FACULTAD DE INGENIERÍA

Carrera de Ingeniería Informática y de Sistemas

APLICACIÓN DE DEEP LEARNING SOBRE IMÁGENES TOPOGRÁFICAS PARA MEJORAR LA PRECISIÓN DEL DIAGNÓSTICO DE QUERATOCONO EN UNA CLINICA DE LIMA

Tesis para optar el Título Profesional de Ingeniero Informático y de Sistemas

BRYAN FABIÁN JULCA VILLARREAL
0000-0001-8873-046X

Asesor:
Mg. David Alejandro Tinoco Reyes
0000-0002-5394-7033

Lima - Perú
2020

JURADO DE LA SUSTENTACIÓN ORAL

……..…….

Presidente

……..…….

Jurado 1

……..…….

Jurado 2

Entregado el:___/___/ 2020 **Aprobado por:**

…… ……

Graduando **Asesor de Tesis**

UNIVERSIDAD SAN IGNACIO DE LOYOLA
FACULTAD DE INGENIERIA

DECLARACIÓN DE AUTORIDAD

Yo, JULCA VILLARREAL BRYAN FABIÁN, identificada con DNI N.º 72878530. Bachiller del Programa Académico de la Carrera de Ingeniería Informática y de Sistemas de la Facultad de Ingeniería de la Universidad San Ignacio de Loyola, presento mi tesis titulada: “APLICACIÓN DE DEEP LEARNING SOBRE IMÁGENES TOPOGRÁFICAS PARA MEJORAR LA PRESICIÓN DEL DIAGNÓSTICO DE QUERATOCONO EN UNA CLINICA DE LIMA”.

Declaro en honor a la verdad, que el trabajo de tesis es de mi auditoría; que los datos, los resultados y su análisis e interpretación, constituyen mi aporte. Todas las referencias han sido debidamente consultadas y reconocidas en la investigación.

En tal sentido, asumo la responsabilidad que corresponda ante cualquier falsedad u ocultamiento de la información aportada. Por todas las afirmaciones, ratifico lo expresado, a través de mi firma correspondiente.

Lima, Enero de 2020

……..

Julca Villarreal Bryan Fabián

DNI N.º 72878530

EPÍGRAFE

“El investigador sufre las decepciones, los largos meses pasados en una dirección equivocada, los fracasos. Pero los fracasos son también útiles, porque, bien analizados, pueden conducir al éxito. Y para el investigador no existe alegría comparable a la de un descubrimiento, por pequeño que sea...”

(Sir Alexander Fleming)

ÍNDICE DE CONTENIDOS

INTRODUCCIÓN **16**

PROBLEMA DE INVESTIGACIÓN **17**

IDENTIFICACIÓN DEL PROBLEMA 17

FORMULACIÓN DEL PROBLEMA 19

Problema general. *19*

Problemas específicos. *19*

MARCO REFERENCIAL **20**

ANTECEDENTES 20

Internacionales. *20*

Nacionales. *21*

ESTADO DEL ARTE 22

MARCO TEÓRICO 23

Queratocono. *23*

Inteligencia Artificial. *25*

Aprendizaje Automático. *25*

Redes Neuronal Artificiales. *27*

Aprendizaje Profundo. *30*

OBJETIVOS **37**

OBJETIVO GENERAL 37

OBJETIVOS ESPECÍFICOS 37

JUSTIFICACIÓN **38**

TEÓRICA 38

PRÁCTICA 38

SOCIAL 38

HIPÓTESIS **39**

MATRIZ DE CONSISTENCIA **40**

ALCANCES **41**

LIMITACIONES **41**

MARCO METODOLÓGICO **42**

METODOLOGÍA 42

PARADIGMA 42

ENFOQUE 42

MÉTODO 42

VARIABLES **43**

INDEPENDIENTE 43

DEPENDIENTE 43

POBLACIÓN Y MUESTRA **43**

POBLACIÓN 43

MUESTRA 43

UNIDAD DE ANÁLISIS **43**

INSTRUMENTOS Y TÉCNICAS **44**

INSTRUMENTOS 44

TÉCNICAS 44

PROCEDIMIENTOS Y MÉTODO DE ANÁLISIS **46**

PROCEDIMIENTOS 46

MÉTODO DE ANÁLISIS 51

RESULTADOS **54**

DISCUSIÓN **65**

CONCLUSIONES **66**

RECOMENDACIONES **67**

REFERENCIAS **68**

ANEXOS **72**

ÍNDICE DE FIGURAS

Figura 1. Comparación córnea normal y queratocono. Adaptado de “Queratocono”, por Clínica Rementeria 18
Figura 2. Diagram de Ishikawa de clínica Exilaser, elaboración propia 19
Figura 3. Crecimiento anual de datos de Global Datasphere. Adaptado de “Annual Size of the Global Datasphere”, por Seagate, 2018 23
Figura 4. Diseño de una red neuronal artificial con una capa oculta. Adaptado de *“Python Deep Learning: Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and TensorFlow”*, elaboración propia 27
Figura 5. Neurona biológica. Adaptado de *“Wikipedia”*, elaboración propia 28
Figura 6. Neurona Artificial. Adaptado de *“Python Deep Learning: Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and TensorFlow”*, elaboración propia 28
Figura 7. Convolución aplicada a una imagen 5x5x1 con 1 desplazamiento y un filtro de 3x3. Adaptado de *“Python Deep Learning: Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and TensorFlow”*, elaboración propia 32
Figura 8. Resultado de una capa de convolución. Adaptado de *“Python Deep Learning: Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and TensorFlow”*, elaboración propia 33
Figura 9. Agrupación máxima aplicando un filtro de 2x2. Adaptado de *“Python Deep Learning: Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and TensorFlow”*, elaboración propia 33
Figura 10. Pasos de la Metodología Fundamental para la Ciencia de Datos. Adaptado de “Metodología Fundamental para la Ciencia de Datos”, por Jhon B. Rollins, 2013 46
Figura 11. Programa para la extracción de imágenes, elaboración propia 47
Figura 12. Reportes generados por el equipo OPD Scan III, elaboración propia 47
Figura 13. Extracción de imágenes, elaboración propia 48
Figura 14. Lista generada en base a las imágenes extraídas, elaboración propia 48
Figura 15. Proceso de generación de imágenes, elaboración propia 48
Figura 16. Demostración del procesamiento de una imagen utilizando OpenCV, elaboración propia 49
Figura 17. Demostración de la generación de imágenes utilizando la técnica Aumento de Datos con la librería Keras, elaboración propia 49
Figura 18. Procesamiento de imágenes con difuminado Gaussiano, elaboración propia 51
Figura 19. Procesamiento de imágenes con detección de bordes, elaboración propia 51

Figura 20. Modificación de permisos, elaboración propia 56
Figura 21. Conexión SSH, elaboración propia 56
Figura 22. Instalación base de Amazon EC2, elaboración propia 56
Figura 23. Aplicación de tmux, elaboración propia 57
Figura 24. Acceso Jupyter Notebook de Amazon EC2, elaboración propia 57
Figura 25. Carga de archivos a Jupyter Notebook, elaboración propia 58
Figura 26. Instalación librerías adicionales en Jupyter Notebook, elaboración propia 58
Figura 27. Flujo de trabajo de los diseños propuestos, elaboración propia 59
Figura 28. Arquitectura de red neuronal convolucional del Diseño 1, elaboración propia 60
Figura 29. Procesamiento de la imagen en la red neuronal convolucional, elaboración propia 61
Figura 30.Resultados de predecir la data de prueba, elaboración propia 62

ÍNDICE DE TABLAS

Tabla 1 30
Tabla 2 31
Tabla 3 39
Tabla 4 40
Tabla 5 50
Tabla 6 52
Tabla 7 55
Tabla 8 59
Tabla 9 62
Tabla 10 63

ÍNDICE DE ANEXOS

Anexo 1: Arquitectura del Diseño 2 72
Anexo 2: Arquitectura del Diseño 3 72
Anexo 3: Arquitectura del Diseño 4 72
Anexo 4: Arquitectura del Diseño 5 73
Anexo 5: Grafica precisión y pérdida del Diseño 1 73
Anexo 6: Grafica precisión y pérdida del Diseño 2 74
Anexo 7: Grafica precisión y pérdida del Diseño 3 75
Anexo 8: Grafica precisión y pérdida del Diseño 4 76
Anexo 9: Grafica precisión y pérdida del Diseño 5 77

DEDICATORIA

Dedico esta tesis en primer lugar a Dios por guiarme e iluminarme para cumplir cada una de mis metas asimismo a mi familia, quienes nunca han dejado de apoyarme y alentarme a seguir cada uno de mis sueños, brindarme su amor incondicional y sus sabios consejos para seguir creciendo profesionalmente. Esto es posible por y para ustedes.

AGRADECIMIENTO

Agradezco a Dios por guiarme y permitirme culminar mi carrera, a mis padres por darme siempre su apoyo y por la confianza puesta en mí, gracias por enseñarme siempre cumplir mis metas con todo el esfuerzo que se uno puede emplear. A mi familia, por los alientos y el apoyo que me brindaron para seguir con este trabajo de investigación. A mí enamorada, por acompañarme durante estos años dándome su amor y apoyo en los diversos retos que me he propuesto. Así mismo a mí asesor por haberme guiado y compartido sus conocimientos durante este proyecto de investigación.

RESUMEN

En los últimos años, el uso de las tecnologías de la información y comunicación ha generado diversos beneficios en una gran variedad de industrias siendo una de ellas el sector salud, por ello en el presente trabajo de investigación se centró en realizar clasificaciones de imágenes topográficas utilizando algoritmos de Deep Learning en la clínica Exilaser, centro oftalmológico especializado en cirugía refractiva. Por esta razón las imágenes generadas por el equipo OPD - Scan III fueron recolectadas para realizar una categorizaron de ojos sanos y aquellos que padecen de queratocono.

El objetivo principal del presente trabajo de investigación es emplear herramientas tecnológicas que puedan clasificar correctamente imágenes topográficas de la enfermedad del queratocono, mediante el aprendizaje de detección de patrones que posteriormente servirán para realizar detecciones preventivas.

El algoritmo que se utilizó para realizar las clasificaciones ha sido de redes neuronales convolucionales así como un pre-procesamiento de imágenes para retirar el ruido que puede limitar el aprendizaje de la red neuronal, dando como resultado 1908 imágenes correctamente clasificadas de un total de 2073 imágenes distribuidas entre ojos normales y que padecen de queratocono, lo que equivale a un 92.04%

Palabras claves: Aprendizaje profundo, queratocono, redes neuronales convolucionales e imágenes topograficas

ABSTRACT

In recent years, the use of information and communication technologies has generated various benefits in a wide variety of industries, one of them being healthcare industry, therefore in the present research work focused on classifying topographic images using Deep Learning algorithms at the Exilaser clinic, an ophthalmological center specialized in refractive surgery. For this reason, the images generated by the OPD - Scan III equipment were collected to perform a categorization of healthy eyes and those with keratoconus.

The main objective of this research work is to use technological tools that can correctly classify topographic images of keratoconus disease by learning to detect patterns that will later be used for preventive detection.

The algorithm used to perform the classifications has been convolutional neural networks as well as image pre-processing to remove noise that can limit the learning of the neural network, resulting in 1908 correctly classified images out of a total of 2073 images distributed between normal eyes and those with keratoconus, which is equivalent to 92.04%

Keywords: Deep learning, keratoconus, convolutional neuronal network and topographic images

INTRODUCCIÓN

El sector salud se ha apoyado durante años sobre herramientas tecnológicas que permiten agilizar los procesos de análisis e inclusive aquellos manuales que generaban inconvenientes al momento de diagnosticar alguna enfermedad en las diversas ramas que existen en el sector salud.

El presente trabajo de investigación se ha centrado en realizar el diagnóstico de una enfermedad que afecta a la córnea denominada queratocono mediante el uso algoritmos de aprendizaje profundo para la clínica Exilaser. El trabajo de investigación se ha dividido en 5 capítulos, los cuales se detallan a continuación:

El primer capítulo, describe el problema de investigación y la formulación del problema, donde se describe el problema general así como los específicos que nacen a partir de la presente investigación.

En el segundo capítulo, se presenta el marco referencial, en donde se detallan los antecedentes, el estado de arte y marco teórico que sustentan el trabajo de investigación.

El tercer capítulo, se explica sobre los objetivos específicos como el general, la justificación y las hipótesis.

Posteriormente, en el cuarto capítulo, se desarrolla el marco metodológico, así como las variables asociadas al trabajo de investigación, del mismo modo se detallan los instrumentos y técnicas que se realizan a la unidad de análisis.

Por último, en el quinto capítulo se muestra el procedimiento de las imágenes topográficas en redes neuronales convolucionales también los resultados de los datos obtenidos y el análisis de éstos.

CAPÍTULO I

PROBLEMA DE INVESTIGACIÓN

Identificación del problema

Las Tecnologías de la Información y Comunicación (TIC) en el Perú se han implementado en diferentes sectores de las sociedades, con la finalidad de automatizar procesos, generar valor agregado a los usuarios internos y externos de las empresas, reducir costos y en otros casos ayudar en el replanteamiento de nuevas estrategias con la finalidad de ser eficientes. En el sector público de la salud ha generado grandes cambios con respecto al uso de las TIC, por ello se han ejecutado diversos proyectos, tales como Sistema de Registro del Certificado de Nacido Vivo en Línea, TeleMedicina, Sistema de Información en Salud (SIS), Redes Integrales de Salud, Repositorio Único de Información de Salud (REUNIS), entre otras más.

Durante el año 2019 el Ministerio de Salud (MINSA) presentó el proyecto: “Estrategia de Gobierno Digital del Sector Salud 2019 - 2022”, el cual tiene como objetivos realizar despliegues de Historias Clínicas Electrónicas, expandir TeleSalud en otras regiones del Perú y toma de decisiones clínicas en base a los datos obtenidos de los pacientes, los cuales serán explotados con tecnologías como Big Data, Machine Learning y otros. La implementación de este proyecto se ha definido bajo la Resolución Ministerial N° 1367-2018/MINSA, en donde no se especifica a que áreas se aplicará la mejora de toma de decisiones clínicas o si será utilizado para realizar diagnósticos para la detección y prevención de enfermedades.

De la misma manera el sector privado ha hecho uso de las TIC en proyectos de digitalización de procesos, como lo hizo la Clínica Internacional en el año 2018 al digitalizar las historias clínicas para tener disponible la información del paciente , de esta manera dar seguimiento y realizar diagnóstico más precisos, la Clínica SANNA en el año 2019 optó por implementar las TIC en sus procesos, orientado a las tomografías computarizadas y resonancias magnéticas que son utilizadas para realizar diagnósticos más precisos así como identificar el tratamiento adecuado para el paciente asimismo la red peruana de centro de salud AUNA emplea las TIC para el diagnóstico por imágenes, tales como ecografías, tomografías, mamografía y equipos de rayos X.

En relación al problema de investigación, el queratocono es una enfermedad que afecta a la córnea; la cual es una estructura del ojo que permite la percepción de la luz del exterior al interior del ojo, las principales funciones son de protección y refracción. El queratocono consiste en la degradación progresiva de la córnea con tendencia a volverse débil y deformarse hasta el punto de adquirir forma de un cono, las causas que generan esta enfermedad son desconocidas pero se han identificado algunos factores que pueden incrementar la deformación de la córnea, los cuales son:

Antecedentes familiares con queratocono

Frotarse los ojos constantemente

Sobre exposición a los rayos ultravioleta del sol

Como consecuencia disminuye la visión y posiblemente la aparición o aumento repentino de astigmatismo

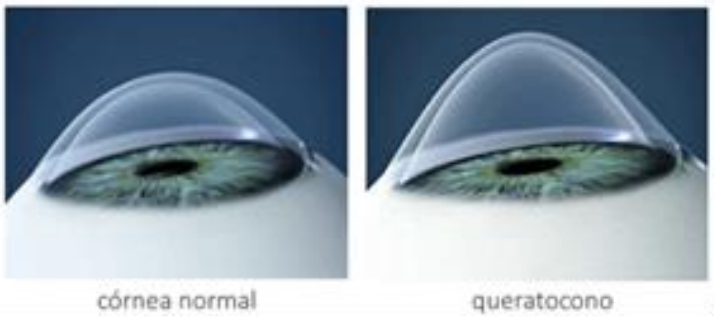


Figura 1. Comparación córnea normal y queratocono.
Adaptado de “Queratocono”, por Clínica Rementeria

El Instituto Nacional de Oftalmología del Perú (INO) informó que de un total de 200,520 atenciones, 1938 han sido por patología de queratocono, lo cual equivale a 1% de atenciones realizadas en diferentes grupos etarios en el año 2017. No obstante, algunas instituciones públicas tanto como privadas poseen equipos sofisticados en los cuales se puede optimizar el uso de equipos biomédicos que permitan evaluar mejor la córnea, entre las cuales tenemos el topógrafo corneal (videoqueratoscopio) y la tomografía ocular.

En la presente investigación se tiene como finalidad realizar diagnósticos de queratocono empleando una técnica de Deep Learning. El diseño de la solución del sistema de detección de queratocono se realizará en la clínica Exilaser con el propósito de ser una herramienta de soporte para el experto en cuanto a la toma de decisiones.

