más métodos para evitar el sobre entrenamiento, pero son más específicos y algunos se derivan de los ya antes mencionados.

6.5. Procesamiento de los datos

En la mayoría de los casos si no es que, en todos, se debe de realizar un procesamiento de los datos. Por lo general esto consta de un acondicionamiento, limpieza y verificación de datos que no se deseen tener para entrenamiento. Por lo general se selecciona y se dividen los datos que se utilizaran en el entrenamiento, validación y posteriormente en las pruebas.

Este es un punto crítico en la realización del entrenamiento y se llegó a considerar que el preprocesamiento de cualquier base de datos es el comienzo para tener un satisfactorio entrenamiento y en ocasiones suele arreglar problemas como el sobre entrenamiento.

Al trabajar con imágenes se pretende tener una imagen limpia y sin ruido. Incluso en ocasiones el redimensionamiento de las imágenes es indispensable ya que por lo general al trabajar con técnicas como AP el ahorrar memoria suele ser un problema de suma importancia. Para esto es necesario el preprocesamiento, aplicar algún tipo de filtro que mejore las condiciones generales de las imágenes.

6.5.1. Aumento de brillo y contraste

Este proceso consta de incrementar los valores en los canales RGB. El aumento de brillo se lleva a cabo a partir de la matriz de luminancia, que es igual a:

$$\frac{r'(x, y)}{r(x, y)} = \frac{g'(x, y)}{g(x, y)} = \frac{b'(x, y)}{b(x, y)} = G(x, y) \quad (6.7)$$

Donde $r'(x, y)$, $g'(x, y)$, $b'(x, y)$ se refieren al pixel con mayor valor de la matriz RGB. Para obtener la matriz de luminancia es necesario transformar la matriz RGB a un espacio de colores HSV, donde V corresponde al canal de luminosidad, H al matiz y S a la saturación. La transformación se realiza, mediante la función en OpenCV `CV_BGR2HSV`. De esta manera se obtuvo el canal $V(x, y)$, ahora con la ecuación 6.7 se obtiene la matriz de ganancia de luminancia.

$$G(x, y) = \frac{V'(x, y)}{V(x, y)} = \frac{V'(x, y)}{\max(r(x, y), g(x, y), b(x, y))} \quad (6.8)$$

Donde $V'(x, y)$ es la intensidad de luminosidad del pixel en la posición (x, y) . Una vez obtenida la matriz de luminancia se aplica la ecuación 6.4 para obtener la curva de transición. Donde $U \in [0, 1]$ que denota la normalización del pixel en el canal de luminosidad y Y es una constante típicamente mayor a 1, donde se puede variar para obtener un factor más alto.

Con este proceso se logra el aumento de brillo de una imagen. No es

la única manera de lograr un aumento de brillo, existen más formas de poder realizar este proceso. pero para fines de este trabajo se optó por utilizar este pequeño algoritmo.

En ocasiones no solo es necesario con aumentar el brillo en las imágenes para lograr obtener una buena resolución. Para esto se necesitan aplicar otros filtros que en este caso se tomó la decisión de utilizar un aumento de contraste de acuerdo a las librerías que incorpora OpenCV

6.5.2. Software especializado en aprendizaje profundo

El gran éxito que han tenido las técnicas de IA ha dado paso a la creación de herramientas que faciliten el desarrollo e implementación de los algoritmos en los que se basan las IA. Algunas herramientas de software se especializan en determinadas técnicas, por lo tanto, no existe una opción definitiva al momento de elegir un auxiliar en el desarrollo de modelos basados en IA. Incluso si dos frameworks se especializan en la misma técnica, pueden brindar ventajas diferentes ya que lo pueden abordar de manera, y en ocasiones puede resultar mejor una que otra todo dependerá del problema a resolver.

Algunas de las herramientas más utilizadas se muestran en la tabla 6.5 donde se describe brevemente algunas ventajas, así como las compañías más relevantes que hacen uso de sus servicios.

Tabla 6.5: Software especializado en AP

Nombre	Características	Clientes
Tensorflow	<ul style="list-style-type: none"> - Creación fácil de modelos. - Creación de modelos robustos en la nube. - Herramientas diseñadas para la investigación. - Una gran cantidad de información disponible para la creación de modelos. 	<p>Airbnb Coco-Cola DeepMind Twitter Intel</p>
Caffe	<ul style="list-style-type: none"> - Diseño de software de alto nivel - Demuestra una gran velocidad para realizar experimentos - Es una herramienta creada por académicos para académicos. - La mayoría de sus tutoriales están disponibles de manera gratuita 	<p>A9 Amazon NVIDIA</p>
MXNET	<ul style="list-style-type: none"> - Interfaz híbrida entre un modo imperativo y simbólico. - Brinda un entrenamiento distribuido entre diferentes dispositivos. - Puede incorporarse a 8 diferentes lenguajes de programación. 	<p>Pioneer Amazon Acer Wolfram</p>
PyTorch	<ul style="list-style-type: none"> - Facilita la escritura creando un sistema fácil y rápido de leer. - El ecosistema cuenta con un gran catálogo de librerías de apoyo. - La mayoría de plataformas en la nube cuentan con soporte para PyTorch 	<p>Udacity Stanford University</p>
NVIDIA TensorRT	<ul style="list-style-type: none"> Una gran optimización de memoria GPU. Minimiza el uso de memoria dinámica Brinda un diseño escalable múltiple 	<p>BOSH Continental Ford DarwinAI IBM</p>

Hipotesis

Es posible realizar una arquitectura de red neuronal profunda para detectar microaneurismas y exudados los cuales se encuentran en la parte posterior del ojo, que permita detectar factores de enfermedades oculares.

Objetivos

Objetivo general

Detectar patrones típicos causados por la retinopatía diabética u otras enfermedades oculares, que ayude a realizar un diagnóstico temprano. (Anterior).

Objetivos particulares.

- Realizar un informe con lo encontrado por el modelo, que este informe sea fácil de entender.
- Probar que el modelo detecta patrones o anomalías causadas por la retinopatía diabética.
- Procesar las imágenes para así encontrar una configuración adecuada que ayude al entrenamiento del modelo.

Metodología

En el presente trabajo se utilizó una base de datos que se obtuvo de Kaggle una página dedicada a la elaboración de concursos donde se pueden descargar diversas bases de datos de todo tipo. El conjunto de imágenes se llama “Diabetic Retinopathy Detection”, dicha base de datos contiene alrededor de 70,000 imágenes, de las cuales 35,000 están etiquetadas para entrenamiento. (Kaggle, 2015) Las restantes imágenes no tienen clasificación y son solo para predicción. Cada imagen tiene una dimensión diferente, teniendo resoluciones de (1080 x 720) hasta (320 x 240). Haciendo que algunas imágenes sean casi perfectas para clasificación y en cambio otras se podrían considerar como ruido. La cantidad de imágenes en cada categoría son las siguientes.

- RD Ausente 25,000 imágenes.
- RD Leve 2,000 imágenes.
- RD Moderada 5,000 imágenes.
- RD Severa No Proliferativa 807 imágenes.
- RD Severa Proliferativa 708 imágenes.

Las imágenes que se tienen en la base de datos de kaggle están centradas en la macula. También cuentan con una característica que ayuda a crear

un sistema más robusto el cual es que las imágenes no fueron tomadas con la misma cámara, esto se refiere a que no tienen el mismo grado de amplitud.

La RD proliferativa es la etapa más violenta de la enfermedad, en este punto se requiere de una intervención quirúrgica en casi todos los casos. Es de suma importancia la atención de esta enfermedad en su grado proliferativo, de no ser así se corre el riesgo de perder parcial o en su totalidad la visión.

Debido a todo lo presentado en este trabajo se decidió dividir la retinopatía en dos categorías. El primer conjunto consta de RD ausente, leve, moderada y severa no proliferativa. El segundo conjunto son todas aquellas imágenes con RD proliferativa. Esto crea un desbalance en los datos muy amplio, ya que se tendrían 2807 imágenes para la primera categoría y 708 para la segunda. Por eso se tomó la decisión de balancear los datos reduciendo el número de imágenes de la primera categoría a 4000 imágenes y dejando la segunda categoría con su total de imágenes de RD proliferativa. La fig. 9.1 muestra el antes y después del balanceo de los datos.

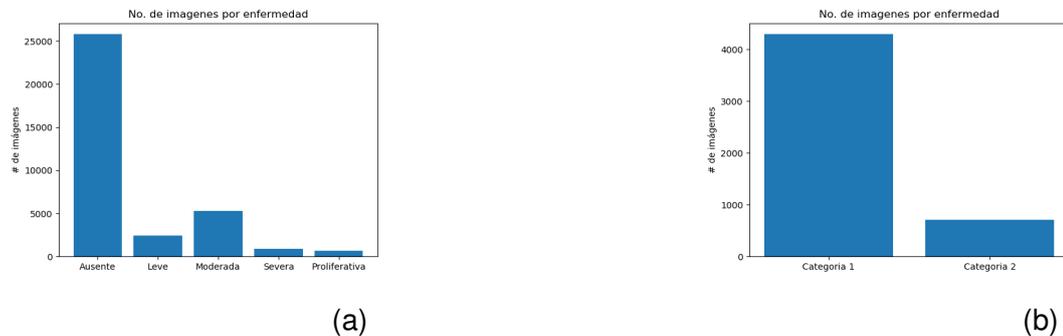


Figura 9.1: La gráfica del lado izquierdo muestra el total de imágenes en sus cinco categorías, la gráfica de la derecha representa el número de imágenes en dos categorías tomando en cuenta el desbalance de los datos.

