

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



TESIS

**CLASIFICACIÓN DE MAMOGRAFÍAS MEDIANTE
REDES NEURONALES CONVOLUCIONALES**

PRESENTADA POR

ING. MAYRA CRISTINA BERRONES REYES

**COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS EN INGENIERÍA DE SISTEMAS**

JULIO, 2019

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA
SUBDIRECCIÓN DE ESTUDIOS DE POSGRADO



TESIS

**CLASIFICACIÓN DE MAMOGRAFÍAS MEDIANTE
REDES NEURONALES CONVOLUCIONALES**

PRESENTADA POR

ING. MAYRA CRISTINA BERRONES REYES

**COMO REQUISITO PARCIAL PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS EN INGENIERÍA DE SISTEMAS**

JULIO, 2019


Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica
Subdirección de Estudios de Posgrado

Los miembros del Comité de Tesis recomendamos que la Tesis «Clasificación de mamografías mediante redes neuronales convolucionales», realizada por el alumno Ing. Mayra Cristina Berrones Reyes, con número de matrícula 1646291, sea aceptada para su defensa como requisito parcial para obtener el grado de MAESTRÍA en Ciencias en Ingeniería de Sistemas.


El Comité de Tesis



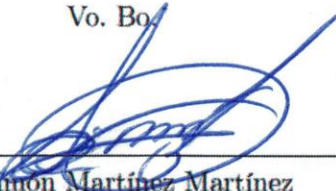
Dra. María Angélica Salazar Aguilar
Directora



Dra. Cristián Castillo Olea
Co-Directora



Dr. Romeo Sánchez Nigenda
Revisor

Vo. Bo.


Dr. Simón Martínez Martínez
Subdirección de Estudios de Posgrado



San Nicolás de los Garza, Nuevo León, julio 2019

A mis abuelos.

ÍNDICE GENERAL

Agradecimientos	XIII
Resumen	xv
1. Introducción	1
1.0.1. Descripción del problema	1
1.0.2. Motivación	4
1.0.3. Hipótesis	6
1.0.4. Estructura de la tesis	6
2. Antecedentes	8
2.1. Tipos de redes neuronales.	12
2.1.1. Redes neuronales convolucionales	12
2.1.2. Anatomía de métricas de redes neuronales	20
2.2. Redes Neuronales en el ámbito de la medicina	21
2.2.1. Mamografía	25
2.2.2. Métodos estudiados para clasificación	26

2.2.3. Métodos automatizados para clasificación	29
2.2.4. Detección asistida por computadora	29
2.3. Redes neuronales convolucionales existentes	32
2.3.1. LeNet (1998)	34
2.3.2. AlexNet (2012)	35
2.3.3. VGGNet (2014)	36
2.3.4. GoogLeNet (2014)	38
2.3.5. ResNet(2015)	39
2.4. Estado del Arte	40
3. Metodología	56
3.1. Bases de datos	58
3.1.1. Mini-mias	59
3.1.2. DDSM	59
3.1.3. INCAN	60
3.2. Metodología propuesta	60
3.2.1. Pre-procesamiento	61
3.2.2. Extracción de características	64
3.2.3. Clasificación	65
3.3. Especificaciones de Software	66
3.3.1. Python	67

3.3.2. KERAS	68
3.3.3. TensorFlow	73
3.3.4. Especificaciones de Hardware	74
4. Diseño de experimentos	75
4.1. Parámetros experimentales	76
4.2. Parámetros fijos	77
4.3. Métricas de salida	81
5. Análisis de Resultados	84
6. Conclusiones y trabajo futuro.	95
A. Códigos en python	100
A.1. Filtros de imágenes	100
A.2. Redimensión de imágenes	101
A.3. División de los conjuntos	102
A.4. Modelo de red	102
A.5. Resultados conjunto de Prueba	104
A.5.1. Re nombrar imágenes	104
A.5.2. Prueba	105
A.6. Pruebas estadísticas en Python	106
B. Códigos en R	108

B.1. Gráficas y pruebas estadísticas	108
B.1.1. Respuesta Terminal	109
C. Resultados completos	114

ÍNDICE DE FIGURAS

2.1. Inspiración biológica de la estructura normal de una neurona. Fuente: Soberanis (2018)	9
2.2. Estructura perceptrón de Rosenblatt. Fuente: Soberanis (2018)	11
2.3. Arquitecturas existentes de redes neuronales, y los tipos de neuronas que estas utilizan. Fuente: AsimovInstitute (2017)	13
2.4. Visualización de una imagen por medio de una computadora. Fuente: AI (2017)	15
2.5. Ejemplo de visualización de como se leen los filtros de cada "parche" de la imagen recibida. Fuente: AI (2017)	17
2.6. Ejemplo de la función y diferencia entre <i>Max pooling</i> y <i>Average pooling</i> . Fuente: Yu et al. (2014)	18
2.7. Ejemplo de función de aplastamiento. Fuente: SuperDataScience (2018)	20
2.8. Ejemplo gráfico de flujo para las métricas de desempeño de una red neuronal. Fuente: Chollet (2018)	21
2.9. Análisis gráfico del comportamiento de células normales y malignas. Fuente: Eldridge (2019).	23
2.10. Anatomía de la mama femenina. Fuente: INC (2018)	24

2.11. Posibles factores que influyen para la aparición de cáncer de mama. Fuente: González (2019)	25
2.12. Mamografía normal (izquierda) y mamografía con anomalía (derecha). Fuente: Suckling (1994)	26
2.13. Mapa de México representando los estados con mayor índice de mortalidad por cáncer de mama. Fuente: INEGI (2019)	30
2.14. Diagrama de bloques de un sistema de CAD para mamografías. Fuente: Cheng et al. (2003).	31
2.15. Diferencias entre las redes ganadoras por cada año de competencia del ImageNet. Fuente: Deng et al. (2009).	33
2.16. Arquitectura utilizada para la estructura de la red neuronal convolucional LeNet-5. Fuente: Max and Pechyonkin (2018)	35
2.17. Arquitectura utilizada para la estructura de la red neuronal convolucional AlexNet. Fuente: Krizhevsky et al. (2012)	36
2.18. Arquitectura utilizada para la estructura de la red neuronal convolucional VGG. Fuente: Krizhevsky et al. (2012)	37
2.19. Arquitectura utilizada para la estructura de la red neuronal convolucional GoogleNet. Fuente: Tsang (2018)	39
2.20. Arquitectura utilizáda para la estructura de la red neuronal convolucional ResNet. Fuente: He et al. (2015)	40
3.1. Estructura de pasos llevados a cabo por los CAD's tradicionales.	57
3.2. Descripción de los pasos utilizados tradicionalmente por las herramientas de diagnóstico y detección.	58
3.3. Estructura de la metodología propuesta para el desarrollo de CAD's.	61

3.4. Ejemplo de estructura de red neuronal convolucional que se desarrolla en este trabajo.	64
3.5. Herramientas de software utilizadas para llevar a cabo la red neuronal.	66
4.1. Distribución de los conjuntos de imágenes para el entrenamiento, desarrollo y prueba de la red.	78
4.2. Exactitud en las diferentes etapas de réplicas del conjunto de entrenamiento.	80
4.3. Exactitud en las diferentes etapas de réplicas del conjunto de desarrollo	80
4.4. Clasificación de las imágenes para el entrenamiento de la red con los valores de la ecuación (4.2)	82
5.1. Efectos univariantes de los factores utilizados en la arquitectura de los modelos.	89
5.2. Interacciones entre los factores de los modelos.	92

ÍNDICE DE TABLAS

2.1. Artículos científicos relacionados	41
4.1. Distribución de características de cada modelo.	76
5.1. Descripción del nombre numérico de los modelos.	86
5.2. Porcentaje de exactitud de modelos en el conjunto de validación.	87
5.3. Resultados del impacto que tienen los factores individuales en los modelos.	90
5.4. Modelos seleccionados para la segunda parte de pruebas y su exactitud.	94
C.1. Desempeño de los modelos en los conjuntos de entrenamiento y vali- dación.	114

AGRADECIMIENTOS

Quisiera iniciar agradeciendo a mis padres Miguel Angel Berrones Jasso y Martha Cecilia Reyes Ornelas, quienes siempre se dieron tiempo de escucharme cuando mi proyecto llegaba a un punto de crisis, ayudándome a encaminarlo por la ruta más adecuada, y por siempre recordarme lo valioso de mis ideas. A mi hermana, Jessica Cecilia Berrones Reyes, por ser una gran fuente de inspiración para nuestra familia. No dejaré de admirar la forma en que allanaste el camino para Victor y para mí, y la manera en que sigues elevando los estándares que tenemos. A mi hermano Victor Hugo Berrones Reyes, que siempre ha sido una fuente de apoyo. Aspiro siempre a tener tu misma tenacidad y confianza en mis habilidades.

A mis abuelos, a los que les dedico también este trabajo. Les agradezco las enseñanzas que les dieron a mis padres. Estas siguen siendo y serán siempre las razones por las que hemos llegado tan lejos como personas. Gracias por ser tan comprensivos y apoyarnos cuando quisimos dar el salto lejos de casa. Ese apoyo incondicional lo llevaré siempre en mi corazón.

A mis compañeros de generación en la maestría. Por convertir esos momentos de estrés y frustración durante la maestría en anécdotas divertidas. Gracias por ofrecerme su amistad, y compartir conmigo sus conocimientos. Lograron elevar mi estancia en la maestría y llenarla de memorias compartidas.

A mi asesora, Dra. María Angélica Salazar Aguilar por alimentar mis ideas y enriquecerlas con las suyas. Su apoyo a lo largo de mi carrera me impulsó a ser mejor estudiante, y el ejemplo de sus logros a esforzarme más como persona.

A la Dra. Begoña García Zapirain y la Dra. Cristián Castillo por haberme facilitado la realización de una estancia en la Universidad de Deusto, laborando codo a codo con su equipo de trabajo. Esta oportunidad abrió considerablemente el paradigma de mi entorno, dándome ocasión de relacionarme con expertos de todas partes del mundo en el tema de mi proyecto.

Al Dr. Eric Ortiz Valdez por facilitarme las imágenes que hicieron posible mejorar mi investigación. Le agradezco inmensamente el tiempo que se tomó para dar revisión a mi herramienta y confiarme con esta tarea tan importante que nos ayudará a salvar vidas.

Por último, mi más grande agradecimiento al CONACYT por permitirme desarrollarme como profesionalista, mediante una beca de manutención y una beca de movilidad.

RESUMEN

Ing. Mayra Cristina Berrones Reyes.

Candidato para obtener el grado de MAESTRÍA en Ciencias en Ingeniería de Sistemas.

Universidad Autónoma de Nuevo León.

Facultad de Ingeniería Mecánica y Eléctrica.

Título del estudio: CLASIFICACIÓN DE MAMOGRAFÍAS MEDIANTE REDES NEURONALES CONVOLUCIONALES.

Número de páginas: 129.

OBJETIVO Y MÉTODOS DE ESTUDIO: Implementar un clasificador de mamografías por medio de herramientas de inteligencia artificial con un nivel de exactitud competente para auxiliar a expertos en el área de medicina en la detección de cáncer de mama, dando como resultado la probabilidad de que exista o no anomalía.

OBJETIVOS ESPECÍFICOS

- Diseñar una estructura de red neuronal con herramientas de software libre.
- Identificar los parámetros adecuados para que la estructura de red clasifique correctamente las imágenes con el menor procesamiento computacional posible.

MÉTODOS DE ESTUDIO: Para poder realizar el clasificador se utilizan herramientas de inteligencia artificial (IA), como lo son las redes neuronales convolucionales (Abreviada CNN por sus siglas en inglés, *Convolutional Neural Network*). A lo largo del trabajo se dará una descripción a detalle del porqué se utiliza este tipo de herramientas a diferencia de las que existen actualmente en el ámbito de clasificación.

En la parte de diseño de experimentos se lleva a cabo un diseño factorial, cuyo propósito es el de identificar cuáles son los parámetros de la arquitectura de la red que proporcionan mejor desempeño en cuanto a la métrica de exactitud de la clasificación.

Finalmente en el apartado de análisis de resultados, se utilizan métricas comunes para el estudio de exactitud de robustez de herramientas en el ámbito del aprendizaje de máquina, y tomando en cuenta estos análisis se llegará a la conclusión de si la herramienta creada en este trabajo tiene una aportación favorable en el tema de clasificación de imágenes médicas.

CONTRIBUCIONES Y CONCLUSIONES: La contribución de este trabajo es una herramienta de clasificación basada en una red convolucional simple, que permite clasificar mamografías (normales o anómalas) con un nivel de exactitud aceptable, en un tiempo de procesamiento corto. Dicha herramienta puede ser utilizada para brindar soporte a los expertos en el análisis de mamografías y como paso previo a la segmentación en la metodología tradicional utilizada por las herramientas de detección asistida por computadora.

Cabe señalar que la herramienta fue probada en bases de datos existentes en la literatura, así como en una base de datos nueva, proporcionada por el INCAN (Instituto Nacional de Cancerología de México) ¹

El porcentaje de exactitud alcanzado en este trabajo es de 99% en el conjunto de validación y 72% en el conjunto de prueba, y la estructura de red que alcanzó


¹Número de registro de base de datos: BI-360-19. Nombre: YERAL.

este desempeño es la red convolucional con las características de optimizador Adam con un tamaño de 160×400 .

Firma de asesores



Dra. María Angelica Salazar Aguilar



Dra. Cristián Castillo Olea

CAPÍTULO 1

INTRODUCCIÓN

En el presente capítulo se describen los puntos base para la realización del trabajo, iniciando con una breve descripción de las redes neuronales, ilustrando la problemática que se desea resolver y cómo es que estas herramientas de inteligencia artificial pueden ayudar a obtener buenos resultados.

Después se describe a grandes rasgos la motivación detrás de esta investigación, remarcando en los puntos que muestran potencial para mejora, y que pueden verse beneficiados con las nuevas tecnologías de inteligencia artificial. Se sigue con las propuestas de hipótesis para los resultados del trabajo, finalizando con la estructura de los capítulos que conforman la totalidad de la tesis.

1.0.1 DESCRIPCIÓN DEL PROBLEMA

Las redes neuronales son un tópico que ha sido utilizado a lo largo del tiempo, caracterizando cada una de sus etapas con avances exponenciales, seguidos por grandes periodos de pausa en su uso. Recientemente se ha iniciado una de las etapas de avances, logrando convertir de nueva cuenta el tópico de redes neuronales y aprendizaje profundo en una línea de interés orientados hacia los temas de aprendizaje de máquina e inteligencia artificial con el enfoque a la resolución de problemas.

Una definición común para describir a las redes neuronales en la literatura es la de una serie de algoritmos que intenta reconocer las relaciones existentes en un conjunto de datos por medio de un proceso que imita la manera en que el cerebro humano trabaja para la resolución de problemas. Un ejemplo de esta definición puede encontrarse en el artículo de Matich (2001). Sin embargo, no fue hasta finales del año 2010 que se desarrollaron nuevos estudios en temas relacionados con el reconocimiento de imágenes utilizando redes neuronales, gracias a la creación de bases de datos anotadas y públicas para facilitar el entrenamiento sin supervisión. Una descripción a fondo sobre una de las primeras bases de datos de este tipo disponibles para el público en general fue desarrollada por los autores Deng et al. (2009).

Una red neuronal tal y como se conocía hace 30 años no poseía la capacidad de estudiar y extraer información relevante de imágenes debido a que el procesamiento para llevarlo a cabo era considerado pesado y costoso para las máquinas en aquel entonces, las cuales no contaban con la capacidad de los núcleos y procesadores que existen hoy en día.

Otra de las razones principales por las cuales las redes neuronales volvieron a ganar popularidad en los círculos de investigación es la forma exponencial en los niveles de información disponibles en el internet siguen aumentando. Lo que se conoce actualmente como *Big Data* pasó de ser un problema de esta generación a convertirse en una oportunidad dentro de diferentes campos de estudio para extraer información relevante. Entre algunas de estas áreas de aplicación se encuentran la de identificación de sistemas, control de procesos, aplicaciones financieras, reconocimiento de patrones, diagnóstico médico, etc.

Todas estas áreas de aplicación mencionadas se han intentado abordar con una gran cantidad de herramientas computacionales e incluso con simple esfuerzo humano. Objetivamente, la cantidad de datos existentes en cualquiera de estos ámbitos hoy en día es demasiada para que incluso un grupo grande de personas puedan sacarle provecho, y es aquí en donde entra la relevancia del aprendizaje profundo, la

minería de datos y el desarrollo de redes neuronales.

Siendo el objetivo de una red neuronal imitar el comportamiento humano en la toma de decisiones para la resolución de problemas, lo que se busca en este trabajo es realizar una herramienta de apoyo para la clasificación de imágenes médicas. Esto con el propósito de agilizar el proceso de clasificación y diagnóstico, con un estándar de exactitud y un nivel de procesamiento aceptable para los niveles de precisión y exactitud que se tienen de herramientas pasadas. Ejemplos de algunos valores de exactitud encontrados en la literatura pueden ser revisados en artículos de autores como Cheng et al. (2003), Freer and Ulissey (2001), Warren Burhenne et al. (2000), Sampat et al. (2005), Henriksen et al. (2019).

En la mayor parte de la literatura de estudio de antecedentes, existe una serie de pasos que se llevan a cabo para la clasificación de imágenes en el ámbito de diagnóstico de cáncer, que es el tema que motiva este trabajo. Primero, existe un pre-procesamiento de la imagen, seguido por una segmentación. Después se muestra la extracción de características, su selección, y por último su clasificación. Algunos autores como Gur et al. (2004), Ganesan et al. (2012) demuestran este seguimiento de pasos en el desarrollo de sus respectivas herramientas.

Los clasificadores que se desarrollan con esta metodología aún siguen mostrando un alto nivel de complejidad computacional y procesamiento, además de que su exactitud muestra tener un gran potencial para mejorar sus resultados, por lo cual en este trabajo se propone una alternativa con la cual se busca elevar este nivel de exactitud para hacer más atractivo el uso de esta herramienta de manera cotidiana entre los expertos.

En la etapa de segmentación es donde se gasta mayor trabajo computacional siguiendo el orden de los pasos que se ven para las herramientas de clasificación. Por esto se propone un cambio de dicha metodología. Lo que se plantea en este trabajo es una serie de pasos extras antes de que se tome en cuenta la etapa de segmentación, básicamente haciendo un filtro de imágenes.

Los pasos descritos a lo largo de este trabajo son los de una herramienta utilizando redes neuronales para clasificar mamografías separándolas en dos categorías, imágenes sin anomalía e imágenes con anomalía. Esto con el fin de que el esfuerzo computacional de la segmentación no se desperdicie en imágenes que no necesitan ser analizadas para buscar anomalías ya que cuentan con una probabilidad muy alta de que esta no contenga ninguna anomalía y que en todo caso pueden terminar provocando lo que se conoce como falsos positivos, indicando segmentaciones en la imagen que no pertenecen a la categoría de anomalías.

1.0.2 MOTIVACIÓN

Como es mencionado en la sección de descripción del problema, el motivo principal por el que las redes neuronales estén entrando en una nueva etapa de avance, se debe a que hoy en día existe una mina de datos de la cual se pueden extraer características y patrones que facilitarían una gran cantidad de tareas, las cuales en este punto se han vuelto imposibles de procesar solo contando con esfuerzo humano. Una herramienta capaz de sacar provecho de este bloque de información en poco tiempo y haciendo buen uso de recursos computacionales puede llegar a ser de gran utilidad.

Enfocando este problema de exceso de información a un tema más específico, se puede tomar en cuenta el ámbito de salud en donde existen cantidad de enfermedades, las cuales podrían beneficiarse de herramientas que utilizan las redes neuronales de base, ya que, añadiéndole al problema de la cuantía de datos de pacientes, en el área médica existe una cantidad reducida de expertos que puedan hacer uso correcto de esa información, ya que en el tema de clasificación de enfermedades se requiere personal altamente especializado tanto en la patología como en el área del cuerpo en la que se ubica.

Generalizando aún más el tópico de herramientas computacionales en el ámbito

de la medicina, surge el tema del cáncer, que es una de las enfermedades con mayor tasa de mortalidad cuanto más tiempo tarde en llegar la detección y el diagnóstico de dicho mal. En 2015 se atribuyeron a esta enfermedad 8.8 millones de defunciones según investigaciones de la OMS (2019) (Organización Mundial de la Salud).

Algunas herramientas de diagnóstico han intentado generalizar la detección y clasificación de cáncer en imágenes, como se puede observar en el artículo realizado por Kumar et al. (2017), más en este trabajo se especializan sus pasos para ser capaz de detectar el cáncer de mama, el cual a la fecha se encuentra en el quinto lugar de cánceres con mayor cantidad de defunciones alrededor del mundo según la OMS (2019).

El mes de octubre a nivel mundial, se ha caracterizado por ser el mes de sensibilización sobre el cáncer de mama, lo que ha contribuido a una nueva cultura de revisión temprana entre las comunidades, facilitada por las campañas que se llevan a cabo a lo largo de este mes, ofreciendo exámenes gratuitos, y ofreciendo más información acerca de este tipo de cáncer, lo que trajo como consecuencia una alza considerable en la cantidad de información que se tiene al respecto, siendo una de ellas las imágenes médicas llamadas mamografías.

Cada año existen alrededor de 1.38 millones de nuevos casos de cáncer de mama, y alrededor de 458 000 de esos casos resultan en muertes de acuerdo con investigaciones realizadas por la División de Prevención y Control del Cáncer (2018). Aunque los conocimientos sobre las causas del cáncer de mama son insuficientes en la actualidad, de acuerdo con BreastCancerOrganization (2015) esta nueva actitud de prevención ha ayudado a disminuir considerablemente el nivel de mortalidad entre los pacientes afectados.

En México un estudio realizado por la UNAM (2018) (Universidad Autónoma de México) estimó que en el año 2014 existían alrededor de 689 mastógrafos a nivel nacional, ubicados principalmente en la Ciudad de México. Esto parece un número bastante reducido de expertos para la cantidad abrumadora de información que

reciben cada año.

Como cualquier otro tipo de cáncer, la clasificación y detección temprana de dichas anomalías juega un papel de vital importancia en cuanto a la supervivencia del paciente, por lo cual, una herramienta que ayude a agilizar este proceso tiene bastante relevancia.

1.0.3 HIPÓTESIS

- Es posible encontrar la estructura de un modelo de red neuronal que pueda clasificar mamografías con una exactitud cercana al 80 %, que es el porcentaje de exactitud que se presenta comúnmente en las herramientas computacionales para diagnóstico y clasificación.

1.0.4 ESTRUCTURA DE LA TESIS

En el primer capítulo se presenta una descripción concisa del proyecto que se lleva a cabo en la tesis, mencionando brevemente la problemática, la motivación para desarrollar dicho trabajo, los objetivos que quieren alcanzar y la hipótesis que se espera cumplir al finalizar la experimentación y el análisis de los resultados.

En el segundo capítulo se ven los antecedentes del problema a resolver, llegando a explicar brevemente la problemática del cáncer de mama, y otros esfuerzos que se han realizado para la ayuda de detección y diagnóstico de imágenes en el área médica, además de una descripción de las redes neuronales convolucionales y el funcionamiento de cada una de sus partes, para lograr un mejor entendimiento del porqué se escogen los parámetros utilizados en este trabajo para intentar lograr una arquitectura simple y que a su vez genere buenos resultados.

Como complemento del capítulo de antecedentes se realiza una investigación

de los trabajos científicos que hacen uso de herramientas similares o diferentes a las redes neuronales convolucionales para la resolución del problema que se presenta en este trabajo, explicando sus características, similitudes, diferencias y mostrando el nivel de exactitud alcanzado por estas.

El tercer capítulo describe la metodología tradicional, utilizada en herramientas computacionales médicas para la detección y el diagnóstico. Después se sigue a explicar a detalle las herramientas utilizadas para construir el clasificador, iniciando por las bases de datos de donde se tomaron las imágenes, seguido por la descripción de las diferentes arquitecturas de red utilizadas. Esta sección finaliza con las especificaciones del software y hardware utilizados en la realización del trabajo.

En el cuarto capítulo se describen los parámetros fijos y los experimentales de la estructura de la red neuronal, con los cuales se busca encontrar la configuración óptima de características que permita alcanzar el nivel de exactitud deseado en la parte de prueba. En esta sección se describen también los cálculos que se realizan para obtener las métricas de salida que se observan como resultado.

En el quinto capítulo se explica el desarrollo de análisis de resultados, argumentando el porqué de las decisiones de despreciar algunas estructuras, además de explicar el razonamiento detrás de la decisión de escoger ciertos modelos para realizar pruebas posteriores a estos resultados.

En el sexto capítulo se presentan las conclusiones y el trabajo a futuro. Se discuten los resultados obtenidos, resaltando las posibles respuestas al comportamiento de algunos modelos, así como las posibles mejoras a la herramienta propuesta.

Para finalizar se tienen los apéndices A, B, y C, en donde se muestran los programas en Python y R, utilizados para la experimentación y el análisis de los resultados. Además de la explicación detallada de los modelos de red analizados en el presente trabajo.

CAPÍTULO 2

ANTECEDENTES

En este capítulo se entra más a detalle en las herramientas utilizadas para resolver la problemática presentada en este trabajo. Se revisan los componentes de las redes neuronales convolucionales para describir la importancia de cada una de sus partes antes de iniciar con la arquitectura de la red propuesta en este trabajo.

Además de describir las herramientas, se tiene un panorama más grande del problema de cáncer de mama, y cuáles son las principales razones por las que se considera que tiene relevancia crear una herramienta que ayude a expertos a agilizar el proceso de detección.

Por último se tienen las diferentes redes convolucionales de las cuales se tomó inspiración en su arquitectura basándose en el desempeño presentado en los retos de clasificación de imágenes normales (no médicas) en los que fueron introducidas por primera vez.

†

A lo largo del tiempo, la ciencia ha tomado inspiración en la naturaleza para resolver una amplia cantidad de problemas, ya que esta representa un buen punto de referencia para poder llegar a diseñar herramientas que faciliten muchas de las tareas cotidianas. En el área de la inteligencia artificial, existen algoritmos que se

basan en modelos naturales, buscando adaptarlos a las necesidades del problema que se quiere resolver, minimizando sus errores y volviéndolos más eficientes en sus tareas.

Hoy en día, las palabras como aprendizaje de máquina o aprendizaje profundo son temas clave en las ramas de ciencias de la computación, que recientemente han ido tomando un mayor reconocimiento en los círculos de investigadores y programadores. El aprendizaje profundo, o mejor conocido como *Deep Learning*, es uno de los sub campos del tema de inteligencia artificial que toma inspiración en el ámbito de las redes neuronales, que en sí mismo se basa en el funcionamiento biológico del cerebro humano.

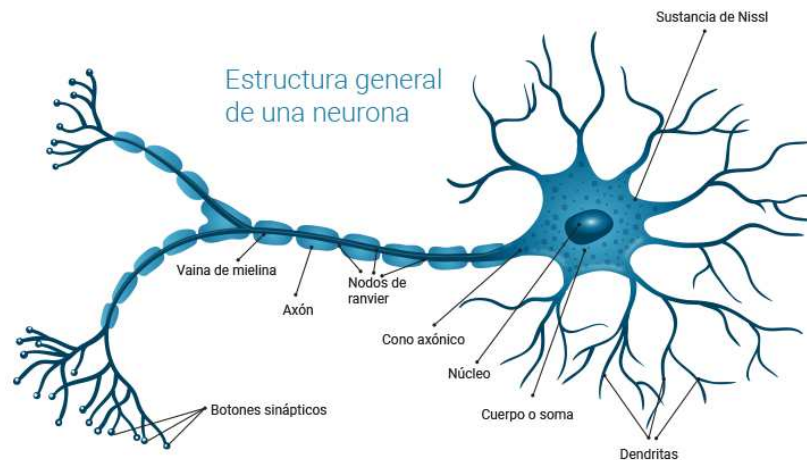


Figura 2.1: Inspiración biológica de la estructura normal de una neurona. Fuente: Soberanis (2018)

Las neuronas se conectan unas con otras por medio de conexiones llamadas sinápsis con las cuales intercambian señales eléctricas entre el emisor sináptico de una neurona, hacia el receptor dendrítico de alguna otra. Sus partes y estructura se muestran en la Figura 2.1. Cuando la neurona recibe un estímulo eléctrico acumula dicha energía, y cuando esta llega a tomar cierta cantidad, la neurona descarga por medio del axón a las neuronas circundantes, formando una conexión sináptica. Para profundizar más en el tema, se puede consultar el trabajo de Basheer and Hajmeer

(2000).

Tomando en cuenta el principio básico de este comportamiento, los científicos Warren McCulloch (1986) crearon un modelo artificial de una neurona. En este primer modelo creado, se tenían algunas ideas en cuanto a cuál era el comportamiento de la neurona que querían traducir en un algoritmo computacional, entre ellas se encuentran:

- La actividad de una neurona es un proceso de todo o nada.
- Debe de haber una cierta cantidad de sinápsis excitada para que una neurona pueda activarse, y esta cantidad deberá ser independiente a cualquier actividad anterior o posición de la neurona dentro de la estructura.
- La estructura de la red no cambia con el tiempo.