De esta manera se presenta en la Figura 2, el Diagrama de Ishikawa en donde se detectan las causas que ocasionan que la baja precisión en la detección del queratocono:

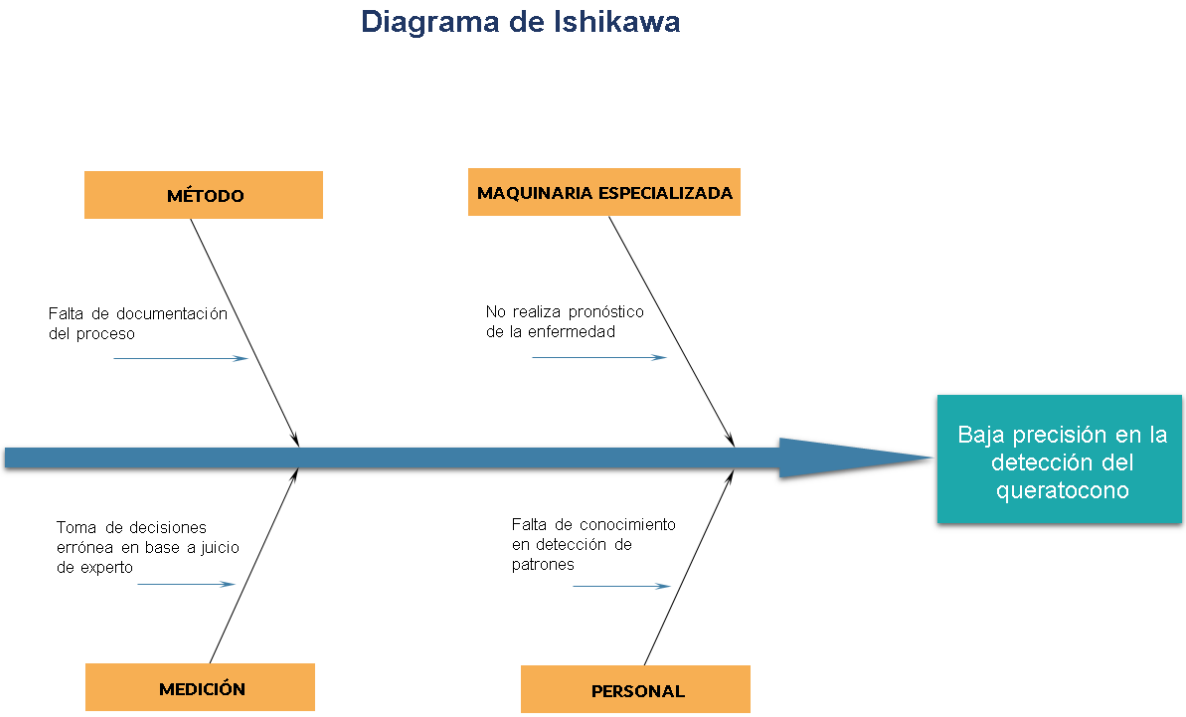


Figura 2. Diagram de Ishikawa de clínica Exilaser, elaboración propia

Formulación del problema

Problema general.

¿En qué medida el diseño y análisis de una solución informática mediante la aplicación de Deep Learning sobre imágenes topográficas mejorará la precisión del diagnóstico de queratocono en la clínica Exilaser?

Problemas específicos.

¿Cuál será el algoritmo a emplear en el diagnóstico de queratocono para mejorar el nivel precisión?

¿Cuál será el diseño del algoritmo para mejorar la precisión del diagnóstico del queratocono?

¿Qué tipo de pruebas se deben realizar a la aplicación de Deep Learning para mejorar la precisión del diagnóstico de queratocono?

¿Cuál será el rendimiento del algoritmo al clasificar las imágenes topográficas?

CAPÍTULO II

MARCO REFERENCIAL

Antecedentes

Internacionales.

En BRAC University ubicada en Bangladés, Rakib Hossain Khan (2018) realizó una tesis de bachiller sobre “Deep Learning Based Medical X-ray Image Recognition and Classification”. El objetivo de la tesis fue proporcionar al personal médico una herramienta que permita identificar e interpretar las imágenes de rayos X del tórax. En la tesis se concluye que el uso de redes neuronales convolucionales cumple con las funciones de reconocimiento de enfermedades del tórax siendo aplicadas a imágenes, en donde se obtuvo un diagnóstico con una precisión de un 86% en el procesamiento de más de 112,000 imágenes de rayos X de 30,000 pacientes. En la tesis se recomienda tener un mayor número de imágenes de radiografías del tórax para obtener mejores resultados.

En Chalmers University of Technology ubicada en Suecia, Charlotta Aguirre Nilson y Medina Velic (2018) realizaron una tesis de maestría sobre “Classification of Ulcer Images Using Convolutional Neural Networks”. El objetivo de tesis fue realizar diagnósticos de úlceras venenosas en las piernas de los pacientes mediante un algoritmo automatizado. En la tesis se concluye que al utilizar redes neuronales convolucionales con VGG19 como base de la arquitectura se puede conseguir una precisión de 82% en el diagnóstico de úlceras venenosas utilizando 300 imágenes para los diagnósticos. En la tesis se recomienda incrementar el número de imágenes de úlceras en piernas para obtener mayor precisión asimismo realizar un pre-entrenamiento que puede mejorar el rendimiento con la mejora de un hiperparámetro.

En Università degli Studi di Padova ubicada en Italia, Enrico Vicenzi (2017) realizó una tesis de titulación sobre “A deep learning approach to the analysis of retinal images”. El objetivo de la tesis fue implementar una red neuronal convolucional que permita clasificar la retinopatía diabética de imágenes retinianas así como discriminar cuales contienen marcadores inteligentes que fueron creados a partir de las imágenes originales. En la tesis se concluye que el diagnóstico de retinopatía diabética obtuvo un 70% de certeza debido problemas de sobreajuste u overfitting, siendo esto perjudicial para el aprendizaje de la red neuronal que tiene como base el diseño VGG16, por otra parte la clasificación de marcadores inteligentes dio como resultado un 100% en las pruebas realizada. En la tesis se recomienda enfocarse en los filtros de red neuronal, ya que permiten medir el rendimiento del algoritmo en la etapa de entrenamiento.

En Linköping University ubicada en Suecia, Philip Johansson (2019) realizó una tesis de maestría sobre “Incremental Learning of Deep Convolutional Neural Networks for Tumour Classification in Pathology Images”. El objetivo de la tesis fue clasificar imágenes histopatológicas utilizando métodos de aprendizaje incremental. En la tesis se concluye que el uso de métodos como Learning without Forgetting (LwF) y Generative Adversarial Networks (GAN) permiten entrenar el algoritmo menor tiempo a comparación de una red desde cero, el método LwF es un 60% más rápido mientras que GAN es 45% menos eficiente en la clasificación de imágenes histopatológicas. En la tesis se recomienda que se enfoque en la combinación equilibrada de la destilación y perdida entropía cruzada con diferentes funciones de perdida para la mejora del rendimiento.

En Technischen Universität Wien ubicada en Alemania, Philipp Seeböck (2015) realizó una tesis de licenciatura sobre “Deep Learning in Medical Image Analysis”. El objetivo de la tesis fue clasificar imágenes de tomografías computarizados del pulmón así como la clasificación de dígitos escritos a mano. En la tesis se concluye que el uso de diversos algoritmos para la clasificación de tomografías computarizadas de pulmón y las imágenes dígitos escritos a mano varía acorde al algoritmo a utilizar así como la identificación de patrones, en el caso del algoritmo K-Nearest Neighbor el resultado de la clasificación fue de un 15% para la clasificación de tomografías computarizadas mientras que las imágenes de dígitos escritos a mano fue de 96%. En la tesis se recomienda que se escoja el algoritmo adecuado de acuerdo al nivel de complejidad, la cantidad de imágenes a procesar y los hiperparámetros para no perjudicar la precisión del algoritmo.

Nacionales.

En Pontifica Universidad Católica del Perú, Gabriel Jiménez (2019) realizó una tesis de maestría sobre “Deep Learning for Semantic Segmentation versus Classification in Computational Pathology: Application to mitosis analysis in Breast Cancer grading”. El objetivo de la tesis fue clasificar de manera correcta el cáncer de mama utilizando algoritmos de Deep Learning. En la tesis se concluye que el uso de redes neuronales convolucionales bajo 2 arquitecturas, tales como AlexNet, donde se realizaron pre-procesamientos a las imágenes para alcanzar una precisión de 95.08% y la arquitectura U-Netv, donde no se realizaron pre-procesamientos obteniendo una precisión de 97.73% siendo esta la más robusta entre las dos propuestas y la cual obtuvo un mayor nivel de precisión.

En la Universidad Tecnológica del Perú, Giorzinio Tejeda y Renzo Gonzales (2020) realizaron una tesis de titulación sobre “Arquitectura de red neuronal convolucional para diagnóstico de cáncer de piel”. El objetivo de la tesis fue detectar cáncer de piel mediante una red neuronal aplicando Deep Learning. En la tesis se concluye que de los diseños propuestos, el tercer diseño obtuvo un 97% de precisión al clasificar las imágenes utilizando las capas básicas de redes neuronales convolucionales y aplicando la técnica e aumento de datos. En la tesis se recomienda que se utilice una etiqueta tipo categórica para realizar la carga de diversas imágenes mediante Keras, librería utilizada en la tesis.

En la Universidad Nacional del Altiplano, Erech Ordoñez (2020) realizó una tesis de doctorado sobre “Deep Learning para la visión artificial e identificación del personal administrativo y docente de la Universidad Nacional Micaela Bastidas de Apurímac 2018”. El objetivo de la tesis fue identificar al personal de la institución educativa mediante la identificación de una videocámara para detectar al personal con la finalidad de capturar la imagen y realizar la clasificación del personal. En la tesis se concluye que los dos modelos de redes neuronales convolucionales propuesto basados en VGG16 y DenseNet121 generaron un nivel de precisión de 98.05% y 99.32% respectivamente mediante un aprendizaje supervisado. En la tesis se recomienda emplear un aprendizaje no supervisado para que la recolección de las imágenes se haga de manera automática.

En la Universidad San Ignacio de Loyola (2016), Kevin Ayuque (2016) realizó una tesis de titulación sobre “Diseño de un sistema de clasificación de señales de tránsito vehicular utilizando redes neuronales convolucionales”. El objetivo de la tesis fue realizar la correcta clasificación de señales de tránsito mediante el uso de redes neuronales convolucionales. En la tesis se concluye que los diseños presentados han generado un alto nivel de precisión, resaltando uno de ellos con el valor de 95.29% al realizar el aprendizaje con imágenes de señales de tránsito de Alemania y hacer las pruebas con señales de tránsito locales. En la tesis se recomienda realizar diverso pre-procesamientos a las imágenes así como procesar imágenes con mayor dimensión debido a que el algoritmo a presentar mayor complejidad en el diseño puede aprender mejor.

Estado del arte

En los últimos años los datos han crecido abruptamente, la empresa International Data Corporation (IDC) realizó un pronóstico de acuerdo a la velocidad de crecimiento de los datos, en donde se pronostica que para el año 2019 los datos tendrán un peso aproximado de 40 Zettabytes (equivalente a un Zettabyte es de 10^{21} bytes) así mismo para el año 2025 el crecimiento de datos será de 175 Zettabytes.

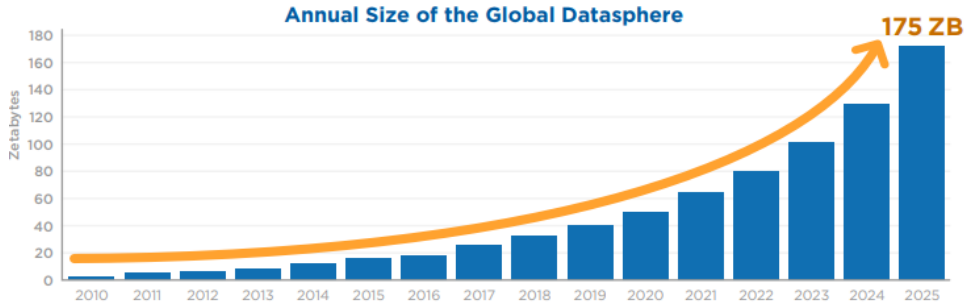


Figura 3. Crecimiento anual de datos de Global Datasphere. Adaptado de "Annual Size of the Global Datasphere", por Seagate, 2018

Estudios realizados por investigadores demuestran que el crecimiento de datos puede ser aprovechado de tal manera que el uso de algoritmos de aprendizaje profundo permitan realizar predicciones con un alto nivel de precisión en base a los datos, es por ello que los últimos estudios realizados están enfocados en el uso de redes neuronales convolucionales como el algoritmo óptimo para la clasificación de imágenes.

El estudio realizado por Lavric y Valentin (2019) denominado "KeratoDetect: Keratoconus Detection Algorithm Using Convolutional Neural Networks", evidencia que el uso de redes neuronales convolucionales aplicadas a imágenes es efectivo, dando como resultado una precisión de 99.33%, debido a la arquitectura de algoritmo en donde se utilizan capas de agrupación máxima (max pooling) y de normalización para la optimización la red.

De la misma manera Aranha et al. (2019) Con su trabajo denominado "CorneaNet: fast segmentation of cornea OCT scans of healthy and keratoconic eyes using deep learning", evidenció que el uso de redes neuronales convolucionales aplicadas a imágenes dan como resultado una precisión de 99.56%, en este estudio se evaluaron diversas arquitecturas en donde se utilizaron capas de agrupación promedio (average pooling) y normalización así como el rendimiento del ordenador al procesar más de 140,000 imágenes.

Marco teórico

Queratocono.

Sánchez, Álvarez, Benavides, Sánchez y Zambrano (2018) explican que queratocono es una enfermedad atípica que afecta a la córnea, adelgazando su grosor y deformándolo hasta tomar forma de cono generando de esta manera una disminución gradual de la visión, visión borrosa o distorsionada. Las causas de esta enfermedad aún no se han identificado por lo que existen diversos factores, los cuales son:

Edad del paciente

Antecedentes familiares con queratocono

Frotarse los ojos constantemente

Estabilidad refractiva

Así mismo Sánchez et al. (2018) indica que la enfermedad no distingue el género y suelen aparecer entre la edad de 20 a 40 años, si bien es cierto que su desarrollo comienza en la niñez, toma aproximadamente 10 años en presentarse. La detección de la enfermedad en una etapa inicial es complicada debido a que recién se manifestarán las incomodidades con la visión o se puede confundir con otra enfermedad, es por ello que se utilizan equipos especiales como el topógrafo corneal, ya por el contrario en un estado avanzado es más fácil de detectar.

La Clínica Mayo explica que la identificación del queratocono se puede realizar a través de diversos métodos, los cuales son:

Refracción ocular:
Se realizan pruebas con un equipo especial para medir los ojos y detectar problemas de visión.

Examen con lámpara de hendidura:
Se utiliza un equipo especial que emite una luz vertical hacia el ojo asimismo un microscopio para observarlo.

Queratometría:
Se emplea un equipo que emite una luz en forma de círculo, la cual enfoca a la córnea y mide el reflejo para visualizar la forma de la córnea.

Mapeo computarizado de la córnea:
Se utiliza un equipo especializado para generar imágenes, tales como la tomografía de coherencia óptica y la topografía corneal también se utiliza para grabar imágenes de la córnea que permiten visualizar de manera más detallada la superficie, así mismo se utiliza para medir el grosor de la córnea.

Sánchez et al. (2018) indican que el topógrafo corneal computarizada y la videoqueratoscopia han sido los métodos más empleados para la detección del queratocono, así mismo mencionan los diversos tratamientos para controlar el avance de esta enfermedad y en casos extremos eliminarlo, los cuales son:

Lentes de contacto

Anillos intracorneales

Cross-Linking corneal

Trasplante de córnea o queratoplastia

Inteligencia Artificial.

Inteligencia artificial está asociado en simular el razonamiento humano mediante el uso de máquinas que están programadas para pensar e imitar sus acciones, lo cual está asociado con el aprendizaje y resolución de problemas para lograr un objetivo específico con la mejor oportunidad que se presente. (Frankenfield, 2019)

Ibertech (2016) indica que existen diferentes métodos de análisis de datos, los cuales son:

Análisis descriptivo:
El análisis descriptivo es aquel que se encarga de estudiar y describir lo que ha pasado, utilizando tablas, gráficos e informes.

Análisis predictivo:
Son aquellos componentes o materia prima que han sufrido cambios en sus procesos, pero no están culminados, los cuales son almacenados para un posterior proceso de transformación.

Análisis prescriptivo:
Contribuye a comprender que debemos realizar para conseguir los resultados que quisiéramos en el futuro. Eso significa que vamos a utilizar la Inteligencia Artificial para hallar qué tenemos que hacer para lograr los resultados que quisiéramos.

Aprendizaje Automático.

Es una rama de la inteligencia artificial que busca desarrollar técnicas o algoritmos, las cuales permitan a un sistema o máquina aprender patrones de los datos brindados para predecir el comportamiento futuro. (Sancho, 2017)

Aprendizaje automático o Machine Learning (ML) cuenta con dos grandes áreas para predecir eventos en base a la data, los cuales son: aprendizaje supervisado y aprendizaje no supervisado. La base para cada una de las áreas mencionadas previamente parte desde la intervención humana, tal es el caso del aprendizaje supervisado en donde se puede realizar predicciones con intervención humana mientras

que en el aprendizaje no supervisado no existe la intervención humana para las predicciones. (González, 2014)

Aprendizaje supervisado:

Aprendizaje supervisado o Supervised Learning, utiliza datos etiquetados para conocer cuáles son sus características con la finalidad de que se puedan clasificar datos similares pero no los que no hayan sido etiquetados o tengan modelo con el cual compararlo. (Vasilev, Slater, Spacanga, Roelants y Zocca, 2019)

Según Vasilev et al. (2019), los algoritmos de aprendizaje supervisado más conocidos son los siguientes:

Regresión lineal y logística.

Máquina de vectores de soporte.

Arboles de decisiones.

Bayesiano ingenuo.

Redes neuronales.

Aprendizaje no supervisado:

Aprendizaje no supervisado o Unsupervised Learning, utiliza datos sin etiquetas, esto quiere decir que la máquina dará una predicción en base a los patrones detectados (Vasilev et al., 2019).

Según Vasilev et al. (2019), los algoritmos de aprendizaje no supervisado más conocidos son los siguientes:

K-means

C-means.

Aprendizaje semi-supervisado:

Aprendizaje semi-supervisado o Semi-supervised Learning, hereda características del aprendizaje supervisado y no supervisado por lo que existen datos que están asociados a etiquetas y son pocas las que no se les ha asociado alguna. (Predycz y Ming, 2020)

Según Predycz y Chen (2020), el uso del aprendizaje semi-supervisado tiene más beneficios que al usar aprendizaje supervisado por los siguientes aspectos

Mayor precisión

Menor tiempo de implementación

Dentro del aprendizaje supervisado se encuentra las redes neuronales, que serán utilizadas en el presente trabajo de investigación.