Se desarrollaron dos CNN para tratar la clasificación de las imágenes. La primera red consta de una variante de red lineal y la denominamos CNN-1, la

cual consta de capas de neuronas una sobre otra. Lo cual implica que una capa tras otra está anidada. Para evitar sobre entrenamiento se implementaron regularizaciones y batch normalization entre algunas capas, así como un dropout. Una representación gráfica de la red se puede observar en la siguiente figura.

En la segunda arquitectura se utilizó una CNN de estilo residual y se denominó CNN-2. Este tipo de redes convergen más rápido que las redes lineales, debido a que trata el problema de degradación que presentan las redes tradicionales. (Krizhevsky et al., 2017) El bloque de aprendizaje residual se puede observar en la fig. 9.3.

La topología de la CNN-2 utilizada en este proyecto se puede observar en la figura siguiente. Cada bloque de inyección A es el mismo. Al igual que los bloques B y C, los cuales se pueden observar más a fondo en las figuras.

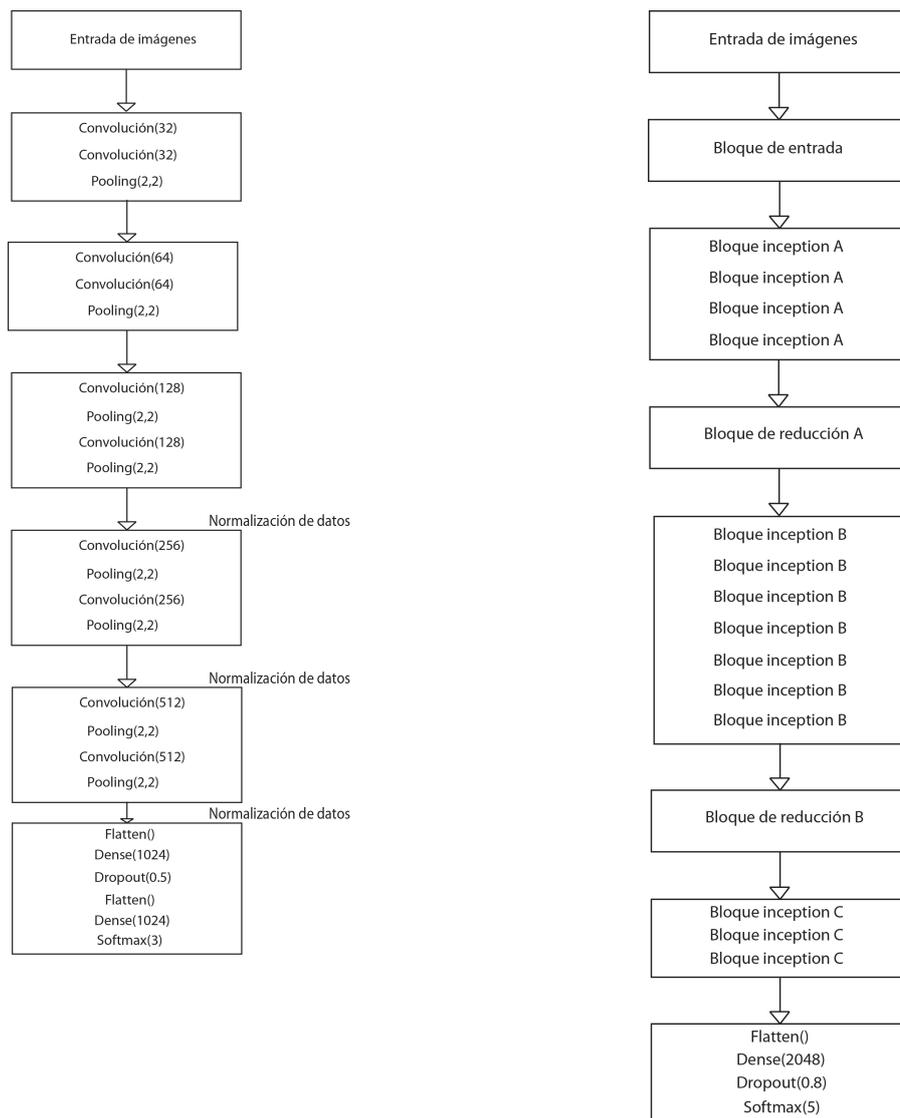


Figura 9.2: La primera imagen muestra la arquitectura de la CNN-1. Mientras que la imagen del lado derecho da lugar a la CNN-2.

9.0.1. Procesamiento de imágenes

Las imágenes medicas por lo regular suelen tener ruido o no ser de la mejor calidad, por ende un procesamiento de estas mismas debería ser requerido para mejorar la calidad o si es su caso aumentar el numero de imágenes para un

futuro entrenamiento. A la base de datos utilizada en este trabajo se le aplico un tratamiento antes de su entrenamiento. La base de datos tienen imágenes muy borrosas y algunas cuentan con poca luz. Por lo tanto se opto por aumentar tanto el brillo y el contraste en sus tres canales.

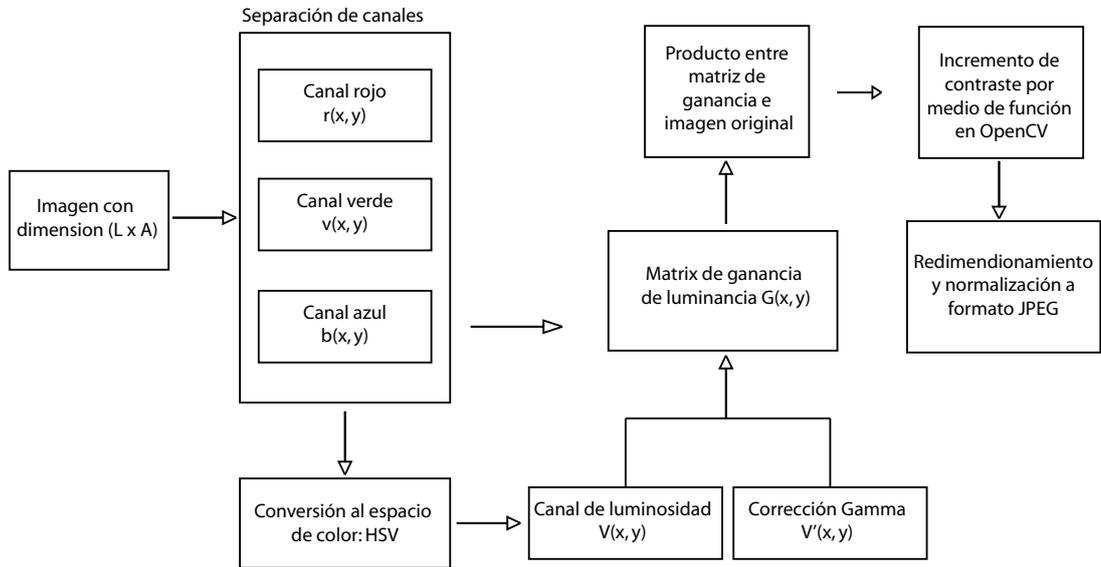


Figura 9.3: Metodología de tratamiento de imágenes.

Una vez aumentado el brillo y el contraste de cada imagen de la base de datos se elimina el exceso de fondo negro y se redimensiona la imagen de acuerdo al tamaño que se utiliza. De esta forma se aprovecha al máximo la dimension de cada imagen. El resultado final se puede observar en la figura 9.4

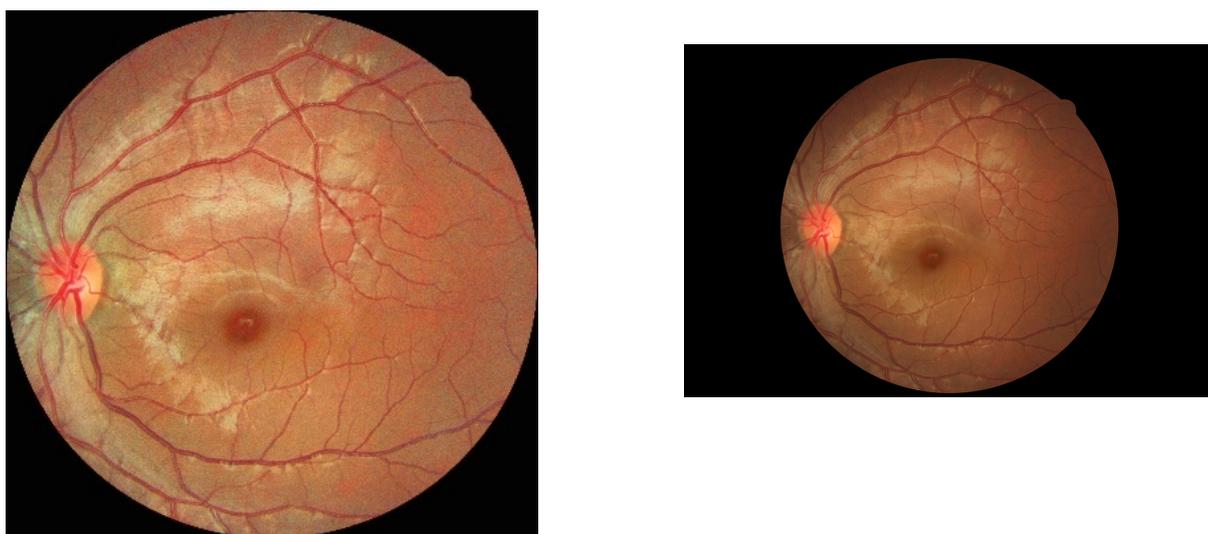


Figura 9.4: La imagen de la izquierda es el resultado final del procesamiento y se puede observar la diferencia con la original del lado derecho.