Estas características se presentan en el conocido modelo matemático M-P. Un modelo M-P de una neurona es un elemento con n entradas y una sola salida. La forma general para representar esta función es la siguiente, de acuerdo con el trabajo elaborado por Zhang and Zhang (1999):

$$y = \text{sgn}(wx - \varphi) \tag{2.1}$$

En donde:

$x = (x_1, x_2, \dots, x_n)^T$ ← un vector de entrada

$w = (w_1, w_2, \dots, w_n)$ ← un vector de pesos

φ ← un umbral

Años después, esta idea fue retomada por el científico Frank Rosenblatt, quien tomó en sus manos la tarea de ilustrar de manera más generalizada algunas de las propiedades fundamentales que se observan en sistemas inteligentes, sin adentrarse demasiado a los detalles o condiciones específicas que se encuentran en los organismos biológicos. En este concepto, se opuso al análisis de McCulloch, el cual se basaba en

una lógica un poco más simbólica. Estas comparaciones y mejoras pueden estudiarse más a detalle en el artículo elaborado por Anderson and Rosenfeld (2000).

Rosenblatt (1958) propuso que un análisis de probabilidades era una herramienta más apropiada, lo que lo llevó a una teoría de separabilidad estadística. De esta manera desarrolló la estructura que hoy en día conocemos como perceptrón. El perceptrón es una estructura con varias entradas, una función de activación intermedia, y por último, una salida. Este comportamiento se muestra de manera gráfica en la Figura 2.2.

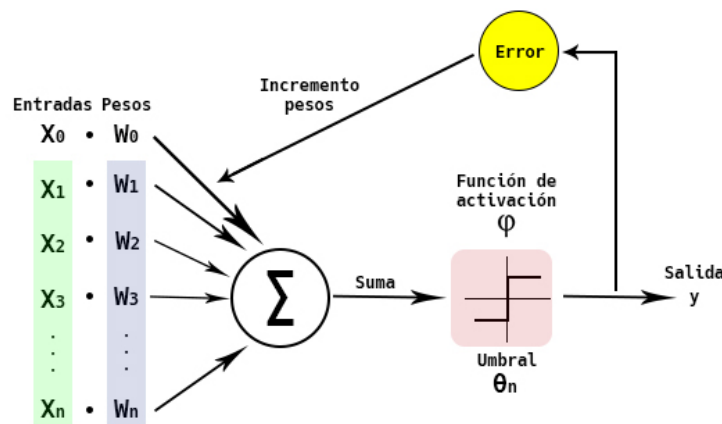


Figura 2.2: Estructura perceptrón de Rosenblatt. Fuente: Soberanis (2018)

Comparando esta nueva estructura de Rosenblatt con la estructura propuesta por McCulloch, podemos comparar la capa de entrada del perceptrón como las dendritas, que actúan como señales de entrada hacia la neurona. La función intermedia de activación tomará el papel de la sumatoria que decide en que momento se activa la señal eléctrica mediante el umbral seleccionado en la parte de la función de activación, y por último la salida se percibe como el axón que cuando se activa envía la información a las neuronas adyacentes con las que está conectada.

A partir de estas descripciones, se puede observar el comportamiento de una red neuronal en su forma más simplificada. Estos trabajos formaron los cimientos en los cuales hoy en día se basan una gran cantidad de arquitecturas de red, cada una de ellas intentando ser mucho más particular en su estructura para resolver los

problemas cada vez más específicos, presentes en la actualidad.

2.1 TIPOS DE REDES NEURONALES.

Siguiendo el tema de redes que evolucionan para concentrarse en trabajos más específicos, nos encontramos con la gran variedad de redes neuronales que existen hoy en día. Gracias a esta cantidad de redes, cada una diferenciándose por sus características y funcionalidades, puede ser complicado y en ocasiones abrumador el hecho de dar búsqueda a una de ellas para resolver cualquier problema.

Ejemplos de los diferentes tipos de redes que existen hoy en día se pueden mostrar en la Figura 2.3. Para el desarrollo de la presente tesis, se utilizan las redes neuronales del tipo convolucionales.

2.1.1 REDES NEURONALES CONVOLUCIONALES

Las redes convolucionales son utilizadas en su mayoría para la clasificación de imágenes. La eficiencia que han mostrado en sus resultados es una de las razones principales por las que el aprendizaje profundo y el aprendizaje de máquina han tenido un nuevo auge en la investigación.

Para crear un panorama claro de la descripción de los modelos desarrollados en este trabajo, se realiza una revisión de la estructura básica de una red neuronal convolucional. Primeramente, una de las cosas más importantes a entender es que las redes convolucionales no perciben las imágenes como lo hacen los seres humanos.

A pesar de que el nivel de memoria y procesamiento de cálculo con el que cuenta cualquier computadora es superior a la de un ser humano por un gran margen, tareas que pudieran parecer poco complejas, como clasificar objetos en una imagen, reconocer el rostro de una persona, y hacerlo con la facilidad y rapidez con la

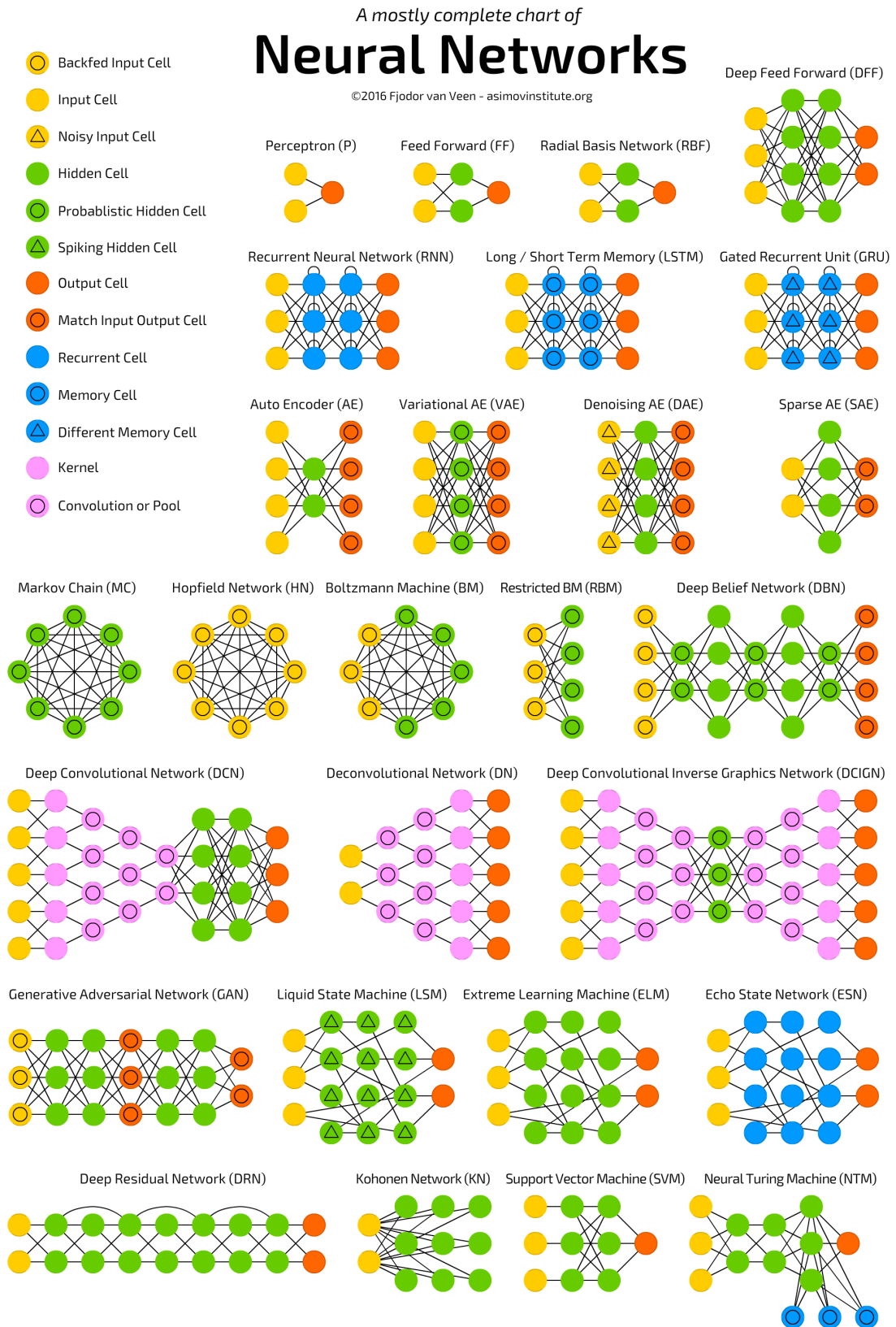


Figura 2.3: Arquitecturas existentes de redes neuronales, y los tipos de neuronas que estas utilizan. Fuente: AsimovInstitute (2017)

que la puede hacer cualquier humano, las computadoras tienen más dificultad para afrontarlas.

Otra de las características con las que cuentan los ordenadores, es que harán exactamente lo que se les pida y en el orden especificado, lo cual presenta tanto ventajas como desventajas. Las ventajas se muestran cuando los trabajos que se requieren son repetitivos o matemáticamente complejos. En cambio, cuando se quiere que la computadora se adapte, y tome decisiones que mejoren su resultado sin tener que controlar manualmente estos cambios, es donde entran las desventajas.

Lo que se puede hacer en estos casos es deshacer la información y comandos que se le dan al ordenador y reconstruirlos de manera que estos puedan procesarla. Una computadora no puede "ver" una imagen de la misma forma en que lo hace un ser humano, ya que no cuenta con ojos en sí. La manera de resolver este problema es pensar un método distinto para alimentar la imagen a una red neuronal, de forma que potencialice sus atributos, como lo es el procesamiento matemático y la cantidad de memoria. Esto quiere decir que tenemos que traducir la imagen en un lenguaje que cualquier computadora entienda.

La manera de traducir dichas imágenes es convertirlas en números, como se muestra en la Figura 2.4 en donde se representa una imagen por medio de una computadora, representado como un cubo donde las capas en donde los ejes x , y , z son los canales de colores, y los números dentro de los cuadros representan el gradiente de dicho color. A partir de este punto, el papel que representan las redes neuronales convolucionales es el de reducir la imagen a una forma o estructura que sea más fácil de procesar, sin que se pierdan características importantes para la final clasificación de la imagen.

Las redes convolucionales reciben las imágenes como si fueran volúmenes, en lugar de cuadros fijos en donde solo vemos la altura y anchura. Estas redes también reciben un tercer valor, que es el color de la imagen, y lo manejan como si fueran tres capas separadas puestas unas encima de otras.

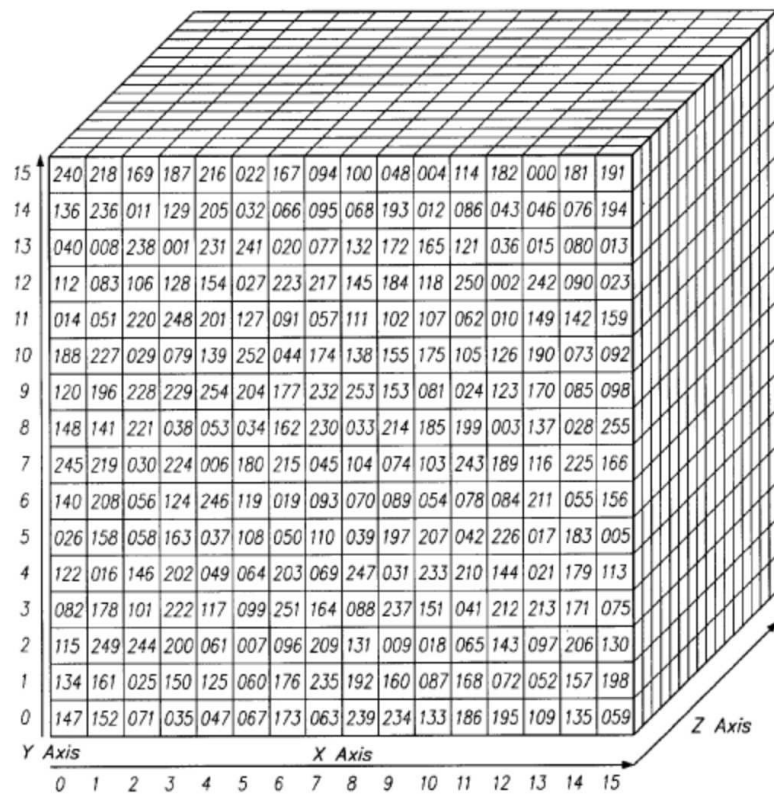


Figura 2.4: Visualización de una imagen por medio de una computadora. Fuente: AI (2017)

De esta manera las redes convolucionales reciben una imagen como si fuera una caja rectangular en la cual las medidas de altura y anchura se miden por el número de pixeles que conforman el perímetro de dichas dimensiones, y cuya profundidad se traduce en las capas de colores que tiene. Estas capas de profundidad se le conocen como canales.

La primer capa de la red se encarga de recibir los datos de la imagen, expresadas matemáticamente como una matriz describiendo sus dimensiones. Primero es la altura de la imagen, después la anchura, y por último se tiene el tipo de imagen, ya sea RGB, blanco y negro, etc. De capa a capa las dimensiones de dicha matriz tienden a cambiar.

Para expresar de manera matemática una imagen a una computadora se le da el valor de un gradiente, el cual se representa como un número que cambia dependiendo de la capa de color en la que se encuentre y la intensidad o luminosidad que presente. En el caso de las imágenes a blanco y negro, este valor es el de la capa principal que va en una escala de 0 a 255. Estos números son características sensoriales que alimentan a la red convolucional, y el propósito de esta red es encontrar cuales de esos números representan señales significativas para ayudar a clasificar la imagen de manera correcta.

Las siguientes capas, que se conocen como capas ocultas, no se concentran en un pixel a la vez como la capa de entrada, si no que toman lo que llamamos "parches" de pixeles, que después tienen que pasar por un filtro. Dicho filtro es una matriz cuadrada, que es más pequeña que la imagen de entrada, y en todos los casos los filtros se muestran del mismo tamaño. Este comportamiento se describe de manera gráfica en la Figura 2.5.

Para poder seguir a la siguiente capa, se toman en cuenta lo que se conocen como funciones de activación, con las cuales la red decide si las neuronas a las que se conecta van a activarse o no. Se puede imaginar que se inicia la revisión de la imagen desde una de las esquinas, y se mueve paso a paso hasta llegar a la otra esquina,

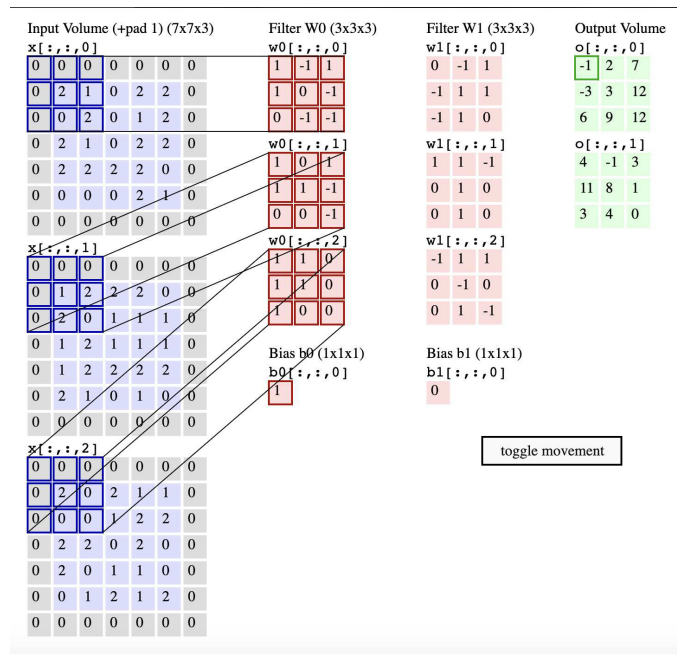


Figura 2.5: Ejemplo de visualización de como se leen los filtros de cada "parche" de la imagen recibida. Fuente: AI (2017)

cubriendo todos los pixeles de la imagen en el proceso. El tamaño de este paso se le llama zancada o *stride* como se le conoce en inglés.

Como es lógico, entre más grande la zancada, es menor la cantidad de pasos que tienen que darse para completar la revisión total de la imagen, y el mapa de aplastamiento es más pequeño ya que tiene menos características que resumir. De la misma manera, una zancada larga se traduce en menos tiempo y procesamiento de computación.

Después de las capas convolucionales u ocultas que se acaban de describir, se tiene la siguiente capa, llamada comúnmente capa de agrupación, que también puede conocerse por sus otros nombres en inglés, *pooling*, *downsampling* y *subsampling*. Esta capa, al igual que la capa de convolución, es responsable de reducir la capacidad computacional que se requiere para procesar la imagen, ya que una de sus funciones más importantes es la de extraer las características predominantes de los parches de entrada.

En el caso de extracción de características, existen tres tipos distintos de *pooling* a elegir, en los que cada uno de ellos dan prioridades a diferentes atributos del mapa de activación que reciben. Al igual que las capas de convolución este método se va aplicando un parche a la vez. A continuación se describen los tres diferentes tipos de *pooling* que pueden elegirse para las capas ocultas, se detallan sus características para ejemplificar su importancia en el modelo.

- **MaxPooling:**

El *Maxpooling* toma los valores más altos del parche de imagen que cubre el parche que se le indica. En los parámetros de entrada debe especificarse el tamaño de la matriz de salida en donde van a acomodarse los valores más altos de otros parches.

La información que no es utilizada al detectar los valores más altos, simplemente se descarta.

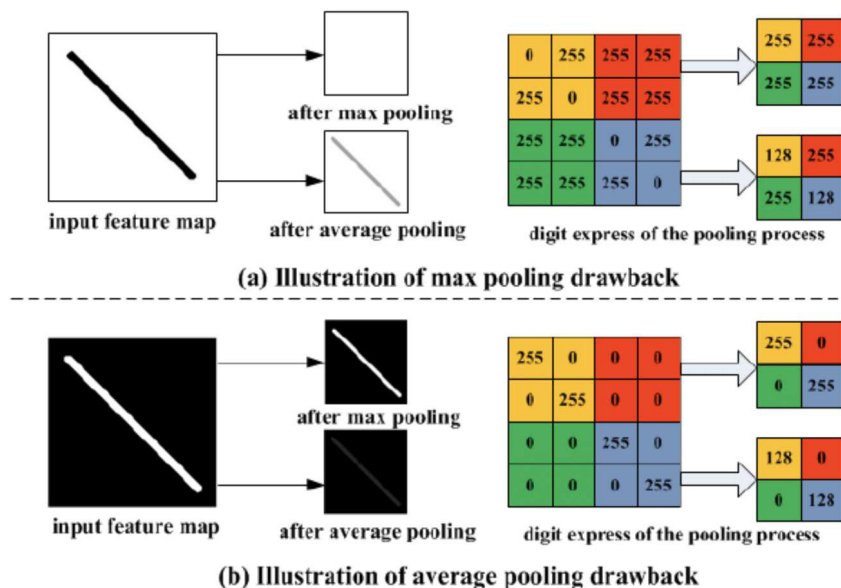


Figura 2.6: Ejemplo de la función y diferencia entre *Max pooling* y *Average pooling*.

Fuente: Yu et al. (2014)

- **Average Pooling:**

A diferencia del *Maxpooling*, el *Average pooling* toma en cuenta todo los datos para hacer su selección de características que pasarán a la siguiente matriz de atributos, no solo las mayores, como se muestra en la Figura 2.6. Este tipo de función toma en cuenta todos los datos y los deja pasar a la siguiente capa, lo cual nos deja con un sistema muy generalizado para la clasificación, lo que puede traer malos resultados cuando no es necesario tomar en cuenta toda la información de entrada.

- **Global Max/Average pooling:**

El *Global Maxpooling* y *Global Average pooling* tiene las mismas funciones de extracción de características que se mencionan anteriormente, sin embargo, este toma menos parámetros de salida, por ejemplo, en donde el *Maxpooling* o *Average pooling* toman una matriz de salida, el *Global* toma el valor generalizado de este, con solo un valor de salida.

En general, muchos de los valores que no tienen tanto impacto en el parche que se les asignó, se pierden en este paso, lo cual ha intentado mejorarse en trabajos recientes, como por ejemplo, el trabajo realizado por Singla (2017). La razón por la que se sigue usando en redes convolucionales hoy en día, es por la gran ventaja que presenta de bajar considerablemente el nivel de procesamiento y memoria requeridos, lo que aún sobrepasa a la posible información que se pierde en este caso.

La penúltima capa que se necesita en una red neuronal convolucional es la capa de aplastamiento, o como se conoce en inglés, *Flattening*. (Figura 2.7) Al finalizar los pasos que se llevan a cabo en las capas convolucionales ocultas y de agrupamiento, esta capa es la encargada de organizar los mapas de atributos en una matriz de una sola columna, de ahí se toma el nombre de capa de aplastamiento.

Esta capa no realiza ningún cálculo matemático ni tiene alguna función interna especial. Simplemente se encarga de ordenar los datos para darlos de entrada a

la última capa. Al final de todas estas capas se tiene la predicción de salida, y dependiendo de los parámetros que se decida dar a la red neuronal, esta va a especializarse en una métrica final para saber que tan efectiva es su predicción final, comparándola con la respuesta real.

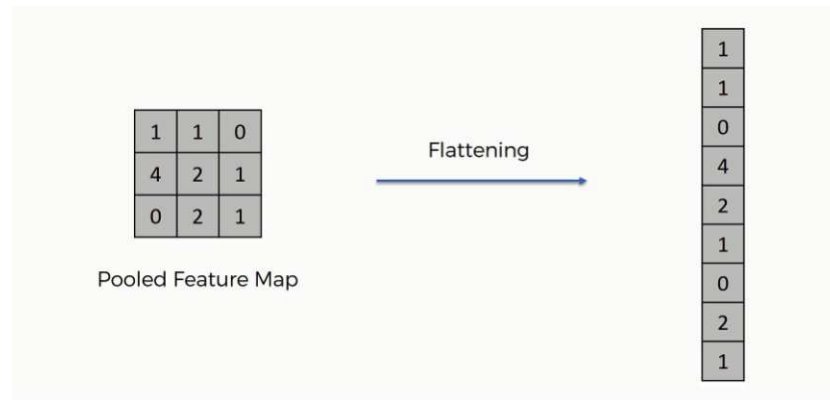


Figura 2.7: Ejemplo de función de aplastamiento. Fuente: SuperDataScience (2018)

2.1.2 ANATOMÍA DE MÉTRICAS DE REDES NEURONALES

Ya que se tiene una idea general del funcionamiento de las redes neuronales convolucionales, podemos profundizar en el tema de la salida final, o más específicamente, la métrica utilizada para medir el rendimiento de dicha red en el análisis de experimentos.

Las dos funciones que ayudan a saber que tan bien se desempeña una red neuronal son la función de pérdida y la de exactitud. Se sabe que las redes neuronales se desarrollan de acuerdo a los siguientes elementos:

- Capas, las cuales se combinan para formar el tipo de red que se esté utilizando.
- Los datos de entrada, relacionados a la predicción deseada, o el objetivo a llegar.

- La función de pérdida, la cual define la señal de aprendizaje que se utiliza en el entrenamiento.
- El optimizador, que determina el ritmo de aprendizaje que se quiere llevar a cabo. Encargado de disminuir el error y maximizar la exactitud.

Un ejemplo visual de como se desarrollan las métricas del optimizador y de pérdida, pueden ser vistas en la Figura 2.8.²

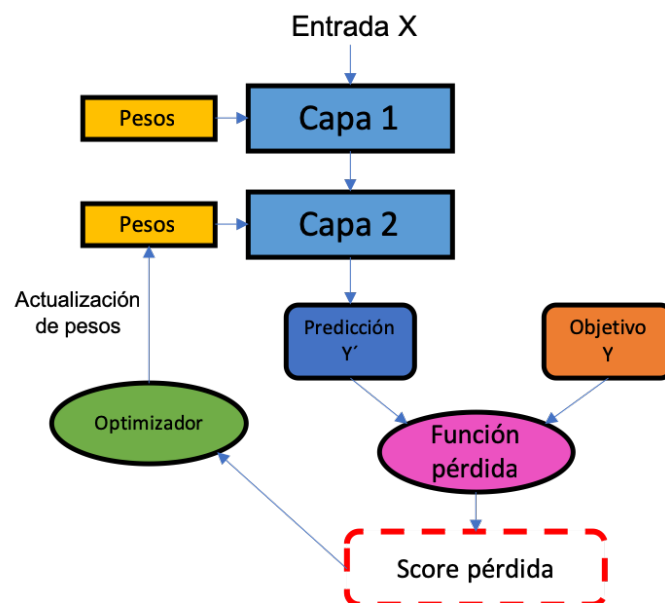


Figura 2.8: Ejemplo gráfico de flujo para las métricas de desempeño de una red neuronal. Fuente: Chollet (2018)

2.2 REDES NEURONALES EN EL ÁMBITO DE LA MEDICINA

Mientras que en el campo de las ciencias computacionales se empieza a observar un avance exponencial en su capacidad de procesamiento dentro de temas como aprendizaje de máquina y aprendizaje profundo, en el campo de la medicina existe un

²Imagen tomada de Chollet (2018) traducida al español

alza de información gracias a la nueva tendencia de prevención que se ha desarrollado en los últimos años.

Automatizar procesos es una de las mayores ventajas que conlleva la mejora del procesamiento computacional, con lo cual ha sido posible agilizar muchos otros procesos, que con simple esfuerzo humano hubieran resultado demasiado costosos. Pero para que estos procesos puedan llevarse a cabo por medio de herramientas computacionales, fue imperativo que mostraran un nivel de confianza en la exactitud de sus resultados superiores a la exactitud a la que es capaz de alcanzar un ser humano.

A pesar de que existen ya herramientas que ayudan al diagnóstico y detección de enfermedades, estas no tienen la exactitud suficiente para que los expertos las utilicen de manera rutinaria. La mayoría de ellas sobrepasan el umbral de exactitud alcanzado por el promedio de expertos que se encuentra dentro del promedio de un 70 % a un 80 % de acuerdo con algunas investigaciones las cuales se pueden consultar en el trabajo de Kohli and Jha (2018), mas con este nuevo avance de tecnologías se pueden explorar este tipo de métodos y mejorarlos tanto en exactitud, como en costo de recursos utilizados tales como, tiempo y procesamiento computacional.

En cualquier tipo enfermedad es de vital importancia la pronta detección, al igual que el diagnóstico oportuno. Una de las enfermedades en las que más se beneficia de tener un margen grande en el tiempo de detección y diagnóstico, para que así el paciente tenga una mayor probabilidad de supervivencia, es el cáncer.

El cáncer es lo que resulta de las mutaciones de los genes encargados de regular el crecimiento de las células. Un comportamiento normal en este tipo de mutaciones es que las células del cuerpo entren en un proceso de renovación, en el cual, nuevas células sanas crecen para reemplazar aquellas que son viejas y mueren. Ejemplo de este comportamiento mostrado por células cancerosas se representa en la Figura 2.9.

3

³Imagen tomada del trabajo de Eldridge (2019) y traducida al español.

Sin embargo, algunas de estas mutaciones pueden activar genes de manera descontrolada, en cuyo caso, las células producidas toman una capacidad de reproducirse sin ningún tipo de orden, lo cual resulta en un tumor de acuerdo con información extraída de ACS (2019).

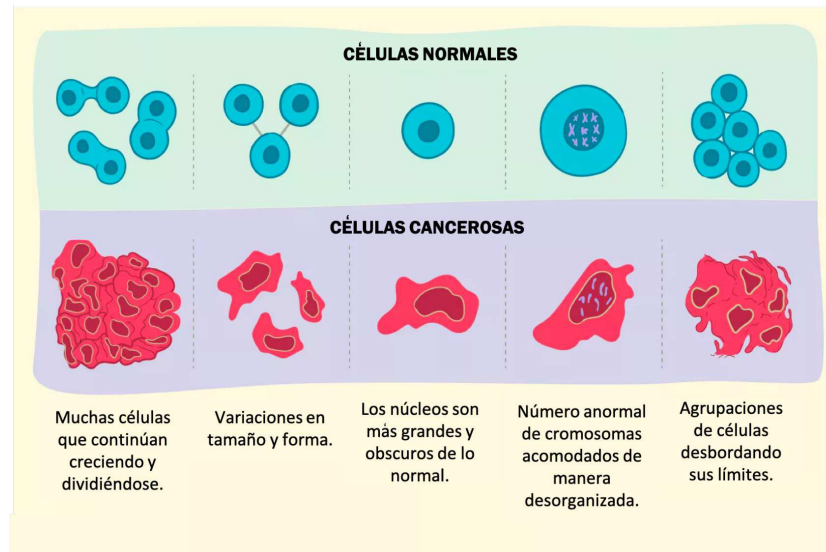


Figura 2.9: Análisis gráfico del comportamiento de células normales y malignas. Fuente: Eldridge (2019).