Redes Neuronal Artificiales.

Redes neuronales artificiales o Artificial Neural Network (ANN), es un modelo matemático que permite procesar y aprender de la misma manera en que lo haría el cerebro, se utilizan elementos simples denominados neuronas artificiales, las cuales están interconectadas entre sí con la finalidad de intercambiar la información procesada aplicando señales de activación en cada neurona y determinar la unidad de salida (Vasilev et al., 2019).

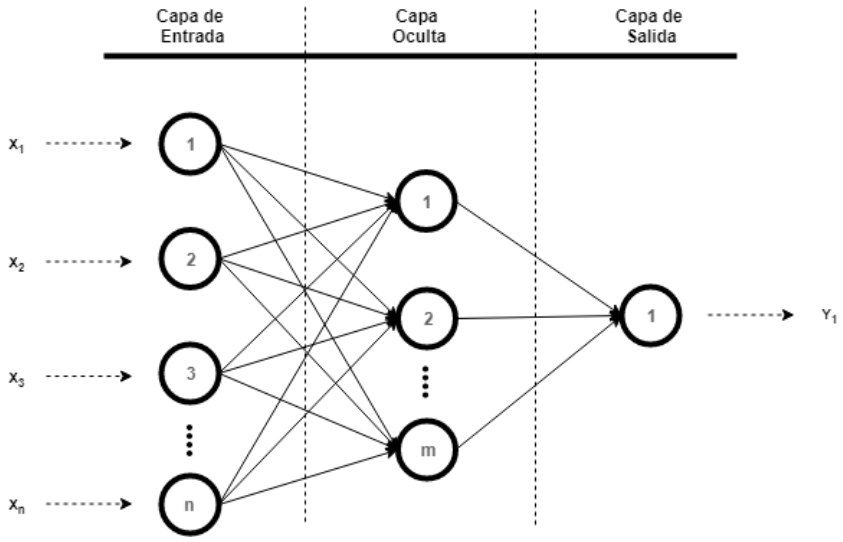


Figura 4. Diseño de una red neuronal artificial con una capa oculta. Adaptado de *"Python Deep Learning: Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and TensorFlow"*, elaboración propia

La Neurona artificial es la unidad base de una red neuronal artificial que nace bajo el concepto de una neurona biológica, el cual es una función matemática que puede tomar varios valores de entrada y dar como resultado un único valor numérico de salida (Vasilev et al., 2019).

Elena Verger (2017), explica sobre las partes de una neurona, las cuales son 4 unidades base:

Dendritas:

Su función es de recibir las señales de otra neurona para así conducirlo al cuerpo celular o soma.

Cuerpo celular (Soma):

Su función es de mantener con vida a la neurona para que pueda continuar con su función.

Axón:

Su función es de conducir las señales de la neurona a través del soma hacia otras neuronas.

Sinapsis:

Su función es de comunicación entre neuronas mediante la conexión de las partes terminales del axón y el cuerpo celular o las dendritas.

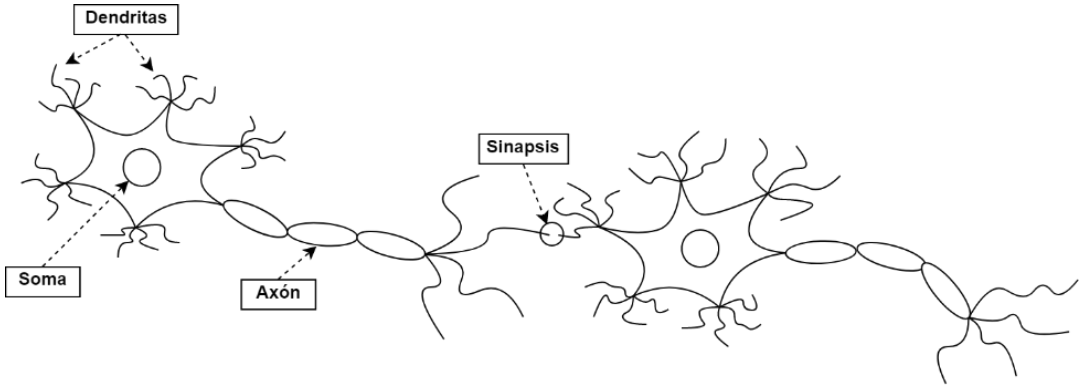


Figura 5. Neurona biológica. Adaptado de *"Wikipedia"*, elaboración propia

El diseño de una neurona artificial tomando en cuenta las unidades bases mencionadas previamente es el siguiente:

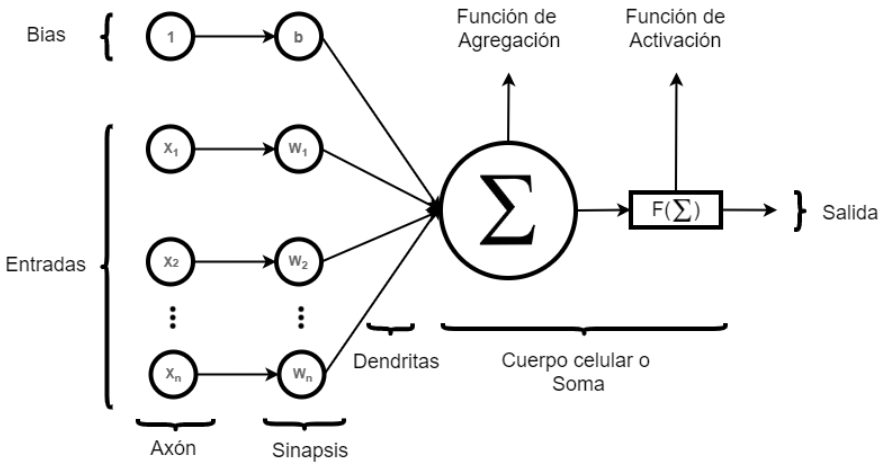


Figura 6. Neurona Artificial. Adaptado de *"Python Deep Learning: Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and TensorFlow"*, elaboración propia

La unidad de salida de la neurona depende la función de activación a utilizar.

$$y = f\left(\sum_{i=1}^{n} x_i w_i + b\right)$$

En donde:

y = Unidad de salida

n = Cantidad de datos de entrada

w_i = Peso de conexión

x_i = Activación de unidad de entrada

b = Peso sesgado o bias

Función de activación

La función de activación es la encargada de generar la unidad de salida utilizando funciones no lineales para resolver problemas complejos en base a lo aprendido asimismo es quién decide si una neurona debe de ser activada o no. (Gupta, 2017)

De la misma manera Calvo (2018) indica que la función de activación es aquella que retorna valores entre los rangos [0 a 1] o [-1 a 1] a partir del valor de entrada, así mismo se detalla que las funciones de activaciones más utilizadas en redes neuronales son:

Tabla 1

Comparación de funciones de activación

Función de activación	Descripción	Características	Formulas
Sigmoidea	La activación genera un valor que oscila entre [0, 1]	El gradiente sufre de saturación hasta el punto de morir La convergencia tiende a ser lenta El rendimiento de la función es óptima en la última capa	$f(x) = \frac{1}{1+e^{-x}}$
Tangente hiperbólica	La activación genera un valor que oscila entre [-1, 1]	Se asemeja a la función Sigmoidea El rendimiento de la función de activación es óptima en redes neuronales recurrentes	$f(x) = \frac{e(x) - e(-x)}{e(x) + e(-x)}$ $f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$
ReLU (Rectified Linear Unit)	La activación genera un valor que oscila entre [0,∞]	La activación de la neurona se da solo con valores positivos por lo que pueden morir bastantes neuronas El rendimiento de la función de la activación es óptima en redes convolucionales	$f(x) = max(0, x)$ $f(x) = \begin{cases} 0\ para\ x < 0 \\ x\ para\ x \geq 0 \end{cases}$
Leaky ReLU	Se asemeja a ReLU pero utiliza un coeficiente de rectificación	Se asemeja a la función ReLU con la diferencia de que los valores negativos son penalizados	$f(x) = \begin{cases} 0\ para\ x < 0 \\ x * \alpha\ para\ x \geq 0 \end{cases}$
Softmax	La activación genera un valor que oscila entre [0, 1].	Es presentando mediante probabilidades por lo que es diferenciable El rendimiento de la función es óptima en la última capa ya que las multiclases son normalizadas	$f(Z)_j = \frac{e^{Z_j}}{\sum_{K=1}^{K} e^{Z_k}}$

***Nota**:* Adaptado de *"Función de activación – Redes neuronales"*, elaboración propia

Dentro de la comparativa de funciones de activación se utilizarán las funciones de ReLU y softmax para el presente de investigación.

Aprendizaje Profundo.

Aprendizaje profundo o Deep Learning (DL) es un subcampo del aprendizaje automático que busca emplear algoritmos que imiten el funcionamiento del cerebro, lo que permitiría que las máquinas lleguen a ser inteligentes. (Moolayil, 2019)

Vasilev et al. (2019), explica que existen diversos algoritmos que se enfocan en partes específicas de la simulación de las funciones del cerebro, los cuales son:

Tabla 2

Comparación de algoritmos de aprendizaje profundo

Características	Perceptrón Multicapas	Redes Neuronales Convolucionales	Redes Recurrentes	Autoencoders
Detalle	Red que clasifica los datos utilizando capas ocultas	Red que aprende de las características extraídas en las capas de convolución para interpretar imágenes	Red que aprende de manera secuencial, los pesos son compartido a través de las neuronas	Red que se entrena para extraer y reducir las dimensiones de las características donde el número de entradas es igual al de salidas
Aplicación	Reconocimiento de voz y traducción automática	Reconocimiento de imágenes	Reconocimiento de voz y texto	Modelos generativos
Tipo de datos	Conjunto de datos tabulares	Imágenes	Voz y texto	Imágenes
Pros	Es el algoritmo más utilizado con un alto nivel de precisión	Buen rendimiento y el aprendizaje del modelo es rápido	Existen diversas variantes para la implementación	No necesita datos etiquetados
Contras	Procesamiento de los datos es lento	Necesita gran cantidad de datos para la clasificación	Necesita gran cantidad de datos y cuenta con problemas de gradiente de fuga	Necesita gran cantidad de datos para la clasificación

***Nota**:* Adaptado de *"Python Deep Learning: Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and TensorFlow"*, elaboración propia

El algoritmo a emplear en el trabajo de investigación es de redes neuronales convolucionales debido a que es el más eficiente en el reconocimiento de imágenes.

Redes Neuronales Convolucionales

Es una variante de las redes neuronales multicapas o perceptrón multicapa, el cual tiene como principal objetivo el reconocimiento de imágenes, en donde el uso de capas especiales, tales como las de convolución, reducción y totalmente conectada permiten reconocer los patrones en cualquier sección de la imagen brindada, la cual puede ser en escala a grises o a colores. (Vasilev et al., 2019).

Capa convolucional

Capa convolucional o convolutional layer, realiza la extracción de características mediante la aplicación de filtros que se desplazan (stride) cierta cantidad de pixeles sobre imagen de entrada que puede ser a colores o en escala a grises, en cada ubicación se realiza una operación matricial y una suma dando como resultado un mapa de característica, la cual es enviada a la función de activación ReLU, dicha función es más utilizada en estos casos asimismo se puede utilizar un relleno (padding) para seguir con la extracción de características de bajo nivel debido a que la aplicación de capas de convolución disminuye velozmente el volumen de la imagen de entrada. (Vasilev et al., 2019)

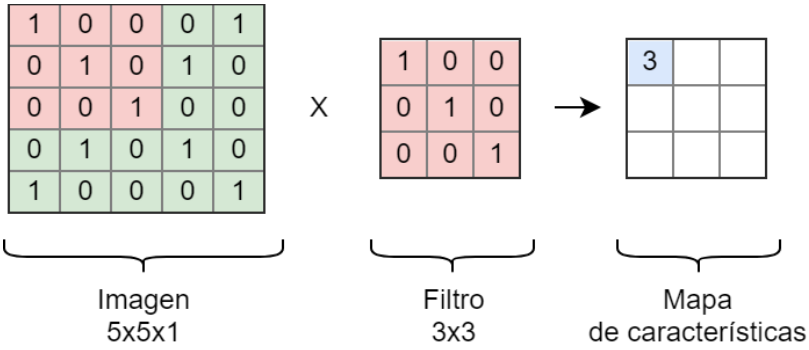


Figura 7. Convolución aplicada a una imagen 5x5x1 con 1 desplazamiento y un filtro de 3x3. Adaptado de *"Python Deep Learning: Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and TensorFlow"*, elaboración propia

Skalski (2019) indica que el método para obtener el mapa de características mediante la siguiente formula:

$$G[m,n] = (f * h)[m,n] = \sum_j \sum_k h[j,k] * f[m-j, n-k].$$

En donde:

f = Tamaño de la imagen

h = Tamaño del filtro

m = Numero de fila del mapa de características

n = Numero de columna del mapa de características

j = Contador de numero de filas

k = Contador de numero de columnas

Como resultado se obtiene el siguiente mapa de características:

3	0	1
0	3	0
1	0	3

Figura 8. Resultado de una capa de convolución. Adaptado de *"Python Deep Learning: Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and TensorFlow"*, elaboración propia

Capa de agrupación

Capa de agrupación o pooling layer, reduce el tamaño de la imagen tras el proceso de convolución con la finalidad de utilizar una menor cantidad de parámetros en la red, existen dos tipos de capa de agrupación, tales como de agrupación máxima (max pooling) y agrupación promedio (average pooling), donde la más utilizada es la agrupación máxima, que busca almacenar solo los valores máximo. (Vasilev et al., 2019)

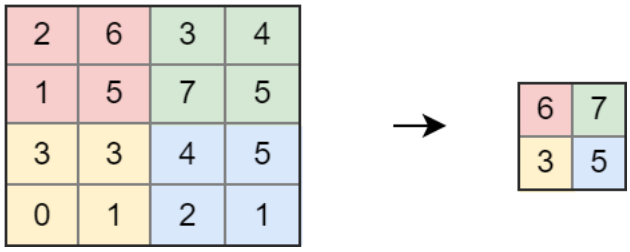


Figura 9. Agrupación máxima aplicando un filtro de 2x2. Adaptado de *"Python Deep Learning: Exploring deep learning techniques, neural network architectures and GANs with PyTorch, Keras and TensorFlow"*, elaboración propia

Capa totalmente conectada

Capa totalmente conectada o fully connected layer, conecta todas las neuronas de entrada que sirven para alimentar a las neuronas de salida, esto permite que el gradiente al hacer su retorno contribuya a la generación de una nueva neurona (Vasilev et al., 2019)

Vasilev et al. (2019) explica que existen diversas técnicas para mejorar el rendimiento de una red neuronal convolucional, las cuales son:

Pre procesamiento de datos

El procesamiento de imágenes maneja un rango de pixeles que oscilan entre 0 a 255, los cuales pueden influir negativamente al aprendizaje del modelo según intensidad que tengan los canales asociados a la imagen, por ello se debe de normalizar los datos antes de realizar el procesamiento en la red, los tipos de normalización que se pueden utilizar son los siguientes:

Escalado de características

Realiza una conversión de los pixeles de entrada en un rango de 0 a 1 mediante la siguiente formula:

$$x = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Considerando que se tiene un pixel con una intensidad de 100, se aplicará un ejemplo práctico.

$$x = \frac{100 - 0}{255 - 0}$$

$$x = \frac{100}{255} = 0.39$$

Este tipo de normalización es uno de los más implementados así como el más rápido de utilizar, por ello se utilizará para el procesamiento del presente trabajo de investigación.

Puntuación estándar

Normaliza los datos utilizando el promedio y la desviación estándar aplicándolo a cada canal de la imagen exclusivamente aquellas imágenes que será utilizada para entrenar y probar el algoritmo, mediante la siguiente formula:

$$x = \frac{x - \mu}{\sigma}$$

Regularización

En los algoritmos pueden ocurrir dos tipos de problemas, tales como sobreajuste (overfitting), el cual se genera cuando los datos de entrenamiento no cubren datos nuevos al realizar las pruebas, y subajuste (underfitting), este

se genera de manera contraria al sobreajuste, ya que se debe a la falta de datos de prueba durante el entrenamiento.

Es por ello que existen diversas técnicas que permiten mitigar estos problemas durante el aprendizaje del modelo, los cuales son:

Pérdida de peso

Esta técnica también conocida como regularizador L2, penaliza los grandes pesos en la red previniendo el uso de algunas características asociadas a estos pesos grandes, logrando de esta manera reducir el sobreajuste de la red.

Dropout

Elimina de manera aleatoria las neuronas con las conexiones de entrada y salida permitiendo de esta manera que la red no sufra de sobreajuste. La capa dropout se puede aplicar en las capas de convolución, agrupación o totalmente conectadas.

Aumento de datos

Se utiliza cuando el tamaño de datos para entrenar es pequeño y puede generar sobreajuste, la finalidad de esta técnica consiste en el aumento del tamaño de la data de entrenamiento a través de diversos métodos, tales como rotación, vuelta horizontal y vertical, aumentar y disminuir el enfoque de la imagen, cortar, sesgar y el incremento de brillo y contraste.

Normalización por lotes

Permite normalizar la salida de las neuronas por cada lote manteniendo la función de activación cerca de 0 y la desviación estándar cerca de 1. La capa de normalización por lotes o batch normalization tiene una similitud a la puntuación estándar explicada en el pre procesamiento de datos.

Moolayil (2019) indica que se puede ajustar o mejorar el rendimiento del algoritmo a utilizar utilizando los hiperparámetros, algunos de ellos son:

Número de capas

La cantidad de capas implementadas en el diseño del algoritmo afecta el rendimiento, es por ello que si se utiliza un diseño complejo se necesitará de un mayor recurso computacional, por otro lado el uso de unos diseños con

pocas capas pueden generar overfitting y se necesitan utilizar técnicas específicas como dropout y L2 para mitigarlo.

Número de épocas

Está asociado al número de veces que un conjunto de datos se ha propagado en la red neuronal por lo que si se incrementa el número de épocas puede obtenerse mejores resultados aunque puede necesitarse un mayor recurso computacional e incrementa el tiempo de aprendizaje.

