



**MODELO DE PREDICCIÓN DE DETECCIÓN DE MELANOMAS A PARTIR DE
CARACTERÍSTICAS EN IMÁGENES DE LESIONES CUTÁNEAS**

ABRAHAM DAVID ALTUVE MOUTHON

ELÍAS ENRIQUE COLÓN MUÑOZ

**UNIVERSIDAD DEL SINÚ ELÍAS BECHARA ZAINÚM
ESCUELA DE INGENIERÍA DE SISTEMAS
CARTAGENA-COLOMBIA
DICIEMBRE 2021**



**MODELO DE PREDICCIÓN DE DETECCIÓN DE MELANOMAS A PARTIR DE
CARACTERÍSTICAS EN IMÁGENES DE LESIONES CUTÁNEAS**

ESTUDIANTES:

Abraham David Altuve Mouthon

Elías Enrique Colón Muñoz

**Director trabajo de grado:
Luis Fernando Murillo**

**UNIVERSIDAD DEL SINÚ ELÍAS BECHARA ZAINÚM ESCUELA DE INGENIERÍA DE
SISTEMA
CARTAGENA-COLOMBIA
DICIEMBRE 2021**

AGRADECIMIENTOS

Expresamos nuestro agradecimiento a nuestro director de tesis Ingeniero Luis Fernando Murillo Murillo por su paciencia, consejos y guiarnos en el desarrollo de esta investigación. También queremos agradecer a la Ingeniera Eugenia Arrieta Rodríguez que con sus conocimientos y motivaciones nos ayudó a llevar a cabo el proyecto, hizo que sea posible que tuviésemos la experiencia de presentar esta investigación y difundir el conocimiento, es una profesora que nos aportó tanto a nivel académico como personal fortaleciendo nuestras competencias y cumplir con cada objetivo planteado en este proyecto, muchas gracias por todo.

Por último y no menos importante agradecer a la universidad por todas las herramientas que aportó a nivel tecnológico para capacitaciones, a amigos y familiares que estuvieron pendiente de todo el proceso que conllevo realizar este proyecto, gracias a todos.

Contenido

RESUMEN	6
INTRODUCCIÓN	7
1. DISEÑO METODOLÓGICO	8
1.1. Descripción del problema	8
1.2. Justificación	9
1.3. Formulación del problema	10
1.4. Alcance.....	10
1.5. Objetivos	10
1.5.1. General.....	10
1.5.2. Específicos	10
1.6. Estado del arte	11
1.7. Marcos de referencia	12
1.7.1. Marco teórico.....	12
1.7.2. Marco conceptual	24
1.7.3. Marco legal y consideraciones éticas.....	25
1.8. Metodología.....	25
2. DESARROLLO	28
2.1. Construcción del conjunto de datos.....	28
2.2. Construcción del modelo	29
2.2.1. Caracterización de las imágenes	29
2.2.2. Creación del modelo	34
2.2.3. Entrenamiento	35
3. VALIDACIÓN.....	38
CONCLUSIONES Y RECOMENDACIONES	46
BIBLIOGRAFÍA.....	47

TABLA DE ILUSTRACIONES

Figura 1. Disposición de las células de la piel. Fuente [2]	8
Figura 2. Distribución de casos nuevos de cáncer de piel por diagnóstico histológico, Instituto Nacional de Cancerología, 1996-2010 Fuente [3]	9
Figura 3. Incidencia estimada del melanoma por ente geográfico en Colombia, 2002-2006 Fuente: [3]	9
Figura 4. Factores de riesgo para melanoma Fuente [8]	13
Figura 5. Técnicas de biopsia en lesiones sospechosas de melanoma Fuente [8]	14
Figura 6. Criterios ABCDE para el diagnóstico de melanoma Fuente: [9]	15
Figura 7. Significado de la T en la clasificación del melanoma. Fuente [10]	15
Figura 8. Significado de la N en la clasificación del melanoma. Fuente [10]	16
Figura 9. Significado de la M en la clasificación del melanoma. Fuente [10]	16
Figura 10. Clasificación de los estadios del melanoma. Fuente [10]	17
Figura 11: Clasificación en base a expresiones faciales empleando un modelo de Deep Learning	18
Función 12: de activación lineal	18
Figura 13: Función de activación Sigmoide (No lineal)	19
Figura 14: Función de activación Tangencial Hiperbólica (No lineal)	19
Figura 15: Función de activación ReLU (No lineal)	20
Figura 16: Resultado de convolución con un kernel de 3x3.	21
Figura 17: Operación entre imágenes. fuente [30]	23
Figura 18: Convolución entre dos imágenes	23
Figura 19: Histograma en escala de colores grises	29
Figura 20: Histograma en escala de colores RGB.	30
Figura 21: melanoma segmentado en color blanco.	30
Figura 22: melanoma con etiquetas en escala a color y con valores binarios	31
Figura 23: melanoma con etiquetas de mayores pixeles sin el resto.	31
Figura 24: melanoma relleno de píxeles.	32
Figura 25: imagen convexa de melanoma.	32
Figura 26: Melanomas con ConvexHull y BoundingBox	33
Figura 27: Melanoma con Kernel (3x3).	33
Figura: 28: Funcionamiento de una neurona artificial [15]	34
Figura 29: Proceso del funcionamiento de una red neuronal convolucional [15]	35
Figura 30 Gráfica de aprendizaje con 2 épocas. Fuente propia.	42
Figura 31 Gráfica de aprendizaje con 4 épocas. Fuente propia.	42
Figura 32 Gráfica de aprendizaje con 6 épocas. Fuente propia.	43
Figura 33 Gráfica de aprendizaje con 8 épocas. Fuente propia.	43
Figura 34 Gráfica de aprendizaje con 10 épocas. Fuente propia.	43
Figura 35 Gráfica de aprendizaje con 12 épocas. Fuente propia.	44
Figura 36 Gráfica de aprendizaje con 14 épocas. Fuente propia.	44
Figura 37 Gráfica de aprendizaje con 16 épocas. Fuente propia.	44
Figura 38 Gráfica de aprendizaje con 18 épocas. Fuente propia.	44
Figura 39 Gráfica de aprendizaje con 20 épocas. Fuente propia.	44
Figura 40: Gráfica de tasa de aprendizaje del modelo. Fuente propia	45

RESUMEN

En este documento se presenta un proyecto de investigación en el área de Inteligencia Artificial, en el cual se desarrolla un modelo de Deep Learning para el aprendizaje automático del mismo, la función de este es la detección de melanoma por medio del análisis de imágenes.

La problemática para tratar en este trabajo es la detección de melanoma, el cual es uno de los distintos cánceres de piel que existen, el cual cuenta con una gran tasa de mortalidad a pesar de su bajo índice.

Este trabajo se encuentra organizado en 3 capítulos, iniciando con el capítulo de diseño metodológico, en el cual se encuentra definido el problema, la justificación, el alcance y la formulación del problema, además, se definen los objetivos tanto general como específicos, también se encuentra definido el marco de referencia, formado por marco teórico, marco conceptual y marco legal y por último las consideraciones éticas dando a conocer las leyes que rigen esta investigación y se explican los conceptos básicos utilizados en esta investigación.

En el segundo capítulo se encuentra la construcción del conjunto de datos en el cual se explican los datos que se utilizarán dentro del modelo y, además, en el cual se explica el desarrollo del modelo utilizando técnicas de Deep Learning, en él también se encuentra el proceso realizado para el entrenamiento del modelo.

Por último, se encuentran los resultados obtenidos por parte del modelo.

INTRODUCCIÓN

El proceso natural de la vida de las células en el cuerpo humano es que estas crezcan, se dividan para crear nuevas células y mueran cuando cumplan su función, sin embargo, cuando una persona presenta algún tipo de cáncer este proceso sufre de alteraciones teniendo un descontrol en la cantidad de células que se encuentran en el cuerpo, sobrepasando la cantidad de células anormales a la cantidad de células normales. Durante esta alteración las células que deberían morir siguen existiendo a pesar de que ya no son necesarias y de que han cumplido con su función y, además, se generan nuevas células a pesar de que éstas no sean necesarias. La acumulación de estas células genera tumores cancerosos en la zona afectada del cuerpo, esto se puede presentar en cualquier zona del cuerpo. Estos tumores al ser malignos pueden infectar a otros tejidos utilizando como medio el sistema circulatorio o el sistema linfático lo que le permite generar nuevos tumores lejos en otros órganos, a este proceso se le conoce como metástasis, principal característica del cáncer y una de las mayores causas de muerte de esta enfermedad.

«El diagnóstico del cáncer en una fase tardía y la imposibilidad de recibir tratamiento condenan a muchas personas a sufrimientos innecesarios y a una muerte precoz», afirma el Dr. Etienne Krug, director del Departamento de la OMS de Enfermedades No Transmisibles, Discapacidad, Violencia y Prevención de Lesiones. [1]

Entre los diferentes tipos de cáncer que existen se encuentra el cáncer de piel, el cual es el más común de todos. A su vez existen 4 tipos principales de cáncer de piel: carcinoma de células escamosas, carcinoma de células basales y melanoma. Esta investigación estará centrada en este último debido a la mortalidad que causa.

Dentro de los de los tipos de cáncer de piel el melanoma es uno de los menos comunes, pero esto no quita no le quita el hecho de que es el que mayor tasa de mortalidad tiene dentro del cáncer de piel debido a que cuando los melanocitos empiezan a presentar anomalías, es decir, multiplicarse de forma descontrolada, es más probable que estas células infectan tejidos cercanos y se expanden a través del cuerpo. A pesar de que el melanoma se presenta mayormente en la piel (melanoma cutáneo) también hay casos en los que aparece en otros tejidos tales como los ojos (melanoma intraocular o melanoma ocular) o la garganta.

Teniendo esto en cuenta se busca desarrollar un algoritmo que permita identificar y extraer las características más importantes y de mayor riesgo a la hora de la detección de un melanoma a partir de imágenes tomadas al paciente por parte de un especialista, con el objetivo de que a futuro este algoritmo se puede usar para desarrollar un modelo predictivo que estas características las analice y permita saber la posibilidad que tiene el paciente de padecer esta enfermedad.

1. DISEÑO METODOLÓGICO

1.1. Descripción del problema

Hoy en día el cáncer es el responsable de cerca de 1/6 parte de la mortalidad mundial. En los últimos años se ha diagnosticado a más de 14 millones de personas por año con alguno de los diferentes tipos de cáncer que existen, pero se prevé según la Organización Mundial de la Salud (OMS) que esta cifra ascendería a 21 millones por año para el 2030 [29].

En contraparte la OMS establece que la cifra de personas que mueren anualmente por cáncer asciende a 8,8 millones de personas alrededor del mundo, una de las principales razones de esto sería el no detectar las irregularidades en las células en las etapas tempranas del cáncer, etapas que son mucho más fáciles de tratar, con mejores resultados y sumado a esto el impacto financiero y físico que le presentan al paciente son mucho menores.

A pesar de que la edad promedio de las personas que han contraído melanoma es de 45 años en las últimas décadas ha aumentado el número de casos presentados en la infancia o adolescencia [29].

La piel está dividida en varias capas, pero las más importantes son la dermis que sería su capa profunda y la epidermis que sería su capa superficial, es en esta última donde se presenta el cáncer de piel. Esta está formada por 3 diferentes tipos de células:

- “Células escamosas”: células delgadas y planas que forman la capa superficial de la epidermis.
- Células basales: células redondas que están por debajo de las células escamosas.
- Melanocitos: células que producen melanina y se encuentran en la parte profunda de la epidermis. La melanina es el pigmento que determina el color natural de la piel. Cuando la piel se expone al sol o a luz artificial, los melanocitos producen más pigmento y oscurecen la piel.”

En la siguiente imagen se puede observar la disposición de estas células

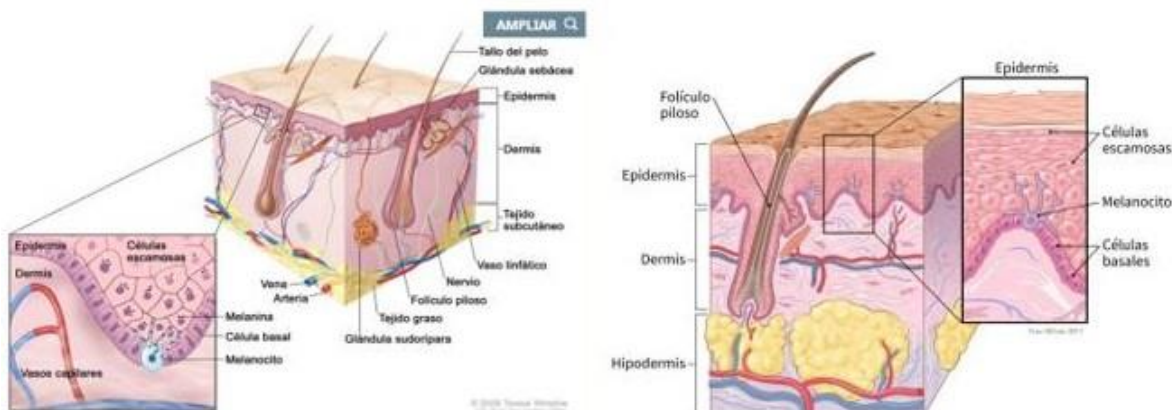


Figura 1. Disposición de las células de la piel. Fuente [2].

El Instituto Nacional de Cancerología ha registrado desde 1996 hasta 2010 79.381 casos nuevos de cáncer, de los cuales 11.780 (14,9 %) fueron casos nuevos de cáncer de piel, con un promedio de 841,4 por año.

DISTRIBUCIÓN DE CASOS NUEVOS DE CÁNCER DE PIEL POR DIAGNÓSTICO HISTOLÓGICO, INSTITUTO NACIONAL DE CANCEROLOGÍA, 1996-2010																
SUBTIPO HISTOLÓGICO	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	Total
Basocelular	395	384	335	221	235	221	313	411	460	523	542	500	506	540	785	6.371
Escamocelular	145	151	160	116	137	139	164	183	210	219	213	176	174	172	249	2.608
Melanoma	114	100	98	100	84	97	127	162	148	175	124	118	120	143	143	1.853
Linfoma	12	6	17	10	16	18	21	30	38	25	22	20	22	32	38	327
Sarcoma de Kaposí	10	13	11	4	6	8	15	13	14	38	12	22	13	9	13	201
DFSP*	4	5	2	2	3	4	7	9	6	6	1	10	9	5	9	82
Células de Merkel	0	2	2	2	3	0	0	3	3	2	1	5	3	1	5	32
Tumores de los anexos	1	2	1	4	2	2	11	4	14	7	17	4	7	12	5	93
Otras neoplasias	36	39	9	10	8	12	7	13	13	16	24	16	26	40	26	295
Total	717	702	635	469	494	501	665	828	906	1.011	956	871	880	954	1.273	11.862

* DFSP: dermatofibrosarcoma protuberans

Figura 2. Distribución de casos nuevos de cáncer de piel por diagnóstico histológico, Instituto Nacional de Cancerología, 1996-2010 Fuente [3]

En la siguiente tabla se puede ver la incidencia de melanoma en las principales regiones del país.

INCIDENCIA ESTIMADA DEL MELANOMA POR ENTE GEOGRÁFICO EN COLOMBIA, 2002-2006		
Localidad	Mujeres	Hombres
Bogotá	9.3	6.9
Antioquia	4.5	6.1
Valle del Cauca	3.9	5.0
Atlántico	0.6	0.1
Bolívar	0.4	1.2
Norte de Santander	1.1	5.9
Santander	5.3	2.6
Cundinamarca	8.0	4.7
Meta	9.3	5.2
Caldas	7.0	5.5

Figura 3. Incidencia estimada del melanoma por ente geográfico en Colombia, 2002-2006 Fuente: [3]

1.2. Justificación

La motivación principal que lleva a realizar este trabajo o investigación es que se ha detectado que el melanoma se encuentra entre las principales causas de mortalidad por cáncer alrededor del mundo por ende existe una necesidad de crear métodos que faciliten su detección temprana lo que generaría tratamientos más eficientes, eficaces y económicos. Labor realizada en conjunto entre las áreas de Ciencias de la computación y la salud. Siendo que durante el desarrollo de este trabajo se tocará el manejo de imágenes con posibles patrones de melanoma.