Algunos de estos tumores pueden resultar ser benignos (que no representan un riesgo a la salud) o maligno (que puede resultar potencialmente dañinos). En el caso de los tumores benignos, por ejemplo, no se consideran un riesgo ya que las células van creciendo muy lentamente y no invaden otro tejido ni se propagan, al contrario de los tumores malignos que tienden a extenderse a otras partes del cuerpo. En la Figura 2.9 tomada del trabajo de Eldridge (2019) se muestra un ejemplo de las formas más comunes que toman las células cancerosas y las células normales.

De la gran variedad existente de tipos de cáncer, este trabajo se concentra en el cáncer de mama. Este es considerado actualmente como uno de los cánceres más frecuentes y el que presenta mayor índice de mortalidad entre las mujeres alrededor del mundo de acuerdo con investigaciones realizadas por López (2012). De acuerdo con datos recolectados entre los años 2013 y 2015 de la organización Breast-CancerOrganization (2015) se tiene la predicción de que el 12.4% de mujeres serán

diagnosticadas con cáncer de mama en algún punto de sus vidas.

Uno de los comportamientos característicos de las células cancerosas, es el de invadir tejidos circundantes al afectado, y en el caso del cáncer de mama existen órganos como los ganglios linfáticos (órganos encargados de eliminar sustancias extrañas dentro del cuerpo) que se encuentran en el área de las axilas. Estos órganos forman parte del sistema linfático, lo cual le da a este tipo de cáncer una línea de acceso a otras partes del cuerpo, como se ilustra en la Figura 2.10.

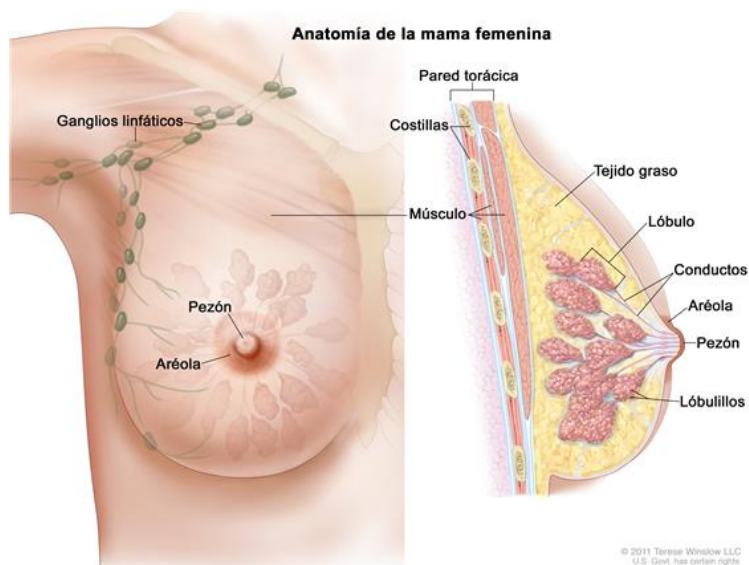


Figura 2.10: Anatomía de la mama femenina. Fuente: INC (2018)

Aunque el cáncer de mama es mucho más común en las mujeres, los hombres también pueden padecerlo, representando menos del 1% de todos los casos que se encuentran de este tipo de cáncer. Gracias a estudios realizados por CDC (2018) se sabe que aproximadamente de un 5% a un 10% de casos de cáncer de mama son producto de una anomalía genética heredada de alguno de los padres, sin embargo, aproximadamente en un rango del 80% hasta un 90% de los casos presentados son por el desgaste natural del cuerpo debido a la edad.

En la actualidad, para el cáncer de mama se desconocen las causas y/o los medios de prevención específicos para evitar padecerlo. Entre algunas de las causas conocidas para tener un riesgo más alto de presentar cáncer se muestran en la Figura

2.11, por lo que se recurren a exámenes de detección temprana para contrarrestar sus efectos. Uno de estos exámenes son los de tacto, los cuales se recomienda iniciarlos desde los 25 años de manera periódica, o desde la edad en la que se tenga su primer embarazo. Si se encuentra cualquier tipo de abultamiento o se tiene una sensación de dolor se recomienda hacer una segunda clase de examen más reconocido para detectar anomalías, como lo son las imágenes tomadas por rayos X llamadas mamografías.

Este tipo de exámenes para detección de cáncer de seno se han convertido en una práctica común en los últimos años, por lo que existe una alza en cuanto a la información disponible para los expertos, en donde la mamografía se coloca como uno de los métodos que demuestra más efectividad para detectar el cáncer de mama en sus etapas iniciales, a veces hasta tres años antes de que pueda sentirse en un examen sencillo de tacto.













 <ul style="list-style-type: none"> • Antecedentes familiares 	 <ul style="list-style-type: none"> • Ser mujer 	 <ul style="list-style-type: none"> • Envejecer 	 <ul style="list-style-type: none"> • Exposición a sustancias químicas y radioactivas 	 <ul style="list-style-type: none"> • Consumo de alcohol y tabaco 	 <ul style="list-style-type: none"> • No haber tenido hijos
 <ul style="list-style-type: none"> • Menstruación antes de los 12 años 	 <ul style="list-style-type: none"> • Factores dietéticos 	 <ul style="list-style-type: none"> • Sobrepeso y obesidad 	 <ul style="list-style-type: none"> • Primer hijo después de los 35 años 	 <ul style="list-style-type: none"> • No amamantar 	 <ul style="list-style-type: none"> • Uso de anticonceptivos o reemplazo hormonal

Figura 2.11: Posibles factores que influyen para la aparición de cáncer de mama.
Fuente: González (2019)

2.2.1 MAMOGRAFÍA

De acuerdo con información de AmericanCancerSociety. (2006) se tiene como definición que la mamografía es un examen radiográfico de la glándula mamaria que tiene como objetivo buscar por anomalías en sus tejidos. Estas se dividen en dos categorías, las mamografías selectivas de detección o las mamografías de diagnóstico.

En el primer caso también se les conoce como preventivas y pueden ser utilizadas para buscar anomalías incluso en pacientes que no presentan signos o síntomas de enfermedad. En este caso pueden ser detectados tumores que no se pueden palpar con el examen de auto exploración. En otros casos pueden encontrarse micro-calcificaciones (pequeños depósitos de calcio) que pueden indicar la presencia de cáncer en el seno, tanto como un tumor de clase benigna.

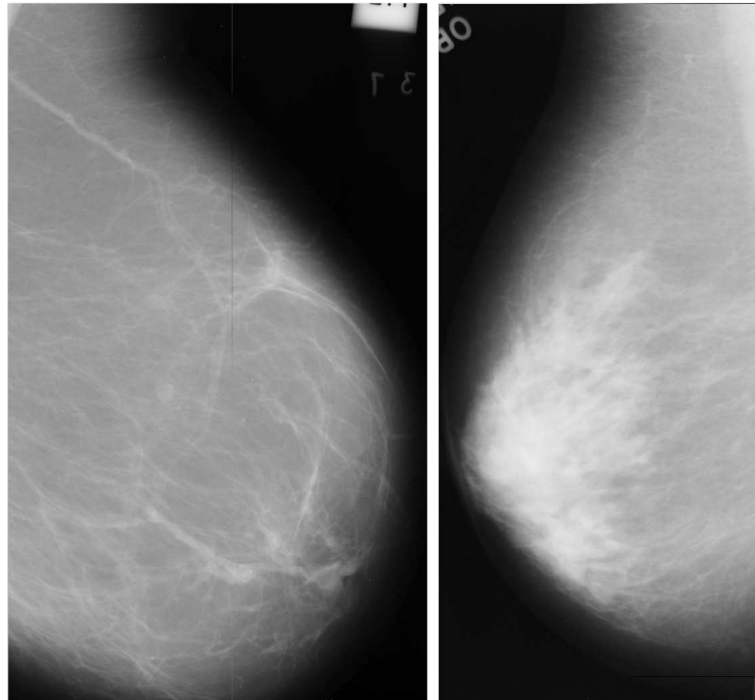


Figura 2.12: Mamografía normal (izquierda) y mamografía con anomalía (derecha).

Fuente: Suckling (1994)

En el caso de las mamografías de diagnóstico, como las que se muestran en la Figura 2.12, se usan para buscar cáncer de seno después de haber encontrado algún abultamiento u otro signo o síntoma de enfermedad.

2.2.2 MÉTODOS ESTUDIADOS PARA CLASIFICACIÓN

Algunos de los signos que los radiólogos buscan mediante el análisis de las imágenes de mamografías cuando intentan encontrar anomalías son las calcificaciones,

nódulos, masas, asimetrías en la densidad y/o distorsiones en la anatomía general del seno.

A pesar de que existen una variedad de diferencias entre los distintos tipos de cánceres, en todos los casos se comparten algunas características y propiedades, las cuales ayudan a construir un conocimiento base para muchos tratamientos e investigaciones contra el cáncer. En el caso del estudio de cáncer por medio de imágenes médicas es importante tomar en cuenta aspectos básicos y compartidos por todos los cánceres a nivel celular.

En la Figura 2.9 se puede apreciar la forma en que se comportan las células cuando se presenta una anomalía maligna. En general, la célula cancerosa se identifica por tener bordes irregulares, y sus núcleos están dispuestos de manera desorganizada. Una vez que estas características son ubicadas, el especialista puede formar su propia interpretación y consecuentemente recomendar la serie de pasos a seguir a partir de esta.

Para generalizar las diferentes interpretaciones que pueden surgir por distintos expertos, se sugiere basar sus especificaciones en el sistema de BI-RADS (Sistema de reportes y datos de imágenes de seno) como se menciona en el artículo de oncología de: *Clasificación radiológica y manejo de las lesiones mamarias* Aibar et al. (2011) "El BI-RADS es un método para clasificar los hallazgos mamográficos que actualmente se consideran como el idioma universal en el diagnóstico de patología mamaria". Este método ha permitido estandarizar la terminología de cómo categorizar las lesiones, estableciendo en cada caso el grado de sospecha y los pasos a seguir si alguna de estas categorías se presenta en el paciente.

De acuerdo con el artículo de Dorsi et al. (1998), acerca de la categorización de las lesiones mamarias, se tiene una escala de reconocimiento y manejo según el BI-RADS. En este estándar se describen las características que debe de tener la imagen para poder ser clasificada en alguna de las categorías, adicionalmente se le da una serie de pasos a seguir al experto una vez que se ha realizado la clasificación.

- **BI-RADS 0:** No concluyente por lectura incompleta.
- **BI-RADS 1:** Mama normal
- **BI-RADS 2:** Benigna.
- **BI-RADS 3:** Hallazgos probablemente benignos.
- **BI-RADS 4:** Anomalía sospechosa de malignidad.
- **BI-RADS 5:** Altamente sugestiva malignidad.
- **BI-RADS 6:** Malignidad comprobada.

Sin embargo, a pesar de existir un modelo estandarizado por el que se deben guiar los expertos para localizar y clasificar las anomalías, muchas veces existen complicaciones en el nivel de complejidad para la interpretación de dichas imágenes. Esto se debe al rango variado de apariencia en la anomalías, en las cuales no es tan sencillo identificarlas, ya sea por las características del tejido mamario de cada paciente, o por la diferencia en la calidad de las imágenes.

En el aspecto de diferencias de tejido mamario, el problema que se describe en el artículo de Bozek et al. (2009) recae en que las lesiones son demasiado pequeñas para ser apreciadas, o porque la mama está construida por tejidos muy similares entre sí, por lo que se confunde el tejido canceroso con el sano.

Por otra parte, la interpretación visual para determinar la lesión realizada por un experto depende mucho de la experiencia que este tenga. De acuerdo con Cheng et al. (2006) existe un porcentaje que va desde un 65% a un 75% en cuanto a exactitud de diagnóstico incluso dentro de los radiólogos más especializados.

Eliminar la incertidumbre en este tipo de herramientas se vuelve muy importante cuando tomamos en cuenta que un falso negativo significa que una anomalía va a pasar desapercibida, o que en caso contrario un falso positivo pedirá el uso innecesario de una biopsia.

2.2.3 MÉTODOS AUTOMATIZADOS PARA CLASIFICACIÓN

Para cualquier tipo de cáncer, uno de los factores más importantes para que el paciente tenga más posibilidades de supervivencia, es la de encontrar a tiempo las anomalías presentes en el cuerpo, y el primer problema que se encuentra en este tema de detección, es la del factor humano ya que el resultado de un diagnóstico correcto depende en gran medida de la experiencia del radiólogo que se encuentra analizando las imágenes, e incluso la calidad de este criterio profesional puede verse disminuido por circunstancias externas como el cansancio en la etapa de revisión (la inspección de las imágenes es un proceso extenuante y que consume mucho tiempo), la calidad de la imagen de mamografía, etc.

Hoy en día existen sistemas de diagnóstico asistido por computadora, mejor conocidos como *Computer Aided Diagnostic* (CAD's por sus siglas en inglés), al igual que los sistemas de detección asistida por computadora, llamados *Computer Aided Detection* (CADx's por sus siglas en inglés), que se utilizan como herramientas auxiliares en la tarea de interpretación de las mamografías para poder encontrar las masas y subsecuentemente clasificarlas. Estas herramientas, en conjunto con la opinión experta, agilizan el proceso de detección y diagnóstico, lo cual es de vital importancia para los pacientes que requieren llevar a cabo procesos de tratamientos oportunos.

2.2.4 DETECCIÓN ASISTIDA POR COMPUTADORA

Como se menciona en secciones anteriores, la mayor parte de los casos en los que las mamografías llegan a ser clasificadas de manera incorrecta cae en los hombros del nivel de experiencia que tenga el radiólogo encargado de hacer el estudio, e incluso entonces su desempeño puede verse disminuido por fatiga o cantidad excesiva de trabajo. En el artículo realizado por Cheng et al. (2003) se tiene como porcentaje que alrededor de un 10 % a un 30 % de lesiones de mama son clasificadas como falsos

negativos o falsos positivos durante el estudio de la imagen hecho por un experto.

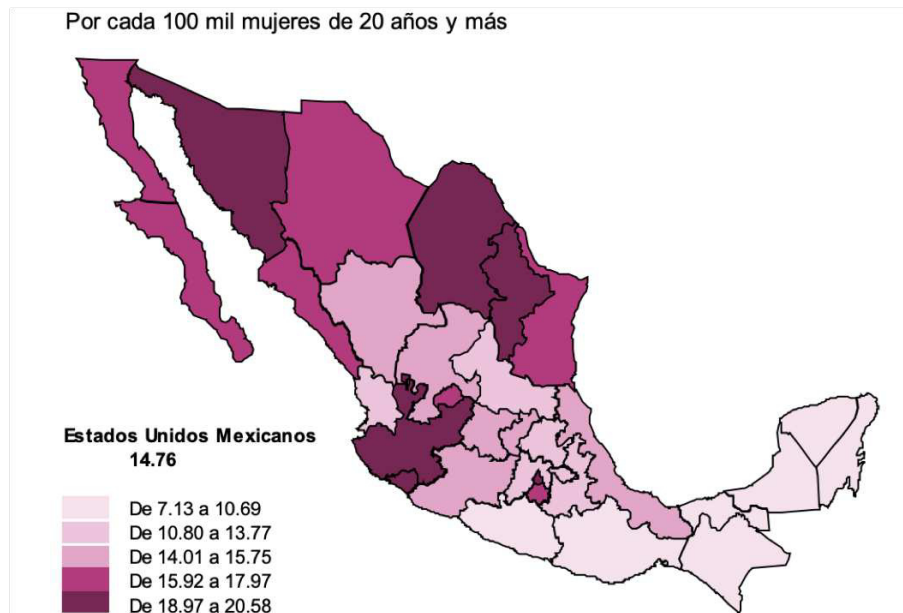


Figura 2.13: Mapa de México representando los estados con mayor índice de mortalidad por cáncer de mama. Fuente: INEGI (2019)

De acuerdo a estadísticas realizadas en México, en el año 2014 se estimó una cantidad de 700 mastógrafos, y de estos su mayoría se encuentran en el estado de México (ver artículo completo publicado por la UNAM (2018)). Estos números, comparados con la porción de casos nuevos que se presentan cada año, hace que la cantidad de expertos para catalogar estas imágenes se vuelva insuficiente, ya que estadísticamente de forma anual, alrededor de seis mil mujeres en México pierden la batalla contra el cáncer de mama, ubicando esta patología como una de las principales causas de muerte entre mujeres a nivel nacional, según datos del INEGI (Instituto Nacional de Estadística y Geografía) tomados en el año 2015. Tomando en cuenta la cifra, al día se registrarían 16 muertes, en promedio por diagnósticos tardíos de acuerdo con estudios realizados por el INEGI (2019), como se puede ver en la Figura 2.13.

Hoy en día existen herramientas de procesamiento de imágenes que ayudan a los radiólogos a mejorar el porcentaje de exactitud de su diagnóstico. Dichas herramientas, llamados sistemas de detección asistida por computadora (CAD), siguen

un patrón de pasos fijos para ser considerados como un CAD, mencionados en la Figura 2.14.⁴



Figura 2.14: Diagrama de bloques de un sistema de CAD para mamografías. Fuente: Cheng et al. (2003).

En el diagrama de bloques se puede apreciar en la Figura 2.14 representa el tipo de metodología que se utiliza como base para ayudar a la detección y diagnóstico de cualquier tipo de imagen médica. La parte de pre-procesamiento se utiliza para digitalizar y mejorar la calidad de las imágenes a utilizar.

Después se pasa al bloque de detección, en donde se localizan las regiones de interés, para posteriormente poder realizar la segmentación, que se encarga de extraer y seleccionar las características necesarias para dividir la clasificación de dicho elemento como benigno, maligno o normal.

Entre bloque y bloque el proceso puede resultar largo, ya que existen diferentes formas selección de herramientas de procesamiento, como a las distintas maneras de realizar el paso de segmentación. Se sabe que es bastante difícil distinguir entre una anomalía maligna y una benigna, y en la actualidad existen diferentes métodos que se han tomado en cuenta en el tema de los clasificadores, siendo las redes neuronales una de las más recientes.

Uno de los objetivos que impulsa este proyecto de tesis es el de encontrar herramientas más efectivas para la detección, clasificación y diagnóstico de enfermedades, y una de las herramientas que muestra mayor promesa para poder cumplir este objetivo, son las redes neuronales.

⁴Imagen tomada de Cheng et al. (2003) traducida al español.

2.3 REDES NEURONALES CONVOLUCIONALES EXISTENTES

Como se mencionó anteriormente, las redes neuronales convolucionales son mayormente utilizadas para la clasificación de imágenes, ya que sus características las hacen una herramienta ideal para problemas de visión computacional.

Por los inicios del año 2000 las redes neuronales en general perdieron un poco su auge debido a que los niveles de procesamiento que se tenían en las computadoras de ese entonces no eran lo suficientemente fuertes como para hacer una gran diferencia en el resultado de una red neuronal con muchas imágenes.

Hoy en día los ordenadores cuentan con núcleos y procesadores más potentes, que han provocado que las redes neuronales vuelvan a tener un alce en cuanto a temas de investigación. Y al igual que este alce en el área de calidad de computadoras, también en años recientes se experimentó un aumento de información de una gran variedad de campos.

Empresas grandes como *Google*, *Amazon*, etc. se dieron cuenta del potencial que representa esta carga de información, y se dieron a la tarea de encontrar el beneficio en su relación con el cliente. Con esto surgieron competencias entre grupos de programadores para hacer uso de las redes neuronales, el *Deep learning* y la minería de datos para sacarle provecho a toda esta información.

De competencias realizadas por Deng et al. (2009) resultaron algunas de las redes neuronales convolucionales, que hasta la fecha, siguen siendo tema de investigación en los círculos de programadores de temas de inteligencia artificial. Una de las contiendas más famosas para la clasificación de imágenes es la llamada *ImageNet*.

Esta competición se llevó a cabo desde el año 2010 al 2017, fecha en la que se pasó a la plataforma de *Kaggle* por Goldbloom et al. (2019). Esto trajo consi-

go un cambio en el modo en que se implementaban las redes neuronales gracias a la eficiencia que mostraron los procesadores gráficos (GPU's) al igual que las más recientes funciones de activación, regularizaciones de sobre-ajuste de datos, y una manera efectiva de aumentar los datos disponibles para entrenamiento.

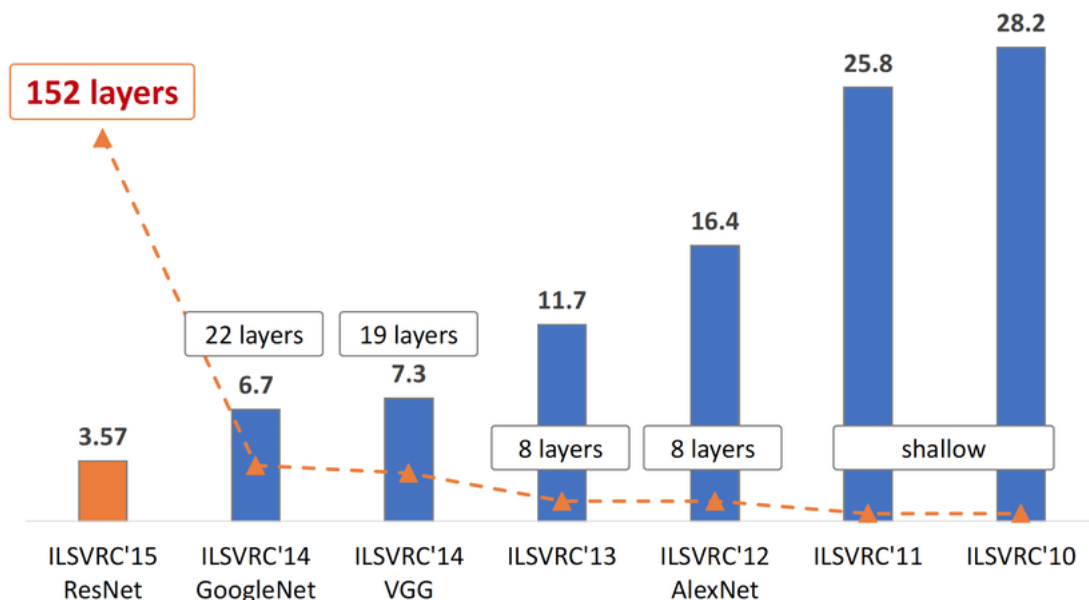


Figura 2.15: Diferencias entre las redes ganadoras por cada año de competencia del ImageNet. Fuente: Deng et al. (2009).

Algunas de las redes que surgieron de esta competencia, como las que se muestran en la Figura 2.15 se llevaron a la investigación dentro del campo de la medicina, ya que, para poder entrenar una red desde cero es necesario una cantidad grande de imágenes anotadas, lo que significa que deben de tener una clasificación establecida para poder entrenar la red con ellas, y se creía que la transferencia de aprendizaje era la manera de mejorar las herramientas de diagnóstico en imágenes médicas, ya que a pesar de que existe un conjunto de información grande, de momento no es del todo simple su diagnóstico, ya que hay pocos especialistas para anotarlas.

La respuesta a esto fue el aumento de datos anotados existentes. La augmentación de una base de datos es una práctica común en el entrenamiento de redes neuronales con información limitada. Por información limitada en este trabajo se

hace referencia a la escasa cantidad de imágenes anotadas por expertos con la cual se puede entrenar la red. Lo que se hace en esta augmentación es "modificar" la imagen de manera que la red la interprete como si fuera otra totalmente diferente. De esta forma se "engaña" a la red a pensar que se tiene más datos que con los que en realidad se cuenta. Estas modificaciones no son más que manipulaciones de la imagen, como girarla, transponerla, agregarle filtros, etc.

Como se menciona en la parte de introducción, las herramientas computacionales auxiliares para el diagnóstico y detección de enfermedades se han ido desarrollando a lo largo de los años. Algunas de las redes neuronales convolucionales que se han estudiado para evaluar su desempeño en la clasificación de imágenes médicas se describen a continuación.

2.3.1 LENET (1998)

La arquitectura de red LeNet creada por LeCun et al. (1998), representada en la Figura 2.16 es una de las clásicas utilizadas para las redes neuronales convolucionales. Los investigadores Yann LeCun, Leon Bottou, Youshua Bengio y Patrick Haffner propusieron este tipo de arquitectura de red en el año 1990, originalmente diseñada para reconocer caracteres escritos a mano y en máquina. Una visión general sobre esta red y sus características puedes ser encontradas en el artículo realizado por Rizwan (2018).

La estructura de la red LeNet-5 consiste en dos capas convolucionales y capas *Average pooling*. La diferencia entre las *Maxpooling* y las *Average pooling*, es que, cuando se les pasa los parches de imagen, el *Maxpooling* toma el valor más grande de la matriz, mientras que el *Average pooling* suma todos los elementos de la matriz y los divide entre su número total. Al finalizar la red, se tiene una capa de *Softmax*. Es importante también mencionar que todas las capas tienen una función de activación de tanh o tangente hiperbólica.

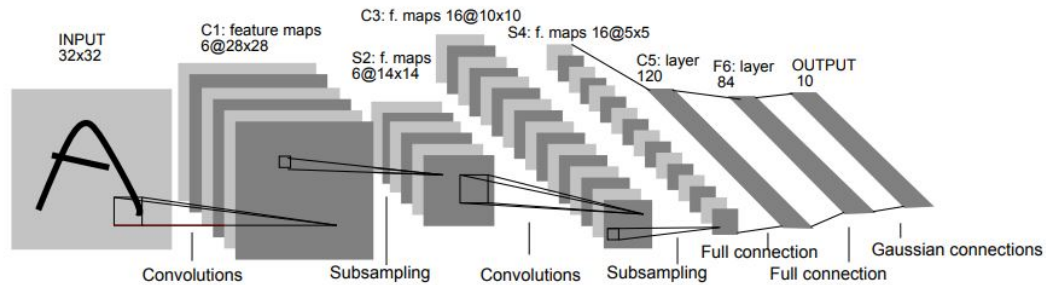


Figura 2.16: Arquitectura utilizada para la estructura de la red neuronal convolucional LeNet-5. Fuente: Max and Pechyonkin (2018)

Tomando en cuenta la información dada por Calvo (2018) la función de tangente hiperbólica transforma los valores que recibe en una escala de -1 a 1. Es muy similar a la función sigmoide, tiene como características que su tiempo de convergencia es lento, y que muestra un mejor desempeño en redes recurrentes.

Esta red no se utilizó para el reto de *ImageNet*, por lo que su experimentación se llevó a cabo con el conjunto de datos llamada MNIST, que consiste en números escritos a mano en imágenes en blanco y negro. La habilidad de procesar imágenes con mayor resolución requiere una mayor cantidad de capas convolucionales, por lo cual esta red quedó restringida por su disponibilidad a recursos de procesamiento.

2.3.2 ALEXNET (2012)

AlexNet creada por Krizhevsky et al. (2012), representada en la Figura 2.17 es una red neuronal que fue la ganadora de la competición *ImageNet* en el año 2012 por una diferencia de 15.3% contra el segundo mejor que fue de 26.2% en margen de error. Creada por Alex Krizhevsky, Ilya Sutskever y Geoffrey E. Hinton. Esta red neuronal convolucional fue entrenada para clasificar las 1.3 millones de imágenes de alta resolución contenidas en el conjunto de entrenamiento del *ImageNet*, dividiéndolas en 1000 clases distintas.

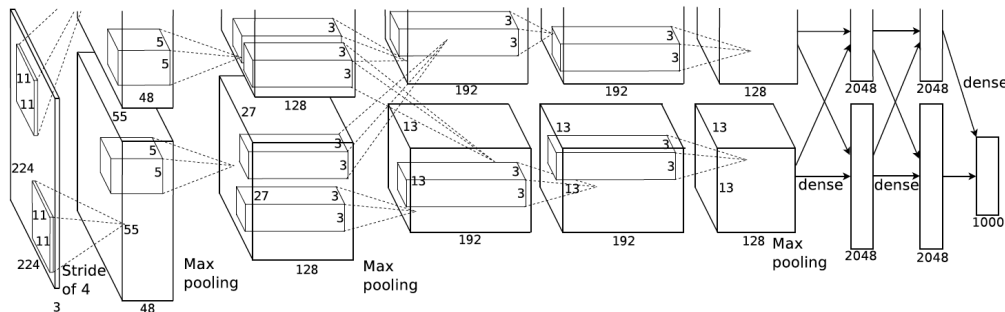


Figura 2.17: Arquitectura utilizada para la estructura de la red neuronal convolucional AlexNet. Fuente: Krizhevsky et al. (2012)

La red neuronal consiste en 60 millones de parámetros y 500 000 neuronas. Dentro se tienen cinco capas convolucionales, algunas de las cuales se conectan directamente con una capa de *Maxpooling*, y dos capas conectadas de manera global, terminando con un *Softmax* con 1000 caminos finales. Para evitar el sobre-ajuste de datos se utilizó un método de regularización conocido como *Dropout*, que prueba ser bastante eficiente, pero sube el nivel de procesamiento y el tiempo de computación.