Inicialización de pesos

Un uso correcto de inicialización de pesos permite al algoritmo acelerar el proceso de entrenamiento así como la reducción de puntos muertos.

Tamaño de lote

Cantidad de elementos a procesar en la red neuronal. Un tamaño adecuado permite al algoritmo aprender de manera progresiva, es recomendable utilizar un tamaño de 32 a 64.

Tasa de aprendizaje

Permite controlar la velocidad de aprendizaje, ya que se utiliza para actualizar los pesos en cada iteración.

Optimizadores

Permite cambiar los parámetros de la red neuronal, tales como Adam, Adamax, Nadam, SGD, RMSprop, entre otros. En el presente trabajo de investigación se utilizará Adam.

CAPÍTULO III

OBJETIVOS

Objetivo general

Diseñar y analizar una solución informática mediante la aplicación de Deep Learning sobre imágenes topográficas para mejorar la precisión del diagnóstico de queratocono en la clínica Exilaser.

Objetivos específicos

Seleccionar el algoritmo a emplear en el diagnóstico de queratocono, utilizando el criterio de nivel de precisión.

Diseñar la arquitectura del algoritmo seleccionado para mejorar la precisión del diagnóstico del queratocono.

Realizar pruebas a la aplicación de Deep Learning para mejorar la precisión de diagnóstico del queratocono.

Evaluar el rendimiento del algoritmo al clasificar las imágenes topográficas.

JUSTIFICACIÓN

Teórica

Este trabajo de investigación tiene como propósito fomentar el conocimiento de Deep Learning, el cual emplea redes neuronales artificiales para realizar procesamiento de texto, reconocimiento de imágenes, objetos y voz; la particularidad de la aplicación de estos sistemas es que el algoritmo que se emplee aprende por sí mismo simulando de esta manera las decisiones tomadas por los seres humanos, es por ello que a mayor cantidad de datos se tenga, la precisión será más alta. El tema principal del trabajo de investigación está centrado en realizar diagnósticos empleando Deep Learning para analizar las imágenes generadas por el topógrafo corneal OPD - Scan III.

Práctica

Los problemas identificados son:

La precisión de los diagnósticos

El equipo especializado no detecta correctamente la enfermedad denominada queratocono en una etapa inicial por lo que se necesita identificar patrones para realizar un diagnóstico más preciso.

Tiempo de análisis

El ser humano no puede procesar grandes cantidades de datos para la identificación patrones sin ser eficiente en tiempo y recursos.

Esta solución tendrá diagnósticos más precisos dependiendo de la cantidad de imágenes topográficas que cuente el sistema de detección de queratocono por lo que permitirá reducir costo, tiempo y uso de recursos además de servir de apoyo para los médicos tratantes en la detección de la enfermedad.

Social

La implementación de la solución será aplicada en la clínica Exilaser en un primer plano con tendencia a expandirse a un nivel nacional, en donde los usuarios internos y externos se verán beneficiados, esto se debe a que con más datos generados de los pacientes la precisión de diagnósticos será más alta asimismo los tratamientos serán más certeros por parte del médico tratante.

HIPÓTESIS

Tabla 3

Hipótesis

Tipo de Hipótesis	Hipótesis Nula	Hipótesis Alterna
Hipótesis General	El diseño y análisis de una solución informática mediante la aplicación de Deep Learning sobre imágenes topográficas no mejorará el diagnóstico de queratocono en la clínica Exilaser	El diseño y análisis de una solución informática mediante la aplicación de Deep Learning sobre imágenes topográficas mejorará el diagnóstico de queratocono en la clínica Exilaser
Hipótesis Especifica	El algoritmo seleccionado para el diagnóstico de queratocono no utiliza el criterio de nivel de precisión	El algoritmo seleccionado para el diagnóstico de queratocono utiliza el criterio de nivel de precisión
	La arquitectura del algoritmo no mejorará el nivel precisión del diagnóstico del queratocono	La arquitectura del algoritmo mejorará el nivel precisión del diagnóstico del queratocono
	Las pruebas realizadas a la aplicación de Deep Learning no mejorará la precisión del diagnóstico de queratocono	Las pruebas realizadas a la aplicación de Deep Learning mejorará la precisión del diagnóstico queratocono
	El rendimiento del algoritmo no será optimo al clasificar las imágenes topográficas	El rendimiento del algoritmo será optimo al clasificar las imágenes topográficas

Nota: Tabla de hipótesis, elaboración propia

MATRIZ DE CONSISTENCIA

Tabla 4

Matriz de consistencia

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES	INDICADORES
Problema General:	**Objetivo General:**	**Hipótesis General**:	**Independiente:**	**Variable independiente:**
¿En qué medida el diseño y análisis de una solución informática mediante la aplicación de Deep Learning sobre imágenes topográficas mejorará la precisión del diagnóstico de queratocono en la clínica Exilaser?	Diseñar y analizar una solución informática mediante la aplicación de Deep Learning sobre imágenes topográficas para mejorar la precisión del diagnóstico de queratocono en la clínica Exilaser	El diseño y análisis de una solución informática mediante la aplicación de Deep Learning sobre imágenes topográficas mejora el diagnóstico de queratocono en la clínica Exilaser	Aplicación de Deep Learning sobre imágenes topográficas	Cantidad de diagnósticos correctos de queratocono
Problemas Específicos:	**Objetivos Específicos:**	**Hipótesis Específicas:**	**Dependiente:**	**Variable dependiente:**
¿Cuál será el algoritmo a emplear en el diagnóstico de queratocono para mejorar el nivel precisión?	Seleccionar el algoritmo a emplear en el diagnóstico de queratocono, utilizando el criterio de nivel de precisión	El algoritmo seleccionado para el diagnóstico de queratocono utiliza el criterio de nivel de precisión	Precisión del diagnóstico de queratocono	Cantidad de casos de éxito aplicando algoritmo de redes neuronales convolucionales
¿Cuál será el diseño del algoritmo para mejorar la precisión del diagnóstico de queratocono?	Diseñar la arquitectura del algoritmo seleccionado para mejorar la precisión del diagnóstico de queratocono	La arquitectura del algoritmo mejorará el nivel precisión del diagnóstico de queratocono		Cantidad de capas a implementar
¿Qué tipo de pruebas se deben realizar a la aplicación de Deep Learning para mejorar la precisión del diagnóstico de queratocono?	Realizar pruebas a la aplicación de Deep Learning para mejorar la precisión de diagnóstico del queratocono	Las pruebas realizadas a la aplicación de Deep Learning mejorará el nivel de precisión del diagnóstico de queratocono		Precisión mayor al 90%
¿Cuál será el rendimiento del algoritmo al clasificar las imágenes topográficas?	Evaluar el rendimiento del algoritmo al clasificar las imágenes topográficas	El rendimiento del algoritmo será optimo al clasificar las imágenes topográficas		Tiempo de procesamiento

***Nota**:* Matriz de consistencia, elaboración propia

ALCANCES

La elaboración de la presente investigación se llevará a cabo en la clínica Exilaser que se encuentra ubicada en la provincia de Lima, distrito de Santiago de Surco, en donde se centrará exclusivamente en el diagnósticos de las imágenes generadas por el topógrafo corneal OPD - Scan III.

LIMITACIONES

Como limitaciones se han identificado los siguientes puntos:

La disponibilidad que el especialista pueda brindar para la recopilación de datos debido a que se trataran datos sensibles en el trabajo de investigación.

Cantidad de imágenes generadas por el topógrafo corneal OPD – Scan III.

Los trabajos de investigación de Deep Learning relacionados al sector salud son escasos por lo que es una tecnología emergente.

En base a la Metodología Fundamental para la Ciencia de Datos, la aplicación de Deep Learning se desarrollará hasta la etapa de evaluación del modelo propuesto, dado que no se realizará el análisis de implementación en la clínica Exilaser. La metodología mencionada se puede encontrar en el Capítulo V.

CAPÍTULO IV

MARCO METODOLÓGICO

Metodología

El trabajo de investigación empleará la investigación no experimental, ya que no se harán modificaciones directas a las variables de estudio para su posterior análisis. (Hernández, 2014), se contemplará un alcance correlacional porque se analizará cómo se comportará la propuesta de “aplicación de Deep Learning sobre imágenes topográficas” (variable independiente) sobre las “la precisión del diagnóstico del queratocono “(variable dependiente).

Así mismo se utilizará la Metodología Fundamental para la Ciencia de Datos, ya que proporcionará estrategias para solucionar problemas de ciencia de datos de manera agnóstica a la tecnología, datos o enfoques. (Rolling, 2015)

Paradigma

El paradigma es positivista. Sánchez explica que:

> Persigue la verificación rigurosa de proposiciones generales (hipótesis) a través de la observación empírica y el experimento en muestras de amplio alcance, y desde una aproximación cuantitativa, con el fin de verificar y perfeccionar leyes referidas a lo educativo. Su finalidad es verificar y controlar los fenómenos. (2013, p. 4).

Enfoque

El enfoque del trabajo de investigación es de tipo cuantitativo. Hernández, Fernández y Baptista explican que “Utiliza la recolección de datos para probar hipótesis con base en la medición numérica y el análisis estadístico, con el fin establecer pautas de comportamiento y probar teorías” (2014, p. 4)

Método

El método será de carácter no experimental, esto se debe a que las variables a analizar no se verán afectadas durante todo el proceso. Hernández et al. Sostienen que “Estudios que

se realizan sin la manipulación deliberada de variables y en los que sólo se observan los fenómenos en su ambiente natural para analizarlos.” (2014, p. 152)

VARIABLES

Independiente

Aplicación de Deep Learning sobre imágenes topográficas porque es el fenómeno que condicionará el resultado de la variable dependiente a través de acciones que se realizarán para cumplir con lo propuesto en el trabajo de investigación

Dependiente

Precisión del diagnóstico de queratocono porque es el fenómeno que se verá afectado por las acciones realizadas de la variable independiente.

POBLACIÓN Y MUESTRA

Población

La población del presente trabajo de investigación está enfocada en la cantidad de imágenes topográficas que cuente la clínica Exilaser, donde se identificaron un total de 456 imágenes, las cuales están se distribuyen en 228 ojos sanos y 228 con queratocono.

Muestra

La muestra será intencional dado que se necesita la totalidad de la población para obtener mejores resultados en el análisis de las imágenes

UNIDAD DE ANÁLISIS

La unidad de análisis del presente trabajo de investigación serán las imágenes topográficas ya que a partir de ellas se procederá a realizar los diagnósticos de queratocono.

INSTRUMENTOS Y TÉCNICAS

Instrumentos

Los instrumentos están relacionados a las herramientas que se utilizarán para realizar el diagnóstico de queratocono, por ello en el presente trabajo de investigación las herramientas a utilizar son:

Keras

Es una librería de redes neuronales de alto nivel diseñado bajo el enfoque de desarrollo rápido, en donde detecta automáticamente la disponibilidad de un GPU e intenta utilizarlo de lo contrario, utiliza el CPU para continuar con el correcto funcionamiento, esta librería soporta tanto redes convolucionales como recurrentes (Vasilev et al., 2019).

OpenCV

Es una librería de código abierto que brinda soporte a aplicaciones de visión por computadora en tiempo real así como mejorar el rendimiento del ordenador con respecto a su percepción de imágenes. (OpenCV, s. f.)

Python

Es un lenguaje de programación orientado a la programación orientada a objetos, el cual permite realizar desarrollos rápidos de aplicaciones web o servidor debido a la simplicidad de su sintaxis. (Python, s. f.)

Jupyter Notebook

Es una aplicación web de código abierto que permite compilar código de diferentes lenguajes de programación, dentro de un mismo documento de trabajo o libro se puede integrar textos imágenes o graficas que pueden ser compartidos por diferentes medios. (Jupyter Notebook, s. f.)

Técnicas

En la presente investigación las técnicas a utilizar están asociadas a la interpretación de las imágenes a través del programa que se implementará así como la generación de más imágenes debido a que la cantidad de imágenes es pequeña y esto puede repercutir en el aprendizaje de la red neuronal convolucional, las librerías que permitirán aplicar lo mencionado son las siguientes:

Keras

Una de las diversas funcionalidades que se utilizará de esta librería, es la que permite generar una gran variedad de imágenes con diversos filtros a partir de una imagen como muestra por lo que se puede incrementar el tamaño de la muestra de manera sustancial.

OpenCV

Interpretará las imágenes a través de los pixeles y la intensidad de cada uno de ellos, los valores de los pixeles oscilan entre 0 a 255 así mismo el canal que presente la imagen influirá significativamente en el programa.

CAPÍTULO V

PROCEDIMIENTOS Y MÉTODO DE ANÁLISIS

Procedimientos

Como se explicó en el Capítulo IV, la Metodología Fundamental para la Ciencia de Datos proporciona estrategias que se dividen en 10 etapas, las cuales se representan en la siguiente imagen.

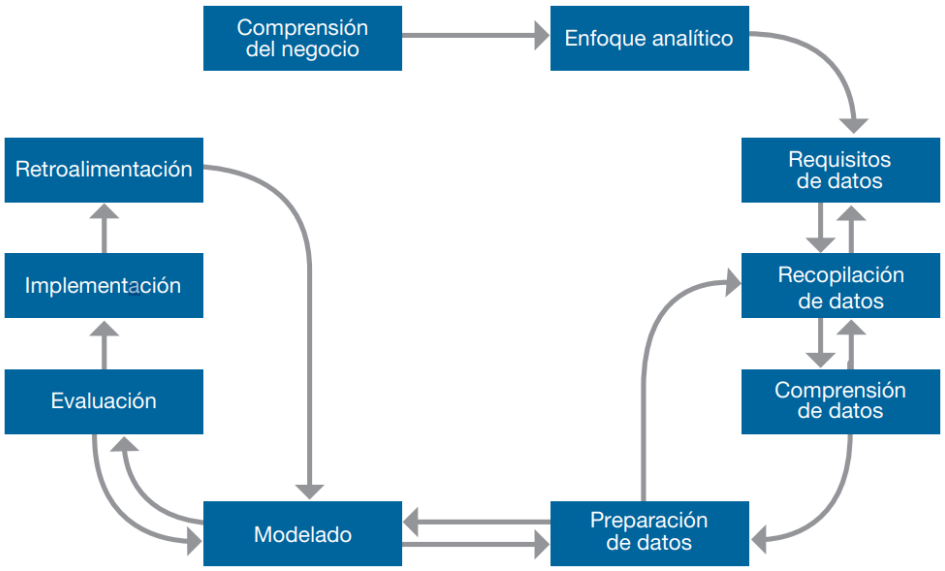


Figura 10. Pasos de la Metodología Fundamental para la Ciencia de Datos. Adaptado de "Metodología Fundamental para la Ciencia de Datos", por Jhon B. Rollins, 2013

El primer paso de la metodología es la comprensión del negocio, que busca definir cuál es la problemática que se presenta y como se puede dar solución. Esta etapa ha sido cubierta en el Capítulo I y III donde se explica sobre el problema de la investigación y los objetivos que se buscan alcanzar.

La siguiente etapa es el enfoque analítico, donde se busca adaptar la solución mediante técnicas de aprendizaje automático. Una de las técnicas que se utilizarán es de redes neuronales convolucionales, la cual forma parte del aprendizaje profundo, parte la interna del aprendizaje automático, con la finalidad de realizar una detección de queratocono con un alto nivel de precisión así como una detección temprana.

Los requisitos de datos, definen qué tipo de datos son los necesarios para la solución, los cuales pueden ser archivos multimedia, archivos planos, entre otros más. El tipo de dato que se necesita para la solución de aprendizaje profundo son imágenes por lo que se necesita los archivos generados por el equipo OPD Scan III.

La etapa de recopilación de datos, está enfocado en obtener los datos necesarios para el algoritmo, por otra parte en la etapa comprensión de datos, se utilizan técnicas que permiten evaluar la calidad de los datos, por lo que si se encuentran fallos en los datos recopilados puede retornar a la etapa anterior mencionada. En las etapas mencionadas se desarrolló un programa con Python que permite extraer la información necesaria de lo obtenido por el equipo OPD Scan III, ya que el equipo generaba múltiples reportes en un solo archivo.

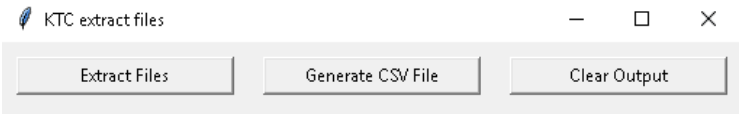


Figura 11. Programa para la extracción de imágenes, elaboración propia

Los archivos exportados del equipo especializado contienen múltiples reportes por lo que solo se consideraron aquellos que contiene “EY” en el nombre, ya que dentro de ese archivo se encuentran las imágenes necesarias.

Nombre	Tipo	Tamaño
AR0000010001L.CAB	Archivo .CAB	6 KB
AR0000010001R.CAB	Archivo .CAB	7 KB
ED0000010001L.cab	Archivo .CAB	31 KB
ED0000010001R.CAB	Archivo .CAB	35 KB
EY0000010001L.CAB	Archivo .CAB	105 KB
EY0000010001R.CAB	Archivo .CAB	130 KB
HD0000010001L.DAT	Archivo DAT	2 KB
HD0000010001R.DAT	Archivo DAT	2 KB
PE0000010001L.CAB	Archivo .CAB	1 KB
PE0000010001R.CAB	Archivo .CAB	1 KB
PP0000010001L.CAB	Archivo .CAB	237 KB
PP0000010001R.CAB	Archivo .CAB	243 KB
PR0000010001L.CAB	Archivo .CAB	1 KB
PR0000010001R.CAB	Archivo .CAB	1 KB
PU0000010001L.CAB	Archivo .CAB	220 KB
PU0000010001R.CAB	Archivo .CAB	225 KB
RA0000010001L.CAB	Archivo .CAB	24 KB
RA0000010001R.CAB	Archivo .CAB	28 KB
RE0000010001L.CAB	Archivo .CAB	123 KB
RE0000010001R.CAB	Archivo .CAB	128 KB

Figura 12. Reportes generados por el equipo OPD Scan III, elaboración propia

Al ejecutar la extracción de archivos este genera una carpeta nueva donde se almacenarán las imágenes que se utilizarán en el trabajo de investigación.

