ECyT
UNSAM

PROYECTO FINAL INTEGRADOR - INGENIERÍA BIOMÉDICA

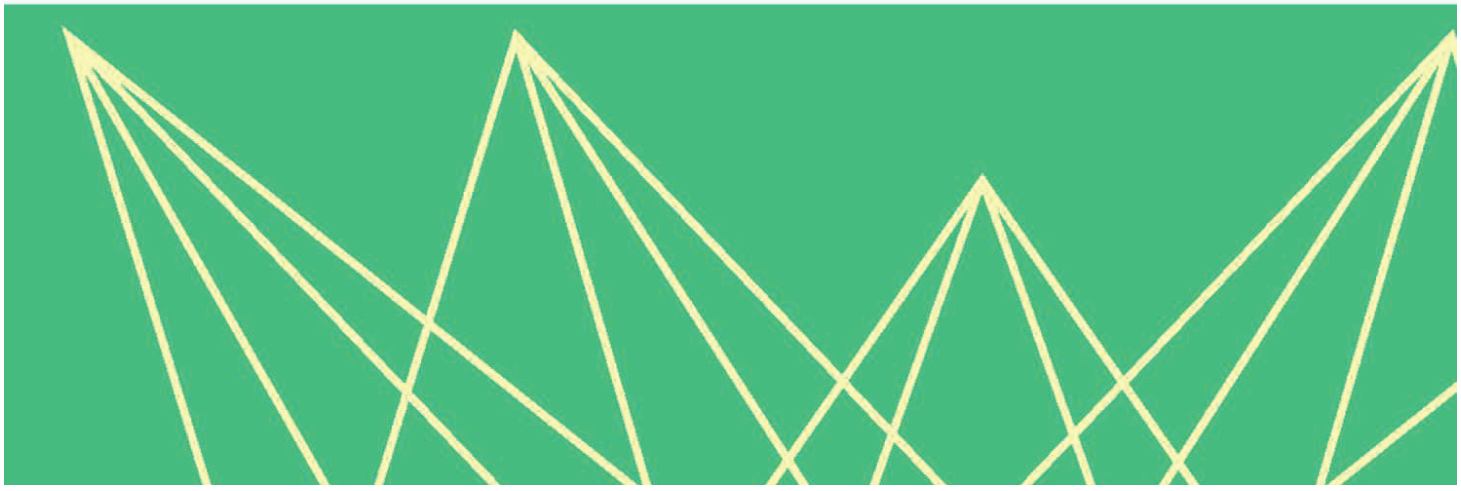
Herramienta de reconocimiento y clasificación de lesiones cutáneas basada en redes neuronales convolucionales para asistir en el diagnóstico de cáncer de piel tipo melanoma

ESTUDIANTE: Alexandra Binder
LEGAJO: CYT-8153

SUPERVISOR: Dr. Bioing. Marcelo Raponi

LUGAR DE TRABAJO: Escuela de Ciencia y Tecnología, UNSAM

FECHA: noviembre 2021



"Más vale prevenir que curar"
– Proverbio

"Siempre parece imposible hasta que se hace"
– Nelson Mandela

Agradecimientos

A mi familia, por haber estado presente todos estos años, brindándome las herramientas para seguir estudiando. Gracias por ayudarme a ser quien soy hoy.

A Facu por estar siempre a mi lado, brindándome palabras de aliento, escuchándome y dándome la contención que necesitaba.

A Gianni por aguantarme, básicamente.

A Herno por estar siempre presente.

A mis amigos y compañeros de la facultad con quienes disfruté de esta travesía.

A Marcelo por su predisposición y acompañamiento. Gracias por ser mi guía en este proyecto y por haberme introducido al mundo del DL, al cual hoy en día apunto a dedicarme.

A Agustina por su soporte y paciencia a lo largo de este proceso. Gracias por el tiempo que le dedicaste y por el apoyo moral que me diste. No te das una idea lo mucho que me ayudaste.

Al Dr. Escalada por haberme regalado su tiempo para escuchar sobre mi proyecto y darme sugerencias para mejorarlo.

A la universidad por abrirme sus puertas y por haberme brindado todas las herramientas adquiridas a lo largo de estos años.

Por último, gracias a todos quienes me hayan ayudado con la realización de este proyecto y a quienes hayan hecho posible navegar estos años con una sonrisa en la cara.

Índice

Resumen

La incidencia del cáncer de piel aumenta cada año. Este puede ser clasificado en dos clases: melanoma (mayor índice de mortalidad) y no melanoma (mayor incidencia). El melanoma tiene un comportamiento muy agresivo, llevando a la muerte en un plazo de varios meses después de su diagnóstico, cuando éste es tardío. La detección temprana es fundamental, ya que la tasa estimada de supervivencia a 5 años para el melanoma disminuye desde un 99% (si se detecta en sus primeras etapas) hasta aproximadamente 14% (si se detecta en sus últimas etapas).

La inspección visual (imagen clínica al ojo desnudo) y la dermatoscopia son herramientas fundamentales para la sospecha diagnóstica de cáncer de piel que eventualmente llevaría al profesional solicitar una biopsia de piel para la confirmación del diagnóstico por análisis histopatológico. Así es como estas dos herramientas clínicas determinan el umbral para solicitar una biopsia. Las lesiones de piel asociadas al cáncer tipo melanoma suelen ser visualmente muy similares a las lesiones benignas, lo cual dificulta la detección y el diagnóstico. Por esta razón, el análisis clínico depende en gran medida de la experiencia del profesional de salud, haciendo que la evaluación sea subjetiva.

Actualmente se están desarrollando herramientas computacionales basadas en *Deep Learning* (DL), que permiten realizar un seguimiento sistemático de la evolución de las lesiones cutáneas y detectar de manera temprana aquellas que son potencialmente malignas. El DL es una forma de Inteligencia Artificial (IA) que busca realizar predicciones derivadas del análisis de un amplio conjunto de datos. Su base de aplicación son las redes neuronales convolucionales (CNN), las cuales han revolucionado el concepto de detección de patrones en imágenes médicas. Estas redes combinan múltiples capas de aprendizaje especializadas y jerarquizadas con filtros sofisticados que forman redes neuronales multicapas para la detección de formas complejas. Las CNNs brindarían a los especialistas una nueva herramienta para facilitar los diagnósticos, proporcionando una probabilidad diagnóstica. No obstante, el éxito de la IA depende de la colaboración del dermatólogo en el desarrollo de algoritmos adecuados y del acceso a grandes bases de datos que permitan generalizar los resultados. En la práctica, se requieren más criterios que el morfológico para la toma de decisiones médicas, como por ejemplo los datos clínicos del paciente.

En este trabajo se desarrolló una herramienta de asistencia diagnóstica, basada en procesamiento de imágenes dermatológicas e inteligencia artificial que asiste al dermatólogo en el diagnóstico de cáncer de piel tipo melanoma. A la misma se accede mediante una aplicación, fácil de utilizar, donde el profesional puede gestionar los datos clínicos del paciente y generar reportes de análisis. De esta forma, el médico dermatólogo no solo tiene a disposición una herramienta que le brinda una segunda opinión en aquellos casos donde la lesión sea dificultosa de diagnosticar, sino también una herramienta que le permite al paciente disminuir los casos de extracción innecesaria de lesiones.

El algoritmo de clasificación binaria suministrado a la herramienta analiza y clasifica las lesiones cutáneas melanocíticas proporcionando una probabilidad diagnóstica de pertenencia a la clase "benigna" o "maligna". Este fue desarrollado para el sistema operativo Windows, empleando lenguaje *open-source* (*Python*) y herramientas específicas del área de la IA (*TensorFlow* y *Keras*). Para entrenar la CNN se utilizaron imágenes dermatoscópicas digitales obtenidas de un repositorio de imágenes internacional libre denominado ISIC. Se obtuvieron métricas comparables al desempeño de dermatólogos profesionales y a aquel de otros modelos citados en la bibliografía. Las métricas AUC obtenidas fueron 0.88 para la ROC AUC y 0.90 para la ROC PR. Por otro lado, se calcularon los valores de exhaustividad (TPR), especificidad (TNR) y exactitud, siendo estos 91.15%, 61.98% y 77.20% respectivamente.

Por otra parte, la aplicación se diseñó y desarrolló utilizando lenguaje y herramientas open-source: *Qt* y *Qt Designer* para el diseño de la interfaz de usuario, Python para atribuirle su funcionalidad y *SQLite* para su conexión con bases de datos. Para mejorar la usabilidad de la aplicación se entrevistó a un médico dermatólogo. De esta manera, se logró desarrollar una herramienta de asistencia al diagnóstico capaz de realizar un seguimiento sistemático y objetivo de la evolución de las lesiones cutáneas para detectar de manera temprana aquellas potencialmente malignas.

Glosario

- ANN: Red Neuronal Artificial
- AUC: Área bajo la curva
- Batch size: tamaño de lote
- BCC: Carcinoma basocelular
- Callback: Función interna del programa
- Checkpoint: época o punto específico de entrenamiento del modelo
- CNN: Red Neuronal Convolucional
- Colab: Google Colaboratory
- CPU: Unidad Central de Procesamiento
- Data augmentation: Aumentación de datos
- Dataset: Conjunto o sub-conjunto
- DL: Deep Learning o aprendizaje profundo
- Epoch: época
- Fine-tuning: Ajuste fino
- GPU: Unidad de Procesamiento de Gráficos
- Ground-truth: etiquetas verdaderas
- GUI: Graphical User Interface o Interfaz Gráfica de Usuario
- IA: Inteligencia Artificial
- IDE: Entorno de Desarrollo Integrado
- ISIC: Archivo llamado "The International Skin Imaging Collaboration"
- Kernel: Filtro
- Learning rate: tasa de aprendizaje
- ML: Machine Learning o aprendizaje automático
- Over-fitting: Sobreajuste
- PR: Precision-Recall o precision-exhaustividad
- ReLu: Función de activación llamada Unidad Lineal Rectificada
- Rescale: Cambio de escala
- ROC: Receiver Operating Characteristic o Características Operativas del Receptor
- SCC: Carcinoma epidermoide
- Transfer Learning: Transferencia de Aprendizaje
- Under-fitting: Ajuste insuficiente
- UV: Rayos ultravioletas

1. Introducción

El informe está estructurado en seis capítulos. El primero es la introducción donde se describe principalmente el marco contextual médico del proyecto. Es decir, se hace hincapié en la patología de interés y en los métodos y criterios que los médicos dermatólogos emplean en su diagnóstico. Por otra parte, se introducen diferentes trabajos de aprendizaje profundo o *Deep Learning* (DL) aplicado a la dermatología donde se describe brevemente qué es el DL y qué son las redes neuronales artificiales (ANN). La introducción de este proyecto solo describe brevemente el DL y las ANN ya que el capítulo 3 profundiza sobre los conceptos referidos a este ámbito. En el capítulo 2 se exponen los objetivos del proyecto para luego presentar en el capítulo 3 y en el capítulo 4 el desarrollo del algoritmo de clasificación y el de la aplicación, respectivamente. El capítulo 5 corresponde a la discusión del proyecto realizado y finalmente el capítulo 6 cierra el proyecto con una conclusión.

1.1. Marco contextual dermatológico

La piel es el órgano más extenso del cuerpo humano. Cubre la superficie externa del cuerpo cumpliendo la función de barrera de protección del medio externo; protege contra agresiones físicas (por ejemplo, la abrasión, la pérdida de agua, la radiación ultravioleta, la invasión por agentes extraños), y constituye la primera línea de defensa contra los microorganismos. Además, cumple la función de regular la temperatura corporal, proporcionar información sensitiva del medio circundante, excretar y absorber sustancias y sintetizar vitamina D [1].

La piel está dividida en tres capas principales: la *epidermis*, la *dermis* y el *tejido celular subcutáneo*. La porción superficial, denominada *epidermis*, es una capa fina compuesta por tejido epitelial que puede dividirse en cuatro estratos: el córneo, el granular, el espinoso y el basal [2]. La epidermis consiste en epitelio pavimentoso estratificado queratinizado y contiene cuatro tipos principales de células: queratinocitos, melanocitos, células de Langerhans y células de Merkel (Figura 1). La *dermis*, por otro lado, es la parte más profunda y gruesa de tejido conectivo. Debajo de la dermis se encuentra el tejido subcutáneo, cuyo elemento constitutivo básico es el adipocito, y sus principales funciones son la protección contra traumatismos, la conservación del calor corporal y la reserva energética. Las fibras que se extienden desde la dermis fijan la piel al

tejido subcutáneo, el cual a su vez se adhiere a la fascia subyacente, que está compuesta por tejido conectivo que rodea los músculos y los huesos [1].

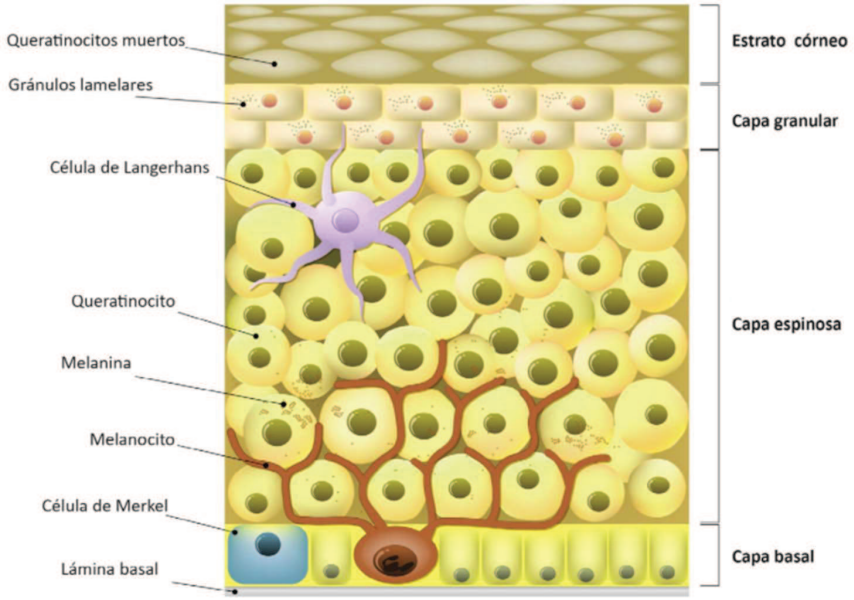


FIGURA 1: CORTE TRANSVERSAL DE LA PIEL Y LAS CAPAS DE LA EPIDERMIS. EXTRAÍDO DE [2]

Los queratinocitos conforman el 90% de las células epidérmicas. Ellos se encuentran distribuidos en cuatro o cinco capas en el estrato espinoso y producen una proteína denominada queratina. Esta es una proteína fibrosa y resistente que ayuda con la protección de la piel y los tejidos subyacentes de las abrasiones, el calor, los microorganismos y los compuestos químicos. A su vez, estas células producen gránulos lamelares que liberan sustancias que disminuyen la permeabilidad de sustancias hidrofílicas como el agua (evita la pérdida y entrada de agua por la piel) y el ingreso de cuerpos extraños [1].

Los melanocitos se encuentran en la capa basal y conforman el 8% de las células epidérmicas [2]. Estas células son las principales responsables de la producción de un pigmento llamado *melanina*. Dicho pigmento de color amarillo-rojizo o pardo-negruzco contribuye a otorgarle color a la piel y absorbe los rayos ultravioletas (UV) nocivos. Los melanocitos son células dendríticas cuyas proyecciones se extienden entre los queratinocitos y les transfieren gránulos de melanina. Una vez dentro de ellos, los gránulos se agrupan para formar un velo protector sobre el núcleo, hacia la superficie de la piel, proporcionando una protección del daño de la luz UV al ADN nuclear. Sin embargo, los melanocitos propiamente dichos son altamente susceptibles al daño por radiación UV [1].

Las células de Langerhans, o células dendríticas epidérmicas, se encuentran en el estrato espinoso y conforman una pequeña fracción de las células epidérmicas [2]. Estas células participan en la respuesta inmunitaria contra los microorganismos que invaden la piel, ayudando a otras células del sistema mencionado a reconocer microorganismos invasores y destruirlos. Además, son muy sensibles a la luz UV [1].

Por último, las células de Merkel son las más escasas de la epidermis. Se encuentran en la capa más profunda de la epidermis, en el estrato basal, donde entran en contacto con prolongaciones de las neuronas sensitivas denominadas discos táctiles de Merkel. Al ser éstas células nerviosas, junto con las células de Merkel, perciben las sensaciones táctiles [1].

1.1.1. Melanocitos, melanina y daño solar

Los melanocitos usualmente se encuentran en abundancia en la epidermis del pene, los pezones y las aréolas mamarias, la cara, los miembros y en algunas membranas mucosas. Los diferentes colores de la piel son una consecuencia de la cantidad de pigmento producido y transferido por los melanocitos hacia los queratinocitos [1], [2].

Según la predisposición genética de la persona, la melanina puede acumularse en parches llamados efélides o pecas. Estas pueden tener un color rojizo o marrón y tienden a ser más visibles en el verano que en el invierno. Las máculas seniles también son resultado de la acumulación de la melanina. Estas son imperfecciones aplanadas parecidas a las pecas, pero varían del color pardo al negro. Es decir, son más oscuras que las efélides y son más frecuentes en adultos mayores a cuarenta años. Los lunares o nevos suelen desarrollarse en condiciones normales en la niñez o en la adolescencia y con áreas circulares, planas o elevadas, formadas por una proliferación benigna y localizada de melanocitos [1].

La síntesis de melanina se realiza, en un organela denominado melanosoma, a partir del aminoácido tirosina en presencia de la enzima tirosinasa. La exposición a la luz UV incrementa la actividad enzimática de los melanosomas y consecuentemente también la producción de melanina. Tanto la cantidad como la oscuridad de la melanina aumentan por la exposición a los rayos UV. El pigmento absorbe la radiación UV para prevenir el daño del ADN de las células epidérmicas y neutraliza los radicales libres generados en la piel debido a dicha exposición [1].

A pesar de la función protectora de la melanina, y de la necesidad de exponer la piel al sol para la síntesis de vitamina D, la sobreexposición de la piel a la radiación UV puede causar cáncer

de piel [1]. Según la Agencia Internacional de Investigación sobre Cáncer (IARC), en Argentina se atribuye el 52,8% de los casos de cáncer tipo melanoma a la exposición a rayos UV [3].

Existen dos tipos de radiación UV que afectan la salud de la piel: los rayos ultravioletas A (UVA) y los rayos ultravioletas B (UVB). Los rayos ultravioletas C (UVC) tienen más energía que los otros tipos de radiación UV, pero como no penetran la atmósfera, no se encuentran la luz solar; razón por la cual no son los que normalmente causan cáncer de piel [4].

La radiación UVA poseen una longitud de onda más larga (entre 315 y 400 nm) y constituyen casi el 95% de la radiación UV que alcanza la superficie terrestre [1], [5]. Al no ser retenida por la capa de ozono de la atmósfera, sus rayos penetran la piel donde son absorbidos por los melanocitos. De esta manera la radiación UVA interviene en el bronceado de la piel [1]. A pesar de ser la forma menos dañina de la radiación UV, debido a que presenta menos energía que la radiación UVB, la exposición sostenida a la misma produce, en el largo plazo, daño en las células de la piel y son la causa fundamental de la inmunosupresión. Este último se debe a la producción de radicales libres que rompen el colágeno y las fibras elásticas de la piel, acelerando la aparición de arrugas, el envejecimiento de la piel [1], [6].

La radiación UVB posee una longitud de onda más corta (entre 280 y 315 nm). Al ser parcialmente absorbidos por la capa de ozono constituyen el otro 5% de radiación UV que alcanza la superficie terrestre. Esta radiación penetra la piel de manera más profunda por lo que es responsable de la generación de quemaduras solares, desarrollo de cataratas y posibles daños celulares de carácter degenerativo. Debido a sus niveles de energía, la radiación UVB puede romper los enlaces de las moléculas del ácido desoxirribonucleico (ADN), las cuales son portadoras moleculares de nuestro codificador genético [1], [6].

En conclusión, la sobreexposición a largo plazo a los rayos UVA y UVB ocasiona diferentes tipos de daños y efectos en la piel, entre ellas el cáncer de piel [1].

1.1.2. Cáncer de piel

Según la Sociedad Americana del Cáncer, esta enfermedad se caracteriza por el desarrollo de células anormales, que se dividen, crecen y diseminan sin control en cualquier parte del cuerpo. Mientras que las células normales se dividen y mueren durante un periodo de tiempo programado (apoptosis) [7]. El cáncer de piel puede aparecer en cualquier individuo, tanto en hombres como mujeres de cualquier edad, e independientemente de su color de piel [8].

La incidencia del cáncer de piel aumenta cada año, particularmente en personas con ascendencia europea y de etnia caucásica [10]. A nivel mundial, se estima que en el 2018 hubo aproximadamente 290.000 casos nuevos de melanoma, donde 61.000 fallecieron por esta causa [10]. La incidencia del melanoma reportada a nivel mundial varía ampliamente según regiones. Por ejemplo, las incidencias de melanoma reportadas en Asia e India rondan entre 0,2 y 0,3 cada 100.000 personas por año; mientras que Nueva Zelanda arroja valores de hasta 55 cada 100.000 personas por año. Los datos locales se asemejan a las reportadas en Europa y Canadá. Se estima que la tasa incidente ajustada a la población de Buenos Aires es de 13.4 por casa 100.000 personas por año [11]–[13]. En Argentina, según datos del Ministerio de Salud y Desarrollo Social (informe emitido en 2015), fallecen cada año unas 550 personas debido al cáncer de piel tipo melanoma [14]. Debido a que aproximadamente el 90% de los cánceres de piel son ocasionados por la sobreexposición de la piel a la radiación solar UV [15], es importante generar conciencia colectiva respecto al riesgo de estar expuesto a dicha radiación, y la necesidad de realizarse los controles dermatológicos anuales correspondientes [16].

Para prevenir el cáncer de piel es importante conocer cuáles son los factores de riesgo más comunes para desarrollarlo [1]. Estos se encuentran enumerados a continuación:

- Fototipo: las personas con fototipos más claros y sensibles, quienes generalmente sufren quemaduras solares antes de llegar a broncearse, corren mayor riesgo de contraer esta enfermedad [1].
- Exposición al sol: las personas con mayor exposición solar (ya sea por su ubicación geográfica o por las actividades que desarrolla) y aquellas que viven en lugares a grandes altitudes (donde la radiación UV es más intensa) poseen mayor riesgo de desarrollar cáncer de piel [1]. La exposición solar en la infancia y en la adolescencia cumplen un rol fundamental en el desarrollo de esta patología [17]. A su vez, aquellas personas con antecedentes de quemaduras solares también son más propensas a contraer la enfermedad [1].
- Antecedentes familiares: como suele ocurrir con varias otras enfermedades, la propensión genética también entra en juego cuando se habla del riesgo de desarrollar cáncer de piel. Algunos genes asociados con la pigmentación o con los nevos, junto con los genes de reparación del ADN (y otros), aumentan el riesgo de melanoma

hereditario. Alrededor del 5 al 10% de los pacientes con melanoma cutáneo tienen antecedentes familiares con la enfermedad [1], [10]

- Edad: las personas mayores que tuvieron una gran exposición solar de jóvenes son más propensas al cáncer de piel debido al daño acumulativo de la radiación solar [1].
- Estado inmunológico: las personas inmunodeficientes tienen mayor incidencia de contraer cáncer de piel [1].

El cáncer de piel puede clasificarse en dos grandes grupos: cáncer de tipo melanoma y cáncer de tipo no melanoma [18]. Esta patología representa un desafío particular al momento de estimar su incidencia por varias razones. En primer lugar, existen múltiples subtipos de cáncer de piel, lo cual puede prestar a dificultades al momento de recopilar datos. Por ejemplo, los registros de cáncer no suelen rastrear el cáncer de piel no melanoma, o estos suelen estar incompletos porque la mayoría de los casos se tratan con éxito mediante cirugía, ablación u otros procedimientos y fármacos. Además, varios casos de cáncer no son identificados ni registrados. Esto puede deberse a la falta de registros de cáncer en algunos países o en algunas regiones específicas de un país. Por otro lado, algunas personas con cáncer pueden no haber consultado a un médico. Debido a estos factores, la incidencia mundial notificada de cáncer de piel se encuentra subestimada. Además, la clase "no melanoma" también suele omitirse en las clasificaciones comparativas de los cánceres más comunes [19].

A pesar de las dificultades en la recopilación de información de la enfermedad, se estima que en el año 2018 el cáncer de tipo melanoma conformó el 22% de los diagnósticos de cáncer de piel; mientras que los tumores cancerígenos de tipo no melanoma respondieron al 78% [18]. Estas estadísticas se ven afectadas tanto por la presencia de los diferentes tipos de cáncer, como por la dificultad al momento de diagnosticar las lesiones debido a su gran variabilidad de manifestaciones y su amplia similitud mediante inspección visual entre las dos variantes [9]. En la Figura 2 puede observarse una gráfica que denota una clasificación generalizada de esta patología [18].

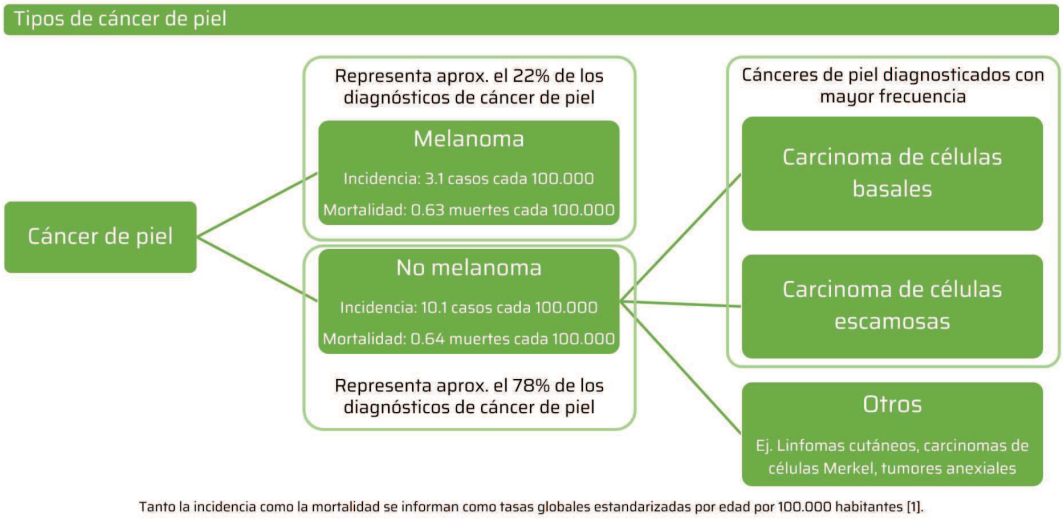


FIGURA 2: CLASIFICACIÓN DE CÁNCER DE PIEL. ADAPTADO DE [18]

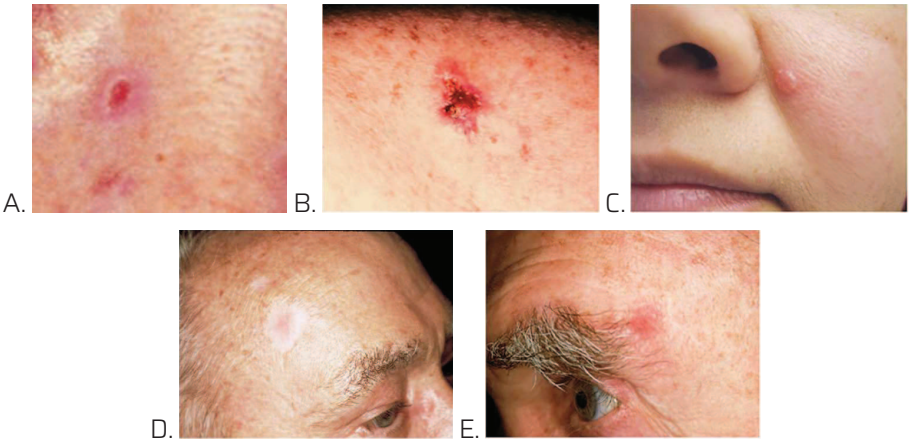


Figura 3: Subtipos del carcinoma basocelular: A) Pápula eritematosa con bordes sobreelevados y erosión central. B) Ulceración que no cicatriza. C) Pápula eritematosa perlada. D) Área cicactrizal de coloración blanquecina – rosada. E) Mácula eritematosa con erosión central. ADAPTADO DE [20]

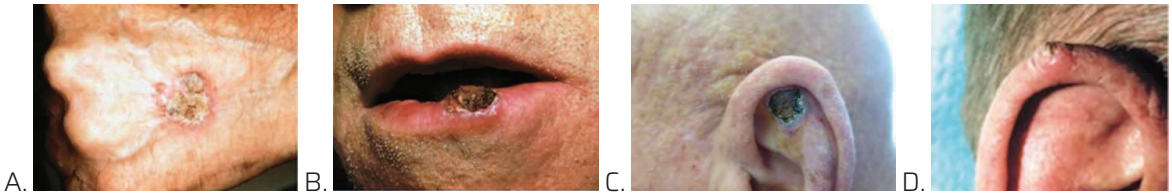


FIGURA 4: SUBTIPOS DEL CARCINOMA EPIDERMOIDE: A) ULCERACIÓN DE BORDES IRREGULARES CON FONDO FIBRINOSO. B) TUMORACIÓN CON ULCERACIÓN CENTRAL DE RÁPIDO CRECIMIENTO. C) ULCERACIÓN DE FONDO FIBRINOSO, QUE A VECES SANGRA, DE TÓRPIDA EVOLUCIÓN. D) PÁPULA ERITEMATOSA CON COSTRA CENTRAL Y SANGRADO OCASIONAL. ADAPTADO DE [20]

El cáncer de piel tipo no melanoma posee una alta incidencia en la población mundial, pero su tasa de mortalidad es menor a la del cáncer de piel tipo melanoma [17]. A pesar de esto, el no melanoma impone una pesada carga financiera a los sistemas de salud, debido a su frecuencia y los costos concomitantes del diagnóstico y la cirugía [10]. Esta patología puede a su vez subdividirse en tres grandes grupos: carcinoma basocelular, carcinoma epidermoide o de células escamosas y otros cánceres de menor incidencia tales como los linfomas cutáneos, carcinomas de células Merkel y tumores anexiales [18].