9.0.2. Entrenamiento

El utilizar bases de datos grandes favorece el uso de AP, pero si no se tiene un buen manejo de los datos de entrada no servirá de nada el uso de RP ya que, si no se alimentan de forma adecuada y con datos útiles, solo estaremos gastando tiempo y recursos innecesariamente. En la fig 9.5 se puede observar el proceso completo del trabajo realizado. Empezando desde la parte del tratamiento de los datos, hasta obtener la información final de cada clasificación.

Existen diversas formas de abordar el desarrollo de un modelo basado en redes neuronales profundas ya que en la actualidad existen diferentes frameworks o APIs especializadas en inteligencia artificial. Algunos APIs se especializan en aprendizaje automático o en otros tipos de técnicas. Este trabajo se desarrolló en TensorFlow y Keras, ambas librerías se especializan en AP. Se dice que se especializan en AP debido a que están diseñados para trabajar con ten-

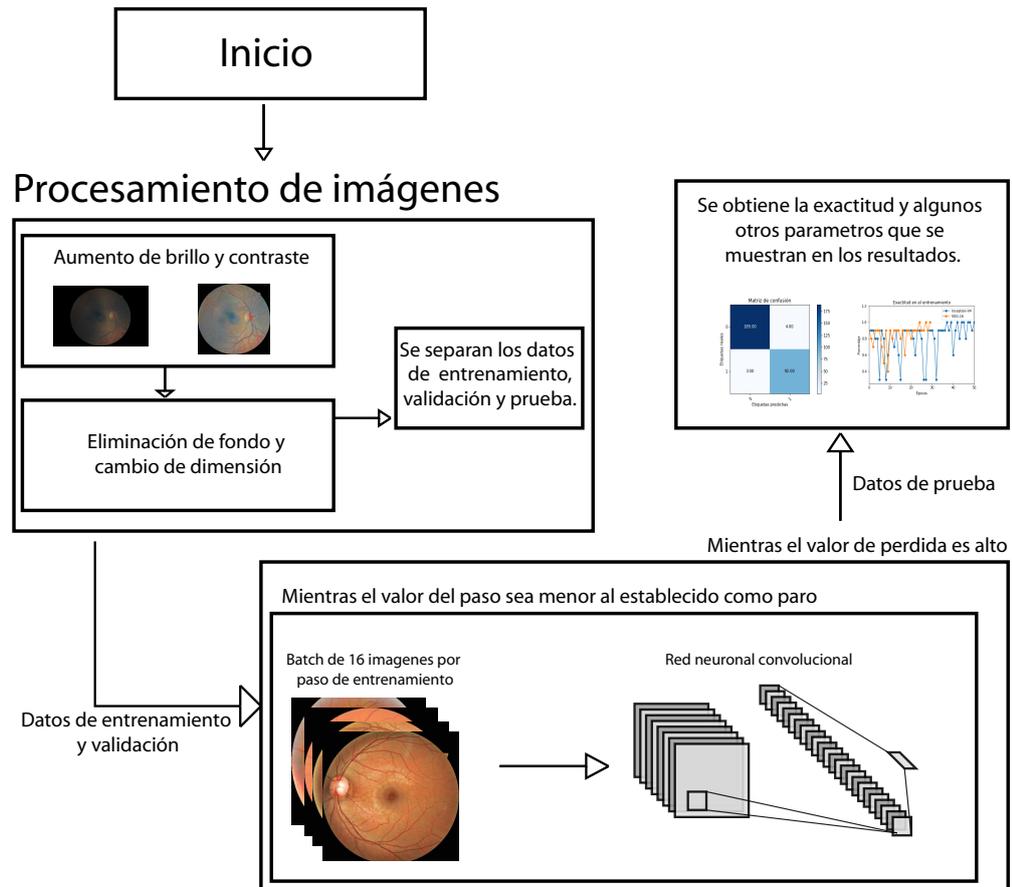


Figura 9.5: Proceso completo del trabajo realizado.

sores desde un principio, así como el cargar los datos y procesarlos. Ambos APIs se implementaron en Python como lenguaje de programación base. El hardware que se utilizó fue una tarjeta NVIDIA GTX 1060 de 6 GB de VRAM. No todas las partes de la red neuronal se cargaron a la memoria de video, las capas pequeñas se calcularon por un procesador Intel core I3 de octava generación, de esta manera se liberó carga de la tarjeta de video.

El proceso de trabajo consta de una etapa de procesamiento de las imágenes, donde se incrementa el brillo, el contraste y para finalizar se redimensionan las imágenes. Posteriormente se crean etiquetas de acuerdo a la clasificación que se requiere hacer. Por ejemplo, para la clasificación de retinopatía proliferativa se realiza un etiquetado binario donde 0 son todas aquellas imágenes donde

no se presenta la RD proliferativa y con un 1 aquellas que si lo presentan.

El siguiente paso es el entrenamiento de las RP, el cual consta de épocas. Cada época representa un entrenamiento con todas las imágenes, al final de cada época se realiza una corrida de validación en la cual se utilizan imágenes que no se encuentran en la base de datos de entrenamiento y por lo regular son el equivalente al 10 % del total de la base de entrenamiento. El criterio en el cual se basa para determinar si la CNN esta aprendiendo son los resultados de la perdida y la exactitud en los datos de validación. A continuación Se muestra el pseudocodigo de entrenamiento.

entrada: Base de datos de imágenes con tamaño (400,400,3)

salida : Valor de la exactitud y perdida

mientras *El valor del modelo no sea el deseado* **and** *no existan indicios de sobre entrenamiento* **hacer**

para $i \leftarrow 0$ **a** *Numero de pasos* **hacer**

 Entrenamiento con entrada de tensor (No. de imágenes por paso, 400, 400, 3);

fin

 Prueba de la CNN con datos de validación;

si *El valor de perdida mejoro* **entonces**

 Se guarda el valor de los pesos y el optimizado;

en otro caso

 No se guarda ningún valor;

fin

si *El valor de perdida no mejora en cinco épocas* **entonces**

 Se disminuye el learning rate

en otro caso

 Se conserva el learning rate

fin

fin

Algoritmo 1: Pseudocodigo del entrenamiento

El monitoreo de los resultados con los datos de entrenamiento y validación es de suma importancia, ya que, si la perdida y la exactitud mejora en cada época con los datos de entrenamiento, pero no en los datos de validación y esto se repite durante varias épocas quiere indicar un posible sobre entrenamiento,

lo cual quiere decir un aprendizaje en patrones que no tienen nada que ver con lo que en realidad estamos buscando. Cuando se obtiene un bajo resultado de perdida y un alto resultado de exactitud se para el entrenamiento.

Para finalizar se realiza una corrida de prueba con imágenes no utilizadas durante el entrenamiento y validación, de estos datos se obtienen los resultados mostrados en la tabla 10.2 y las matrices de confusión que se muestran en la fig.10.2

Resultados

Ya que las RN se entrenaron y estas alcanzaron buenos resultados se implementaron algunas pruebas. La prueba se realizó con 300 imágenes, donde 200 pertenecen al grupo de RD no proliferativa y los 100 restantes son RD proliferativa. Los parametros que se midieron son los siguientes.

- Exactitud: Es el porcentaje total de elementos clasificados correctamente.

$$= \frac{TP + TN}{TP + TN + FP + FN} \quad (10.1)$$

- Especificidad o ratio: Es el numero de items correctamente identificados como negativos fuera del total de negativos.

$$= \frac{TN}{TN + FP} \quad (10.2)$$

- Sensibilidad o recall: Es el numero de elementos identificados correctamente como positivos del total de positivos verdaderos.

$$= \frac{TP}{TP + FN} \quad (10.3)$$

- Precisión: Es el numero de elementos identificados correctamente como

positivo de un total de elementos identificados como positivos.

$$= \frac{TP}{TP + FP} \quad (10.4)$$

- Valor de predicción negativa: Es la probabilidad de que dada una predicción negativa, la realidad sea también negativa.

$$= \frac{TN}{TN + FN} \quad (10.5)$$

- Error de clasificación: Porcentaje de errores del modelo.

$$= \frac{FP + FN}{TP + TN + FP + FN} \quad (10.6)$$

- Prevalencia: La probabilidad de un positivo en el total de la muestra.

$$= \frac{TP + FN}{TP + TN + FP + FN} \quad (10.7)$$

Donde:

- TP = Verdaderos positivos.
- FP = Falsos positivos.
- FN = Falsos negativos.
- TN = Verdaderos negativos.