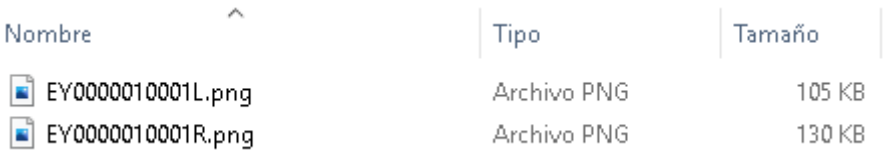


Figura 13. Extracción de imágenes, elaboración propia

El siguiente proceso fue listar los casos de ojos sanos y aquellos que padecen de queratocono con la finalidad de que otro proceso sea el encargado de realizar la distribución y clasificación de imágenes a la categoría respectiva.

Figura 14. Lista generada en base a las imágenes extraídas, elaboración propia

La siguiente etapa es la de preparación de datos, en donde se busca realizar tratamientos a la imagen con el objetivo de que se puedan utilizar las secciones más útiles, así como la definición del conjunto de datos a utilizar en la etapa de modelado. De esta manera se implementaron las técnicas mencionadas en el capítulo anterior con la finalidad de generar diversos casos, ya que la cantidad de la muestra es pequeña, por ello se realizará un proceso para incrementar la cantidad de imágenes:

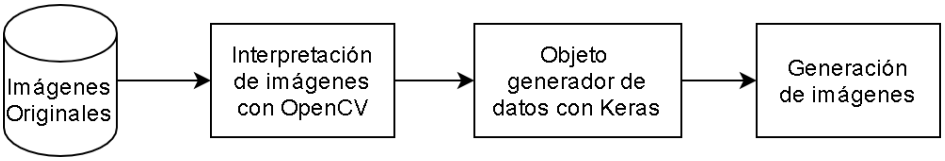


Figura 15. Proceso de generación de imágenes, elaboración propia

Como primer paso el programa debe de obtener la ruta de la carpeta donde se encuentran las imágenes dado que la librería OpenCV lo necesita para interpretarlo como una matriz de n por m, haciendo referencia a la anchura y altura respectivamente de la imagen.

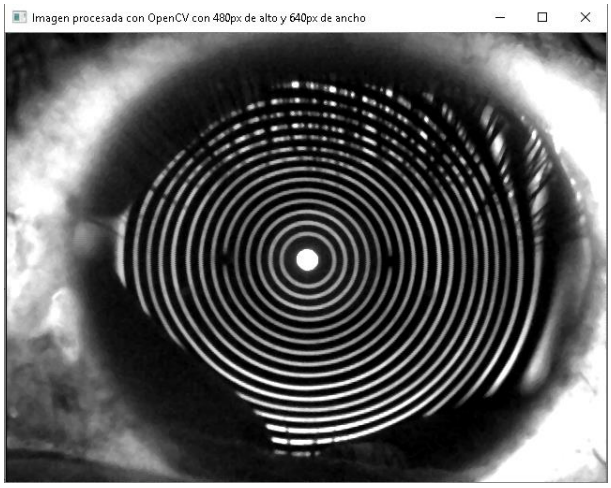


Figura 16. Demostración del procesamiento de una imagen utilizando OpenCV, elaboración propia

A continuación se utilizó la librería Keras para realizar el aumento de datos en donde se procedió a hacer una rotación aleatoria de 45 grados así como el giro horizontal de la imagen.

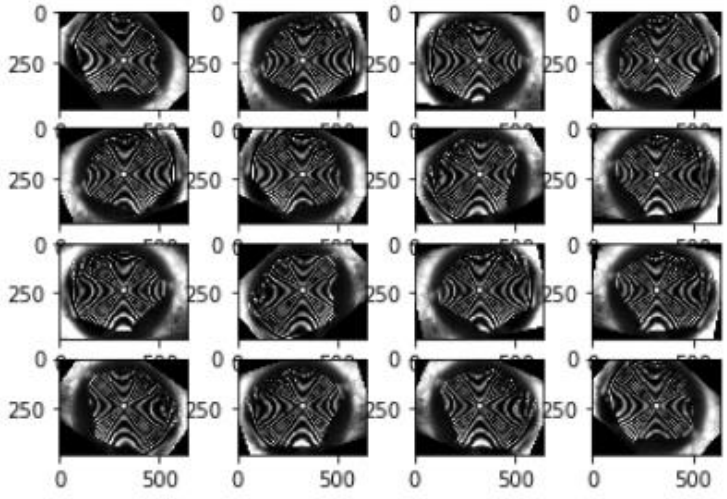


Figura 17. Demostración de la generación de imágenes utilizando la técnica Aumento de Datos con la librería Keras, elaboración propia

Al hacer uso de la técnica de aumento de datos generó un incremento significativo dando como resultado un total de 6609 imágenes que serán distribuidas en 3 conjuntos de datos para que el algoritmo pueda aprender correctamente, los cuales son:

Entrenamiento

Conjunto de datos que se utilizará para el aprendizaje del modelo propuesto.

Validación

Conjunto de datos que se utilizará para validar lo que el modelo ha aprendido en base al conjunto de datos de entrenamiento, logrando así que el modelo aprenda a clasificar de manera incorrecta.

Pruebas

Conjunto de datos que se utilizará para evaluar el rendimiento del modelo con la finalidad de validar si ha clasificado correctamente o no en base a los datos brindados.

La distribución de las categorías entrenamiento y validación forman parte de un solo grupo de datos, dado que la cantidad de imágenes que cuentan es demasiado alta como para realizar una adecuación manualmente es por ello que se ha realizado una distribución de 70% y 30% respectivamente mientras que el conjunto de datos pruebas es independiente, la cantidad de imágenes para cada conjunto de datos que se utilizarán en la red neuronal convolucional es la siguiente:

Tabla 5

Distribución de datos

Entrenamiento	Validación	Prueba
3175	1361	2073

Nota: Distribución de datos, elaboración propia

De esta manera al contar con una mayor cantidad de imágenes a procesar se procederá a realizar una comparativa de diseños de redes neuronales convolucionales para medir el rendimiento y la precisión de la clasificación de imágenes.

Al realizar la generación de imágenes se realizó un pre-procesamiento a las imágenes mediante el uso de la librería OpenCV, dado que permite aplicar filtros para reducir el ruido o segmentos de la imagen que no son útiles lo que conlleva a ocasionar que el modelo no obtenga un buen rendimiento, los filtros a aplicar son los siguientes:

Difuminado Gaussiano

Permite aplicar un desenfoque a la imagen con la finalidad reducir el ruido para que un proceso posterior pueda continuar con el tratamiento.

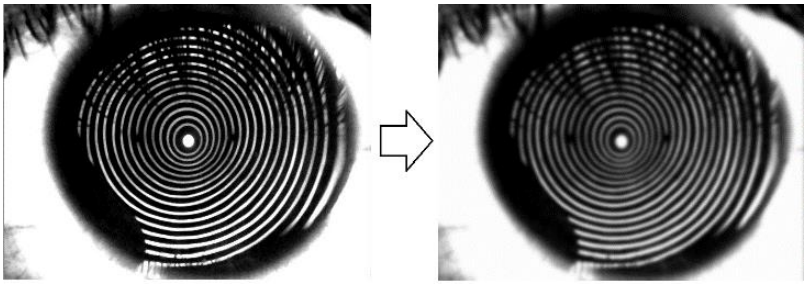


Figura 18. Procesamiento de imágenes con difuminado Gaussiano, elaboración propia

Detección de bordes

Permite identificar los bordes de la imagen con la finalidad de solo tratar las partes más significativas.

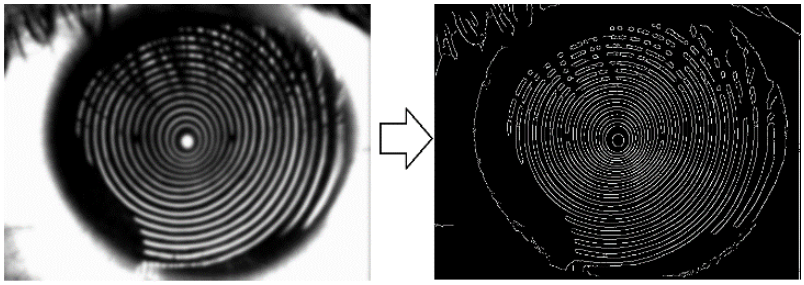


Figura 19. Procesamiento de imágenes con detección de bordes, elaboración propia

Las imágenes son procesadas en su tamaño original dado que posteriormente se redimensiona según el parámetro de entrada de los diseños.

Método de análisis

Continuando con el uso de la Metodología Fundamental para la Ciencia de Datos (Figura 9), la etapa de modelado es aquella donde se utilizarán los conjuntos de datos definidos en la etapa previa. Es por ello que el presente trabajo de investigación se propondrán 5 diseños con diversas arquitecturas de redes neuronales así como la dimensión que tendrá cada la imagen con la finalidad de validar cuál de ellos brindará el más alto nivel de precisión con respecto al aprendizaje y prueba de la red neuronal, los diseños planteados para el presente trabajo de investigación son los siguientes:

Tabla 6

Propuestas de diseños

Diseños	Pre procesamiento	Entrada	Modelo	Híper parámetros
Diseño 1	Difuminado Gaussiano Detección de bordes Escalado de características	60 x 80 x 1	Convolución: 3 x 3 kernel, 16 filtros y holgura en bordes Activación: ReLU MaximaAgrupación: 2 x2 kernel Dropout: 25% Convolución: 3 x 3 kernel y 32 filtros Activación: ReLU MaximaAgrupación: 2 x2 kernel Dropout: 25% Convolución: 3 x 3 kernel y 64 filtros Activación: ReLU MaximaAgrupación: 2 x2 kernel Dropout: 25% TotalmenteConectada: 128 Activación: ReLU Dropout: 50% TotalmenteConectada: 2 Activación: Softmax	Épocas: 100 Porcentaje de aprendizaje: 0.0001 Tamaño de lote: 32 Optimizador: Adam Detener cuando no haya mejora en la función de perdida
Diseño 2	Difuminado Gaussiano Detección de bordes Escalado de características	60 x 80 x 1	Convolución: 5x 5 kernel, 24 filtros y holgura en bordes Activación: ReLU MaximaAgrupación: 2 x 2 kernel Convolución: 5 x 5 kernel y 48 filtros Activación: ReLU MaximaAgrupación: 2 x 2 kernel Convolución: 5 x 5 kernel y 64 fltros Activación: ReLU MaximaAgrupación: 2 x2 filtro TotalmenteConectada: 128 Activación: ReLU TotalmenteConectada: 2 Activación: Softmax	Épocas: 200 Porcentaje de aprendizaje: 0.0001 Tamaño de lote: 32 Optimizador: Adam Detener cuando no sea mejora en la función de perdida
Diseño 3	Escalado de características	60 x 80 x 1	Convolución: 5 x 5 kernel, 16 filtros y holgura en bordes Activación: ReLU Normalización MaximaAgrupación: 2 x 2 kernel Convolución: 5 x 5 kernel y 32 filtros Activación: ReLU Normalización MaximaAgrupación: 2 x 2 kernel Convolución: 5 x 5 kernel y 64 fltros Activación: ReLU Normalización MaximaAgrupación: 2 x 2 filtro	Épocas: 200 Porcentaje de aprendizaje: 0.0001 Tamaño de lote: 32 Optimizador: Adam Detener cuando no haya mejora en la función de perdida

			TotalmenteConectada: 2 Activación: Softmax	
Diseño 4	Escalado de características	30 x 40 x 1	Convolución: 3 x 3 kernel, 16 filtros y holgura en bordes Activación: ReLU MaximaAgrupación: 2 x 2 kernel Convolución: 3 x 3 kernel y 32 filtros Activación: ReLU MaximaAgrupación: 2 x 2 kernel Convolución: 5 x 5 kernel y 64 fltros Activación: ReLU MaximaAgrupación: 2 x 2 filtro TotalmenteConectada: 128 Activación: ReLU TotalmenteConectada: 2 Activación: Softmax	Épocas: 200 Porcentaje de aprendizaje: 0.0001 Tamaño de lote: 32 Optimizador: Adam Detener cuando no haya mejora en la función de perdida
Diseño 5	Escalado de características	180 x240 x1	Convolución: 3 x 3 kernel, 24 filtro y holgura en bordes Activación: ReLU MaximaAgrupación: 2 x 2 kernel Dropout: 25% Convolución: 3 x 3 kernel, 48 filtro y holgura en bordes Activación: ReLU MaximaAgrupación: 2 x 2 kernel Dropout: 25 Convolución: 3 x 3 kernel, 64 filtro y holgura en bordes Activación: ReLU MaximaAgrupación: 2 x2 kernel Dropout: 25 TotalmenteConectada: 1024 Normalización Activación: ReLU Dropout: 30% TotalmenteConectada: 128 Normalización Activación: ReLU Dropout: 50% TotalmenteConectada: 2 Activación: Softmax	Epocas: 100 Porcentaje de aprendizaje: 0.0001 Tamaño de lote: 32 Optimizador: Adam Detener cuando no haya mejora en la función de perdida

***Nota**:* Propuesta de diseños, elaboración propia

La arquitectura de los diseños propuestos se puede encontrar en los Anexos.

RESULTADOS

Situación Actual

En la clínica Exilaser se utiliza el equipo especializado OPD Scan III para identificar diferentes enfermedades siendo una de ellas el queratocono mediante la funcionalidad de topógrafo corneal con el que el equipo cuenta, el procedimiento para identificar la enfermedad consiste en medir el grosor de la córnea. El motivo principal por el cual se opta por realizar este procedimiento se debe a que el queratocono degrada el grosor por lo que tiende a deformarse en una forma de cono.

Una de las particularidades del equipo OPD Scan III es que permite identificar las partes más significativas de los anillos y regiones donde se presentan deformidades, dando como resultado un valor expresado en valor porcentual que debe superar el umbral mínimo de 70% para considerar que el ojo presenta de queratocono. Partiendo de este apartado el equipo puede generar un valor menor al umbral mínimo y se opta por dar un seguimiento periódico para ver la evolución del ojo analizado y en algunos casos optar por tratamientos preventivos bajo el criterio del especialista.

Desarrollo de la propuesta

El presente trabajo de investigación tiene como objetivo realizar una clasificación correcta de la enfermedad queratocono que conlleve a realizar una detección temprana en base al histórico de los pacientes que presentan la enfermedad con la finalidad de realizar los tratamientos en una etapa inicial.

Es por ello que se han propuesto diversos diseños que se han ejecutado en Amazon Elastic Compute Cloud (EC2), plataforma que proporciona recurso computacional en la nube, que permite realizar la implementación de manera más eficiente a comparación de un ordenador propio. Debido a la escalabilidad que ofrece en cuanto a la construcción de la máquina virtual que uno necesite, las características de la máquina virtual adquirida son:

Tabla 7

Características de la máquina virtual en EC2

Conceptos	Características
Imagen	Deep Learning AMI (Amazon Linux) Version 28.1 - 64 bits
Instancia	4 VCPU - 2.7 GHz - memoria 61 GiB p2.xlarge GPU - Nvidia Tesla K80 Accelerators
Almacenamiento	100 GiB – General Purpose SSD (gp2)
Costo	Suscripción mensual “Developer”– $ 29 Costo del uso de máquina virtual por hora – $ 0.271

Nota: Características de la máquina virtual adquirida en AWS EC2, elaboración propia

Las características adquiridas en EC2 son suficientes para realizar el procesamiento de las imágenes de acuerdo a la dimensión establecida de cada diseño en la Tabla 6. El uso de una GPU en lugar de CPU marca una diferencia significativa en cuanto al tiempo de ejecución del aprendizaje, validación y pruebas de los diseños planteados, esto se debe a que una GPU permite realizar múltiples tareas de manera simultánea mientras que una CPU maneja las tareas de manera secuencial, esto puede generar que un modelo que se ejecute en 2 horas utilizando una GPU se convierta en 20 horas en una CPU.

Para realizar la configuración de la máquina virtual adquirida en Amazon EC2, primero se debe de tener configurado los siguientes componentes:

Key Pair

Componente que genera una llave publica alojada en Amazon Web Service y una llave privada que permitirá establecer la comunicación a la máquina virtual a través de protocolo SSH.

Security Group

Componente que permite o deniega el acceso a la máquina virtual, realiza el trabajo de un firewall virtual.

Una vez configurado los componentes, se procede a dar permiso de solo lectura a la llave privada generada utilizando el comando “chmod 400”.