Actualmente existen diferentes formas de diagnosticar si un paciente posee o no de melanoma siendo los más utilizados revisar los antecedentes de la persona, realizar un examen físico, realizar un examen de la piel y por último una biopsia. Durante el examen físico se revisa el estado general de salud de la persona para identificar posibles masas o anomalías. En el examen de piel un especialista de la salud revisa la superficie de la piel de la persona en busca de cambios o alteraciones, en su color tamaño textura o forma, de los lunares o marcas de nacimiento del paciente. Por último, tendríamos la biopsia que sería el examen más invasivo dentro del cuerpo del paciente debido a que se le debe

extirpar parte del tejido que se consideré anormal y parte del tejido sano para que posteriormente uno o varios patólogos lo examinen. A partir de estos exámenes se puede determinar el estadio en el que se encuentra el melanoma lo cual es de mucha ayuda para elegir el tratamiento que mejor se adecua a la etapa en la que se encuentra el melanoma. Teniendo esto en cuenta, el algoritmo que se plantea realizar sería de gran ayuda a la hora de realizar los exámenes de piel mencionados anteriormente ya que ayudaría a identificar de una forma más ágil y rápida las anomalías que se presenten en la piel, identificando sus características principales como son el tamaño del lunar, la pigmentación del lunar, la forma del lunar la tez de la persona y si en este lunar hay indicios de ulceración. Al contar con la identificación de dichas características se puede plantear a futuro la creación de un modelo predictivo que procese y analice estos datos con el objetivo de estimar la probabilidad de que un paciente posea o no un melanoma, permitiendo así dar un diagnóstico y tratamiento oportuno.

1.3. Formulación del problema

¿Cómo desarrollar un modelo de predicción para detección de melanomas a partir de características en imágenes de lesiones cutáneas en Colombia utilizando técnicas de *deep learning*?

1.4. Alcance

Este proyecto tuvo como objetivo el desarrollo de un algoritmo que permita la identificación y extracción de características a partir de imágenes de posibles casos de melanoma, para ello se eligió un conjunto de imágenes que cumplan con las características solicitadas para poder llevar a cabo el entrenamiento del modelo realizado. Luego se eligió un conjunto diferente de imágenes con el cual se probó el modelo para comprobar el correcto funcionamiento de este.

Este proyecto devuelve únicamente como resultado las características importantes al momento de detectar el melanoma, sin llegar a realizar un análisis de estos resultados, como podría ser estimar la probabilidad de que el paciente posea melanoma, estimar el grado en el que se encuentra la enfermedad.

1.5. Objetivos

1.5.1. General

Desarrollar un modelo de predicción para detección de melanomas a partir de características en imágenes de lesiones cutáneas en Colombia utilizando técnicas de *deep learning*.

1.5.2. Específicos

- Construir un conjunto de datos mediante la recolección de imágenes en repositorios públicos para el entrenamiento y prueba del modelo.
- Procesar las imágenes mediante la librería *OpenCv*.
- Seleccionar la arquitectura del modelo de *Deep Learning* para la construcción del modelo de predicción.
- Validar la precisión del modelo a partir de un conjunto de imágenes no usadas en el

entrenamiento del modelo, que permita evaluar la predicción en la detección de melanomas

1.6. Estado del arte

Actualmente las investigaciones realizadas sobre la detección de cáncer de piel de melanoma mediante IA (inteligencia artificial) han mostrado avances significativos puesto que las técnicas utilizadas han dado resultados prometedores, la detección temprana juega un papel clave en el diagnóstico del cáncer y puede mejorar las tasas de supervivencia a largo plazo esto se puede validar de manera más eficiente en los resultados de una búsqueda realizada de forma cronológica en los últimos años.

Una de las investigaciones más destacadas es *Deep Learning for Image-based Cancer Detection and Diagnosis — A Survey* en el año 2017 por Alexandra Elbakyan la cual plantea que la formación de imágenes es una técnica muy importante para la detección y el diagnóstico precoz del cáncer. Como es bien sabido, la imagenología se ha utilizado ampliamente para la detección temprana del cáncer, la monitorización y el seguimiento después de los tratamientos. Sin embargo, la interpretación manual de una enorme cantidad de imágenes médicas puede resultar tediosa y llevar mucho tiempo y causa fácilmente prejuicios y errores humanos. Por lo tanto, desde principios de la década de 1980, se han implementado los sistemas de diagnóstico asistido por computadora (CAD)” [4].

En el mismo orden de ideas surgió una investigación más reciente por parte de Alexandra Elbakyan para el año 2019 en la que expone un claro avance sobre la detección del melanoma, afirma que recientemente la utilización del procesamiento de imágenes y la visión artificial en aplicaciones médicas está aumentando basado en eso propusieron un método basado en el procesamiento para la detección temprana del cáncer de piel. El método utiliza una óptima Red neuronal convolucional (RNC) para este propósito. En este artículo, el algoritmo de optimización de ballenas mejorado es utilizado para optimizar la RNC. Los resultados de la simulación muestran que el método propuesto tiene superioridad sobre otros métodos comparados [5].

De las investigaciones más recientes realizado por Instituto de Investigaciones Biomédicas Maimónides de Córdoba quien en el año 2020 presentó la investigación titulada Redes convolucionales plantean que el estudio experimental comprendió cinco fases, donde primero se analizó la sensibilidad de los modelos con respecto al algoritmo de optimización utilizado para su entrenamiento, y luego se analizó el impacto en el rendimiento al utilizar diferentes técnicas como el aprendizaje sensible al costo, el aumento y la transferencia de datos. El estudio realizado confirmó la utilidad, eficacia y solidez de diferentes arquitecturas convolucionales para resolver el problema del diagnóstico de melanoma [6].

Una investigación la cual tuvo mucho protagonismo fue la del Instituto de Biociencia Molecular, Universidad de Queensland hecha en el año 2020 que se centraron en aplicar métodos de aprendizaje profundo interpretables simultáneamente a los cánceres de piel más comunes (carcinoma de células basales, carcinoma de células escamosas y carcinoma intraepidérmico) en un entorno histológico. Teniendo muy grandes resultados tales como demostrar que la mayoría del trabajo de dermatopatología es susceptible de análisis automático por máquina. Al caracterizar el tejido clasificándose en 12 clases dermatológicas significativas. Esto proporcionó resultados altamente interpretables ya que la red está entrenada para representar el dominio del problema con una alta precisión de la imagen (93,6-97,9%) [7].

1.7. Marcos de referencia

1.7.1. Marco teórico

Los algoritmos han existido a lo largo de la historia del ser humano debido a que estos son una serie de pasos ordenados que se deben seguir para poder conseguir un resultado, que se emplean desde las actividades más básicas en la vida cotidiana hasta las más complejas. Con el surgimiento de las tecnologías y los avances en la ciencia estos algoritmos se llevaron a lenguaje de programación para permitir la comunicación entre el humano y la máquina. En sus inicios estos eran muy básicos siendo que se usaban cintas para transmitirle las instrucciones al computador, pero con el paso del tiempo estos fueron evolucionando hasta convertirse en los lenguajes de programación que conocemos hoy en día que poseen una sintaxis mucho más fácil de interpretar y de manejar, pareciéndose en algunos casos a un lenguaje que se utilizará en la vida cotidiana. Existe una amplia variedad entre estos lenguajes de programación, contando cada uno con beneficios y cualidades que le permiten ser más eficientes y eficaces para ciertas tareas, cómo podrían ser aplicaciones de escritorio, aplicaciones móviles, aplicaciones web, videojuegos, inteligencia artificial, etcétera. Durante el mediados del siglo pasado hubo un auge de una de una de estas tecnologías debido a la publicación de un artículo por parte de un matemático británico llamado Alan Turing, conocido hoy en día como el padre de la informática, en el que planteaba la pregunta de qué tanto podría pensar una máquina por sí misma, esta tecnología es la inteligencia artificial. Hoy en día existe el desarrollo de muchos modelos de inteligencia artificial realizados por parte de grandes empresas como por ejemplo Google o Microsoft. Uno de los lenguajes más utilizados actualmente para el desarrollo de Inteligencia Artificial y Machine Learning es Python debido a las ventajas que éste ofrece. Este lenguaje fue desarrollado por el informático holandés Guido Van Rossum, se desarrolló durante la década de los 90 y continuó durante la década del 2000, donde alcanzó su auge con la salida de su tercera versión en el 2008. Para potenciar las capacidades de los lenguajes de programación existen librerías que sirven para utilizar funcionalidades que ya han sido desarrolladas por otros programadores y emplearlas en el nuevo código que se esté realizando, brindando así una mayor rapidez en el desarrollo y un sistema mucho más robusto.

Para una mejor comprensión de los temas principales de este trabajo, a continuación, se explicarán los conceptos de melanoma y redes neuronales convolucionales.

Melanoma

El melanoma es un tipo de cáncer de piel, formado por un tumor maligno que se origina en los melanocitos, los cuales son las células de la piel encargadas de la pigmentación de esta. A pesar de que representa un bajo porcentaje (4%) dentro de los distintos cánceres de piel, se le atribuye un gran porcentaje de decesos dentro de estos, alcanzando aproximadamente el 80% [29].

Es uno de los principales ejemplos de que la detección temprana es un factor fundamental para la supervivencia del paciente, ya que los pacientes que tienen se le detecta en uno de los primeros estadios tienen una tasa de supervivencia cercana al 100% mientras que aquellos que se les se les detecta en un estadio avanzado tienen una probabilidad de supervivencia menor, de alrededor del 10%.

Existen diferentes factores de riesgo para él la aparición de melanoma siendo el principal la existencia de un melanoma previo, ya que sí se ha presentado con anterioridad existe

un gran riesgo de volver a contraerlo. Seguido a este, el riesgo aumenta si existen antecedentes familiares con personas con melanoma.

Las personas con una gran cantidad de lunares o nevos atípicos según su nombre médico tienen una mayor probabilidad de contraer melanoma, además, también influye el tamaño de estos.

La exposición a la radiación ultravioleta provocada por el sol es el principal factor ambiental para la causa de melanoma, llegando incluso a causar quemaduras solares.

Si bien existen factores individuales como pueden ser el tono de la piel o el color del pelo de los ojos que pueden influir en la obtención de melanoma debido a los bajos índices de melanina en el cuerpo.

En la siguiente tabla se puede ver reflejados los factores de riesgo antes mencionados junto a su riesgo estimado.

Factor de riesgo	Riesgo relativo estimado	
Xeroderma pigmentoso	1.000	
Nevus displásicos más melanoma previo más historia familiar	500	
Nevus displásicos, sin melanoma previo más historia familiar	148	
Historia familiar positiva (más de 3 parientes afectados)	35-70	
Nevus displásicos	11-27	
Múltiples nevos benignos (más de 100)	11	
Nevus melanocíticos congénitos (grandes)	17 -21	
Historia personal de melanoma	8,5 -9	
Inmunosupresión	Trasplantes	3
	VIH	1,5
Fototipo 1 de piel	1,7	
Ojos azules	1,6	
Pelirrojos	2,4	
Historia de quemaduras intensas	2,5	

Figura 4. Factores de riesgo para melanoma Fuente [8]

El melanoma posee varios tipos que son:

- De Extensión superficial
- Nodular
- Lentiginoso de las extremidades

Para detectar un posible melanoma lo primero que se realiza es un examen físico alrededor de la lesión en busca del tumor y lesiones que indiquen metástasis. Para tener un mejor análisis se suele utilizar lupas o lentes de aumento con el fin de poder detallar mejor la zona afectada. Para llevar un control sobre el avance del melanoma se suelen realizar fotografías periódicas sobre dicha área. Otra técnica para el análisis de melanoma muy utilizada, principalmente en Europa, es mediante ultrasonidos en el cual los melanomas se pueden observar con una mayor claridad. La dermatoscopia es una técnica no invasiva y muy utilizada en la actualidad, que utiliza un instrumento llamado dermatoscopia el cual está equipado con una luz polarizada y un juego de lentes con aumentos, sirve para

determinar el lugar óptimo para la realización de una biopsia ya que permite la visualización in vivo de las estructuras de la dermis y epidermis que no son visibles a simple vista. Como ultima paso para su diagnóstico esta la biopsia, que consiste en extraer una muestra de la zona afectada para realizarle un estudio e identificar cómo tratarlo.

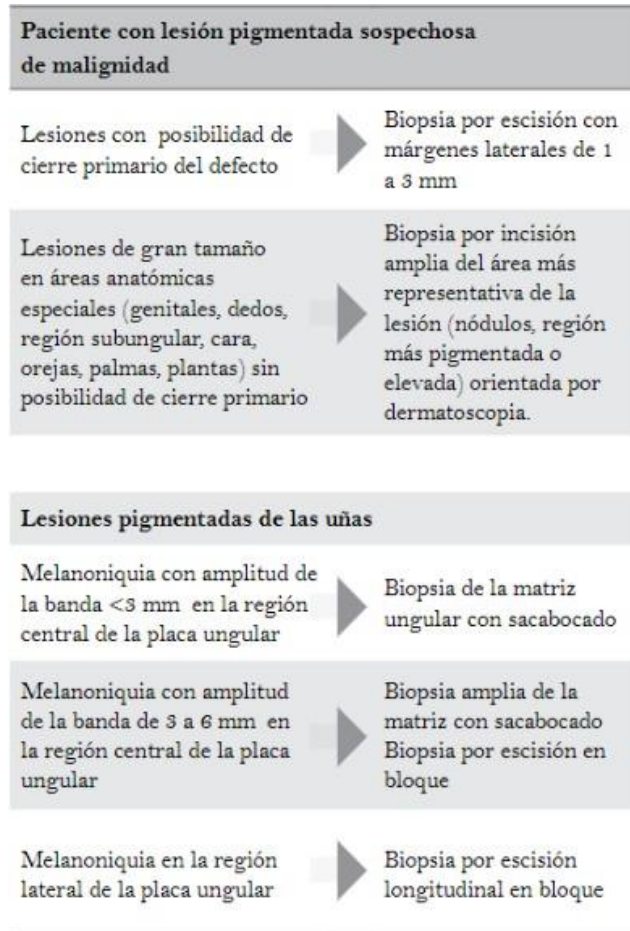
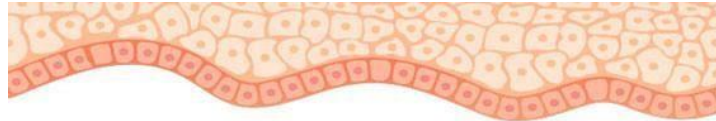


Figura 5. Técnicas de biopsia en lesiones sospechosas de melanoma
Fuente [8]

Desde 1985, para el diagnóstico clínico se ha implementado el sistema ABCDE del melanoma (ver figura 6):

- A: Asimetría. La lesión NO es redondeada.
- B: Bordes. Los bordes son irregulares.
- C: Color. La lesión presenta distintos colores, no homogéneos.
- D: Diámetro. El tamaño de la lesión es mayor de 6 mm.
- E: Evolución. Cualquiera de las características anteriores ha presentado cambios en el tiempo.



SIGNOS DEL MELANOMA

El ABCDE del melanoma te puede ayudar a detectarlo a tiempo:
En los melanomas las características que presentan las lesiones con mayor frecuencia vienen determinadas por la regla del A, B, C, D.



Figura 6. Criterios ABCDE para el diagnóstico de melanoma
Fuente: [9]

Cuando la detección del melanoma ha dado positivo se procede a su estadificación, siendo los estadios I, II, III y IV los posibles. Para determinar en qué estadio se encuentra se rigen para dos o 3 factores que se denominan **TNM**:

- T hace referencia a la extensión local del propio melanoma.

CLASIFICACIÓN T	SIGNIFICADO
TX	No se ha podido determinar el grosor
T0	No se ha podido determinar que haya melanoma o el melanoma ha regresado espontáneamente
Tis	Melanoma in situ, está confinado en la primera capa de la piel sin atravesarla
T1	El melanoma mide 1 o menos milímetros. Se divide en a y b
T1a	El melanoma mide menos de 0.8 milímetros y no tiene ulceración.
EN ESTOS CASOS DE ARRIBA ES DONDE HABITUALMENTE NO SE REALIZA GANGLIO CENTINELA EN LOS SIGUIENTES CASOS ES DONDE SE REALIZA HABITUALMENTE GANGLIO CENTINELA	
T1b	El melanoma mide menos de 0.8 milímetros y tiene ulceración. O bien el melanoma mide más de 0.8 a 1 milímetros sin ulceración
T2	El melanoma mide entre más de 1 y 2 milímetros (inclusive) Si no presenta ulceración, es T2a Si la presenta, es T2b
T3	El melanoma mide entre más de 2 y 4 milímetros (inclusive) Si no presenta ulceración, es T3a Si la presenta, es T3b
T4	El melanoma mide más 4 milímetros Si no presenta ulceración, es T4a Si la presenta, es T4b

Figura 7. Significado de la T en la clasificación del melanoma.
Fuente [10]

- N hace referencia a los ganglios.