La técnica del *Dropout* se utiliza como herramienta de regularización y se desarrolla seleccionando de manera aleatoria neuronas que se ignoran durante la etapa de entrenamiento de la red. *Dropout*, introducido inicialmente por Srivastava et al. (2014) se traduce crudamente como deshechar, que es lo que pasa con estas neuronas que se seleccionan. Al finalizar el entrenamiento, los pesos de la red deben de estar familiarizados con este tipo de falta de información que se crea al ignorar algunas de las neuronas, lo que debería hacer más robusta la red en general.

2.3.3 VGGNET (2014)

VGGNet, creada por Simonyan and Zisserman (2014), representada en la Figura 2.18 es una abreviación para el grupo que la inventó, *Visual Geometry Group* de la Universidad de Oxford. En la tarea de clasificación mejoró significativamente

los resultados de la red de AlexNet y la ZFNet (ganadora del año 2013). Este tipo de red tiene diferentes versiones, siendo las más destacadas en el ámbito de reconocimiento de imágenes la VGG-16 y la VGG-19. Una descripción más detallada sobre esta arquitectura de red neuronal puede encontrarse en el artículo realizado por Tsang (2018).

La red neuronal VGG se caracteriza por su simplicidad, ya que, a diferencia del AlexNet, esta red utiliza una arquitectura más compacta de 3x3 capas convolucionales, puestas unas sobre de otras, añadiendo con esto un nivel mayor de profundidad a la red. Dos de estas capas, totalmente conectadas le sigue un clasificador tipo *softmax*.

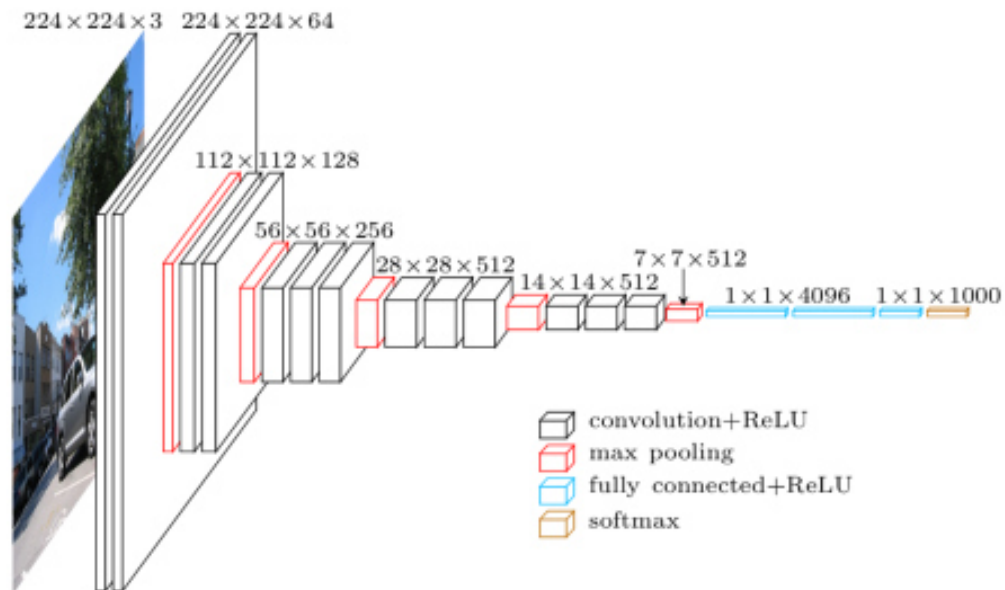


Figura 2.18: Arquitectura utilizada para la estructura de la red neuronal convolucional VGG. Fuente: Krizhevsky et al. (2012)

Es una de las redes más utilizadas en la comunidad por su capacidad de extraer características relevantes de las imágenes, sin embargo, esta red consiste en más de 140 millones de parámetros, por lo que podrá resultar un poco difícil de manejar.

Entre los trabajos que se llevaron a cabo dentro de esta distribución de red neuronal fue la de la experimentación en cuanto al número de capas utilizadas en

la arquitectura, como el realizado por Tsang (2018) demostrando que el VGG-16 mejora el rendimiento y la tasa de error al añadir capas a la arquitectura VGG-11, mientras que la arquitectura VGG-19 empeora sus resultados al hacer lo mismo, por lo que los autores dejaron de añadir capas.

2.3.4 GOOGLNET (2014)

GoogLeNet, creado por Szegedy et al. (2015) y representado en la Figura 2.19 fue el ganador del concurso *ImageNet* del año 2014, alcanzando un nivel de error de 6.67%, que años después fue capaz de mejorar y bajarlo a un 5.1%. Su exactitud llegó acercarse bastante a la de un ser humano.

Esta red se inspiró en la arquitectura de LeNet-5, pero añadió algunos elementos que aumentaron el nivel de profundidad de la red. Utilizó la normalización de parches y el optimizador de RMSProp. Redujo también, drásticamente la cantidad de parámetros utilizados gracias a su arquitectura que se basa en convoluciones pequeñas. En comparación con el AlexNet y sus 60 millones de parámetros, la GoogleNet tiene 4 millones.

Otra diferencia de redes como la VGGNet, ZFNet y la AlexNet, es que se utiliza el *Global average pooling* al final de la red en lugar de utilizar las capas de conexión final tradicionales. Lo que caracteriza esta red con respecto a otras es la creación de su módulo llamado *Inception module* que basa su funcionamiento en tener diferentes tipos y tamaños de convoluciones para la misma entrada, y guarda sus diferentes valores de salida como parte del entrenamiento, una descripción más detallada de esta arquitectura de red se puede consultar con el autor Tsang (2018).

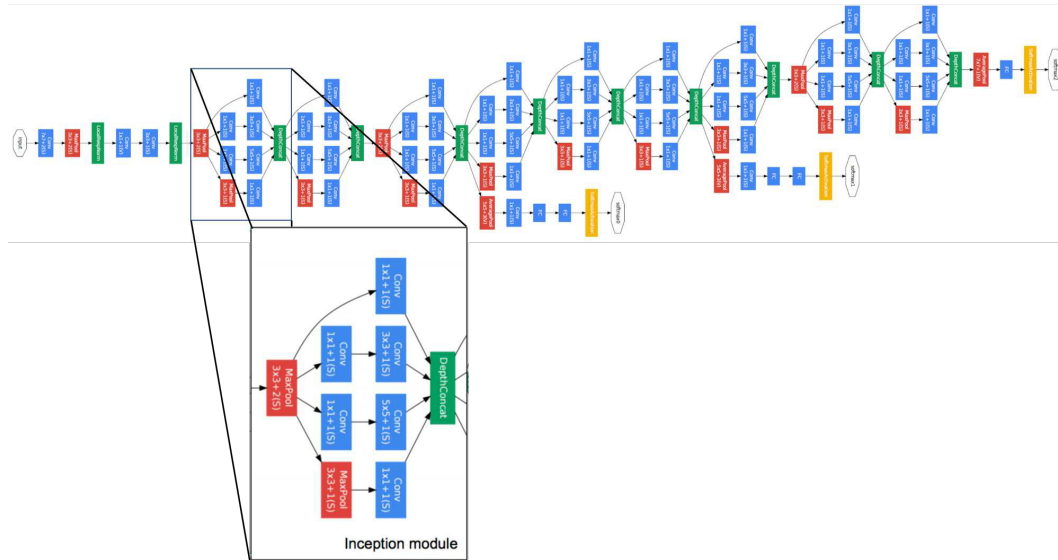


Figura 2.19: Arquitectura utilizada para la estructura de la red neuronal convolucional GoogleNet. Fuente: Tsang (2018)

2.3.5 RESNET(2015)

ResNet fue creada por He et al. (2015), representada en la Figura 2.20. Llamada de esa manera por su nombre en inglés de *Residual Neural Network* (red neuronal residual) realizada por el investigador Kaiming. Él propuso una nueva arquitectura basada en pasar conexiones y características más potentes para normalizar los parches de imágenes. El pase de conexión se pueden describir como unidades cerradas o unidades recurrentes cerradas que han sido aplicadas recientemente en las redes neuronales recurrentes.

Gracias a esta nueva arquitectura de red, le fue posible entrenar una red neuronal con 152 capas, y aún así tener una menor complejidad computacional a la red de VGG. Tuvo un porcentaje de 3.57% en rango de error, lo cual logró vencer el nivel de rendimiento de un ser humano por primera vez en una herramienta de procesamiento y reconocimiento de imágenes.

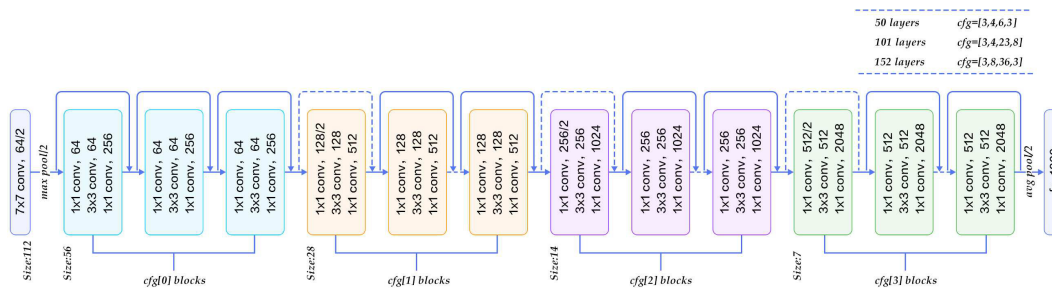


Figura 2.20: Arquitectura utilizada para la estructura de la red neuronal convolucional ResNet. Fuente: He et al. (2015)

2.4 ESTADO DEL ARTE

En esta sección se compila la investigación realizada del estado del arte para herramientas computacionales de diagnóstico y detección en el área médica. Son pocas las investigaciones encontradas que se concentren más en la clasificación inicial en la que se enfoca este trabajo, siendo el enfoque de la mayoría un diagnóstico de la anomalía ya localizada.

Se incluyen también las publicaciones científicas que utilizaron la metodología tradicional, al igual que las que utilizan herramientas alternativas a las redes neuronales convolucionales. Con esto se quiere tomar una referencia de cuál es un umbral aceptable de exactitud. Se describen por el nombre del artículo en el que fueron publicados los descubrimientos, la fecha de su publicación, el equipo de trabajo involucrado, las herramientas y metodologías utilizadas para desarrollarlo y, en caso de contenerla, se agrega la exactitud encontrada después de su experimentaciones.

En muchos casos se trabaja con las mismas bases de datos utilizadas en este trabajo, sin embargo existen algunos casos en los que se cuenta con una base de datos propia. En estos casos se realizó una investigación de estos conjuntos, más ninguno de ellos aparece con un enlace público.

En general todos los títulos de los artículos se dejaron en el idioma en el que

se encontraron para evitar malas traducciones.

Tabla 2.1: Artículos científicos relacionados

Artículo	Descripción
<p>An ensemble of Fine - Tuned Convolutional Neural Networks for Medical Image Classification. Autores: Kumar et al. (2017)</p>	<ul style="list-style-type: none"> <li data-bbox="753 491 1192 527">▪ Base de datos: <i>ImageCLEF</i>. <li data-bbox="753 575 1422 722">▪ Utiliza el 90 % de base de datos para conjunto de entrenamiento, 10 % para conjunto de validación. <li data-bbox="753 770 1422 980">▪ Herramientas utilizadas: <i>Transfer Learning</i>, <i>SVM (Suport Vector Machine)</i> y <i>Fine tuning</i> de redes neuronales existentes como AlexNet, GoogLeNet. <li data-bbox="753 1029 1422 1176">▪ Exactitud alcanzada: <i>Transfer learned + SVM: 96 %</i>, <i>Fine tuned + Softmax: 94 %</i>, <i>Fine tuned + SVM: 96 %</i>.

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
<p>Deep convolutional Neural Networks for image classification, a comprehensive review. Autores: Rawat and Wang (2017)</p>	<ul style="list-style-type: none"> ▪ Base de datos: Variadas. (Imágenes a color, blanco y negro, escrituras, Alta definición) ▪ Hace un estudio de redes neuronales con arquitecturas simples. ▪ Herramientas utilizadas: Canonical view DCNN: 7 Capas, Exactitud: 96 %. DeepFace: 8 capas, Exactitud: 97 %. DeepID: 7 capas, Exactitud: 97 %. FaceNet: 22 capas, Exactitud: 99 %
<p>DL in Mammography. Diagnostic Accuracy of a Multipurpose Image Analysis Software in the Detection of Breast Cancer. Autores: Becker et al. (2017)</p>	<ul style="list-style-type: none"> ▪ Base de datos: <i>Breast Cancer Digital Repository</i> (BCDR) 143 pacientes diagnosticados con cáncer de mama, dando un total de 286 casos de estudio. ▪ Hace una investigación con redes neuronales artificiales, separando sus resultados con diferente tipo de densidad de tejido mamario. ▪ Herramientas utilizadas: Deep learning Artificial Neural Networks (DANN) ▪ Exactitud alcanzada: 82 %

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
<p>Probabilistic Visual Search for Masses Within Mammography Images using Deep Learning. Autores: Ertosun and Rubin (2015)</p>	<ul style="list-style-type: none"> ▪ Base de datos: DDSM <i>Digital Database for Screening Mammography</i> ▪ Dividiendo los conjuntos de entrenamiento, validación y prueba en 80 %, 10 % y 10 %. ▪ Herramientas utilizadas: En este artículo experimentan con las arquitecturas de red de AlexNet, GoogLeNet y VGG Net-16, mostrando el resultado de clasificación de mamografía con y sin anomalía. ▪ Exactitud alcanzada: 84 %, 85 % y 82 % respectivamente.
<p>A preliminary study on breast cancer risk analysis using deep neural network. Autores: Sun et al. (2016)</p>	<ul style="list-style-type: none"> ▪ Base de datos: <i>Full Field Digital Mammography</i> (FFDM) 420 casos con un total de 840 mamografías disponibles, haciendo un recorte de las imágenes a 52×52. ▪ Herramientas utilizadas: El experimento es de una red neuronal convolucional con 4 capas y una capa de <i>Softmax</i>. ▪ Exactitud Alcanzada: 70 %

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
Automatic medical image classification and abnormality detection using KNN. Autores: Ramteke and Monali (2012)	<ul style="list-style-type: none"> ▪ Base de datos: No menciona la fuente, pero se sabe que son imágenes de escáner CT del cerebro con y sin anomalías. ▪ Herramientas utilizadas: Utiliza la herramienta de <i>K-Nearest Neighbour</i> para hacer una clasificación de anomalías en imágenes de escaneo de cerebro. ▪ Exactitud Alcanzada: 80 %
Using machine learning algorithms for breast cancer risk prediction and diagnosis. Autores: Asri et al. (2016)	<ul style="list-style-type: none"> ▪ Base de datos: <i>Wisconsin Breast Center</i> con 644 instancias de imágenes con y sin anomalías. ▪ Comparación de 4 herramientas para la clasificación de la base de datos. ▪ Herramientas y exactitud: Árbol de desición C4.5 (94.5%), Máquina de soporte vectorial(95%), <i>Naive Bayes</i> (96%) y <i>k-Nearest Neighbour</i> (94.5%)

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
Breast Cancer classification using deep belief networks. Autores: Abdel-Zaher and Eldeib (2016)	<ul style="list-style-type: none"> ▪ Base de datos: <i>Wisconsin Breast Cancer dataset</i> (WBCD) Repositorio de datos de anomalías de pacientes. ▪ Uso de datos de texto sobre características de biopsia. ▪ Exactitud Alcanzada: 99.5 %
Mammogram classification using CNN. Autores: Zhou et al. (2017)	<ul style="list-style-type: none"> ▪ Base de datos: <i>MIAS Mini Mammogram Image Analysis Society</i> y <i>DDSM Digital database for screening Mammography</i> ▪ Herramientas utilizadas: Creación de una red neuronal convolucional. Dos capas ocultas con un <i>Maxpooling</i> en ambas. Contiene 32 y 64 <i>Kernels</i> en cada una. Valor de <i>Dropout</i> de 0.5. Capa final de <i>Flatening</i> de 1024 neuronas. Entrenamiento de 5000 pasos. ▪ Exactitud alcanzada: 59.6 % solo malignos. 60.9 % general

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
<p>Digital mammographic tumor classification using transfer learning from DCNN. Autores: Huynh et al. (2016)</p>	<ul style="list-style-type: none"> ▪ Base de datos: Propia y no pública. Obtenida del Centro Médico de la Universidad de Chicago, aprobado bajo Revisión Institucional. 219 mamografías con lesiones confirmadas por expertos. ▪ Herramientas utilizadas: Utilizan tres métodos de modelos. Red neuronal convolucional pre entrenada (SVM1), Extracción analítica de características(SVM2), Clasificador ensamblado (SVM1 y SVM2). ▪ Exactitud alcanzada: SVM1: 81 %, SVM2: 81 %, y Clasificador ensamblado: 86 %.
<p>Breast image analysis for risk assessment, detection, diagnosis and treatment of cancer. Autores: Giger et al. (2013)</p>	<ul style="list-style-type: none"> ▪ Revisión de diferentes herramientas para el diagnóstico y detección de cáncer, su índice de exactitud, y cómo han ayudado a radiólogos a mejorar su nivel de exactitud en la detección y diagnóstico.

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
Breast Image Feature Learning with Adaptive Deconvolutional Networks. Autores: Jamieson et al. (2012)	<ul style="list-style-type: none"> ▪ Base de datos: UCMC <i>University of Chicago Medical Center</i> Imágenes de lesiones de seno. FFDM <i>Diagnostic full field digital mammography</i>. Imágenes de ultrasonidos. ▪ Herramientas utilizadas: Se explora el uso de ADN (Adaptive Deconvolutional Networks). Estas redes fueron propuestas recientemente, usando modelos sin supervisión que descomponen una imagen mediante la convolución de <i>Sparse coding</i> y el <i>Maxpooling</i>. ▪ Exactitud alcanzada: 95 % general. 83 % ultrasonidos.
How transferable are features in deep neural networks? Autores: Yosinski et al. (2014)	<ul style="list-style-type: none"> ▪ Explicación a detalle de cuáles son las características que cambian cuando se hace un <i>Transfer learning</i> a una red neuronal pre-entrenada. Ventajas y desventajas de utilizar el <i>Transfer learning</i> y en cuáles problemas se desempeñan mejor.

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
<p>Xnor-net: Imagenet classification using binary convolutional neural networks. Autores: Rastegari et al. (2016)</p>	<ul style="list-style-type: none"> ▪ Base de datos: <i>ImageNet</i> (Imágenes. no médicas) ▪ Herramientas y exactitud: En este artículo realizan una herramienta de clasificación de imágenes normales (reto <i>Imagenet</i>) combinando arquitecturas de red, con redes binarias y Xnor. ResNet18 teniendo una exactitud de 69.3% y 89.2% respectivamente, y GoogLeNet un 71.3% y 90% en las dos combinaciones.
<p>Modality Classification and Concept Detection in Medical Images Using Deep Transfer Learning. Autores: Singh et al. (2018)</p>	<ul style="list-style-type: none"> ▪ Base de datos: Imágenes de diferente formato sacadas de la base de datos conjunta del NIH (<i>National Institute of Health</i>). ▪ Clasificador de <i>transfer learning</i> que ordena imágenes por el tipo al que representa, siendo estos MRI, CT, X ray, Ultrasonido, etc. ▪ Herramientas y exactitud: VGG16 62%, VGG19 98.18%, ResNet50 90%, Inception v3 99.45%, Xception 98.36%, MobileNet 98.73%, Inception ResNetv2 98.18%.

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
<p>Unsupervised Deep Transfer Feature Learning for Medical Image Classification. Autores: Ahn et al. (2019)</p>	<ul style="list-style-type: none"> ▪ Base de datos: <i>ImageCLEF</i>. ▪ Herramientas utilizadas: Se utiliza una metodología en la que colocan una capa convolucional encima de una red pre-entrenada de un dominio distinto. La red seleccionada es la de AlexNet. ▪ Exactitud alcanzada: Se muestran los resultados de esta y otras metodologías, teniendo el tercer mejor resultado de exactitud (81.33%) sobrepasado solo por dos metodologías del conjunto de Alexnet y GoogLeNet (82.48%)y GoogLeNet y ResNet (83.14%).
<p>Computationally-efficient wavelet-based characterization of breast tumors using conventional B-mode ultrasound images. Autores: Mahmoud et al. (2019)</p>	<ul style="list-style-type: none"> ▪ Bases de datos: WAFI (<i>Woman and Fetal Imaging. National Cancer Institute in Egypt</i>) ▪ Herramientas y exactitud: Se utiliza el método de <i>wavelet packet transform</i> en conjunto con características como <i>Support Vector Machine</i> 96.9%, <i>k- Nearest Neighbour</i> 96.3%, <i>Árbol de decisión</i> 89.1%, <i>Naive Bayes</i> 87.2%.

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
Breast Cancer Classification with Statistical Features of Wavelet Coefficient of Mammograms. Autores: Lahamage and Patil (2014)	<ul style="list-style-type: none">▪ Base de datos: Mini Mias.▪ Herramientas utilizadas: Utilizan los pasos comunes de un CAD (<i>Computer Aided Detection</i>), solo que la parte de extracción de características utilizan el método de <i>Wavelet coefficient</i>. Al final también realizan clasificación para saber que tipo de anomalía es.▪ Exactitud alcanzada: 90.4 %.

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
<p>Using generative adversarial networks and transfer learning for breast cancer detection by convolutional neural networks. Autores: Lahamage and Patil (2014)</p>	<ul style="list-style-type: none"> <li data-bbox="753 394 1427 485">▪ Base de datos: DDSM (<i>Digital Database for Screening Mammography</i>). <li data-bbox="753 537 1427 1024">▪ Herramientas utilizadas: Se utiliza el método GAN. Modelo de red neuronal que aprende la probabilidad de la distribución de los datos reales y crea datos prueba simulados con una distribución parecida. Esto con el fin de aumentar la base de datos existente y evitar el <i>overfitting</i>. Se utiliza en conjunto con redes pre-entrenadas. Clasificación entre imágenes con y sin anomalía. <li data-bbox="753 1077 1427 1283">▪ Exactitud alcanzada: VGG16 + ORG ROIs: 91.5%, VGG16 + ORG ROIs: + GAN: 92.1% , CNN (Convolutional Neural Network) de cero + doble GAN: 98 %

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
<p>Investigation of the effect of image resolution on automatic classification of mammary gland density in mammography images using deep learning. Autores: Oshima et al. (2019)</p>	<ul style="list-style-type: none"> <li data-bbox="753 394 1427 541">▪ Base de datos: <i>University of Medica Science, Department of Radiological Technology.</i> 1106 imágenes anómalas. <li data-bbox="753 594 1427 1031">▪ Herramientas utilizadas: Separación de 4 tipos de anomalías (BI-RADS). Uso de algoritmo propio para pre-procesamiento. Extracción de características por medio de arquitectura de red de AlexNet, utilizando optimizador Adam y <i>Dropout</i> de 0.5. Se compara con diferente tamaño de imagen, utilizando imágenes normales y pre-procesadas por el algoritmo propuesto. <li data-bbox="753 1083 1427 1346">▪ Exactitud alcanzada: Tamaño de la imagen – exactitud: (474x354 – 90.4% – 89.5%) (316x236 – 90.8% – 91.5%) (237x177 – 90.2% – 90.6%) (158x118 – 89.5% – 89.0%) (79x59 – 87.3% – 89.3%).

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
<p>Batch normalization: Accelerating deep network training by reducing internal covariate shift. Autores: Ioffe and Szegedy (2015)</p>	<ul style="list-style-type: none"> ▪ Estudio sobre el problema llamado <i>Internal Covariate Shift</i> (ICS) en el conjunto de entrenamiento. Se utiliza el <i>batch normalization</i> en cada uno de los batch de entrenamiento, lo cual hace innecesario el uso de <i>Dropout</i>. Se prueba en herramientas de clasificación para su comparación. El error encontrado es de 4.82 %, siendo el anterior reportado de 4.9 %.
<p>Deep learning for breast cancer classification with mammography. Autores: Yang et al. (2019)</p>	<ul style="list-style-type: none"> ▪ Base de datos: Breast Cancer Digital Repository, INBreast y CBIS-DDSM. ▪ Herramientas utilizadas: En este artículo se desprecian las imágenes clasificadas en el nivel 0 BI-RADS, ya que son consideradas como anomalías benignas, para entrenar una red más específica en zonas de BI-RADS como la 3 y 4, que son las que presentan las anomalías malignas. ▪ Exactitud alcanzada: Se compara contra expertos humanos en la clasificación de BI-RADS nivel 3 y 4. Expertos: 54.1 %, Herramienta 75.7 %.

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
<p>Medical image retrieval using Resnet-18 for clinical diagnosis. Autores: Ayachamy et al. (2019)</p>	<ul style="list-style-type: none"> ▪ Base de datos: Las imágenes se recolectan de diferentes bases de datos. BRATS, LITS, PROMISE, Head and Neck dataset, base de datos extra de la Universidad de Cornell y un repositorio TCIA. ▪ Herramientas utilizadas: Se utiliza la red pre entrenada de ResNet-18 para clasificar una base de datos multimodal, clasificando cada una de las imágenes por regiones. ▪ Exactitud alcanzada: En promedio con todas las regiones se obtiene una clasificación del 92 %. Esta herramienta se dispone para ayudar al entrenamiento de radiólogos para diagnóstico de imágenes médicas.
<p>National performance benchmarks for modern diagnostic digital mammography. Autores: Sprague et al. (2017)</p>	<ul style="list-style-type: none"> ▪ Este artículo da el porcentaje aceptable para cada uno de los valores de desempeño mínimo que deben de tener los CADs. Detección de cáncer 34.7 de 1000 (95 % de confianza), Falsos negativos 4.8 de 1000. Sensibilidad 87.8 %, Especificidad: 90.5 %.

Continúa en la próxima página

Tabla 2.1 – *Continuación*

Artículo	Descripción
<p>Deep learning for identifying breast cancer malignancy and false recalls: a robustness study on training strategy. Autores: Clancy et al. (2019)</p>	<ul style="list-style-type: none"> <li data-bbox="753 394 1422 485">▪ Base de datos: ImageNet, DDMS, ChestX-Ray8 y Breast Density. <li data-bbox="753 537 1422 856">▪ Herramientas utilizadas: Se exploran distintos métodos para el entrenamiento de una red neuronal convolucional. Se realizan 7 modelos distintos, en los cuales varían las bases de datos, y los parámetros de entrenamiento cambiando también entre dos redes pre-entrenadas. <li data-bbox="753 909 1422 1056">▪ Exactitud alcanzada: AlexNet y ResNet-10. Modelos: A: 75 %, B: 75 %, C: 80 %, D: 66 %, E:67 % , F: 67 %, G: 66 %.

CAPÍTULO 3

METODOLOGÍA

En este capítulo se describen las bases de datos utilizadas para la experimentación, además de entrar a detalle con las herramientas de software utilizadas para armar la arquitectura final de la red. Se presenta una descripción general de los optimizadores que se utilizan dentro de la red, sus características, ventajas y desventajas. Finalmente se detallan las características del hardware que se utilizó en la etapa de entrenamiento de modelos y las pruebas finales.

†

Teniendo en cuenta lo que se describe en los antecedentes de este trabajo, se sabe que cualquier tipo de CAD (Detección ó diagnóstico asistido por computadora) se lleva a cabo con una serie de pasos definidos, los cuales se pueden apreciar en la Figura 3.1⁵.

Se inicia con el pre-procesamiento de la imagen, seguido de la etapa de segmentación que tiene como finalidad extraer las características relevantes, dando paso a la parte de detección de zonas de interés, y poder llegar a la última etapa, que es la clasificación de dicha imagen. La descripción de estos pasos se muestra con más detalle en la Figura 3.2⁶.

⁵Imagen propia.

⁶Imagen propia.



Figura 3.1: Estructura de pasos llevados a cabo por los CAD's tradicionales.

En este trabajo se propone un cambio a esta metodología conocida, dando una primera ramificación antes de la parte de segmentación. El objetivo de esta experimentación es poder estructurar una herramienta de clasificación con redes neuronales que tome la menor cantidad de recursos computacionales y de tiempo, y que a la vez tenga un nivel de exactitud razonable para una herramienta de detección y diagnóstico médico.

La idea detrás de esta motivación, es que los CAD's desperdician un tanto de nivel de procesamiento en la parte de segmentación, ya que, si se quiere como herramienta de detección de anomalías, se puede suponer que van a utilizarse imágenes que no han sido etiquetadas por un experto. Esto quiere decir que hay una probabilidad de que la imagen no contenga ninguna anomalía. En estos casos, la computadora intentará realizar segmentaciones en imágenes que no lo requieren.