Figura 20. Modificación de permisos, elaboración propia

Solo si la llave privada tiene permisos de lectura se puede realizar la conexión SSH a la maquia virtual de la siguiente manera:

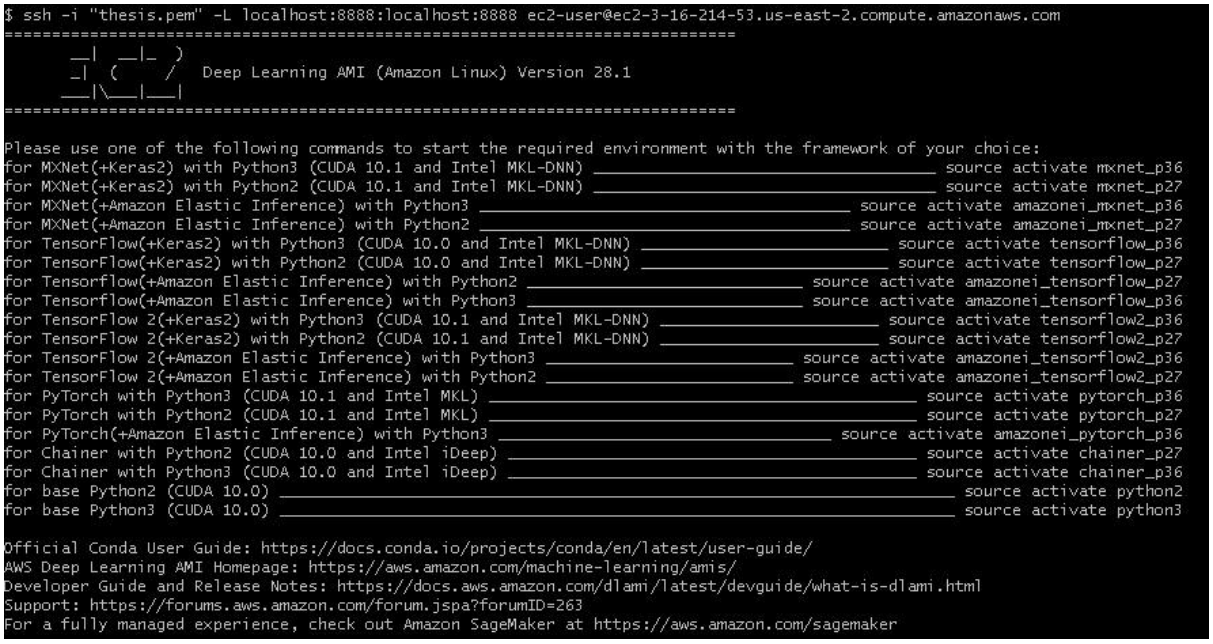


Figura 21. Conexión SSH, elaboración propia

La exposición del puerto remoto 8888 a través del puerto 8888 de la máquina donde se ha realizado la conexión, se debe a que se utilizará la aplicación Jupyter Notebook para compilar código Python. El puerto por defecto que utiliza la aplicación para ejecutarse es 8888.

Seguidamente se realiza la ejecución del comando “source activate tensorflow_p36”, el cual instala las configuraciones bases que proporciona Amazon EC2.

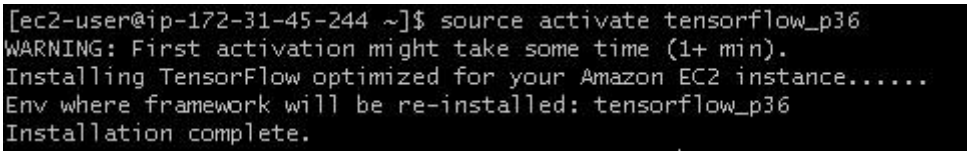


Figura 22. Instalación base de Amazon EC2, elaboración propia

Una característica resaltante de las instalaciones bases que se hicieron previamente, es el uso de “tmux”, el cual es un programa que permite tener múltiples sesiones virtuales a través de una sola terminal dando como resultado que se pueda activar Jupyter Notebook, visualizar el consumo del GPU y ejecutar comandos propios de Linux o relacionados a Python.

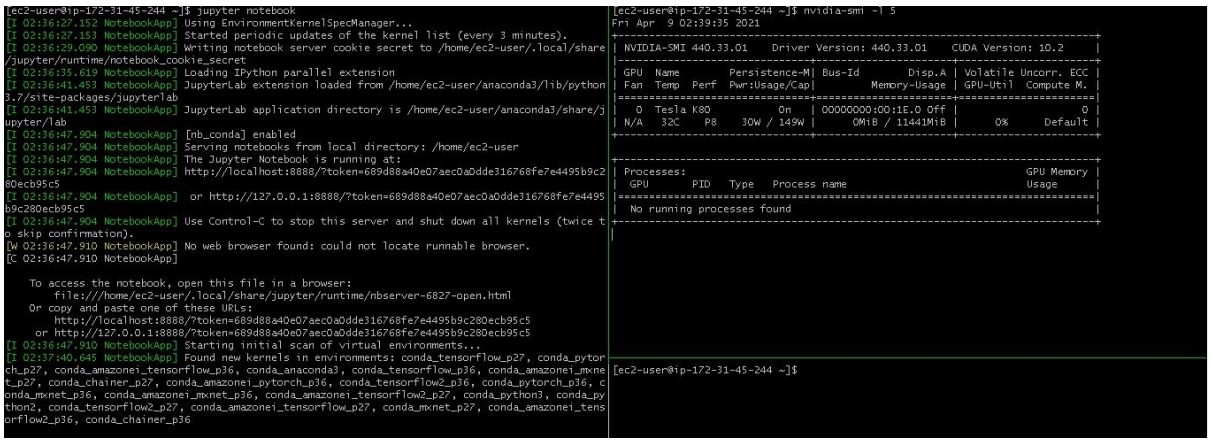


Figura 23. Aplicación de tmux, elaboración propia

Al haber ejecutado la aplicación Jupyter Notebook se genera una URL con un código de acceso, el cual permite el acceso a la aplicación que se expuso localmente al iniciar la conexión SSH.

Figura 24. Acceso Jupyter Notebook de Amazon EC2, elaboración propia

Se procede a realizar la carga del archivo donde se ejecutará el código Python así como el archivo comprimido que contiene las imágenes generadas previamente.

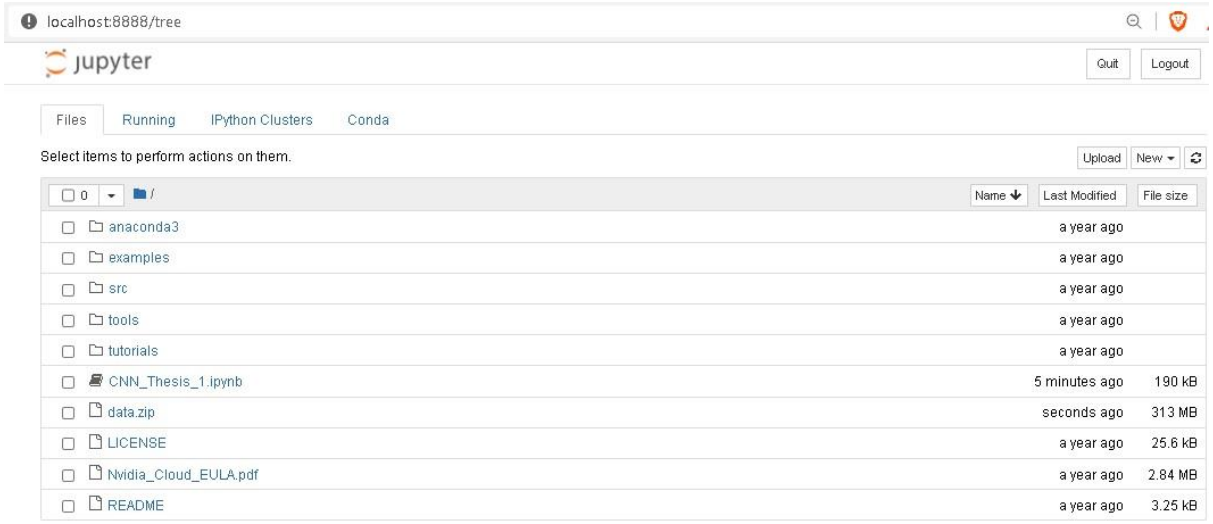


Figura 25. Carga de archivos a Jupyter Notebook, elaboración propia

Para concluir la configuración del entorno se debe de instalar las siguientes librerías a través de la terminal o de Jupyter Notebook, con la finalidad de no generar errores por fallos de dependencias. En esta ocasión se utilizó Jupyter Notebook para la instalación, la cual solicita un reinicio del Kernel, el cual es el encargado de ejecutar los códigos por parte del usuario.

```
In [1]: pip install opencv-python

Collecting opencv-python
  Using cached opencv_python-4.5.1.48-cp37-cp37m-manylinux2014_x86_64.whl (50.4 MB)
Requirement already satisfied: numpy>=1.14.5 in ./anaconda3/lib/python3.7/site-packages (from opencv-python) (1.19.5)
Installing collected packages: opencv-python
Successfully installed opencv-python-4.5.1.48
Note: you may need to restart the kernel to use updated packages.

In [2]: pip install Keras

Collecting Keras
  Downloading Keras-2.4.3-py2.py3-none-any.whl (36 kB)
Requirement already satisfied: numpy>=1.9.1 in ./anaconda3/lib/python3.7/site-packages (from Keras) (1.18.1)
Requirement already satisfied: pyyaml in ./anaconda3/lib/python3.7/site-packages (from Keras) (5.3.1)
Requirement already satisfied: scipy>=0.14 in ./anaconda3/lib/python3.7/site-packages (from Keras) (1.4.1)
Requirement already satisfied: h5py in ./anaconda3/lib/python3.7/site-packages (from Keras) (2.10.0)
Requirement already satisfied: six in ./anaconda3/lib/python3.7/site-packages (from h5py->Keras) (1.14.0)
Installing collected packages: Keras
Successfully installed Keras-2.4.3
Note: you may need to restart the kernel to use updated packages.

In [3]: pip install tensorflow

Installing collected packages: six, absl-py, termcolor, flatbuffers, numpy, keras-preprocessing, grpcio, wheel, markdown, o
authlib, requests-oauthlib, pyasn1, rsa, cachetools, pyasn1-modules, google-auth, google-auth-oauthlib, tensorboard-plugin-
wit, tensorboard, gast, astunparse, opt-einsum, google-pasta, tensorflow-estimator, tensorflow
  Attempting uninstall: six
    Found existing installation: six 1.14.0
    Uninstalling six-1.14.0:
      Successfully uninstalled six-1.14.0
  Attempting uninstall: numpy
    Found existing installation: numpy 1.18.1
    Uninstalling numpy-1.18.1:
      Successfully uninstalled numpy-1.18.1
  Attempting uninstall: wheel
    Found existing installation: wheel 0.34.2
    Uninstalling wheel-0.34.2:
      Successfully uninstalled wheel-0.34.2
Successfully installed absl-py-0.12.0 astunparse-1.6.3 cachetools-4.2.1 flatbuffers-1.12 gast-0.3.3 google-auth-1.28.1 goog
le-auth-oauthlib-0.4.4 google-pasta-0.2.0 grpcio-1.32.0 keras-preprocessing-1.1.2 markdown-3.3.4 numpy-1.19.5 oauthlib-3.1.
0 opt-einsum-3.3.0 pyasn1-0.4.8 pyasn1-modules-0.2.8 requests-oauthlib-1.3.0 rsa-4.7.2 six-1.15.0 tensorboard-2.4.1 tensorb
oard-plugin-wit-1.8.0 tensorflow-2.4.1 tensorflow-estimator-2.4.0 termcolor-1.1.0 wheel-0.36.2
Note: you may need to restart the kernel to use updated packages.
```

Figura 26. Instalación librerías adicionales en Jupyter Notebook, elaboración propia

El flujo de trabajo del presente trabajo de investigación se puede representar de la siguiente manera:

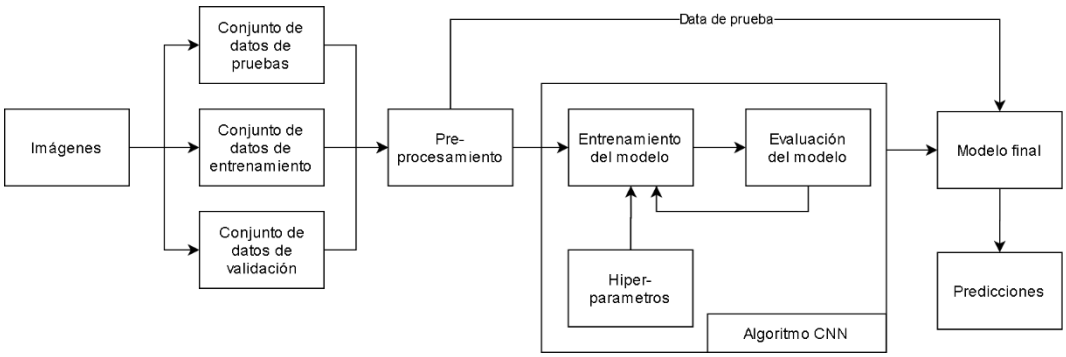


Figura 27. Flujo de trabajo de los diseños propuestos, elaboración propia

A continuación se presentará los resultados obtenido de los modelos al realizar la validación de la data de prueba así como el tiempo de ejecución:

Tabla 8

Resultados de modelos propuestos

	Data de prueba	**Tiempo de ejecución**	**Cantidad de capas**
Diseño 1	92.04%	97 minutos	17
Diseño 2	89.00%	43 minutos	13
Diseño 3	84.90%	18 minutos	14
Diseño 4	73.85%	4 minutos	13
Diseño 5	78.19%	7 minutos	22

Nota: Resultados de los modelos propuestos, elaboración propia

En base a los resultados presentados se decide rechazar las hipótesis específicas presentadas en la Tabla 3 del capítulo III, debido a que se evidencia que el algoritmo de redes neuronales convolucionales ha realizado la correcta clasificación de las imágenes debido al uso adecuado de capas en el diseño de la arquitectura, logrando de esta manera un resultado de 92.04% al realizar la clasificación de ojos sanos y aquellos que padecen de queratocono, una de las partes más significativas de los resultados presentados es el tiempo de ejecución el cual en el diseño 1 no se ha visto afectado durante el total de épocas definidas logrando de esta manera diferenciarse de los resultados de los demás diseños.

Respecto al uso de la Metodología Fundamental para la Ciencia de Datos (Figura 9), el resultado obtenido hace referencia a la etapa de evaluación donde se utiliza el conjunto de datos de pruebas para validar el rendimiento del modelo propuesto, siendo en este caso el Diseño 1 aquel que cumplió con el enfoque esperado.

La arquitectura del Diseño 1 será presentada de manera gráfica para visualizar el procesamiento a través de cada capa implementada así como el tamaño asociado a cada una de ellas:

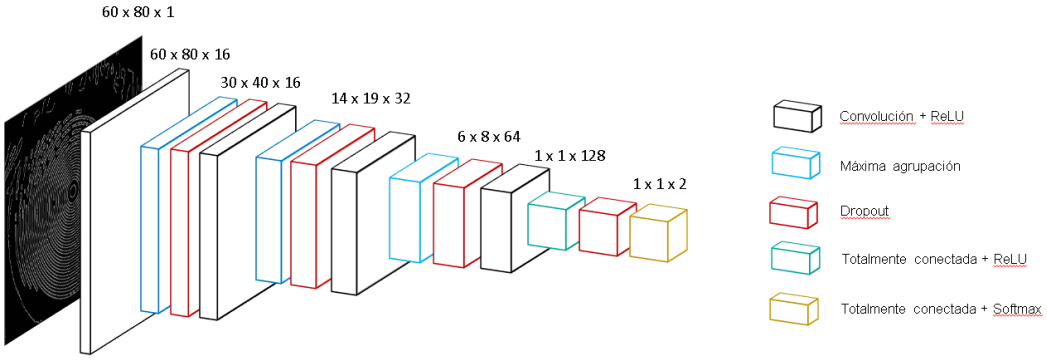


Figura 28. Arquitectura de red neuronal convolucional del Diseño 1, elaboración propia

Como se puede observar la disminución del tamaño se realiza progresivamente al ejecutarse la capa “máxima agrupación”, de igual manera en el uso de las “capa totalmente conectada”, las cuales se encuentran en la parte final de la arquitectura ya que permiten conectar todas la neuronas y dar como resultado una nueva neurona, en donde se indicará la clasificación que se esté realizando. Es por ello que el procesamiento de las imágenes que se realiza en cada capa es la siguiente:

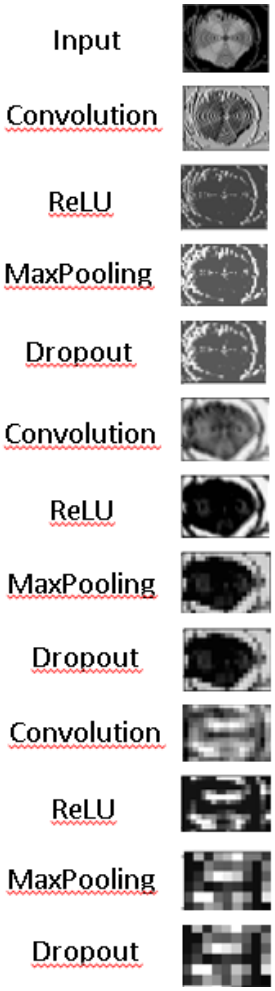


Figura 29. Procesamiento de la imagen en la red neuronal convolucional, elaboración propia

La ultimas capas relacionadas a “totalmente conectadas” no se presentan en la Figura 11 debido a que son las encargadas realizar la clasificación en base a la neuronas que se le asignaron. Al entender cómo se realiza el procesamiento de las imágenes se presentará el resultado del Diseño 1 con imágenes de prueba.