CLASIFICACIÓN N	SIGNIFICADO
NX	No se ha podido determinar el estado de los ganglios
N0	No hay afectación del melanoma en los ganglios
N1	Hay solo un ganglio afectado. Este puede ser N1a: se ha detectado por el ganglio centinela N1b: se ha detectado por la exploración física
N1c	No hay ganglios afectados, pero se ha detectado metástasis en tránsito, satélites o microsatélites (ver*)
N2	Hay 2 o 3 ganglios afectados. Puede ser N2a: se han detectado por el ganglio centinela N2b: se ha detectado por la exploración física
N2c	Hay un ganglio afectado (independientemente de cómo se detecte) y además hay metástasis en tránsito, satélites o microsatélites (ver*)
N3	Este subgrupo es más complejo, y comprende N3a: hay cuatro ganglios afectados detectados por el ganglio centinela N3b: hay cuatro ganglios afectados detectados por la exploración física, o hay cualquier número de ganglios en forma de conglomerado.
N3c	N3c: cualquier número superior a 2 ganglios, independientemente de cómo se detecten, o cualquier número de ganglios si estos están en forma de conglomerado, en los que se añaden metástasis en tránsito, satélites o microsatélites (ver*)

*Las metástasis en tránsito, satélites o microsatélites, son una forma específica de metástasis del melanoma, que suceden cuando afectan a los vasos linfáticos de la piel antes de llegar al ganglio en el que se detendrían primero.

Quando están presentes, le aportan la letra **c** a la **N**.

Figura 8. Significado de la N en la clasificación del melanoma.

Fuente [10]

- M hace referencia a las metástasis en otros órganos.

CLASIFICACIÓN M	SIGNIFICADO
M0	Se han podido descartar con pruebas de imagen y exploración física la presencia de metástasis
M1a	Las metástasis están en la piel, en el tejido graso, en los músculos o en ganglios que están lejos del territorio ganglionar del melanoma. - M1a0 si LDH normal - M1a1 si LDH elevada
M1b	La o las metástasis está en el pulmón (puede haber también metástasis del tipo M1a, pero no del tipo M1c) - M1b0 si LDH normal - M1b1 si LDH elevada
M1c	La o las metástasis está en cualquier órgano distinto a las que hay en M1a y/o distinto al pulmón (por ejemplo hígado, bazo, páncreas, aparato digestivo, etc), con la excepción del cerebro - M1c0 si LDH normal - M1c1 si LDH elevada
M1D	La o las metástasis está en el cerebro, independientemente de que haya en sitios como M1a o M1b - M1d0 si LDH normal - M1d1 si LDH elevada

Figura 9. Significado de la M en la clasificación del melanoma.

Fuente [10]

Una vez conocido el estado de cada una de las letras se procede a clasificar los diferentes estadios.

T	N	M	ESTADIO
T1a	0	0	IA
T1b	0	0	
T2a	0	0	IB
T2b	0	0	
T3a	0	0	IIB
T3b	0	0	
T4a	0	0	
T4b	0	0	IIC
T0 hasta T2a	N1a o N2a	0	IIIA
T0 hasta T2a	N1b, N1c o N2b	0	IIIB
T0	N1b o N1c	0	IIIB
T1 hasta T2a	N1b, N1c o N2b	0	IIIB
T2b, T3a	N1a hasta N2b	0	IIC
T3b, T4a	Cualquier N mayor que 0	0	IIC
T0	N2b, N2c, N3b o N3c	0	IIC
T4b	N1a hasta N2b	0	IIID
T4b	N3 (a, b o c)	0	
Cualquier T	Cualquier N	1	IV

Figura 10. Clasificación de los estadios del melanoma.

Fuente [10]

Los tratamientos que existen hoy en día para el melanoma son:

- Inmunoterapia, es un tratamiento por vía intravenosa en la cual se administran anticuerpos que no atacan directamente al melanoma, sino que desbloquea a las células del sistema inmune para que combatan contra las células malignas
- Terapia Dirigida, son dos medicamentos que se toman diariamente por vía oral.
- Radioterapia, consiste en sesiones diarias de radiación de una duración de 15-30 minutos, dependiendo el caso.
- Ensayos Clínicos, son nuevas investigaciones que se realizan para dar un mejor tratamiento al melanoma.

Deep Learning

El aprendizaje profundo (*Deep Learning*) define un conjunto de técnicas de la Inteligencia Artificial (AI) que consiste en una técnica de aprendizaje automático que enseña a los ordenadores a hacer lo que resulta natural para las personas: aprender mediante ejemplos. Los modelos de *Deep Learning* se entrenan mediante un amplio conjunto de datos etiquetados y arquitecturas de redes neuronales o *Neural Networks* (NN) formadas por distintas capas, pudiendo obtener resultados con una precisión que incluso supere el rendimiento humano. Estas capas se organizan en una jerarquía de creciente complejidad y abstracción, de forma que el nivel inicial de la red aprende conceptos simples, el siguiente nivel toma esta información sencilla, la combina, compone una información algo un poco más compleja y se lo pasa al tercer nivel, y así sucesivamente (Figura 11).

Las redes neuronales están basadas en los sistemas nerviosos biológicos y, por lo tanto, están constituidas por un conjunto de unidades llamadas neuronas o nodos conectados unos con otros. Las neuronas reciben señales (inputs) de otras neuronas y en función de las señales recibidas, una neurona envía a su vez una señal a otras neuronas. Profundizaremos en las redes neuronales convolucionales o *Convolutional Neural Networks* (CNN), que son un tipo de redes neuronales profundas [11].

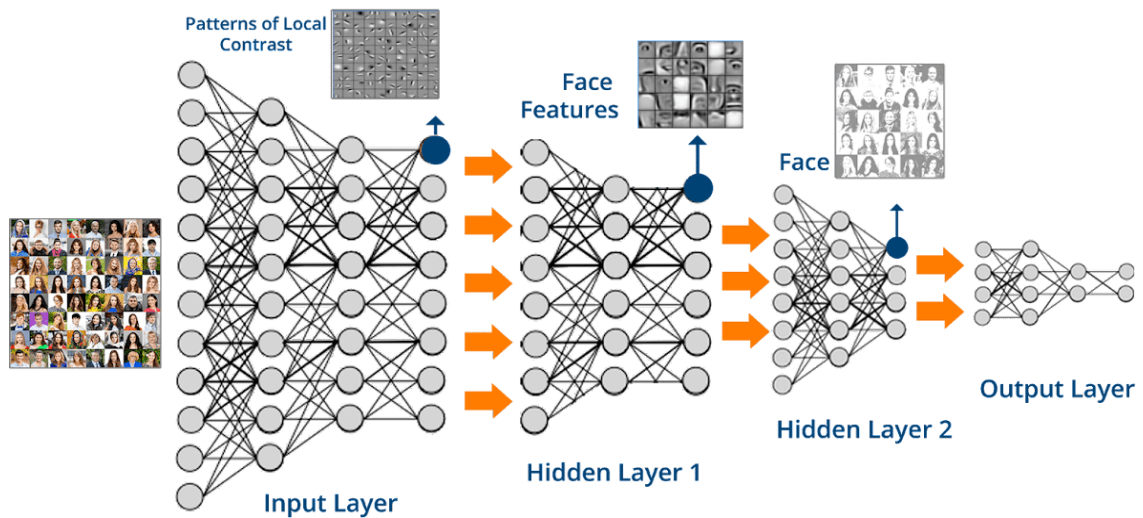


Figura 11: Clasificación en base a expresiones faciales empleando un modelo de *Deep Learning*. Muestra las imágenes de entrada, así como las diferentes características que se extraen en cada capa. Extraído de [11].

Funciones de activación

Las funciones de activación deberían ser no lineales para aumentar el rendimiento de la red. No obstante, existen funciones de activación lineales que no son frecuentemente utilizadas, tal cual indica la figura 11. En dicha figura observamos la función identidad, con lo cual tener esta función de activación es equivalente a no tener una función de activación, ya que el valor de la entrada se traduce en el mismo valor de la salida

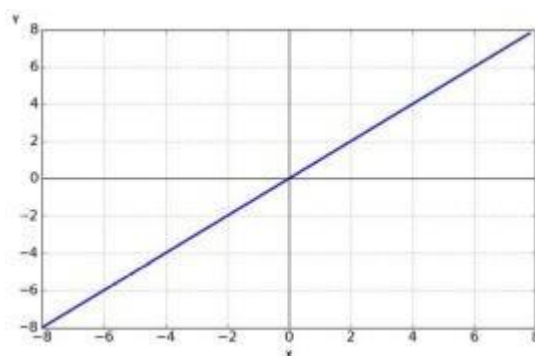


Figura 12: Función de activación lineal

Por otro lado, están las funciones no lineales donde resulta deseable que cumpla los siguientes requisitos:

- **Función monótona:** Esta propiedad garantiza que la superficie del error asociada a una sola capa del modelo sea convexa. Lo que permite una convergencia hacia el mínimo más fácilmente [12]
- **Derivable:** Esta propiedad se requiere para utilizar métodos de optimización como el del gradiente descendente [13]

Las funciones de activación no lineales más utilizadas son las siguientes:

Función Sigmoide o Logística

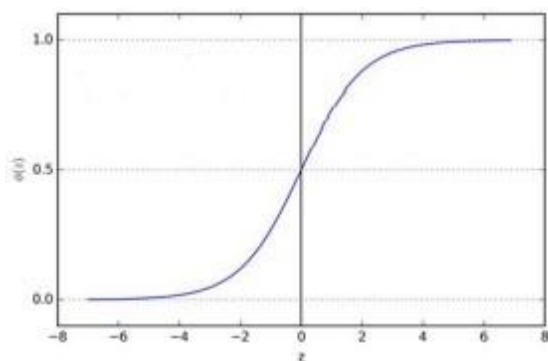


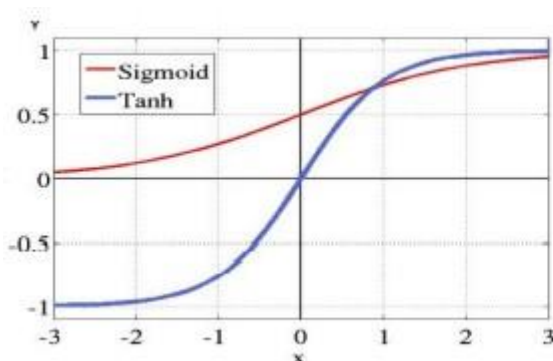
Figura 13: Función de activación Sigmoide (No lineal)

$$\sigma(z) = \frac{1}{1 + e^{-cz}}$$

Ecuación 1.1

La función Sigmoide (ver figura 13) dispone del parámetro C (ver ecuación 1.1) que permite controlar el crecimiento de la función. La función es monótona y diferenciable. En lo que respecta a su utilización en las capas ocultas, debido a la saturación que produce esta función cuando la entrada tiene un valor absoluto grande (ya sea positivo o negativo) hace que el entrenamiento basado en el gradiente descendente sea dificultoso. Ya que su rango se encuentra en (0,1), se suele utilizar en la capa de salida cuando se requiere calcular la probabilidad asociada a una clasificación, siempre y cuando se utilice una función de pérdida que elimine el efecto de saturación.

Función de activación Hiperbólica



$$y = \frac{e^{zx} - e^{-zx}}{e^{zx} + e^{-zx}}$$

Ecuación 1.2

Figura 14: Función de activación Tangencial Hiperbólica (No lineal).

Esta función se puede reescribir utilizando la función Sigmoide, con lo cual se encuentra íntimamente relacionada. A diferencia de la función Sigmoide, la función Tangente Hiperbólica (ver figura 14) tiene su rango en (-1,1) permitiendo mapear valores de entradas negativas a valores de salida de su mismo signo. Por otro lado, al igual que la función sigmoide dispone del parámetro Z (ver ecuación 1.2), el cual permite calcular la velocidad de crecimiento de la función. Generalmente donde funciona bien la función sigmoide, la función Hiperbólica funciona mejor [11].

Función ReLU (Unidad lineal rectificada)

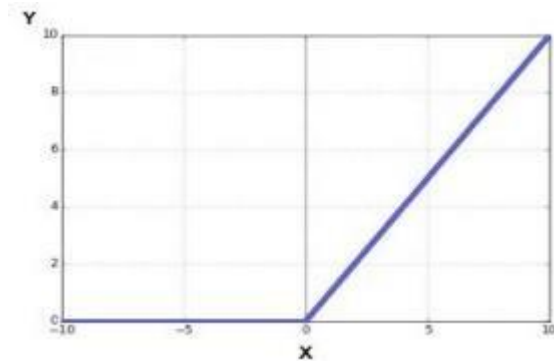


Figura 15: Función de activación ReLU (No lineal). Fuente [31]

La función ReLU (*Rectified Linear Unit* – ver figura 15) es la más empleada (en el momento que escribe el presente trabajo) ya que se utiliza en casi todos los tipos de redes como así también en otros tipos de redes de aprendizaje profundo. Es derivable y es monótona (su derivada también es monótona). Una desventaja de la función ReLU es que los pesos podrían actualizarse de cierta forma en la que la neurona nunca se active, es decir, que la salida de esa neurona siempre sea cero y cuando se llega a esa condición durante el entrenamiento no se puede salir de ella [14]. No obstante, tiene ventajas importantes comparadas con la función Sigmoide o Tangencial Hiperbólica, como la aceleración en la convergencia de la red (mediante el gradiente descendente estocástico) debido a que no implica cálculos costosos en el entrenamiento [11].

Inicialización de una red neuronal

Antes de entrenar una red neuronal, el primer paso a llevar a cabo es la inicialización de los parámetros, puesto que, si se realiza correctamente, la optimización se logrará en menos tiempo, ya que en el entrenamiento se tratará de adaptar estos parámetros (pesos y sesgos) de forma que la red se comporte de la manera deseada. Se define como la matriz de pesos y como el vector de sesgos. Se emplean funciones de activación que utilizan una suma ponderada, es decir, se encarga de devolver una salida (generalmente en un rango determinado) a partir de un valor de entrada. Para un número de dimensiones i , el resultado predicho se podría expresar como z . Podemos interpretar que cada peso w_i representa la influencia relativa de la entrada por la cual se multiplica, x_i

Diferentes espacios de colores.

Conjunto de validación (*validation*): conjunto de datos con el que se validan los modelos finales del conjunto de entrenamiento y se determina el modelo definitivo según: modelo con mejor ajuste a los datos, modelo más “cercano” al problema de predicción. Se suele utilizar para el ajuste de los hiperparámetros según los resultados obtenidos.

Conjunto de prueba (*test*): se realizan pruebas de funcionamiento del modelo y se obtiene el error “real” cometido respecto al modelo seleccionado. Solo se aplica una vez que el modelo ya esté completamente entrenado. Los datos del conjunto de prueba deberían ser de la misma distribución de datos que los del conjunto de validación.

Regularización: Dropout

La regularización se emplea para solucionar o prevenir el problema de alta varianza (*overfit*) y reducir los errores en la red neuronal. El parámetro de regularización se define como λ . Con la regularización lo que se pretende es, además de minimizar la función de costo J ,

reducir la complejidad, lo que se denomina minimización del riesgo estructural. El *dropout* es un tipo de regularización que consiste en recorrer cada una de las capas de la red y establecer la probabilidad de eliminar algunos de los nodos de cada una. En cada iteración (*mini-batch*), se adjudica a cada neurona de capa una probabilidad p de ser eliminados, empleando normalmente valores pequeños en las primeras capas y valores mayores en las capas intermedias.