El carcinoma basocelular (BCC) comienza en las células basales de la piel, encontradas en la parte interior de la epidermis. Estos tumores crecen de manera lenta y metastatizan ocasionalmente, por lo que su probabilidad de letalidad es escasa [10], [21]. La presentación más común suele ser una pápula o placa rosada y perlada en la piel expuesta al sol (Figura 3) [21]. El BCC suele desarrollarse 80% de la veces en sitios de piel expuestos al sol, particularmente en el área de la cabeza, cuello, hombros, pecho y nariz [17]. Este tipo de cáncer se encuentra asociado principalmente con la exposición intermitente a altas dosis de radiación solar en comparación con dosis similares administradas de forma más continua; y suele desarrollarse a edades más tempranas que el carcinoma de células escamosas [10], [17], [21].

El carcinoma epidermoide (SCC) comienza en las células escamosas de la piel, encontradas en la parte exterior de la epidermis. Estos tumores crecen rápidamente por lo que poseen una alta probabilidad de hacer metástasis [17]. Generalmente se presenta como una pápula escamosa, placa o nódulo en la piel [21]. Surgen en sitios habitualmente expuestos al sol, particularmente en la cara, orejas, cuello y superficies expuestas de las extremidades (Figura 4) [10]. A su vez, se presentan en edades mayores que los BCC. El efecto de la edad y la exposición continua a los rayos solares es mucho más fuerte para los SCC que para los BCC. En consecuencia, la proporción de BCC a SCC cambia rápidamente, de aproximadamente 10:1 a los 40 años a aproximadamente 3:1 a los 60 años [1], [10], [17], [21].

Por último, el cáncer de tipo melanoma es aquel que surge de una mutación en los melanocitos en la epidermis [22]. Este tipo de cáncer de piel es el menos común pero, es el más peligroso ya que posee una alta tasa de mortalidad [10], [17], [22]. La razón se debe a que tiene la capacidad de hacer metástasis con mayor rapidez que el cáncer de tipo no melanoma. Aproximadamente un 95% de los melanomas son tumores cutáneos que surgen en las superficies de la piel expuestas al sol (Figura 5), pero también pueden aparecer en varias otras

partes de la piel, ya sean superficies expuestas a los rayos solares o no. Por ejemplo, en la piel de las palmas de las manos y de las plantas de los pies, en los ojos, meninges, membranas mucosas, tractos gastrointestinales e incluso tractos genitales [1], [10], [17].

El caso más común de cáncer tipo melanoma es el de extensión superficial, que suele ser plano e irregular, tanto en forma como en color. El melanoma nodular suele observarse como un bulto o área elevada de color oscuro (azul, negro) que, en general, presenta un crecimiento vertical que tiende a la diseminación hacia otras regiones del organismo. En personas de edad avanzada puede detectarse el melanoma lentigo maligno, plano y de color marrón, producto del daño ocasionado por la exposición al sol. Por último, el menos común es el melanoma lentiginoso acral, que aparece en las plantas de los pies y debajo de las uñas. También existe una variedad que carece de pigmento, denominado melanoma amelanótico. La gran variabilidad de manifestaciones de esta lesión y su amplia similitud mediante inspección visual a lesiones benignas, hace que el diagnóstico sea permeable a confusiones [9].

Se prevé que en los próximos 20 años, el número de nuevos casos de melanoma aumente a más de 450.000 casos incidentes por año a nivel mundial [18]. Según la IARC la proporción de casos de cáncer tipo melanoma en Argentina en personas mayores a 30 años, atribuibles a la exposición a la radiación UV, se encuentra entre el 49.5 y el 76.9% [23]. Es por esta razón que no solo es importante tomar conciencia colectiva respecto al riesgo de estar expuesto a dicha radiación y realizarse los controles dermatológicos anuales correspondientes, sino también generar y adoptar nuevas tecnologías que faciliten el diagnóstico de esta patología [16].

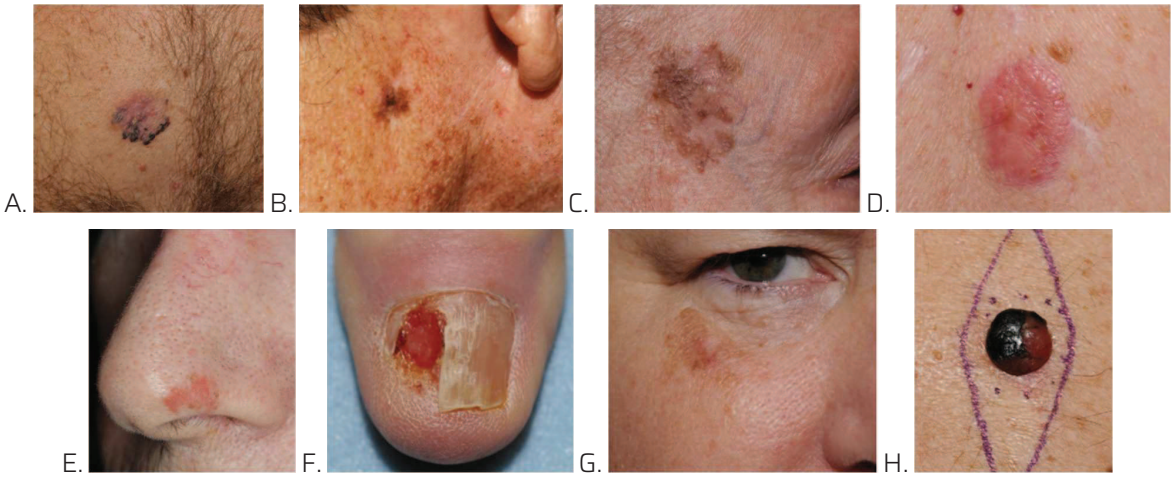


FIGURA 5: SUBTIPOS DE MELANOMA: A) PLACA ERITEMATOSA DE LÍMITES IRREGULARES E HIPERPIGMENTACIÓN EN LA PERIFERIA. B) MÁCULA DE COLORACIÓN AMARRONADA CON ACENTUACIÓN DEL PIGMENTO A NIVEL CENTRAL. C) MÁCULA IRREGULAR HIPERPIGMENTADA. D) PÁPULA ERITEMATOSA OVALADA. E) PÁPULA ERITEMATOSA DE LÍMITES IRREGULARES. F) TUMORACIÓN ERITEMATOSA SUBUNGUEAL. G) MÁCULA HIPERPIGMENTADA EN REGIÓN INFRAORBITARIA DERECHA. H) TUMORACIÓN REDONDEADA DE COLORACIÓN ROJIZA-NEGRUZCA. EXTRAÍDO DE [21].

1.1.3. Diagnóstico de cáncer de piel

La detección temprana del cáncer tipo melanoma es fundamental: la tasa estimada de supervivencia a 5 años para el melanoma disminuye desde un 99% (si se detecta en sus primeras etapas) hasta aproximadamente 14% (si se detecta en sus últimas etapas) [17]. Es por esta razón que es imprescindible consultar con un médico dermatólogo al menos una vez al año y controlar el estado de la piel y sus lesiones.

A la hora de diagnosticar lesiones cutáneas malignas los médicos dermatólogos emplean diferentes métodos, siendo el examen físico (inspección visual aislada, la dermatoscopía y, eventualmente, la biopsia de piel (para la confirmación del diagnóstico por análisis histopatológico), las herramientas más utilizadas [21], [24]. Al ser las lesiones dermatológicas malignas y benignas visualmente similares, su detección y diagnóstico se torna dificultosa [25], dependiendo en gran medida de la experiencia del profesional de salud, haciendo de esta evaluación subjetiva [26].

La herramienta primordial que asiste al médico en la detección de una lesión maligna es la biopsia. Este es el único método que define el diagnóstico [24], lo cual implica tiempo, dinero y una cicatriz que en muchos casos no era necesaria [25], [27]. Otras herramientas existentes contribuyen a que el médico especialista sea más objetivo en su trabajo de detección visual y lo

asisten a la hora de tomar la decisión de realizar la extracción de la lesión [28] pero, no obstante, siguen dependiendo de la interpretación de la lesión por parte del especialista.

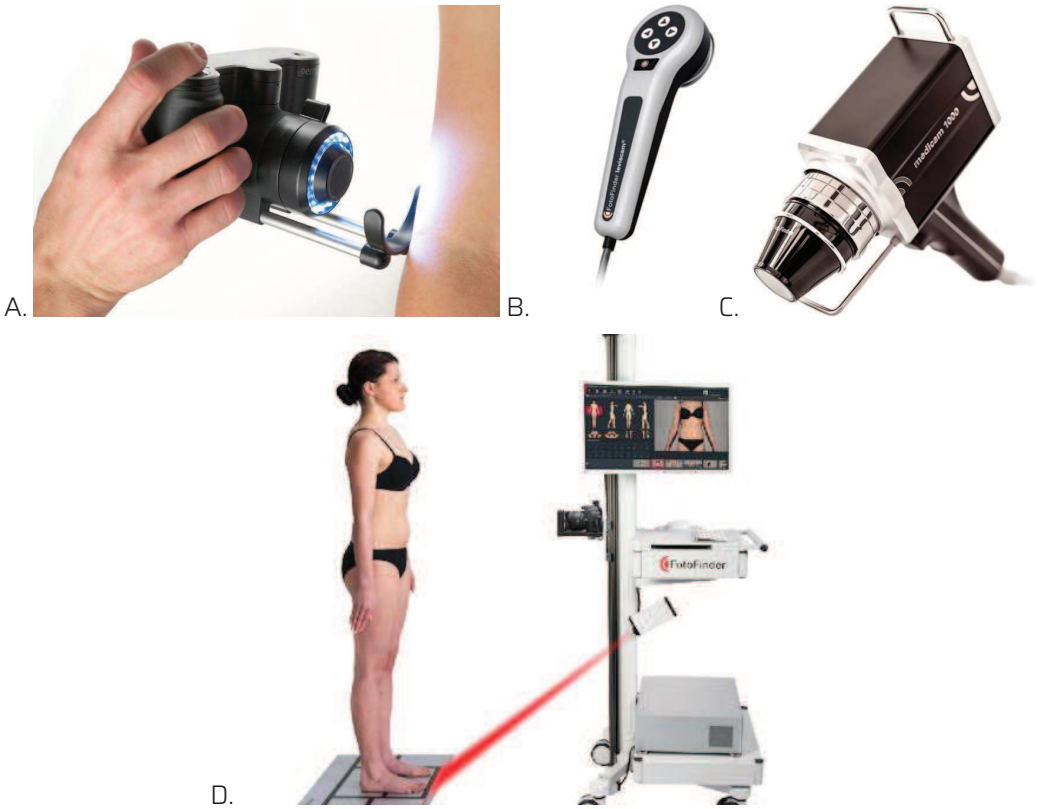


FIGURA 6: ADAPTACIONES DE DERMATOSCOPIOS: A) MARCA: DERMLITE, MODELO: FOTO SYSTEM [29]; B) MARCA: FOTOFINDER, MODELO: LEVIACAM [30]; C) MARCA: FOTOFINDER, MODELO: MEDICAM 1000 [30]; D) MARCA: FOTOFINDER, MODELO: BODYSTUDIO [31]

Por lo general, existe una aproximación estructurada y metodológica para realizar un diagnóstico dermatológico. En una primera instancia se intuye la etiología de la lesión combinando el análisis de la historia clínica del paciente, junto a la realización de un examen físico para identificar lesiones y patrones morfológicos y topográficos. Posteriormente se realizan estudios complementarios para confirmar o descartar el diagnóstico inicial. Estos pueden incluir diascopía[1], exámenes que iluminan las lesiones con diversas fuentes lumínicas, estudios de laboratorio, exámenes epicutáneos, dermatoscopia digital, microscopia confocal y biopsias cutáneas [9].

[1] Técnica utilizada en dermatología donde el médico aprieta una lesión cutánea con una lámina de cristal ligeramente abombada con el objetivo de expulsar la sangre de la zona y visualizar el comportamiento de la piel. De esta manera el dermatólogo puede determinar si la lesión es debida a una dilatación de los vasos o si se trata de una lesión que no desaparece al ceder la presión [32].

La dermatoscopía es una de las técnicas mayormente utilizadas debido a su sencillez, eficacia y rapidez [33]. Esta es una técnica diagnóstica no invasiva para el estudio de las lesiones cutáneas (Figura 6.A. y Figura 6.B.) [34]. Para su realización se utiliza un dermatoscopio, un dispositivo manual que cuenta con un sistema de amplificación de imagen de hasta 140X [28] y un sistema de iluminación mediante luz polarizada para eliminar la distorsión producto de la reflexión y refracción de la luz en la piel [9]. Mediante esta técnica se logra que el estrato córneo de la piel se torne traslúcido, permitiendo la visualización detallada de la estructura de la dermis, la epidermis y de patrones de vascularización y de pigmento, lo cual agrega exactitud diagnóstica respecto de la inspección visual aislada [34].

Actualmente, la adquisición de imágenes dermatoscópicas se puede realizar de forma digital (Figura 6.C. y Figura 6.D.). Esta nueva técnica se denomina videodermatoscopía, "mapeo de lunares" o DIAR-D. El estudio consiste en hacer un paneo videográfico de cuerpo entero para conocer la ubicación exacta de cada lesión y posteriormente obtener imágenes de cada una de ellas para poder analizarlas con mayor detenimiento. Al permitir el registro fotográfico de las lesiones, posibilita la realización de estudios longitudinales, que permiten evaluar la evolución de una lesión durante un período definido de tiempo. El estudio puede realizarse a cualquier edad, no es invasivo y no requiere preparación previa [34][35].

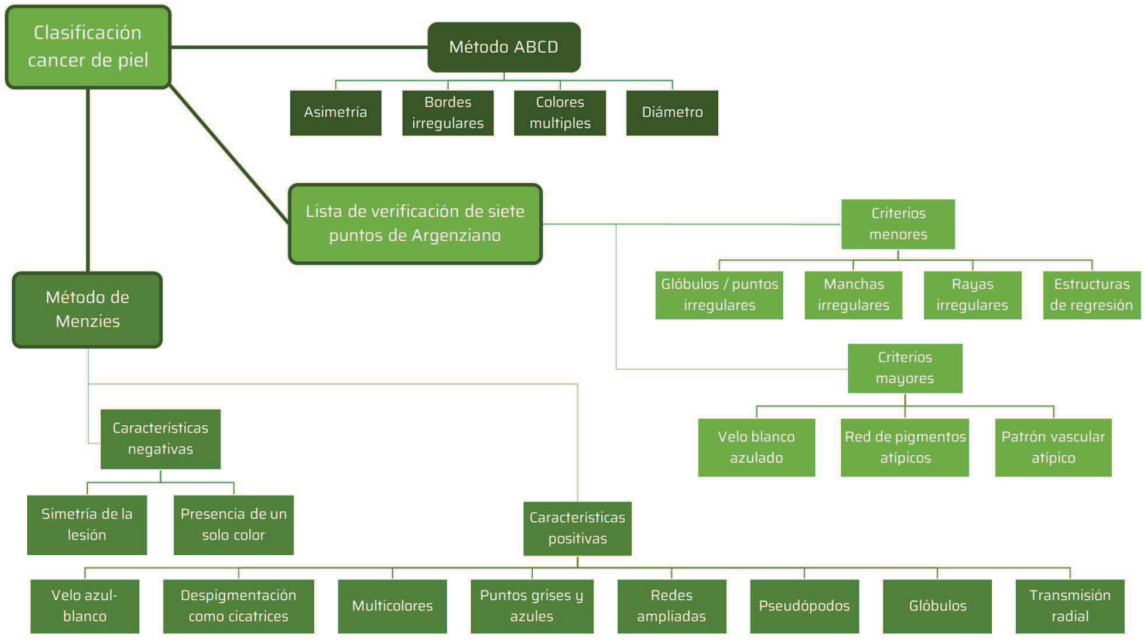


FIGURA 7: MÉTODOS DE DIAGNÓSTICO DE CÁNCER DE PIEL COMÚNMENTE UTILIZADOS. EXTRAÍDO DE [36].

Con la observación en mayor detalle y el reconocimiento de nuevos patrones que permite la dermatoscopía, se definen criterios específicos para la identificación de las distintas lesiones cutáneas. En particular, para el diagnóstico del cáncer tipo melanoma se utilizan con mayor frecuencia los siguientes algoritmos diagnósticos: el método ABCD de Stolz, la lista de verificación de siete puntos de Argenziano y el método de Menzies, entre otros. [36]. La Figura 7 muestra un esquema de los diferentes métodos comúnmente utilizados para diagnosticar esta patología.

La más utilizada es una versión simplificada del método ABCD. Esta no implica un método de asignación de puntajes a las características de una lesión. Además, suele enseñarse a la población con el propósito de concientizar en la detección precoz del melanoma. De esta forma, la persona logra realizar un chequeo personal complementario a la evaluación profesional a partir de la inspección visual aislada de la lesión. Este criterio se denomina ABCDE: (A) asimetría, (B) bordes irregulares, (C) colores múltiples, (D) diámetro mayor de 6 mm y (E) evolución en términos de cambios en los síntomas y el aspecto [9], cualquier cambio en el tamaño, forma, color o elevación de una mancha en la piel, o cualquier síntoma nuevo en ella, como sangrado, picazón o costras, puede ser una señal de advertencia de melanoma [37]. La Figura 8 permite visualizar los diferentes aspectos de la lesión analizados mediante el método ABCDE.

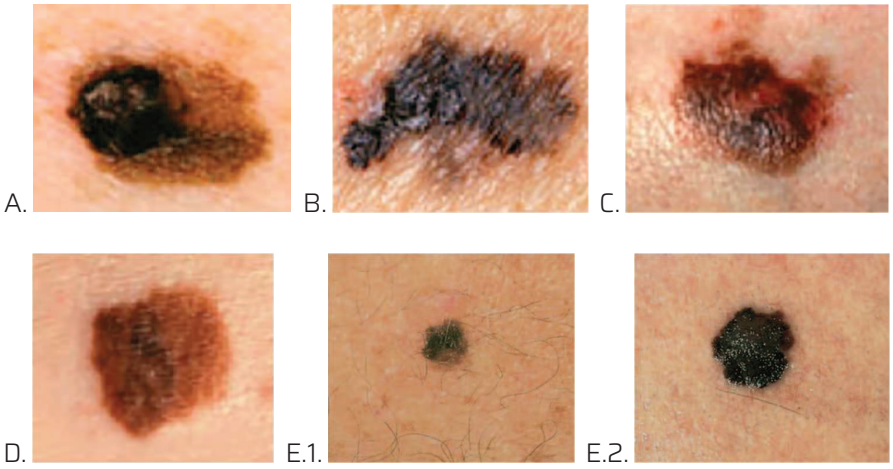


Figura 8: Ejemplos del método ABCDE: A) Asimetría; B) Bordes; C) Color; D) Diámetro; E.1) Evolución (antes); E.2) Evolución (después). Extraído de [37].

La lista de verificación de siete puntos de Argenziano consiste en puntuar una lesión según tres criterios mayores y según cuatro criterios menores (Tabla 1). Los criterios mayores tienen una puntuación máxima de 2 puntos, mientras que los criterios menores, de 1 punto cada

uno. Si el puntaje final es menor a 3 puntos, la lesión es considerada benigna; en caso de ser igual o superior a este valor, la lesión será calificada como maligna [9], [36].

Por último, el método de Menzies analiza si la lesión posee características positivas y/o características negativas. La presencia de cualquier negativo declara que la lesión es melanoma maligno. Sería benigno si ambos negativos están ausentes y uno o más positivos son verdaderos. Estos se resumen en la Tabla 2.

En los últimos años los nuevos avances tecnológicos permitieron aumentar la capacidad de procesamiento de las computadoras, impulsando la aplicación de visión computacional[2] mediante inteligencia artificial en el campo de las imágenes médicas; con el objetivo final de generar herramientas de asistencia y soporte para la toma de decisiones clínicas [9]. De esta manera surgieron nuevas herramientas de procesamiento de imágenes y de aprendizaje automático, como por ejemplo el Deep Learning, aplicadas al diagnóstico de diferentes patologías, incluyendo el diagnóstico de cáncer de piel tipo melanoma [38].

TABLA 1: CRITERIOS DE CLASIFICACIÓN PARA EL MÉTODO DE LA LISTA DE VERIFICACIÓN DE SIETE PUNTOS. EXTRAÍDO DE [36].

Criterios mayores	Criterios menores
– Velo blanco azulado: manchas azules con bruma blanca alrededor sin estructura definida. – Red de pigmentos atípicos: la lesión tiene una distribución asimétrica dentro de la red junto con líneas reticulares, mientras que el color y el grosor son de naturaleza heterogénea. – Patrón vascular atípico: vasos globulares o punteados irregulares que tienen linealidad.	– Glóbulos / puntos irregulares: los puntos tienen forma, color y distribución irregulares con un tamaño menor a 0.1 mm, mientras que el tamaño de los glóbulos es mayor a 0.1 mm. – Manchas irregulares: áreas de diferentes colores (blanco, negro o marrón) y no tienen una forma o regularidad determinada (sin distribución o estructura definida). – Rayas irregulares: cuando el melanoma comienza a crecer radialmente, forma un patrón de rayas radiales y seudópodos, que se encuentran en los bordes del área de la lesión. – Estructuras de regresión: cicatrices como la pigmentación, particularmente de color blanco.

[2] "La visión computacional tiene como propósito emular, mediante algoritmos y programas computacionales, la capacidad de los ojos humanos y del sistema visual, no solo para ver, sino también para interpretar el contenido en imágenes y video" [9].

Tabla 2: Método de Menzies. Extraído de [36].

Características positivas	Características negativas
– Velo azul-blanco – Despigmentación como cicatrices – Multicolores – Puntos grises y azules – Redes ampliadas – Pseudópodos – Glóbulos – Transmisión radial	– Simetría de la lesión – Presencia de un solo color

1.2. Deep Learning aplicado a la dermatología

El *Deep Learning* (DL), también conocido como aprendizaje profundo, es un área del *Machine Learning* (ML) o aprendizaje automático, el cual a su vez es un subcampo de la Inteligencia Artificial (IA). El objetivo principal de la IA es proporcionar un conjunto de algoritmos y técnicas que puedan ser utilizadas para resolver problemas que los humanos realizan de forma intuitiva y casi automática, pero que son desafiantes de resolver para las computadoras [38].

Por otro lado, el ML suele aplicarse para el reconocimiento de patrones y el aprendizaje a través de los datos. Para ello se emplean herramientas tales como las redes neuronales artificiales (ANN, por sus siglas en inglés), las cuales son una clase de algoritmos de ML inspirados en la estructura y funcionamiento del cerebro [38]. Las ANN intentan simular la percepción de los objetos conectando neuronas artificiales o nodos dentro de capas que podrían extraer características de los objetos [39].

Si bien esta herramienta se inspira en el cerebro humano y en cómo interactúan sus neuronas entre sí, las ANN no están destinadas a ser modelos realistas del cerebro. En cambio, son una inspiración, lo que permite establecer paralelismos entre un modelo muy básico del cerebro y cómo imitar parte de este comportamiento a través de ANN. Esto significa que cuando se aplica a la clasificación de imágenes, esta herramienta busca tomar un conjunto de imágenes e identificar patrones que puedan usarse para discriminar entre sí varias clases u objetos [38].

El campo del DL existe desde la década de los años noventa y pertenece a la familia de algoritmos de ANN. Los algoritmos de ANN clásicos utilizan funciones diseñadas a mano para cuantificar el contenido de una imagen; mientras que el enfoque mediante las redes neuronales convolucionales (CNN) es diferente, pudiendo realizar la extracción de características de una imagen de manera automática durante el proceso de entrenamiento de la red [38].

El DL, que se discutirá con mayor profundidad en el Capítulo 3, busca entender los problemas en términos de una jerarquía de conceptos. Cada concepto se construye sobre los demás. Aquellos en las capas de nivel inferior de la red, más cercanas a las entradas o *inputs* de la misma, codifican alguna representación básica del problema; mientras que las capas de nivel superior, más cercanas a las salidas u *outputs*, usan las capas básicas para formar conceptos más abstractos. Además, en los últimos años los avances tecnológicos facilitaron tener a disposición hardware especializado, más veloz y con una mayor cantidad de datos de

entrenamiento disponibles, permitiendo de esta manera entrenar redes con un gran número de capas ocultas, lo cual implica una mayor capacidad de aprendizaje jerárquico [38].

Debido a que el área de la dermatología suele realizar un diagnóstico utilizando una aproximación estructurada y metodológica, esta especialidad es un área médica que, por su propia naturaleza de búsqueda y reconocimiento de patrones en los datos, sea promisoria para la aplicación de herramientas de IA con propósito de facilitar el trabajo de los profesionales de salud y asistir al diagnóstico de cáncer de piel [9]. Además, como la interpretación y el análisis de los resultados de las imágenes médicas todavía dependen en gran medida tanto de la disponibilidad como de la experiencia de los expertos médicos, las herramientas de IA pueden definir un criterio objetivo al diagnóstico de cáncer de piel y acercar estas herramientas a lugares donde la disponibilidad de médicos especializados sea escasa [16].

Estas herramientas pueden facilitar el seguimiento sistemático de la evolución de las lesiones cutáneas y detectar de manera temprana aquellas que son potencialmente malignas. Por esta razón, el uso de algoritmos en la práctica clínica potencia la aplicación de la IA en la toma de decisiones médicas. De esta manera, las CNN brindarían a los especialistas una nueva herramienta para la realización de los diagnósticos [40], [41]. El diagnóstico de imágenes empleando métodos basados en CNN ha tenido bastante éxito. Esto se debe en gran parte al hecho de que las CNN han logrado un desempeño a nivel humano en tareas de clasificación de objetos, donde una CNN aprende a clasificar el objeto contenido en una imagen [42]–[44].

En el año 2017, un grupo de investigadores describieron la aplicación de las CNN para la evaluación clínica de lesiones cutáneas. En este trabajo la capacidad diagnóstica de 21 dermatólogos expertos se vio superada por la capacidad diagnóstica de una CNN, para la clasificación de lesiones de origen epidérmico y melanocítico (dermatólogos: 95% tumores malignos y 76% lesiones benignas; CNN: 96% y 90% respectivamente) [45]. En otro trabajo, un algoritmo de ML superó el rendimiento de la mayoría de los dermatólogos en la clasificación de 100 imágenes dermatoscópicas de nevus y melanoma [46]. Por otro lado, un trabajo multi-clase del 2019 concluyó que la combinación de hombre y máquina logró una exactitud del 82,95% y que este resultado fue un 1,36% más alto que el mejor de las dos partes individuales (81,59% logrado por la CNN) [47]. Otros trabajos llegaron a resultados de precisión del 74.2% ([48]) y otros a una sensibilidad del 86.4% y una especificidad del 85.5% [48]–[50]. Por último, otro trabajo del 2019 comparó el desempeño entre médicos con diferente experiencia profesional,

junto al algoritmo de clasificación desarrollado y concluyó que su CNN superó a 136 de los 157 dermatólogos en términos de especificidad y sensibilidad media (dermatólogos: 74.1% y 60% vs CNN: 86.5% y 74.1%) [51]. No obstante, el éxito de la IA depende de la colaboración del dermatólogo en el desarrollo de algoritmos adecuado, además del acceso a grandes bases de datos que permitan generalizar los resultados [50], [52]. La mayoría de estos sistemas no han sido evaluados en un entorno clínico real, donde se requieren más criterios que el morfológico para la toma de decisiones médicas, como por ejemplo los datos clínicos del paciente [45], [53] .

Un software basado en IA pensado para brindar al dermatólogo una herramienta de análisis de imágenes dermatológicas permitiría ahorrar tiempo y esfuerzo, además de disminuir los casos de extracción innecesaria de lesiones. Esta herramienta computacional debería analizar y clasificar las lesiones cutáneas, determinando la probabilidad que dicha lesión sea benigna o maligna. De esta forma, el médico dermatólogo tendría a disposición una herramienta que le brinde una segunda opinión en aquellos casos donde la lesión sea dificultosa de diagnosticar.

2. Objetivos

2.1. Objetivo general

Diseñar una herramienta basada en procesamiento de imágenes dermatológicas e inteligencia artificial que permita clasificar lesiones de piel melanocíticas benignas y malignas, proporcionando una probabilidad diagnóstica para asistir al personal médico en el diagnóstico de cáncer de piel tipo melanoma.

2.2. Objetivos específicos

La herramienta a implementar consta de dos etapas: el desarrollo de un modelo de inteligencia artificial y el diseño y desarrollo de una aplicación. A continuación, se enuncian los objetivos específicos del proyecto donde los primeros cuatro se asocian a la primera etapa y los últimos tres a la segunda.