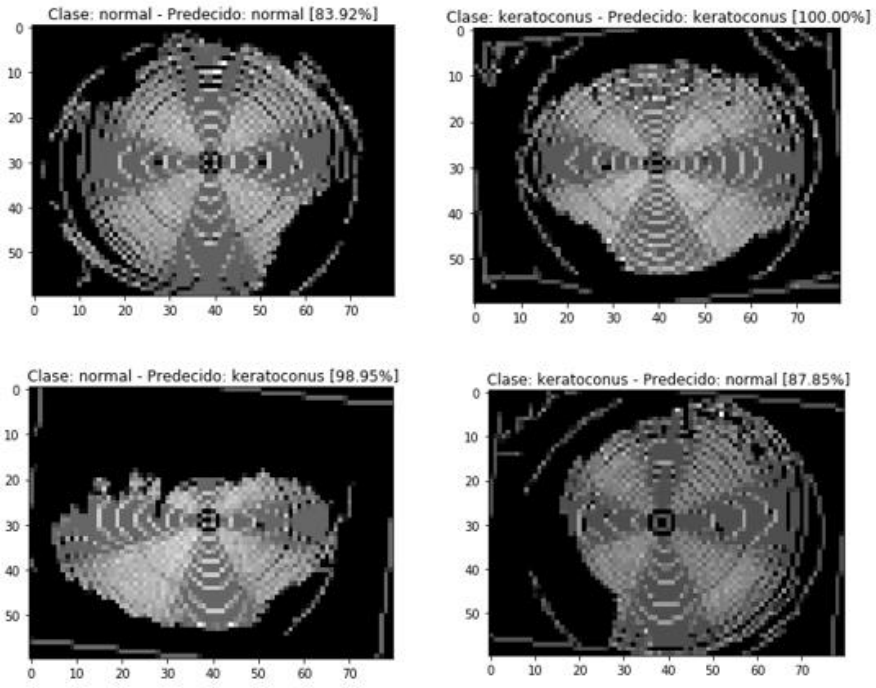


Figura 30.Resultados de predecir la data de prueba, elaboración propia

El diseño 1 realizó la correcta clasificación de 1908 imágenes de un total de 2073, el detalle de cada cantidad de ojos con queratocono y normal se detalla de la siguiente manera:

Tabla 9

Cantidad de imágenes clasificadas por categoría

Categoría	**Imágenes clasificadas correctamente**	**Total de imágenes**
Normal	968	1035
Queratocono	940	1038

Nota: Cantidad de imágenes clasificadas por categoría, elaboración propia

A continuación se realizará la matriz de confusión para evaluar el algoritmo implementado:

Tabla 10

Matriz de confusión

	Queratocono (Positivo)	Normal (Negativo)	Total
Queratocono (Positivo)	940	98	1038
Normal (Negativo)	67	968	1035
Total	1007	1066	2073

Nota: Matriz de confusión, elaboración propia

En base a lo presentado en la Tabla 10 se da a conocer los siguientes enunciados:

Positivo + Positivo = Verdadero Positivo (VP)

Positivo + Negativo = Falso Negativo (FN)

Negativo + Positivo = Falso Positivo (FP)

Negativo + Negativo = Verdadero Negativo (VN)

Conociendo las combinaciones generados por la matriz de confusión se procederá generar las métricas del algoritmo:

Exactitud

Indica el número de clasificaciones correctas realizadas.

$$exactitud = \frac{VP + VN}{VP + FN + FP + VN}$$

$$exactitud = \frac{940 + 968}{940 + 968 + 67 + 98}$$

$$exactitud = 92.04\%$$

El valor generado en esta métrica es el mismo obtenido en la Tabla 8 del presente capítulo.

Sensibilidad

Indica el número de verdadero positivos que el algoritmo ha clasificado.

$$sensibilidad = \frac{VP}{VP + FN}$$

$$sensibilidad = \frac{940}{940 + 98}$$

$$sensibilidad = 90.56\%$$

Puntaje F1

Indica el valor en conjunto obtenido de la sensibilidad y exactitud.

$$f1 = 2 * \frac{exactitud * sensibilidad}{exactitud + sensibilidad}$$

$$f1 = 2 * \frac{92.04 * 90.56}{92.04 + 90.56}$$

$$f1 = 91.29\%$$

En relación con la problemática expuesta sobre los diagnósticos de queratocono aplicando Deep Learning, en primer lugar, en conjunto con el experto se identificaron los ojos sanos y aquellos que padecen de queratocono, para obtener el 100% de casos correctamente clasificados. Posteriormente se busca lograr que se realicen las evaluaciones de la solución propuesta y de este modo se pueda cumplir con el criterio de aceptación establecido, el cual fue 90% de casos correctamente clasificados.

Los resultados presentados en la Tabla 9, 10 y los indicadores sirven de apoyo para rechazar la hipótesis nula general debido a que se demuestra que la aplicación de Deep Learning clasifica de manera correcta y con un alto nivel de precisión los casos de queratocono así como superar el umbral de criterio de aceptación propuesto.

Las últimas etapas de la Metodología Fundamental para la Ciencia de Datos, implementación y retroalimentación, están asociados a puesta en marcha del entorno productivo y recopilar los resultados obtenidos del modelo para reforzar el aprendizaje. Estas dos etapas no han sido contempladas en el presente trabajo de investigación no obstante se podría implementar en un trabajo futuro, donde se podría optar por exponer el modelo a través de una API para que sea consumido por aplicaciones web o dispositivos móviles con la finalidad de que el usuario pueda brindar comentarios de si la clasificación ha sido la correcta o no.

DISCUSIÓN

De acuerdo a Rakib (2018), quién implementó una red neuronal convolucional para la clasificación de imágenes de rayos X, afirma que el uso de este algoritmo puede realizar la clasificación de imágenes de manera correcta dando como resultado valores aceptables pero que se puede mejorar si se obtuviera más imágenes. De igual manera en este presente trabajo de investigación se utilizaron técnicas para incrementar la cantidad de imágenes y obtener valores altos en la clasificación de los diversos modelos propuestos, por ello concuerdo con Rakib de ser necesario una gran cantidad de imágenes para procesar.

En el trabajo de investigación de Nilson y Velic (2018), se indica que el método de transferencia de aprendizaje es utilizado para realizar un pre-entrenamiento de la red neuronal, lo que conlleva a tener mejores resultados y asimismo el uso de hiperparámetros que pueden mejorar el rendimiento de la red neuronal. En base a los resultados obtenidos en la presente tesis se concuerda con Nilson y Velic sobre el uso de hiperparámetros que permiten al algoritmo realizar ajustes necesarios para mejorar el rendimiento e inclusive la precisión.

Vicenzi (2017), explica que el uso adecuado de los filtros puede mejorar el rendimiento de la red neuronal durante la fase de entrenamiento dado que existe el caso de sufrir de sobreajuste u overfitting, lo que limita el aprendizaje y puede generar resultados con un valor bajo. Concuerdo con lo indicado por Vicenzci respecto al uso correcto de los filtros en una red neuronal dado que puede limitar el aprendizaje, es por ello que en el presente trabajo de investigación se utilizó la capa de dropout con la finalidad de no sufrir de overfitting durante el aprendizaje.

De acuerdo a Seeböck (2015), quien realizó pruebas de clasificación de imágenes utilizando diversos algoritmos para validar el rendimiento siendo uno de ellos el algoritmo K-Nearest Neighbor aplicando a tomografías computarizadas, dando como resultado una precisión baja y a dígitos escritos a mano, donde se obtuvo un porcentaje alto, partiendo de esto Seeböck hace hincapié en el uso adecuado de los algoritmos así como los hiperpárametros a utilizar. Por ello en el presente trabajo de investigación se realizó un análisis comparativo de los algoritmos para elegir el que mejor se adecuaba a la necesidad de clasificar imagenes siendo este la de redes neuronales convolucionales por lo que se concuerda con lo indicado Seeböck.

CONCLUSIONES

De los objetivos planteados así como los resultados que se obtuvieron en los capítulos III y V permitieron llegar a las siguientes conclusiones:

El uso de redes neuronales convolucionales permitieron validar la correcta clasificación de ojos que presentan queratocono, el cual se puede apreciar en la Tabla 9 donde de un total 1038 imágenes se han clasificado correctamente 940. De este modo se puede concluir que el diagnóstico de queratocono utilizando un deep learning o aprendizaje profundo ha sido satisfactorio, logrando cumplir con lo propuesto en el presente trabajo de investigación.

En base a los trabajos de investigación presentados en el marco teórico así como en el estado de arte se puede concluir que el algoritmo predilecto para realizar una clasificación de imágenes es el de redes neuronales convoluciones, de igual forma en el presente trabajo de investigación se realizó un análisis comparativo sobre los algoritmos a utilizar, dando como resultado la aplicación del algoritmo de redes neuronales convolucionales, el cual puede ser encontrado en la Tabla 2 ubicado en el capítulo IV.

Se concluye que los diseños presentados en el trabajo de investigación han generado diversos resultados, de los cuales el diseño 1 fue construido teniendo en cuenta la mitigación del overfitting así como un pre-procesamiento correcto que permitieron al algoritmo continuar con el aprendizaje sin ser detenido durante las épocas definidas, el cual puede ser encontrado en el Anexo 5.

Se propusieron 5 diseños con diferentes arquitecturas, procesamiento de imágenes e hiperparámetros para identificar cuál de ellos generará un el mayor nivel de precisión, dando como resultado que el diseño 1 obtuvo un 92.04% de precisión en la clasificación de imágenes, el diseño se encuentra detallado en la Tabla 8 del capítulo V.

En base a los resultados obtenidos se ha evidenciado que el tiempo de ejecución del diseño 1 presenta un mayor tiempo a comparación de los demás diseños debido a que el proceso de aprendizaje no se ha detenido al mantener mejora constante en el valor de la función de activación del conjunto de datos de validación, el tiempo de ejecución se puede encontrar en la Tabla 8 del capítulo V.

RECOMENDACIONES

Se recomienda utilizar el algoritmo de redes neuronales convolucionales para la clasificación de imágenes, dado que se ha demostrado que en el presente trabajo de investigación se ha obtenido un alto nivel de precisión utilizando diversas arquitecturas así como los hiperparámetros del mismo modo es necesario tener una guía que permita tener una visión más clara de cómo dar solución a un problema, para ello se recomienda utilizar metodologías enfocadas a problemas de ciencia de datos, tal como se usó en la presente tesis donde la metodología aplicada fue la Metodología Fundamental para la Ciencia de Datos

Así como en el presente trabajo de investigación se realizó un análisis comparativo de los algoritmos de deep learning además de utilizar una buena referencia en los antecedentes como en el estado de arte, se recomienda realizar la investigación necesaria del algoritmo que se desee utilizar como también la aplicación de ello dado que no todos los algoritmos tiene la misma aplicación, un ejemplo claro es el uso de redes neuronales convoluciones cuya aplicación está enfocada en la clasificación de imágenes.

Es conveniente tener referencias del estado del arte al problema que se necesita dar solución, dado que los diseños nacen a partir de ello con algunas variaciones dependiendo de los parámetros de entrada, ya sea que se necesite utilizar capas específicas como Dropout o en algunos casos agregar diversas capas totalmente conectadas con la finalidad de diseñar una arquitectura robusta.

Se sugiere para obtener un alto nivel de precisión es recomendable utilizar diversas técnicas que permitan al algoritmo mitigar los problemas como overfitting como pérdida de peso (weight decay) o como se aplicó en el presente trabajo de investigación la capa de Dropout y normalización por lotes (batch normalization), así mismo para el problema de underfitting se optaría por la técnica de aumento de datos aplicando diversos filtros para que la red neuronal genere un valor con alto nivel de precisión.

Es recomendable hacer un análisis respecto a si el algoritmo debe tener un rápido aprendizaje o si necesita aprender correctamente, debido a que existen diseños de redes neuronales convolucionales con una gran complejidad, tales como VGG16, VGG19, Resnet-50 entre otros más que pueden dar buenos resultados al utilizarlos pero a coste de la rapidez.

REFERENCIAS

Aguirre, N. C., & Velic, M. (2018). Classification of Ulcer Images Using Convolutional Neural Networks (Tesis de maestría). Recuperada de http://publications.lib.chalmers.se/records/fulltext/255746/255746.pdf

AUNA (s.f). Diagnóstico por Imágenes [AUNA]. Recuperado de http://auna.pe/servicios/diagnostico-por-imagenes/

Ayuque, K. J. (2016). Diseño de un sistema de clasificación de señales de tránsito vehicular utilizando redes neuronales convolucionales (Tesis de titulación). Recuperada de http://repositorio.usil.edu.pe/bitstream/USIL/2480/1/2016_Ayuque_Dise%c3%b1o_de_un_sistema_de_clasificacion.pdf

Calvo, D. (2018, 07 de Diciembre). Función de activación – Redes neuronales [Diego Calvo]. Recuperado de https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/

Clínica Internacional (2018, 03 de Julio). 7 beneficios de la digitalización en nuestros procesos médicos [Clínica Internacional]. Recuperado de https://www.clinicainternacional.com.pe/blog/beneficios-digitalizacion-procesos-medicos/

Clínica Rementeria (2015, 08 de Octubre). Queratocono [Clínica Rementeria]. Recuperado de: https://www.clinicarementeria.es/tratamientos/queratocono-2

Curioso, W. H. (2014). eSalud en Perú: implementación de políticas para el fortalecimiento de sistemas de información en salud (Volumen 35, N° 5-6). Washington, D.C: Pan American Health Organization

Frandsen, A. J. (2016). Machine Learning for Disease Prediction (Tesis de maestría). Recuperada de https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=6974&context=etd

Frankenfield J. (2019, 13 de Junio). Artificial Intelligence (AI) [Investopedia]. Recuperado de https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp

González A. (2018, 30 de Julio). Conceptos básicos de Machine Learning [CleverData]. Recuperado de https://cleverdata.io/conceptos-basicos-machine-learning/

Gupta D. (2017, 23 de Octubre). Fundamentals of Deep Learning – Activation Functions and When to Use Them? [Analytics Vidhya]. Recuperado de https://www.analyticsvidhya.com/blog/2017/10/fundamentals-deep-learning-activation-functions-when-to-use-them/

Hernández, R., Fernández, C., & Baptista, P. (2014). Metodología de la investigación (Sexta edición). México D.F. Mc Graw Hill

Hossain, K. R. (2018). Deep Learning Based Medical X-ray Image Recognition and Classification (Tesis de bachiller). Recuperada de http://dspace.bracu.ac.bd/xmlui/bitstream/handle/10361/11426/14301110_CSE.pdf?sequence=1&isAllowed=y

Ibertech (2016, 01 de Junio). Analítica descriptiva, predictiva y prescriptiva [Ibertech]. Recuperado de https://www.ibertech.org/analitica-descriptiva-predictiva-y-prescriptiva/

Instituto Nacional de Oftalmología (INO). (2018). Plan operativo institucional. Recuperado de http://www.ino.gob.pe/transparencia/planeamiento/poi/POI%202018.pdf

Jiménez G. (2019). Deep Learning for Semantic Segmentation versus Classification in Computational Pathology: Application to mitosis analysis in Breast Cancer grading. Recuperada de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/13969/JIMENEZ_GARAY_GABRIEL_DEEP_LEARNING_SEMANTIC.pdf?sequence=1&isAllowed=y

Johansson P. (2019). Incremental Learning of Deep Convolutional Neural Networks for Tumour Classification in Pathology Images. Recuperada de http://liu.diva-portal.org/smash/get/diva2:1331382/FULLTEXT01.pdf

Lavado, L. (s.f). Cornea [Universidad Nacional Mayor de San Marcos]. Recuperado de http://sisbib.unmsm.edu.pe/bibvirtual/libros/medicina/cirugia/tomo_iv/cornea.htm

Mayo Clinic (2019, 13 de Setiembre). Queratocono [Mayo Clinic]. Recuperado de: https://www.mayoclinic.org/es-es/diseases-conditions/keratoconus/diagnosis-treatment/drc-20351357

Ministerio de Salud (2018, 19 de Diciembre). Promulgan Ley que establece la conformación y el funcionamiento de las Redes Integradas de Salud (RIS). Recuperado de

https://www.gob.pe/institucion/minsa/noticias/23830-promulgan-ley-que-establece-la-conformacion-y-el-funcionamiento-de-las-redes-integradas-de-salud-ris

Ministerio de Salud (2019, 03 de Enero). Minsa publica proyecto de Estrategia de Gobierno Digital del Sector Salud 2019 - 2022. Recuperado de https://www.gob.pe/institucion/minsa/noticias/24257-minsa-publica-proyecto-de-estrategia-de-gobierno-digital-del-sector-salud-2019-2022

Moolayil, J. (2019). Learn Keras for Deep Neural Networks (Primera edición). New York. Apress Media

Ordoñez, E. (2020). Deep Learning para la visión artificial e identificación del personal administrativo y docente de la Universidad Nacional Micaela Bastidas de Apurímac 2018 (Tesis de doctorado). Recuperada de http://repositorio.unap.edu.pe/bitstream/handle/UNAP/13523/Erech_Ordoñez_Ramos.pdf?sequence=3&isAllowed=y

Pedrycz, W. & Chen, S. (2020). Deep Learning: Concepts and Architectures. Springer.

Rollins, J. B. (2015) Metodología Fundamental para la Ciencia de Datos. Recuperado de https://www.ibm.com/downloads/cas/WKK9DX51

RPP (2019, 22 de Enero). Tecnología médica que cambia la vida de los pacientes [RRP]. Recuperado de https://rpp.pe/campanas/contenido-patrocinado/tecnologia-medica-que-cambia-la-vida-de-los-pacientes-noticia-1175179

Sánchez L., Álvarez P., Benavides P., Sánchez H. & Zambrano D. (2018). El queratocono, su diagnóstico y manejo. Una revisión bibliográfica, Enfermería Investiga, Investigación, Vinculación, Docencia y Gestión-Vol. 3 Sup.1, p. 1-8

Sánchez, J. (2013). Paradigmas de Investigación Educativa: de las leyes subyacentes a la modernidad reflexiva. Epistemología, 16, 4-5.