Como efectos, el *dropout* hace que las neuronas sean menos dependientes de la salida de las neuronas conectadas, además, la red logra aprender características más robustas que son útiles para más subconjuntos de neurona, y puede promediar sobre diferentes redes entrenadas con diferentes inicializaciones aleatorias

Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales son un tipo de redes neuronales multicapa que se usan para procesar imágenes. Están basadas en operaciones de convolución, lo cual consiste en filtrar una imagen utilizando una máscara, y cada píxel de salida es una combinación lineal de los píxeles de entrada (Figura 16). Dicha entrada es, generalmente, una imagen $W \times H \times D$ (ancho \times alto \times número de canales). Al emplear convoluciones se reduce la carga computacional del sistema.

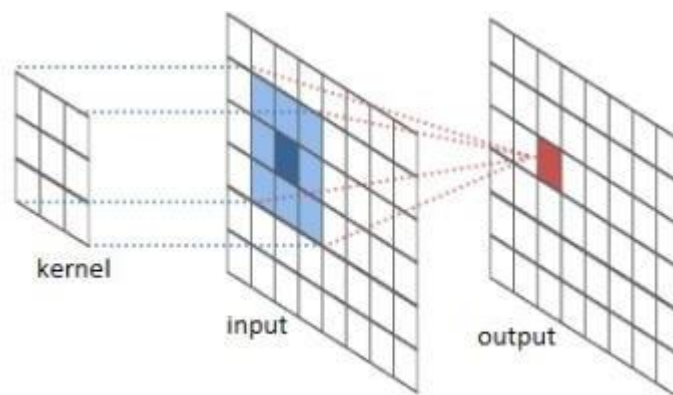


Figura 16: Resultado de convolución con un kernel de 3x3. Tomando el píxel mostrado en azul oscuro, se calcula la media ponderada de sus vecinos determinados por el tamaño y forma del kernel y se produce el píxel de salida (color rojo). Fuente [15].

Capa convolucional

La capa convolucional es el bloque de construcción central de una red convolucional que realiza la mayor parte del trabajo pesado computacionalmente. Estas capas tienen K filtros (o *kernels*) de dimensión $(W \times H \times D)$, elegida por el diseñador. Cada filtro, mediante convolución, genera un mapa de rasgos o características de tamaño $(W+1) \times (H+1) \times P$, donde P es el número de filtros que se utilizarán. El modo de operación en estas capas consiste en superponer el filtro y la imagen de entrada, se calcula la convolución entre los elementos de la imagen y del filtro, y el resultado se va almacenando en la llamada matriz de activación.

Capa Pooling

Se sub-muestra cada resultado generado en las capas convolucionales mediante la operación *max pooling* o *average pooling* sobre regiones contiguas de tamaño $(P \times P)$. *Max pooling* consiste en seleccionar únicamente el valor mayor de una ventana o región de la matriz de activación obtenida en la capa de convolución, descartando el resto de los

valores. *Average pooling* opera de la misma forma, pero el valor obtenido es la media de los valores de la ventana o región.

Imagen

Una imagen es una representación visual de un objeto o lugar que no cambia con el tiempo.

Imágenes en escalas grises

Cada píxel almacena un nivel de gris:

- Intensidad mínima: Negro
- Intensidad máxima: Blanco
- Intensidades medias: Grises medios

Para 8 bits, cada píxel puede tomar 256 niveles de gris, donde 0 es el negro y 255 es el blanco.

Que estamos analizando una imagen donde para cada valor de X sea una columna y cada valor de G una fila, un valor de intensidad, la intensidad mínima que espero me da me va a dar un color negro.

Para 1 bit, cada píxel solo puede tomar 2 valores, o (negro) o 255(blanco). Se conoce como imagen binaria

Imágenes a color

Cada píxel almacena un color. Los colores pueden representar en diferentes espacios de color.

Espacios RGB

Cada color es representado como una combinación lineal de tres componentes: Rojo (Larga), Verde (Media), Azul (corta)

Espacio HSI

Cada color es representado por medio de 3 características: hue, saturación e intensidad.

- Tono (hue): Es el color de acuerdo con su longitud de onda
Ejemplo: Rojo, Verde, Azul, Amarillo
- Saturación: es la cantidad presente del respectivo color. Un ejemplo sería poder distinguir entre el rojo y el rosado.
- Intensidad: es la cantidad de luz. Un ejemplo sería poder distinguir un rojo oscuro de un rojo claro.

Operación entre imágenes

Suma de imágenes a color

$$\begin{array}{|c|c|c|} \hline 3,2,1 & 5,2,4 & 3,8,1 \\ \hline 3,4,9 & 2,1,0 & 4,0,0 \\ \hline 5,8,1 & 3,3,3 & 3,5,2 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 3,1,0 & 1,1,1 & 5,1,0 \\ \hline 1,2,0 & 2,1,0 & 0,0,0 \\ \hline 0,0,0 & 0,0,0 & 0,0,0 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 6,3,1 & 6,3,5 & 8,9,1 \\ \hline 4,6,9 & 4,2,0 & 4,0,0 \\ \hline 5,8,1 & 3,3,3 & 3,5,2 \\ \hline \end{array}$$

Suma de imágenes en escala de grises

$$\begin{array}{|c|c|c|} \hline 3 & 5 & 3 \\ \hline 3 & 2 & 4 \\ \hline 5 & 3 & 3 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 2 & 1 & 3 \\ \hline 0 & 2 & 0 \\ \hline 3 & 3 & 3 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 5 & 6 & 6 \\ \hline 3 & 4 & 4 \\ \hline 8 & 6 & 6 \\ \hline \end{array}$$

Si la suma da más de 255 al mostrar la imagen se observará como una imagen blanca

Multiplicación de imágenes

$$\begin{array}{|c|c|c|} \hline 3 & 5 & 3 \\ \hline 3 & 2 & 4 \\ \hline 5 & 3 & 3 \\ \hline \end{array} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 3 & 5 & 0 \\ \hline 0 & 0 & 0 \\ \hline 1 & 1 & 0 \\ \hline \end{array}$$

Imagen en escala de grises o algún plano de RGB Imagen binaria

Es muy difícil multiplicar la imagen binaria, pero esta debe estar entre 0 y 1

Figura 17: Operación entre imágenes. Fuente [30]

Una vez hecha las operaciones entre las imágenes se procede a sacar el promedio entre la última matriz seguido de la varianza para saber que tanto se están desviando los datos de las imágenes, para calcular la varianza se utiliza la siguiente fórmula.

Una vez tengamos la varianza podemos saber cuál es la desviación estándar, con esto podremos hacer un análisis más a fondo de la imagen al interpretar los valores que se estén alejando es decir desde qué píxeles de la imagen comienza a cambiar esta misma.

Histograma

El histograma da información importante sobre la distribución de los niveles de gris en una imagen.

Operación de convolución 2D

La operación de convolución 2D nos permite hacer operaciones entre imágenes y así obtener una nueva.

Ejemplo de Convolución entre dos imágenes

$$\begin{array}{|c|c|c|c|c|} \hline 3 & 3 & 2 & 1 & 0 \\ \hline 0 & 0 & 1 & 3 & 1 \\ \hline 3 & 1 & 2 & 2 & 3 \\ \hline 2 & 0 & 0 & 2 & 2 \\ \hline 2 & 0 & 0 & 0 & 1 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline 0 & 1 & 2 \\ \hline 2 & 2 & 0 \\ \hline 0 & 1 & 2 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 12 & 12 & 17 \\ \hline 10 & 17 & 19 \\ \hline 9 & 6 & 14 \\ \hline \end{array}$$

Imagen en escala de grises o un canal de RGB Kernel, mascara de convolución o filtro Imagen filtrada

Figura 18: Convolución entre dos imágenes. Fuente [30].

1.7.2. Marco conceptual

Python

Python es un lenguaje de programación multiplataforma y de código abierto que puede utilizarse tanto para desarrollo web, creación de software y procesamiento de datos, entre muchos otros propósitos [16].

Librería

Una librería es un código que está a la disposición de todas las personas para que puedan incluirla dentro de su proyecto, el objetivo principal de estas librerías es realizar funciones en específico para agilizar el proceso de desarrollo.

Open cv

Es una biblioteca de código abierto especializada en el sistema de visión artificial y *machine learning*.

Estos algoritmos permiten identificar objetos, caras, clasificar acciones humanas en vídeo, hacer tracking de movimientos de objetos, extraer modelos 3D, encontrar imágenes similares, eliminar ojos rojos, seguir el movimiento de los ojos, reconocer escenarios [17].

Tensorflow

TensorFlow es una plataforma de código abierto de extremo a extremo para el aprendizaje automático. Cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos de la comunidad que permite que los investigadores innoven con el aprendizaje automático y los desarrolladores creen e implementen aplicaciones con tecnología de AA (aprendizaje automático) fácilmente [18].

Scikit-image

scikit-image es una colección de algoritmos para el procesamiento de imágenes. Está disponible de forma gratuita y sin restricciones. Incluye algoritmos de segmentación, transformaciones geométricas, manipulación del espacio de color, análisis, filtrado, morfología, detección de características [19].

Scikit-learn

Scikit-Learn es una librería gratuita para Python, cuenta con algoritmos de clasificación, regresión, clustering y reducción de dimensionalidad [20].

Numpy

NumPy es una librería de Python especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos.

Incorpora una nueva clase de objetos llamados arrays que permite representar colecciones de datos de un mismo tipo en varias dimensiones, y funciones muy eficientes para su manipulación [21].

Matplotlib

Matplotlib es una librería de Python especializada en la creación de gráficos en dos dimensiones.

Permite crear y personalizar los tipos de gráficos más comunes, entre ellos: Diagramas de barras, Histogramas Diagramas de sectores, Diagramas de caja y bigotes, Diagramas de

dispersión o puntos, Diagramas de líneas, Diagramas de áreas, Diagramas de contorno, Mapas de color y combinaciones de todos ellos [22].

Scipy

SciPy es una librería de Python que proporciona algoritmos para optimización, integración, interpolación, problemas de valores propios, ecuaciones algebraicas, ecuaciones diferenciales, estadísticas y muchos otros [23].

1.7.3. Marco legal y consideraciones éticas

Actualmente Colombia no cuenta con una legislación o marco legal que rija sobre el uso de la Inteligencia Artificial, ni la implementación de esta en proyectos del área de la salud. En este trabajo de investigación se garantiza la integridad en el tratamiento de los datos de los pacientes basándose en la ley estatutaria 1581 “Ley de Habeas Data” la cual trata sobre el manejo adecuado de la información recolectada, tiene por objeto desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos, y los demás derechos, libertades y garantías constitucionales a que se refiere el artículo 15 de la Constitución Política; así como el derecho a la información consagrado en el artículo 20 de la misma. Por lo que los datos personales de los pacientes no serán revelados ya que serán tratados de forma anónima. Además, no existe riesgo biológico ni ambiental en el desarrollo de este proyecto,

1.8. Metodología

La Facultad de Ingeniería cuenta con un grupo de investigación llamado DEARTICA, y este cuenta con varias líneas: Desarrollo de software, Inteligencia Artificial, Redes de Cómputo. Con el proyecto algoritmo para la extracción de características en imágenes de lesiones cutáneas en Colombia utilizando técnicas de visión artificial se tomó como línea de investigación la Inteligencia Artificial, debido a la que el producto final será un programa en la cual se necesita la implementación de diferentes tipos de algoritmos y funciones capaces de procesar información y llevar a cabo el análisis de imágenes.

Este trabajo implica la interacción con seres humanos puesto que se necesitan para la extracción de las imágenes tomadas por médicos a pacientes que nuestro algoritmo tendrá la tarea de analizar y arrojar una predicción.

Esta investigación es de tipo aplicada y de desarrollo tecnológico ya que sirve para generar conocimientos que se puedan poner en práctica en el sector productivo, con el fin de disminuir el tiempo en diagnóstico de este tipo de cáncer de piel el cual es muy importante detectarlo en el menor tiempo posible. Es una investigación cuantitativa ya que se tiene en cuenta la recolección de datos y se pone en práctica el uso de herramientas matemáticas, estadísticas e informáticas para medirlos. Esto permite hacer conclusiones generalizadas que pueden ser proyectadas en el tiempo. Ya que el estudio se realizará en un momento concreto se considera una investigación transversal durante un periodo determinado.

Este proyecto plantea una metodología estructurada en fases la cual toma como base fundamental el método de desarrollo de prototipado, que permitirá realizar pruebas y

correcciones al proyecto a lo largo de su desarrollo, con el fin de cumplir los objetivos planteados. A continuación, se describen cada una de las fases:

Fase I Investigación preliminar. Esta fase debe permitir desarrollar el primer objetivo propuesto de construir un conjunto de datos mediante la recolección de imágenes en repositorios públicos y capturas propias, con esto se propone buscar y clasificar las imágenes son más útiles en varios repositorios para así tener una mayor base a la hora de arrojar un resultado.

Fase II. Procesamiento de imágenes, en esta fase se hará el proceso en el cual las imágenes pasan a ser analizadas con el algoritmo de aprendizaje mediante la librería *OpenCv* que es una biblioteca libre de visión artificial originalmente desarrollada por Intel y así poder hacer la extracción de las características.

Fase III. Eliminación de características no deseadas, teniendo en cuenta el tercer objetivo del proyecto, en esta fase se busca eliminar todo tipo de detalle el cual no permite analizar bien las imágenes, dando como resultado una imagen limpia de la cual poder obtener mejores resultados.

Fase IV: Validación de características, en esta etapa final se realizan pruebas a un conjunto de imágenes para así probar la eficiencia que tiene el algoritmo.

A continuación, se presenta un cuadro metodológico en el cual se describe la articulación entre las fases de la metodología y los objetivos del proyecto, junto con las actividades que permitirán dar cumplimiento a los mismos.

Fases	Objetivo específico	Actividades
Investigación preliminar	<p>Determinar cuáles son las imágenes más pertinentes a la hora de analizar, mediante la búsqueda e investigación en diferentes repositorios.</p> <p>Determinar las mejores técnicas de procesamiento de imágenes mediante visión artificial.</p>	<ol style="list-style-type: none"> 1. Buscar repositorios de imágenes sobre melanoma. 2. Investigar técnicas de IA para el tratamiento de imágenes. 3. Clasificar las imágenes que cumplan con las características.
Procesamiento de imágenes	<p>Procesar las imágenes que se hayan seleccionado para su debida extracción de información.</p>	<ol style="list-style-type: none"> 1. Limpieza de las imágenes obtenidas en la Fase I por medio del algoritmo. 2. Eliminación de ruido en las imágenes

		3. Extraer características relevantes de las imágenes.
Eliminación de características no deseadas	Eliminar datos no necesarios de las imágenes para un procesamiento adecuado	1. Suprimir características no relevantes de las imágenes.
Validación de características	Validación de las características obtenidas y hacer testeos para determinar cuánta es su efectividad a la hora de dar una predicción.	<ol style="list-style-type: none"> 1. A través de diferentes pruebas validar los resultados. 2. Hacer una tabla de resultados para hacer un análisis de su evolución. 3. Elaborar un informe de la eficiencia del algoritmo.

Tabla 1. Descripción fases de la metodología del proyecto. Fuente propia.

2. DESARROLLO

Este capítulo está dividido en dos partes, empezando por la explicación de conjunto de datos que se utilizaron para probar el modelo una vez desarrollado, en este apartado se encuentran las columnas que conforman este conjunto de datos y de donde fue extraído; y una segunda parte en la cual se describe el proceso realizado para la construcción del modelo para dar solución a la problemática planteada anteriormente. Además, se explica la forma en la cual fue entrenado este modelo.

2.1. Construcción del conjunto de datos

Un dataset es un conjunto de datos contenidos en un único archivo, en este caso nuestro dataset corresponde a un grupo de imágenes de melanoma para la detección de este mismo, el dataset se divide en dos carpetas diferentes una para el entrenamiento de la red neuronal y otra la para la validación o prueba del entrenamiento, cada carpeta contiene imágenes específicas para su respectivo uso.