- Generar un dataset propio a partir de repositorios de imágenes internacionales libres.
- Acondicionar las imágenes para utilizarlas como elementos de entrada a una CNN.
- Desarrollar una CNN para clasificar imágenes dermatológicas en lesiones cutáneas melanocíticas del tipo benigno y maligno.
- Entrenar y validar la red neuronal utilizando las imágenes preprocesadas.
- Diseñar y desarrollar una aplicación que permita al personal médico administrar la información del paciente, sus imágenes dermatológicas y sus estudios, además de utilizar la herramienta como asistente de diagnóstico.
- Evaluar el correcto funcionamiento del software mediante asistencia médica.
- Elaborar un manual de usuario para la utilización de la herramienta desarrollada.

3. Desarrollo del algoritmo de clasificación binario para lesiones cutáneas benignas y malignas

En este capítulo se describirá el desarrollo del algoritmo de clasificación utilizado en este proyecto con el objetivo de clasificar lesiones cutáneas benignas y malignas. El capítulo se organizó emulando el orden en el que se desarrolla el código del algoritmo, intercalando en cada paso la teoría correspondiente a los conceptos de ML y DL involucrados en cada uno de ellos.

El algoritmo fue desarrollado con la intención de desarrollar un software que permita al médico dermatólogo acceder a una herramienta de trabajo que lo asista en el proceso de decisión que conlleva el diagnóstico de cáncer de piel.

3.1. Entorno de programación

Existen diversos lenguajes de programación para desarrollar algoritmos de DL, entre ellos *Python* y *R* son los más conocidos. En el caso de este proyecto se optó por utilizar Python ya que es un lenguaje fácil de aprender debido a su sintaxis simple e intuitiva [38].

Al momento de programar algoritmos de DL, se emplean varias librerías de programación. Cada una de ellas especializada en distintos elementos utilizados en diferentes etapas del desarrollo del código tal como se observa en la Figura 9.

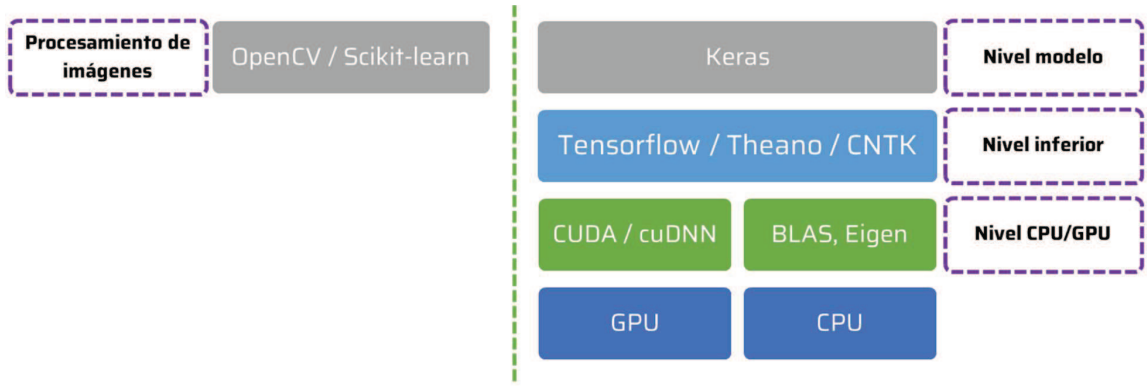


FIGURA 9: LIBRERÍAS DE PROGRAMACIÓN UTILIZADAS EN EL DESARROLLO DE ALGORITMOS DE DL. ADAPTADA DE [54].

Para el procesamiento de imágenes se utilizaron las librerías *OpenCV* y *Scikit-learn,* ambas de código abierto y compatibles con el lenguaje Python. La primera se utiliza para importar imágenes desde la computadora, mostrarlas en pantalla y realizar operaciones básicas de procesamiento de imágenes ya que permite representarlas como matrices *NumPy*, que también es una librería. Por su parte, la segunda se utiliza como complemento a OpenCV debido a que facilita la creación de las divisiones o *datasets* de entrenamiento / prueba / validación y validar la precisión de los modelos de DL [38].

En cuanto al desarrollo de las redes neuronales, *Keras*, es la librería utilizada por excelencia, la cual proporciona una manera ágil de definir y entrenar los modelos. Como se observa en la figura anterior, Keras trabaja a nivel modelo [38], [54]. Permite definir la arquitectura del mismo, especificando el tipo de entrada a recibir, la cantidad de capas, el optimizador a utilizar, entre otros parámetros que se detallarán más adelante en este capítulo.

Estrechamente vinculadas a Keras, otras librerías tales como *TensorFlow*, *Theano* y *CNTK* manejan operaciones de nivel inferior o de *backend*. Estas se encargan de realizar la manipulación y diferenciación de los datos de bajo nivel ya que se encuentran especializadas y optimizadas para realizar el procesamiento matemático que conlleva desarrollar las redes neuronales [38], [54]. A pesar de que Keras es completamente compatible con cualquiera de estos backends, TensorFlow es el más adoptado, debido a su escalabilidad [38], [54], razón por la cual se decidió utilizar TensorFlow como backend en este proyecto.

Para llevar a cabo la manipulación de datos, es necesario contar con un alto poder de procesamiento, el cual es proporcionado por la Unidad Central de Procesamiento (CPU) de la computadora. Generalmente tanto Keras como TensorFlow son instaladas en la CPU, pero de contar con una placa de video del tipo *NVIDIA* y librerías tales como *CUDA* y *cuDNN*, entonces es posible utilizar el poder de cómputo paralelo de la Unidad de Procesamiento de Gráficos (GPU) [38], [54]. Combinar el uso de una o más GPUs junto a la CPU de la computadora permite acelerar el funcionamiento de las aplicaciones de ML y DL. Esto se logra captando los aspectos de la aplicación donde la computación es más intensiva, mientras que el resto del código se ejecuta en la CPU; obteniéndose una aplicación de procesamiento más veloz a nivel usuario [55].

Por último, es necesario contar con un entorno de desarrollo integrado (IDE) para el diseño y desarrollo del código. Existen varios IDEs compatibles con Python, entre los cuales podemos mencionar a PyCharm, Google Colaboratory (Colab), Jupyter Notebook, Spyder, etc. Para el

desarrollo de este proyecto se optó por Colab ya que se encuentra preparado para escribir y ejecutar códigos de Python en el navegador. Además, es una herramienta adecuada para realizar tareas de ML y de análisis de datos, ya que cuenta con una facilidad que los otros IDEs carecen: ofrece acceso gratuito a recursos informáticos, como GPUs. A pesar de que este sistema sea de libre acceso, utilizar la GPU de Colab tiene sus limitaciones: funciona con un sistema de prioridades donde aquellas personas que hayan utilizado menos recursos recientemente, tendrán prioridad para acceder a las GPUs, de otra forma se restringirá temporalmente el acceso a las mismas. A su vez, Colab no permite seleccionar el tipo de GPU, sino que se asigna de manera aleatoria. Estas suelen ser los modelos Nvidia K80 (24 GB), T4 (16 GB), P4 (8 GB) y P100 (16 GB) y varían a lo largo del tiempo. Por otro lado, el ciclo de vida máximo de las máquinas virtuales donde se ejecutan los cuadernos es de 12 horas [56].

3.2. Redes neuronales artificiales

El cerebro humano puede describirse como una red neuronal biológica o como una red interconectada de neuronas que transmiten patrones elaborados de señales. En términos generales, el funcionamiento de las neuronas es el siguiente: las dendritas reciben señales de entrada y, basándose en esas entradas, disparan una señal de salida a través de un axón; posteriormente esta señal es captada por las dendritas de la neurona subsiguiente. Las redes neuronales artificiales son algoritmos computacionales basados en el cerebro humano que buscan resolver problemas “simples para los seres humanos, pero difíciles para las máquinas”, los cuales generalmente llevan el nombre de *reconocimiento de patrones* [57].

Una red neuronal es un sistema computacional basado en una red de nodos interconectados que procesan la información de forma colectiva y en paralelo, tal como se observa en la Figura 10 [57]. Las líneas que conectan a los nodos en la imagen representan las conexiones entre dos neuronas e indican el camino por donde fluye la información. Estas poseen pesos de conexión que pueden modificarse mediante un algoritmo de aprendizaje. Los pesos positivos amplifican la señal, indicando que esta es muy importante al realizar una clasificación, mientras que los pesos negativos la disminuyen, determinando de esta manera que la salida del nodo es menos importante en la clasificación final [38].

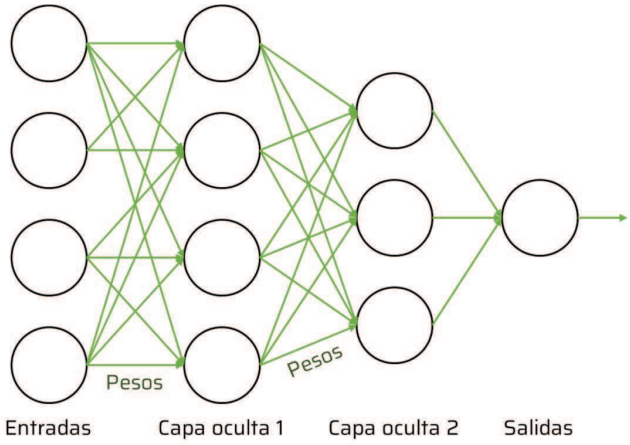


FIGURA 10: ARQUITECTURA DE RED NEURONAL SIMPLE. ADAPTADO DE [38].

3.2.1. El Perceptrón

La configuración más simple de una ANN se conoce como Perceptrón. Este es un modelo computacional de una sola neurona o nodo, donde se realiza una analogía paralela a una única neurona biológica. Esta responde como una simple puerta lógica con salidas binarias; múltiples señales llegan a las dendritas, luego se integran en el cuerpo celular y, si la señal acumulada excede un cierto umbral, se genera una señal de salida que será transmitida por el axón. Es decir, el modelo consiste de una o más entradas, un procesador y una única salida, tal como se observa en la Figura 11 [57], [58]. Un perceptrón sigue el modelo *feed forward*, lo que significa que las entradas se envían a la neurona, se procesan y dan como resultado una salida. Esto significa que la información de la red fluye de izquierda a derecha [57].

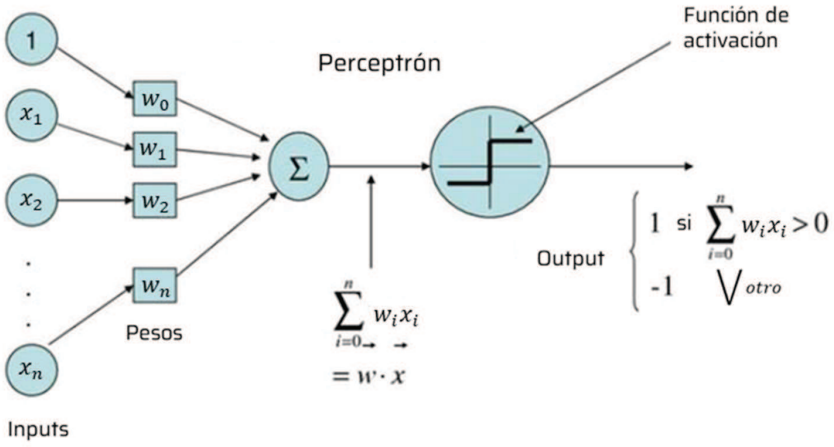


FIGURA 11: CONFIGURACIÓN DEL PERCEPTRÓN CLÁSICO. ADAPTADO DE [59].

En la Figura 11 se observa que el impulso de entrada x_i es multiplicado por un peso w_i que representa la estructura sináptica de la neurona. Posteriormente, se obtiene una combinación lineal de los valores de entrada x_i y su correspondiente peso w_i, denominada entrada neta z (ecuación 1). Tanto los pesos w_i como un elemento denominado sesgo (w_0 *o* b, por su palabra en inglés *bias*) son aprendidos en la fase de entrenamiento. Este último término se le suma a la multiplicación matricial previamente mencionada. La salida del modelo (ecuación 3) responde según la función de activación f (ecuación 2) elegida, la cual en el caso del Perceptrón es una función signo [57], [60].

$$z = b + w_1 x_1 + \cdots + w_n x_n \qquad \text{(ECUACIÓN 1)}$$

$$f(z) = \begin{cases} 1 & si\ z \geq 0 \\ -1 & \vee\ otro \end{cases} \qquad \text{(ECUACIÓN 2)}$$

$$y(x, w) = f\left(\sum_{i=1}^{N-1} x_i w_i + b\right) \qquad \text{(ECUACIÓN 3)}$$

En el contexto de una clasificación binaria donde se tienen dos clases (una "positiva" o "maligna" y otra "negativa" o "benigna") se le asocia a cada una de ellas una salida de la red diferente: 1 para la clase positiva y -1 para la clase negativa. Al ser la función de activación una función signo, de ser la entrada neta z de una muestra particular mayor a 0, se predice la clase 1, y de no ser así, se predice la clase 0 [58].

El término de sesgo b agregado a la entrada neta z permite discriminar aquellos casos donde no es claro a qué clase debería pertenecer cierto parámetro de entrada. Permite cambiar y trasladar la función de activación en una dirección u otra sin influir en los pesos asociados a cada entrada [38].

3.2.2. Funciones de activación

Cambiando la función de activación signo por otras, se mejora el rendimiento de la red. No obstante, la decisión de qué función utilizar depende del objetivo de cada ANN y de los datos que esta reconocerá. Las funciones de activación forman las capas no lineales en todos los modelos de DL; y sus combinaciones con otras capas se utilizan para simular la transformación no lineal de la entrada a la salida. Por lo tanto, se puede lograr una mejor extracción de características seleccionando las funciones de activación apropiadas [38].

La función de activación sigmoidea, también conocida como función logística, es no lineal por naturaleza. Su salida permanece siempre dentro del rango (0, 1) y suele ser útil en la última

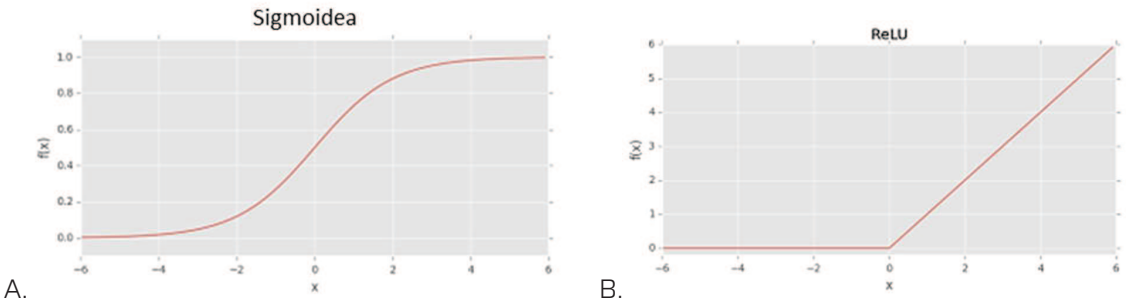
capa de salida cuando se busca calcular la probabilidad asociada a una clasificación binaria. Una ventaja que presenta esta función de activación ante la función signo, es que es menos sensible a observaciones individuales. La función sigmoidea se describe mediante la ecuación 4, donde a es la entrada de la capa de entrada [36], [39], [61].

$$f(a) = \frac{1}{1+e^{-a}} \quad \text{(ECUACIÓN 4)}$$

La función de activación unidad lineal rectificada, también conocida como ReLU, (ecuación 5) y sus variantes muestran un rendimiento superior en muchos casos y son las funciones de activación más populares en DL hasta ahora [39]. Su salida es a, si a es positiva, y si no, es 0. La ReLU es de naturaleza no lineal y su combinación también lo es, lo que significa que permite apilar diferentes capas. Su rango es de 0 a infinito, lo que puede saturar la activación [36]. Esta función de activación posee una desventaja que es que los pesos podrían actualizarse de cierta forma en la que la neurona nunca se active. Esto ocurre cuando la salida de la neurona es igual a 0. Una vez que se llega a esta condición durante el entrenamiento, no es posible revertirla. No obstante, como se mencionó anteriormente, posee un rendimiento mayor que otras funciones de activación debido a que no implica cálculos costosos en el entrenamiento [61].

$$f(a) = max\,(0, a) \quad \text{(ECUACIÓN 5)}$$

En este proyecto se utilizaron principalmente dos funciones de activación: la función sigmoidea y la ReLU. La primera fue utilizada en la capa de salida del modelo ya que es una práctica estándar cuando se trata de un algoritmo de clasificación binaria, es decir de dos clases. De haber sido una clasificación multi-clase, es decir, dos o más clases, la función de activación por defecto es la *Softmax*. Esta última también podría haber sido utilizada en este proyecto ya que te permite obtener a la salida un vector de predicciones con la misma cantidad de columnas como clases involucradas. En el caso de una función de activación sigmoide se obtiene a la salida un vector de una única columna con las predicciones de la clase positiva (en este caso "maligna") y para obtener la predicción de la clase negativa se calcula el complemento (predicción benigna = 1 – predicción maligna). La función ReLu por otra parte, fue utilizada en las capas ocultas de la red debido a que es la que se utiliza mayormente en otros trabajos [62], [63]. En la Figura 12 puede observarse la representación gráfica de las funciones de activación mencionadas:



A. B.

FIGURA 12: DESCRIPCIÓN GRÁFICA DE LAS FUNCIONES DE ACTIVACIÓN DE PASO A) SIGMOIDEA Y B) RELU. ADAPATADO DE [38].

3.2.3. Redes neuronales multi-capa y mecanismos de aprendizaje

Las redes neuronales se forman a partir de un ordenamiento secuencial de capas compuestas por nodos ordenados de manera paralela (Figura 13). Esta disposición, donde la salida de un conjunto de neuronas se da secuencialmente como entrada al siguiente conjunto de neuronas, permite resolver problemáticas más complejas que un único perceptrón no podría. En este caso, cada nodo puede enfocarse en una característica particular del objeto de entrada y no en la salida final. El resultado final se predecirá en función del resultado de estas características. De esta manera, las redes neuronales pueden tomar decisiones sofisticadas con técnicas básicas de aprendizaje automático [38].

Para poder resolver problemas no lineales, las redes tienen que ser capaces de *aprender* (mediante la realización de ajustes en su estructura a lo largo del tiempo). El algoritmo de retropropagación o de *back propagation* es el que permite entrenar redes neuronales de múltiples capas, tal como la que se observa en la Figura 13. Este algoritmo combinado con funciones de activación no lineales ha posibilitado la resolución de problemas más complejos y ha habilitado un área de investigación completamente nueva en las redes neuronales [38].

Existen varias estrategias para aprender, entre ellas se encuentran (1) el aprendizaje supervisado, (2) el aprendizaje no supervisado y por último (3) aprendizaje semi-supervisado o por refuerzo [57]. Como el algoritmo de este proyecto pertenece a la categoría de visión computacional[3], la estrategia de aprendizaje empleada es de aprendizaje supervisado. En este caso, el algoritmo recibe un conjunto de entradas, junto a sus etiquetas verdaderas o de *ground-*

[3] "La visión computacional o visión artificial (*computer vision*) es un conjunto de métodos y algoritmos que buscan adquirir, procesar, analizar y comprender las imágenes del mundo real, buscando simular la percepción y la comprensión visual humana" [9].

truth y luego "aprende" patrones que pueden usarse para *mapear* automáticamente los puntos de datos de entrada a su *ground-truth* correcto. La estrategia es que algoritmo intente clasificar correctamente el *input* basándose en los conocimientos adquiridos en su entrenamiento. De equivocarse, se modifican ciertos componentes en su arquitectura para lograr realizar una mejor predicción la próxima vez [38]. En otras palabras, al algoritmo se le entregan los datos de entrenamiento junto a sus valores de *ground-truth* correspondientes; este realiza sus conjeturas y luego las compara frente a los valores verdaderos. Una vez hecho esto realiza ajustes en sus pesos de acuerdo a sus errores [57].

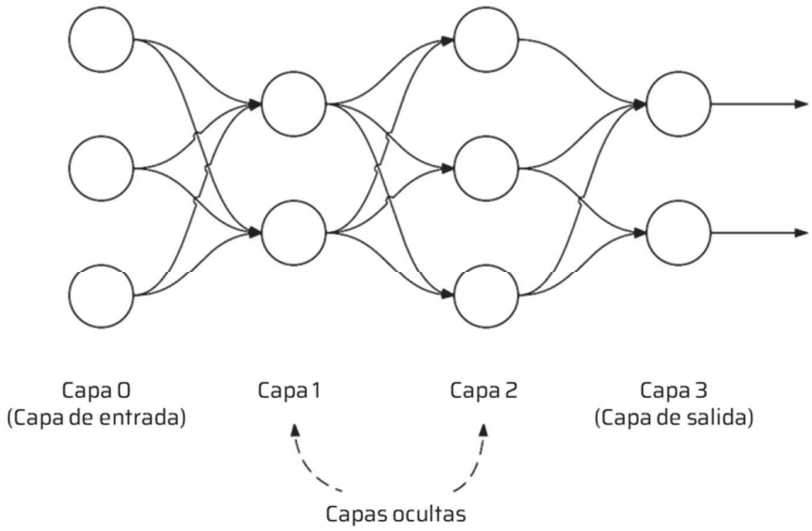


FIGURA 13: ARQUITECTURA DE RED MULTICAPA *FEED FORWARD* CON UNA CAPA DE ENTRADA (3 NODODS), DOS CAPAS OCULTAS (2 NODOS EN LA PRIMERA CAPA Y 3 NODOS EN LA SEGUNDA CAPA) Y UNA CAPA DE SALIDA (2 NODOS). EXTRAÍDO DE [38].

3.2.4. Redes neuronales convolucionales

Actualmente el aprendizaje profundo o DL se encuentra en auge debido a la disponibilidad tanto de hardware especializado (de mayor velocidad) como de datos de entrenamiento. Esto hace posible entrenar redes con varias capas ocultas que son capaces de aprender de manera jerárquica tomando conceptos simples de las capas inferiores para formar patrones más abstractos en las capas superiores de la red [38], [64].

El ejemplo por excelencia de DL aplicado al aprendizaje de características es la red neuronal convolucional (CNN, de su inglés *Convolutional Neural Network*). Este tipo de red aprende automáticamente patrones discriminatorios de imágenes (llamados filtros) al apilar capas secuencialmente una encima de la otra [38]. A diferencia de otras estructuras de DL, las

CNN extraen características de pequeñas porciones de imágenes de entrada, denominadas *campos receptivos*. Este tipo de extracción de características se inspiró en los mecanismos visuales en los organismos vivos, donde las células en la corteza visual son sensibles a pequeñas regiones del campo visual [39]. Como se mencionó en el Capítulo 1, en muchas aplicaciones las CNN actualmente se consideran el clasificador de imágenes más poderoso y actualmente son responsables de impulsar el estado del arte en los subcampos de visión computacional que aprovechan el aprendizaje automático [38].

Uno de los principales beneficios del DL y las CNN es que permiten omitir el paso de extracción de características, que para otros algoritmos de ML se realiza de manera manual, y permiten al desarrollador centrarse en el proceso de entrenamiento de la red para aprender estos filtros. La capa de entrada recibe los valores de intensidad de los píxeles de las imágenes que se emplean para entrenar. Luego, los filtros en las capas ocultas de niveles inferiores detectan regiones con forma de bordes y de esquinas, y posteriormente, las capas de nivel superior usan estas características para aprender conceptos más abstractos, tales como "partes del objeto", útiles para discriminar entre clases de imágenes. Finalmente, la capa de salida logra clasificar la imagen y obtener a su salida una distribución de probabilidad sobre las etiquetas de clase [38].

Las CNN son mayormente utilizadas para problemas de reconocimiento de imágenes o del habla. Esto se debe a que las CNN superan a las ANN en este tipo de problemas en cuanto a exactitud. Incluso, como se mencionó en el Capítulo 1, la exactitud de algunos modelos de CNN en el reconocimiento de imágenes llega a superar a la de los humanos [64]. Es por esta razón que tener *hardware* especializado disponible es crucial al momento de trabajar con herramientas de aprendizaje profundo.

Una mayor capacidad de cálculo permite trabajar con una cantidad de datos más amplia, en comparación a aquellas utilizadas previamente. A su vez los avances tecnológicos permiten emplear redes más complejas, con una mayor profundidad o cantidad de capas. En general, cuanto más grande sea la disponibilidad de estos dos elementos, mayor será la precisión de clasificación. Este comportamiento es diferente al de los algoritmos tradicionales de ML (tales como, regresión logística, SVM, árboles de decisión, etc.) donde se alcanza una meseta en el rendimiento incluso cuando aumentan los datos de entrenamiento disponibles [38]. La Figura 14 muestra el aumento en la precisión de clasificación de los algoritmos tradicionales y de DL en

función de la cantidad de datos de entrenamiento disponibles. Se observa que las CNN obtienen una mayor precisión de clasificación, mientras que los métodos anteriores se estabilizan en cierto punto. Debido a la relación entre el aumento de la precisión junto al aumento en la cantidad de datos, se debe asociar el DL también con grandes conjuntos de datos [38].

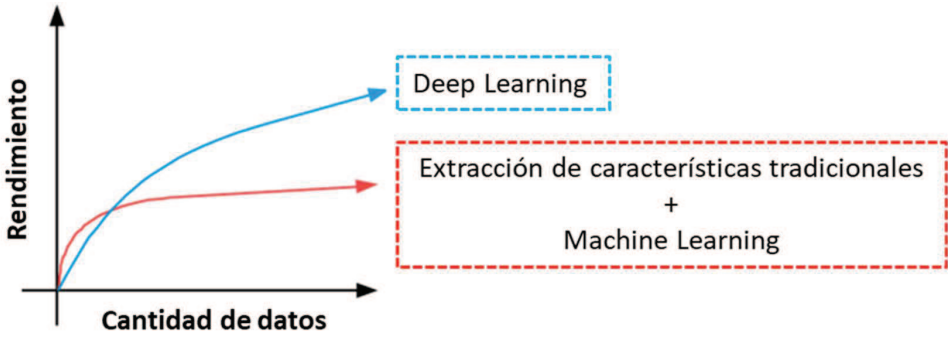


FIGURA 14: GRÁFICA DE COMPARACIÓN DE RENDIMIENTO ENTRE ALGORITMOS DE DL Y ALGORITMOS TRADICIONALES DE ML. EL RENDIMIENTO DE LOS ALGORITMOS DE DL AUMENTA CON LA CANTIDAD DE DATOS, MIENTRAS QUE LOS ALGORITMOS TRADICIONALES EVENTUALMENTE LLEGAN A UNA MESETA QUE IMPIDE MEJORAR SU RENDIMIENTO. ADAPTADO DE [38].

3.3. Dataset de entrenamiento y evaluación

Para que las computadoras interpreten el contenido de una imagen es necesario utilizar visión computacional y algoritmos de ML. Esta tarea se denomina *clasificación de imágenes*, la cual busca asignar una etiqueta a una imagen de un conjunto predefinido de clases descubriendo patrones subyacentes en el conjunto de datos que permitan clasificar correctamente los elementos [9], [38].

La diferencia entre cómo un humano percibe el contenido de una imagen y cómo se puede representar para que la computadora logre interpretarla, se denomina *brecha semántica*. Las imágenes pueden ser descritas de varias maneras:

1. Espacial: se diferencian regiones según su ubicación en la imagen (por ejemplo, el cielo está en la parte superior de la imagen y el océano en la parte inferior);

2. Color: las regiones se reconocen acorde al color que poseen (por ejemplo: el cielo es azul oscuro y el agua del océano es de un azul de diferente tonalidad);

3. Textura: las regiones se distinguen según los patrones que poseen (por ejemplo, el cielo tiene un patrón relativamente uniforme, mientras que la arena tiene un patrón granulado) [38].

Las imágenes deben ser codificadas para que las computadoras puedan interpretarlas. Esto se realiza mediante la extracción de características, lo cual es el proceso mediante el cual se toma una imagen de entrada, se le aplica un algoritmo y se obtiene un vector de características que cuantifica la imagen. Para lograr este proceso de manera automática, se aplica DL [38].

Además de la brecha semántica, las computadoras deben poder manejar factores de variación en cómo aparece una imagen u objeto. Estos factores incluyen: (1) variación de perspectiva; (2) escala; (3) deformación; (4) oclusiones; (5) cambios en la iluminación; (6) desorden de fondo; y por último (7) variación intraclase. A su vez, el sistema de clasificación de imágenes debería ser resistente a tanto las variaciones mencionadas de manera independiente como a la combinación de las mismas [38]. Por esta razón es importante definir correctamente el enfoque de los problemas de clasificación. De adoptarse uno demasiado amplio, donde podría haber cientos de objetos posibles, es poco probable que el sistema de clasificación funcione bien.

De enmarcar el problema y reducir su alcance, aumenta la probabilidad de que el sistema sea preciso y exitoso [38].

Por otra parte, pueden distinguirse dos tipos de imágenes diferentes: aquellas en escala de grises o las imágenes a color o *RGB* [54]. En una imagen en escala de grises, cada píxel contiene un único valor de intensidad de ese mismo píxel. Por lo general, se ve en una escala de 0 a 255. Pero en una imagen a color, cada píxel contiene tres valores en lugar de uno. Estos son rojo, verde y azul, o R, G y B por sus nombres en inglés. Por lo tanto, para una imagen de clase uint8, cada píxel porta tres valores entre 0 y 255, un valor para cada uno de los colores, y dependiendo de su combinación es posible obtener el color correspondiente del píxel [54]. Entonces puede decirse que las imágenes RGB están compuestas por tres capas: una compuesta por valores de color rojo llamada *canal rojo*, otra compuesta por valores de color verde llamada *canal verde* y una última capa compuesta por valores de color azul llamada *canal azul*, tal como se observa en la Figura 15. Por otro lado, una imagen en escala de grises, posee una única capa [54].