Los datos anteriormente mostrados se pueden obtener de la matriz de confusión. La cual evalúa el desempeño de un algoritmo de clasificación, ya que se puede observar cómo está clasificando dicho algoritmo, a partir de un conteo

de los aciertos y errores de cada una de las clases en la clasificación. En la tabla 10.1 se puede observar la configuración de una matriz de confusión.

Tabla 10.1: Descripción de matriz de confusión.

		CLASIFICADOR	
		+	-
Valor real	+	TP	FN
	-	FP	TN

En la Fig. 10.1 se muestra la exactitud durante el entrenamiento de ambas CNN, es importante recalcar que el tiempo de entrenamiento fue casi el mismo, pero que el número de épocas que transcurrieron fue considerablemente diferente, esto demuestra que la topología de las CNN influye en el tiempo de entrenamiento, así como en la exactitud de cada red.

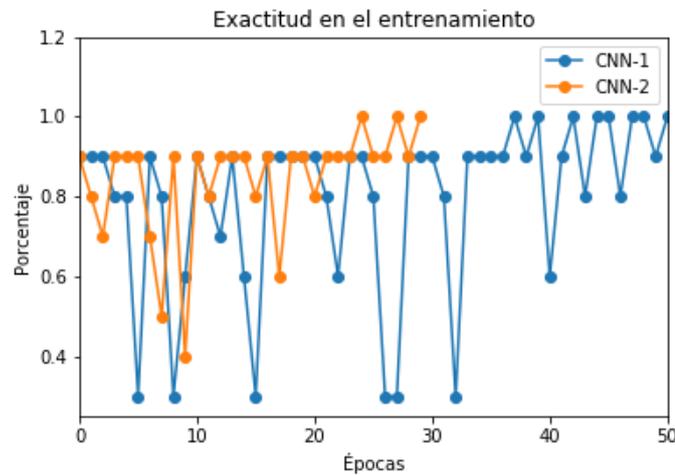


Figura 10.1: La gráfica muestra la exactitud durante el entrenamiento de ambas redes, este valor se obtuvo con los datos de validación.

En la Fig. 10.2 se muestra la matriz de confusión (Guo et al., 2016) de ambas CNNs para los datos de prueba, los cuales sus resultados se muestran en la tabla 10.2. Se puede observar que la red residual no presenta falsos negativos dando una sensibilidad del 100%, en cambio la especificidad del 92.3% esto nos indica que esta CNN es susceptible a confundir algunos patrones de RD

Tabla 10.2: Resultados obtenidos con los datos de prueba.

	Porcentaje red lineal	Porcentaje red residual
Exactitud	93.6 %	91.3 %
Especificidad	96.8 %	100 %
Sensibilidad	97.9 %	92.3 %
Precisión positiva	98.4 %	100 %
Precisión negativa	95.8 %	97.27 %
Error de clasificación	2.3 %	4.6 %
Prevalencia	64.3 %	64 %

leve, moderada o grave con patrones asociados a RD proliferativa. En el caso de la red lineal la sensibilidad y especificidad están más balanceadas y esto se ve reflejado con las 7 imágenes mal clasificadas, 3 como falsas negativas y 4 como falsos positivos. Esto claramente nos arroja una mejor exactitud y nos asegura una clasificación con menos errores.

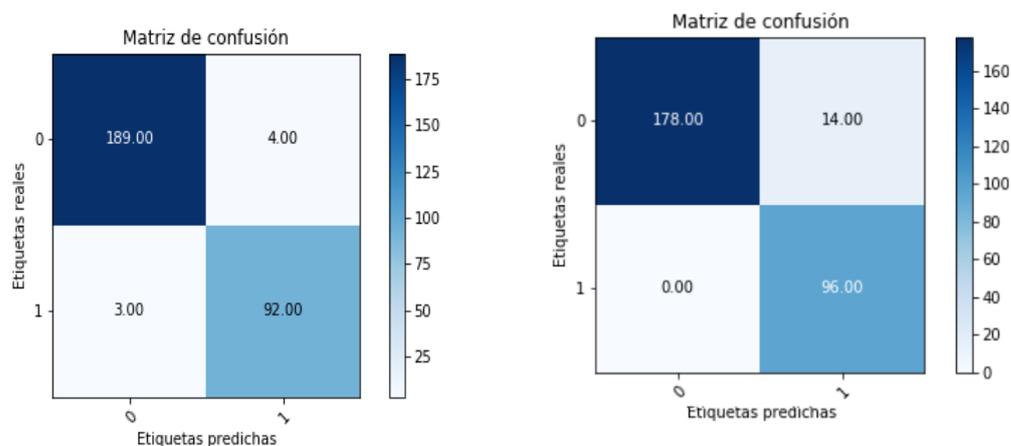


Figura 10.2: Matrices de confusión para la CNN-1 y la CNN-2 respectivamente.

Conclusiones

Se puede observar que el utilizar diferentes topologías de redes neuronales convolutivas puede llegar a dar diferentes resultados para un mismo problema, tanto en el tiempo de entrenamiento como en la exactitud, esto reafirma el tener que probar diferentes arquitecturas para así determinar la que mejor desempeño tenga para el problema. Se mostró que el aprendizaje de patrones notables en las imágenes de fondo de ojo puede llegar a alcanzar una exactitud alta. Esto quiere decir que los patrones asociados con RD Proliferativa son detectables para las CNNs.

El preprocesamiento de los datos es un papel importante en el uso de estas redes neuronales profundas ya que los datos requieren de limpieza e incluso mejoras antes de entrar a los modelos antes presentados. Cabe mencionar que, para mejorar la exactitud y el entrenamiento de los modelos se debe de tener una base de datos con más ejemplos de cada categoría, esto con el fin de no crear un desbalance significativo y así evitar el sobre entrenamiento. El hardware utilizado nos limita a utilizar imágenes de poca resolución, con una tarjeta de video más potente se podrían usar imágenes de alta resolución, así como también añadir más capas internas a las CNNs, ayudaría a encontrar patrones que con la resolución utilizada son muy difíciles de observar. Para concluir solo resta mencionar que el uso de aprendizaje profundo es una alternativa muy llamativa en

trabajos donde se cuente con gran cantidad de datos, ya que como se mostro no es necesario el señalar las características que se buscan para cada clasificación.

11.1. Trabajo futuro

Recientemente se realizó un convenio entre el Instituto Mexicano de Oftalmología (IMO) y la Universidad Autónoma de Querétaro para la colaboración y asesoramiento en proyectos donde se requieran especialistas del área. De entre los muchos servicios que ofrece el IMO, una parte importante en la institución es el diagnóstico de RD por lo que cuentan con un gran número de imágenes de fondo de ojo ya etiquetadas. Además, que las imágenes son centradas a la macula y otras al nervio óptico esto resulta en un beneficio a la hora de entrenar el clasificador ya que se puede obtener un sistema más robusto.

Trabajar con imágenes de mayor tamaño ayudaría a tener más información en las imágenes que muy posiblemente se pierdan cuando se redimensionan las imágenes a un menor tamaño. Por ende, el obtener un hardware con mayor capacidad de cálculo ayudaría a entrenar las CNN con imágenes de mayor calidad.

Además, que para brindar una clasificación ideal a este problema se debe de clasificar por las cinco categorías, esto implica un gran número de ejemplos para cada grado de RD, pero gracias al convenio con el IMO se podría llegar a conseguir un numero amplio de imágenes para el entrenamiento.

El lograr una clasificación escocesa totalmente automatizada brindaría grandes oportunidades para una comercialización de un producto que en un futuro pueda ser accesible para todas las personas. La investigación que se requiere es grande pero las recompensas son mucho mayores.

Referencias

- Catania, L. J. & Nicolitz, E. (2018). Artificial Intelligence and Its Applications in Vision and Eye Care. *Advances in Ophthalmology and Optometry*, 3(1), 21–38.
- Chitrakar, R. & Chuanhe, H. (2012). Anomaly detection using Support Vector Machine classification with k-Medoids clustering. *Asian Himalayas International Conference on Internet*, 1–5.
- Dabeer, S., Khan, M. M., & Islam, S. (2019). Cancer diagnosis in histopathological image: CNN based approach. *Informatics in Medicine Unlocked*, 16(August), 100231.
- De Oliveira Martins, L., Braz Junior, G., Correa Silva, A., Cardoso de Paiva, A., & Gattass, M. (2009). Detection of Masses in Digital Mammograms using K-Means and Support Vector Machine. *ELCVIA Electronic Letters on Computer Vision and Image Analysis*, 8(2), 39.
- Dong, J., Du, H. S., Wang, S., Chen, K., & Deng, X. (2004). A framework of Web-based decision support systems for portfolio selection with OLAP and PVM. *Decision Support Systems*, 37(3), 367–376.
- Du, C. J. & Cheng, Q. (2014). *Computer vision*.
- García-Aguirre, G., Henaine-Berra, A., Fromow-Guerra, J., Martínez-Castellanos, M. A., Salcedo-Villanueva, G., & Morales-Cantón, V. (2017). Imagen de retina de campo ultra-amplio. *Revista Mexicana de Oftalmología*, 91(6), 286–296.