Si se ven las bases de datos como ejemplos de la distribución con las que se encuentran las imágenes de mamografías, se puede apreciar que hay una cantidad muy grande de imágenes normales, y una cantidad menor de imágenes con anomalías. Con esto podemos suponer que cuando los CAD's realizan la parte de segmentación están obligando a la computadora a realizar una tarea en la cual es bastante más

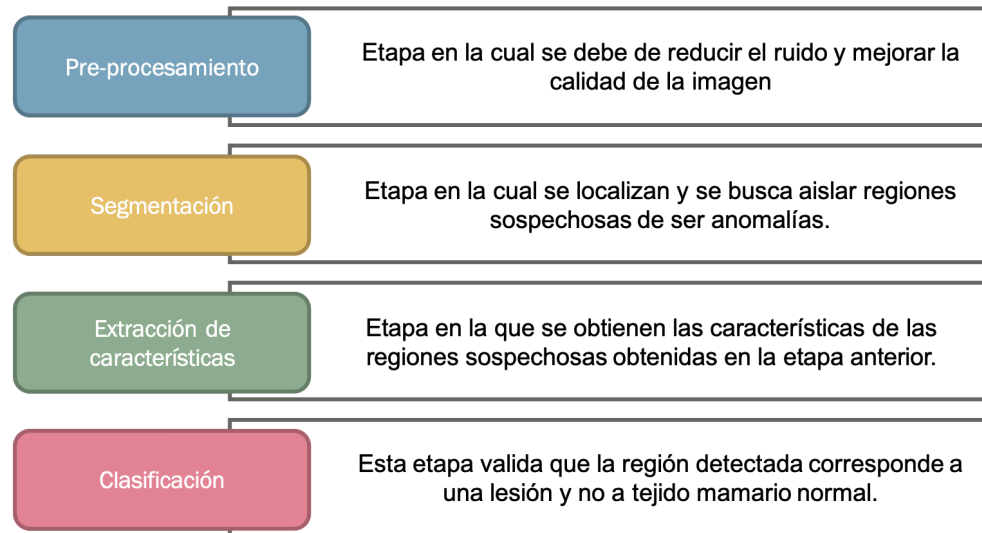


Figura 3.2: Descripción de los pasos utilizados tradicionalmente por las herramientas de diagnóstico y detección.

probable que no encuentren resultados positivos de anomalías. Esto es procesamiento desperdiciado, por lo que en este trabajo se propone posponer este paso, y enfocar primero el esfuerzo computacional en dividir la porción de imágenes que se le da de entrada en dos, creando un lote para imágenes con anomalías, y otro lote con imágenes sin anomalías.

Con esto se pretende dar una herramienta que ayude a priorizar el tiempo de los expertos, ya que, como se menciona en la parte de motivación y antecedentes, en contraste entre la cantidad de información que requiere revisión y la cantidad de expertos capaces de analizarla, se tiene en manos un desbalance abrumador.

3.1 BASES DE DATOS

Antes de iniciar con la explicación de la metodología, en esta sección se explica la distribución de la que vienen las diferentes mamografías con y sin anomalías que conforman la base de datos total de este trabajo. Dos de las bases de datos utilizadas son públicas y reconocidas por diferentes investigaciones de clasificación y

diagnóstico del cáncer de mama. Ambas han estado a disposición de investigadores desde el año 2000, por lo que han pasado por extensas revisiones.

3.1.1 MINI-MIAS

Mammographic Image Analysis Society (MIAS) (ver Suckling (1994)), traducido crudamente como sociedad de análisis de imágenes de mamografías, es una organización de grupos de investigación en el Reino Unido, cuyo trabajo ha generado como resultado una base de datos de mamografías digitales que pueden ser utilizadas como punto de referencia para la investigación del cáncer de mama creada en el año de 1994. Se pueden encontrar más detalles sobre esta base de datos en Michael Heath (2001).

En esta base de datos existen 322 imágenes de mamografías, de las cuales 121 presentan anomalía, mientras que el resto se encuentran sin anomalías. Dentro de este conjunto se encuentra un archivo de texto que especifica también la posición y el tamaño de la anomalía en caso de tenerla, además de especificar la clase de anomalía que presenta. Sus imágenes están en formato .png en canal de blanco y negro con una dimensión de 1024×1024 píxeles.

3.1.2 DDSM

DDSM es un acrónimo de sus siglas en inglés a *Digital Database for Screening Mammography* (ver Michael Heath (2001)). Esta base de datos libre fue realizada gracias a la colaboración y esfuerzo del hospital general de Massachusetts, y la Universidad del Sur de Florida, departamento de ingeniería y ciencias computacionales creada en el año 2000. Esta base de datos contiene alrededor de 2 500 estudios, en el cual se incluyen dos imágenes de cada seno, incluyendo información relevante del paciente (edad en el momento del estudio, densidad de anomalía si es que la tiene,

etc.) sin revelar ninguna otra característica personal que pueda identificarlo.

Para este trabajo de esta base de datos se tomaron solo las imágenes normales, ya que las imágenes que presentaban una anomalía ya estaban segmentadas, por lo cual no se creyó conveniente agregarlas a los conjuntos de entrenamiento y desarrollo. Las imágenes contenidas en esta base de datos se encuentran en formato PNG, en canales RGB.

3.1.3 INCAN

Esta base de datos fue proporcionada en el año 2012 por el Instituto Nacional de Cancerología (INCAN) ⁷ ubicado en la Ciudad de México. Al igual que las imágenes en la base de datos de MINI-MIAS, estas imágenes tienen el tamaño estándar de 1024×1024 , contando con una cantidad de 200 mamografías en total. Todas ellas fueron revisadas por un especialista, por lo cual han sido clasificadas como anómalas. La ventaja de tener esta base de datos dentro de los conjuntos de entrenamiento y prueba es que, al ser más recientes, al momento de modificarlas conservan mejor la información, ya que su formato ya no es en los canales de blanco y negro, si no que están en los de RGB (color) y cuentan con una resolución de calidad mucho más clara.

3.2 METODOLOGÍA PROPUESTA

El cambio que propone este trabajo es una especie de filtro de imágenes antes de que entren a la parte de segmentación. Como se muestra en la Figura 3.3⁸. este filtro se coloca justo antes del paso de segmentación en la sección de división de base de datos. El resto de de los pasos para formar un CAD se seguirá respetando,

⁷Número de registro de base de datos: BI-360-19. Nombre: YERAL.

⁸Imagen propia.

tomando en cuenta los pasos de extracción de características, y clasificación para realizar un clasificador de imágenes que nos divida la base de datos entrante en imágenes con anomalía y sin anomalía.

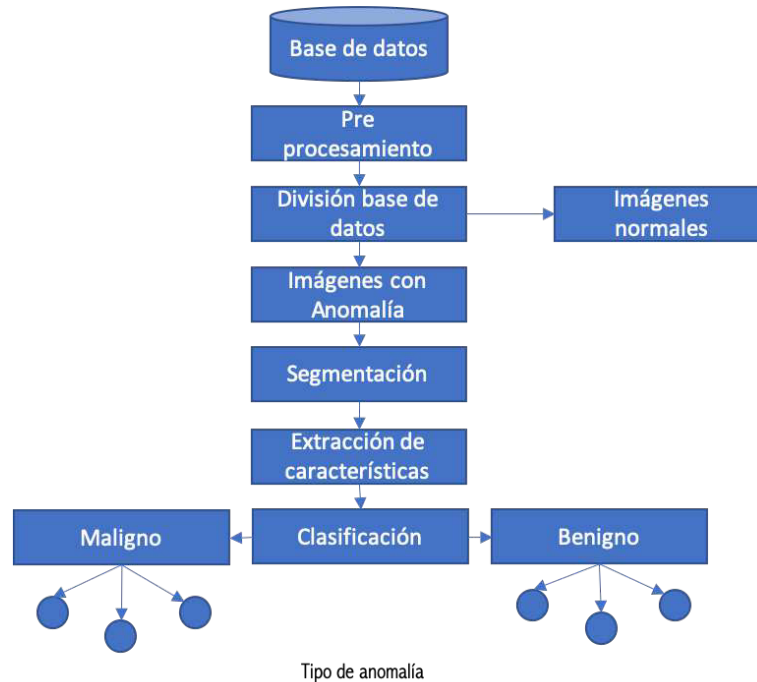


Figura 3.3: Estructura de la metodología propuesta para el desarrollo de CAD's.

3.2.1 PRE-PROCESAMIENTO

El pre-procesamiento que se le realiza a las imágenes antes de que entren al clasificador es necesario para disminuir la capacidad de procesamiento que se le pide a la computadora para tener una clasificación satisfactoria, por lo que el tipo de pre-procesamiento que se le da es para recortar las zonas oscuras, o mejor conocidas como máscaras negras que rodean la imagen de mamografía.

Para desechar las máscaras negras de la imagen se utiliza un código en python, el cual se describe en los apéndices, cuya base es la de recibir los datos de la imagen como si fueran matrices. El programa lee cada columna y se encarga desechar todas aquellas columnas que solo tienen ceros, ya que, como se explicó en la sección de

anteriores, las imágenes utilizadas en blanco y negro van en un gradiente de 0 a 255, siendo 0 el color negro y 255 el valor del blanco.

En el caso de las imágenes de la base de datos del INCAN primero se tuvieron que colocar parches negros en las esquinas superiores, ya que todas aún cuentan con algunos datos del paciente y características de la imagen, como densidad del tejido, escala, etc. Después de este procedimiento, ya se pudo recortar el resto del espacio en negro.

Una vez realizados los recortes, se utilizan algunos métodos de manipulación de imágenes para poder ampliar la base de datos con la que se cuenta. Aquí se hace uso de una librería en Python llamada PIL. Esta librería cuenta con algunas de estas herramientas las cuales pueden ser utilizadas para mejorar la calidad de la imagen, ayudando a resaltar algunas de sus características. Además a probado ser una buena opción para el recorte y redimensión de imágenes, ya que respeta lo mejor posible sus atributos.

Otro de los cambios que se hicieron a las imágenes fue la de normalizar todas a un solo canal de color. Se menciona en la descripción de las bases de datos que las imágenes del conjunto de MINI-MIAS estaban en formato PMG. Al convertirlas al formato PNG, que es como se tienen todas las demás, estas pasaron de ser de canal blanco y negro a tener el canal RGB.

Se decidió modificar las imágenes de las bases de datos antiguas porque se quería utilizar a nuestro favor el valor de calidad que traían consigo las imágenes nuevas del conjunto del INCAN, por lo que se decidió modificarlas en lo más mínimo.

Dentro de la red neuronal existe una función que cambia las imágenes sin modificar su contenido, rotando, trasponiendo, alejando, acercando, etc. Todo esto sigue con la finalidad de aumentar nuestros datos.

La última transformación que se le da a las imágenes resultantes es el del cambio de tamaño. Al tratarse de un clasificador con redes neuronales, es necesario

establecer dimensiones fijas para el tamaño de las imágenes. En este caso se manejan dos tipos de dimensiones, para establecer mediante el análisis de resultados, cual de ellos muestra un resultado más favorable y si se existe un impacto importante cuando estos tamaños cambian.

En todas las distribuciones de imágenes el tamaño original era de 1024×1024 . Como primer paso en la experimentación se intentó entrenar la red con las imágenes enteras de dimensiones originales, pero el procesamiento resultó ser demasiado, incluso para un ordenador con una capacidad superior a la encontrada en computadores normales y añadiéndole el impulso que se obtiene de poder utilizar el GPU. Por este motivo se realizaron varios experimentos para poder llegar al tamaño más pequeño de imagen conservando la mayor cantidad posible de información de ellas.

Se inició con un tamaño mínimo encontrado en un artículo científico Sun et al. (2016) que trabaja también con redes neuronales para la clasificación de imágenes de una base de datos de mamografías digitales. El tamaño fue de 52×52 pero después de realizar el recorte de máscaras negras, las imágenes resultantes quedaban aplastadas. Esto ocurrió como resultado del procedimiento de remover los bordes de máscara negra que se realizó en el paso anterior al re-dimensionamiento. En general, la altura quedaba igual, mientras que de ancho perdía dimensión, por lo que al darle una dimensión cuadrada, daba la impresión de un seno más alargado de lo que debería.

A partir de este primer experimento, se decidió que las dimensiones de las imágenes nuevas deberían ser diferentes, siendo más pequeño el tamaño de anchura a comparación de la altura. Se probó el resultado de los tamaños con diferente optimizador, dejando las capas y neuronas tanto en capa oculta como en la final con las dimensiones del modelo más sencillo que se describe en este trabajo.

Una vez que se estableció que la dimensión de la imagen sería de 80×200 , por los buenos resultados del entrenamiento, como el tiempo final de procesamiento, se eligió un segundo tamaño con el fin de saber si existe un nivel alto o bajo de significancia en cuanto al tamaño, siguiendo con los mismos parámetros para imágenes pero con

dimensiones de 160×400 .

3.2.2 EXTRACCIÓN DE CARACTERÍSTICAS

En la sección de extracción de características existen otras herramientas para localizar los atributos más importantes de una imagen, como lo son las SVM (*Support Vector Machine*), árboles de decisión bayesianos, algoritmos de agrupamiento, etc. En este trabajo, se utilizará como herramienta las redes neuronales convolucionales.

Para la arquitectura de la red se tiene en mente una estructura simple para ahorrar procesamiento a la computadora en la etapa del entrenamiento. Los pasos se pueden observar en la Figura 3.4⁹ y se describen como:

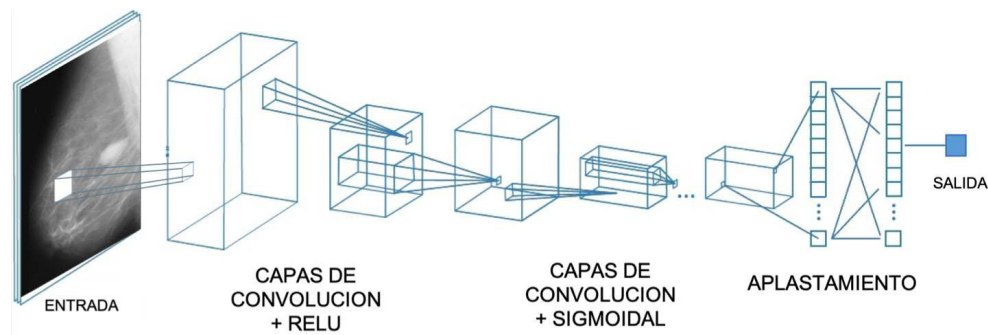


Figura 3.4: Ejemplo de estructura de red neuronal convolucional que se desarrolla en este trabajo.

- Como se explica en la parte de antecedentes, primero se tiene la imagen de entrada, la cual es revisada para sacar sus atributos. El pequeño cuadro que se toma representa el filtro que pasa a la siguiente capa.
- Se observan a continuación las capas de convolución, en las cuales se encuentran las funciones de activación para los filtros que se van a utilizar.

⁹Imagen inspirada en el trabajo en AI (2017)

- Después se tiene la capa de submuestreo, conocida como aplastamiento o *pooling*, en donde se desprecian algunas de las características que presentan menor relevancia en la clasificación final.
- Por último se tiene la capa de clasificación, que dependiendo del atributo que se quiera de salida, ya sea exactitud, pérdida, etc, nos da una predicción del resultado de la imagen de entrada.

En la parte de las capas ocultas se mencionan las funciones de activación. Estas se explicaron en la parte de antecedentes, la razón de tenerlas distribuidas de esa manera es porque, cuando se recibe la imagen, a pesar del primer pre-procesamiento que se le aplica, aún sigue existiendo una gran porción de la imagen que no tiene información relevante para el entrenamiento (la parte oscura de la imagen), por lo que primero se coloca una función ReLu que utiliza una función de 0 a máximo, lo que significa que si el parche de imagen que se le pasa a la neurona contiene al menos un número de escala de color, este parche pasará a la siguiente capa, si no, será descartado.

De esta manera se crea un filtro para que la segunda capa oculta solo reciba parches de imágenes de los cuales puede extraer información. La siguiente capa, a diferencia de la que se acaba de describir, tiene la función sigmoideal. Esta función tiene como salida un gradiente o porcentaje de que tan alto o bajo es el nivel de pixel que contiene su parche. En este caso, pasan este parche de información a la última capa, que es el *maxpooling*, que se encarga de encontrar en que parches existen características más relevantes para traducirlas en la capa final en la mejor predicción que se pueda sacar.

3.2.3 CLASIFICACIÓN

En la parte final de la herramienta desarrollada en este trabajo, tenemos el modelo de la red entrenado con las imágenes que se le dieron en la parte de extracción

de características. A este modelo se le alimentarán otras imágenes, guardadas en un conjunto de imágenes extras, y que de preferencia sean de una distribución distinta a las que se tiene en el conjunto de entrenamiento y desarrollo. Esto quiere decir que ninguna de las imágenes utilizadas en la parte de prueba puede estar en el conjunto de entrenamiento ni desarrollo.

El resultado que dará este modelo de red es una probabilidad de que sea normal o anómala. La librería de KERAS tiene un umbral de decisión en la etapa de entrenamiento y desarrollo de 0.5, lo que significa que si la probabilidad es mayor que el umbral, se toma como un 1, y si es menor se toma como un 0. Se replica este umbral para los resultados del conjunto de prueba. El código para realizar esta parte de clasificación con los modelos obtenidos se tiene en la parte de apéndices de programas en Python. En caso de que la predicción sea normal su valor será de 1, y en caso de que resulte ser anómala, el resultado es un valor cercano al 0.

3.3 ESPECIFICACIONES DE SOFTWARE

Para mejorar la explicación de los diferentes modelos de redes neuronales que se prueban en este documento primero se quiere hacer un breve análisis de las herramientas que se utilizaron para poder llevar a cabo este proyecto. Un ejemplo gráfico de las herramientas más utilizadas se muestran en la Figura 3.5¹⁰.

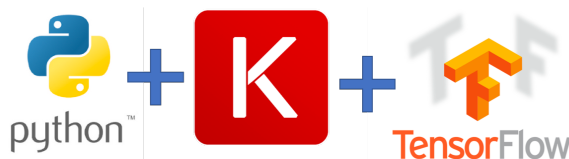


Figura 3.5: Herramientas de software utilizadas para llevar a cabo la red neuronal.

¹⁰Imagen propia formada con logos de las herramientas de software.

3.3.1 PYTHON

En cuanto a la construcción de los modelos, se utilizó el lenguaje de programación interpretado de distribución libre llamado Python. Este lenguaje de programación tiene una base de documentación sólida, además de que su comunidad es bastante grande, lo que ayuda a que sea fácil de entender para otros expertos y programadores.

La razón de elegir Python sobre otros lenguajes de programación que existen se puede encasillar en varios puntos:

- Python es un lenguaje de programación orientado a objetos, que tiene un alto nivel de abstracción a nivel máquina. Esto nos quiere decir que su código se modela alrededor de sus clases, las cuales pueden ser mucho más sencillas de remodelar en caso de que el proyecto que se está realizando cambie a ser más grande de cuando se inició. Más información puede ser encontrado en Foundation (2019).
- Últimamente a tomado un papel importante dentro de los círculos de desarrollo web, robótica, inteligencia artificial, etc. Lo cual ha llevado a este lenguaje a ampliar su entorno, dándole robustez a sus funciones.
- También nos encontramos que la idea detrás de la creación de este lenguaje de programación de código abierto fue la de asemejar lo mayor posible a la sintaxis humana, lo que lo convierte en uno de los lenguajes más sencillos de aprender, al mismo tiempo que asegura que otras personas puedan tener un mayor entendimiento de tu código, lo cual lo hace versátil para compartirlo con la comunidad investigadora.

3.3.2 KERAS

La segunda herramienta que va de la mano con el lenguaje de Python, es la librería de KERAS (2019), la cual se puede describir como una librería de aprendizaje profundo escrita en este mismo lenguaje de programación que facilita la creación y manipulación de modelos de redes neuronales convolucionales y recurrentes.

Existen otras librerías que se desarrollan dentro del esquema de aprendizaje profundo, como por ejemplo Pytorch. La razón para elegir KERAS sobre cualquier otro entorno de librería para redes neuronales recae en el hecho de que KERAS es mucho más concisa en el sentido de que solo se tiene que cambiar los parámetros de sus funciones. Pytorch suele utilizarse cuando la estructura de la red ya es fija, y los únicos parámetros que tienen que cambiarse son detalles pequeños dentro de esta arquitectura.

Para poder crear un modelo con la herramienta de KERAS, es necesario tener las librerías instaladas en la computadora, además de una idea de como se va a estructurar la red neuronal, ya que Keras solamente tiene las funciones puestas en su documentación. La decisión de dónde y cómo acomodar las capas queda enteramente en las manos del programador.

Para llevar a cabo este trabajo se utilizaron las siguientes funciones en la librería de KERAS:

- **ImageDataGenerator:** Se tiene esta función para hacer más grande la base de datos. Su tarea principal es la de modificar las imágenes recibidas de entrada, para que el modelo las vea como imágenes diferentes. Estos cambios realizados son los parámetros que pueden agregarse a esta función, tales como rotar, acercar, alejar, transponer, etc.
- **Sequential:** En KERAS existen dos maneras de crear redes neuronales. Con la función de capa secuencial o la funcional. Ambas sirven para diferentes tipos

de redes. La secuencial, por ejemplo permite crear un modelo de capa por capa que logra cubrir la mayoría de los problemas. La funcional, por otra parte, se utiliza cuando se tienen múltiples salidas, lo que traduce en una mayor complejidad de red resultante. Este último tipo funcional se adapta mejor a redes residuales o las redes siameses.

- **Conv2D:** Convolución en 2D es la capa de KERAS que se especializa en la recepción de imágenes. Cuando se utiliza como primera capa para recibir la información es necesario llenar algunos de los espacios adicionales con los que cuenta, como el *input shape*, que es el encargado de comunicarle a la red con que tamaño de imagen se va a estar trabajando.
- **MaxPooling2D:** Como se menciona en la parte de antecedentes de este trabajo, la capa de *Max pooling* es utilizada para tomar los valores más prominentes de los parches de imagen que recibe como entrada, descartando los valores con menos impacto. Al igual que la función de Conv2D, en este caso se tiene el 2D porque se está trabajando con imágenes. El 1D está designado para datos de texto, mientras que el 3D es para videos.
- **Flatten:** Esta capa no cuenta con parámetros que puedan cambiarse. Se manda a llamar cuando se requiere aplastar la información de la capa anterior. Como se explica en el apartado de antecedentes, este aplastamiento no es más que re-ordenar los datos del mapa de activación que se tiene de salida de la capa anterior. Esta función no afecta el tamaño del parche de información recibido.
- **Dense:** Esta función es una capa densa para conectar las capas finales. Esta capa contiene la matriz de valores en los cuales se guardan los parámetros entrenables que se van actualizando durante el proceso de propagación hacia atrás, la cual es una función necesaria en una red neuronal para reducir el parámetro de error en el modelo.

- **Compile:** En esta función es donde se cambian los parámetros de salida del modelo. Aquí es donde se debe de especificar cual es el valor que se quiere recibir. En este caso, se quiere medir la exactitud del modelo, por lo que en el parámetro de métricas, se le da ese valor.

Otro parámetro importante es el del optimizador. Este es el encargado de optimizar los valores de los pesos para disminuir el valor del error final, y de esta manera aumentar la exactitud del modelo.

KERAS maneja siete optimizadores diferentes, los cuales cuentan con características diferentes. A continuación se da una breve introducción a cada uno de ellos.

ADAM: El algoritmo de optimización Adam es una extensión del descenso gradiente estocástico, el cual ha ampliado su área de uso en aplicaciones de aprendizaje de máquina y visión computacional. Su nombre es un derivado de estimación adaptativa por momento (*adaptive moment estimation*).

Este optimizador es utilizado en lugar de los procesos clásicos de descenso gradiente estocásticos para actualizar los pesos del modelo de red para entrenar los datos. En los procesos clásicos, el ritmo de aprendizaje se mantiene fijo durante todas las actualizaciones de pesos. Adam calcula un promedio exponencial en movimiento del gradiente y el gradiente al cuadrado, teniendo dos parámetros para controlar el ritmo de caída de estos promedios en movimiento. Se puede encontrar información más detallada en el artículo por Kingma and Ba (2014).

Los parámetros que tiene por defecto en KERAS son: $learning\ rate = 0,001$, $beta_1 = 0,9$, $beta_2 = 0,999$, $epsilon = 1e - 08$, $decay = 0,0$.

ADAMAX: Este optimizador proviene de la distribución de algoritmo similar a Adam, tomando inspiración es las fórmulas utilizadas por este, pero dándole prioridad a la maximización de la regla de actualización. Esto hace que la función final no

sea tan susceptible a la varianza de los datos.

Al igual que el algoritmo de Adam, Adamax cuenta con dos valores para el ritmo de aprendizaje. Ver Baidu (2018) para profundizar en las características de este optimizador.

NADAM: En el caso de Nadam, se describe por sus siglas en inglés, *Nesterov-accelerated Adaptive Moment Estimation*, en el cual se combinan características de los algoritmos Adam, Rmsprop y SGD+Nesterov. Al utilizar el momento para avanzar, le añade a las funciones de dichos algoritmos un paso que se da en la dirección del vector de momento anterior, y otro paso hacia la dirección del gradiente actual.

Esto permite corregir el valor de la varianza, que en los algoritmos de los que toma inspiración, tienen problemas para tener un buen estimado en la exactitud del entrenamiento. El artículo por Timothy (2016) habla en más detalle sobre las características de este optimizador.

SGD: El nombre de este optimizador viene de las siglas en inglés *Stochastic Gradient Descent* (Descenso de gradiente estocástico). Este parámetro está atado de manera cercana a las características que se le da en un inicio. Si el ritmo de aprendizaje se le da un número alto, llegará más rápido a una respuesta satisfactoria, pero es mucho más probable que se pase del objetivo, y una vez que pasa esa marca, no es posible que tenga un resultado satisfactorio. En cambio, si se tiene un ritmo de aprendizaje pequeño, aunque con un nivel de convergencia más lento, tiene mayor seguridad de llegar al valor deseado.

Por esto es que escoger un nivel de ritmo de aprendizaje es un poco complicado para este optimizador. Asimismo, este ritmo elegido al inicio de los parámetros se aplica en todas las transformaciones del modelo, lo que significa que si los datos de entrada tienen frecuencias distintas, este optimizador no desempeñará satisfactoriamente. Para profundizar más en el tema, se puede consultar el artículo por Chengwei (2018).

ADAGRAD: Este es un optimizador que intenta mejorar el problema de SGD, eliminando la necesidad de cambiar el parámetro del ritmo de aprendizaje de manera manual. Las implementaciones más recientes toman el valor de 0.01. Este optimizador se desarrolla a su mejor potencial cuando recibe datos escasos de entrada.

Su mayor desventaja, sin embargo, es que tiende a acumular gradientes cuadrados en su denominador. Esto causa que el ritmo de aprendizaje se haga muy pequeño, y eventualmente se vuelva minúsculo. Llegando a este punto el algoritmo no puede adquirir conocimientos adicionales. Más características se describen en el artículo por Roan and Gylberth (2018).

ADADELTA: En el caso de Adadelata, se puede tomar como una extensión del algoritmo de Adagrad, que intenta implementar una metodología para reducir la manera agresiva en la que va a disminuyendo el ritmo de aprendizaje. Donde Adagrad guarda todos los gradientes cuadrados que dan de salida en cada iteración de entrenamiento, Adadelata restringe la cantidad de datos que pueden guardarse en este denominador, poniendo un tamaño fijo en este valor.

Con Adadelata tampoco se tiene que establecer un ritmo de aprendizaje por defecto, ya que este es eliminado de la regla de actualización dentro del algoritmo. Se puede consultar el artículo por Zeiler (2012) para profundizar en las características de este optimizador.

RMSPROP: Al igual que el algoritmo de optimización de Adadelata, RMSprop se ha desarrollado de manera independiente alrededor del problema de Adagrad que disminuye el ritmo de aprendizaje, y eventualmente lo estanca. A diferencia Adadelata, este optimizador sí utiliza un valor de inicio para su ritmo de aprendizaje. Existen varias investigaciones que intentan estimar cual es el valor óptimo para empezar este valor, aunque siempre se considera como buen inicio un valor bajo, como por ejemplo 0.0001. El artículo por Vitaly (2018) describe a mayor detalle las características de este optimizador.

3.3.3 TENSORFLOW

En este punto cabe mencionar la tercer herramienta que es una pieza clave para la mejora de la velocidad de procesamiento y rendimiento de la máquina en donde se desarrolla la red. TensorFlow (2019), se describe como una fuente de software libre para la computación de gráficos numéricos usando flujos paralelos.

Se menciona que TensorFlow es la clave en la velocidad de procesamiento, porque las máquinas que se manejan hoy en día tienen una nueva y mejor capacidad tecnología de procesamiento, y esta herramienta nos da la oportunidad de explotar esa ventaja en otro nivel.