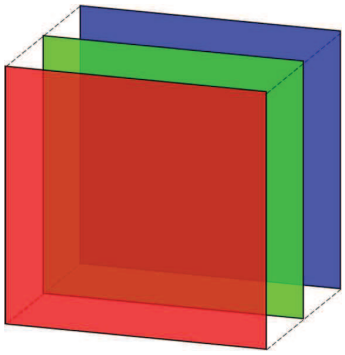


FIGURA 15: ESQUEMA DE IMAGEN RGB COMPUESTA POR TRES CANALES O CAPAS: CANAL ROJO, CANAL VERDE Y CANAL AZUL.

3.3.1. Armado de conjunto de datos

Para generar un clasificador basado en DL con aprendizaje supervisado se debe adoptar un enfoque basado en datos: proporcionando a la red ejemplos de cada clase y luego entrenándola a reconocer la diferencia entre ellos. Dichos ejemplos representan un conjunto de datos de entrenamiento de imágenes etiquetadas, donde cada elemento consta de una imagen y una etiqueta asociada a la clase de la imagen [9], [38].

Entonces, el primer componente de la construcción de una red de DL es recopilar el conjunto de datos inicial. Es decir, se deben recolectar tanto las imágenes en sí, así como las

etiquetas asociadas a cada imagen [38]. Dichas etiquetas deben provenir de un conjunto finito de clases tales como “perros” y “gatos”. Además, el número de imágenes para cada clase debe ser aproximadamente uniforme, es decir, se debería recolectar la misma cantidad de imágenes por cada una de ellas. De tener las cantidades de imágenes desbalanceadas, el clasificador tenderá a sobreajustar hacia la clase más representada. El desequilibrio de clases es un problema común en el ML [38].

En este proyecto se trabajó con imágenes dermatoscópicas obtenidas del repositorio de libre acceso denominado *The International Skin Imaging Collaboration* (ISIC). Las imágenes seleccionadas fueron descargadas en formato *.jpg*, y son todas de calidad dermatoscópica, con confirmación de diagnóstico histopatológico. Además, pertenecen a individuos entre los 5 y 85 años y corresponden varios sitios anatómicos (torso, cabeza, cuello, palmas de la mano, palmas de los pies, y extremidades inferiores y superiores). ISIC permite filtrar las imágenes según diferentes atributos, tales como: de diagnóstico, clínicos, tecnológicos (en cuanto a la calidad de las imágenes) y atributos referidos a sub-conjuntos o *datasets*. Como el objetivo del algoritmo es clasificar de manera binaria entre lesiones “benignas” y “malignas”, se seleccionaron imágenes con dichas características; este fue el único filtro aplicado, además de aquellos referidos a los datasets mencionados a continuación. El repositorio ISIC ofrece más imágenes de las que fueron utilizadas en este proyecto, pero debido al costo computacional que implica el procesamiento de imágenes, se decidió trabajar con una muestra de las mismas. Las imágenes pre-seleccionadas pertenecen en su mayoría al dataset denominado ISIC HAM10000. De acuerdo a la revisión bibliográfica llevada a cabo, este es el dataset más comúnmente utilizado en desarrollos similares al propuesto en el presente trabajo, motivo que justificó su elección [24], [47], [48], [50], [65]. Como la cantidad de imágenes benignas superaba considerablemente a la cantidad de imágenes malignas, se decidió recolectar imágenes de otros datasets para la segunda clase. La selección de los dataset realizada para aumentar la cantidad de datos de lesiones malignas fue hecha de manera aleatoria, filtrando la búsqueda por lesiones con diagnóstico melanoma y por cantidad de imágenes. A modo resumen, en una primera instancia el dataset de este proyecto se conformó por imágenes, pertenecientes a los datasets HAM10000, MSK, UDA y VIENNA bajo el filtro de diagnóstico “benigno” y “maligno”. La Tabla 3 muestra la distribución de imágenes por *dataset*.

Tabla 3: Distribución por dataset de la pre-selección de las imágenes

Dataset	Benigna	Maligna
HAM 10000	2240	1113
MSK	-	822
UDA	-	196
VIENNA	-	43
Total	2240	2174

Una vez pre-seleccionadas, las imágenes fueron revisadas de a una por vez para asegurarnos de la calidad de las mismas y evitar repeticiones de una misma lesión, aunque esto último no fue posible asegurarlo debido a que la planilla de metadatos no indicaba un id para cada paciente. De poseer artefactos tales como reglas de medición, mala definición o recortes importantes de la lesión, estas imágenes serían borradas. A su vez, existían varias imágenes muy parecidas entre sí, o incluso repetidas, las cuales también fueron eliminadas. En cuanto al artefacto de la regla, se observó que esta aparecía en mayor medida en las imágenes de lesiones malignas que en las de lesiones benignas, como esto podría generar un sesgo al momento del entrenamiento de la red, se decidió eliminar las imágenes que tuvieran reglas muy visibles. En general las imágenes benignas estaban libres de estos artefactos, pero para conservar la relación 1:1 entre ambas clases, se eliminaron de manera aleatoria 336 imágenes, tal como se observa en la Tabla 4.

Tabla 4: Distribución final de imágenes por clase de clasificación

	Benigna	Maligna
Pre-selección	2240	2174
Eliminadas	336	263
Selección final	1904	1904

El siguiente paso es dividir el conjunto de datos inicial en tres partes: (1) conjunto de entrenamiento, (2) conjunto de validación, (3) conjunto de prueba. El conjunto de entrenamiento es utilizado por la red para “aprender” cómo se ve cada clase al hacer predicciones sobre los datos de entrada y luego corregirse cuando las predicciones son incorrectas. Una vez que se ha entrenado al clasificador, es posible evaluar su rendimiento con un conjunto de prueba. El conjunto de entrenamiento y el conjunto de prueba deben ser independientes entre sí, es decir, no deben superponerse [9]. Esto se debe a que la red no debe conocer las etiquetas de clase

verdaderas de aquellas imágenes que se van a utilizar para validar la red. Los tamaños de división comunes para conjuntos de entrenamiento y prueba son 66,6% / 33,3%, 75% / 25% y 90% / 10%, respectivamente [38].

El conjunto de validación sirve para ajustar los *hiperparámetros* de la red de manera tal de obtener un rendimiento óptimo (se desarrollará sobre los hiperparámetros en la Sección 3.5.3.). Este conjunto de datos normalmente proviene de los datos de entrenamiento y es utilizado como "datos de prueba falsos" para poder ajustar los hiperparámetros de la red. Solo después de haber determinado los valores de los hiperparámetros utilizando este conjunto, se recopilan los resultados de evaluación final de la red utilizando los datos de prueba. Normalmente se asignan aproximadamente el 10-20% de los datos de entrenamiento para la validación [38].

En este proyecto se realizó una división 75% / 25% para los conjuntos de entrenamiento y prueba, y el conjunto de validación se formó a partir del 10% de las imágenes del conjunto de entrenamiento. Los tres conjuntos son independientes entre sí y poseen la misma proporción de imágenes "malignas" y "benignas", junto a una misma proporción de imágenes pertenecientes a los diferentes datasets del ISIC (HAM 10000, MSK, UDA, VIENNA). Durante el entrenamiento la red visualizó los conjuntos de entrenamiento y de validación y posteriormente al evaluar los resultados del entrenamiento se utilizó el conjunto de prueba. En resumen, se trabajó con un total de 2571 imágenes de entrenamiento, 285 imágenes de validación y 952 imágenes de prueba.

3.3.2. Preparación de datos para el entrenamiento

Tamaño de las imágenes

La mayoría de las ANN y las CNN aplicadas a la tarea de clasificación de imágenes asumen una entrada de tamaño fijo, lo que significa que las dimensiones de todas las imágenes que pasan por la red deben ser las mismas [38]. En el caso de este proyecto, se escogió 224 × 224 píxeles como tamaño de entrada de las imágenes, debido a la arquitectura de red seleccionada, que se detallará más adelante. Como las imágenes utilizadas poseen un tamaño mayor, fue necesario cambiar su tamaño disminuyéndola en términos de ancho y alto. Se mantuvo la relación de aspecto de las imágenes para evitar que se vean comprimirlas y distorsionarlas, ya que en este conjunto de datos la forma de los elementos en las imágenes es relevante para la clasificación.

Rescale: factor de cambio de escala

El *rescale* es un valor por el que se multiplican los datos antes de cualquier otro procesamiento. Las imágenes originales consisten en componentes RGB, donde cada píxel posee un valor de intensidad en el rango 0-255. Como tales valores son demasiado altos para que los modelos los procesen, suele trabajarse con valores entre 0 y 1, aplicando un factor de escala de 1/255 [66].

Trabajar con esta herramienta tiene varias ventajas, tales como: todas las imágenes se tratan de la misma manera, es decir, escalar las imágenes al mismo rango [0,1] permite que las mismas contribuyan de manera más uniforme a la actualización de pesos de las neuronas. Por otro lado, habilita el uso de valores típicos en el entrenamiento de la red que se observan en otros trabajos, siempre y cuando estos también hayan aplicado esta transformación a su conjunto de datos de imágenes. De lo contrario, se deberá trabajar con tasas de aprendizaje diferentes [66].

Preprocesamiento de las imágenes

A pesar de que la mayoría de las lesiones se encuentran en el centro de la imagen, la magnificación de la misma difiere de manera sistemática entre clases, haciendo del dataset uno bastante heterogéneo. Tal como se observa en la Figura 16, las imágenes benignas poseen una mayor cantidad de píxeles alrededor de la lesión, mientras que, en el caso de las lesiones malignas, estas se encuentran magnificadas de manera tal que ocupan una mayor proporción de la imagen. Por esta razón no fue posible utilizar una técnica de preprocesamiento de recorte de bordes de manera centralizada y se optó por trabajar con las imágenes crudas obtenidas del repositorio de imágenes. No obstante, este obstáculo fue resuelto mediante el empleo de la técnica denominada aumentación de datos o *data augmentation*.

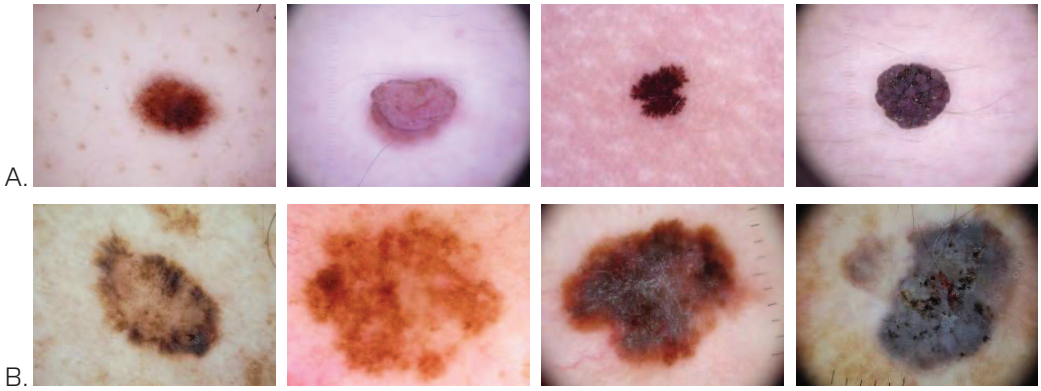


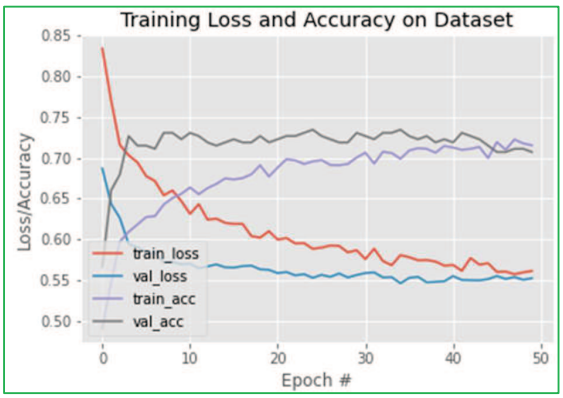
FIGURA 16: IMÁGENES PERTENECIENTES AL DATASET DE ENTRENAMIENTO OBTENIDAS DE [67] : A) LESIONES BENIGNAS, B) LESIONES MALIGNAS.

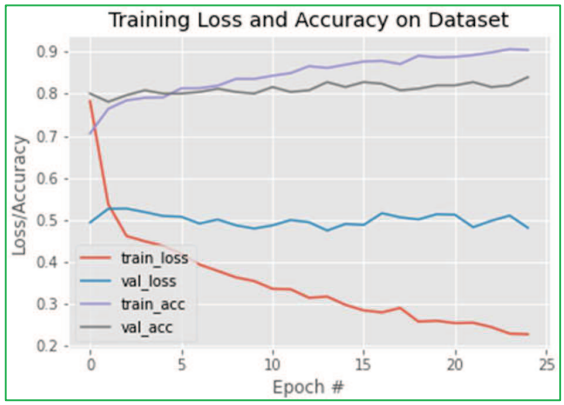
Data augmentation

La capacidad de una red para generalizar y predecir correctamente la etiqueta de clase de una imagen que no existe como parte de sus datos de entrenamiento o prueba, es el aspecto más importante de cualquier algoritmo de IA. Cuando las predicciones de la red son incorrectas, se debe a un problema denominado falta de generalización. Existen varias estrategias en ML que están diseñadas explícitamente para reducir el error de prueba, a expensas de un aumento en el error de entrenamiento. Estas estrategias son comúnmente conocidas como *regularización* [38].

De no aplicar la regularización, el modelo puede volverse demasiado complejo y puede sobre ajustarse (*over-fitting*) a los datos de entrenamiento, en cuyo caso se pierde la capacidad de la red de generalizar a los datos de prueba y a imágenes fuera de dicho conjunto. Sin embargo, demasiada regularización puede llevar a un ajuste insuficiente (*under-fitting*), en cuyo caso el modelo tiene un rendimiento deficiente en los datos de entrenamiento y no es capaz de modelar la relación entre los datos de entrada y las etiquetas de clase de salida [38].

La Figura 17 muestra tres ejemplos diferentes donde se puede observar un caso de *under-fitting* (A), *over-fitting* (B) y generalización (C). El objetivo al crear clasificadores de DL es obtener un modelo que se adapte bien a los datos de entrenamiento, evitando el sobreajuste. La regularización facilita la obtención del tipo de ajuste deseado.

A.


B.


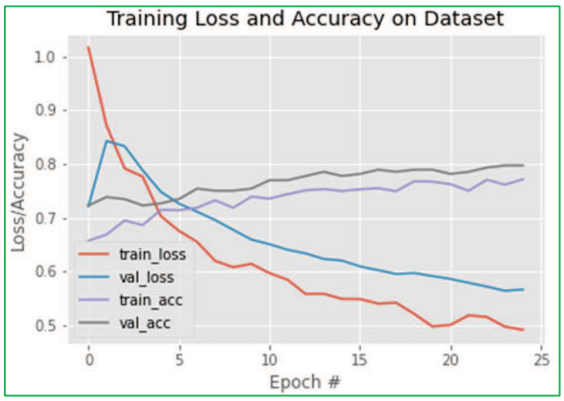
C.


FIGURA 17: EJEMPLO DE A) UNDER-FITTING, B) OVER-FITTING Y C) GENERALIZACIÓN. EN LAS TRES IMÁGENES SE OBSERVAN CUATRO CURVAS: LA FUNCIÓN DE PÉRDIDA DE ENTRENAMIENTO (ROJA), LA FUNCIÓN DE EXACTITUD DE ENTRENAMIENTO (LILA), LA FUNCIÓN DE PÉRDIDA DE VALIDACIÓN (AZUL), LA FUNCIÓN DE EXACTITUD DE VALIDACIÓN (GRIS). ESTOS CONCEPTOS SE ABORDAN EN LA SECCIÓN 3.5.

Existen varios tipos de técnicas de regularización, como la regularización L1, la regularización L2 y Elastic Net, que son mecanismos de regularización parametrizadas que se utilizan actualizando una función, denominada *función de pérdida*, la cual será mencionada en el capítulo de entrenamiento de la red. También existen tipos de regularización que se pueden agregar explícitamente a la arquitectura de la red; las capas *dropout* son el ejemplo por excelencia de dicha regularización (el funcionamiento de las mismas será desarrollado en la Sección 3.4.4.). Luego existen formas implícitas de regularización que se aplican durante el proceso de entrenamiento. Los ejemplos de regularización implícita incluyen el aumento de datos y la detención anticipada o *Early Stopping* [38].

En este trabajo la principal estrategia utilizada para mejorar la generalización del algoritmo fue el aumento de datos. Esta herramienta abarca una amplia gama de técnicas utilizadas para generar nuevas muestras de entrenamiento a partir de las originales mediante la

aplicación de perturbaciones y fluctuaciones aleatorias de modo que las etiquetas de las clases no se modifiquen. El objetivo al aplicar el aumento de datos es incrementar la generalización del modelo, y en parte, alivianar la necesidad de recopilar más datos de entrenamiento. Dado que la red ve constantemente versiones nuevas y ligeramente modificadas de los puntos de datos de entrada, puede aprender patrones más robustos que contribuyen a la correcta discriminación de las diferentes clases. Al evaluar la red ya entrenada no se aplica el aumento de datos ya que para obtener las métricas de evaluación alcanza con que la red visualice las imágenes originales del conjunto de prueba [68].

Es decir, para aumentar la generalización del clasificador, es posible alterar aleatoriamente los puntos a lo largo de la distribución agregando valores extraídos de una distribución aleatoria (Figura 18, derecha). La gráfica de la distribución sigue una distribución normal, pero con alteraciones. De esta manera es más probable que un modelo entrenado con estos datos se generalice a puntos de datos de ejemplos no incluidos en el conjunto de entrenamiento [68].

En el contexto de la visión por computadora utilizar el aumento de datos es una práctica común y simple de aplicar. Por ejemplo, es posible obtener datos de entrenamiento adicionales de las imágenes originales aplicando transformaciones geométricas simples y aleatorias tales como: traslaciones, rotaciones, cambios de escala, transformaciones de corte, volteo (*flip*) horizontal y vertical [68].

A pesar de que cada imagen aumentada puede considerarse una imagen "nueva", aumentar los datos no implica incrementar la cantidad real de imágenes vistas por el algoritmo. En cada iteración del algoritmo a cada imagen se le aplica una transformación diferente, logrando visualizar a partir de una única imagen varias variaciones de la misma por iteración [69].

Debido a las características inherentes de las imágenes utilizadas, fue posible realizarles las diferentes transformaciones mencionadas. Al utilizar la transformación de magnificación o *zoom* se logra disminuir la discrepancia entre las relaciones lesión/*background* en las imágenes mencionadas en el segmento anterior. En este caso, al aplicar la transformación *zoom*, la red ve un espectro de variaciones de las imágenes con diferentes niveles de magnificación.

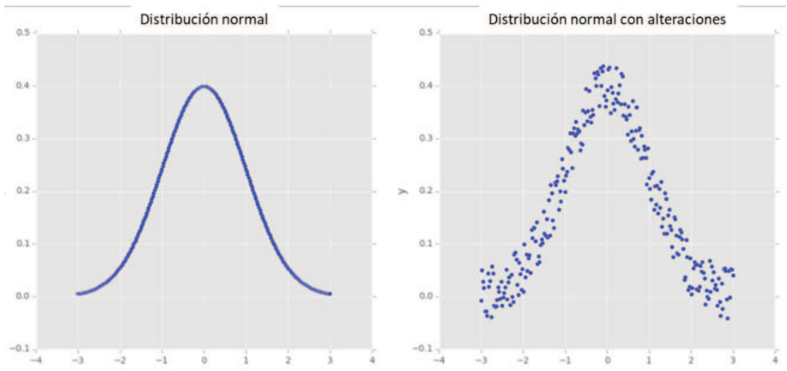


FIGURA 18: IZQUIERDA: DISTRIBUCIÓN NORMAL DE MUESTRA. DERECHA: DISTRIBUCIÓN NORMAL CON ALTERACIONES ALEATORIAS. ADAPTADO DE [68].

3.4. Arquitectura

Antes de entrenar el modelo es necesario definir la arquitectura del mismo. Es decir, se debe establecer la cantidad de neuronas de entrada, el número de capas ocultas del modelo, la función de activación de dichas capas y la cantidad de neuronas de salida, junto a su función de activación. En este apartado se describe la arquitectura elegida para el modelo empleado en el proyecto, pero antes es necesario repasar ciertos conceptos respecto a las redes neuronales convolucionales.

3.4.1. Redes neuronales convolucionales

Los modelos con CNNs poseen una estructura diferente a aquellos con ANNs convencionales. La Figura 19 representa un esquema genérico de un modelo de DL. En primera instancia se tiene la imagen de entrada a analizar. A continuación, le sigue la arquitectura propiamente dicha del modelo, que puede dividirse en dos partes: una red convolucional para la extracción de características, seguida de una red de clasificación. A la salida se obtiene la predicción de clasificación de la imagen de entrada [9].

Las ANNs convencionales no poseen el bloque de la red convolucional ya que la extracción de características es realizada en una etapa previa, utilizando algoritmos convencionales o estadísticos. Pero en el caso de las CNNs, la extracción se realiza de manera automática, utilizando unas capas especializadas denominadas capas convolucionales y capas de *pooling* [9]. A continuación, se desarrollará cómo funcionan para comprender cómo es que las CNNs logran extraer las características de manera automática.

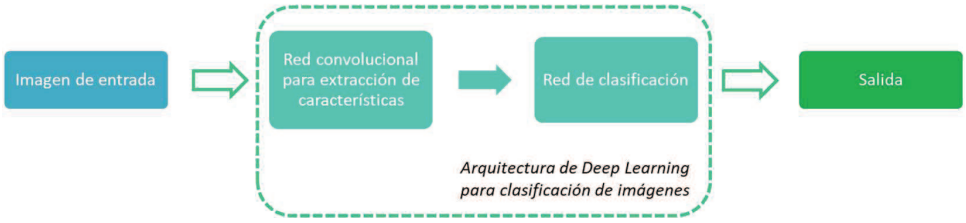


FIGURA 19: ESQUEMA GENÉRICO DE UNA ARQUITECTURA DE DL PARA CLASIFICACIÓN DE IMÁGENES. ADAPTADO DE [9]

Capas convolucionales

Mientras que las ANN convencionales obtienen datos de cada píxel de la imagen como entrada a la primera capa de neuronas, sin considerar el efecto de los píxeles vecinos; las CNN logran emular el funcionamiento el cerebro humano. Esto se logra reconociendo un patrón en un grupo de píxeles. Incluso, pueden simular cómo responden las neuronas de la corteza visual a diferentes patrones. Algunas responden a líneas horizontales, algunas a líneas verticales y otras a patrones complejos. La salida de las neuronas de nivel inferior es luego procesada por neuronas de nivel superior para identificar objetos en ese campo visual [38], [54], [64].

Las CNN se inspiran en este concepto. En lugar de observar cada píxel individual, observan un grupo de píxeles, aumentando la probabilidad de detectar diferentes características de los objetos en la imagen. Una vez obtenidas las características, es más probable lograr predecir el objeto en la imagen [54], [64], [68].

Las CNN contienen capas especializadas denominadas *capas convolucionales*. En el esquema de Figura 20 se observa en la parte inferior la imagen de entrada a la red y, encima de ella, la primera capa convolucional. Esta capa está formada por neuronas que captan información de un grupo de píxeles en la capa anterior, es decir, información espacial. Al conjunto de píxeles sobre los cuales la neurona de la capa siguiente extrae información de la capa anterior, se lo llama *campo receptivo* [38], [54], [64].

En la primera capa convolucional, las neuronas obtienen información almacenada en el campo receptivo asociado a esa neurona en particular como entrada. De manera similar, en la segunda capa convolucional, la información de todas las neuronas en el campo receptivo de la primera capa convolucional es tomada como entrada por esta neurona. Esta arquitectura permite que la red se concentre en entidades de nivel inferior en la primera capa y luego las ensamble en entidades de nivel superior y más grandes en la siguiente capa oculta y así sucesivamente [38], [54], [64]

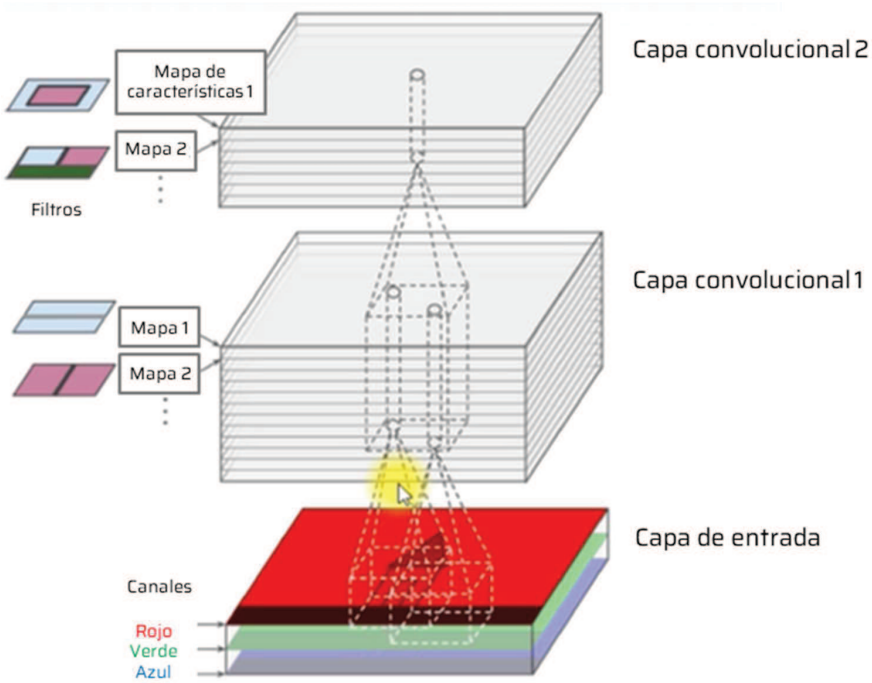


FIGURA 20: ILUSTRACIÓN DE UNA CAPA DE ENTRADA DE TRES CANALES, DOS CAPAS CONVOLUCIONALES SECUENCIALES, Y LOS FILTROS Y MAPAS DE CARACTERÍSTICAS ASOCIADAS A LAS MISMAS. ADAPTADO DE [54].

Para trasladar de manera representativa la información de los píxeles dentro de un campo receptivo a una única neurona, se emplean filtros o *kernels*. Estos son convolucionados con cada valor de píxel de la ventana del campo receptivo para obtener lo que se conoce como *mapa de características*. Se le llama mapa de características porque cada filtro resalta alguna característica particular de la imagen de entrada. Entonces, al utilizar varios tipos de filtros, es posible obtener varios mapas de características donde cada uno contiene información diferente. Esto significa que la capa convolucional es, a fin de cuentas, un paquete de mapas de características, donde cada uno de ellos contiene alguna característica particular resaltada [38], [54], [64].

El proceso se puede resumir de la siguiente manera: se aplica un filtro en la capa de datos anterior para extraer características, obteniéndose un mapa de características. Se aplican varios tipos de filtros para extraer características diferentes a partir de cada uno de ellos. Esto resulta en un paquete de mapas de características. El primer paquete de mapas de características de la Figura 20 se llama capa convolucional 1, mientras que la capa convolucional 2 trabaja con estas características extraídas para extraer características de nivel aún más alto [38], [54], [64]. Para más información en el Anexo 2 se desarrolla estos conceptos con mayor profundidad. Además,

se describen otros conceptos clave sobre el funcionamiento de las CNN como el *stride*, el *padding* y la capa de *pooling*.

3.4.2. Resnet

Existen varias arquitecturas de DL de libre acceso mediante las librerías de Keras y TensorFlow, tales como Xception, Inception, VGG, DenseNet, EfficientNet, entre otras. En el caso de este proyecto se optó por la ResNet50, tomando en consideración que había sido utilizada en varios trabajos [47], [49], [51], [65].

ResNet, del acrónimo en inglés *Residual Network*, es una red neuronal innovadora que fue presentada por primera vez por Kaiming He, Xiangyu Zhang, Shaoqing Ren y Jian Sun en su artículo de investigación de visión computacional de 2015 titulado "Aprendizaje residual profundo para el reconocimiento de imágenes". La ResNet fue notoriamente exitosa, habiendo ganado la primera posición en la competencia de clasificación ILSVRC 2015 con un error de solo 3.57%. Además, también ocupó el primer lugar en la detección de ImageNet y COCO[4], la localización de ImageNet y la segmentación de COCO en las competencias ILSVRC y COCO de 2015 [64], [70].

Este modelo tiene muchas variantes donde la diferencia principal reside en la cantidad de capas. Resnet50 se utiliza para indicar la variante que puede funcionar con 50 capas de red neuronal. Cuando se trabaja con CNNs profundas para resolver problemas relacionados a la visión computacional, los expertos de ML suelen desarrollar modelos más profundos, apilando varias capas. Estas capas adicionales ayudan a resolver problemas complejos de manera más eficiente, ya que las diferentes capas podrían entrenarse para diversas tareas a fin de obtener resultados altamente precisos [64], [70].

Si bien la cantidad de capas apiladas puede enriquecer las características del modelo, una red más profunda puede mostrar el problema de la *degradación*. En otras palabras, a medida que aumenta el número de capas de la red neuronal, los niveles de precisión pueden saturarse y degradarse lentamente después de un cierto punto. Como resultado, el rendimiento del modelo se deteriora tanto en los datos de entrenamiento como en los de prueba. Esta degradación no es

[4] ImageNet y COCO son *datasets* de imágenes ampliamente utilizados en el desarrollo de modelos de DL. Es común utilizarlos en competencias de ML.

el resultado de un sobreajuste. En cambio, puede resultar de la inicialización de la red, la función de optimización o, más importante, el problema de la extinción del gradiente [70].