- García Bello, L., Torales Salinas, J., Giménez, M. B., Flores, L. E., Gómez de Ruiz, N., & Centurión, O. A. (2016). The risk of those who care for risk: FIDRISK in healthcare personnel. *Revista Virtual de la Sociedad Paraguaya de Medicina Interna*, 3(2), 71–76.
- Gascueña, J. M., Fernández-caballero, A., & González, P. Ontologies for Student and Domain Models in Adaptive and Collaborative Learning System.
- Gollapudi, S. & Gollapudi, S. (2019). *Deep Learning for Computer Vision*.
- Guo, T., Guo, B., Zhang, J., Yu, Z., & Zhou, X. (2016). *Crowdtravel: Leveraging heterogeneous crowdsourced data for scenic spot profiling and recommendation*, volume 9917 LNCS.
- Hrbáček, J., Ripel, T., & Krejsa, J. (2010). Ackermann mobile robot chassis with independent rear wheel drives. *Proceedings of EPE-PEMC 2010 - 14th International Power Electronics and Motion Control Conference*, 46–51.
- Hsu, C. R., Chen, Y. T., & Sheu, W. H. (2015). Glycemic variability and diabetes retinopathy: A missing link. *Journal of Diabetes and its Complications*, 29(2), 302–306.
- Ingolfsson, A. & Sachs, E. (1993). Stability and Sensitivity of an EWMA Controller. *Journal of Quality Technology*, 25(4), 271–287.
- Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4, 237–285.
- Kaggle (2015). Diabetic retinopathy detection. <https://www.kaggle.com/c/diabetic-retinopathy-detection>.

- Kalisiak, M. & De Van Panne, M. (2001). A grasp-based motion planning algorithm for character animation. *Journal of Visualization and Computer Animation*, 12(3), 117–129.
- Kong, J. S., Rezaei, B. A., Sarshar, N., & Roychowdhury, V. P. (2006). E-Mail Networks. *Computer*, (August), 67–73.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Convolutional Neural Networks ImageNet Classification with Deep Convolutional Neural Network. *Communications of the Acm*, 60(06).
- Kulkarni, P. (2012). *Reinforcement and Systemic Machine Learning for Decision Making*.
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234(October 2016), 11–26.
- Mathworks (2017). Introducing Deep Learning with MATLAB. *Introducing Deep Learning with MATLAB*, 15.
- Mustafa, W. A., Abdul-Nasir, A. S., & Yazid, H. (2018). Diabetic Retinopathy (DR) on Retinal Image: A Pilot Study. *Journal of Physics: Conference Series*, 1019(1).
- Norouzi, A., Rahim, M. S. M., Altameem, A., Saba, T., Rad, A. E., Rehman, A., & Uddin, M. (2014). Medical image segmentation methods, algorithms, and applications. *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, 31(3), 199–213.

- Okamura, A. M. (2004). Methods for haptic feedback in teleoperated robot-assisted surgery. *Industrial Robot*, 31(6), 499–508.
- Patterson, J. & Gibson, A. (2008). *Deep Learning A practitioner's approach*, volume 14.
- Prado-Serrano, A., Guido-Jiménez, M. A., & Camas-Benítez, J. T. (2019). Prevalencia de retinopatía diabética en población mexicana. *Revista Mexicana de Oftalmología*, 83(5), 261–266.
- Pratt, H., Coenen, F., Broadbent, D. M., Harding, S. P., & Zheng, Y. (2016). Convolutional Neural Networks for Diabetic Retinopathy. *Procedia Computer Science*, 90(July), 200–205.
- Quellec, G., Charrière, K., Boudi, Y., Cochener, B., & Lamard, M. (2017). Deep image mining for diabetic retinopathy screening. *Medical Image Analysis*, 39, 178–193.
- Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M. P., & Ng, A. Y. (2017). CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning, 3–9.
- Ramachandran, N., Hong, S. C., Sime, M. J., & Wilson, G. A. (2018). Diabetic retinopathy screening using deep neural network. *Clinical and Experimental Ophthalmology*, 46(4), 412–416.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9351, 234–241.

- Shi, S., Wang, Q., Xu, P., & Chu, X. (2017). Benchmarking state-of-the-art deep learning software tools. *Proceedings - 2016 7th International Conference on Cloud Computing and Big Data, CCBD 2016*, 99–104.
- Srivastava, R., Duan, L., Wong, D. W., Liu, J., & Wong, T. Y. (2017). Detecting retinal microaneurysms and hemorrhages with robustness to the presence of blood vessels. *Computer Methods and Programs in Biomedicine*, 138, 83–91.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07-12-June*, 1–9.
- Tenorio, G. (2010). Retinopatía diabética ; conceptos actuales. *73*(3), 193–201.
- Welikala, R. A., Dehmeshki, J., Hoppe, A., Tah, V., Mann, S., Williamson, T. H., & Barman, S. A. (2014). Automated detection of proliferative diabetic retinopathy using a modified line operator and dual classification. *Computer Methods and Programs in Biomedicine*, 114(3), 247–261.
- Yamashita, O. M., Betoni, J. R., Guimarães, S. C., & Espinosa, M. M. (2009). Influência do glyphosate e 2,4-D sobre o desenvolvimento inicial de espécies florestais. *Scientia Forestalis/Forest Sciences*, (84), 359–366.
- Zachariah, S., Wykes, W., & Yorston, D. (2015). Grading diabetic retinopathy (DR) using the Scottish grading protocol. *Community Eye Health Journal*, 28(92), 72–73.

Anexos

CoICACI2019

IEEE COLOMBIAN CONFERENCE ON APPLICATIONS IN COMPUTATIONAL INTELLIGENCE



IEEE
Computational
Intelligence
Society



Bogotá D.C., Mayo 30 de 2019

Ingeniero
Rafael Ortiz
Facultad de Ingeniería
Universidad Autónoma de Querétaro

Para la IEEE *Colombian Conference on Applications of Computational Intelligence* (IEEE CoICACI 2019) es un placer anunciar que el trabajo titulado “*Classification of proliferative diabetic retinopathy using deep learning*” con ID 54 y autores: Rafael Ortiz, Saúl Tovar, Juan Ramos and Efrén Gorrostieta ha sido **ACEPTADO** para presentación **ORAL** en la conferencia.

Todos los trabajos enviados al IEEE CoICACI 2019 tuvieron que pasar un proceso riguroso que incluyó por lo menos dos revisores especializados en el área. Solicitamos amablemente que tenga en cuenta las revisiones realizadas y ajuste su artículo para mejorar la calidad y someterlo nuevamente para el *camera-ready*.

La versión final deberá enviarse como máximo para el 22 de mayo de 2019. De esta forma podrá ser incluida en la versión electrónica de los *proceedings* y posteriormente en la librería digital de IEEE Xplore. Recuerde que, si el artículo está en español, por lo menos el título, resumen y palabras clave deben estar en inglés, de otra forma el artículo no será publicado. También, al menos uno de los autores deberá estar registrado bajo la categoría R1 o R2 y el trabajo deberá ser presentado durante la conferencia.

Si necesita cuenta de cobro o factura, por favor comuníquese al correo colombia@ieee.org con información de la institución (dirección, número telefónico, NIT, etc).

Fechas importantes:

Camera-ready Submisión: Mayo 22 de 2019.
Registro *deadline*: Mayo 22 de 2019
Conferencia: Junio 5 al 7 de 2019

Recuerde: solo artículos presentados en la conferencia son elegibles para publicar en el IEEE Xplore.

En caso de tener más dudas, por favor contáctese a cis@ieee.org

Agradeciendo la atención prestada y esperando contar con su participación en el evento.

Cordialmente,

Alvaro David Orjuela Cañón
General Co-Chair
IEEE Colombian Conference on Applications in Computational Intelligence – IEEE CoICACI 2019
dorjuela@ieee.org

IEEE Sección Colombia: Calle 67 no. 6 - 60 Oficina 904, Edificio Torre Ejecutiva. Tel: + 57 1 2122088
Correos Electrónico: colombia@ieee.org

Classification of proliferative diabetic retinopathy using deep learning.

Rafael Ortiz-Feregrino, Saúl Tovar-Arriag, IEEE Senior Member, Juan Ramos-Arreguin, IEEE Senior Member, Efrén Gorrostieta, IEEE Senior Member.

Facultad de Ingeniería
Universidad Autónoma de Querétaro
Querétaro, México.
rafaortizferegrino@gmail.com

Abstract— Diabetic retinopathy (DR) is a disease with different degrees of severity. The patterns that denote the presence of the disease are varied; when the disease is in low level severity it's hard to find the features that represent the presence of DR. In this paper, two convolutional neuronal networks (CNN) are proposed to classify proliferative diabetic retinopathy. For the CNNs training and evaluation the Kaggle public data base that has around 35,000 training images was used. The obtained classification accuracy using the two CNNs architectures was of 96.37% and 97.38% respectively.