TensorFlow permite el acceso a las tarjetas GPU que existen dentro del ordenador. Usualmente este tipo de tarjetas se utiliza solamente para mejorar los gráficos del monitor y optimizar el funcionamiento de la PC, pero al tener acceso a ella cuando se esta trabajando redes neuronales, el modelo puede correr con el triple de velocidad que lo haría en un CPU normal. Utilizando una técnica conocida como programación en paralelo.

Una de las funciones en las que la programación en paralelo muestra uno de sus mejores rendimientos de procesamiento computacional es en operaciones matemáticas con matrices que se tienen que repetir muchas veces. Por esto es de gran utilidad al momento de utilizarlas para entrenar redes neuronales, que no son más que muchas operaciones con matrices.

No todas las computadoras tienen acceso a su tarjeta de gráficos. Dicha tarjeta tiene que ser compatible con la distribución de CUDA y CUDNN, que son otras dos herramientas que se utilizan para poder enlazar el GPU al TensorFlow. Tanto CUDA como CUDNN son distribuciones que están tratando de impulsar el aprendizaje profundo y de máquina entre la comunidad de programadores.

3.3.4 ESPECIFICACIONES DE HARDWARE

Para realizar los experimentos descritos en este trabajo, se utilizó una estación de trabajo HP Z420 con las siguientes características:

- NVIDIA Tesla K20c 1st *Compute Processor*
- NVIDIA Quadro K600 1GB 1st GFX Spl
- 500GB 7200 RPM SATA 1st HDD
- 16X SuperMulti DVDRW SATA 1st ODD

CAPÍTULO 4

DISEÑO DE EXPERIMENTOS

En este capítulo se describen los experimentos realizados para llegar a la estructura de red neuronal con mejor desempeño, especificando sus parámetros fijos tanto como los parámetros variables. Se da una explicación de las métricas utilizadas para la medición de valores como la pérdida y la exactitud dentro del entrenamiento. Por último, se presenta una explicación breve de los resultados obtenidos que serán analizados en capítulos subsecuentes.

†

La estructura de la red neuronal creada en este trabajo se hizo lo más simple posible con la finalidad de hacer un procesamiento sencillo, y a su vez hacer posible un diseño de experimentos en el cual se variaron algunos de los parámetros de la arquitectura para poder decidir al final, con un análisis de resultados, cuáles de estos modelos son los adecuados para realizar réplicas en el futuro, y seguir experimentando con sus parámetros hasta encontrar el que demuestre tener el mejor resultado.

4.1 PARÁMETROS EXPERIMENTALES

Para la variedad de arquitecturas en los modelos, se utilizaron diferentes características en cada uno de ellos, las cuales se explican en la Tabla 4.1 y se describen con más detalle a continuación.

Tamaño de la Imagen	Optimizador	Neuronas segunda capa convolucional	Neuronas capa de aplastamiento
80×200	Adam	32	128
160×400	Adamax	64	240
	Nadam	128	380
	RMSProp		
	SGD		
	Adadelata		
	Adagrad		

Tabla 4.1: Distribución de características de cada modelo.

TAMAÑO DE LA IMAGEN: Como se mencionó en la parte de metodología, los dos tamaños de imágenes que se utilizan en este trabajo se establecieron gracias a varios experimentos a escala en los que se determinó que se podrían reducir las imágenes, sin sacrificar la información que esta pueda proporcionar. Al finalizar estos experimentos se observó que el tamaño de 80×200 era la escala mínima a la que se puede llegar sin que esto afectara significativamente el resultado del procesamiento. Por procesamiento se quiere ejemplificar el cambio por el cual tuvieron que pasar las imágenes para crear una base de datos suficientemente grande para entrenar a una red neuronal.

Como se explica en el apartado de metodología, para poder tener una fuente razonable de información, se recurrió a la modificación de las imágenes para hacer

una base de datos más robusta. Estas modificaciones se realizaron en ambos tamaños.

OPTIMIZADOR: Como se menciona en la parte de antecedentes, el optimizador es el encargado de actualizar los pesos de la red, basándose en los resultados de la función de pérdida. KERAS maneja varios optimizadores que son variantes del método de descenso gradiente estocástico.

NEURONAS CAPA CONVOLUCIONAL Y DE APLASTAMIENTO: En los casos de la capa convolucional y de aplastamiento, se seleccionaron tres números arbitrarios, para experimentar con qué cantidad de neuronas en estas capas el modelo se desempeña de mejor manera.

4.2 PARÁMETROS FIJOS

En cada caso, las características que permanecen fijas son la de cantidad de capas convolucionales, ya que en todo el experimento se mantienen en dos. El segundo parámetro fijo es el de distribución de los conjuntos de datos. En este trabajo se utilizan tres conjuntos, los cuales son el de entrenamiento, desarrollo y prueba. En este primer lote de experimentación se utilizó la distribución de 70 % – 25 % – 5 % del total de las imágenes disponibles, para cada conjunto respectivamente.

Esto se puede ilustrar en la Figura 4.1¹¹ en donde se puede apreciar la división que se menciona. De la totalidad de imágenes normales, se realiza una división del 70 %, 25 % y 5 % de ellas para ponerlas en la carpeta de normales del conjunto de entrenamiento, desarrollo y prueba. Lo mismo se realiza con la totalidad de imágenes con anomalías, solo que estas van en la carpeta de anómalas de cada uno de sus respectivos conjuntos.

Esta nueva metodología de tres conjuntos se presentó en los trabajos del libro

¹¹Imagen propia.

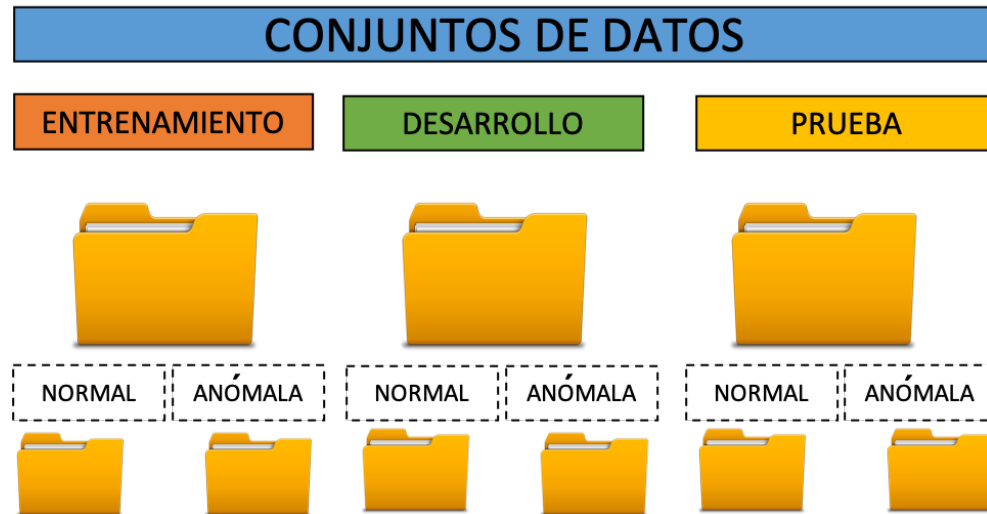


Figura 4.1: Distribución de los conjuntos de imágenes para el entrenamiento, desarrollo y prueba de la red.

de Andrew Ng para su curso titulado *Machine Learning Yearning*, Ng (2018) en donde se utilizan los dos conjuntos tradicionales de entrenamiento y desarrollo, más una pequeña porción de datos fuera de estos conjuntos para poder armar el conjunto de prueba. Este conjunto es utilizado en este trabajo exclusivamente como prueba de fallos, y se utilizó en las primeras fases del armado de la estructura de red para encontrar el tamaño mínimo de dimensión de imagen que se podía utilizar.

De este conjunto podemos extraer lo que se le conoce en los trabajos de Andrew Ng como el *"eye ball set"* traducido crudamente a conjunto de ojo, que se explica como un conjunto reducido de las imágenes contenidas dentro del conjunto de prueba en el cual el programador puede comparar contra los resultados reales las predicciones que realiza el modelo, y de esta manera decide cuáles son las características que la red encuentra más difícil de clasificar. Esto nos sirve para tener una manera más sencilla de encontrar el porqué de los fallos o dificultades que pueda encontrarse la red neuronal en el momento de procesar los datos de los conjuntos de entrenamiento y desarrollo.

Este tipo de conjunto extra se ejemplifica en el conjunto utilizado en el cla-

sificador que se desarrolla en el libro de *Machine Learning Yearning*, el cual es un clasificador de imágenes de perros y gatos. En este caso es entendible que se utilice este tipo de conjunto, pero en el caso de este trabajo, este conjunto solo puede ser revisado y analizado por un experto, ya que el área de especialidad no es dominada por los programadores de la herramienta.

Para poder comparar los resultados de los modelos obtenidos en este trabajo con otras herramientas utilizadas en la literatura, se tuvo que hacer el análisis de experimentos en resultado de exactitud del conjunto de desarrollo, ya que la mayoría de los artículos publicados solo cuentan con los dos conjuntos tradicionales.

El tercer elemento que se mantiene fijo en todos los modelos realizados es la cantidad de réplicas o mejor conocidas como *epochs* y las iteraciones cortas. En el primer caso, en todos los modelos, para el conjunto de entrenamiento se le dan 10 *epochs*, con 1000 iteraciones dentro de cada una de ellas. Para validar cada *epoch*, se tienen 2000 iteraciones cortas que se le dan al conjunto de desarrollo. Estas cantidades fueron elegidas después de un análisis en las iteraciones de cada modelo.

En las Figuras 4.2 y 4.3¹² se puede apreciar el avance en exactitud que van teniendo los modelos conforme avanza el número de *epochs*. En ambos casos la exactitud máxima mejora a cada iteración, pero en el caso de la gráfica de desarrollo, el cuartil superior de la décima iteración disminuye a comparación del noveno, por lo cual se puede concluir que 10 réplicas son suficientes para tener un resultado favorable.

¹²Ambas imágenes son propias.

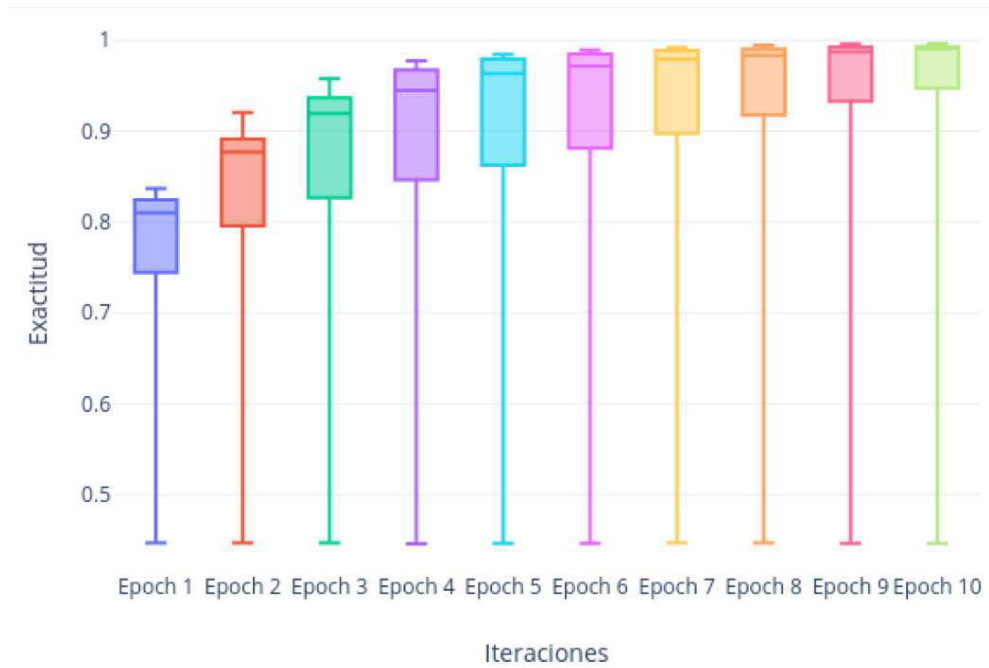


Figura 4.2: Exactitud en las diferentes etapas de réplicas del conjunto de entrenamiento.

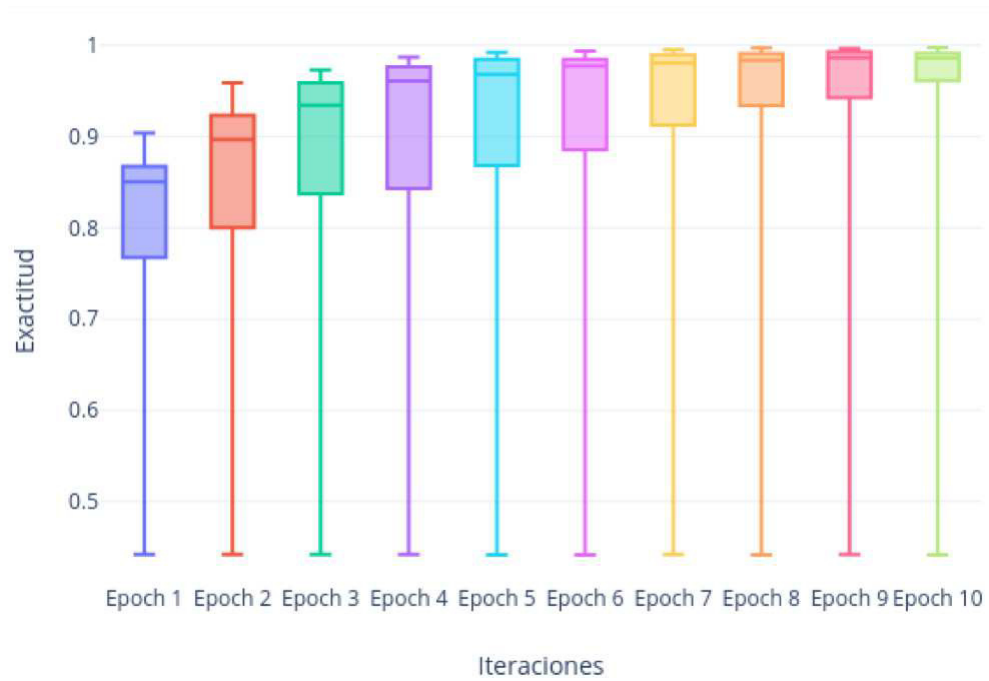


Figura 4.3: Exactitud en las diferentes etapas de réplicas del conjunto de desarrollo

4.3 MÉTRICAS DE SALIDA

Se menciona en los antecedentes que las métricas utilizadas para la salida de una red neuronal son la función de pérdida y la exactitud. En el caso de la función de pérdida, se tiene que junto con el optimizador elegido, son las que ayudan a mejorar la predicción final en el entrenamiento. La exactitud es el porcentaje de predicciones correctas que se tienen en el conjunto de desarrollo.

En el caso de la función de pérdida, se considera la crosentropía binaria, dado que se trata de una clasificación entre dos clases. La función de crosentropía binaria, o como se le conoce dentro de KERAS, *binnary crossentropy*, es utilizada en casos en donde se tiene que clasificar entre dos salidas 0 o 1, que en este caso representan como anómala ó normal, respectivamente.

La fórmula que utiliza KERAS para calcular el valor de pérdida se describe por Godoy (2018) como:

$$Hp(q) = -\frac{1}{N} \sum_{i=1} y_i \cdot \log(p(y_i)) + (1 + y_i) \cdot \log(1 - p(y_i)) \quad (4.1)$$

Donde:

y_i = Es la etiqueta o la clasificación correcta.

$p(y_i)$ = Predicción de probabilidad de que la entrada sea 0 o 1.

N = Número total de entradas.

La métrica de exactitud de KERAS tiene un procedimiento específico para definir una imagen como anómala o normal. El modelo de red nos da como resultado la probabilidad de que la imagen de entrada esté cerca del valor normal, que es 1, o del valor de anomalía, que es 0.

Lo que hace la fórmula de exactitud es redondear esta predicción mencionada;

si es mayor que .5 se considera como 1, y si es menor, se considera como 0. Después del redondeo, se compara con el valor real de la imagen. El valor de 0.5 se le llama umbral, y se utiliza este mismo valor para comprobar las predicciones dentro del conjunto de prueba. Esta comparación es posible por la manera en que están separadas las imágenes en los conjuntos de entrenamiento y prueba, gracias a la función de *binary crossentropy*, que indica que solo hay dos opciones en las que pueden ser clasificados los datos de entrada (Godoy (2018)).

$$K.mean(K.equal(y_{true}, K.round(y_{pred}))) \quad (4.2)$$

El resultado de la ecuación (4.2), se puede identificar de manera más gráfica con la tabla de confusión que se muestra en la Figura 4.4 en donde se enumeran los verdaderos positivos y verdaderos negativos (color azul) que es cuando la predicción y la etiqueta de la imagen coinciden en el resultado. En los cuadros rojos se tienen los falsos positivos y falsos negativos, que es cuando la predicción y la etiqueta de la imagen son diferentes.

$y_{true} = 0$ $y_{pred} = 0$ True Negative	$y_{true} = 0$ $y_{pred} = 1$ False Positive
$y_{true} = 1$ $y_{pred} = 0$ False Negative	$y_{true} = 1$ $y_{pred} = 1$ True Positive

Figura 4.4: Clasificación de las imágenes para el entrenamiento de la red con los valores de la ecuación (4.2)

Esta tabla de confusión se utilizará más adelante en el análisis de resultados, para obtener los resultados de sensibilidad, especificidad, exactitud, precisión, y la

puntuación F1. Estos son los datos reconocidos para calificar los resultados de cualquier herramienta que se basa en el aprendizaje de máquina.

La sensibilidad es la métrica que calificará cuántas imágenes con anomalía de verdad fueron clasificadas en el conjunto de prueba. La métrica de especificidad indica la tasa de verdaderos negativos, es decir, los negativos que fueron correctamente clasificados. La exactitud es la más conocida, ya que en esta se basan los conjuntos de entrenamiento y desarrollo, y es qué tan cerca está el valor real del valor que se predijo. Después se tiene la precisión, la cual se encarga de medir qué tan dispersas se encuentran las predicciones. Entre menor dispersión, mejor la precisión.

Por último se tiene el valor F1, que utiliza los valores de salida de todas las métricas anteriores, para dar una calificación balanceada entre los verdaderos positivos y negativos, y los falsos positivos y negativos.

CAPÍTULO 5

ANÁLISIS DE RESULTADOS

En este capítulo se hace un análisis de los resultados obtenidos al finalizar el entrenamiento de los diferentes modelos creados en la etapa de diseño de experimentos. Aquí se muestran los diferentes procesos que se llevaron a cabo para eliminar algunos de los modelos, para poder continuar con la experimentación final solamente con los que demostraron tener mejores resultados.

Esta experimentación final se lleva a cabo con el conjunto final de prueba, cuyas características se describieron en la sección de metodología, y el cual sirve como guía para saber qué tanta exactitud tiene el modelo en cuestión. Por último se tienen las tablas de resultados en dichos modelos seleccionados.

†

Para poder realizar el análisis de resultados, se tomaron las puntuaciones de exactitud del conjunto de desarrollo de cada combinación de modelo realizada. En total con todos los modelos se tienen 126 resultados. Para recortar significativamente la cantidad de modelos, se realiza primero un análisis estadístico en RStudio para saber cuales son las características más influyentes en la estructura del modelo, y de esta manera empezar a descartar los modelos cuyos resultados no favorezcan la clasificación final.

Para realizar este experimento se toma como estrategia el diseño factorial. Este tipo de diseño puede tener distintos niveles y factores, sin embargo, a partir de una cantidad mayor de tres factores, el resultado del experimento se vuelve más complicado de controlar e interpretar. Esto se dificulta incluso más si existe un desbalance en la cantidad de niveles en cada factor.

A partir de los resultados finales de exactitud de los modelos, antes de iniciar con los pasos para llevar a cabo la experimentación del análisis de resultados se decide despreciar los valores de todos los modelos que tengan el optimizador de Adagrad ya que ninguno demostró una clasificación con probabilidad mayor al 50 % de exactitud en las etapas de entrenamiento y desarrollo, lo que nos dice que sus resultados para la parte de prueba van a ser totalmente aleatorios.

Como se puede apreciar en las Tablas 5.1 y 5 los factores y niveles se pueden describir de la siguiente manera:

- **Factor 1:** Tamaño de la imagen.
 - **Nivel 1:** 80×200
 - **Nivel 2:** 160×400
- **Factor 2:** Neuronas capa convolucional.
 - **Nivel 1:** 32
 - **Nivel 2:** 124
 - **Nivel 3:** 240
- **Factor 3:** Neuronas capa de aplastamiento.
 - **Nivel 1:** 128
 - **Nivel 2:** 240
 - **Nivel 3:** 380
- **Factor 4:** Optimizador.

Tabla 5.1: Descripción del nombre numérico de los modelos.

Tamaño de imagen	Capa convolucional	Capa de aplastamiento	Optimizador
1 (80 × 200)	1 (32)	1 (128)	1 (Adam)
2 (160 × 400)	2 (64)	2 (240)	2 (Adamax)
	3 (128)	3 (380)	3 (Nadam)
			4 (RMSProp)
			5 (SGD)
			6 (Adadelata)

- **Nivel 1:** Adam
- **Nivel 2:** Adamax
- **Nivel 3:** Nadam
- **Nivel 4:** RMSProp
- **Nivel 5:** SGD
- **Nivel 6:** Adadelata

Para poder hacer referencia a los modelos, se les designó un nombre numérico, el cual se puede describir de la manera en que se muestra en la Tabla 5.1. Por ejemplo, si un modelo se llama **2331** significa que sus características pertenecen a imágenes del tamaño 160×400 (**2**), con 240 neuronas de la segunda capa convolucional (**3**), 380 de capa de aplastamiento (**3**) y que utiliza el optimizador Adam (**1**). Este identificador es con la finalidad de no escribir las características de cada modelo en todas las tablas, y que sean más sencillos de identificar en los resultados.

En la Tabla 5 podemos apreciar los porcentajes de exactitud obtenidos en el conjunto de validación en cada uno de los modelos, y a partir de estos resultados es donde se eligen los modelos que pasan por la última parte de prueba.

Tabla 5.2: Porcentaje de exactitud de modelos en el conjunto de validación.

Modelo	Exactitud	Modelo	Exactitud	Modelo	Exactitud	Modelo	Exactitud
1 1 1 1	99.15 %	1 1 1 4	98.64 %	2 1 1 1	99.69 %	2 1 1 4	51.68 %
1 2 1 1	99.32 %	1 2 1 4	96.90 %	2 2 1 1	98.95 %	2 2 1 4	98.00 %
1 3 1 1	98.63 %	1 3 1 4	96.98 %	2 3 1 1	99.37 %	2 3 1 4	51.67 %
1 1 2 1	98.85 %	1 1 2 4	96.48 %	2 1 2 1	99.21 %	2 1 2 4	99.00 %
1 2 2 1	99.11 %	1 2 2 4	99.37 %	2 2 2 1	100.00 %	2 2 2 4	97.04 %
1 3 2 1	99.02 %	1 3 2 4	44.22 %	2 3 2 1	51.66 %	2 3 2 4	48.35 %
1 1 3 1	98.68 %	1 1 3 4	99.28 %	2 1 3 1	99.79 %	2 1 3 4	99.37 %
1 2 3 1	99.41 %	1 2 3 4	55.81 %	2 2 3 1	51.68 %	2 2 3 4	51.65 %
1 3 3 1	99.28 %	1 3 3 4	97.65 %	2 3 3 1	51.68 %	2 3 3 4	48.34 %
1 1 1 2	97.99 %	1 1 1 5	96.08 %	2 1 1 2	99.37 %	2 1 1 5	97.63 %
1 2 1 2	55.80 %	1 2 1 5	97.61 %	2 2 1 2	99.10 %	2 2 1 5	97.31 %
1 3 1 2	98.56 %	1 3 1 5	97.66 %	2 3 1 2	92.78 %	2 3 1 5	96.52 %
1 1 2 2	99.40 %	1 1 2 5	89.45 %	2 1 2 2	51.66 %	2 1 2 5	96.27 %
1 2 2 2	99.15 %	1 2 2 5	97.11 %	2 2 2 2	99.58 %	2 2 2 5	94.99 %
1 3 2 2	55.81 %	1 3 2 5	96.28 %	2 3 2 2	48.35 %	2 3 2 5	97.05 %
1 1 3 2	98.90 %	1 1 3 5	96.85 %	2 1 3 2	48.32 %	2 1 3 5	97.83 %
1 2 3 2	99.57 %	1 2 3 5	96.59 %	2 2 3 2	51.66 %	2 2 3 5	95.41 %
1 3 3 2	55.83 %	1 3 3 5	97.57 %	2 3 3 2	48.35 %	2 3 3 5	97.04 %
1 1 1 3	98.98 %	1 1 1 6	99.45 %	2 1 1 3	48.34 %	2 1 1 6	51.65 %
1 2 1 3	98.60 %	1 2 1 6	99.36 %	2 2 1 3	51.64 %	2 2 1 6	99.42 %
1 3 1 3	98.77 %	1 3 1 6	99.11 %	2 3 1 3	51.63 %	2 3 1 6	99.58 %
1 1 2 3	98.85 %	1 1 2 6	98.81 %	2 1 2 3	51.68 %	2 1 2 6	99.32 %
1 2 2 3	99.71 %	1 2 2 6	99.57 %	2 2 2 3	48.34 %	2 2 2 6	99.63 %
1 3 2 3	98.85 %	1 3 2 6	99.11 %	2 3 2 3	48.32 %	2 3 2 6	99.58 %
1 1 3 3	99.10 %	1 1 3 6	99.75 %	2 1 3 3	51.67 %	2 1 3 6	51.64 %

Continúa en la próxima página

Tabla 5.2 – *Continuación*

Modelo	Exactitud	Modelo	Exactitud	Modelo	Exactitud	Modelo	Exactitud
1 2 3 3	98.50 %	1 2 3 6	99.40 %	2 2 3 3	51.66 %	2 2 3 6	99.84 %
1 3 3 3	44.18 %	1 3 3 6	99.44 %	2 3 3 3	51.67 %	2 3 3 6	99.15 %

Para poder llevar a cabo la experimentación con el último conjunto de prueba que se guardó separado de los conjuntos de entrenamiento y desarrollo, se realiza una gráfica en el lenguaje de programación R para identificar los modelos con buen desempeño y los de bajo desempeño. Con estas gráficas lo que se quiere explorar es el promedio de exactitud de los modelos tomando en cuenta las características de su arquitectura. Primero, se realizan gráficas de los efectos univariantes, mostrados en la Figura 5.1.

La media divide visiblemente los optimizadores y el tamaño de las imágenes, mientras que la cantidad de neuronas en capa oculta y final parecen estar un poco más compactas entre ellas. Para poder saber cuales de todos elementos son los que tienen mayor relevancia, se realiza un ANOVA a los datos. Análisis de varianza (ANOVA) es una herramienta para cuantificar sí o no una variable o factor tiene un efecto estadísticamente significativo en una variable de interés.

Hay tres tipos de ANOVA, cuyo uso depende de si se espera contar con interacciones entre las variables de entrada y si los datos con los que se cuenta son balanceados. En nuestro caso se espera tener una relación fuerte entre el optimizador y el tamaño de la imagen, ya que en ambos casos, la Figura 5.1 muestra un margen grande entre los optimizadores con mejores resultados, al igual que en el tamaño que presentó mejor rendimiento.

La hipótesis nula en el ANOVA es que no haya diferencias en las medias de los

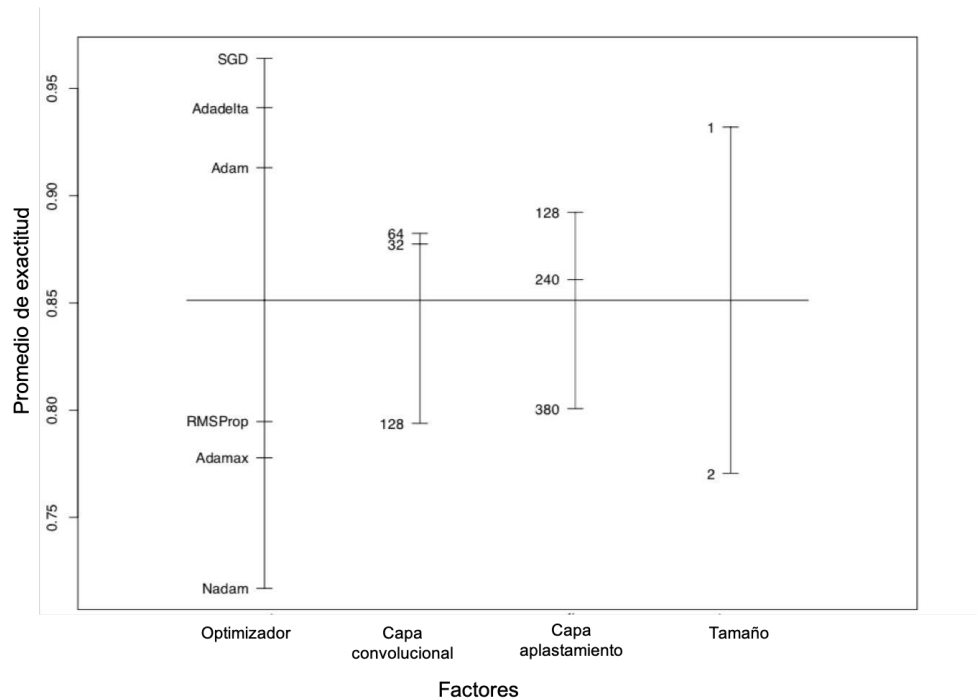


Figura 5.1: Efectos univariantes de los factores utilizados en la arquitectura de los modelos.

factores; si el valor P de un factor es menor a la significancia establecida entonces se rechaza la hipótesis nula, concluyendo que hay diferencia en las medias y que sí tiene un efecto estadísticamente significativo. En este trabajo se utilizó un valor de $P = 0,01$, de acuerdo a lo recomendado por RStudio (2017)

Esta herramienta de diseño de experimentos se eligió con el objetivo de saber cuales son los factores del experimento que influyen en el desempeño de los modelos y cuales de ellos pueden considerarse como parámetros fijos, a lo largo de experimentaciones futuras.