Los datos han sido extraídos de un dataset en específico el cual fue proporcionado por **SIIM – ISC Melanoma Classification**. El conjunto de datos contiene 33.126 imágenes de entrenamiento dermatoscópicas de lesiones cutáneas benignas y malignas únicas de más de 2.000 pacientes. Cada imagen se asocia con una de estas personas mediante un identificador de paciente único. Todos los diagnósticos malignos se han confirmado mediante histopatología y los diagnósticos benignos se han confirmado mediante el acuerdo de expertos, el seguimiento longitudinal o la histopatología [24]

El conjunto de datos fue generado por la *International Skin Imaging Collaboration (ISIC)*, *Society for Imaging informatics in Medicine (SIIM)* y las imágenes provienen de las siguientes fuentes: *Hospital Clínic de Barcelona, Medical University of Vienna, Memorial Sloan Kettering Cancer Center, Melanoma Institute Australia, University of Queensland y University of Athens Escuela de Medicina* [25].

El dataset está distribuido con imágenes y metadatos con un formato de imágenes médicas de uso común con un tamaño de 1024 x 1024 uniforme [26]. Los metadatos tienen un formato DICOM [27], en archivos CSV que contienen diferentes columnas para la predicción del melanoma las cuales son:

- **image_name**: identificador único, apunta al nombre de archivo de la imagen DICOM relacionada
- **patient_id**: identificador de paciente único
- **sex**: el sexo del paciente (cuando se desconozca, estará en blanco)
- **age_approx**: edad aproximada del paciente en el momento de la obtención de imágenes
- **anatom_site_general_challenge**: ubicación del sitio de la imagen
- **diagnosis**: información de diagnóstico detallada (solo tren)
- **benign_malignant**: indicador de malignidad de la lesión fotografiada
- **target**: versión binaria de la variable de destino

Cabe resaltar que para el alcance solo se usara el dataset con las imágenes para su detección, el resto de información del CVS sería muy útil para que el proyecto tenga más variables que analizar, es decir, poder clasificar el melanoma y dar el diagnostico teniendo en cuenta ya sea el sexo, la edad u etnia el cual pueda generar estadísticas para su respectivo análisis.

Se cuenta con otro dataset proporcionado por el curso *Computer Vision y Machine Learning con Python 2021* [15] el cual contiene un conjunto de imágenes divididas en dos carpetas una de 2637 imágenes para el entrenamiento y otra de 660 imágenes para las pruebas y

validaciones.

2.2. Construcción del modelo

Este modelo se desarrolló con el lenguaje de programación Python en su versión 3.9.6 a través del entorno de desarrollo integrado multiplataforma de código abierto para programación científica Spyder. Usando librerías para el manejo de Inteligencia Artificial tales como *Open CV*, *Tensorflow*, *scikit-image*, *scikit-learn*, *Numpy*, *MatplotLib*, *Scipy* que utilizaremos en nuestro producto final.

En primera instancia se busca analizar la imagen y todas sus características (forma, borde, contorno, ruido etc.) para tener un mejor entrenamiento de nuestro algoritmo, con esto teniendo mejores resultados en los diagnósticos. A continuación, vemos que características analizamos de las imágenes para luego con una red neuronal convolucional y Deep Learning nuestro modelo pueda diferenciar entre un melanoma maligno y uno benigno con su tasa de efectividad a la hora de dar este diagnóstico.

2.2.1. Caracterización de las imágenes

Para la caracterización se creó un método que analiza cada detalle de la imagen y las ajusta de acuerdo con el objetivo final que es tratar esas imágenes y generar un estándar el cual todo nuestro dataset las incorpore para posteriormente sirvan para el entrenamiento. El método recibe una imagen de entrada y retorna una imagen con el estándar que necesitamos de salida, a continuación, están descritos cada una de las funciones del algoritmo para hacer la caracterización, se usaron imágenes de pruebas del dataset para la explicación de este método.

Hietograma: Se genera un histograma con una imagen de prueba en este caso un melanoma benigno aplicando la función *Flatten* las cuales va a dejar los datos en un arreglo en secuencia que tiene todos los valores que tenía la imagen. Generamos dos gráficas una en escala de colores grises [Figura 19] y otra gráfica en escala de colores RGB [Figura 20] que nos permite ver la distribución de las frecuencias de una muestra.

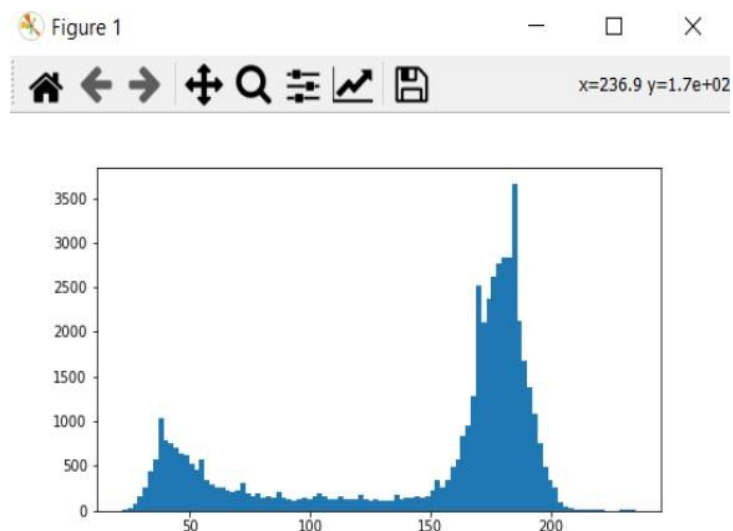


Figura 19: Histograma en escala de colores grises Fuente propia

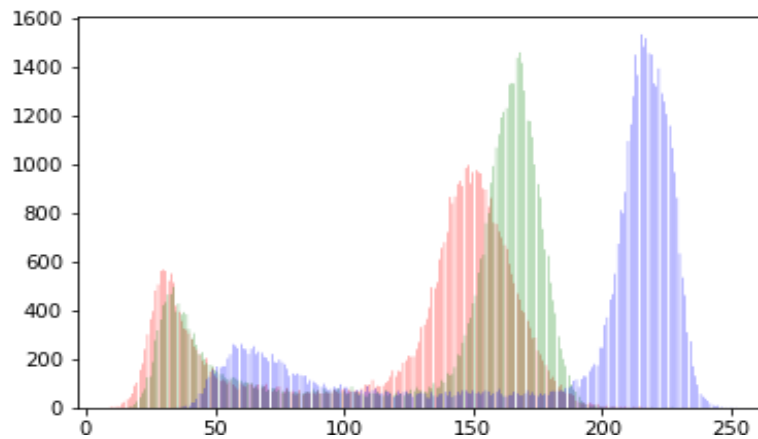
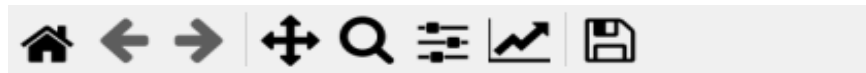


Figura 20: Histograma en escala de colores RGB. Fuente propia.

Segmentación: Se segmenta la imagen en escala de colores grises de la muestra generando una máscara con la información de la forma del objeto con el que vamos a trabajar algunas características como el tamaño del objeto, contorno, borde entre otras que podemos utilizar para hacer una limpieza de esta misma es decir enfocando solo lo que necesitamos analizar [Figura 21].



Figura 21: melanoma segmentado en color blanco. Fuente propia.

Detección de características: digamos que cuando estamos hablando de detección de cáncer, en este caso, cáncer de piel (melanoma), uno de los pasos más importantes que debemos hacer es detectar el lunar de la imagen sobre el fondo, o sea, sobre la piel sana. Y para eso tenemos que segmentar el lunar y dejar el fondo aparte. Esto se hace haciendo el uso de etiquetas, nombrando a cada conjunto de pixeles agrupados [Figura 22] cada figura en amarillo son etiquetas que luego usando unas funciones de *Open CV* podemos calcular cual es la que tiene el mayor número de pixeles y eliminar el resto.

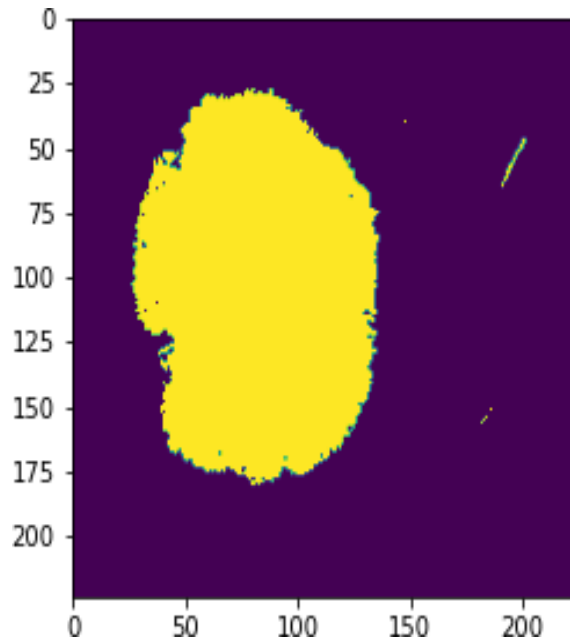


Figura 22: melanoma con etiquetas en escala a color y con valores binarios Fuente propia.

El resultado de este método ayuda a tener una imagen con menos objetos que analizar a continuación, vemos como quedaría nuestra máscara al solo dejar la imagen que queremos analizar (melanoma) [Figura 23] y el resto de los píxeles que había se han ido.

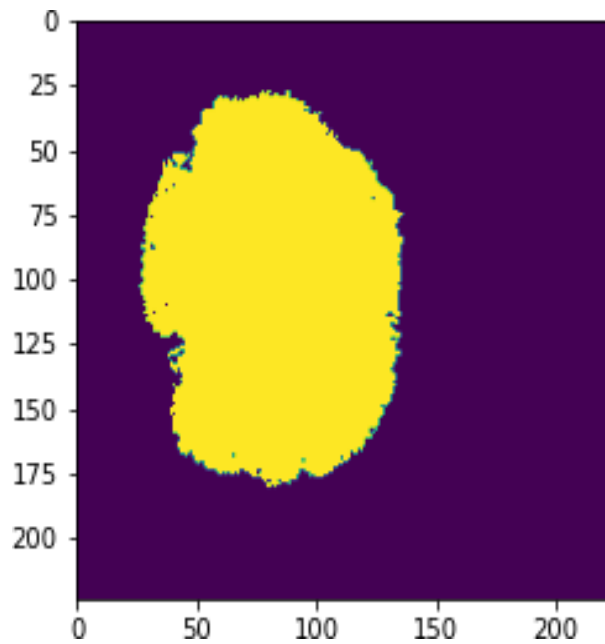


Figura 23: melanoma con etiquetas de mayores pixeles sin el resto. Fuente propia.

Relleno de contorno: podemos mejorar nuestra máscara rellenando todos los huecos que estén en la imagen para así enfocarnos en todo el contorno, para esto usamos la librería *ndimage* que nos permite rellenar estos vacíos de píxeles y nos quede una imagen con más forma y posibilidad de analizar [Figura 24].

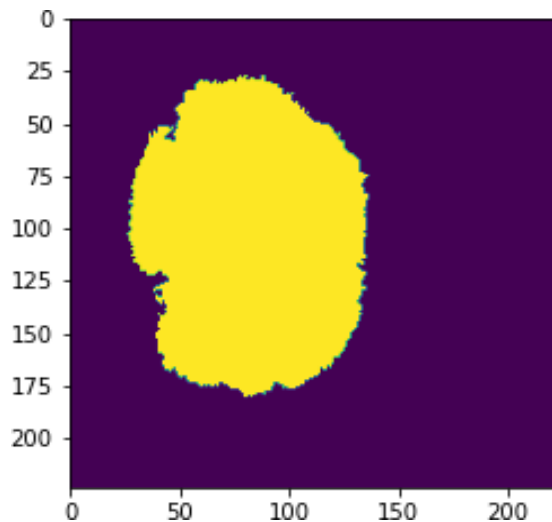


Figura 24: melanoma relleno de píxeles. Fuente: Fuente propia.

Generar una imagen convexa: Hacer mediciones en nuestra imagen, es muy importante puesto que nos da información del objeto, y a partir de estas características podemos realizar predicciones con varios métodos de Inteligencia Artificial. Una de esas mediciones es el *ConvexHull* que básicamente lo que hace es encontrar el polígono convexo más pequeño que encierra el objeto y el *ConvexImage* que rellena el polígono convexo para tener una imagen convexa. Un ejemplo de esto sería la aplicación del *ConvexHull* sobre este tipo de problemas de detección de cáncer, es que si la imagen del lunar tiene pedazos donde haya más entradas, pues es más probable que sea cáncer.

Para poder analizar las características de esta última imagen es muy importante saber su perímetro y dimensiones esto con el fin de saber si información o saber qué tipo de objeto es la imagen para procesar, un rectángulo triángulo etc. Para aclarar esto no funciona para todo tipo de muestra, hay imágenes de melanomas que no tienen ningún tipo de figura geométrica con esto ignorando este paso.

Esto se hace con el *BoundingBox*, nos permite tomar todas las mediciones de nuestra máscara y generar un rectángulo con cada una de sus dimensiones, aplicando este método nos quedaría una máscara con una forma y convexa para su enfoque [Figura: 25].

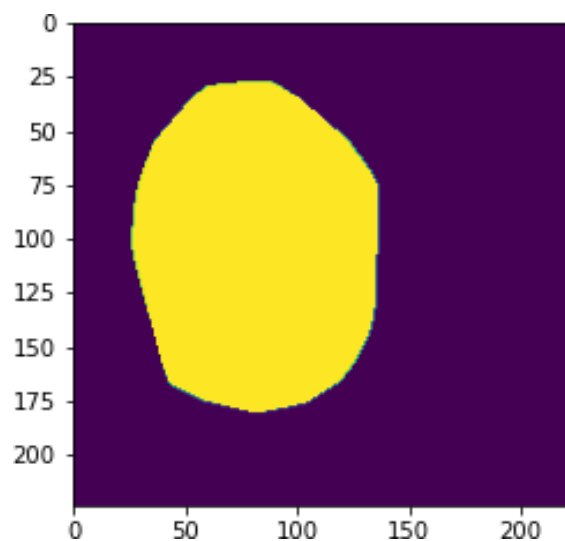


Figura 25: imagen convexa de melanoma. Fuente propia

Al observar nuestra imagen nuevamente generamos un rectángulo el cual solo detecta nuestro cáncer de melanoma e ignorando el resto de piel para así poder analizarlo [Figura 26].

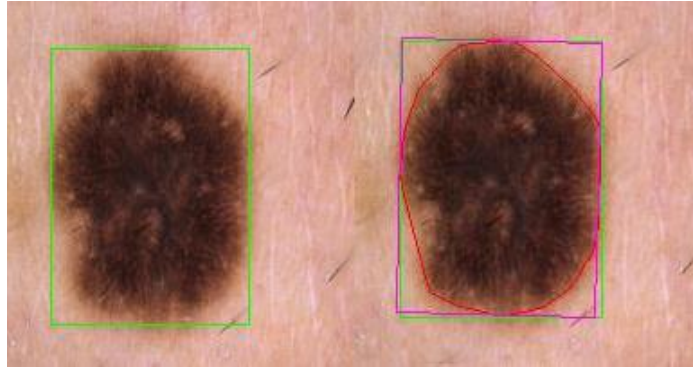


Figura 26: Melanomas con ConvexHull y BoundingBox Fuente propia.

Filtrado de imagen: Este filtro nos ayuda a mejorar el ruido, difuminación y el peso de la imagen para dar una mejor muestra que analizar, los filtros que se utilización son los siguientes:

Filtro promedio: Se utiliza la imagen con un kernel con distintos tipos de máscaras, podemos utilizar una de 3x3, 5x5, 15x15 y 35x35 dependiendo entre mayor sea el kernel la imagen quedará más difuminada.

Detección de bordes: Es muy importante para el procesamiento de bordes puesto que nos permiten distinguir las imágenes de otras.

Filtro promedio Gaussiano: funciona de forma similar al filtro promedio, pero con la diferencia de que se les da más peso a los valores de la imagen del centro, es decir en la matriz de pixeles el valor central será el mayor y no causa ningún tipo de distorsión.

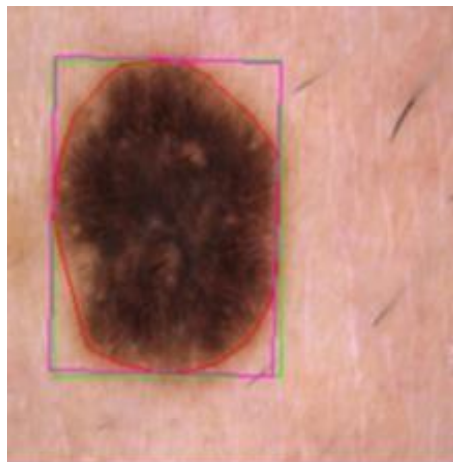


Figura 27: Melanoma con Kernel (3x3). Fuente: Se obtuvo del autor a través del melanoma. Fuente propia.