ResNet se creó con el objetivo de abordar este problema. Las redes residuales profundas utilizan bloques residuales para mejorar la precisión de los modelos. El concepto de "*skip connections*", que se encuentra en el núcleo de los bloques residuales, es la fuerza de este tipo de red neuronal. Las *skip connections* o conexiones de acceso directo funcionan de dos formas. En primer lugar, alivian el problema de la desaparición del gradiente configurando un atajo alternativo para que pase el gradiente. Por otro lado, permiten que el modelo aprenda una función de identidad. Esto asegura que las capas superiores del modelo no se comporten peor que las capas inferiores [64], [70]. En resumen, los bloques residuales facilitan considerablemente que las capas aprendan las funciones de identidad. Como resultado, ResNet mejora la eficiencia de las redes neuronales profundas con más capas neuronales en simultáneo a minimizar el porcentaje de errores. En otras palabras, las *skip connections* agregan las salidas de las capas anteriores a las salidas de las capas apiladas, lo que hace posible entrenar redes mucho más profundas de lo que era posible anteriormente [64], [70].

Arquitectura ResNet-50

Si bien la arquitectura ResNet50 se basa en el modelo ResNet34, existe una diferencia importante: el bloque de construcción se modificó en un diseño de cuello de botella debido a preocupaciones sobre el tiempo necesario para entrenar las capas. Esto consistió en modificar las conexiones de acceso directo a saltar tres capas en vez de solamente dos. A su vez, también se agregaron capas de convolución 1x1 (ver Anexo 1: ResNet50). Por lo tanto, cada uno de los bloques de 2 capas en Resnet34 fue reemplazado por un bloque de cuello de botella de 3 capas, formando la arquitectura ResNet50. Esto tiene una precisión mucho mayor que el modelo ResNet de 34 capas [71]. En el Anexo 1 encontrará más información respecto a la arquitectura de la ResNet50.

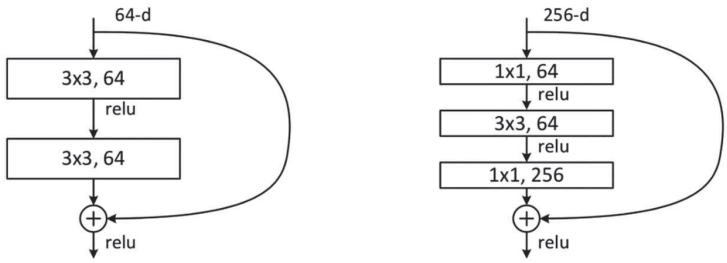


FIGURA 21: COMPARACIÓN ENTRE LA RESNET50 (DERECHA) Y LOS MODELOS DE RESNET PREVIOS [71].

3.4.3. Transfer learning

Los seres humanos tenemos la habilidad nata para abstraer y transferir conocimientos de una tarea a otra. Logramos utilizar el conocimiento adquirido al aprender a realizar una tarea determinada para resolver otras similares. La transferencia de aprendizaje, también conocida como *transfer learning* es una técnica de DL que busca emular esta habilidad, ampliamente utilizada en visión computacional [9]. El transfer learning busca utilizar conocimientos adquiridos para una tarea en la resolución de otra tarea similar y superar de esta manera el esquema de entrenamiento aislado. En visión computacional, la intuición detrás de esta técnica es que, si un modelo se entrena en un conjunto de datos lo suficientemente grande y general, puede servir como un modelo genérico del mundo visual [9], [38], [64].

La motivación principal del transfer learning es que conseguir datos etiquetados para modelos supervisados es una de las grandes limitaciones al momento de desarrollar algoritmos de ML. Existen datasets públicos para tareas generales de visión computacional; entre ellos, ImageNet, construido por Stanford. ImageNet cuenta con 1.300.000 de imágenes etiquetadas según 1.000 clases de objetos. Sin embargo, para tareas específicas se necesitan modelos de DL especializados en un dominio particular de imágenes, y construir datasets de esas dimensiones es un trabajo considerablemente laborioso [9], [68].

Una estrategia ampliamente utilizada para entrenar modelos de DL es usar un modelo previamente entrenado (*pre-trained*), cuyos pesos se encuentran ajustados para otro objetivo. Se asume que el dominio fuente es similar al dominio objetivo. En el campo de la visión computacional, los modelos generalmente están previamente entrenados en algún dataset público de gran volumen, como ImageNet, CIFAR-100 o COCO, entre otros [9], [68].

Se pueden distinguir dos formas de utilizar los modelos previamente entrenados: extractores de características existentes o ajuste fino. Los primeros, también conocidos como *off-the-shelf,* consisten en usar la red convolucional previamente entrenada para convertir las imágenes de alta dimensión en un mapa de características de menor dimensión. Con las características extraídas se pueden entrenar modelos de clasificación o regresión utilizando un dataset que corresponda a la tarea de interés que se desea resolver. Por otro lado, el ajuste fino o *fine-tuning* consiste en reentrenar ciertas capas de la red utilizando el dataset correspondiente a la tarea de interés. Si la arquitectura tiene capas densas finales, sus pesos se reajustan para minimizar el error de predicción en la tarea deseada. También es posible reentrenar capas

convolucionales, en particular las últimas. Las capas cuyos pesos no se reentrenan se conocen como capas congeladas [9].

3.4.4. Arquitectura empleada en el algoritmo de clasificación

En este proyecto se optó por aplicar Transfer Learning, importando el modelo ResNet50, con los pesos del modelo entrenado con el dataset *ImageNet*. A su vez, se eliminaron las últimas capas del modelo. Esto es una práctica común cuando se realiza la transferencia de aprendizaje ya que las últimas capas de cualquier modelo de clasificación se especializan en el objetivo de clasificación del entrenamiento realizado, y es necesario adaptar el modelo al nuevo objetivo de clasificación. Al haber eliminado la cabeza del modelo, se decidió agregar un nuevo bloque de capas al final de la arquitectura para reemplazarla. Este se compuso por seis capas vírgenes, tal como se observa en la Figura 22.

```
model.add(conv_base)
model.add(layers.Flatten())
model.add(layers.BatchNormalization())
model.add(layers.Dense(256, activation='relu'))
model.add(layers.Dropout(0.5))
model.add(layers.BatchNormalization())
model.add(layers.Dense(config.DENSE_N, activation=config.ACTIV_F))
```

FIGURA 22: NUEVA CABEZA DEL MODELO COLOCADA AL FINAL DE LA ARQUITECTURA.

Las capas seleccionadas para la cabeza son capas utilizadas en algoritmos de clasificación convencionales. Como el objetivo de la cabeza es utilizar las características extraídas en la ResNet para luego clasificar, no fue necesario emplear capas convolucionales y de pooling. A su vez, como la ResNet ya es un modelo de cierta complejidad, para evitar el over-fitting se buscó generar una cabeza relativamente simple [38], [62].

La capa *flatten* recibe como entrada la salida de la base convolucional que en este caso es la ResNet. Este tipo de capas convierten datos multidimensionales en un único vector para ser procesado por futuras capas densas. La información en una red convolucional se analiza matricialmente, pero, de aplicar una capa flatten, se obtiene a la salida de la misma un único vector [63]. Las capas flatten no llevan asociadas pesos, solamente se encargan de convertir el input multidimensional en un output unidimensional [72].

Por otro lado, las capas densas son capas cuyas neuronas se encuentran completamente conectadas a las activaciones de la capa anterior. Este tipo de capa siempre se coloca al final de la red, junto a la función de activación correspondiente al objetivo del trabajo a realizar.[68].

En la Sección 3.3.3. se mencionó que las capas *dropout* son una forma de regularización que tiene como objetivo ayudar a prevenir el sobreajuste. Esto lo hace aumentando la precisión de las pruebas, algunas veces a expensas de la precisión del entrenamiento. Por cada lote del conjunto de entrenamiento, las capas dropout desconectan de manera aleatoria y según una probabilidad *p*, las entradas de la capa anterior a la siguiente. La desconexión aleatoria garantiza que ningún nodo de la red sea responsable de la activación cuando se le presente un patrón determinado. En cambio, la deserción garantiza que haya múltiples nodos redundantes que se activarán cuando se les presenten entradas similares, lo que a su vez ayuda en la generalización del modelo [9], [68].

Por último, se encuentran las capas de *batch normalization*. Estas se utilizan para normalizar las salidas de las funciones de activación de la capa anterior y estimar su media y varianza. Luego de normalizar dichas salidas, la capa toma estos valores y los multiplica por algún parámetro arbitrario para después sumarle otro parámetro arbitrario a este producto resultante. De esta manera, el cálculo con los dos nuevos parámetros establece una nueva desviación estándar y media para los datos, donde la media es cero y la varianza unitaria. La media, la desviación estándar y los dos parámetros establecidos arbitrariamente son todos entrenables. Es decir, también se optimizan durante el proceso de entrenamiento. Este proceso hace que los pesos dentro de la red no se desequilibren con valores extremadamente altos o bajos ya que la normalización está incluida en el proceso de entrenamiento. Utilizar capas de batch normalization en los modelos de DL puede aumentar en gran medida la velocidad del entrenamiento y aumentar la capacidad de detener pesos fuera de rango, los cuales influirían en exceso al proceso de entrenamiento [9].

3.5. Entrenamiento del algoritmo de clasificación

El tercer paso de la construcción de una red de DL es entrenarla. El objetivo del entrenamiento es que la red aprenda a reconocer cada una de las clases de los datos etiquetados. Cuando el modelo comete un error, aprende de ese error y mejora. Para poder aumentar la exactitud de clasificación se busca optimizar los valores de los pesos y los sesgos de

la red mediante la utilización de la función de exactitud y la función de pérdida. Dicha optimización requiere de la utilización de algoritmos de optimización tales como el descenso de gradiente [38]. En la Sección 3.6.1. se definen las métricas de evaluación del algoritmo, incluyendo la exactitud.

3.5.1. Función de pérdida

La función de pérdida cuantifica qué tan bien concuerdan las etiquetas de clase predichas con las etiquetas verdaderas o *ground truth*. Cuanto mayor sea el nivel de concordancia entre los dos conjuntos de etiquetas, menor será la pérdida, y mayor será la exactitud de clasificación en el conjunto de entrenamiento. El objetivo al entrenar un modelo de aprendizaje automático es minimizar la función de pérdida, aumentando así la exactitud de clasificación. Sin embargo, no es posible asegurar la eficacia de un modelo observando únicamente la función de pérdida; existen otras características a analizar como por ejemplo si el modelo sobreajusta o no a los datos de entrenamiento [38].

Para una clasificación binaria, como es el caso en este proyecto, la función de pérdida típica es la entropía cruzada binaria. Al combinarla con una función de activación sigmoide a la salida del modelo, se obtienen las probabilidades predichas de pertenencia a la clase positiva (clase “maligna” en este caso). Dado que el objetivo es calcular una función de pérdida mínima, la entropía cruzada binaria busca penalizar las predicciones erróneas. Es decir, si la probabilidad asociada con la clase verdadera es alta, la función de pérdida debe ser baja y viceversa [38], [73].

3.5.2. Método de optimización

El descenso de gradiente es una técnica de optimización para encontrar el mínimo de una función. Es particularmente utilizada cuando se trabaja con una gran cantidad de características y relaciones complejas ya que muestra un buen rendimiento computacional, permitiendo entrenar rápidamente el modelo. [38], [54]. Aunque existen otros métodos de optimización que suelen tener un mejor rendimiento, la selección de qué optimizador utilizar depende del objetivo de cada proyecto. En general, estos métodos comparten una estructura básica de funcionamiento. , razón por la cual a continuación se describe el método denominado descenso de gradiente.

Partiendo del gráfico de la izquierda de la Figura 23, se desea encontrar el mínimo global de la función (punto B). Para hacerlo se comienza en un punto aleatorio (A) y se intenta averiguar en qué dirección está la pendiente. Si esta es negativa, el algoritmo da un paso en aquella dirección. Esto lo hace hasta finalmente encontrar el punto que referencia el mínimo global de la función. Al llegar a este punto, moverse en cualquier dirección solamente aumenta el valor de la función, razón por la cual el proceso se detiene [38], [54].

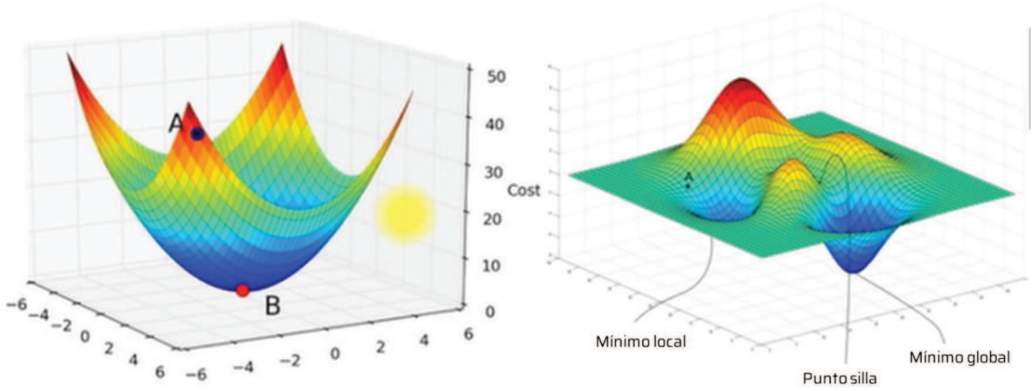


FIGURA 23: DESCENSO DE GRADIENTE. LA SUPERFICIE REPRESENTA LA FUNCIÓN DE COSTO O DE PÉRDIDA. ADAPTADO DE [54].

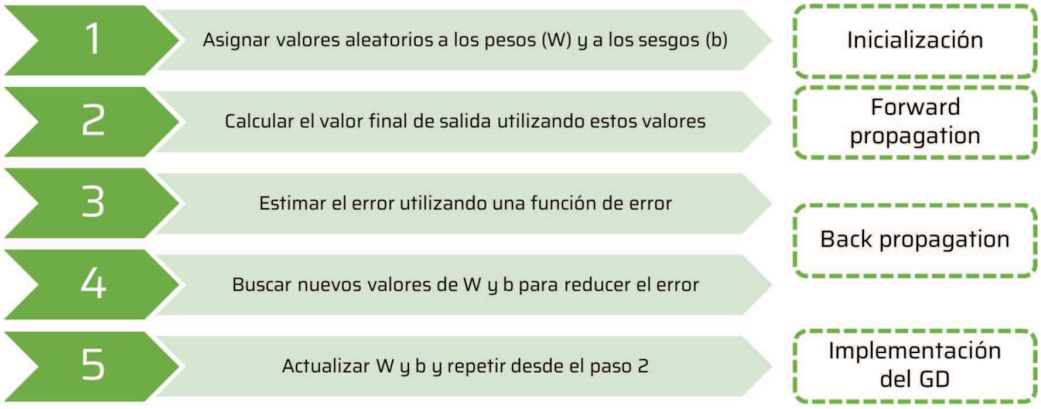


FIGURA 24: PROCESO DE IMPLEMENTACIÓN DEL DESCENSO DE GRADIENTE. ADAPTADO DE [54].

La Figura 24 muestra cómo se implementa el descenso de gradiente. Se comienza asignando pesos y valores de sesgo aleatorios a los nodos de la red. Dado que todas las ponderaciones y los valores de sesgo están disponibles, el modelo está listo para dar una salida. El segundo paso es ingresar un ejemplo de entrenamiento. Se utilizan los valores de pesos y sesgo del entrenamiento y se calcula la salida final de la red. El tercer paso es comparar los valores predichos con los valores reales para conocer la diferencia entre estos dos utilizando

alguna función de error. Al conocer el valor real del elemento ingresado a la red, se logra dar una retroalimentación a la misma sobre qué tan bueno o malo está siendo su desempeño. El cuarto paso es tratar de descubrir esos cambios de peso y sesgos, para reducir la función de error. Por último, se actualizan los valores de ponderaciones y sesgos y se repite este proceso desde el segundo paso. Este bucle continúa hasta llegar a un mínimo de la función de pérdida que puede representar un mínimo local profundo o el mínimo global de la misma [38], [54].

El primer paso se llama Inicialización, donde se le asignan valores iniciales aleatorios a los pesos y los sesgos. En el segundo paso, llamado *forward propagation*, se ingresa un valor a la entrada de la red, la cual es luego procesada sistemáticamente por el resto de las capas hasta obtener una salida final predicha. Como se avanza hacia adelante en términos de capas de la red, se dice que la propagación es hacia adelante. El tercer y cuarto paso se denominan *back propagation*. En estos pasos se obtiene una función de error final y realiza un ciclo de realimentación donde se averigua qué pesos y sesgos tienen el máximo impacto en la función de error. Una vez que se establecen, estos se actualizan ligeramente para reducir el error [38], [54]. El término correcto para describir esto es en realidad descenso de gradiente estocástico ya que se mencionó que se toma cada registro de entrenamiento individual y se actualizan los pesos y sesgos en función de este. Pero de ejecutar la propagación hacia adelante para todo el conjunto de entrenamiento, y buscar el error promedio de este, se lo denomina descenso de gradiente [38], [54].

Existe otra variación en la que se aplica el descenso de gradiente a lotes del conjunto de entrenamiento en lugar de al conjunto completo. Esta se llama *descenso de gradiente por lotes*. El descenso de gradiente estocástico encuentra dificultad para converger debido a que actualiza los pesos y los sesgos rápidamente. Por otro lado, el descenso de gradiente, a pesar de tener una muy buena convergencia, es considerablemente más lento en comparación, debido a que en cada ciclo debe pasar por todo el conjunto de entrenamiento. El método por lotes permite ajustar las mejores propiedades de los dos métodos [38], [54].

En este proyecto se probaron varios optimizadores: entre ellos SGD (descenso de gradiente estocástico), SGDR (descenso de gradiente estocástico con restart)[5] y Adam. El

[5] Método utilizado para evitar caer en un mínimo local profundo de la función de costo. El SGDR supera este problema incrementando repentinamente en algunos pasos de tiempo específico la tasa de aprendizaje [51].

optimizador llamado Adam fue seleccionado para la versión final del algoritmo debido a que se obtuvieron mejores resultados en cuanto a la función de pérdida y a la de exactitud.

Este es un método de descenso de gradiente estocástico que se basa en la estimación adaptativa de momentos [74]. Además, según Kingma et al. [75] el optimizador Adam es computacionalmente más eficiente que otros optimizadores como el SGD y el RMSProp, tiene pocos requisitos de memoria y es muy adecuado para problemas donde se trabaja con grandes cantidades de datos. A diferencia del descenso de gradiente estocástico, donde se mantiene una misma tasa de aprendizaje para cada peso de la red y esta se adapta por separado a medida que se desarrolla el aprendizaje, el optimizador Adam calcula diferentes tasas de aprendizaje de manera adaptativa e individual para diferentes parámetros a partir de estimaciones del primer y segundo momento de los gradientes. En lugar de adaptar las tasas de aprendizaje de los parámetros en función del primer momento promedio (la media) como en el optimizador RMSProp, Adam también hace uso del promedio de los segundos momentos de los gradientes (la varianza no centrada) [76].

3.5.3. Hiperparámetros

Los hiperparámetros son parámetros preestablecidos del proceso de aprendizaje. Es decir, no son parámetros internos del modelo encontrados durante dicho proceso, sino que deben establecerse antes del entrenamiento [38].

Tal como indica la palabra, se habla de hiperparámetros debido a la cantidad de parámetros disponibles para modificar el rendimiento del modelo de aprendizaje. No existen reglas estrictas sobre cómo configurar estos parámetros. Varían según el objetivo del proyecto y según los recursos disponibles para su desarrollo; razón por la cual se deben probar diferentes valores de los mismos y optar por el que mejor funciona para el problema propuesto.

Tamaño de lote o *batch size*

El tamaño de lote o *batch size* es la cantidad de muestras procesadas por el modelo antes de que este se actualice. El tamaño de un lote debe ser mayor o igual a uno y menor o igual al número de muestras en el conjunto de datos de entrenamiento. Generalmente se trabaja con los siguientes valores: 32, 64, 128, 256 [9], [38].

En este proyecto se trabajó con un batch size de 32 ya que no solo aliviana el costo computacional, debido a la cantidad de muestras procesadas en paralelo, sino también le aporta al entrenamiento de la red una cierta cantidad de ruido a la convergencia de la red que le permite encontrar mínimos locales más amplios. Esto último se debe a que la utilización de batches o lotes más chicos, tienen una mayor variación entre sí, por lo que la velocidad y la dirección de convergencia de la función de costo es más variable. En contraste, al utilizar un batch o lote demasiado grande, o incluso todos los datos, se obtiene como resultado una convergencia suave hacia un mínimo local profundo. Por lo tanto, un tamaño de lote más pequeño puede proporcionar una regularización implícita para su modelo [77].

Épocas o *epochs*

El número de épocas es el número de ciclos completos a través del conjunto de datos de entrenamiento. Este parámetro puede tomar un valor entre uno e infinito. Es decir, es posible ejecutar el algoritmo durante el tiempo que se desee e incluso es posible detenerlo utilizando otros criterios además de un número fijo de épocas, como, por ejemplo, un cambio (o falta de cambio) en el error del modelo a lo largo del tiempo [38]. Las épocas se diferencian de las iteraciones ya que definen la cantidad de veces que el conjunto de datos será alimentado a la red, independientemente del tamaño del batch size, es decir, independientemente de si las imágenes serán vistas de a una, por lotes o todas al mismo tiempo. Utilizar épocas permite que la red vuelva a verificar su desempeño con los mismos datos [54].

La cantidad de épocas utilizadas en este proyecto no fue fija ya que, como se verá más adelante, es posible seleccionar los pesos de una época en particular y después seguir entrenando modificando o no los hiperparámetros de la red. Al momento de familiarizarse con estas tecnologías, se probaron diferentes cantidades de épocas para evaluar si era beneficioso entrenar la red por 50-100 épocas. Como el valor de este hiperparámetro afecta de manera directa al costo computacional, se optó por entrenar de a bloques de épocas más chicas, e interconectarlas.

Tasa de aprendizaje o *learning rate*

La tasa de aprendizaje es uno de los parámetros más importantes del modelo [38]. Para explicar su rol en la optimización del algoritmo, es necesario volver a la funcionalidad del descenso de gradiente (Sección 3.5.2.). En el método de descenso de gradiente, la superficie de la

colina es la función de costo, y la altura varía en función de los parámetros del modelo. En cada punto de la superficie se define un gradiente (variación local de pendientes), donde se toma la decisión de hacia dónde avanzar, conociendo únicamente la información local, pero no la solución global. En cada iteración se determina en qué dirección cambiar los parámetros o en qué dirección avanzar [9].

La magnitud de los parámetros es muy importante a la hora de ajustar un modelo. Con el signo del gradiente se define que un parámetro debe reducirse o aumentarse, y con su magnitud, la proporción con la que se ajusta. Esta siempre será proporcional al módulo del gradiente: cuanta mayor influencia tenga un parámetro en el error total, mayor deberá ser la magnitud de variación. Normalmente, este error se multiplica por un coeficiente denominado tasa de aprendizaje o *learning rate*, que determina en qué medida penalizar el error. El learning rate es un hiperparámetro, ya que su valor se define de forma explícita para realizar el entrenamiento del modelo [9].

Entonces, sabiendo que el objetivo es encontrar el mínimo global de la función de costo y que, en cada iteración, el optimizador realiza un salto en esta superficie, la magnitud del salto estará determinada por la tasa de aprendizaje. Si esta es demasiado grande, los saltos sobre la superficie serán demasiado largos, lo cual dificulta encontrar mínimos locales. Por el contrario, de ser demasiado chico, se necesitarán varias iteraciones hasta finalmente llegar a un valor de pérdida óptimo [9], [38], [77]. Los batch size pequeños suelen adaptarse mejor a tasas de aprendizaje chicas dada la ruidosa estimación de la función de costo. Un valor predeterminado tradicional para la tasa de aprendizaje es 0.1 o 0.01 [78].

Puede apreciarse que la tasa de aprendizaje interactúa con varios aspectos del proceso de optimización donde las interacciones pueden ser no lineales. Por esta razón en el proyecto se probaron varios valores de learning rate hasta lograr ajustar en mejor medida el modelo. Finalmente se optó por un learning rate de 0.001 para la primera parte del entrenamiento y un learning rate de 0.0001 en la segunda parte donde se aplicó fine-tuning.

3.5.4. Resultados del entrenamiento

En este proyecto, la red fue entrenada en dos instancias. En la primera se entrenó solamente la cabeza, manteniendo congeladas todas las capas de la ResNet excepto las capas de tipo batch normalization. Se utilizó el optimizador Adam, con una tasa de aprendizaje de 0.001 y un tamaño de lote de 32, y se entrenó hasta la época 17. Luego se tomó este punto para continuar entrenando la red, pero de esta vez descongelando el último bloque convolucional de la ResNet y utilizando una tasa de aprendizaje menor, del orden del 0.0001. El punto seleccionado para ser utilizado en la aplicación fue el de la época 5 de este segundo modelo (marcado como época 22 en la Figura 25). A partir de este momento puede notarse que las curvas sobreajustan, razón por la cual no se optó por un punto en una época posterior.

La Figura 25 muestra un total de cuatro curvas del entrenamiento del algoritmo desarrollado: exactitud de entrenamiento (celeste/azul), función de pérdida o costo de entrenamiento (amarillo/marrón), exactitud de validación (rojo/bordo) y función de pérdida de validación (verde/verde oscuro). A su vez se observan dos rectas verticales grises que representan los puntos o *checkpoints* seleccionados donde, en primera instancia, se modificó el valor de la tasa de aprendizaje y las capas congeladas de la ResNet, y en segunda instancia como checkpoint final para ser utilizado posteriormente en la aplicación. El objetivo es lograr que las curvas de exactitud converjan en un punto máximo, mientras que las funciones de pérdida converjan a un valor mínimo, debido que se busca la mejor exactitud posible y un valor de pérdida que represente un mínimo de la función.

Todas las capas de batch normalization de la ResNet fueron liberadas para poder ser entrenadas, ya que modifican la media y el desvío estándar de los datos procesados en la red y es preferible que calcule estos valores estadísticos para el dataset a ser analizado y no mantener los valores de las imágenes con las que fue pre-entrenada [79]. Ajustar un modelo de DL tiene su parte experimental que se ve afectada directamente por el objetivo del proyecto en cuestión. No obstante, se optó dejar libres las capas de la ResNet en lugar de mantener todo el cuerpo congelado y se lograron mejores resultados como se muestran a continuación.

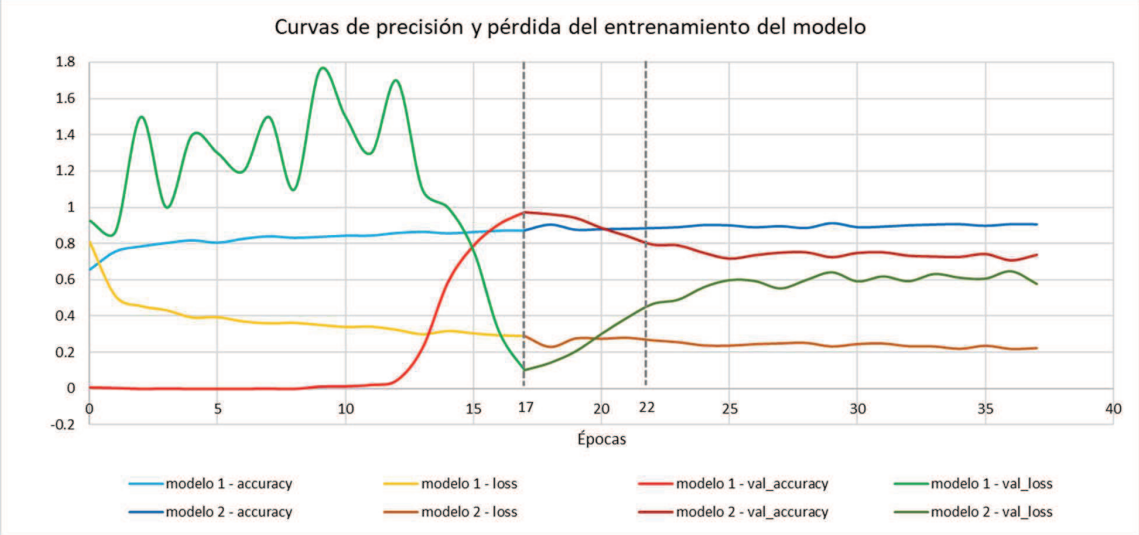


FIGURA 25: CURVAS DE EXACTITUD Y PÉRDIDA DEL ENTRENAMIENTO DEL MODELO COMPLETO. LA GRÁFICA PUEDE DIVIDIRSE EN CUATRO CURVAS: EXACTITUD DE ENTRENAMIENTO (CELESTE/AZUL), FUNCIÓN DE PÉRDIDA O COSTO DE ENTRENAMIENTO (AMARILLO/MARRÓN), EXACTITUD DE VALIDACIÓN (ROJO/BORDO) Y FUNCIÓN DE PÉRDIDA DE VALIDACIÓN (VERDE/VERDE OSCURO). LA RECTA GRIS DE LA IZQUIERDA REPRESENTA EL CHECKPOINT SELECCIONADO DEL MODELO 1 A PARTIR DEL CUAL SE VOLVIÓ A ENTRENAR AL MODELO CON DIFERENTES HIPERPARÁMETROS; MIENTRAS QUE LA RECTA GRIS DE LA DERECHA REPRESENTA EL CHECKPOINT FINAL SELECCIONADO.