Sancho F. (2017, 23 de Setiembre). Introducción al Aprendizaje Automático [Fernando Sancho Caparrini]. Recuperado de http://www.cs.us.es/~fsancho/?e=75

Seeböck P. (2015). Deep Learning in Medical Image Analysis. Recuperada de https://www.cir.meduniwien.ac.at/assets/Uploads/Masterthesis-SeeboeckPhilipp-Version28-03-2015.pdf

Skalski, P. (2019, 12 de Abril). Gentle Dive into Math Behind Convolutional Neural Networks [Towards Data Science]. Recuperado de https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9

Tejeda, G. & Gonzales, R. (2020). Arquitectura de red neuronal convolucional para diagnóstico de cáncer de piel (Tesis de titulación). Recuperada de http://repositorio.utp.edu.pe/bitstream/UTP/3043/1/Giorzinio%20Tejada_Renzo%20Gonzales_Tesis_Titulo%20Profesional_2020.pdf

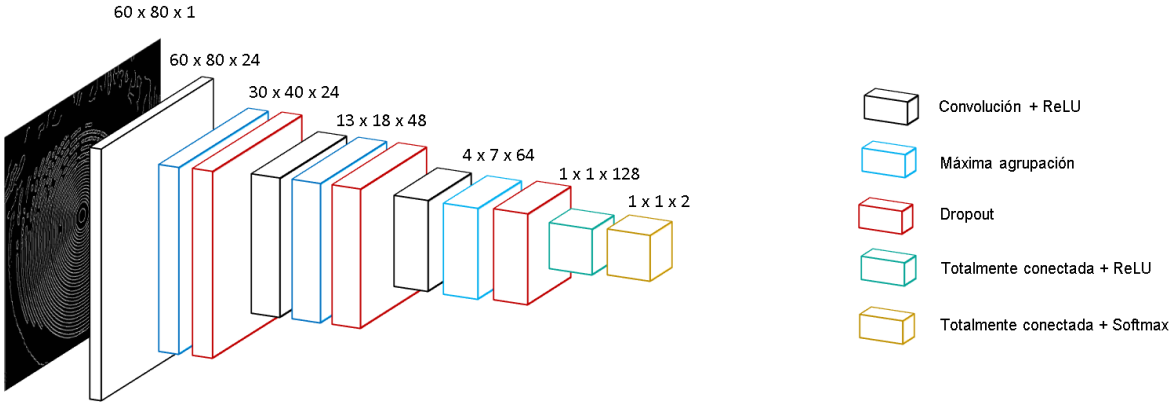
Vasilev, I., Slater, D., Spacagna, G., Roelants, P. & Zocca, V. (2019). Python Deep Learning: Exploring Deep Learning Techniques and Neural Network Architectures with PyTorch, Keras, and TensorFlowPython Deep Learning (Segunda 2 edicióned.). Birmingham. Packt Publishing

Verger E. (2017, 24 de Junio). ¿Qué partes tiene una neurona? ¿Cuál es su función? [CienciaToday]. Recuperado de https://cienciatoday.com/partes-neurona-funcion/

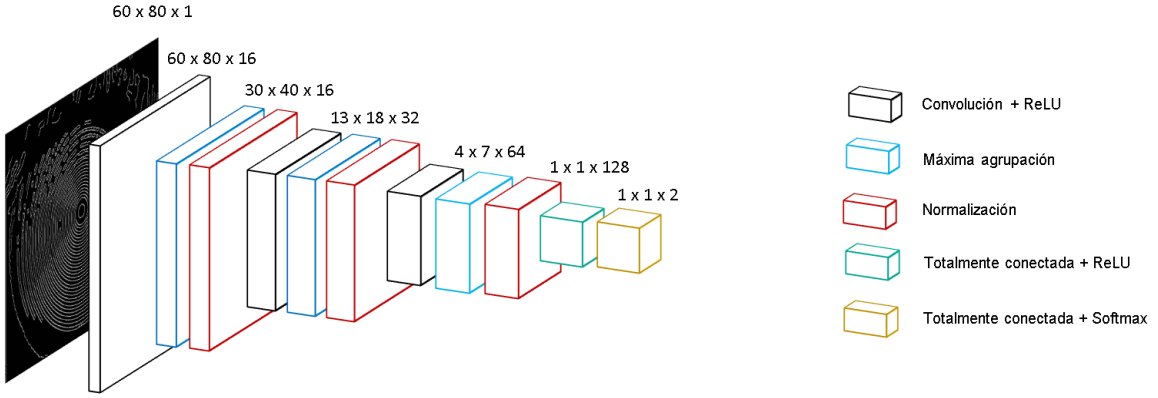
Vicenzi E. (2017). A deep learning approach to the analysis of retinal images (Tesis de titulación). Recuperada de http://tesi.cab.unipd.it/54600/1/enrico_vincenzi_tesi.pdf

ANEXOS

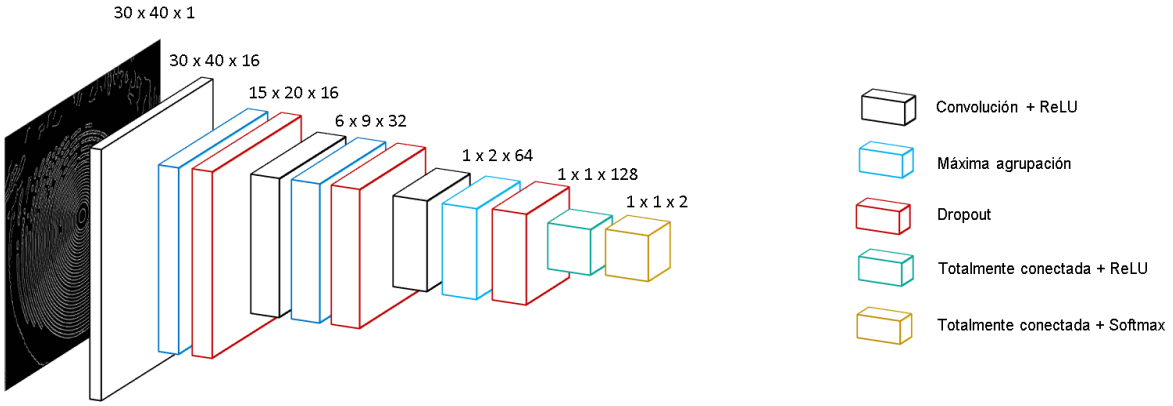
Anexo 1: Arquitectura del Diseño 2



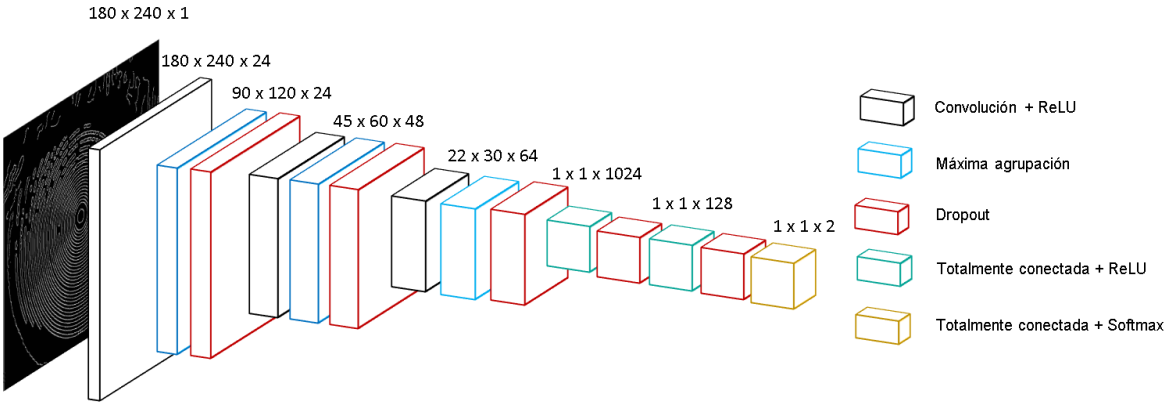
Anexo 2: Arquitectura del Diseño 3



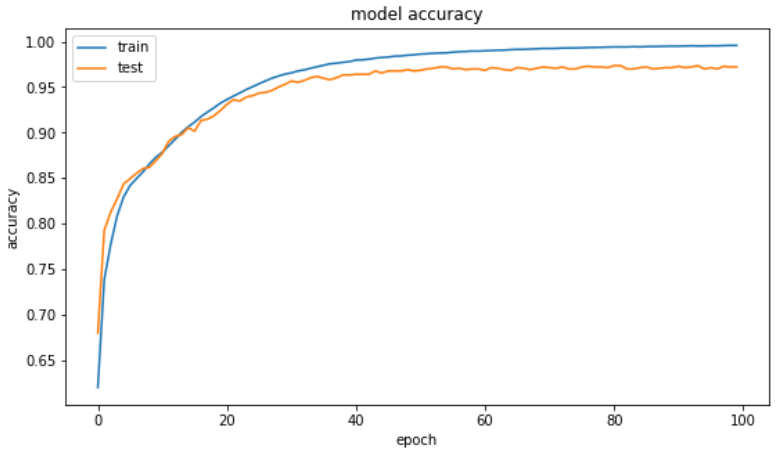
Anexo 3: Arquitectura del Diseño 4

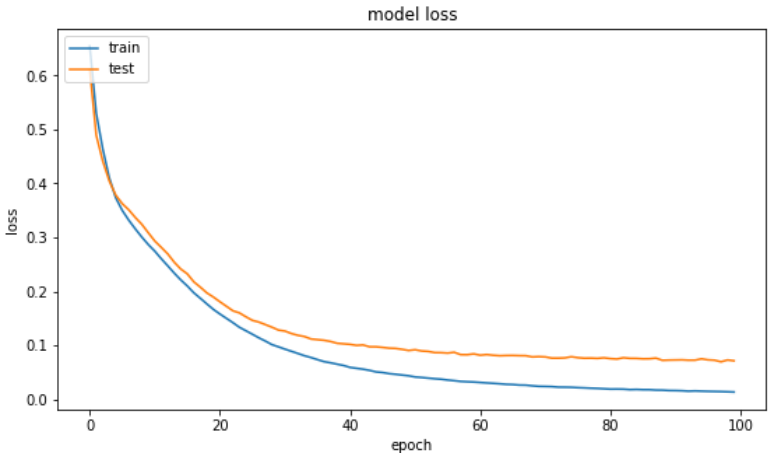


Anexo 4: Arquitectura del Diseño 5

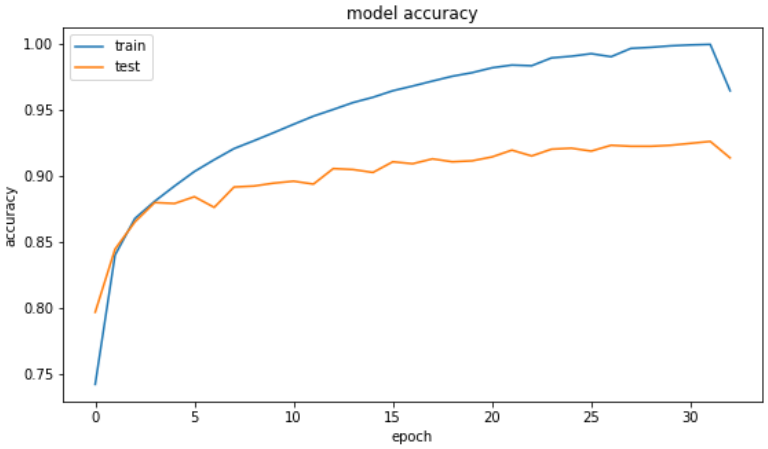


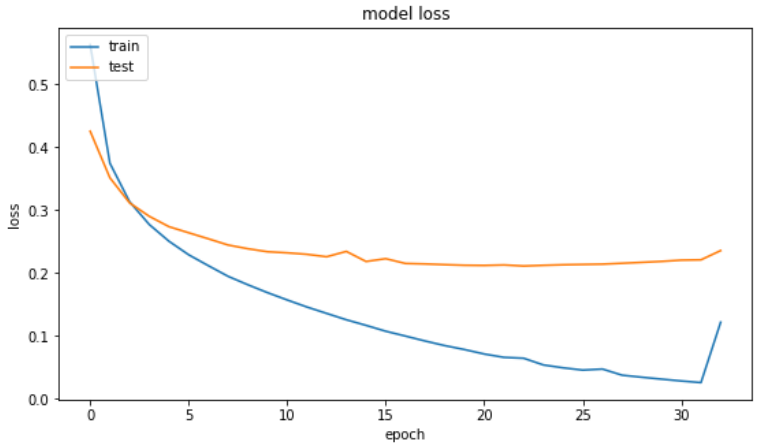
Anexo 5: Grafica precisión y pérdida del Diseño 1



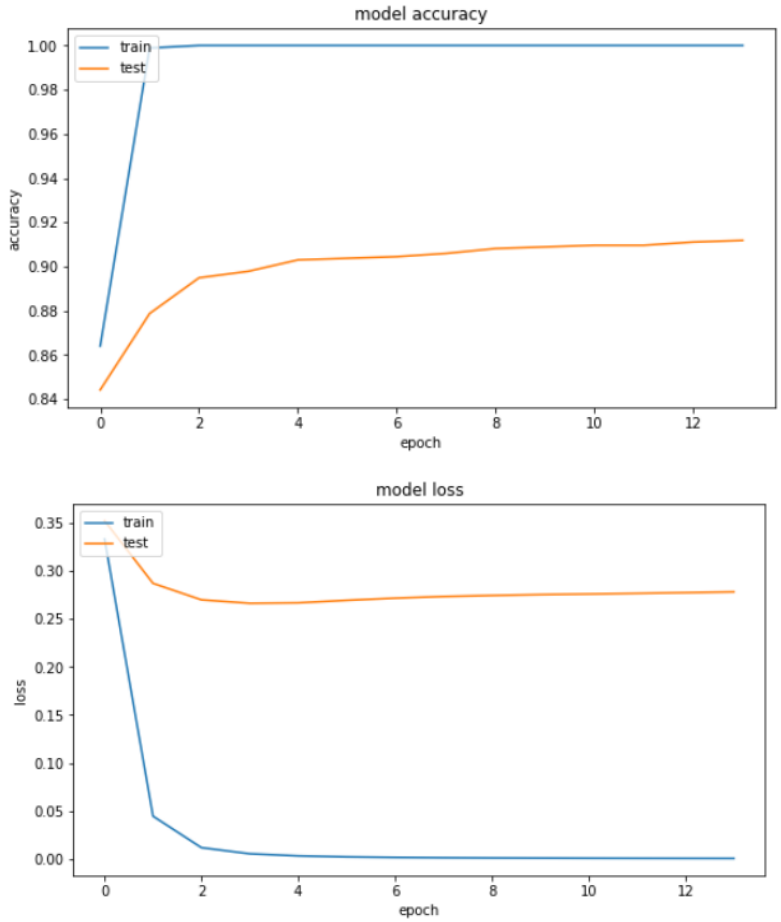


Anexo 6: Grafica precisión y pérdida del Diseño 2

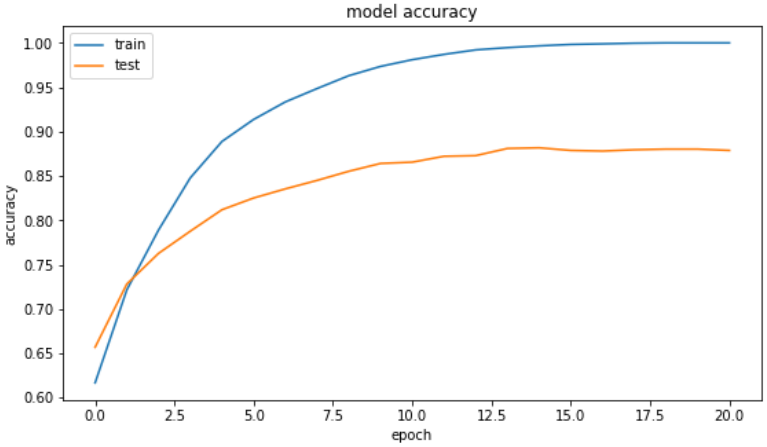


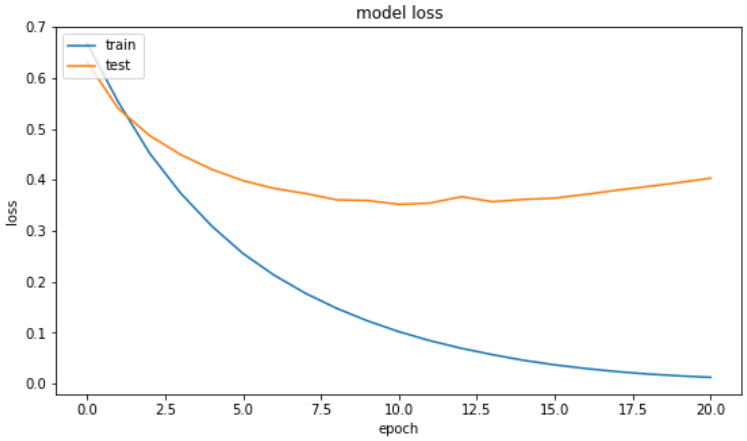
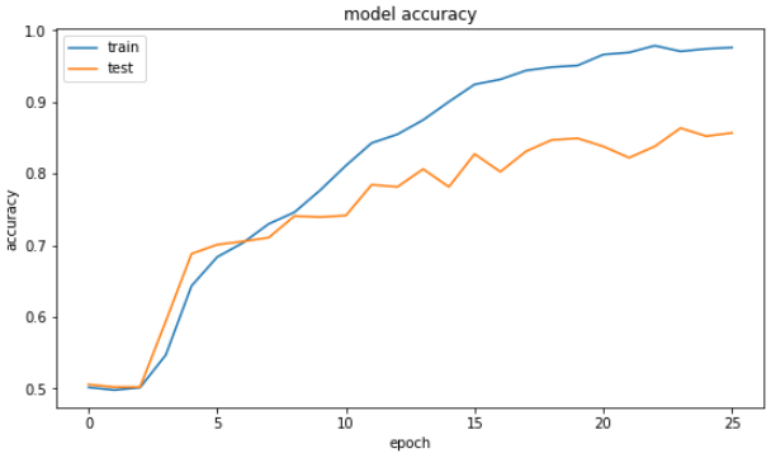
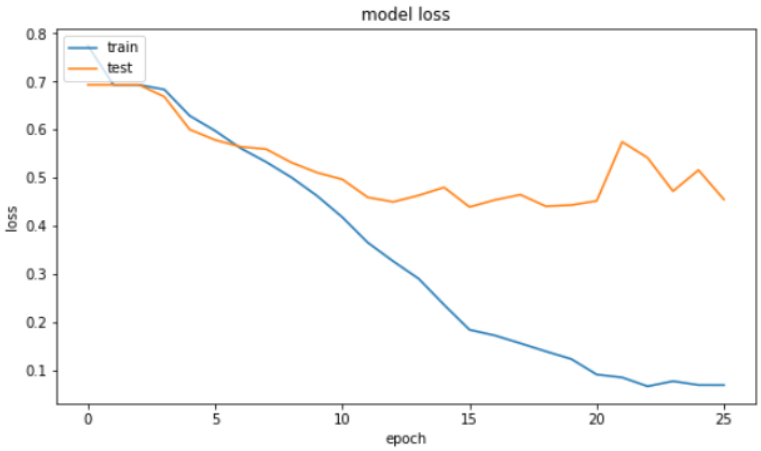


Anexo 7: Grafica precisión y pérdida del Diseño 3



Anexo 8: Grafica precisión y pérdida del Diseño 4





Anexo 9: Grafica precisión y pérdida del Diseño 5