En este caso se optó por usar un filtro con kernel de 3x3 puesto que ayudaba a mejorar la distinción de la imagen creando una nueva con mejor resolución [Figura 27].

Características que no se aplicaron a la imagen: existen muchas más formas de hacer la caracterización de las imágenes, pero a la hora de agregar más y más y no enfocarnos en las que nos son útiles en el proyecto podría causar una deformación de la muestra con está teniendo muy malos resultados a la hora de pasarlas a la red neuronal. Entre esas otras características tenemos el vector gradiente discreto que nos ayuda a

detectar los cambios de intensidad en la imagen es decir los bordes y la homografía que nos permite manipular la perspectiva de cualquier imagen y la separación del melanoma del fondo.

2.2.2. Creación del modelo

Partiendo del hecho de una red neuronal veamos ¿Qué es una neurona artificial? Es básicamente un sistema de características que arroja una salida con un problema binario 0 y 1 [Figura: 28]. A partir de unos pesos, que son unas constantes que se multiplican por cada característica las cuales son w_0 , w_1 , w_2 hasta w_b se intenta obtener una salida en la clasificación.

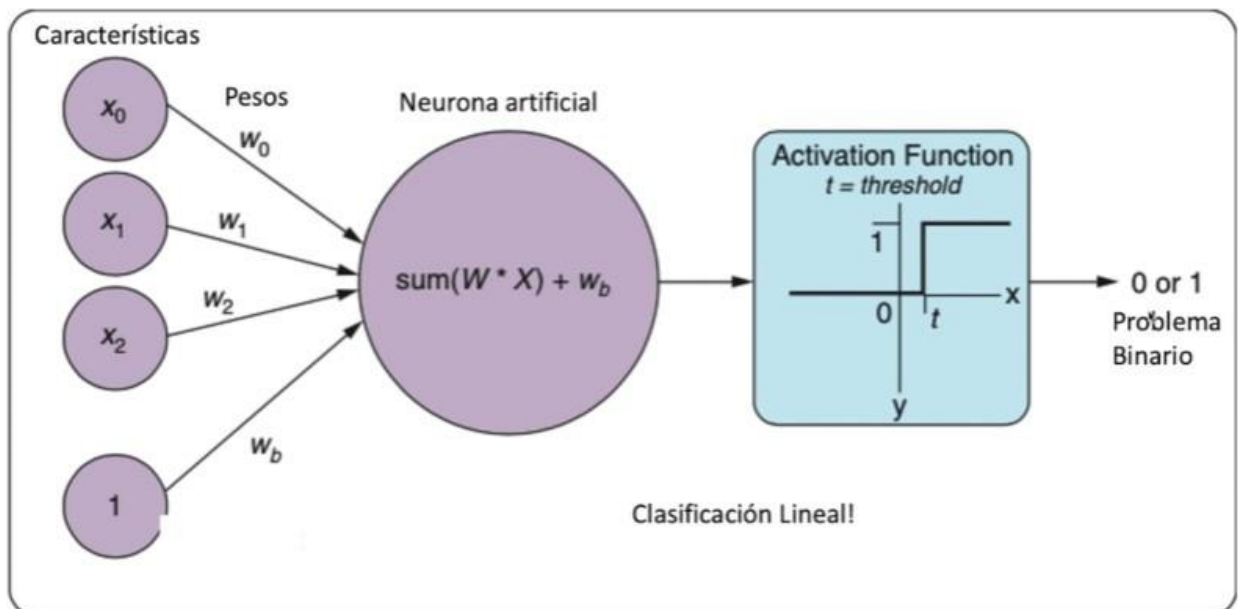


Figura: 28: Funcionamiento de una neurona artificial. Fuente [15].

Ahora bien, al saber que es una neurona artificial entonces ¿qué es una red neuronal artificial? son modelos informáticos de unidades conectadas o nodos diseñados para transmitir, procesar y aprender de la información (datos) de manera similar a cómo funcionan las neuronas (células nerviosas) en los humanos [11].

El éxito de las redes neuronales en temas de inteligencia artificial es que puede modelar sistemas muy complejos variando parámetros.

Parámetros básicos pen una red neuronal

- Cantidad de capas ocultas
- Cantidad de neuronas en cada capa
- Función de activación
- Método de optimización para minimizar el error en la clasificación.

Red neuronal convolucional y Deep Learning

Sabiendo que una convolución es una operación entre imágenes, se comienza con la imagen de entrada y luego se pasa por un kernel y se hace la convolución entre la imagen y el kernel dando de resultado una nueva imagen, esta misma se pasa por la función RELU que quita los números negativos y deja los positivos más adelante se hace el *Max Pooling* que su función es sacar los valores mayores y agruparlos, finalmente se hace el *Flattening* que vuelve la matriz en vector de una sola dimensión para sacar las características de la

imagen [Figura 26].

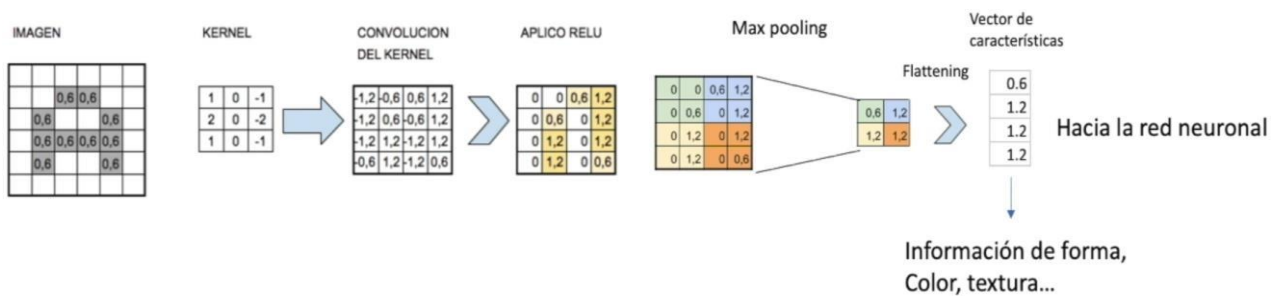


Figura 29: Proceso del funcionamiento de una red neuronal convolucional. Fuente [15].

Haciendo uso de la biblioteca *Tensorflow* para aprendizaje automático en conjunto al *framework Keras* usando librerías como *Sequential* el cual nos ayuda al proceso de experimentación rápida y creación del modelo, *Dense* que nos permite crear nuestras capas de la red, *Convolution2D* que es una capa que crea un núcleo de convolución que se convoluciona con la entrada de la capa para producir un tensor de salidas, *Maxpooling2D* que es una capa que aplica un filtro que reduce la dimensionalidad de la imagen y *Flatten* que serializa un *tensor multidimensional*. Finalmente usamos la librería *Open CV* para leer las imágenes y *Numpy* para hacer operaciones matemáticas.

Una vez tenemos nuestras librerías listas para usar creamos el modelo utilizando *Sequential* y como si fuera una lista comenzamos a añadir capas para nuestra red neuronal. La capa inicial es la que recibe las imágenes de entrada, que se normalizan para su dimensión sea 244x244x3, correspondiendo al alto *H (height)*, ancho *W (width)* y profundidad *D (depth)* o canales que representan los valores RGB (*red, green, blue*). El objetivo de las cinco capas convolucionales es la extracción de características. Las dos primeras incluyen una etapa de *Pooling 2x2* con solapamiento, con el objetivo de reducir las dimensiones de la salida de la red para aumentar la profundidad de la siguiente capa de convolución. Con esto se podrán conseguir características más complejas, aunque se elimine parte de información espacial, cabe aclarar que el tamaño de las imágenes a procesar todas deben tener el mismo tamaño. Se declaran los hiperparámetros como el número de épocas, el tamaño del *batch*, la carpeta donde están cada uno de los datos de las imágenes de entrenamiento y testeo. Luego se carga el conjunto de datos que disponemos para entrenar el modelo. Usaremos el dataset cuyas imágenes son muy pocas cuando se trata de redes convolucionales [28] por primera instancia puesto que la capacidad de nuestro computador no da para el análisis del primer dataset. Este se puede considerar pequeño para un sistema de Deep Learning puesto que solo consta de 2637 imágenes para el entrenamiento y agregarle muchas capas puede hacer que demore muchísimo en converger, por lo tanto, la estructura que usamos es un poco simple.

2.2.3. Entrenamiento

Los datos de entrenamiento se seleccionan de forma aleatoria y se pasan por el método de caracterización para que tenga mejores resultados, se recorren cada una de las imágenes y se guardan en matrices que el algoritmo recorre para su entrenamiento.

Una vez entrenado el modelo se pasa a un texto o validación con datos de prueba recorriendo cada una de las imágenes guardando los resultados en matrices, se valida si es 0 será benigno y si es 1 será maligno.

Tamaño del batch

Pasar todo el conjunto de datos a la red neuronal puede ser muy costoso computacionalmente y muy lento, por lo tanto, lo que se hace es dividirlo en lotes (*batches*) o conjuntos de datos. El tamaño del lote o *batch size* es el número total de datos de entrenamiento que contiene cada *batch*, hay tres opciones:

Modo *batch*: donde el tamaño del *batch* es igual al conjunto de datos total, por lo que los valores de iteración y época son equivalentes.

Modo de *mini-batches*: donde el tamaño del *batch* es mayor que uno, pero menor que el tamaño total del conjunto de datos.

Modo estocástico: donde el tamaño del *batch* es igual a uno, por lo tanto, los parámetros del gradiente y pesos de la red neuronal se actualizarán después de cada muestra. Normalmente las redes neuronales se entrenan más rápido con *mini-batches* debido a que se actualizan los pesos después de cada propagación y, además, requiere menos memoria.

Por lo tanto, el modo empleado en los experimentos será el de *mini-batches*. Lo ideal para conseguir un estudio completo sobre los distintos tamaños de *mini-batch* sería probar los valores del *mini-batch* más grandes llegando incluso a igualar el tamaño del *batch*. Por lo general, a menor tamaño de *mini batch* los pesos oscilarán más sin poder llegar a obtener un resultado generalizado debido a que la cantidad de datos será insuficiente. Según se va aumentando el tamaño, las oscilaciones serán mucho más suaves, pero también se dará una convergencia más lenta.

Iteración y época

Una iteración es una sola actualización de los pesos de un modelo durante el entrenamiento. Consiste en el cómputo de los gradientes de los parámetros con respecto a la pérdida en un solo lote (*batch*) de datos. Para el entrenamiento de datos con una red neuronal lo más normal es que no sea suficiente con pasar a través de todos los datos una sola vez. Una época (*epoch*) es el recorrido de entrenamiento completo por todo el conjunto de datos a través de la red neuronal, realizando una pasada hacia delante y una pasada hacia atrás para actualizar los pesos, por todos los ejemplos de entrenamiento. Por lo tanto, siendo N el número total de ejemplos, las iteraciones necesarias para completar una época resultan $N / (\text{tamaño } batch)$. Por ejemplo, teniendo 1000 datos de entrenamiento, y su tamaño de lote es 500, entonces se necesitarán 2 iteraciones ($1000/500$) para completar 1 época. Según se aumenta el número de épocas, más veces se cambian o actualizan los pesos de las capas de la red neuronal, produciendo una curva desde ajuste insuficiente (*underfitting*), a óptima, e incluso a un ajuste excesivo (*overfitting*).

Tasa de aprendizaje

La tasa de aprendizaje o *learning rate* es un hiperparámetro que controla cuánto estamos ajustando los pesos de la red con respecto al gradiente de pérdida. Cuanto menor sea el valor, más lento viaja a lo largo de la pendiente descendente, lo cual sería una forma de asegurarse de no perder ningún mínimo local, pero podría llevarnos mucho tiempo hasta que llegara a converger, o quedarse estancado y no llegar nunca. Por otro lado, un valor muy grande de *learning rate* podría tener oscilaciones tan grandes que nunca llegue a alcanzar el punto óptimo o incluso divergir.

Función de pérdida

Una función de pérdida, o *loss function*, es una función que evalúa la desviación entre las predicciones realizadas por la red neuronal y los valores reales de las observaciones

realizadas durante el aprendizaje. Cuanto menor es el resultado de esta función, más eficiente es la red neuronal.

Teniendo en cuenta lo anterior para nuestro proyecto usamos un *batch* de 32 con 4 épocas, con validación de aprendizaje *accuracy* y la función de pérdida *binary_crossentropy*. Cabe resaltar que estos datos son experimentales. en los cuales nos dio un resultado de forma gráfica en cuanto a nuestra tasa de pérdida [Figura 28] podemos ver como disminuye por cada época teniendo una mayor efectividad.

3. VALIDACIÓN

A continuación, se encuentran explicado el proceso realizado para validar el funcionamiento del modelo, utilizando para esto distinta cantidad de épocas.

Para realizar las pruebas se ha implementado la técnica de *Deep Learning* mencionada anteriormente en cuanto a las capas entrenadas: se tuvieron varios entrenamientos con diferentes capas de neuronas números de épocas y *bracht* para llegar a la que sea más eficiente a la hora de dar el diagnóstico de si el cáncer de melanoma es benigno o maligno. Se procedió a hacer un estudio y evaluación de las épocas, el tamaño del *mini-batch* y el *learning rate*, de forma que se lleguen a establecer unos valores fijos considerados óptimos para el posterior entrenamiento con diferentes redes y conjuntos de datos.

Para establecer el número de épocas, ya que no hay respuesta correcta a cuál es el adecuado y éste será diferente para cada conjunto de datos, habrá que hacer un estudio del conjunto de datos respecto a nuestro modelo de forma que se consigan las épocas adecuadas para asegurar la convergencia de la red. Se han realizado diferentes pruebas para determinar un número fijo de épocas a utilizar en los diferentes entrenamientos y se ha comprobado que, a partir de 4 épocas, tanto el *accuracy* como el *loss convergen* los datos comienzan a ser más precisos [Figura 27].

Pruebas experimentales cambiando el número de épocas con el mismo *bracht* y misma cantidad de neuronas (128,50,1) y una capa de convolución, pero con épocas que iban ascendiendo de 2 en 2 para ver su evolución.

Ajuste de Hiperparámetros - Prueba 1	
Tamaño de bacht	32
Épocas	2
Función de perdida	2.0216
Tasa de aprendizaje	0.7182

Tabla 2: Valores escogidos para realizar las pruebas con el fin de ajustar los hiperparámetros prueba 1. Fuente propia.

Para las pruebas se comenzó con parámetros básicos como la cantidad de bacht y épocas, en primera instancia se usó 2 en las épocas puesto que se planteó hacer pruebas de intervalo de 2 en 2, en nuestro primer resultado [tabla 2] podemos analizar varios hiperparámetros que nos dio a la hora de depurar nuestro modelo, en primer lugar tenemos el tamaño del bacht que fue de 32 que es el número de datos que tiene cada iteración de un ciclo (*epoch*), luego tenemos las épocas (*epoch*) en este caso de 2, que es la cantidad de veces que pasan los datos por nuestro algoritmo, en el tercer hiperparámetro tenemos la función de perdida (*lost function*) que es la función que nos dice que tan buena es la red neuronal, un resultado alto indica que la red neuronal tiene un desempeño pobre y un resultado bajo indica que la red neuronal está haciendo un buen trabajo y por ultimo tenemos la tasa de aprendizaje (*Learning rate*), este hiperparámetro indica que tan largo será el camino que tome el algoritmo de optimización. Si el valor es muy pequeño la actualización se puede quedar atrapada en un mínimo local y los valores de W y b no cambiarían correctamente, igualmente la red neuronal tardara mucho más tiempo en optimizar. Por otro lado, si los valores son muy altos la actualización puede pasarse del punto perfecto que sería el mínimo global y nunca encontrarlo y aunque el aprendizaje sea más rápido nunca llegara al mínimo global.