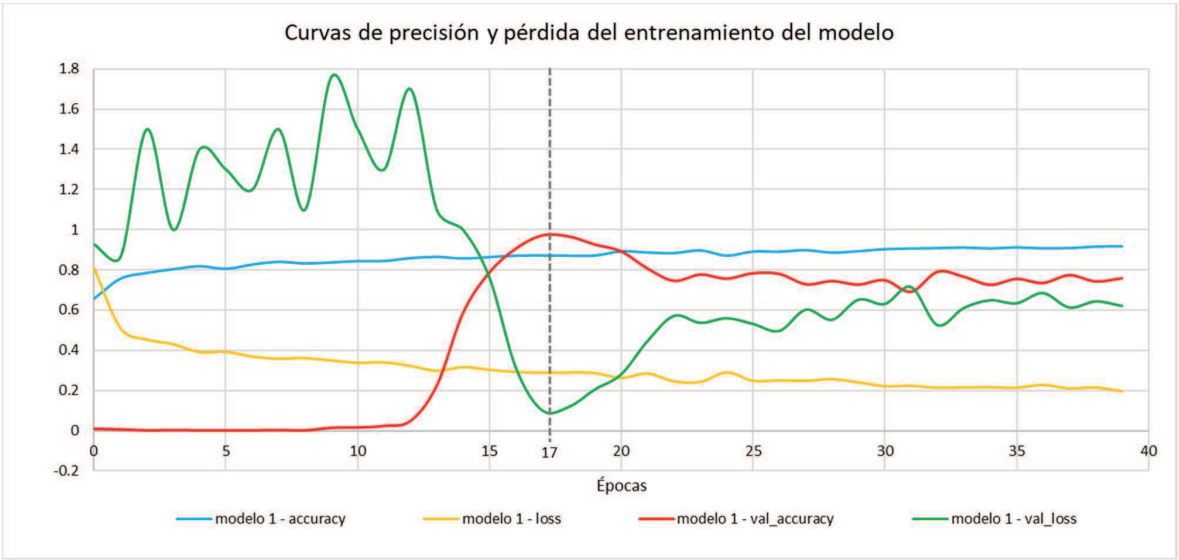


FIGURA 26: CURVAS DE EXACTITUD Y PÉRDIDA DEL ENTRENAMIENTO DEL PRIMER MODELO. LA GRÁFICA PUEDE DIVIDIRSE EN CUATRO CURVAS: EXACTITUD DE ENTRENAMIENTO (CELESTE), FUNCIÓN DE PÉRDIDA O COSTO DE ENTRENAMIENTO (AMARILLO), EXACTITUD DE VALIDACIÓN (ROJO) Y FUNCIÓN DE PÉRDIDA DE VALIDACIÓN (VERDE). LA RECTA GRS MARCA LA ÉPOCA 17, LA CUAL FUE UTILIZADA COMO CHECKPOINT A PARTIR DEL CUAL SE ENTRENÓ LA SEGUNDA PARTE DEL MODELO.

Puede apreciarse que en las primeras 12 épocas las curvas de la Figura 25 poseen un comportamiento errático. Particularmente la curva de la función de pérdida, la cual presenta una oscilación pronunciada, y la curva de exactitud de validación, la cual parece estar estabilizada en 0. Este comportamiento puede indicar que se utilizó una tasa de aprendizaje demasiado alta, donde el algoritmo da saltos largos en la dirección del gradiente, por lo que pierde un mínimo local, intenta volver al mismo en el siguiente paso y posteriormente lo vuelve a perder nuevamente. En simultáneo, como el algoritmo no logra acercarse a dicho punto de manera definitiva, este no aumenta su nivel de aprendizaje, haciendo que la curva de exactitud se mantenga inamovible. No obstante, al haberse aplicado un reductor de la tasa de aprendizaje durante el entrenamiento del modelo finalmente, al finalizar las 40 épocas de entrenamiento, ambas curvas se estabilizan o convergen a un cierto valor Figura 26. Por otro lado, las curvas de la función de pérdida y de exactitud de entrenamiento tuvieron un comportamiento más estable a lo largo de los dos entrenamientos. Se observa que la curva verde presenta un alto nivel de over-fitting a partir de la época 20, razón por la cual se decidió tomar el checkpoint de la época 17 para seguir entrenando el modelo con las modificaciones anteriormente mencionadas.

3.6. Evaluación del algoritmo

Por último, es necesario evaluar la red ya entrenada. Para cada una de las imágenes del conjunto de prueba, se le pide a la red que prediga cuál cree que es la etiqueta de la imagen. Las predicciones del modelo son luego tabuladas. Finalmente, estas se comparan con las etiquetas verdaderas de cada imagen, es decir, con la clase real de pertenencia. A partir de aquí es posible calcular la cantidad de predicciones que el clasificador obtuvo correctamente y calcular informes agregados como exactitud, exhaustividad y varias otras métricas que se utilizan para cuantificar el rendimiento de la red en su conjunto [38].

3.6.1. Métricas

Para construir las métricas de evaluación es necesario primero recordar algunas medidas elementales. Al momento de evaluar el desempeño de un algoritmo de clasificación existen varias métricas diferentes. A continuación, se mencionan aquellas que fueron utilizadas en este proyecto diagnóstico.

Matriz de confusión

La matriz de confusión, también conocida como tabla de contingencia, muestra el desempeño de un algoritmo de clasificación [9], describiendo cómo se distribuyen las predicciones y sus valores reales:

Positivo verdadero (TP): es la correcta clasificación de la clase positiva. Por ejemplo, cuando la imagen corresponde a la clase "maligna" y el modelo clasificó correctamente [9], [36].

Negativo verdadero (TN): es la correcta clasificación de la clase negativa. Por ejemplo, cuando la imagen corresponde a la clase "benigna" y el modelo clasificó correctamente [9], [36]

Falso positivo (FP): es la clasificación incorrecta de la clase positiva. Por ejemplo, cuando la imagen corresponde a la clase "benigna" y el modelo clasificó incorrectamente como "maligna" [36].

Falso negativo (FN): es la clasificación incorrecta de la clase negativa. Por ejemplo, cuando la imagen corresponde a la clase "maligna" y el modelo clasificó incorrectamente como "benigna" [9], [36].

A partir de la matriz es posible calcular otras métricas, tales como la precisión, la exhaustividad, la exactitud y la especificidad

Precisión / *Precision*: muestra la relación entre los elementos positivos predichos correctamente y el total de elementos positivos predichos [9], [36].

$$Precisión = \frac{TP}{TP + FP} \quad \text{(ECUACIÓN 6)}$$

Exactitud / *Accuracy*: determina la relación entre las predicciones correctas (TP y TN) y las predicciones totales. Un clasificador ideal tendría una exactitud de 1 ya que todas las muestras serían clasificadas correctamente [9], [36]:

$$Exactitud = \frac{TP + TN}{TP + TN + FN + FP} \quad \text{(ECUACIÓN 7)}$$

Exhaustividad / Tasa de verdaderos positivos (TPR) / *Recall* / *Sensitivity:* muestra la relación entre los verdaderos positivos (positivos detectados) y los positivos reales (los detectados y los no detectados). Un clasificador ideal tendría una exhaustividad de 1, pues todos los TP serían detectados como tal [9], [36]. Cuanto mayor sea la exhaustividad de un examen,

más enfermos serán diagnosticados adecuadamente, con lo que la tasa de falsos negativos será menor.

$$Exhaustividad = TPR = Recall = Sensitivity = \frac{TP}{TP + FN} \quad \text{(ECUACIÓN 8)}$$

Especificidad / Tasa de verdaderos negativos (TNR) / *Specificity*: muestra la relación entre los verdaderos negativos y los negativos reales (los detectados y los no detectados) [9], [36]. Cuanto mayor sea la especificidad del examen, más sanos serán diagnosticados adecuadamente, con lo que la tasa de falsos positivos será menor.

$$Especificidad = TNR = Specificity = \frac{TN}{TN+FP} \quad \text{(ECUACIÓN 9)}$$

Curva ROC

Las curvas de características operativas del receptor o curvas ROC (por su término en inglés: Receiver Operating Characteristic) representan el rendimiento del modelo para todos los umbrales de clasificación. Es el gráfico de la tasa de verdaderos positivos frente a la tasa de falsos positivos [36].

Los clasificadores binarios calculan una probabilidad de la presencia de un hallazgo de interés. Para obtener una salida binaria (como benigno/maligno) se aplica un punto de corte o umbral donde las imágenes con puntuaciones que lo superan se clasifican dentro de dicha clase. Sin embargo, puede ser de interés representar varias opciones de "umbrales" simultáneamente [9].

Las curvas ROC muestran la relación entre la sensibilidad y la especificidad de una prueba: sensibilidad (o TPR) en función del complemento de la especificidad (1-especificidad o tasa de falsos positivos, FPR). De tomar un punto donde tanto la TPR como la FPR sean bajas, se clasificará a un menor número de lesiones como "malignas" y se asociará con medidas más altas de especificidad (mayor proporción de lesiones benignas correctamente identificadas) y menor sensibilidad. En el otro extremo, de tomar un punto donde tanto la TPR como la FPR son altas, se clasificarán más lesiones como "malignas", aumentando la probabilidad de que una lesión maligna se identifique como tal (mayor sensibilidad), a expensas de más falsos positivos (menor especificidad). Se observa que, a mayor sensibilidad, menor especificidad, y viceversa [9].

El área bajo la curva ROC, también conocida como AUC o AUC ROC refleja la capacidad de discriminación de un clasificador binario. Mientras que un clasificador perfecto tendría un AUC

igual a 1, un clasificador con un AUC igual a 0.5 tendrá la misma capacidad de predicción que dejar la clasificación al azar [9], [36]. Su amplia utilización se debe a que es invariante en escala, lo que significa que verifica qué tan bien predice el modelo en lugar de verificar los valores absolutos; y, además porque comprueba el rendimiento del modelo independientemente del umbral elegido [10]. El AUC se usa para comparar pruebas diagnósticas de clasificación binaria: puede calcularse la diferencia entre las AUC de diferentes pruebas para una misma tarea y con eso ofrecer una comparación cuantitativa de su desempeño [9].

En la práctica es necesario elegir un único umbral para definir cuándo clasificar una observación dentro o fuera de la clase de interés. Dicha elección está sujeta a consideraciones de diferentes tipos. Aunque existen fórmulas que ayudan a definir cuál es el valor que ofrece un mejor balance entre sensibilidad y especificidad, la elección también debería estar sujeta a consideraciones clínicas. Muchas veces las fórmulas que optimizan sensibilidad y especificidad asumen que la prevalencia del hallazgo de interés es del 50%, y que los resultados falsos positivos tienen la misma gravedad que los falsos negativos. En la práctica clínica, este raramente es el caso: muchas veces, un resultado falso negativo puede ser más grave que un falso positivo, o viceversa.

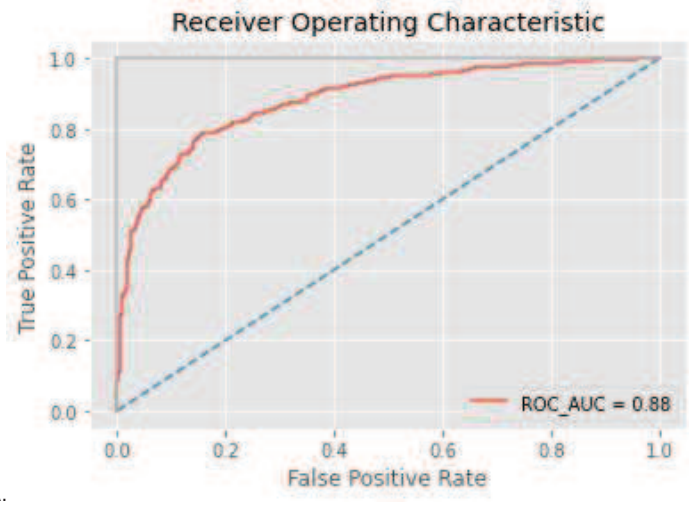
Curva PR

Otra curva comúnmente utilizada para evaluar los algoritmos de DL es la curva de *precision-recall* (PR). Tal como lo indica su nombre, grafica la relación entre la precisión y el recall o exhaustividad para cada valor de umbral. Esta curva permite calibrar el modelo para lograr valores de precisión y recall aceptables. Cuanto más se acerque la curva al punto (1, 1), mayor es la capacidad de detección del modelo en términos generales (independientemente del valor del umbral de decisión). Por ello, a la hora de comparar modelos se suele utilizar el área bajo la curva *precision-recall* (PR-AUC), también conocida como precisión promedio [9], [80].

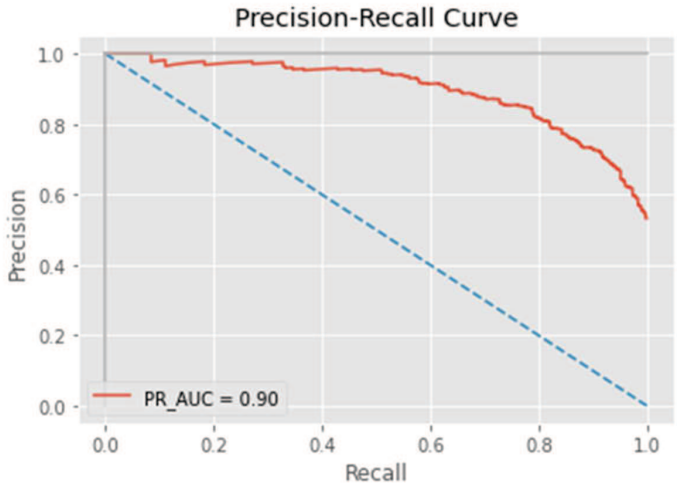
3.6.2. Resultados

La Figura 27 muestra las curvas ROC y PR obtenidas del modelo entrenado, junto a sus valores AUC respectivos. En ambos casos la curva azul representa a un clasificador con un AUC igual a 0.5. El mejor escenario posible, en el caso de la primera figura (A), es representado por una línea completamente vertical desde el punto (0,0) seguido de una línea completamente horizontal a partir del punto (0,1). En el caso de la segunda figura (B), es representado por una

línea completamente horizontal desde el punto (0,1), seguido de una línea completamente vertical a partir del punto (1, 1) hasta el punto (0,1).



A.



B.

FIGURA 27: A) CURVA ROC, JUNTO A SU VALOR ROC_AUC; B) CURVA PR, JUNTO A SU VALOR PR_AUC. EN AMBOS CASOS LA CURVA ROJA ES LA DE INTERÉS, MIENTRAS QUE LA CURVA AZUL REPRESENTA UNA CURVA CON AUC = 0.5. LA MISMA SE COLOCA A MODO DE CONTRASTE.

Puede observarse que en ambos casos las curvas se acercan más al caso ideal que a la línea azul. Además, si se comparan los valores AUC, esta observación se hace más evidente. Donde el ROC AUC es de 0.88 y el PR AUC de 0.9. No obstante, es necesario calcular otras métricas para evaluar el desempeño del modelo.

Utilizando el umbral por defecto, aquellas predicciones mayores a 0.5 fueron clasificadas como malignas y de lo contrario como benignas, obteniéndose de esta manera la matriz de confusión de la Figura 28 y las métricas de la Tabla 5.

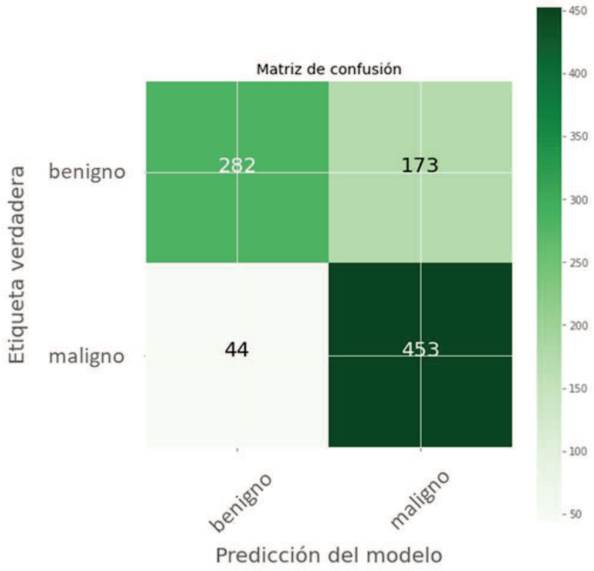


FIGURA 28: MATRIZ DE CONFUSIÓN CALCULADA EN LA ETAPA DE EVALUACIÓN DEL MODELO, UTILIZANDO UN UMBRAL = 0.5. LAS COLUMNAS REPRESENTAN LA CLASE PREDICHA POR EL ALGORITMO DE CLASIFICACIÓN, MIENTRAS QUE LAS FILAS REPRESENTAN LAS ETIQUETAS VERDADERAS O DE *GROUND-TRUTH*. LAS MÉTRICAS OBTENIDAS DIRECTAMENTE DE LA MATRIZ SON: TN = 282, FN = 173, FP = 44 Y TP = 453.

TABLA 5: MÉTRICAS OBTENIDAS UTILIZANDO UN UMBRAL = 0.5

Métrica	Valor	Métrica	Valor
Exhaustividad (TPR)	91.15%	FNR (1 - TPR)	8.85%
Especificidad (TNR)	61.98%	FPR (1 - TNR)	38.02%
	Exactitud	77.20%	

A partir de la matriz de confusión es posible extraer de manera directa las siguientes métricas: TN = 282, FN = 173, FP = 44 y TP = 453. Para poder obtener una mejor percepción del desempeño del algoritmo, es necesario combinar estas métricas utilizando las ecuaciones 8 y 9. De esta manera se calculó una exhaustividad o TPR del 91.15% y una especificidad del 61.98%. A su vez fue posible calcular la tasa de falsos negativos, que se calcula con el complemento de la TPR, y la tasa de falsos positivos, calculada con el complemento de la TNR. Por último, utilizando la ecuación 7 fue posible calcular la exactitud con un valor del 77.20%

Como este proyecto abarca una patología clínica donde su falta de detección es el peor escenario, la exhaustividad es la métrica con mayor importancia al momento de evaluar el modelo. Es decir, es preferible que el modelo detecte erróneamente una lesión benigna a que no detecte correctamente cuando la misma es maligna. En el primer caso al paciente se le recomendaría extraer una lesión potencialmente maligna pero que acaba por ser benigna, mientras que en el segundo caso al paciente no se le realizaría la extracción de una lesión que efectivamente era maligna, con consecuencias graves para el desarrollo de la enfermedad.

La Figura 29 muestra la distribución de las predicciones realizadas por el modelo. Los colores indican la pertenencia de la imagen a la clase de *ground-truth*, azul para la clase negativa (benigna) y naranja para la clase positiva (maligna). El eje de las predicciones es dividido por la mitad, a la altura del valor 0,5 por una recta gris punteada que denota el umbral establecido. Esta recta separa las predicciones negativas (lado izquierdo) de las positivas (lado derecho). Moviendo el umbral hacia la derecha, más lesiones benignas serían identificadas como tal a expensas de identificar correctamente menos lesiones malignas. Inversamente, al correr el umbral hacia la izquierda, más lesiones malignas serían identificadas como tal a expensas de identificar correctamente menos lesiones benignas. En otras palabras, modificando el umbral de detección es posible obtener una exhaustividad mayor, a cambio de precisión (siempre y cuando las clases estén correctamente aprendidas). En la Sección 4.2. se profundiza respecto al umbral de detección seleccionado.

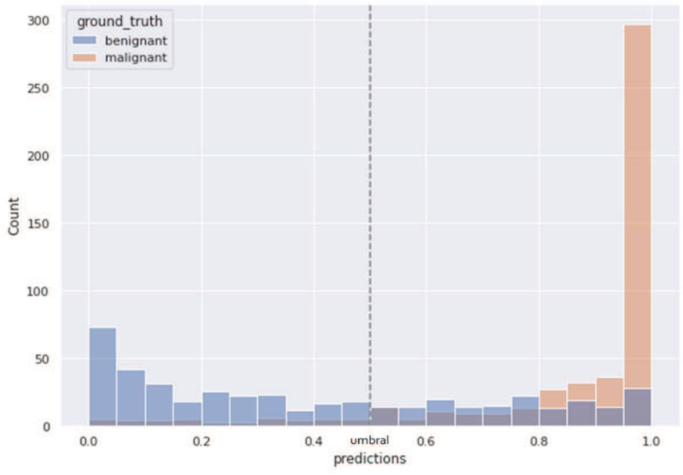


FIGURA 29: PREDICCIONES REALIZADAS POR EL MODELO. EN AZUL PUEDEN OBSERVARSE LAS IMÁGENES CON ETIQUETA DE *GROUND-TRUTH* "BENIGNA" Y EN NARANJA AQUELLAS CON ETIQUETA DE *GROUND-TRUTH* "MALIGNA". LOS MARCADORES CIRCULARES MARCAN LAS IMÁGENES QUE FUERON PREDICHAS "MALIGNAS" Y AQUELLAS CON FORMA DE CRUZ "BENIGNAS". LA LÍNEA PUNTEADA HORIZONTAL IDENTIFICA EL UMBRAL POR DEFINICIÓN DE 0.5.

Comparación con médicos dermatólogos

Para poder asegurar el desempeño del algoritmo es necesario saber si es comparable en cuanto a la capacidad de clasificación de los médicos dermatólogos. De esta manera se optó por comparar las métricas del modelo de este proyecto frente a las métricas de dermatólogos profesionales, obtenidos de la bibliografía. Los mismos se detallan en la Tabla 6.

TABLA 6: MÉTRICAS DE EXHAUSTIVIDAD, ESPECIFICIDAD Y ROC AUC DE DIFERENTES GRUPOS DE DERMATÓLOGOS EXTRAÍDAS DE LA BIBLIOGRAFÍA.

Autor	Exhaustividad	Especificidad	ROC AUC
Brinker et al. [52]	67.20%	62.20%	NA
Brinker et al. [51]	74.10%	60.00%	NA
Haenssle *et al.* [44]	86.60%	71.30%	0.79
Marchetti *et al.* [46]	82.00%	59.00%	0.71
Rezvantalab *et al.* [65]	NA	NA	0.82
Promedio calculado	77.46%	63.13%	0.77
Modelo desarrollado	91.15%	61.98%	0.88

NA: No Aplica, métrica no calculada en el trabajo

El trabajo de Haenssle *et al.* [44] determina que los valores de exhaustividad calculados para 58 dermatólogos de diferentes niveles de estudio, es del 86.60%, mientras que la especificidad es del 71.30%. Esto se tradujo a un valor ROC AUC promedio de 0.75 para los profesionales principiantes, mientras que para los profesionales expertos se tradujo a un valor del 0.82, obteniéndose de esta manera un valor ROC AUC promedio del 0.79.

Por otro lado, trabajos como los de Brinker *et al.* [51], [52] demostraron que 157 dermatólogos tuvieron un desempeño de exhaustividad del 67.20% y una especificidad del 62.20%. Posteriormente, en otra investigación [51], una exhaustividad del 74.10% y una especificidad del 60.00%. En este último trabajo, los médicos con mayor experiencia mostraron una especificidad media mayor al 69,2% con una exhaustividad media del 73,3%; mientras que los médicos principiantes mostraron una exhaustividad del 68,9% y una especificidad del 58%.

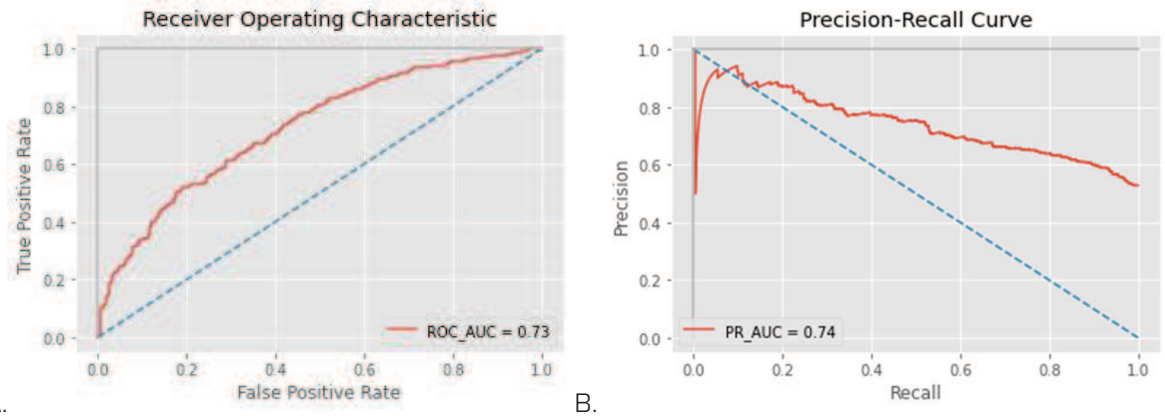
De esta manera fue posible estimar una exhaustividad media para los dermatólogos del 77.46%, una especificidad media del 63.13% y un valor ROC AUC del 0.77. Al comparar estas métricas a aquellas obtenidas del algoritmo desarrollado en este proyecto, puede observarse

una mejora en cuanto a su exhaustividad, mientras que, para la especificidad, la métrica del modelo supera dos de los cuatro casos presentados, y de considerar las métricas de Brinker *et al.* [51] para especialistas con diferentes niveles de experiencia, el algoritmo supera a la especificidad de los médicos principiantes, pero no la de los médicos con mayor experiencia.

Comparación con un modelo de línea de base

Otra manera de evaluar el desempeño de un algoritmo es comparándolo con otro que marque una línea de base la cual debería ser superada por el modelo desarrollado. En este caso se optó por generar un clasificador simple compuesto por 5 capas no convolucionales, entrenado en el mismo set de datos que el algoritmo principal. Para entrenarlo se utilizó el optimizador SGD con un batch size de 64. El algoritmo fue entrenado en dos partes, la primera utilizando un learning rate de 0.001 hasta la época 40, y la segunda, conectada a partir del último checkpoint, con un learning rate de 0.0001 por otras 40 épocas. El checkpoint final elegido fue el último del entrenamiento.

Se obtuvieron las curvas ROC y PR para este algoritmo, junto a sus valores AUC (Figura 30). En primera instancia es posible comparar los algoritmos mediante sus valores AUC. Tanto para el ROC AUC como para el PR AUC, el modelo principal desarrollado superó al modelo de línea de base (Tabla 7). En el caso de la ROC AUC por una diferencia de 0.15 puntos, mientras que para el caso de la PR AUC, por una diferencia del 0.16.



A. B.

FIGURA 30: A) CURVA ROC, JUNTO A SU VALOR ROC_AUC; B) CURVA PR, JUNTO A SU VALOR PR_AUC PARA EL MODELO DE LÍNEA BASE. EN AMBOS CASOS LA CURVA ROJA ES LA DE INTERÉS, MIENTRAS QUE LA CURVA AZUL REPRESENTA UNA CURVA CON AUC = 0.5. LA MISMA SE COLOCA A MODO DE CONTRASTE.

TABLA 7: COMPARACIÓN DE VALORES AUC ENTRE EL MODELO PRINCIPAL Y EL MODELO DE LÍNEA DE BASE

Métrica	Modelo principal (Mp)	Modelo de línea de base (Mb)	Δ (Mp – Mb)
ROC AUC	0.88	0.73	0.15
PR AUC	0.90	0.74	0.16

Empleando un umbral del 0.5, se calculó una matriz de confusión para el segundo modelo. De ella fue posible obtener las métricas previamente calculadas para el primer modelo (Tabla 5). La Tabla 8 compara los valores de exhaustividad, especificidad y exactitud entre los dos algoritmos. Puede observarse que las métricas del modelo principal superan a aquellas del modelo de línea base. En el caso de la exhaustividad por 4.43%, en el de la especificidad por un 21.98% y en el de la exactitud por un 10.20%.

TABLA 8: COMPARACIÓN DE MÉTRICAS ENTRE EL MODELO PRINCIPAL Y EL MODELO DE LÍNEA DE BASE, UMBRAL = 0.5.

Métrica	Modelo principal (Mp)	Modelo de línea de base (Mb)	Δ (Mp – Mb)
Exhaustividad (TPR)	91.15%	86.72%	4.43%
Especificidad (TNR)	61.98%	40.00%	21.98%
Exactitud	77.20%	67.00%	10.20%

Comparación con algoritmos de la bibliografía

Por último, es posible comparar el desempeño del algoritmo desarrollado con aquellos encontrados en la bibliografía. En la Tabla 9 se enuncian las métricas para diferentes algoritmos, junto al tamaño del dataset empleado en cada proyecto. Puede observarse que los resultados de la bibliografía son muy variados y no todos exponen las mismas métricas de evaluación. Por un lado, la exhaustividad del modelo desarrollado en este proyecto supera a la mayoría de los algoritmos de la Tabla 9, pero por el otro lado, tanto el valor ROC AUC como la especificidad es superada en todos los casos. Además, es llamativo que si bien el trabajo de Han et al. [49] muestra tener muy buenas métricas de exhaustividad, especificidad y valor ROC AUC, pero la de exactitud es relativamente baja, de tan solo 56.50%. Cabe aclarar que los valores de estas métricas (salvo la ROC AUC) se modifican según el umbral de clasificación elegido, y en esta oportunidad las métricas del modelo desarrollado fueron obtenidas utilizando el umbral por definición (0.5).

A pesar de no superar todas las métricas de los algoritmos encontrados en la bibliografía, el algoritmo desarrollado en este proyecto sí supera aquellas de los médicos dermatólogos. El objetivo planteado es principalmente aportar una segunda opinión en cuánto al diagnóstico de una lesión maligna, donde la línea base es que este sea comparable a la capacidad de clasificación de un médico dermatólogo. Tomando los resultados de las comparaciones realizadas, es posible decir que el modelo desarrollado en este proyecto es útil para la asistencia en la toma de decisiones clínicas en el diagnóstico de lesiones melanocíticas benignas o malignas.