Palabras clave—, diagnóstico automatizado, aprendizaje profundo, retinopatía diabética, clasificación de imágenes

I. INTRODUCCIÓN

La retina es la zona que se encuentra en la parte posterior del ojo y contiene una red de vasos sanguíneos. En este sector se pueden manifestar algunas enfermedades tales como glaucoma, problemas degenerativos maculares, retinopatía diabética, y más enfermedades [1]. La retinopatía diabética es la principal causa de pérdida visual no recuperable en los países industrializados, prevaleciendo en pacientes entre los 20 y 64 años de edad, siendo responsable del 10% de nuevos casos de ceguera cada año [2]. Así mismo, la retinopatía diabética es la principal afección que daña la retina [1][3].

Existen varias técnicas basadas en visión por computadora que utilizan imágenes de fondo de ojo para clasificar el grado de severidad en la RD [4]. Los métodos más utilizados para la detección de patrones en imágenes son técnicas de inteligencia artificial [5]–[8]. El objetivo de estos sistemas, así como el propuesto en este trabajo, es la creación de nuevas herramientas que faciliten y mejoren la calidad de vida de las personas.

La clasificación de los factores que denotan la RD es un tema que se ha intentado resolver con Aprendizaje Automático (ML). Algunas clasificaciones o búsquedas de patrones se han realizado con ML como en 2016 Srivastava *et al.* que buscaba microaneurismas y hemorragias, a lo cual alcanzo una exactitud de 97%. En 2016, Pratt *et al.* alcanzó una exactitud del 75% utilizando redes neuronales convolutivas con una sensibilidad del 95%. Otro trabajo con Aprendizaje Profundo (DL) en 2017 Quellec *et al.* alcanzó una exactitud de 95% para imágenes con RD proliferativa. También en 2017 Litjens *et al.* demostró una exactitud de 57.6% clasificando con redes neuronales convolutivas. En 2018 Ramachandran *et al.* utilizando redes

neuronales profundas alcanzo un intervalo de confianza de 80.7% - 99.5% en la detección de factores relacionados con la RD. Utilizando múltiples redes neuronales profundas, en 2018 Wang *et al.* alcanzo 95.68% de exactitud. Uno de los artículos más recientes en 2018 por De Fauw *et al.* detectó múltiples enfermedades oculares entre ellas la RD. La exactitud promedio de la arquitectura es de 96% esto incluye todas las enfermedades oculares que se intentan diagnosticar en el documento. La tabla I muestra un resumen de los trabajos mencionados anteriormente en la detección de RD. Cabe mencionar que cada trabajo crea categorías diferentes para la clasificación, con excepción del trabajo realizado por Pratt *et al.* Que se basa en la clasificación escocesa

TABLA I. ESTADO DEL ARTE.

Autor	Técnica	Exactitud	Búsqueda
Srivastava (2016) <i>et al.</i> [9]	Múltiples kernels de aprendizaje	97%	Microaneurismas y hemorragias
Pratt (2016) <i>et al.</i> [10]	Redes neuronales profundas	75%	Clasificación de RD
Litjens (2017) <i>et al.</i> [11]	Aprendizaje profundo	57.60%	Clasificación de RD
Quellec (2017) <i>et al.</i> [12]	Redes neuronales profundas	95.40%	Clasificación de RD
De Fauw (2018) <i>et al.</i> [13]	Aprendizaje profundo	96%	Clasificación de RD
Ramachandran (2018) <i>et al.</i> [14]	Redes neuronales profundas	80.7% - 99.5%	Clasificación de RD

En el desarrollo del presente trabajo se utilizó DL [6], [15], una técnica de ML [16] también conocido como redes neuronales profundas, es una técnica que cae en la categoría de aprendizaje supervisado. Existen diferentes tipos de redes neuronales profundas, pero en este trabajo hace uso de redes neuronales convolutivas (CNN) debido a su buen desempeño en el procesamiento de imágenes.

Como se muestra en la tabla I se han implementado métodos de inteligencia artificial para la clasificación de diferentes características que denotan la RD, algunos trabajos se realizaron con ML y otros con DP, La tabla II, denota las características más significativas entre estos dos métodos.

TABLA II. DIFERENCIAS NOTABLES ENTRE APRENDIZAJE AUTOMÁTICO Y APRENDIZAJE PROFUNDO [17].

<i>Aprendizaje automático</i>	<i>Aprendizaje profundo</i>
+ Buenos resultados con pocos datos	- Requieren grandes cantidades de datos
+ Rápido entrenamiento	- Carga computacional intensiva
- Se necesitan diferentes características y clasificadores para alcanzar buenos resultados	+ Las características y clasificadores se aprenden automáticamente
- La exactitud no mejora después de un punto	+ La exactitud no tiene un límite de mejora

Tiempo atrás era imposible pensar en crear redes neuronales con gran cantidad de parámetros entrenables para la detección de patrones en imágenes, o incluso en cualquier otra tarea que se pretendiera utilizar aprendizaje profundo debido al alto costo computacional de estas topologías. Gracias a los grandes progresos en términos de poder computacional y las grandes cantidades de información que son posibles de recolectar y clasificar en la actualidad, las redes neuronales han tenido un gran auge, pero no solo estos factores han favorecido la continua mejora del aprendizaje profundo. Las nuevas ideas, algoritmos y las diversas arquitecturas de las redes neuronales son así mismo un factor importante que de igual manera mejora el desempeño de las redes neuronales [6], [15].

En las variadas arquitecturas de redes neuronales convolutivas destacan algunas por encima de otras. No solo por su exactitud si no también debido a que algunas cuentan con gran cantidad de parámetros a entrenar, lo que no es un factor favorable, ya que ocasiona que el costo computacional sea más alto. Es necesario recalcar que algunas arquitecturas funcionan bien para algunos problemas y otras no, esto indica que se deben de probar diferentes topologías y dictaminar cual es la más indicada para el problema.

II. METODOLOGIA

En el presente trabajo se utilizó la base de datos de Kaggle llamada "Diabetic Retinopathy Detection", dicha base de datos consta con alrededor de 70,000 imágenes, de las cuales 35,000 están etiquetadas para entrenamiento [18]. Cada imagen tiene una dimensión diferente, teniendo resoluciones de (1080 x 720) hasta (320 x 240). Las categorías que emplea la base de datos son:

- 1) RD Ausente: No hay anomalías
- 2) RD Leve: Microaneurismas
- 3) RD Moderada: Hemorragias y microaneurismas
- 4) RD Severa No Proliferativa: Hemorragias intrarretinianas, cuentas venosas definidos, anomalías microvasculares intrarretiniales, exudados.
- 5) RD Severa Proliferativa: Neovascularización, Hemorragia vítrea/pre-retiniana.

En la Fig. 1 [18] se puede observar la gravedad de la RD según los parámetros de clasificación escocesa [19].

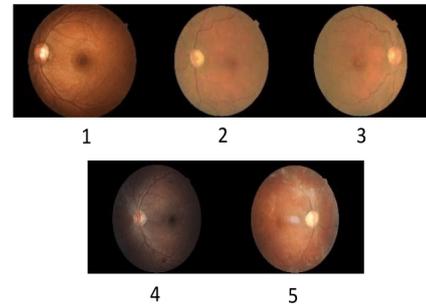


Fig. 1. Ejemplo de severidad en la RD de acuerdo con la clasificación escocesa, 1) Ausente, 2) Leve, 3) Moderada, 4) Severa, 5) Proliferativa. [18]

No obstante, el utilizar una base de datos publica trae consigo muchos problemas, ya que este tipo de bases de datos cuentan con muchas imágenes que son casi imposibles de clasificar, además que la mayoría de las imágenes caen en la categoría de RD ausente, esto crea un sesgo importante.

A. Aprendizaje profundo (DL)

En este trabajo se utilizaron dos arquitecturas diferentes. La CNN-1 está inspirada en las redes residuales e Inception-V4 [6]. No hay trabajos en los cuales se utilicen este tipo de topologías residuales en la clasificación de RD. La CNN-2 está basada en VGG-16, esta topología es la más usada en trabajos relacionados con clasificación de RD [20]. La tabla III muestra el número de parámetros de cada CNN.

TABLA III. NUMERO DE PARÁMETROS DE CADA CNN.

CNN	Parámetros de cada red neuronal convolutiva		
	Parámetros	Parámetros para entrenar	Parámetros para no entrenar
CNN-1	44,271,235	44,263,043	8,192
CNN-2	8,872,355	8,871,331	1,024

Ambas CNNs se utilizaron para clasificar la RD proliferativa. Las clases se agruparon en dos categorías, la primera consta de los grados de RD Ausente, Leve, Moderada, Severa. La segunda categoría se comprende de RD Proliferativa.

La tabla IV. Muestra el número total de imágenes que corresponde a cada categoría, pero no todas las imágenes fueron utilizadas, ya que se espera balancear los datos, por esto mismo se utilizó solo una cantidad de imágenes determinada.

TABLA IV. NUMERO DE IMÁGENES UTILIZADAS.

Categorías	Numero de imágenes	Numero de imágenes utilizadas
Cat 1	34,418	4,293
Cat 2	708	708

No hay un sistema o configuración que nos garantice un buen desempeño de alguna CNN, en la mayoría de las ocasiones solo el probar y modificar los hiperparámetros es la única manera de ajustar la red para obtener los resultados esperados [21], [22]. Existen casos en los que se puede dar una idea de por dónde comenzar, sin embargo, la mayor parte del trabajo consiste en una exploración estocástica.