En esta primera prueba nuestra función de perdida fue de 2.0216 y teniendo en cuenta lo mencionado anteriormente esto es un desempeño no muy favorable puesto que entre mayor se aleje el numero del 0 peor desempeño tendrá, para la tasa de aprendizaje tenemos un 0.7182 si bien no es un numero malo, ya que esto lo podemos traducir a 71%

de aprendizaje tenemos que tener en cuenta que nuestra red neuronal no tendrá buenas predicciones por el simple hecho de que tiene una muy alto número en la función de perdida.

Ajuste de Hiperparámetros - Prueba 2	
Tamaño de batch	32
Épocas	4
Función de pérdida	3.0334
Tasa de aprendizaje	0.8682

Tabla 3: Valores escogidos para realizar las pruebas con el fin de ajustar los hiperparámetros prueba 2. Fuente propia.

Para la segunda prueba [Tabla 3] se aumentó el número de épocas con el doble de iteraciones, es decir en este caso fueron 4, analizando la función de perdida podemos observar como su numero a aumentado con respecto a la primera prueba, sin embargo su tasa de aprendizaje también aumento de tener 0.71 a 0.86, es una diferencia significativa puesto que esto significaría que el modelo tiene un 86% de probabilidad de acertar el la predicción, ahora bien se sabe que si la función se perdida es muy alta esto podría causar muchos inconvenientes puesto que podríamos determinar que el modelo tiene un alto porcentaje de aprendizaje pero la mayor parte de estas predicciones podrían ser un falso positivo, es decir para el modelo podría ser la respuesta correcta pero a la hora de la verdad fue equivocada.

Ajuste de Hiperparámetros - Prueba 3	
Tamaño de batch	32
Épocas	6
Función de pérdida	1.1330
Tasa de aprendizaje	0.7227

Tabla 4: Valores escogidos para realizar las pruebas con el fin de ajustar los hiperparámetros prueba 3. Fuente propia.

En la tercera prueba [Tabla 4] podemos ver un muy buen avance en cuanto la función de perdida, esta fue realizada con un total de 6 épocas las cuales nos dieron resultados más prometedores que pruebas anteriores puesto que la función de perdida disminuyo considerablemente pasó de 3.0 a un 1.1 la diferencia es muy significativa, si bien se sabe que entre menor sea este número hubo un mejor entrenamiento también tenemos que tener en cuenta cuanto fue la tasa de aprendizaje de nuestro modelo que este caso fue de un 0.72 es decir un 72% de efectividad para posibles predicciones, es una cifra en el rango hasta ahora que nos podría ser muy útil para nuestro modelo.

Ajuste de Hiperparámetros - Prueba 4	
Tamaño de batch	32
Épocas	8
Función de pérdida	2.3615
Tasa de aprendizaje	0.7242

Tabla 5: Valores escogidos para realizar las pruebas con el fin de ajustar los hiperparámetros prueba 4. Fuente propia.

Para la prueba número 4 [Tabla 5] según los datos de los hiperparámetros fue de una buena prueba en la anterior a una mala en todos los aspectos ¿Por qué? Comparando la

función de pérdida vemos que en vez de disminuir aumentó, con esto teniendo un alto índice de poco aprendizaje, en la tasa de aprendizaje no vario con la anterior, es decir se mantuvo estable, pero con mayor tasa de pérdida, la prueba 4 con una cantidad de épocas de 8 no es recomendable para posibles predicciones.

Ajuste de Hiperparámetros – Prueba 5	
Tamaño de batch	32
Épocas	10
Función de pérdida	1.1566
Tasa de aprendizaje	0.7879

Tabla 6: Valores escogidos para realizar las pruebas con el fin de ajustar los hiperparámetros prueba 5. Fuente propia.

Con la prueba número 5 [Tabla 6] se obtuvieron resultados más favorables puesto que tuvimos una función de pérdida de 1.1 con esto teniendo una mayor certeza de que nuestro modelo aprendió con pocos menos errores y una tasa de aprendizaje de 0.78 es decir un 78% de probabilidad de que nuestro modelo al compilarlo nos muestre el resultado que es, una cifra muy por encima de otras pruebas anteriores, en esta prueba se utilizaron 10 épocas la cual nos dio como resultado estos datos en los hiperparámetros que nos permiten ser un buen candidato para el modelo final.

Ajuste de Hiperparámetros - Prueba 6	
Tamaño de batch	32
Épocas	12
Función de pérdida	3.2615
Tasa de aprendizaje	0.7667

Tabla 7: Valores escogidos para realizar las pruebas con el fin de ajustar los hiperparámetros prueba 6. Fuente propia.

En la prueba numero 6 con una cantidad de épocas de 12 [Tabla 7] vemos que ha pasado como en casos anteriores que, de pasar de tener buenos resultados a tener datos muy poco prometedores, en este caso en particular la función de pérdida fue muchísimo mas alta que en la anterior y su tasa de aprendizaje disminuyó con esto podemos concluir que la prueba numero 6 no es viable para una posible predicción u adaptación al modelo final.

Ajuste de Hiperparámetros - Prueba 7	
Tamaño de batch	32
Épocas	14
Función de pérdida	1.7801
Tasa de aprendizaje	0.7424

Tabla 8: Valores escogidos para realizar las pruebas con el fin de ajustar los hiperparámetros prueba 7. Fuente propia.

En la séptima prueba se uso una cantidad de épocas de 14 [Tabla 8] teniendo resultados no tan malos, pero no mejor que lo que se esperaba, es decir su función de pérdida no es tan alta pero su tasa de aprendizaje disminuyo en comparación a la prueba 5, se mantiene aun en el promedio, pero no tiene la tasa de aprendizaje suficientemente alta para considerar que podría servir para el modelo final.

Ajuste de Hiperparámetros - Prueba 8	
Tamaño de batch	32
Épocas	16
Función de pérdida	2.3411
Tasa de aprendizaje	0.7652

Tabla 9: Valores escogidos para realizar las pruebas con el fin de ajustar los hiperparámetros prueba 8. Fuente propia.

Para la prueba 8 con 16 épocas [Tabla 9] hay varias cosas que analizar, una de ellas es que si bien subió la tasa de aprendizaje también subió la función de pérdida, en esta iteración los hiperparámetros estuvieron directamente proporcional, es decir si uno subió el otro también, hubo un 76% en el porcentaje de asertividad de nuestro modelo, es una cifra considerablemente acertada pero también hubo una cantidad alta en cuanto a la función de pérdida, es decir que siempre que tengamos esta cifra alta o por encima de 1.1 la cantidad de épocas en este caso 16 no serian aptas para el modelo final.

Ajuste de Hiperparámetros - Prueba 9	
Tamaño de batch	32
Épocas	18
Función de pérdida	2.2417
Tasa de aprendizaje	0.7545

Tabla 10: Valores escogidos para realizar las pruebas con el fin de ajustar los hiperparámetros prueba 9. Fuente propia.

En nuestra penúltima prueba la numero 9 [Tabla 10] con un total de 18 épocas [tabla 10] vemos resultados poco favorables puesto que la tasa de pérdida sigue siendo considerablemente alta para dar predicciones, puede tener un alto porcentaje en la tasa de aprendizaje, pero nuevamente serian resultados de no tan certeros a la hora de dar un diagnóstico.

Ajuste de Hiperparámetros - Prueba 10	
Tamaño de batch	32
Épocas	20
Función de pérdida	1.7279
Tasa de aprendizaje	0.7455

Tabla 11: Valores escogidos para realizar las pruebas con el fin de ajustar los hiperparámetros prueba 10. Fuente propia.

En la ultima prueba con un total de 10 [Tabla 11] se usó un tamaño significativo de épocas puesto que para redes neuronales normalmente se usan grandes cantidades de esta, sin embargo los resultados obtenidos si bien no son los mejores se tiene en cuenta que la función de pérdida no es tan alta en comparación a otras y su tasa de aprendizaje de mantiene en el promedio de todas las pruebas el cual es un 75% de efectividad, que este caso fue de 74%, para esta prueba final podemos concluir que se puede mejorar aun mas el modelo con muchas mas épocas pero con un dataset mucho mas grande que pueda darnos resultados muy favorables para aplicarlos a una mejora del modelo final, con esto teniendo una muy baja función de pérdida y un alto porcentaje en la tasa de aprendizaje.

Para concluir las pruebas realizadas observamos que la prueba numero 5 con un total de 10 épocas fue la que mejor tuvo resultados para nuestro modelo final puesto que su función de pérdida fue la mas baja y su tasa de aprendizaje fue considerablemente buena.

Gráficas de tasas de aprendizaje por cada prueba.

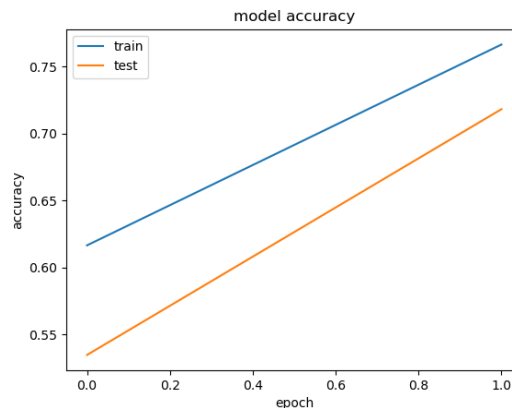


Figura 30 Gráfica de aprendizaje con 2 épocas. Fuente propia.

En la grafica de la prueba 1 [Figura 30] observamos dos rectas que son casi que paralelas, en esta vemos la línea de entrenamiento y la línea de pruebas una vez entrenado el modelo, la línea azul de nuestro entrenamiento se nota como va mejorando desde un punto de vista número, si bien parte con otro valor se nota como mejora, la línea naranja de las pruebas parte de 0 y se observa que tanto se acerca al entrenamiento, es decir que tan bien aprendió el modelo a dar predicciones.

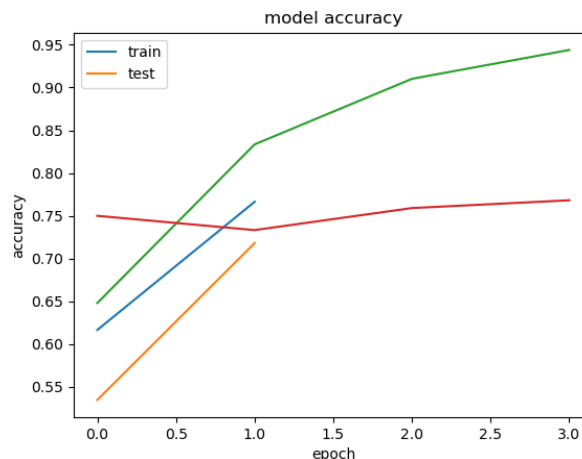


Figura 31 Gráfica de aprendizaje con 4 épocas. Fuente propia.

En la segunda prueba nos dio como resultado la grafica que nos muestra como va aprendiendo tanto en pruebas [Figura 31] como en entrenamiento teniendo en cuenta la cantidad de épocas que fue de 4, que tanto mejoró, cada línea representa cada iteración por época, vemos como en las primeras épocas fue pobre el aprendizaje, pero a medida que avanzaba los resultados iban mejorando.

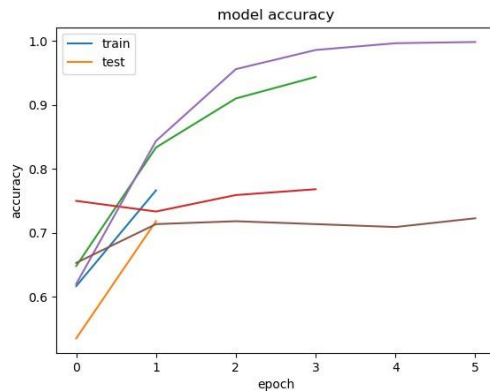


Figura 32 Gráfica de aprendizaje con 6 épocas. Fuente propia.

Como se mencionó anteriormente las líneas representan cada iteración de las épocas en su etapa de aprendizaje, en esta gráfica [Figura 32] se puede observar como desde el primer ciclo hasta el ultimo es decir 6 que fueron las épocas, estos últimos quedan con la mayor tasa de aprendizaje con respecto a las anteriores.

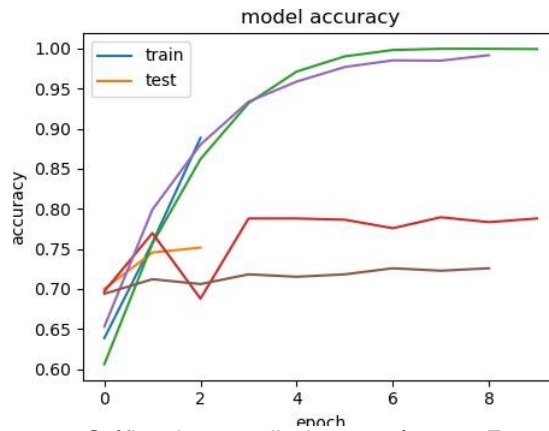


Figura 33 Gráfica de aprendizaje con 8 épocas. Fuente propia.

En esta gráfica [Figura 33] se puede observar cómo el último ciclo es decir 8 que fueron las épocas alcanzó la mayor cantidad en cuanto a los anteriores, estos últimos quedan con la mayor tasa de aprendizaje, en este caso los mejores ciclos fueron en el 7 y 8, si bien el 5 y 6 estuvieron cerca estos tuvieron mejores resultados.

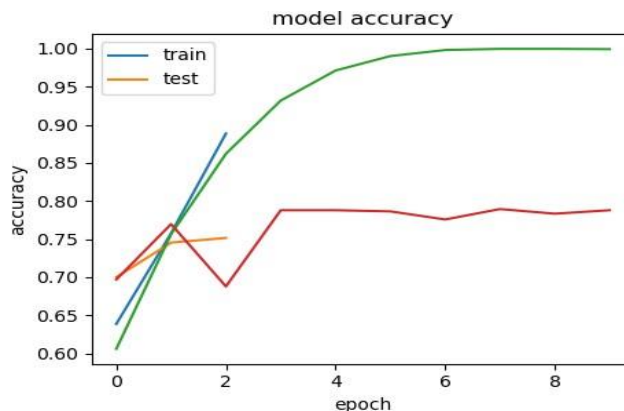


Figura 34 Gráfica de aprendizaje con 10 épocas. Fuente propia.

En esta gráfica [Figura 34] se puede observar cómo en los primeros ciclos la tasa de aprendizaje fue muy pobre, pero en los últimos dejó muy por detrás al resto los ciclos 9 y

10, el ciclo 10 que fue el ultimo tuvo el mejor resultado entre los demás con esto dar mejores predicciones para el modelo.

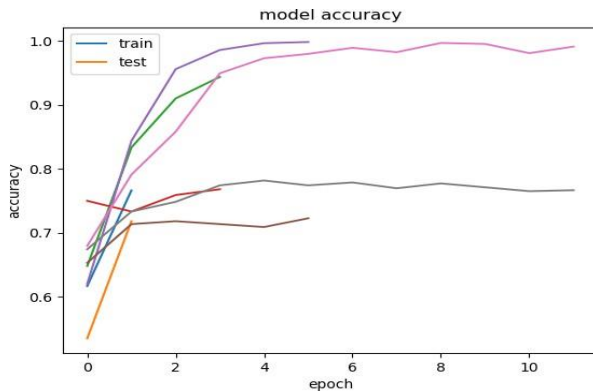


Figura 35 Gráfica de aprendizaje con 12 épocas. Fuente propia

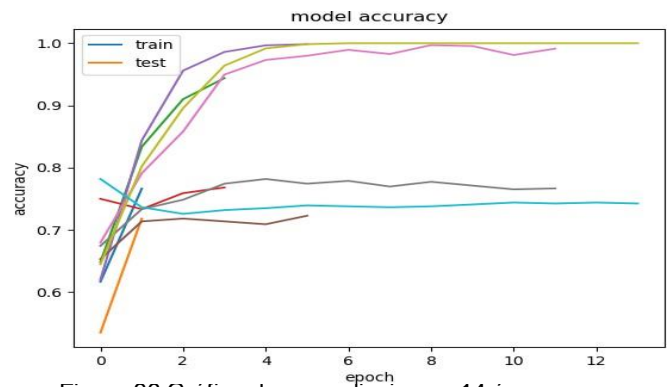


Figura 36 Gráfica de aprendizaje con 14 épocas. Fuente propia

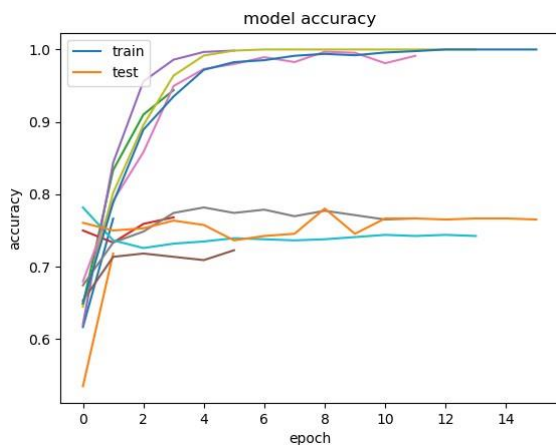


Figura 37 Gráfica de aprendizaje con 16 épocas. Fuente propia.