TABLA 9: COMPARACIÓN DE MÉTRICAS DE ALGORITMOS EXTRAÍDOS DE LA BIBLIOGRAFÍA FRENTE A AQUELLAS DEL MODELO DESARROLLADO.

Autor	Tamaño dataset	Exhaustividad	Especificidad	Exactitud	ROC AUC
Haenssle *et al.* [44]	100.000	95.00%	80.00%	NA	0.95
Han *et al.* * [49]	19.398	85.75%	83.4%	56.50%	0.90
Brinker et al. * [51]	12.378	74.10%	86.5%	NA	NA
Hekler *et al.* * [47]	11.444	86.1%	89.2%	81.59%	NA
Rezvantalab *et al.* * [65]	10.135	NA	NA	NA	0.94
Brinker et al. [52]	4.204	82.30%	77.90%	NA	NA
Modelo desarrollado	3.808	91.15%	61.98%	77.20%	0.88

* Utilizaron alguna variación de la arquitectura ResNet como arquitectura de su modelo

NA: No Aplica, métrica no calculada en el trabajo

3.7. Versión final del algoritmo

La versión final del algoritmo que posteriormente es utilizada en la aplicación fue desarrollada utilizando la arquitectura ResNet50 pre-entrenada con el conjunto de datos ImageNet y utilizando como cabeza un conjunto de capas secuenciales compuesta por una capa flatten, seguida por una capa de batch normalization, una capa densa con 256 nodos de salida con una función de activación ReLu. A esta capa le sigue una capa dropout con una probabilidad del 0.5, otra capa de batch normalization y finalmente la capa de salida densa con una función de activación sigmoide.

La red fue entrenada en dos instancias donde en la primera se entrenó por 17 épocas solamente la cabeza, manteniendo congeladas todas las capas de la ResNet excepto las capas de tipo batch normalization. Se utilizó el optimizador Adam, con una tasa de aprendizaje de

0.001 y un tamaño de lote de 32. Tomando como checkpoint este último punto, se volvió a entrenar la red descongelando el último bloque convolucional de la ResNet y utilizando una tasa de aprendizaje igual a 0.0001. El punto seleccionado para ser utilizado en la aplicación fue el de la época 5 de este entrenamiento.

Las métricas AUC obtenidas para este modelo fueron 0.88 para la ROC AUC y 0.90 para la ROC PR. Por otro lado, se calcularon los valores de exhaustividad (TPR), especificidad (TNR) y exactitud, siendo estos 91.15%, 61.98% y 77.20% respectivamente.

4. Diseño y desarrollo de DermoAnálisis

Para aplicar el algoritmo descrito en el capítulo anterior, se diseñó y desarrolló una aplicación: *DermoAnálisis*. Este software busca asistir a los médicos dermatólogos al momento de analizar y seleccionar el tratamiento adecuado para una lesión dermatológica melanocítica. Para mejorar la usabilidad de la interfaz de usuario se entrevistó a un médico dermatólogo. Las sugerencias aportadas por el profesional serán mencionadas en la Sección 4.2.

La aplicación permite ingresar una imagen de una lesión cutánea y posteriormente calcular una probabilidad diagnóstica asociada a la misma que indicará qué tan probable es que dicha lesión pertenezca a la clase "benigna" o "maligna". A su vez, permite gestionar los datos de los pacientes mediante una base de datos diseñada específicamente para ello, y administrar tanto los informes clínicos realizados mediante el programa, como las imágenes de las lesiones cargadas al mismo.

4.1. Arquitectura de la GUI

La interfaz gráfica desarrollada (Graphical User Interface, GUI), permite mediante la utilización de elementos gráficos como iconos, menús e imágenes, facilitar el acceso del usuario a toda la información disponible. Este tipo de interfaces se basan en la programación basada en eventos. Es decir, al accionar un botón o campo, se genera una señal de evento que activa y ejecuta una función o *callback* asociado a la misma [81]. A modo ejemplo, un evento sería el click del botón del ratón sobre un botón de la interfaz, y el callback sería guardar un archivo.

El *frontend* o diseño de la GUI fue realizado empleando *Qt Designer*, un programa de libre acceso basado en el lenguaje *Qt*; mientras que la funcionalidad o backend se implementó utilizando el lenguaje *open-source* Python versión 3.6.10. Principalmente se utilizaron dos librerías: *PyQt5* para programar las señales de eventos y *SQLite* para el manejo de bases de datos.

La interfaz cuenta con cuatro pestañas. Cada una de ellas permite realizar diferentes acciones las cuales serán mencionadas a continuación (para mayor detalle, remitirse al Anexo 3: Manual de Usuario).

Pestaña 1)	Pacientes	Pestaña 3)	Historial de consultas
Pestaña 2)	Editar pacientes	Pestaña 4)	Nuevo análisis

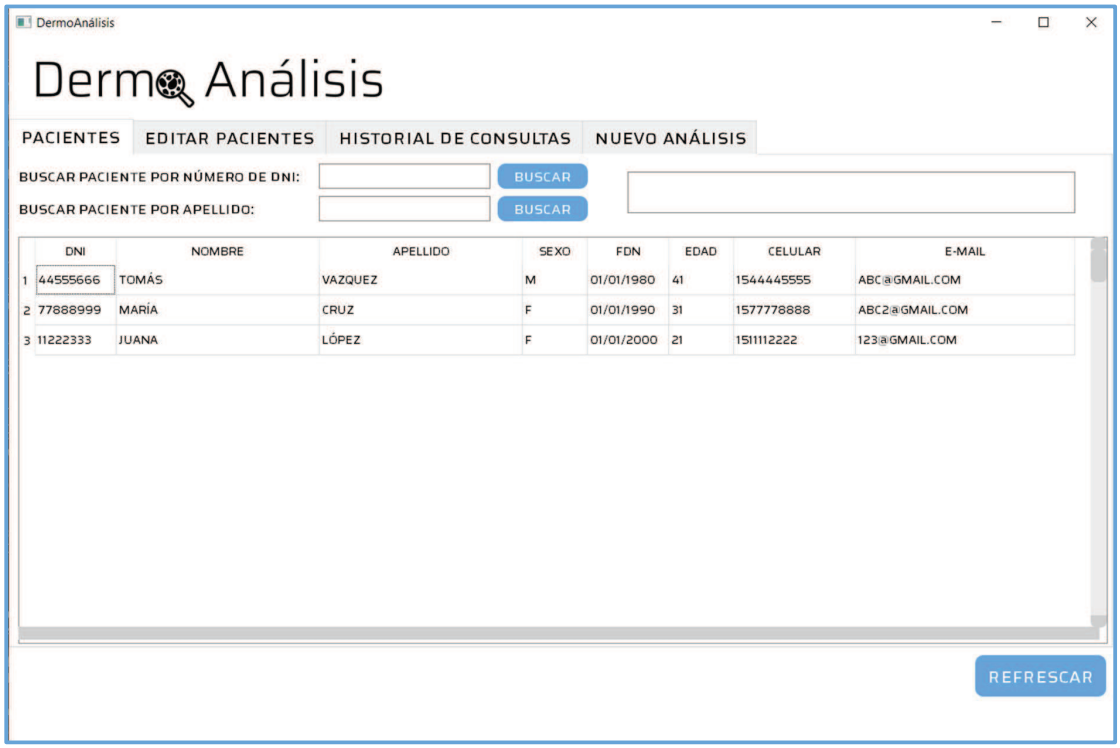


Figura 31: Interfaz de usuario – DermoAnálisis – Pestaña 1: PACIENTES

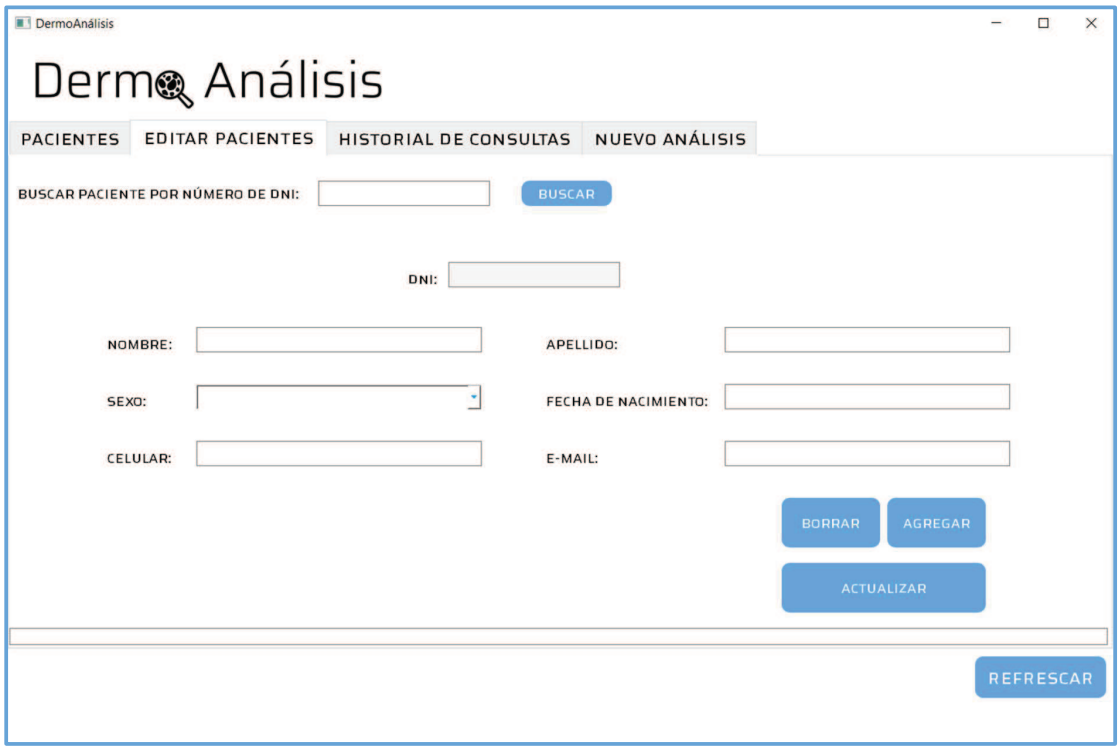


Figura 32: Interfaz de usuario – DermoAnálisis – Pestaña 2: EDITAR PACIENTES

La primera pestaña, denominada PACIENTES (Figura 31), permite visualizar una tabla con los datos de los pacientes. Para facilitar la búsqueda de un registro en particular, es posible utilizar los buscadores de paciente por DNI o por Apellido ubicados por encima de la tabla.

La segunda pestaña, denominada EDITAR PACIENTES (Figura 32), permite agregar, actualizar y borrar registros de la tabla de la pestaña 1. Para facilitar las últimas dos acciones posee un buscador por DNI.

La tercera pestaña, denominada HISTORIAL DE CONSULTAS (Figura 33), permite visualizar una tabla que resume las consultas realizadas mediante la aplicación DermoAnálisis. Para facilitar la búsqueda de un registro en particular, es posible utilizar los buscadores de paciente por DNI o por Apellido ubicados por encima de la tabla. Además, se puede acceder al informe emitido por cada análisis y a la imagen relacionada al mismo seleccionado la palabra PDF o IMAGEN, respectivamente. Por último, permite borrar un registro al pulsar sobre la palabra BORRAR de la fila correspondiente.

Finalmente, la cuarta pestaña, denominada NUEVO ANÁLISIS (Figura 34), permite analizar una lesión de piel mediante el algoritmo de clasificación desarrollado. Para ello es necesario acceder a la imagen y posteriormente pulsar el botón ANALIZAR IMAGEN. Una vez terminado el procesamiento, los campos de Resultados del Análisis se actualizarán con un valor numérico y otro categórico, indicando al profesional la probabilidad diagnóstica de pertenencia a la clase maligna obtenida de la lesión. Al finalizar el ingreso de los datos, y al pulsar el botón AGREGAR, se generará un registro en la tabla de la pestaña 3 y una carpeta de análisis en la carpeta del paciente. Esta última contendrá un informe PDF generado con la información obtenida de la pestaña 4, junto a una copia de la imagen seleccionada.

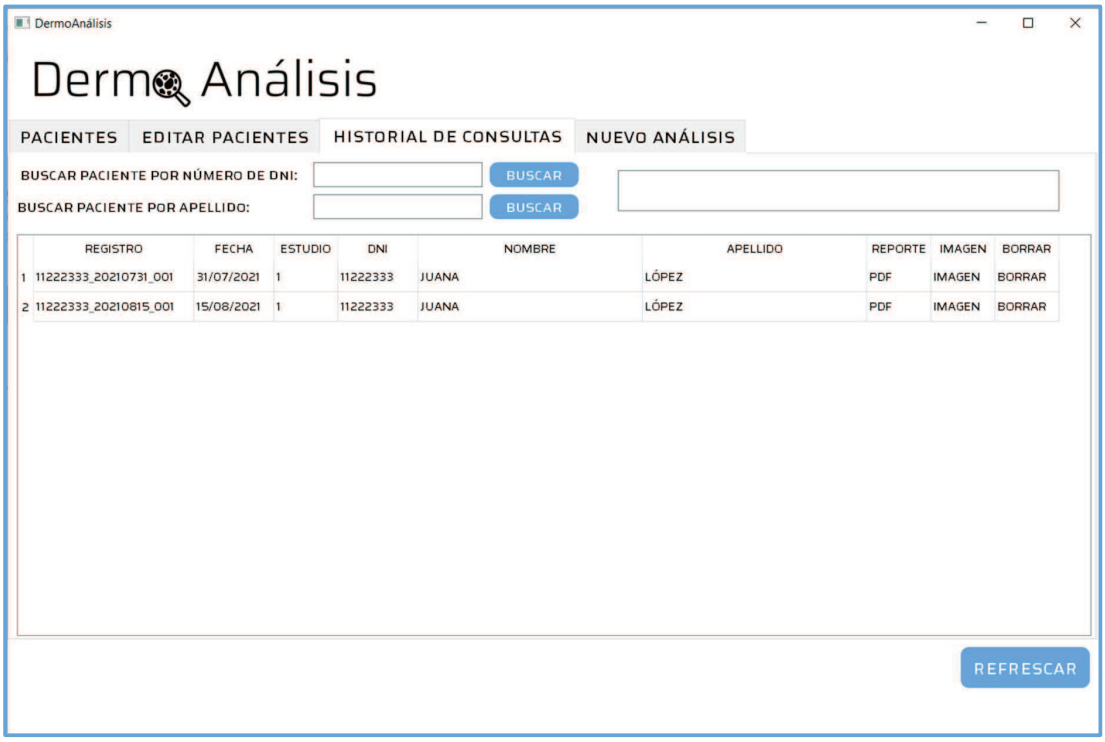


FIGURA 33: INTERFAZ DE USUARIO – DERMOANÁLISIS – PESTAÑA 3: HISTORIAL DE CONSULTAS

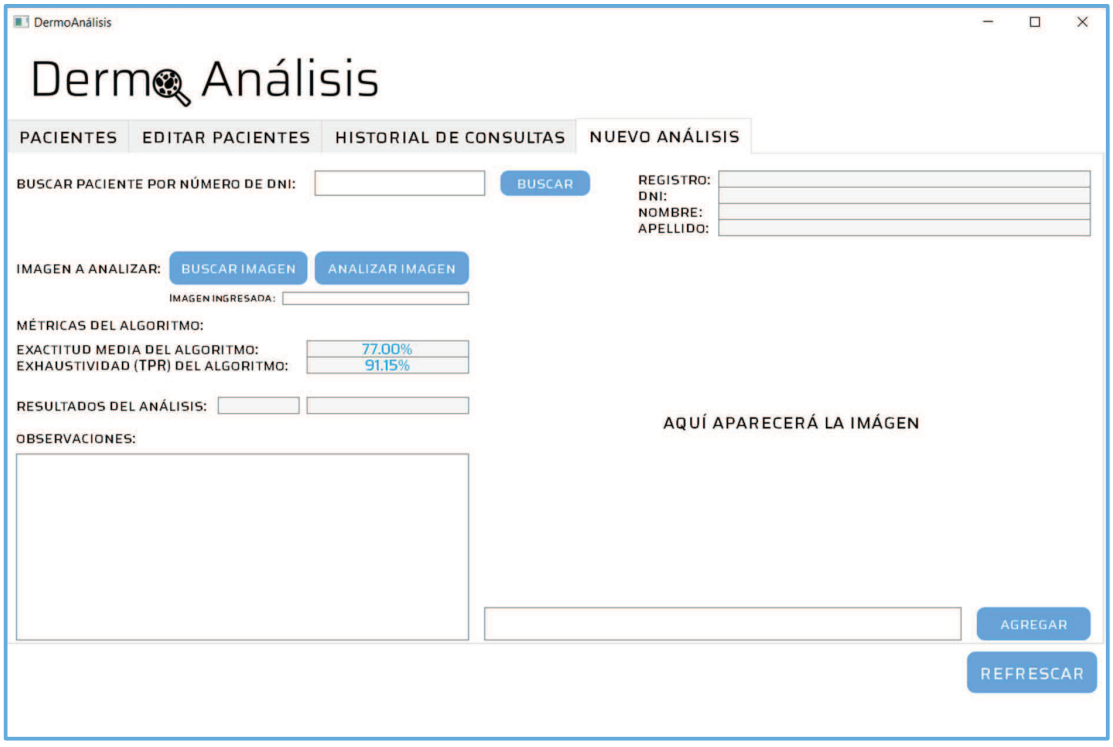


FIGURA 34: INTERFAZ DE USUARIO – DERMOANÁLISIS – PESTAÑA 4: NUEVO ANÁLISIS

4.1.1. Flujo de trabajo de la aplicación

El flujo de trabajo general del programa se describe mediante los diagramas a continuación (Figura 35, Figura 36, Figura 37). En el mismo se indica a qué pestaña recurrir en caso de llevar a cabo una acción de generar, modificar o visualizar un registro. Mayor detalle las acciones será desarrollado en las secciones correspondientes a cada pestaña

De no existir un paciente, es necesario generar un “registro paciente” nuevo en la pestaña 2. De lo contrario, el registro deberá encontrarse en la tabla de la pestaña 1. En caso de necesitar actualizar los datos de un paciente, se deberá buscar dicho paciente en la pestaña 2 y modificar los elementos deseados. Si se requiere generar un análisis o una consulta nueva para un paciente determinado, se deberá acceder a la pestaña 4. Una vez generado dicho registro, el acceso a su información se encontrará disponible desde la pestaña 3.

De necesitar eliminar un registro de un análisis, es necesario ir a la pestaña 3, buscar el registro de interés y presionar la palabra BORRAR en la fila correspondiente. Esta acción no solo eliminará el registro de la tabla registros, sino también la carpeta asociada a dicho registro. En caso de querer eliminar a un paciente, desde la pestaña 2 se busca al paciente en cuestión y se presiona el botón BORRAR. Esta acción eliminará el “registro paciente” de la tabla pacientes, todos los registros de análisis en la tabla registros, y la carpeta principal del paciente, junto a todas las subcarpetas de análisis efectuados.

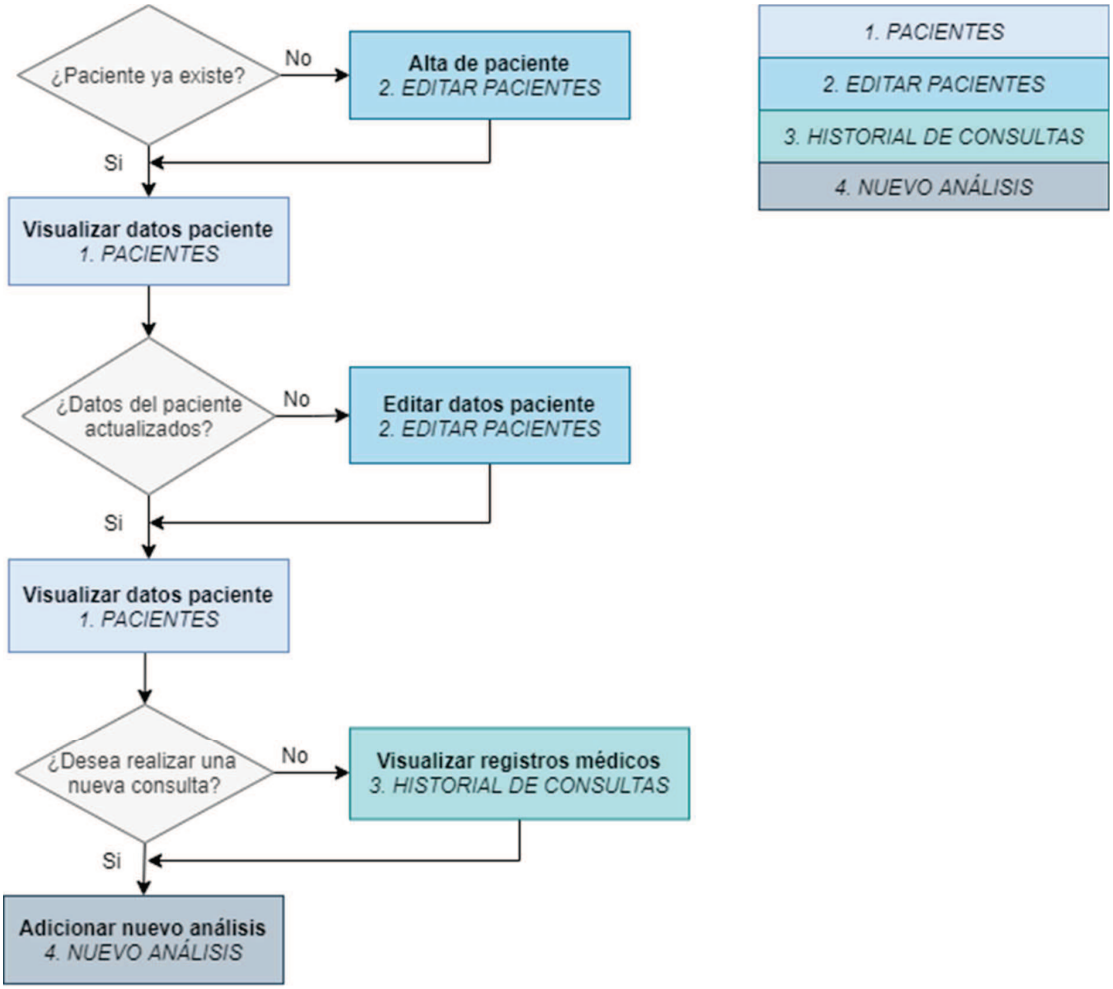


FIGURA 35: DIAGRAMA EN BLOQUES DEL FLUJO GENERAL DE TRABAJO DE DERMOANÁLISIS, PARTE A.

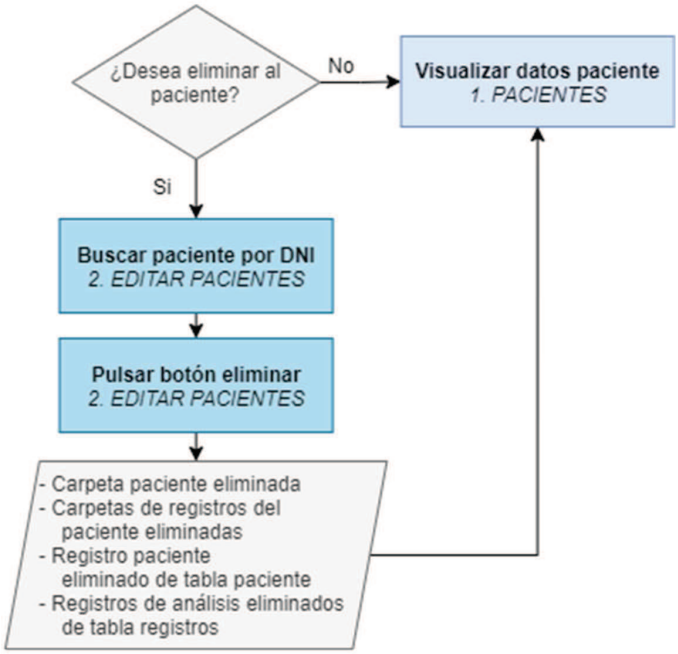


FIGURA 36: DIAGRAMA EN BLOQUES DEL FLUJO GENERAL DE TRABAJO DE DERMOANÁLISIS, PARTE B.

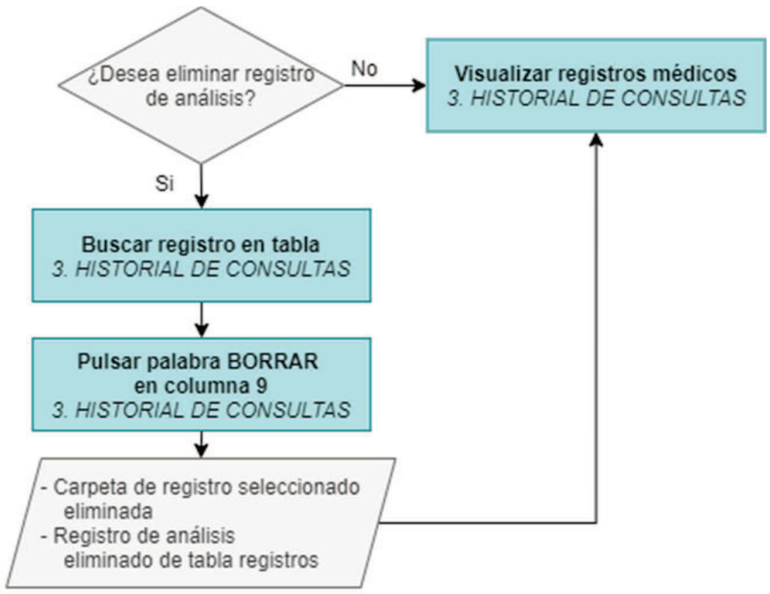


FIGURA 37: DIAGRAMA EN BLOQUES DEL FLUJO GENERAL DE TRABAJO DE DERMOANÁLISIS, PARTE C.

4.1.2. Generación de directorios y registros

El programa trabaja con una base de datos, programada en *DB Browser for SQLite*, compuesta por dos tablas (Figura 38): tabla *pacientes* y tabla *registros*. La primera guarda los datos de los pacientes: DNI, Nombre, Apellido, Sexo, Fecha de Nacimiento, Edad, Celular e E-mail; mientras que la segunda guarda los datos asociados al análisis realizado: Registro, Fecha, Estudio, DNI, Nombre, Apellido, Documento, Imagen y Borrar. Las características particulares de cada uno de estos atributos se encuentran detallados en el Anexo 3.

Para cada paciente deberá existir un único registro en la tabla *pacientes*; mientras que en la tabla *registros*, podrán existir tantos registros cómo análisis realizados con *DermoAnálisis*. Cabe destacar que los últimos tres elementos de la tabla *registros* permiten realizar acciones sobre los datos asociados a ese mismo registro: al pulsar sobre la palabra *PDF* o sobre la palabra *IMAGEN*, se abrirán respectivamente el informe y la imagen asociados a dicho registro. Al seleccionar la palabra BORRAR, se borrará dicho elemento de la tabla y la carpeta vinculada al mismo.

A su vez, el programa permite gestionar los análisis en un sistema de carpetas, tal como se observa en la Figura 39. Al instalar el programa en la computadora del usuario (médico), se establece un directorio principal. Al adicionar pacientes en la tabla *pacientes*, se generará una carpeta con el número de DNI del paciente en el directorio principal seleccionado. A medida que el paciente se realice estudios mediante DermoAnálisis, se generarán nuevas carpetas con el número de registro correspondiente a los análisis efectuados. Dentro de estas últimas carpetas se guardarán el informe de la consulta y la imagen analizada. Estos documentos pueden accederse directamente desde la pestaña 3 del programa, seleccionado PDF o IMAGEN del registro correspondiente. A su vez, el profesional tiene la capacidad de acceder al directorio desde la computadora. Esto le permite manipular los documentos en su interior y adicionar más elementos en caso de desearlo. De modificar los nombres y/o la ubicación de los archivos generados por la aplicación, esta no podrá encontrarlos.