B. Procedimiento

La Fig. 2 muestra la arquitectura del sistema completo. En un principio, puede ser de ayuda el mejorar los datos de entrada, en nuestro caso las imágenes. La base de datos utilizada cuenta con imágenes de diferentes tamaños e intensidades de color, además de contener imágenes las cuales son casi imposibles clasificar, ya que cuentan con destellos de luz, reflejos del dispositivo con el que se capturaron las imágenes y también resulta el caso de imágenes que son casi totalmente oscuras.

El siguiente paso consiste en la entrada de los datos a la CNN. Cada red neuronal se entrenó por separado debido al gran costo computacional que esto requiere, así que al término de un entrenamiento si los resultados son satisfactorios, se prosigue a entrenar la siguiente red, de lo contrario se modifican los hiperparámetros y se ejecuta el entrenamiento nuevamente.

En el último paso, se guardan los valores que corresponden a cada peso de ambas CNNs, con esto se puede seguir entrenando la red neuronal con nuevas imágenes o realizar validaciones con los pesos ya guardados.



Fig. 2. Metodología del trabajo.

Las imágenes de retina suelen tener muchas variaciones en la calidad de la imagen, llegando a tener un contraste muy bajo de colores, ruido, e incluso reflejo del lente que toma la imagen [4]. Mejorar las imágenes con algún filtro o cambiando los canales de color puede llegar a mejorar el entrenamiento, aunque también dejar las imágenes con ruido podría ayudar a crear un sistema más robusto. Lo ideal es llegar a obtener un punto medio, donde se mejore el entrenamiento y a su vez se permita a

la CNN analizar imágenes con ruido y así crear un sistema lo suficientemente robusto.

El preprocesamiento se realiza de la siguiente manera: Se recortó la parte de la retina y se redimensionó su tamaño original a un tamaño estándar, ya que las imágenes cuentan con tamaños diferentes. Para identificar la retina se segmenta toda la imagen, de la cual el objeto segmentado con un área más grande se identifica como la retina. Posteriormente se recorta el área del objeto segmentado más grande y se redimensiona a un tamaño estándar de 400 x 400 píxeles, dejando la imagen en los canales RGB. La Fig. 3 muestra la imagen original y la redimensionada respectivamente.

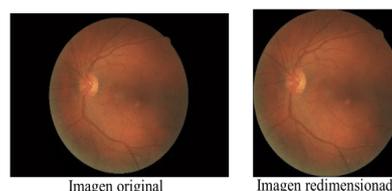


Fig. 3. Ejemplo de redimensión, recortando el contorno de la retina, quitando excesos de fondo negro que no se necesitan.

El redimensionamiento de las imágenes ayuda a que la carga computacional no sea tan alta, pero como consecuencia se pierde información, esto puede afectar el desempeño de las CNNs, lo cual afecta con un impacto más alto a las imágenes que caen en la categoría de RD leve y moderada, ya que los patrones suelen ser muy difíciles de observar con un solo vistazo.

C. Arquitectura CNN-2 y CNN-1

La CNN-2 utiliza 10 capas convolutivas, de forma lineal una de tras de otra. Se utilizó una regularización de Kernel en las últimas cuatro capas de la red, ya que son las capas más grandes de la CNN y se debe tener cuidado del sobre entrenamiento. Se ha demostrado que el aplicar estos métodos ayuda a la prevención de dicho sobre entrenamiento [23]. Además, se incorporó una normalización de los datos en la salida de estas mismas cuatro capas, ya que el mantener los datos normalizados ayuda a un mejor entrenamiento [24]. En la Fig. 4 Se puede visualizar la arquitectura antes mencionada.

A pesar de que se utilizaron técnicas para tratar de evitar el sobre entrenamiento, el desbalance de los datos tiende a causar un gran sesgo a la hora del entrenamiento [25]. Para ayudar un poco más a evitar el sobre entrenamiento se incorporó un peso diferente para cada grupo de imágenes ya que la relación de la categoría 0 con respecto a la otra categoría es de 4:1 aproximadamente. Esto quiere decir que al clasificar una imagen de la categoría 0 de manera correcta, no tendrá tanto impacto como si se clasificara una imagen del grupo 1.

Las redes neuronales que utilizan una arquitectura de tipo residual suelen tener un mejor desempeño a comparación de las que cuentan con capas convolutivas consecutivas [26]. En el caso de la CNN-1 se modificó el tamaño de la conexión total en la parte de detección y se aumentó el "Dropout" para no caer en un sobre entrenamiento. Además, se agregaron kernels de

regularización en la mayoría de sus capas convolutivas. En los bloques de reducción A y B se realiza una normalización de los datos, así como en el último bloque C. La arquitectura se muestra en la Fig. 4. De la misma manera que la red anterior se aplicó un peso diferente para la clasificación.

D. Entrenamiento

Se usaron 4,500 imágenes de entrenamiento y 500 imágenes de validación para cada época, siendo un total de 5,000 imágenes utilizadas. El entrenamiento consta de 281 pasos y la validación de 31. Cada paso tiene de entrada un lote de 16 imágenes, resultando en un tensor de tamaño (16, 400, 400, 3). Al término de todas las épocas de entrenamiento, se utilizan 300 imágenes de predicción.

El optimizador utilizado es ADAM [27]. Con una tasa de aprendizaje de 0.001 como parámetro inicial de entrenamiento. Cuando la CNN deja de mejorar por tres épocas seguidas se disminuye la tasa de aprendizaje a 0.0008, de no mejorar en tres épocas seguidas nuevamente se disminuye la tasa de aprendizaje, si ocurre el caso de una mejora se cuentan cinco épocas para regresar a la tasa de aprendizaje original. Si estas medidas no funcionan por un periodo de trece épocas, se considera que la red se está sobre entrenado y se detiene el entrenamiento. Cabe mencionar que al mejorar el modelo después de una época se guardan los valores de los pesos en esa época.

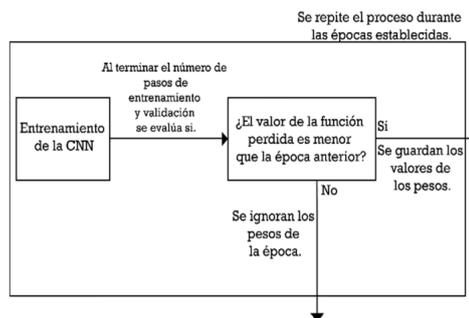


Fig. 5. Diagrama de entrenamiento.

Existen diversas formas de abordar el desarrollo de un modelo basado en redes neuronales profundas ya que en la actualidad existen diferentes frameworks o API's especializadas en redes neuronales. Este trabajo se desarrolló en TensorFlow y Keras, ambas librerías que se especializan en aprendizaje profundo, y Python como lenguaje de programación.

El hardware que se utilizó fue una tarjeta NVIDIA GTX 1060 de 6 GB de VRAM. No todas las partes de la red neuronal se cargaron a la memoria de video, las capas pequeñas se calcularon por un procesador Intel core i3 de octava generación, de esta manera se liberó carga de la tarjeta de video.

III. RESULTADOS

Se obtuvieron los resultados que se muestran en la tabla V. Donde se puede observar la exactitud, sensibilidad, especificidad y el tiempo que duro el entrenamiento. La prueba se realizó con 300 imágenes, donde 200 pertenecen al grupo de RD no proliferativa y los 100 restantes son RD proliferativa.

TABLA V. Resultados obtenidos con las imágenes de prueba.

CNN	Exactitud	Sensibilidad	Especificidad	Tiempo
CNN-1	95.13%	100%	92.7%	11h 39m
CNN-2	97.56%	97.3%	96.9%	11h 55m

En la Fig. 6 se muestra la exactitud durante el entrenamiento de ambas CNNs, es importante recalcar que el tiempo de entrenamiento fue casi el mismo, pero que el número de épocas que transcurrieron fue considerablemente diferente, esto demuestra que la topología de las CNNs influye en el tiempo de entrenamiento, así como en la exactitud de cada red.

También se puede ver que una topología lineal como la de CNN-2 con capas grandes una después de la otra, afectan en el costo computacional que se traduce en tiempo de entrenamiento. En cambio, la CNN-1 que su arquitectura está inspirada en capas residuales su entrenamiento por época es más corto, pero tarda más épocas en converger hacia una buena exactitud debido a la cantidad de parámetros que se deben entrenar.

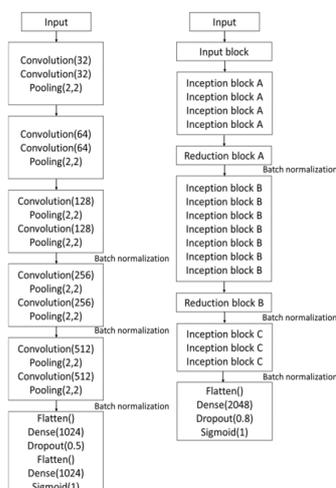


Fig. 4. Arquitectura CNN-2 y CNN-1 respectivamente.