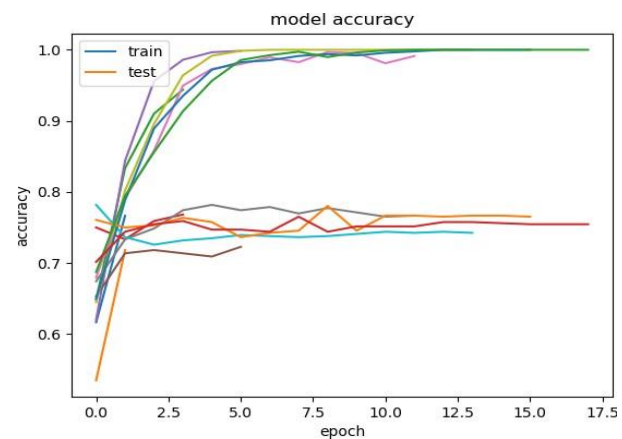


Figura 38 Gráfica de aprendizaje con 18 épocas. Fuente propia.

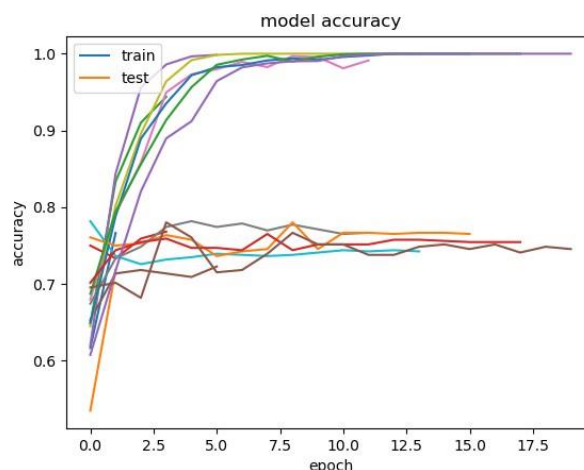


Figura 39 Gráfica de aprendizaje con 20 épocas. Fuente propia.

Para el resto de graficas ocurre exactamente lo mismo que en las primeras, observamos como varia en la cantidad de épocas la tasa de aprendizaje y como va mejorando o en unos casos empeorando ese porcentaje que es muy importante a la hora de dar

diagnósticos, cuando se trata de este tipo de eventos se pretende tener el mayor porcentaje de este puesto que son temas bastante delicados, por eso se opta por mejorar el modelo hasta tener mejores resultados y mejores pruebas con otros dataset o incluso hacer combinaciones y jugar con las capas de las redes para así poder alcanzar y sobrepasar las expectativas de este.

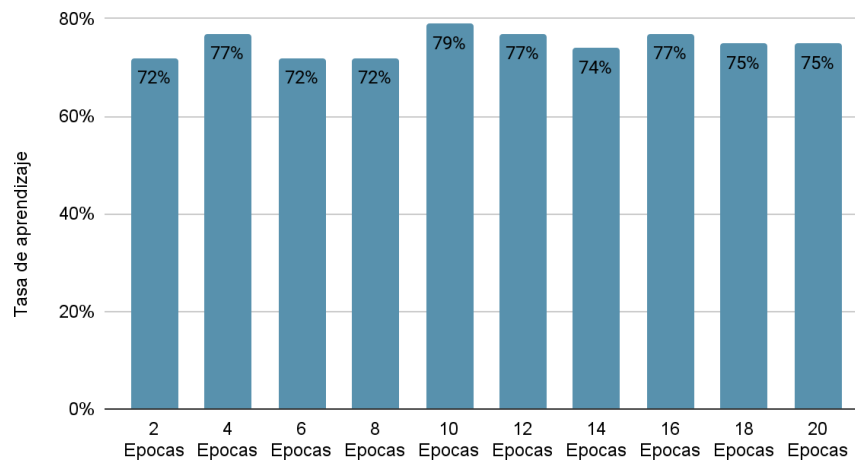


Figura 40: Gráfica de tasa de aprendizaje del modelo. Fuente propia

Analizando todos los resultados obtenidos, tanto de las pruebas como graficas se opto por sintetizar todos los datos y específicamente el hiperparámetro de la tasa de aprendizaje en una sola gráfica la cual muestra que tanto aprendió por época en porcentajes, esto sin tener en cuenta la función de pérdida, esto es simplemente para tener una forma mas sencilla y clara de que cantidad de épocas podrían ser mucho más útiles para futuras pruebas y ensayos, al observar la Figura 40 a simple vista podemos concluir que la que mejor tuvo desempeño fue en la cantidad de épocas con 10 con un 79% de probabilidad de acercarse al diagnóstico, cabe resaltar que los números fueron redondeados para llegar a estas cifras, en segundo lugar hay tres épocas que se mantuvieron a la par la 4, 12 y 16 y del resto de pruebas con épocas fueron resultados poco prometedores para nuestro modelo por esa razón, solo se harían mejoras en las pruebas con mayores resultados mencionadas anteriormente, con el mismo batch y épocas pero ya con capas distintas de neuronas, esto ya sería a futuro, al momento con el dataset que se tiene se lograron estos resultados pero para tener mejores que estos se necesita un mejor equipo de computo y mejorar el modelo y adaptarlo para que analice grandes cantidades de datos para así tener una mejor tasa de aprendizaje con una muy baja función de pérdida.

CONCLUSIONES

El objetivo principal del presente trabajo era elaborar un modelo que identifique imágenes de los melanomas dentro de un conjunto de imágenes con esto permitiendo mejorar la exactitud del diagnóstico de esta lesión. El estudio del estado del arte ha sido un punto clave, ya que un estudio sobre redes convolucionales y los hiperparámetros a entrenar han resultado muy útiles para la comprensión del trabajo.

Durante el desarrollo de este trabajo se logró el desarrollo de un modelo que permita la detección de melanomas a partir de imágenes de lesiones cutáneas mediante el uso de *Deep Learning*.

Para llegar a este resultado se realizó la búsqueda de *datasets* que cumplieran con los requisitos, de los cuales se realizó una división con el fin de generar 3 *dataset*, contando con un primero para el entrenamiento del modelo, un segundo para la validación del modelo y, por último, uno para realizar pruebas y así determinar si el resultado obtenido del modelo es satisfactorio con lo que se busca en este proyecto.

Dentro de las tecnologías de *Deep Learning* se encuentra el uso de la librería *OpenCV* la cual está especializada en el tratamiento de imágenes, por medio del uso de distintas herramientas que esta posee se realizó correctamente el procesamiento de las imágenes de los *datasets* mencionados anteriormente.

Además, se implementó la arquitectura VGG16 debido a que esta permite el uso de las redes neuronales convolucionales, la cual es una parte fundamental dentro de la construcción del modelo de predicción.

Una vez terminado el modelo, se procedió a realizar pruebas con el *dataset* proporcionado el cual hemos dividido de forma que dispusiéramos de dos conjuntos, uno de *train* y otro de test, por lo que el conjunto de entrenamiento ha quedado todavía más reducido.

Los porcentajes obtenidos de las pruebas, tanto en este trabajo como en la competición en general, pueden parecer insuficientes tratándose de un problema tan delicado como es la determinación de un tipo de cáncer en concreto, en el cual sería conveniente tener la máxima precisión posible para poder determinar una cura específica. Sin embargo, gracias al *Deep Learning* con esta clase de herramientas se está ofreciendo un sistema muy poderoso y útil a la hora de determinar el diagnóstico de este tipo de enfermedades, lo cual servirá de gran ayuda a los médicos especialistas a la hora de tomar decisiones finales, proporcionándoles más precisión y rapidez en los diagnósticos.

Teniendo en cuenta los resultados presentados en este trabajo se propone en mejorar el modelo para tener una mayor confianza en este, con un *dataset* mucho más robusto el cual ayude a que el entrenamiento sea más óptimo puesto que para este tipo de redes convolucionales entre más datos más preciso y confiable, otra posible mejor sería probar más capas al final, antes de la clasificación, de forma que vayan reduciendo el número de neuronas en cada capa de forma gradual en lugar de pasar por ejemplo de 80000 a 7. Otro posible trabajo futuro de cara a ver cómo se están comportando los datos, podría ser calcular por separado para cada una de las capas, de forma que se pueda estudiar cuál falla y cuál funciona correctamente, y finalmente usar el *dataset* con mayor cantidad de imágenes a procesar para así tener mejores resultados.

BIBLIOGRAFÍA

- [1] *¿Qué es el cáncer?* (s. f.). Sociedad Americana Contra el Cáncer. <https://www.cancer.org/content/cancer/es/tratamiento/como-comprender-su-diagnostico/que-es-el-cancer.html>
- [2] *Cáncer de piel en Colombia: cifras del Instituto Nacional de Cancerología.* (s. f.). Asociación Colombiana de Dermatología y Cirugía Dermatológica. <https://revistasocolderma.org/articulo-revista/cancer-de-piel-en-colombia-cifras-del-instituto-nacional-de-cancerologia>
- [3] *Instituto Nacional de Cancerología.* (s. f.). Instituto Nacional de Cancerología. <https://www.cancer.gov.co/>
- [4] *Cáncer de piel: para 2025 el número de muertes por melanoma aumentará un 20%.* (s. f.). Infobae. <https://www.infobae.com/salud/2020/06/13/cancer-de-piel-para-2025-el-numero-de-muertes-por-melanoma-aumentara-un-20/>
- [5] Zhang, N., Cai, Y. X., Wang, Y. Y., Tian, Y. T., Wang, X. L., & Badami, B. (2020). Skin cancer diagnosis based on optimized convolutional neural network. *Artificial Intelligence in Medicine, 102*. <https://scihub.se/10.1016/j.artmed.2019.101756>
- [6] *Declaración de Helsinki de la Amm – Principios éticos para las investigaciones médicas en seres humanos.* (s. f.). Asociación Médica Mundial. <https://www.wma.net/es/polices-post/declaracion-de-helsinki-de-la-amm-principios-eticos-para-las-investigaciones-medicas-en-seres-humanos/>
- [7] *Declaración de Helsinki de la asociación médica mundial.* (s. f.). Facultad de medicina Clínica alemana Universidad del desarrollo. https://medicina.udd.cl/centro-bioetica/files/2010/10/declaracion_helsinki.pdf

- [8] Acosta, Á. E. (2009). *Melanoma: patogénesis, clínica e histopatología* | *Revista de la Asociación Colombiana de Dermatología y Cirugía Dermatológica*. La Asociación Colombiana de Dermatología y Cirugía Dermatológica.
<https://revista.asocolderma.org.co/index.php/asocolderma/article/view/156>
- [9] *Cáncer de piel - Melanoma síntomas*. (s. f.). Asociación española contra el cáncer. <https://www.contraelcancer.es/es/todo-sobre-cancer/tipos-cancer/cancer-piel/melanoma/sintomas>
- [10] *Melanoma - SEOM: Sociedad Española de Oncología Médica* © 2019. (2020, 2 marzo). Sociedad Española de Oncología Médica. <https://seom.org/info-sobre-el-cancer/melanoma>
- [11] Ian Goodfellow, Yoshua Bengio, & Aaron Courville. (2017, 29 octubre). *Deep Learning*. Research Gate.
https://www.researchgate.net/publication/320703571_Ian_Goodfellow_Yoshua_Bengio_and_Aaron_Courville_Deep_learning_The_MIT_Press_2016_800_pp_ISBN_0262035618
- [12] Wu, H. (2009). Global stability analysis of a general class of discontinuous neural networks with linear growth activation functions. *Information Sciences*, 179(19), 3432–3441. <https://doi.org/10.1016/j.ins.2009.06.006>
- [13] Arora, J. S. (2006). Jan A. Snyman, Practical Mathematical Optimization: An introduction to basic optimization theory and classical and new gradient-based algorithms. *Structural and Multidisciplinary Optimization*, 31(3), 249.
<https://doi.org/10.1007/s00158-005-0595-0>
- [14] Team, D. S. (2021, 28 marzo). *Función de activación Relu*. DATA SCIENCE.
<https://datascience.eu/es/aprendizaje-automatico/funcion-de-activacion-relu/>

- [15] *Bringing Parallelism to the Web with River Trail*. (s. f.). River Trail. <http://intellabs.github.io/RiverTrail/tutorial/>
- [16] *Python*. (s. f.). Platzi. <https://platzi.com/blog/python/>
- [17] Gracia, L. (2013, 9 octubre). *¿Qué es OpenCV? Un poco de Java*. <https://unpocodejava.com/2013/10/09/que-es-opencv/>
- [18] *TensorFlow*. (s. f.). TensorFlow. <https://www.tensorflow.org/?hl=es-419>
- [19] *scikit-image: Image processing in Python — scikit-image*. (s. f.). <https://Scikit-Image.Org>. <https://scikit-image.org>
- [20] *Scikit-Learn, herramienta básica para el Data Science en Python*. (2019, 8 abril). Universidad de Alcalá. <https://www.master-data-scientist.com/scikit-learn-data-science/>
- [21] Alberca, A. S. (2020b, octubre 4). *La librería Numpy*. Aprende con Alf. <https://aprendeconalf.es/docencia/python/manual/numpy/>
- [22] Alberca, A. S. (2020a, octubre 4). *La librería Matplotlib*. Aprende con Alf. <https://aprendeconalf.es/docencia/python/manual/matplotlib/>
- [23] *SciPy*. (s. f.). SciPy. <https://scipy.org>
- [24] R. (2015, 6 junio). *La regla ABCDE de los lunares ayuda a la detección de un cáncer de piel*. Hoy. <https://www.hoy.es/sociedad/salud/vida-sana/201506/02/regla-abcde-lunares-ayuda-20150602184554-rc.html>
- [25] *Rendimiento de 4 metodos algorítmicos dermatoscopicos realizados por medicos no expertos*. (s. f.). IntraMed. <https://www.intramed.net/contenidover.asp?contenidoid=36479>
- [26] Valverde, J. M. (2018, 27 octubre). *Otsu's method, Python implementation — Lipman's Artificial Intelligence Directory*. Lipman's Artificial Intelligence Directory. <http://laid.delanover.com/otsus-method-python-implementation/>

- [27] *Vision artificial con redes convolucionales (CNN)*. (2019, 21 octubre). El laberinto de Falken. <https://www.ellaberintodefalken.com/2019/10/vision-artificial-redes-convolucionales-CNN.html>
- [28] *Computer Vision y Machine Learning con Python 2021*. (s. f.). Udeemy. <https://www.udemy.com/course/computer-vision-y-machine-learning-con-python/>
- [29] Hernández-Pimentel, F. (2010, 10 20). *Estadísticas del cancer*. A propósito del cáncer. https://www.scielo.sa.cr/scielo.php?script=sci_arttext&pid=S1409-14292010000200001
- [30] Bagnato, J. I. (2018, November 29). *Convolutional Neural Networks: La Teoría explicada en Español*. Aprende Machine Learning. Retrieved November 19, 2021, from <https://www.aprendemachinlearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>
- [31] Hernández-Pimentel, F. (2010, 10 20). *Estadísticas del cancer*. A propósito del cáncer. https://www.scielo.sa.cr/scielo.php?script=sci_arttext&pid=S1409-14292010000200001
- [32] Bagnato, J. I. (2018, November 29). *Convolutional Neural Networks: La Teoría explicada en Español*. Aprende Machine Learning. Retrieved November 19, 2021, from <https://www.aprendemachinlearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>
- [33] Calvo, D. (2018, December 7). *Función de activación - Redes neuronales*. Diego Calvo. Retrieved November 19, 2021, from <https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/>