La Tabla 5.3 se obtuvo mediante unas líneas de código en Python descrito en el Apéndice A, utilizando herramientas de la librería de *statsmodels*, *numpy* y *pandas*; *Ocultas* representa la capa oculta, y *Final* el de la capa de aplastamiento.

Se confirma en los resultados que el tamaño y el optimizador son los dos elementos más influyentes en el resultado final de exactitud. El siguiente paso es buscar

Tabla 5.3: Resultados del impacto que tienen los factores individuales en los modelos.

Nombre de parámetro	Suma cuadrada	Diferencia	Valor F	Valor P ($>F$)	Conclusión
Optimizador	0.921	5.0	6.083	< 0.001	Es significativo
Oculto	0.153	1.0	5.054	0.026	No es significativo
Final	0.153	1.0	5.068	0.026	No es significativo
Tamaño	0.703	1.0	23.200	< 0.001	Es significativo
Oculto + Optimizador	0.321	5.0	2.583	0.032	No es significativo
Tamaño + Optimizador	0.450	5.0	3.624	< 0.001	Es significativo
Oculto + Tamaño	0.002	1.0	0.115	0.735	No es significativo
Optimizador + Final	0.120	5.0	0.970	0.441	No es significativo
Oculto + Final	0.058	1.0	2.343	0.129	No es significativo
Tamaño + Final	0.031	1.0	1.276	0.261	No es significativo
Residual	3.000	99.0	NaN	NaN	No es significativo

las interacciones que tienen todos estos elementos entre sí.

Se utiliza la palabra *oculto* para representar las capas convolucionales, y la palabra *Final* para representar la capa de aplastamiento. La única interacción que muestra ser significativa es la del tamaño y el optimizador. Con la herramienta de R, se realiza ahora una gráfica para analizar los datos del residual de todos estos

datos. Se pueden ver los códigos utilizados en el Apéndice B.

Como se muestra en la gráfica en la parte inferior derecha, el optimizador SGD es el que parece tener más robustez en cuanto a su comportamiento al cambiar el tamaño de la imagen. Adadelta y Adam, a pesar de estar un poco más dispersos que el SGD, son los que tienen dos de los modelos con exactitud más alta. Estos resultados pueden verse en la Figura 5.2.

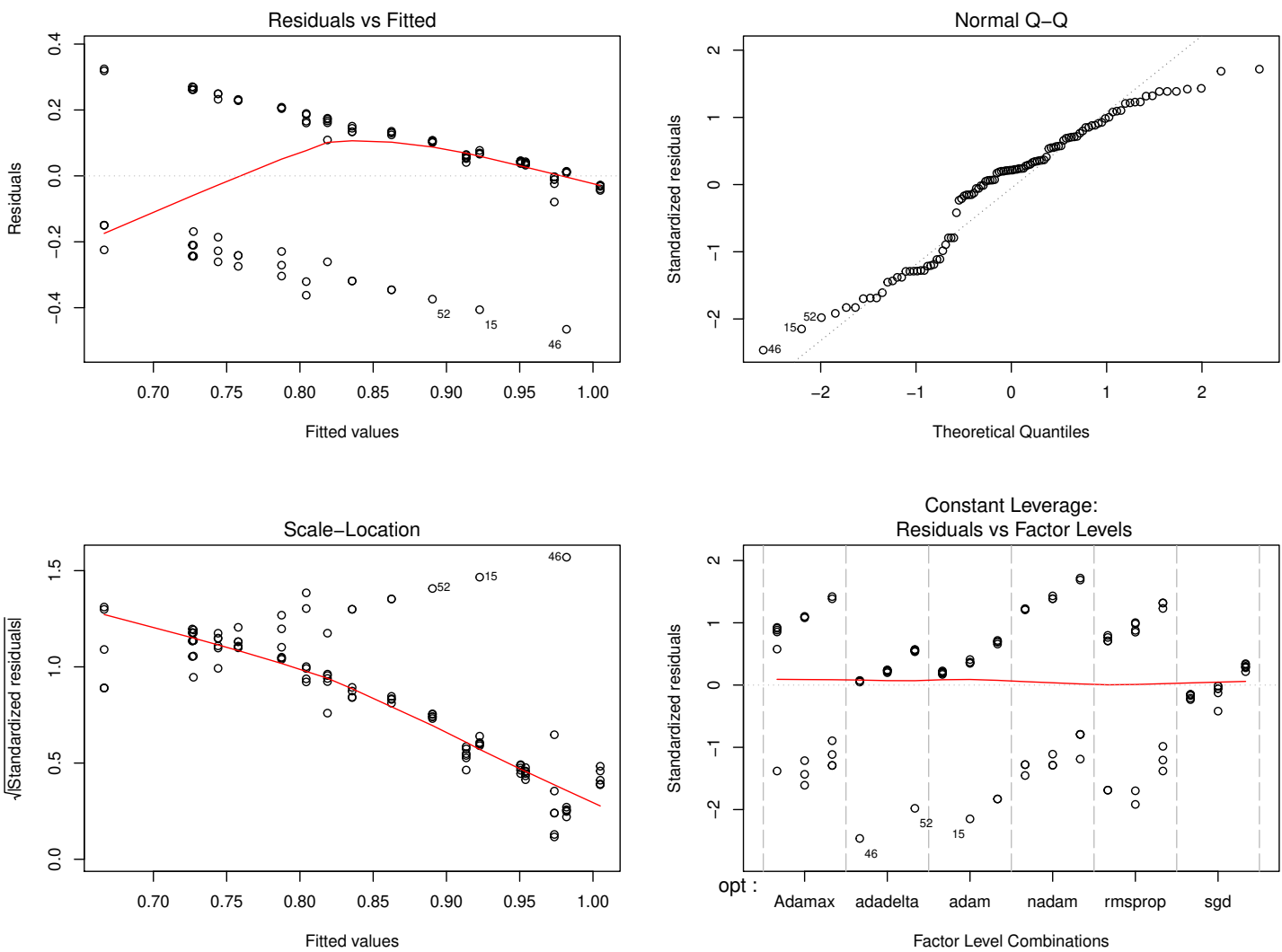


Figura 5.2: Interacciones entre los factores de los modelos.

Para poder pasar a la parte final del análisis de resultados se tienen que elegir una cantidad reducida de modelos para poder probarlos con el conjunto de imágenes de prueba, que como se explicó anteriormente se guardan para ser el último bloque de verificación del clasificador.

Con los resultados recién obtenidos, se puede concluir que los mejores modelos son aquellos que tienen los optimizadores Adadelta, SGD y Adam. El tamaño parece ser un factor influyente cuando se pone en conjunto con el optimizador, por lo que se elige tomar dos modelos de los tres mejores optimizadores, Adadelta, Adam y SGD, en cada uno de los tamaños (80×200 y 160×400) que presenten el valor más alto en su resultado de exactitud del conjunto de desarrollo.

Los modelos seleccionados se describen en la Tabla 5.4 dando la descripción de sus características de arquitectura, y terminando con la exactitud que alcanzaron en el conjunto de prueba. Se pueden apreciar también las métricas que se utilizan normalmente para calificar una herramienta de aprendizaje de máquina, como lo son la precisión, memoria (*recall*) y cómo se distribuyen los verdaderos positivos (A), verdaderos negativos (B), falsos positivos (C) y los falsos negativos (D). El resultado de los porcentajes son los niveles de exactitud usados usualmente por herramientas de aprendizaje de máquina.

Esto nos ayuda a balancear un poco el resultado de exactitud, ya que en el conjunto de prueba se tienen 130 imágenes con anomalía y 30 normales.

Tabla 5.4: Modelos seleccionados para la segunda parte de pruebas y su exactitud.

Modelo	Entrenamiento	Validación	Prueba	Precisión	Memoria	F1	A	B	C	D
1 2 1 1	99 %	99 %	70 %	80 %	83 %	81 %	109	3	27	21
1 2 3 1	99 %	99 %	69 %	80 %	81 %	81 %	106	5	25	24
1 2 1 5	95 %	97 %	68 %	80 %	81 %	80 %	106	4	26	24
1 3 1 5	94 %	97 %	68 %	79 %	81 %	80 %	106	3	27	24
1 2 2 6	99 %	99 %	70 %	79 %	85 %	82 %	111	2	28	19
1 1 3 6	99 %	99 %	66 %	79 %	80 %	79 %	104	3	27	26
2 2 2 1	99 %	100 %	63 %	79 %	75 %	77 %	98	4	26	32
2 1 3 1	99 %	99 %	72 %	80 %	86 %	83 %	113	3	27	17
2 1 1 5	96 %	97 %	61 %	78 %	72 %	75 %	94	4	26	36
2 1 3 5	96 %	97 %	61 %	78 %	72 %	75 %	94	4	26	36
2 2 2 6	99 %	99 %	65 %	78 %	79 %	78 %	103	2	28	27
2 2 3 6	99 %	99 %	64 %	79 %	75 %	77 %	98	5	25	32

Analizando estos resultados que se resaltan en azul en la Tabla 5.4 en podemos observar que el mejor resultado fue el del modelo 2 1 3 1, que es el de optimizador Adam con un tamaño de imagen de 160×400 . Tomando en cuenta los falsos negativos y falsos positivos, es uno de los modelos con resultados más bajos, pero aún así, siguen siendo bastante altos considerando el nivel de confianza que queremos alcanzar.

CAPÍTULO 6

CONCLUSIONES Y TRABAJO FUTURO.

Para finalizar en este capítulo se tienen las conclusiones después de revisar los resultados mostrados en la parte de análisis, tomando en cuenta cada uno de los aspectos que representaron un mayor desafío. Por último, se tienen los planes para trabajo a futuro partiendo de los resultados obtenidos en este trabajo.

†

ADAGRAD: Una de las primeras conclusiones a las que se llegó en este trabajo fue la de por qué el Adagrad resultó no ser un buen optimizador para los modelos. Una vez concluidos los resultados de la modelación y experimentación se buscó una respuesta del porqué el Adagrad podría estar mostrando resultados tan pobres, ya que según muchos artículos previamente revisados los optimizadores que se recomendaban más para realizar experimentaciones con redes neuronales en imágenes son los de métodos adaptativos en su selección de *Learning rate* ya que este es el valor que disminuye la pérdida y mejora el resultado de la exactitud.

Se encontraron varias respuestas para este problema. Una de ellas es que los parámetros que se tienen dentro del algoritmo no son los adecuados para el tipo de información que se le está incluyendo. En el caso de Adagrad, se le toma como un método adaptativo bueno cuando sus parámetros le ayudan a seleccionar un buen

learning rate. Si este es muy pequeño, el resultado se ciclará en algún punto hasta que se convierta en cero, en cuyo punto el algoritmo se estanca y no puede aprender más. En caso contrario, si este parámetro es muy grande, Adagrad hará que el resultado converja demasiado rápido en un resultado que no es bueno.

Para mejorar los resultados de estos modelos se podrían tomar en cuenta algunos ajustes a estos parámetros para observar si en realidad esto es lo que está causando un desempeño malo.

Otra de las razones no está orientada en el algoritmo en sí, mas en la forma en la que la computadora lo maneja. Han sido probados algunos casos en los que el algoritmo no interpreta la ayuda del GPU cuando se corren los modelos, por lo que en lugar de saber que se están haciendo varias líneas de cálculos al mismo tiempo, este solo le da seguimiento a una línea, lo que ocasiona que adapte sus pesos de manera correcta.

Esto puede ser corregido utilizando los valores en el CPU solamente, para que las ventajas de la programación paralela no se vuelvan el problema de que este algoritmo no se desarrolle como debería.

La última opción que se contempla, y esta es simplemente especulativa, es el hecho de que este algoritmo basa sus resultados en la memoria que tiene para recordar los gradientes anteriores que modificó. Esto puede probar a ser una desventaja cuando se sabe que la fortaleza de este algoritmo es cuando la información de entrada tiene muchos campos inservibles. Lo que pasará con darle imágenes en las que cada detalle cuenta, esta pila de memoria se llenará muy rápido, lo que ocasionará que el algoritmo estanque el valor de pérdida, y no mejore su resultado en la exactitud.

Esto podría arreglarse si incluimos el valor de Dropout, que ayudará a quitar un poco esa carga de información que va acumulando el algoritmo al intentar mejorar los pesos del modelo.

BASES DE DATOS: Al momento de realizar la sección de estado del arte, se pudieron sacar conclusiones de algunos de los trabajos. En los que se decidió basarse por la similitudes de objetivos que se tenían se vieron resultados de arriba de un 80 % en la clasificación final como se describen en los artículos mencionados en la sección de antecedentes, mas en todos los casos se utilizaba una sola base de datos. El único artículo encontrado que manejaba dos bases de datos, no tuvo un resultado mejor que el presentado en este trabajo.

Gracias a varias investigaciones realizadas en libros y cursos de Machine Learning podemos afirmar que el uso de más bases de datos dentro de los conjuntos de entrenamiento, desarrollo y prueba para un modelo de red neuronal la hace más robusta, pero a su vez puede ocasionar un bajo nivel de exactitud, ya que se tienen distribuciones, calidades y formatos distintos en cada una de las bases utilizadas.

Una de las opciones presentadas en trabajos futuros será la de manejar solo una de las bases de datos para verificar si los resultados son los mismos o logran mejorar. Otra alternativa es la de concentrarse solamente en la base de datos INCAN, ya que es la más reciente y tiene mejor calidad en sus datos. Para esto se tendrá que trabajar en conjunto con la asociación para que nos permita trabajar no solo con imágenes con anomalía, si no también con imágenes sin ella.

Otra de las preguntas que se presentan al terminar este trabajo es si se tomó la desicion correcta al normalizar las imágenes en la base de datos. Para poder conservar la calidad en las imágenes nuevas del INCAN se decidió modificar las imágenes de las otras dos bases de datos, pero ahora haciendo un poco más de búsqueda en cuando a resolución de imágenes y sus posibles pérdidas de información tal vez sería una mejor idea el dejar las imágenes de las bases de datos antiguas en su formato original (canales blanco y negro) y en lugar modificar las imágenes de la base de datos más actual, ya que sacrificar un poco de las características de calidad que se intentaron proteger en la primer metodología, puede que tenga un mejor resultado en cuanto a exactitud.

CONJUNTO DE PRUEBA: Para realizar la parte de prueba en los modelos seleccionados en el diseño de experimentos se separaron 160 imágenes de la base de datos del INCAN, siendo 30 de ellas normales y las demás con anomalía. El motivo para utilizar solo imágenes del INCAN es porque la herramienta está orientada a ayudar en el proceso de detección para ese hospital.

Siguiendo algunos de los consejos dados en el libro de Machine Learning Yearning de Andrew NG, se revisaron los resultados de las imágenes que más trabajo le costaba a los modelos identificar, y se encontró con que no califica correctamente la mayor cantidad de imágenes normales.

Una de las razones por las que esto pueda pasar es porque dentro de los conjuntos de entrenamiento y desarrollo no se introdujeron imágenes normales de la base de datos del INCAN, ya que en el momento de hacer los experimentos, solo se contaban con 200 imágenes con anomalías proporcionadas por dicho hospital. Las únicas instancias de imágenes normales que se proporcionaron a estos conjuntos venían de las bases de datos Mini mias y DDSM, por lo que se sospecha que su falta de calidad tuvo un valor importante en la parte de su desempleo en el conjunto de prueba.

OPTIMIZADOR SGD: Uno de los optimizadores que obtuvo mejores resultados en el diseño de experimentos en las dos dimensiones de tamaño, a pesar de no tener los mejores resultados en cuanto a exactitud comparado con los demás optimizadores es el SGD.

Haciendo una investigación más a fondo sobre este optimizador nos damos cuenta que es uno de los algoritmos más confiables en cuanto a resultados positivos para el entrenando de una red neuronal con imágenes, más su tiempo estimado para poder converger en uno de estos resultados es mucho más largo que los otros optimizadores.

Una propuesta más para trabajo a futuro es explorar estos modelos de optimi-

zador SGD pero dándole más iteraciones para darle la oportunidad de converger en un buen resultado.

CANTIDAD DE CAPAS: El modelo inicial que se tenía en mente para este trabajo es el de una arquitectura lo más simple posible para una buena clasificación en imágenes de mamografías, por lo que se empezó un la arquitectura más básica para una red neuronal convolucional, con dos capas.

TRABAJO FUTURO: Una propuesta para trabajo a futuro es explorar una mayor cantidad de capas dentro de la arquitectura del modelo para ver si eso ayuda a mejorar a los resultados. Teniendo más capas también abre la puerta a utilizar la herramienta del *Dropout* que se ignoró en las experimentaciones de este trabajo porque, al ser una cantidad tan reducida de capas, mostraba ser fuente de problemas para tener un buen desempeño.

APÉNDICE A

CÓDIGOS EN PYTHON

A.1 FILTROS DE IMÁGENES

En este código se utiliza la librería de PIL para aplicar filtros a las imágenes como forma de augmentación de datos. El conjunto de prueba (conjunto final) no pasó por este paso.

```
from PIL import Image, ImageFilter, ImageEnhance
import os
for f in os.listdir('.'):
    if f.endswith('.png'):
        i = Image.open(f)
        fn, fext = os.path.splitext(f)
        i.rotate(180).save('{}-180.png'.format(fn))
        i.filter(ImageFilter.GaussianBlur())./
        save('{}_gauss.png'.format(fn))
        i.filter(ImageFilter.MinFilter(5))./
        save('{}-minfil.png'.format(fn))
        i.filter(ImageFilter.BLUR)./
        save('{}-blur.png'.format(fn))
        i.filter(ImageFilter.EDGE_ENHANCE)./
        save('{}-edge.png'.format(fn))
        i.filter(ImageFilter.MaxFilter(5))./
        save('{}-maxfil.png'.format(fn))
```

```
i.filter(ImageFilter.EDGE_ENHANCE_MORE)./  
save('{}-edgemore.png'.format(fn))  
i.filter(ImageFilter.SMOOTH_MORE)./  
save('{}-smoothmor.png'.format(fn))  
i.filter(ImageFilter.SHARPEN)./  
save('{}-sharp.png'.format(fn))
```

A.2 REDIMENSIÓN DE IMÁGENES

En esta parte del código de Python se quitan las máscaras negras a las imágenes y se modifican todas en cuanto a sus dimensiones para normalizarlas y que puedan ser alimentadas a la red neuronal.

```
from PIL import Image, ImageChops  
import os  
  
def trim(im):  
    bg = Image.new(im.mode, im.size, im.getpixel((0,0)))  
    diff = ImageChops.difference(im, bg)  
    diff = ImageChops.add(diff, diff, 2.0, -95)  
    bbox = diff.getbbox()  
    if bbox:  
        return im.crop(bbox)  
  
if __name__ == "__main__":  
    for f in os.listdir('.'):   
        if f.endswith('.png'):  
            img = Image.open(f)  
            fn, fext = os.path.splitext(f)  
            cropped_img = trim(img)  
            resize = cropped_img.resize((80, 200),/  
                Image.ANTIALIAS)  
            resize.save('crop/{f}.png'.format(fn))
```

A.3 DIVISIÓN DE LOS CONJUNTOS

Se utilizaron las siguientes líneas de código para hacer la separación de manera aleatoria de las imágenes dentro de los conjuntos de entrenamiento, validación y prueba.

```
import os
source1 = "/Users/mayraberrones/Desktop/Mini-mias/a-train"
dest11 = "/Users/mayraberrones/Desktop/Mini-mias/a-test"
files = os.listdir(source1)
import shutil
import numpy as np
for f in files:
    if np.random.rand(1) < 0.025:
        shutil.move(source1 + '/' + f, dest11 + '/' + f)
```

A.4 MODELO DE RED

Aquí se describen como están conformadas las capas ocultas y la capa final de la estructura de red. Sobre este modelo se cambiaron todos sus parámetros.

```
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers.core import Activation

from keras.models import load_model

import numpy as np

# Inicializando la RNC
```

```
classifier = Sequential()

# Paso 1- Convolucion
classifier.add(Conv2D(32, (3, 3),
                    input_shape = (80, 200, 3),
                    activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))

# Segunda capa
classifier.add(Conv2D(32, (3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))

classifier.add(Flatten())

# Conexion
classifier.add(Dense(units= 128, activation = 'relu'))
classifier.add(Dense(units = 1, activation = 'sigmoid'))

#Compilar
classifier.compile(optimizer = 'adam',
                  loss = 'binary_crossentropy',
                  metrics = ['accuracy'])

# Part 2 - Fitting the CNN to the images

from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory(
```

```
        '/Users/mayraberrones/Desktop/original/train-s',
        target_size = (80, 200),
        class_mode = 'binary')

test_set = test_datagen.flow_from_directory(
        '/Users/mayraberrones/Desktop/original/dev-s',
        target_size = (80, 200),
        batch_size = 32,
        class_mode = 'binary')

classifier.fit_generator(training_set,
                        steps_per_epoch = 1000,
                        epochs = 10,
                        validation_data = test_set,
                        validation_steps = 2000)

classifier.save('modelo-001.h5')
```

A.5 RESULTADOS CONJUNTO DE PRUEBA

A.5.1 RE NOMBRAR IMÁGENES

Para poder dar la última experimentación con el conjunto de prueba fue necesario cambiarles el nombre a normal y anormal para poder tener ambas en una sola carpeta y de esta manera medir el porcentaje de correctas e incorrectas.

```
from PIL import Image, ImageFilter, ImageEnhance
import os
i = 0
for f in os.listdir('.'):
    if f.endswith('.png'):
        i = i+1
        im = Image.open(f)
        fn, fext = os.path.splitext(f)
```

```
im.save('no-{}.png'.format(i))
```

A.5.2 PRUEBA

Este código es el que se utilizó para mandar a llamar a los modelos seleccionados, y probarlos con imágenes NO contenidas dentro del conjunto de entrenamiento y de validación. El umbral que se eligió fue de 0.5, ya que es el mismo utilizado por KERAS en el conjunto de Validación para realizar el *Binary crossentropy*

```
from keras.models import load_model
from keras.preprocessing import image
import matplotlib.pyplot as plt
import numpy as np
import os

def load_image(img_path, show=False):

    img = image.load_img(img_path, target_size=(160, 400))
    img_tensor = image.img_to_array(img)
    img_tensor = np.expand_dims(img_tensor, axis=0)
    img_tensor /= 255.

    if show:
        plt.imshow(img_tensor[0])
        plt.axis('off')
        plt.show()

    return img_tensor

if __name__ == "__main__":
    from PIL import Image
    model = load_model('/home/mayra/Escritorio/size2-dis1/modelo-064.h5')
    for f in os.listdir('.'):
        if f.endswith('.png'):
```

```

new_image = load_image(f)
fn, fext = os.path.splitext(f)
pred = model.predict(new_image)
if pred > .5:
    print('{}_1'.format(fn))
else:
    print('{}_0'.format(fn))

```

A.6 PRUEBAS ESTADÍSTICAS EN PYTHON

```

import statsmodels.api as sm
from statsmodels.formula.api import ols
from numpy import isnan
import pandas as pd
import ssl

if getattr(ssl, '_create_unverified_context', None):
    ssl._create_default_https_context = ssl._create_unverified_context
o = pd.read_csv("https://raw.githubusercontent.com/
mayraberones94/Modelos-Probabilisticos/master/prueba1/todo-exp.csv")
d = pd.DataFrame(o, columns = ["opt", "ocu", "fin", "tam", "dev"])
d = d.dropna()
m = ols('dev ~ opt + ocu + fin + tam', data = d).fit()
a = sm.stats.anova_lm(m, typ = 2)
print(a)
n = len(a)
alpha = 0.01
for i in range(n):
    print("{}_{:s}es significativo".format(a.index[i],
        " if a['PR(>F)'][i] < alpha else "no"))

m2 = ols('dev ~ ocu * tam + ocu * opt + opt * fin +
ocu * fin + tam * fin + tam * opt',
        data = d).fit()

```

```
a2 = sm.stats.anova_lm(m, typ = 2)
print(a2)
n = len(a2)
alpha = 0.01
for i in range(n):
    print("{:s}_{:s}es_significativo".format(a.index[i],
    " if a['PR(>F)'][i] < alpha else "NO_"))
```

APÉNDICE B

CÓDIGOS EN R

B.1 GRÁFICAS Y PRUEBAS ESTADÍSTICAS

Código para realizar el ANOVA de tres factores en R.

```
datos <- read.table("ultimo.txt", header = TRUE)

opt<- as.factor(datos[,1])
fin <- as.factor(datos[,2])
ocu <- as.factor(datos[,3])
acc <- datos[,4]
datos$ocu = factor(datos$ocu)
datos$fin = factor(datos$fin)
head(datos)
str(datos)
plot.design(acc ~ ., data = datos)

op <- par(mfrow = c(3, 1))
with(datos, {
  interaction.plot(opt, fin, acc)
  interaction.plot(opt, ocu, acc)
  interaction.plot(fin, ocu, acc)
})
)
```

```
par(op)
fm <- aov(acc ~ opt + fin + ocu + opt*fin + opt*ocu +
          fin*ocu + opt*fin*ocu, data = datos)
summary(fm)
fm <- update(fm, . ~ . - opt:fin:ocu )
summary(fm)

fml <- update(fm, .~opt+fin+ocu)
summary(fml)

anova(fm, fml)

model.tables(fml, type = "effects")

model.tables(fml, type = "means")

fm <- aov(acc ~ opt+fin, data = datos)
#fm <- aov(acc ~ opt, data = datos)
summary(fm)

op <- par(mfrow = c(2, 2))
plot(fm)
par(op)
```

B.1.1 RESPUESTA TERMINAL

```
[Workspace loaded from
~/Documents/GitHub/Modelos-Probabilisticos/prueba1/.RData]

> datos <- read.table("ultimo.txt", header = TRUE)
>
> opt <- as.factor(datos[,1])
> fin <- as.factor(datos[,2])
> ocu <- as.factor(datos[,3])
```

```
> acc <- datos[,4]
> datos$ocu = factor(datos$ocu)
> datos$fin = factor(datos$fin)
> head(datos)
  opt fin ocu   acc
1 adam 128  32 0.9915
2 adam 128  64 0.9932
3 adam 128 128 0.9863
4 adam 240  32 0.9885
5 adam 240  64 0.9911
6 adam 240 128 0.9902
> str(datos)
'data.frame':  27 obs. of  4 variables:
 $ opt: Factor w/ 3 levels
 "adadelta","adam" ,...: 2 2 2 2 2 2 2 2 2 2 3 ...
 $ fin: Factor w/ 3 levels
 "128","240","380": 1 1 1 2 2 2 3 3 3 1 ...
 $ ocu: Factor w/ 3 levels
 "32","64","128": 1 2 3 1 2 3 1 2 3 1 ...
 $ acc: num  0.992 0.993 0.986 0.989 0.991 ...
> plot.design(acc ~ ., data = datos)
>
> op <- par(mfrow = c(3, 1))
> with(datos, {
+   interaction.plot(opt, fin, acc)
+   interaction.plot(opt, ocu, acc)
+   interaction.plot(fin, ocu, acc)
+ })
+ )
> par(op)
> fm <- aov(acc ~ opt + fin + ocu + opt*fin + opt*ocu +
+           fin*ocu + opt*fin*ocu, data = datos)
> summary(fm)
              Df  Sum Sq  Mean Sq
opt              2 0.005648 0.0028241
fin              2 0.000651 0.0003255
```

```

ocu          2 0.000710 0.0003552
opt:fin      4 0.000920 0.0002299
opt:ocu      4 0.001135 0.0002836
fin:ocu      4 0.000804 0.0002009
opt:fin:ocu  8 0.001201 0.0001501
> fm <- update(fm, . ~ . - opt:fin:ocu )
> summary(fm)
              Df  Sum Sq  Mean Sq F value  Pr(>F)
opt           2 0.005648 0.0028241  18.819 0.000944 ***
fin           2 0.000651 0.0003255   2.169 0.176763
ocu           2 0.000710 0.0003552   2.367 0.155768
opt:fin       4 0.000920 0.0002299   1.532 0.281097
opt:ocu       4 0.001135 0.0002836   1.890 0.205636
fin:ocu       4 0.000804 0.0002009   1.339 0.335248
Residuals    8 0.001201 0.0001501

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> fml <- update(fm, .~opt+fin+ocu)
> summary(fml)
              Df  Sum Sq  Mean Sq F value  Pr(>F)
opt           2 0.005648 0.0028241  13.917 0.000163 ***
fin           2 0.000651 0.0003255   1.604 0.225918
ocu           2 0.000710 0.0003552   1.750 0.199279
Residuals    20 0.004059 0.0002029

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> anova(fm, fml)
Analysis of Variance Table

Model 1: acc ~ opt + fin + ocu + opt:fin + opt:ocu + fin:ocu
Model 2: acc ~ opt + fin + ocu
  Res.Df    RSS  Df Sum of Sq    F Pr(>F)
1       8 0.0012005
2      20 0.0040585 -12 -0.002858 1.5871 0.2608

```

```
>
> model.tables(fml, type = "effects")
Tables of effects

  opt
opt
  adadelta      adam      sgd
0.011611  0.008778 -0.020389

  fin
fin
      128      240      380
0.003133 -0.006933  0.003800

  ocu
ocu
      32      64      128
-0.007200  0.004367  0.002833
>
> model.tables(fml, type = "means")
Tables of means
Grand mean

0.9817222

  opt
opt
  adadelta      adam      sgd
  0.9933  0.9905  0.9613

  fin
fin
      128      240      380
0.9849 0.9748 0.9855

  ocu
```

```
ocu
  32   64   128
0.9745 0.9861 0.9846
>
> fm <- aov(acc ~ opt+fin, data = datos)
> #fm <- aov(acc ~ opt, data = datos)
> summary(fm)
              Df  Sum Sq  Mean Sq F value  Pr(>F)
opt              2  0.005648  0.0028241  13.028 0.000185 ***
fin              2  0.000651  0.0003255   1.501 0.244759
Residuals      22  0.004769  0.0002168
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> op <- par(mfrow = c(2, 2))
> plot(fm)
> par(op)
```

APÉNDICE C

RESULTADOS COMPLETOS

Modelos descritos por identificador, y sus características en el orden establecido en la sección de metodología con el porcentaje de exactitud en entrenamiento y validación.