La métrica que se evalúa para decidir si la CNN está o no mejorando, es la función de pérdida, pero además se utilizó otra métrica, la exactitud. Pero esta última no tiene un impacto a la hora de evaluar si una red neuronal está mejorando o no, solo sirve como referencia a la hora de evaluar el modelo. La Fig. 5 muestra los pasos descritos anteriormente.

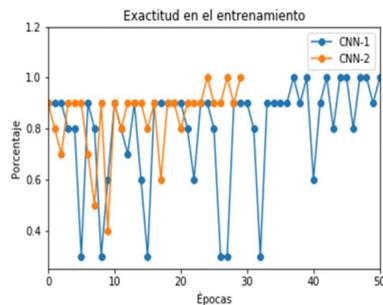


Fig. 6. Exactitud durante cada época en el entrenamiento.

En la Fig. 7 se muestra la matriz de confusión [28] de ambas CNNs para los datos de prueba, los cuales sus resultados se muestran en la tabla V. Se puede observar que la CNN-1 no presenta falsos negativos dando una sensibilidad del 100%, en cambio la especificidad del 92.7% nos indica que esta CNN es susceptible a confundir algunos patrones de RD leve, moderada o grave con patrones asociados a RD proliferativa. En el caso de la CNN-2 la sensibilidad y especificidad están más balanceadas y esto se ve reflejado con las 7 imágenes mal clasificadas, 3 como falsas negativas y 4 como falsos positivos. Esto claramente nos arroja una mejor exactitud y nos asegura una clasificación con menos errores.

Como se mostró en la tabla I, algunos trabajos alcanzan una buena exactitud en la detección de RD o características relacionadas con la enfermedad. Debido a que no todos realizan una clasificación estándar como por ejemplo la clasificación escocesa, la comparación de resultados no es adecuada. El presente trabajo muestra una buena exactitud, sensibilidad y especificidad debido a que los datos utilizados no cuentan con un desbalance muy amplio, así se asegura que esto no juegue un papel importante en la clasificación.

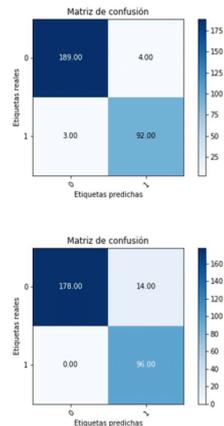


Fig. 7. Matriz de confusión de la CNN-2 y CNN-1 respectivamente.

IV. CONCLUSION

Se puede observar que el utilizar diferentes topologías de redes neuronales convolutivas puede llegar a dar diferentes resultados para un mismo problema, tanto en el tiempo de entrenamiento como en la exactitud, esto reafirma el tener que probar diferentes arquitecturas para así determinar la que mejor desempeño tenga para el problema.

Se mostró que el aprendizaje de patrones notables en las imágenes de fondo de ojo puede llegar a alcanzar una exactitud alta. Esto quiere decir que los patrones asociados con RD Proliferativa son detectables para las CNNs.

El preprocesamiento de los datos es un papel importante en el uso de estas redes neuronales profundas ya que los datos requieren de limpieza e incluso mejoras antes de entrar a los modelos antes presentados. Cabe mencionar que, para mejorar la exactitud y el entrenamiento de los modelos se debe de tener una base de datos con más ejemplos de cada categoría, esto con el fin de no crear un desbalance significativo y así evitar el sobre entrenamiento.

El hardware utilizado nos limita a utilizar imágenes de poca resolución, con una tarjeta de video más potente se podrían usar imágenes de alta resolución, así como también añadir más capas internas a las CNNs, ayudaría a encontrar patrones que con la resolución utilizada son muy difíciles de observar.

Para concluir solo resta mencionar que el uso de aprendizaje profundo es una alternativa muy llamativa en trabajos donde se cuente con gran cantidad de datos, ya que como se mostro no es necesario el señalar las características que se buscan para cada clasificación.

V. REFERENCIAS

- [1] W. A. Mustafa, A. S. Abdul-Nasir, and H. Yazid, "Diabetic Retinopathy (DR) on Retinal Image: A Pilot Study," *J. Phys. Conf. Ser.*, vol. 1019, no. 1, 2018.
- [2] G. Tenorio, "Retinopatía diabética; conceptos actuales," vol. 73, no. 3, pp. 193–201, 2010.
- [3] A. Prado-Serrano, M. A. Guido-Jiménez, and J. T. Camas-Benitez, "Prevalencia de retinopatía diabética en población mexicana," *Rev. Mex. Oftalmol.*, vol. 83, no. 5, pp. 261–266, 2009.
- [4] R. A. Welikala *et al.*, "Automated detection of proliferative diabetic retinopathy using a modified line operator and dual classification," *Comput. Methods Programs Biomed.*, vol. 114, no. 3, pp. 247–261, 2014.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Convolutional Neural Networks ImageNet Classification with Deep Convolutional Neural Network," *Commun. Acm*, vol. 60, no. 06, 2017.
- [6] C. Szegedy *et al.*, "Going deeper with convolutions," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07–12–June, pp. 1–9, 2015.
- [7] R. Ramachandran, D. C. Rajeev, S. G. Krishnan, and P. Subathra, "Deep learning – An overview," *Int. J. Appl. Eng. Res.*, vol. 10, no. 10, pp. 25433–25448, 2015.
- [8] D. Ning and L. Yafen, "Automated Identification of Diabetic Retinopathy Stages Using Support Vector

- Machine.”
- [9] R. Srivastava, L. Duan, D. W. K. Wong, J. Liu, and T. Y. Wong, “Detecting retinal microaneurysms and hemorrhages with robustness to the presence of blood vessels,” *Comput. Methods Programs Biomed.*, vol. 138, pp. 83–91, 2017.
- [10] H. Pratt, F. Coenen, D. M. Broadbent, S. P. Harding, and Y. Zheng, “Convolutional Neural Networks for Diabetic Retinopathy,” *Procedia Comput. Sci.*, vol. 90, no. July, pp. 200–205, 2016.
- [11] G. Litjens *et al.*, “A survey on deep learning in medical image analysis,” *Med. Image Anal.*, vol. 42, no. December 2012, pp. 60–88, 2017.
- [12] G. Quellec, K. Charrière, Y. Boudi, B. Cochener, and M. Lamard, “Deep image mining for diabetic retinopathy screening,” *Med. Image Anal.*, vol. 39, pp. 178–193, 2017.
- [13] J. De Fauw *et al.*, “Clinically applicable deep learning for diagnosis and referral in retinal disease,” *Nat. Med.*, vol. 24, no. 9, pp. 1342–1350, 2018.
- [14] N. Ramachandran, S. C. Hong, M. J. Sime, and G. A. Wilson, “Diabetic retinopathy screening using deep neural network,” *Clin. Exp. Ophthalmol.*, vol. 46, no. 4, pp. 412–416, 2018.
- [15] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [16] S. M. Student, C. Landran, and G. Kaur, “Review on: Detection of Diabetic Retinopathy using SVM and MDA,” *Int. J. Comput. Appl.*, vol. 117, no. 19, pp. 975–8887, 2015.
- [17] Mathworks, “Introducing Deep Learning with MATLAB,” *Introd. Deep Learn. with MATLAB*, p. 15, 2017.
- [18] Kaggle Inc., “Diabetic Retinopathy Detection | Kaggle,” *Diabetic Retinopathy Detection*, 2015. [Online]. Available: <https://www.kaggle.com/c/diabetic-retinopathy-detection>. [Accessed: 21-Mar-2019].
- [19] S. Zachariah, W. Wykes, and D. Yorston, “Grading diabetic retinopathy (DR) using the Scottish grading protocol,” *Community Eye Heal. J.*, vol. 28, no. 92, pp. 72–73, 2015.
- [20] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” pp. 1–14, 2014.
- [21] S. Albelwi, “Hyperparameter Optimization of Deep Convolutional Neural Networks Architectures for Object Recognition,” *ProQuest Diss. Theses*, no. April, p. 105, 2018.
- [22] H. Cui and J. Bai, “A new hyperparameters optimization method for convolutional neural networks,” *Pattern Recognit. Lett.*, 2019.
- [23] S. Han *et al.*, “DSD: Dense-Sparse-Dense Training for Deep Neural Networks,” 2016.
- [24] J. L. Zhou *et al.*, “Deprotonation or protonation: The coordination properties, crystal structures and spectra of cobalt (II) complex with 1-(2-pyridylazo)-2-acenaphthequinol ligand,” *J. Mol. Struct.*, vol. 1134, pp. 63–66, 2017.
- [25] Q. Dong, S. Gong, and X. Zhu, “Imbalanced Deep Learning by Minority Class Incremental Rectification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–14, 2018.
- [26] G. A. Hembury, V. V. Borovkov, J. M. Lintuluoto, and Y. Inoue, “An Acid-Base Controlled Molecular Switch. syn-anti Conformational Switching in a μ -oxo Bis(Iron Porphyrin),” *Chem. Lett.*, vol. 32, no. 5, pp. 428–429, 2003.
- [27] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” pp. 1–15, 2014.
- [28] T. Guo, B. Guo, J. Zhang, Z. Yu, and X. Zhou, *Crowdtravel: Leveraging heterogeneous crowdsourced data for scenic spot profiling and recommendation*, vol. 9917 LNCS. 2016.