Se describe la columna de capa convolucional como oculta, y la capa de aplastamiento como final.

Tabla C.1: Desempeño de los modelos en los conjuntos de entrenamiento y validación.

Modelo	Parámetros				Exactitud	
	Tamaño	Oculto	Final	Optimizador	Entrenamiento	Validación
1 1 1 1	80×200	32	128	Adam	99.19 %	99.15 %
1 2 1 1	80×200	64	128	Adam	99.26 %	99.32 %
1 3 1 1	80×200	128	128	Adam	99.31 %	98.63 %
1 1 2 1	80×200	32	240	Adam	99.30 %	98.85 %
1 2 2 1	80×200	64	240	Adam	99.37 %	99.11 %
1 3 2 1	80×200	128	240	Adam	99.40 %	99.02 %
1 1 3 1	80×200	32	380	Adam	99.25 %	98.68 %
1 2 3 1	80×200	64	380	Adam	99.42 %	99.41 %
1 3 3 1	80×200	128	380	Adam	99.35 %	99.28 %

Continúa en la próxima página

Tabla C.1 – *Continuación*

Modelo	Parámetros				Exactitud	
	Tamaño	Oculto	Final	Optimizador	Entrenamiento	Validación
1 1 1 2	80×200	32	128	Adamax	98.46 %	97.99 %
1 2 1 2	80×200	64	128	Adamax	55.39 %	55.80 %
1 3 1 2	80×200	128	128	Adamax	99.28 %	98.56 %
1 1 2 2	80×200	32	240	Adamax	99.14 %	99.40 %
1 2 2 2	80×200	64	240	Adamax	99.47 %	99.15 %
1 3 2 2	80×200	128	240	Adamax	55.29 %	55.81 %
1 1 3 2	80×200	32	380	Adamax	99.52 %	98.90 %
1 2 3 2	80×200	64	380	Adamax	99.41 %	99.57 %
1 3 3 2	80×200	128	380	Adamax	55.22 %	55.83 %
1 1 1 3	80×200	32	128	Nadam	99.10 %	98.98 %
1 2 1 3	80×200	64	128	Nadam	99.10 %	98.60 %
1 3 1 3	80×200	128	128	Nadam	99.10 %	98.77 %
1 1 2 3	80×200	32	240	Nadam	99.23 %	98.85 %
1 2 2 3	80×200	64	240	Nadam	99.25 %	99.71 %
1 3 2 3	80×200	128	240	Nadam	99.13 %	98.85 %
1 1 3 3	80×200	32	380	Nadam	99.29 %	99.10 %
1 2 3 3	80×200	64	380	Nadam	99.37 %	98.50 %
1 3 3 3	80×200	128	380	Nadam	44.69 %	44.18 %
1 1 1 4	80×200	32	128	RMSProp	98.23 %	98.64 %
1 2 1 4	80×200	64	128	RMSProp	96.69 %	96.90 %
1 3 1 4	80×200	128	128	RMSProp	96.78 %	96.98 %
1 1 2 4	80×200	32	240	RMSProp	98.76 %	96.48 %
1 2 2 4	80×200	64	240	RMSProp	98.50 %	99.37 %
1 3 2 4	80×200	128	240	RMSProp	44.75 %	44.22 %
1 1 3 4	80×200	32	380	RMSProp	98.60 %	99.28 %
1 2 3 4	80×200	64	380	RMSProp	55.33 %	55.81 %

Continúa en la próxima página

Tabla C.1 – *Continuación*

Modelo	Parámetros				Exactitud	
	Tamaño	Oculto	Final	Optimizador	Entrenamiento	Validación
1 3 3 4	80×200	128	380	RMSProp	97.39 %	97.65 %
1 1 1 5	80×200	32	128	SGD	93.62 %	96.08 %
1 2 1 5	80×200	64	128	SGD	95.53 %	97.61 %
1 3 1 5	80×200	128	128	SGD	94.69 %	97.66 %
1 1 2 5	80×200	32	240	SGD	96.18 %	89.45 %
1 2 2 5	80×200	64	240	SGD	95.81 %	97.11 %
1 3 2 5	80×200	128	240	SGD	94.93 %	96.28 %
1 1 3 5	80×200	32	380	SGD	96.20 %	96.85 %
1 2 3 5	80×200	64	380	SGD	96.20 %	96.59 %
1 3 3 5	80×200	128	380	SGD	96.01 %	97.57 %
1 1 1 6	80×200	32	128	Adadelta	99.51 %	99.45 %
1 2 1 6	80×200	64	128	Adadelta	99.25 %	99.36 %
1 3 1 6	80×200	128	128	Adadelta	99.57 %	99.11 %
1 1 2 6	80×200	32	240	Adadelta	99.53 %	98.81 %
1 2 2 6	80×200	64	240	Adadelta	99.65 %	99.57 %
1 3 2 6	80×200	128	240	Adadelta	99.66 %	99.11 %
1 1 3 6	80×200	32	380	Adadelta	99.64 %	99.75 %
1 2 3 6	80×200	64	380	Adadelta	99.60 %	99.40 %
1 3 3 6	80×200	128	380	Adadelta	99.63 %	99.44 %
1 1 1 7	80×200	32	128	Adagrad	55.21 %	55.82 %
1 2 1 7	80×200	64	128	Adagrad	99.07 %	98.25 %
1 3 1 7	80×200	128	128	Adagrad	44.65 %	44.20 %
1 1 2 7	80×200	32	240	Adagrad	44.69 %	44.16 %
1 2 2 7	80×200	64	240	Adagrad	55.29 %	55.78 %
1 3 2 7	80×200	128	240	Adagrad	55.28 %	55.81 %
1 1 3 7	80×200	32	380	Adagrad	55.24 %	55.80 %

Continúa en la próxima página

Tabla C.1 – *Continuación*

Modelo	Parámetros				Exactitud	
	Tamaño	Oculto	Final	Optimizador	Entrenamiento	Validación
1 2 3 7	80×200	64	380	Adagrad	44.73 %	44.19 %
1 3 3 7	80×200	128	380	Adagrad	55.33 %	55.81 %
2 1 1 1	160×400	32	128	Adam	99.44 %	99.69 %
2 2 1 1	160×400	64	128	Adam	99.53 %	98.95 %
2 3 1 1	160×400	128	128	Adam	99.48 %	99.37 %
2 1 2 1	160×400	32	240	Adam	99.33 %	99.21 %
2 2 2 1	160×400	64	240	Adam	99.26 %	100.00 %
2 3 2 1	160×400	128	240	Adam	49.73 %	51.66 %
2 1 3 1	160×400	32	380	Adam	99.41 %	99.79 %
2 2 3 1	160×400	64	380	Adam	49.64 %	51.68 %
2 3 3 1	160×400	128	380	Adam	49.82 %	51.68 %
2 1 1 2	160×400	32	128	Adamax	99.26 %	99.37 %
2 2 1 2	160×400	64	128	Adamax	99.44 %	99.10 %
2 3 1 2	160×400	128	128	Adamax	91.26 %	92.78 %
2 1 2 2	160×400	32	240	Adamax	49.80 %	51.66 %
2 2 2 2	160×400	64	240	Adamax	99.64 %	99.58 %
2 3 2 2	160×400	128	240	Adamax	50.25 %	48.35 %
2 1 3 2	160×400	32	380	Adamax	50.41 %	48.32 %
2 2 3 2	160×400	64	380	Adamax	49.89 %	51.66 %
2 3 3 2	160×400	128	380	Adamax	50.10 %	48.35 %
2 1 1 3	160×400	32	128	Nadam	50.28 %	48.34 %
2 2 1 3	160×400	64	128	Nadam	49.70 %	51.64 %
2 3 1 3	160×400	128	128	Nadam	49.83 %	51.63 %
2 1 2 3	160×400	32	240	Nadam	49.72 %	51.68 %
2 2 2 3	160×400	64	240	Nadam	50.25 %	48.34 %
2 3 2 3	160×400	128	240	Nadam	50.16 %	48.32 %

Continúa en la próxima página

Tabla C.1 – *Continuación*

Modelo	Parámetros				Exactitud	
	Tamaño	Oculto	Final	Optimizador	Entrenamiento	Validación
2 1 3 3	160×400	32	380	Nadam	49.81 %	51.67 %
2 2 3 3	160×400	64	380	Nadam	49.64 %	51.66 %
2 3 3 3	160×400	128	380	Nadam	49.70 %	51.67 %
2 1 1 4	160×400	32	128	RMSProp	49.81 %	51.68 %
2 2 1 4	160×400	64	128	RMSProp	98.10 %	98.00 %
2 3 1 4	160×400	128	128	RMSProp	49.72 %	51.67 %
2 1 2 4	160×400	32	240	RMSProp	98.56 %	99.00 %
2 2 2 4	160×400	64	240	RMSProp	98.32 %	97.04 %
2 3 2 4	160×400	128	240	RMSProp	50.36 %	48.35 %
2 1 3 4	160×400	32	380	RMSProp	98.52 %	99.37 %
2 2 3 4	160×400	64	380	RMSProp	49.77 %	51.65 %
2 3 3 4	160×400	128	380	RMSProp	50.41 %	48.34 %
2 1 1 5	160×400	32	128	SGD	96.63 %	97.63 %
2 2 1 5	160×400	64	128	SGD	96.49 %	97.31 %
2 3 1 5	160×400	128	128	SGD	96.19 %	96.52 %
2 1 2 5	160×400	32	240	SGD	97.35 %	96.27 %
2 2 2 5	160×400	64	240	SGD	96.30 %	94.99 %
2 3 2 5	160×400	128	240	SGD	96.31 %	97.05 %
2 1 3 5	160×400	32	380	SGD	96.85 %	97.83 %
2 2 3 5	160×400	64	380	SGD	97.65 %	95.41 %
2 3 3 5	160×400	128	380	SGD	96.98 %	97.04 %
2 1 1 6	160×400	32	128	Adadelta	49.67 %	51.65 %
2 2 1 6	160×400	64	128	Adadelta	99.38 %	99.42 %
2 3 1 6	160×400	128	128	Adadelta	99.47 %	99.58 %
2 1 2 6	160×400	32	240	Adadelta	99.62 %	99.32 %
2 2 2 6	160×400	64	240	Adadelta	99.60 %	99.63 %

Continúa en la próxima página

Tabla C.1 – *Continuación*

Modelo	Parámetros				Exactitud	
	Tamaño	Oculto	Final	Optimizador	Entrenamiento	Validación
2 3 2 6	160×400	128	240	Adadelta	99.52 %	99.58 %
2 1 3 6	160×400	32	380	Adadelta	49.81 %	51.64 %
2 2 3 6	160×400	64	380	Adadelta	99.56 %	99.84 %
2 3 3 6	160×400	128	380	Adadelta	99.55 %	99.15 %
2 1 1 7	160×400	32	128	Adagrad	50.23 %	48.33 %
2 2 1 7	160×400	64	128	Adagrad	50.37 %	48.35 %
2 3 1 7	160×400	128	128	Adagrad	49.65 %	51.65 %
2 1 2 7	160×400	32	240	Adagrad	49.77 %	51.67 %
2 2 2 7	160×400	64	240	Adagrad	50.36 %	48.35 %
2 3 2 7	160×400	128	240	Adagrad	50.31 %	48.34 %
2 1 3 7	160×400	32	380	Adagrad	49.75 %	51.67 %
2 2 3 7	160×400	64	380	Adagrad	50.22 %	48.33 %
2 3 3 7	160×400	128	380	Adagrad	50.22 %	48.33 %

BIBLIOGRAFÍA

- Abdel-Zaher, A. M. and Eldeib, A. M. (2016). Breast cancer classification using deep belief networks. *Expert Systems with Applications*, 46:139–144.
- ACS (2019). What is cancer? Disponible en: "<https://www.cancer.org/cancer/cancer-basics/what-is-cancer.html>".
- Ahn, E., Kumar, A., Feng, D., Fulham, M., and Kim, J. (2019). Unsupervised deep transfer feature learning for medical image classification. *arXiv preprint arXiv:1903.06342*.
- AI, S. (2017). A beginner's guide to convolutional neural networks (cnns). Disponible en: "<https://skymind.ai/wiki/convolutional-network>".
- Aibar, L., Santalla, A., López-Criado, M., González-Pérez, I., Calderón, M., Gallo, J., and Fernández-Parra, J. (2011). Clasificación radiológica y manejo de las lesiones mamarias. *Clínica e Investigación en Ginecología y Obstetricia*, 38(4):141–149.
- AmericanCancerSociety. (2006). Qué es el cáncer de seno. Disponible en: "<https://www.cancer.org/es/cancer/cancer-de-seno/acerca/que-es-el-cancer-de-seno.html>".
- Anderson, J. A. and Rosenfeld, E. (2000). *Talking nets: An oral history of neural networks*. MIT Press.
- AsimovInstitute (2017). The neural network zoo. Disponible en: "<http://www.asimovinstitute.org/neural-network-zoo/>".

- Asri, H., Mousannif, H., Al Moatassime, H., and Noel, T. (2016). Using machine learning algorithms for breast cancer risk prediction and diagnosis. *Procedia Computer Science*, 83:1064–1069.
- Ayyachamy, S., Alex, V., Khened, M., and Krishnamurthi, G. (2019). Medical image retrieval using resnet-18 for clinical diagnosis. In *Medical Imaging 2019: Imaging Informatics for Healthcare, Research, and Applications*, volume 10954, page 1095410. International Society for Optics and Photonics.
- Baidu (2018). fluid optimizer. Disponible en: "<http://www.paddlepaddle.org/documentation/api/en/0.15.0/optimizer.html#permlink-12-adamaxoptimizer>".
- Basheer, I. A. and Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, 43(1):3–31.
- Becker, A. S., Marcon, M., Ghafoor, S., Wurnig, M. C., Frauenfelder, T., and Boss, A. (2017). Deep learning in mammography: diagnostic accuracy of a multipurpose image analysis software in the detection of breast cancer. *Investigative radiology*, 52(7):434–440.
- Bozek, J., Mustra, M., Delac, K., and Grgic, M. (2009). A survey of image processing algorithms in digital mammography. In *Recent advances in multimedia signal processing and communications*, pages 631–657. Springer.
- BreastCancerOrganization (2015). Factores de riesgo del cáncer de mama. Disponible en: "<https://www.breastcancer.org/es/riesgo/factores>".
- Calvo, D. (2018). Función de activación - redes neuronales. Disponible en: "<http://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/>", journal=Función de activación – Redes neuronales.

- CDC (2018). Cdc - ¿cuáles son los factores de riesgo del cáncer de mama? Disponible en: "https://www.cdc.gov/spanish/cancer/breast/basic_info/risk_factors.html".
- Cheng, H.-D., Cai, X., Chen, X., Hu, L., and Lou, X. (2003). Computer-aided detection and classification of microcalcifications in mammograms: a survey. *Pattern recognition*, 36(12):2967–2991.
- Cheng, H.-D., Shi, X., Min, R., Hu, L., Cai, X., and Du, H. (2006). Approaches for automated detection and classification of masses in mammograms. *Pattern recognition*, 39(4):646–668.
- Chengwei, Z. (2018). Quick notes on how to choose optimizer in keras. Disponible en: "<https://www.dlology.com/blog/quick-notes-on-how-to-choose-optimizer-in-keras/>".
- Chollet, F. (2018). *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG.
- Clancy, K., Zhang, L., Mohamed, A., Aboutalib, S., Berg, W., and Wu, S. (2019). Deep learning for identifying breast cancer malignancy and false recalls: a robustness study on training strategy. In *Medical Imaging 2019: Computer-Aided Diagnosis*, volume 10950, page 1095005. International Society for Optics and Photonics.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- División de Prevención y Control del Cáncer, C. p. e. C. y. l. P. d. E. (2018). Que es la mamografía. Disponible en: "<https://www.cdc.gov>".
- Dorsi, C., Bassett, L., Feig, S., et al. (1998). Breast imaging reporting and data system (bi-rads). *Breast imaging atlas, 4th edn. American College of Radiology, Reston Google Scholar*.

- Eldridge, L. (2019). Cancer cells vs. normal cells: How are they different? Disponible en: "<https://www.verywellhealth.com/cancer-cells-vs-normal-cells-2248794>".
- Ertosun, M. G. and Rubin, D. L. (2015). Probabilistic visual search for masses within mammography images using deep learning. In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1310–1315. IEEE.
- Foundation, P. S. (2019). Python 3.7.2 documentation. Disponible en: "<https://docs.python.org/3/>".
- Freer, T. W. and Ulissey, M. J. (2001). Screening mammography with computer-aided detection: prospective study of 12,860 patients in a community breast center. *Radiology*, 220(3):781–786.
- Ganesan, K., Acharya, U. R., Chua, C. K., Min, L. C., Abraham, K. T., and Ng, K.-H. (2012). Computer-aided breast cancer detection using mammograms: a review. *IEEE Reviews in biomedical engineering*, 6:77–98.
- Giger, M. L., Karssemeijer, N., and Schnabel, J. A. (2013). Breast image analysis for risk assessment, detection, diagnosis, and treatment of cancer. *Annual review of biomedical engineering*, 15:327–357.
- Godoy, D. (2018). Understanding binary cross-entropy log loss: a visual explanation. Disponible en: "<https://towardsdatascience.com/>".
- Goldbloom, A., Hamner, B., and Mosner, J. (2019). Your home for data science. Disponible en: "<https://www.kaggle.com/>".
- González, J. D. R. (2019). Cáncer de mama. Disponible en: "<https://doctorrojas.com.mx/cancer-de-mama/>".
- Gur, D., Sumkin, J. H., Rockette, H. E., Ganott, M., Hakim, C., Hardesty, L., Poller, W. R., Shah, R., and Wallace, L. (2004). Changes in breast cancer detection and mammography recall rates after the introduction of a computer-aided detection system. *Journal of the National Cancer Institute*, 96(3):185–190.

- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385. Disponible en: "<http://arxiv.org/abs/1512.03385>".
- Henriksen, E. L., Carlsen, J. F., Vejborg, I. M., Nielsen, M. B., and Lauridsen, C. A. (2019). The efficacy of using computer-aided detection (cad) for detection of breast cancer in mammography screening: a systematic review. *Acta Radiologica*, 60(1):13–18.
- Huynh, B. Q., Li, H., and Giger, M. L. (2016). Digital mammographic tumor classification using transfer learning from deep convolutional neural networks. *Journal of Medical Imaging*, 3(3):034501.
- INC (2018). Diccionario de cáncer. Disponible en: "<https://www.cancer.gov/espanol/publicaciones/diccionario/def/mama>".
- INEGI (2019). Instituto nacional de estadística y geografía. Disponible en: "<https://www.inegi.org.mx/default.html>".
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
- Jamieson, A. R., Drukker, K., and Giger, M. L. (2012). Breast image feature learning with adaptive deconvolutional networks. In *Medical Imaging 2012: Computer-Aided Diagnosis*, volume 8315, page 831506. International Society for Optics and Photonics.
- KERAS (2019). Keras: The python deep learning library. Disponible en: "<https://keras.io/>".
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kohli, A. and Jha, S. (2018). Why cad failed in mammography. *Journal of the American College of Radiology*, 15(3):535–537.

- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.
- Kumar, A., Kim, J., Lyndon, D., Fulham, M., and Feng, D. (2017). An ensemble of fine-tuned convolutional neural networks for medical image classification. *IEEE journal of biomedical and health informatics*, 21(1):31–40.
- Lahamange, S. and Patil, H. (2014). Breast cancer classification with statistical features of wavelet coefficient of mammograms. *arXiv preprint arXiv:1903.06342*.
- LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- López, V. R. (2012). Análisis de imágenes de mamografía para la detección de cáncer de mama. In *Temas de Ciencia y Tecnología*, volume 15, pages 39 – 45. Universidad de.
- Mahmoud, M. N., Rushdi, M. A., Ewais, I., Hosny, E., Gewefel, H., and Mahmoud, A. M. (2019). Computationally-efficient wavelet-based characterization of breast tumors using conventional b-mode ultrasound images. In *Medical Imaging 2019: Computer-Aided Diagnosis*, volume 10950, page 109502G. International Society for Optics and Photonics.
- Matich, D. J. (2001). Redes neuronales: Conceptos básicos y aplicaciones. *Universidad Tecnológica Nacional, México*.
- Max and Pechyonkin (2018). Key deep learning architectures: Lenet-5. Disponible en: "<https://medium.com/pechyonkin/key-deep-learning-architectures-lenet-5-6fc3c59e6f4>".
- Michael Heath, Kevin Bowyer, D. K. e. a. (2001). The digital database for screening mammography. Disponible en: "<http://www.eng.usf.edu/cvprg/Mammography/Database.html>".

- Ng, A. (2018). Machine learning yearning. Disponible en: "<https://www.mlyearning.org/>".
- OMS (2019). Cáncer. Disponible en: "<https://www.who.int/es/news-room/fact-sheets/detail/cancer>".
- Oshima, A., Shinohara, N., and Kamiya, N. (2019). Investigation of the effect of image resolution on automatic classification of mammary gland density in mammography images using deep learning. In *International Forum on Medical Imaging in Asia 2019*, volume 11050, page 1105018. International Society for Optics and Photonics.
- Ramteke, R. and Monali, Y. K. (2012). Automatic medical image classification and abnormality detection using k-nearest neighbour. *International Journal of Advanced Computer Research*, 2(4):190–196.
- Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. (2016). Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer.
- Rawat, W. and Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9):2352–2449.
- Rizwan, M. (2018). Lenet-5 - a classic cnn architecture. Disponible en: "<https://engmrk.com/lenet-5-a-classic-cnn-architecture/>".
- Roan and Gylberth (2018). An introduction to adagrad. Disponible en: "<https://medium.com/konvergen/an-introduction-to-adagrad-f130ae871827>".
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6).
- RStudio (2017). 3-way anova. Disponible en: "<https://www.r-bloggers.com/raccoon-ch-2-4-3-way-anova/>".

- Sampat, M. P., Markey, M. K., Bovik, A. C., et al. (2005). Computer-aided detection and diagnosis in mammography. *Handbook of image and video processing*, 2(1):1195–1217.
- Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Singh, S., Ho-Shon, K., Karimi, S., and Hamey, L. (2018). Modality classification and concept detection in medical images using deep transfer learning. In *2018 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pages 1–9. IEEE.
- Singla, S. (2017). A new kind of pooling layer for faster and sharper convergence. Disponible en: "<https://hackernoon.com/>".
- Soberanis, M. C. (2018). Inspiración biológica de las redes neuronales artificiales. Disponible en: "<https://medium.com/soldai/inspiraci%C3%B3n-biol%C3%B3gica-de-las-redes-neuronales-artificiales-9af7d7b906a>".
- Sprague, B. L., Arao, R. F., Miglioretti, D. L., Henderson, L. M., Buist, D. S., Onega, T., Rauscher, G. H., Lee, J. M., Tosteson, A. N., Kerlikowske, K., et al. (2017). National performance benchmarks for modern diagnostic digital mammography: update from the breast cancer surveillance consortium. *Radiology*, 283(1):59–69.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- Suckling, J. (1994). The mammographic image analysis society digital mammogram database. Disponible en: "<http://peipa.essex.ac.uk/info/mias.html>".
- Sun, W., Tseng, T.-L. B., Zheng, B., and Qian, W. (2016). A preliminary study on breast cancer risk analysis using deep neural network. In *International Workshop on Breast Imaging*, pages 385–391. Springer.

- SuperDataScience (2018). Convolutional neural networks (cnn): Step 3 - flattening. Disponible en: "<https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening>".
- Szegedy, C., Liu, W., and et al, J. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.
- TensorFlow (2019). About tensorflow. Disponible en: "<https://www.tensorflow.org/about>".
- Timothy, D. (2016). Incorporating nesterov momentum into adam. *OpenReview.net*.
- Tsang (2018). Review: Googlenet (inception v1)- winner of ilsvrc 2014 (image classification). Disponible en: "<https://medium.com/coinmonks/>".
- UNAM (2018). México, con déficit de personal para el manejo de mastógrafos. Disponible en: "<https://www.saludiarario.com/>".
- Vitaly, B. (2018). Understanding rmsprop - faster neural network learning. Disponible en: "<https://towardsdatascience.com/understanding-rmsprop-faster-neural-network-learning-62e116fcf29a>".
- Warren Burhenne, L. J., Wood, S. A., D’Orsi, C. J., Feig, S. A., Kopans, D. B., O’Shaughnessy, K. F., Sickles, E. A., Tabar, L., Vyborny, C. J., and Castellino, R. A. (2000). Potential contribution of computer-aided detection to the sensitivity of screening mammography. *Radiology*, 215(2):554–562.
- Warren McCulloch, W. P. (1986). A logical calculus of the ideas immanent in nervous activity. *Brain Theory*, page 229–230.
- Yang, W.-T., Su, T.-Y., Cheng, T.-C., He, Y.-F., and Fang, Y.-H. (2019). Deep learning for breast cancer classification with mammography. In *International Forum on Medical Imaging in Asia 2019*, volume 11050, page 1105014. International Society for Optics and Photonics.

- Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.
- Yu, D., Wang, H., Chen, P., and Wei, Z. (2014). Mixed pooling for convolutional neural networks. In *International Conference on Rough Sets and Knowledge Technology*, pages 364–375. Springer.
- Zeiler, M. D. (2012). Adadelata: An adaptive learning rate method.
- Zhang, L. and Zhang, B. (1999). A geometrical representation of mcculloch-pitts neural model and its applications. *IEEE Transactions on Neural Networks*, 10(4):925–929.
- Zhou, H., Zaninovich, Y., and Gregory, C. (2017). Mammogram classification using convolutional neural networks. *Vaialable: http://ehnree.github.io/documents/papers/mammogram_conv_net.pdf, 2.*

RESUMEN AUTOBIOGRÁFICO

Ing. Mayra Cristina Berrones Reyes

Candidato para obtener el grado de
MAESTRÍA en Ciencias
en Ingeniería de Sistemas

Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Tesis:

CLASIFICACIÓN DE MAMOGRAFÍAS MEDIANTE REDES NEURONALES
CONVOLUCIONALES

Nombre: Mayra Cristina Berrones Reyes.

Fecha de Nacimiento: 21 de Octubre de 1994

Edad: 24 años.

Lugar de Nacimiento: Cd. Victoria, Tamaulipas, México.

Padre: Miguel Angel Berrones Jasso.

Madre: Martha Cecilia Reyes Ornelas

Educación superior:

Centro de Bachillerato Tecnológico Industrial y de Servicios Num. 119.

Técnico en Informática.

(2009 - 2012)

Universidad Autónoma de Nuevo León. Facultad de Ingeniería Mecánica y Eléctrica.

Ingeniero en Tecnología de Software.

(2012 - 2017)