	DNI	Nombre	Apellido	Sexo	FDN	Edad	Celular	Email
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	44555666	TOMÁS	VAZQUEZ	M	01/01/1980	41	1544445555	ABC@GMAIL.COM
2	77888999	MARÍA	CRUZ	F	01/01/1990	31	1577778888	ABC2@GMAIL.COM
3	11222333	JUANA	LÓPEZ	F	01/01/2000	21	1511112222	123@GMAIL.COM

(I)

	Registro	Fecha	Estudio	DNI	Nombre	Apellido	Documento	Imagen	Borrar
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	11222333_20210731_001	31/07/2021	1	11222333	JUANA	LÓPEZ	PDF	IMAGEN	BORRAR

(II)

FIGURA 38: BASE DE DATOS DE DERMOANÁLISIS: (I) TABLA PACIENTES, (II) TABLA REGISTROS

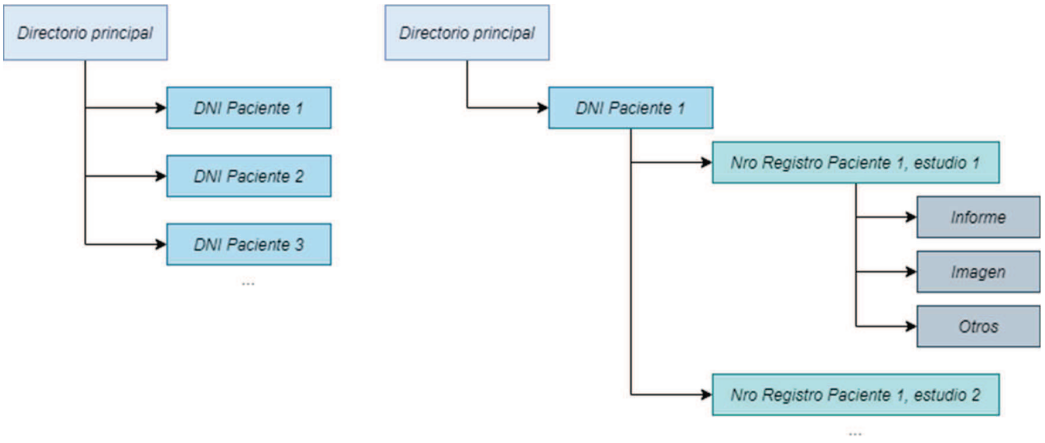


FIGURA 39: DIRECTORIOS GENERADOS POR DERMOANÁLISIS

4.2. Ingreso de nuevo análisis

La rutina para generar un nuevo análisis o estudio, en la pestaña 4, se compone de varias subrutinas. La Figura 40 muestra cuáles son y cómo interactúan entre sí. Para buscar los datos del paciente, el usuario debe ingresar el número de DNI como referencia. Este valor es luego utilizado para buscar el Nombre y el Apellido en la tabla Pacientes. Una vez obtenidos estos datos, se crea un número de registro, el cual se compone por el número de DNI, la fecha del día de la generación del estudio y un número de estudio que permite discriminar entre varios estudios efectuados en un mismo día sobre un mismo paciente. Posteriormente el usuario debe buscar la imagen a analizar mediante una función que permite buscar archivos en la

computadora local. Una vez obtenida la imagen, esta es analizada por el algoritmo de clasificación del software. La Figura 41 muestra la rutina para analizar la imagen seleccionada. Luego de completar el campo de observaciones y guardar el estudio, se genera un registro nuevo en la tabla Registros y se crea una carpeta del estudio en la carpeta del paciente que corresponda. Además, en ella se guarda una copia de la imagen analizada y se genera un informe en formato PDF que se guardará en la misma carpeta.

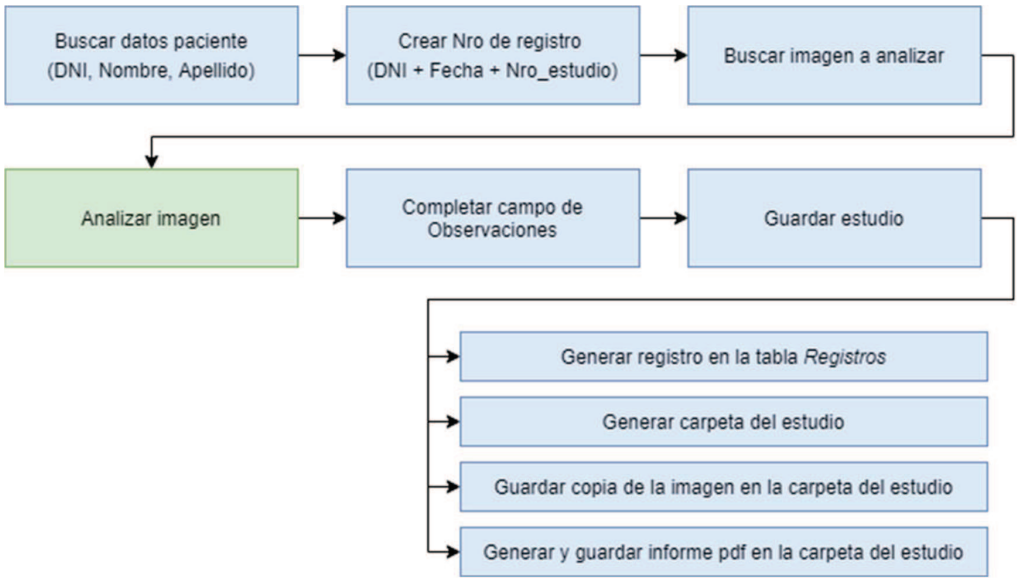


FIGURA 40: RUTINA PARA GENERACIÓN DE NUEVO ANÁLISIS.

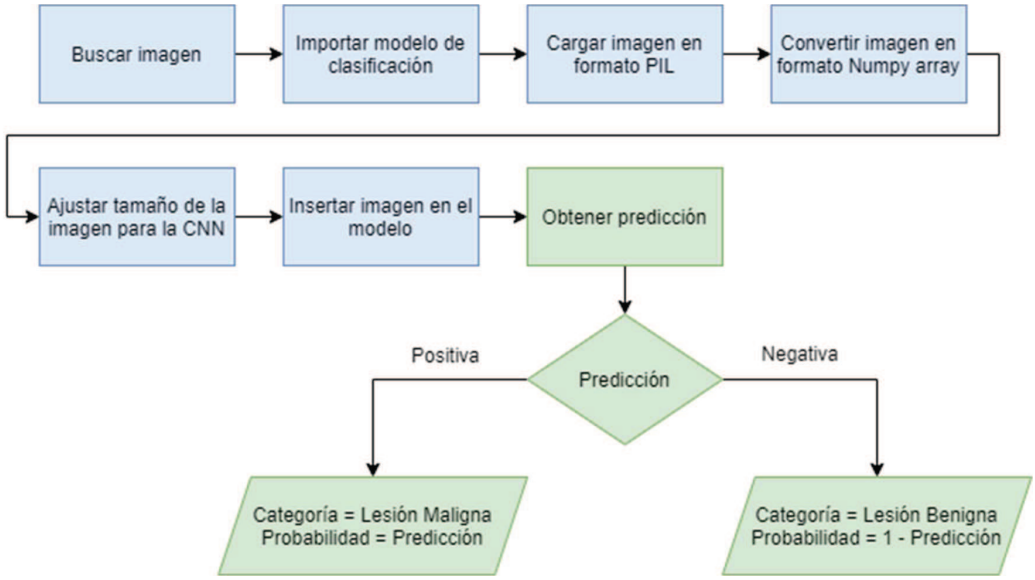


FIGURA 41: RUTINA PARA ANÁLISIS DE IMAGEN MEDIANTE ALGORITMO DE CLASIFICACIÓN.

Al momento de analizar la imagen es necesario acceder a la misma e importar el modelo de DL en formato *.h5* utilizando las librerías Keras y TensorFlow. El siguiente paso es convertir la imagen a formato PIL (*Python Image Library*); esto permite aplicar diferentes métodos de procesamiento de imágenes a la imagen. En el caso de este programa solamente se busca ajustar el tamaño de la misma y normalizarla, adecuándola para el ingreso al algoritmo de clasificación. Una vez obtenida la predicción, se debe interpretar el resultado.

Como el problema presentado a la red es uno de clasificación binaria, el valor obtenido a la salida del algoritmo corresponde a la probabilidad de pertenencia a la clase 1, o se invierte (1 - probabilidad) para dar la probabilidad de pertenecer a la clase 0 [82]. Normalmente el resultado final de una clasificación es únicamente la clase asignada por el umbral de decisión seleccionado, tal como se observa en la Figura 41; pero, al haber obtenido la opinión de un médico dermatólogo, él prefirió visualizar en pantalla la probabilidad asociada a la pertenencia a la clase maligna, independientemente del valor de la misma (Figura 42). Es por esto que finalmente se decidió no aplicar un umbral de decisión a la probabilidad predicha por el algoritmo.

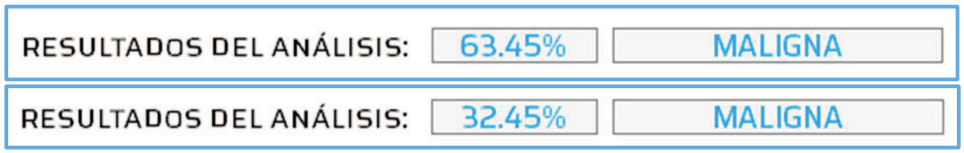
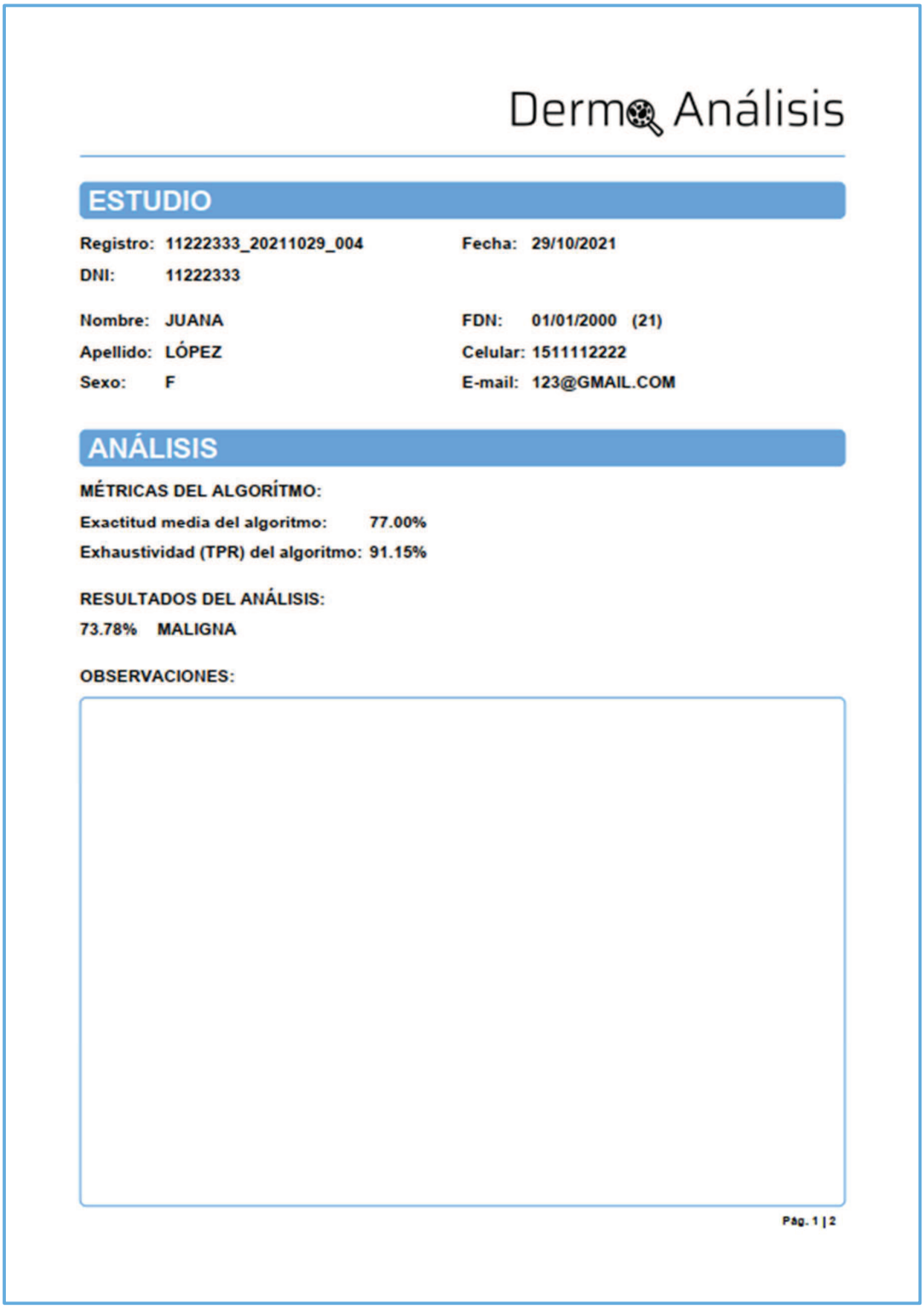


FIGURA 42: PRESENTACIÓN DE RESULTADOS DE CLASIFICACIÓN

4.3. Informe de análisis

Al generar un nuevo análisis se crea una nueva carpeta con el número de registro asociado al análisis y en su interior se guardan una copia de la imagen analizada, junto a un informe en formato PDF.

El informe tiene una estructura muy similar a la pestaña 4 de DermoAnálisis (Figura 44). El mismo se divide en tres partes: la primera, denominada ESTUDIO, resume los datos del paciente, además de incluir la fecha del análisis y el número de registro correspondiente. La segunda, denominada ANÁLISIS, contiene las métricas del algoritmo empleado en el programa, la probabilidad diagnóstica dada por el mismo (expresada porcentualmente) y el campo de observaciones con los comentarios realizados por el profesional médico. Por último, en la segunda hoja del informe (Figura 44), se encuentra el segmento denominado IMAGEN, el cual incluye en el archivo la imagen de la lesión analizada.



Dermo Análisis

ESTUDIO

Registro: 11222333_20211029_004 Fecha: 29/10/2021
DNI: 11222333

Nombre: JUANA FDN: 01/01/2000 (21)
Apellido: LÓPEZ Celular: 1511112222
Sexo: F E-mail: 123@GMAIL.COM

ANÁLISIS

MÉTRICAS DEL ALGORÍTMO:
Exactitud media del algoritmo: 77.00%
Exhaustividad (TPR) del algoritmo: 91.15%

RESULTADOS DEL ANÁLISIS:
73.78% MALIGNA

OBSERVACIONES:

Pág. 1 | 2

Figura 43: Informe generado por DermoAnálisis – Hoja 1.

Dermo Análisis

IMAGEN

Pág. 2 | 2

FIGURA 44: INFORME GENERADO POR DERMOANÁLISIS – HOJA 2.

4.4. Versión final de DermoAnálisis

Para facilitar el acceso a la aplicación se generó un ejecutable que permite abrir la interfaz con un solo *click*, el cual es accesible al descomprimir un archivo .zip y seleccionar el archivo .exe. Al abrir el programa por primera vez, el software corrobora si existe o no una carpeta que sirva como directorio a los estudios de los pacientes. De no existir, este será generado. Dentro del archivo comprimido se encuentran el ejecutable, la base de datos con sus tablas correspondientes (archivo .bd) y el modelo de DL (archivo .h5) que clasificará las lesiones de piel, junto a otros archivos que hacen al correcto funcionamiento de la aplicación y a su estética.

DermoAnálisis permite clasificar lesiones cutáneas melanocíticas benignas y malignas asistiendo al dermatólogo en el diagnóstico de cáncer de piel tipo melanoma. A su vez permite gestionar los datos clínicos del paciente y generar reportes de análisis en formato PDF. La usabilidad de la aplicación fue controlada por un médico dermatólogo y se la adaptó a sus necesidades. Esta fue desarrollada utilizando lenguaje y herramientas open-source: Qt Designer para el diseño de la interfaz, Python para atribuirle su funcionalidad y SQLite para su conexión con bases de datos. En el Anexo 3 encontrará el Manual de Usuario de la aplicación.

5. Discusión

Se ha diseñado y desarrollado una herramienta de asistencia diagnóstica, basada en procesamiento de imágenes dermatológicas e inteligencia artificial (IA) que permite clasificar lesiones cutáneas melanocíticas benignas y malignas, asistiendo al personal médico en el diagnóstico de cáncer de piel tipo melanoma. Esto implicó desarrollar un algoritmo de clasificación binaria basado en redes neuronales convolucionales (CNN) y una aplicación para su utilización y para una integración amigable al flujo clínico de trabajo.

Se utilizaron herramientas propias del aprendizaje profundo (DL), como Keras y TensorFlow, para generar un clasificador binario que logre identificar lesiones benignas y malignas con el fin de asistir al médico dermatólogo en la decisión de realizar o no la escisión de una lesión. Para entrenar y evaluar el algoritmo fue necesario generar un dataset propio a partir de imágenes pertenecientes a repositorios de imágenes internacionales libres. Las mismas fueron obtenidas del archivo ISIC y fueron seleccionadas según subconjunto y según diagnóstico (benigno / maligno). Se trabajó principalmente con el dataset denominado HAM10000 ya que es recurrente en la bibliografía consultada [24], [50], [65]. Como las clases de imágenes se encontraban desbalanceadas, se incluyeron otros ocho subconjuntos del ISIC. Luego de revisar la primera versión del dataset y eliminar imágenes con artefactos no deseados e imágenes repetidas y poco nítidas, el dataset final quedó conformado por 3808 imágenes.

Trabajos como el de Rezvantalab *et al.* [65] conformaron datasets de 10135 imágenes, mientras que otros como el de Esteva *et al.* [24] utilizaron 129450 imágenes y el de Han *et al.* [49] 19398 imágenes. La cantidad mínima de imágenes a utilizar en un trabajo de clasificación no se encuentra definida ya que depende principalmente del objetivo de cada proyecto, seguido de los recursos computacionales con los que se cuenta. En el caso de este proyecto, donde la clasificación es relativamente compleja, se trabajó con un dataset reducido debido a los recursos computacionales disponibles. Probablemente se hubieran obtenido mejores resultados y una mejor capacidad de generalización de la red utilizando un dataset más amplio. A pesar de las diferencias respecto al tamaño del dataset utilizado, los resultados obtenidos en la evaluación del algoritmo alientan a seguir los lineamientos sugeridos en Mongan et al. [83] y Eng et al. [84] en cuanto a la selección de las imágenes y la cantidad.

Al momento de preparar el dataset de entrenamiento, las imágenes fueron sometidas a un cambio de escala (*rescale* o estandarización de datos) y de tamaño (*resize*). Además, se aplicó la técnica de aumentación de datos, para mejorar la generalización del modelo y compensar la reducida cantidad de imágenes utilizadas. Si a futuro se quisiese mejorar el rendimiento del algoritmo, podrían aplicarse técnicas de segmentación de imágenes para aislar las lesiones del fondo de las mismas, tal como se reporta en otros trabajos [26], [27]. Otra posibilidad sería combinar el algoritmo desarrollado con otro que utilice los datos clínicos del paciente como dataset de entrenamiento, es decir, un modelo que utilice la edad, el sexo, la ubicación de la lesión, entre otros elementos, como dataset de entrenamiento.

Para seleccionar la arquitectura de red se tomó como punto de partida aquellas utilizadas por otros proyectos. Se decidió optar por la ResNet50 ya que diferentes trabajos consultados ([47], [49], [51], [65]) mostraron buenos resultados con el uso de esta arquitectura. A futuro podrían utilizarse otras arquitecturas para realizar un trabajo comparativo entre las mismas y seleccionar aquella con mejores métricas o incluso combinarlas. En cuanto a la selección de las capas de la cabeza de la red, se utilizó como guía lo estipulado por Cortés Aguas y Rosebrock [38], [62]. Respecto a los otros parámetros, tales como la función de activación, función de pérdida y el resto de los hiperparámetros utilizados, más allá de lo enunciado en la bibliografía [24], [38], [42], [44], [50], [52], [68], [85], la disciplina del DL requiere realizar pruebas por parte del desarrollador e ir ajustando el modelo para lograr un algoritmo eficaz.

Los resultados obtenidos por la red fueron evaluados en función de tres casos de referencia. En una primera instancia, se comparó el desempeño del algoritmo frente al de los profesionales médicos. Tomando los trabajos [44], [46], [51], [52], [65], se obtuvo una exhaustividad media para los dermatólogos del 77.46% y una especificidad media del 63.13%; en contraste a una exhaustividad y especificidad del modelo del 91.1% y 61.98%. Se concluyó que en cuanto a la métrica de exhaustividad hubo una mejora, mientras que, para la especificidad, la métrica del modelo se encuentra en un rango adecuado debido a que supera dos de los cuatro casos presentados. Además, basándonos en el trabajo de Brinker *et al.* [51], se concluyó que esta métrica equivale a aquella de un dermatólogo promedio, debido a que supera a aquella de los médicos principiantes, pero no la de los médicos con mayor experiencia.

En segunda instancia, se desarrolló una ANN de poca profundidad la cual fue superada tanto en las métricas AUC (ROC AUC 0.73 vs 0.88; ROC PR 0.74 vs 0.90), como en las métricas de

exactitud (67.00% vs 77.20%), exhaustividad (86.72% vs 91.15%) y especificidad (40.00% vs 61.98%) ANN vs CNN respectivamente. Esto demuestra que la complejidad del algoritmo utilizada influyó positivamente en los resultados obtenidos.

Por último, el desempeño del algoritmo de este proyecto fue comparado a aquel de los algoritmos encontrados en la bibliografía [44], [47], [49], [51], [52], [65] donde la mayoría de los trabajos utilizaron la arquitectura ResNet. Se observó que los resultados en la bibliografía son muy variados y que no todos exponen las mismas métricas. La exhaustividad del modelo superó a la mayoría de los algoritmos expuestos ([74.10% - 95.00%] vs 91.15%), mientras que la especificidad y el valor ROC AUC fue superada en todos los casos ([77.90% - 89.2%] vs 61.98%; [0.90 – 0.95] vs 0.88, respectivamente). La métrica de exactitud solo fue reportada en dos de los trabajos analizados ([47], [49]), donde la métrica del algoritmo se encontró en el rango comprendido entre los dos valores ([56.50% - 81.59%] vs 77.20%).

Puede observarse que, a pesar de las limitaciones presentadas, el desempeño del algoritmo es comparable a aquel de los diferentes proyectos mencionados. Como trabajo a futuro podría establecerse un umbral de decisión diferente a 0.5 para mejorar las métricas del del algoritmo de este proyecto y consecuentemente modificar la visualización de los resultados en la interfaz de usuario. No obstante, a diferencia de otros trabajos donde el foco era hacer un escaneo o test de screening, el objetivo de este proyecto era detectar casos positivos, aunque implique un aumento en la tasa de falsos positivos. Por otra parte, como se buscó aportar una herramienta de asistencia diagnóstica, se priorizó que la capacidad de clasificación del algoritmo desarrollado iguale o supere la de los profesionales médicos, objetivo que fue alcanzado.

Se desarrolló una aplicación, con su respectivo manual de usuario, la cual posibilita realizar un seguimiento sistemático de la evolución de las lesiones cutáneas para detectar de manera temprana aquellas que son potencialmente malignas. Este software permite al dermatólogo gestionar los datos clínicos del paciente, administrar su información y sus imágenes médicas y generar reportes de análisis en formato PDF, además de utilizar la herramienta como asistente de diagnóstico mediante el modelo de DL. La aplicación fue desarrollada utilizando lenguaje y herramientas open-source: Qt Designer para el diseño de la interfaz, Python para atribuirle su funcionalidad y SQLite para su conexión con bases de datos. A la misma se accede mediante un archivo ejecutable contenido en un archivo comprimido, donde además se encuentran los archivos de la base de datos y del modelo. La aplicación fue

presentada a un profesional médico, quien estuvo conforme con el entregable final. Una de las solicitudes realizadas fue que, al analizar la imagen de la lesión, el programa muestre la probabilidad diagnóstica asociada a la clase de tipo maligna para que el médico decida por su cuenta el umbral de decisión a utilizar.

Como trabajo a futuro la herramienta podría ser adaptada para analizar más de una imagen a la vez, y permitir realizar anotaciones o transformaciones sobre las mismas. Por otra parte, la herramienta podría ser modificada para ser integrada a un sistema de historia clínica electrónica de una institución de salud. Los códigos de la aplicación utilizados en este proyecto quedan disponibles para continuar siendo perfeccionados y adecuarlos a las necesidades de diferentes grupos de profesionales en salud del ámbito de la dermatología.

6. Conclusión

Se logró desarrollar una herramienta de asistencia al diagnóstico capaz de realizar un seguimiento sistemático y objetivo de la evolución de las lesiones de piel para detectar de manera temprana aquellas que son potencialmente malignas. Además, posibilita reducir los tiempos y cicatrices innecesarias, en los casos donde la lesión de interés sea benigna. DermoAnalisis también permite al médico gestionar la información del paciente y generar una base de datos donde guardar los informes y las imágenes generadas por la aplicación.

La herramienta se compone por una aplicación y un modelo de aprendizaje profundo o Deep Learning (DL). Este modelo es un algoritmo de clasificación binario que permite clasificar lesiones cutáneas melanocíticas benignas y malignas, asistiendo al médico dermatólogo en el diagnóstico de cáncer de piel tipo melanoma. El desempeño de exhaustividad del algoritmo utilizado superó al rendimiento de los profesionales médicos, por lo que se considera que el modelo es eficaz en su tarea. No obstante, el objetivo del algoritmo no es reemplazar al profesional de salud, sino asistirlo. Hay información clínica a la cual solamente tiene acceso el dermatólogo y que el modelo no toma en consideración al momento de clasificar la lesión.

Finalmente, este proyecto me permitió conocer y adentrarme a un área de relevancia para el ámbito de salud ya que permite desarrollar herramientas que asisten a los profesionales médicos de diferentes especialidades y ayudan al paciente a tener un diagnóstico preciso.

7. Anexos

7.1. Anexo 1 – ResNet50

La arquitectura ResNet tiene diferentes versiones, acorde a la cantidad de capas que posee. La Figura 45 muestra las diferentes configuraciones. En ella se puede observar que la ResNet50 posee las siguientes características:

- Una capa convolucional con un kernel de 7x7 y 64 kernels diferentes, todos con un stride de tamaño 2.
- Una capa de max pooling con un stride de tamaño 2
- En la siguiente convolución hay un kernel de 1x1, 64 kernels después de este tiene uno de 3x3, 64 kernels y por último un kernel de 1x1, 256 kernels. Estas tres capas se repiten 3 veces, dando 9 capas finales en este paso.
- Luego se observa un kernel de 1x1, 128 después de eso, un kernel de 3x3, 128 y por último un kernel de 1x1, 512. Este paso se repite 4 veces, lo que da 12 capas en este paso.
- Después de eso, hay un kernel de 1x1, 256 y 2 kernels más de 3x3, 256 y 1x1, 1024 y esto se repite 6 veces, dando un total de 18 capas.
- Y luego de nuevo un kernel de 1x1, 512 con 2 más de 3x3, 512 y 1x1, 2048 y esto se repite 3 veces dando un total de 9 capas.
- Después de eso, se hace un grupo promedio y se termina con una capa completamente conectada que contiene 1000 nodos y al final una función softmax, por lo que se obtiene 1 capa.

Las funciones de activación y las capas de max pooling y average pooling no se cuentan, sumando de esta forma una red convolucional profunda de 1 + 9 + 12 + 18 + 9 + 1 = 50 capas.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

FIGURA 45: DIFERENTES VERSIONES DE LA ARQUITECTURA RESNET [71].

7.2. Anexo 2 – Conceptos básicos sobre CNN

7.2.1. *Stride* y *padding*

Las neuronas de las capas convolucionales toman datos de la capa anterior dentro del campo receptivo de la misma. Cada neurona en la capa convolucional debe ver información diferente al resto para un mejor análisis de la imagen. El concepto denominado *stride* indica el desplazamiento entre los campos receptivos de las neuronas (Figura 46), tanto horizontalmente como verticalmente [38]. Es decir, un stride o paso de 2, implica que la ventana de la siguiente neurona se desplazará dos píxeles hacia la derecha Si el paso es demasiado chico, existirá una mayor superposición entre dos campos receptivos, pero si es demasiado grande, la superposición será más pequeña. El stride determina el tamaño de la capa superior y la cantidad de superposición en los campos receptivos. Por esta razón, cuando la superposición es grande, se requerirán menos neuronas en la siguiente capa convolucional [54].

Trabajar con un stride demasiado grande puede ocasionar que no se llegue a cubrir la imagen en su totalidad. Es decir, al ir desplazando la ventana hacia la derecha, esta visualizará siempre la misma cantidad de pixeles hasta acercarse al borde de la imagen, donde la ventana se achica porque se quedó sin columnas de pixel para analizar [38]. Entonces, como el objetivo es siempre respetar la uniformidad, se pueden dar dos posibles soluciones (Figura 47): La primera opción es ignorar los píxeles adicionales en el borde. Entonces, dado que no se pueden

cubrir las últimas columnas de píxeles, se dejan afuera algunas columnas tanto de la izquierda como de la derecha de la imagen. Por lo tanto, al eliminarlas, la imagen termina teniendo un tamaño menor a la original. Esta opción es conocida como *Valid Padding*, e implica solamente utilizar ubicaciones de ventana válidas e ignorar los píxeles adicionales en el borde. La segunda opción es agregar filas y columnas adicionales de píxeles ficticios o píxeles en blanco. Este tipo de relleno se denomina *Same Padding,* que significa rellenar de tal manera que tenga una salida que pueda ser cubierta por una ventana del mismo ancho y alto. Los argumentos de *padding* en las funciones de Keras trabajan con Valid Padding de forma predeterminada, de creer que los bordes de las imágenes almacenan información importante, es posible modificar este parámetro a Same Padding [54].

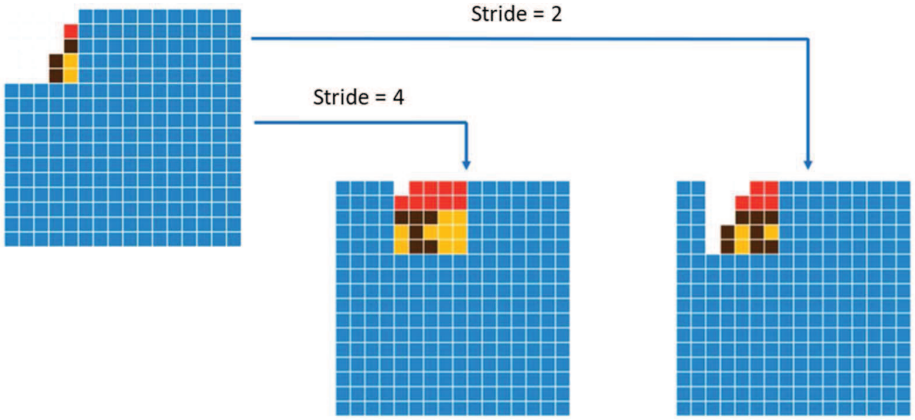


FIGURA 46: VISUALIZACIÓN DEL STRIDE. ADAPTADO DE [54]

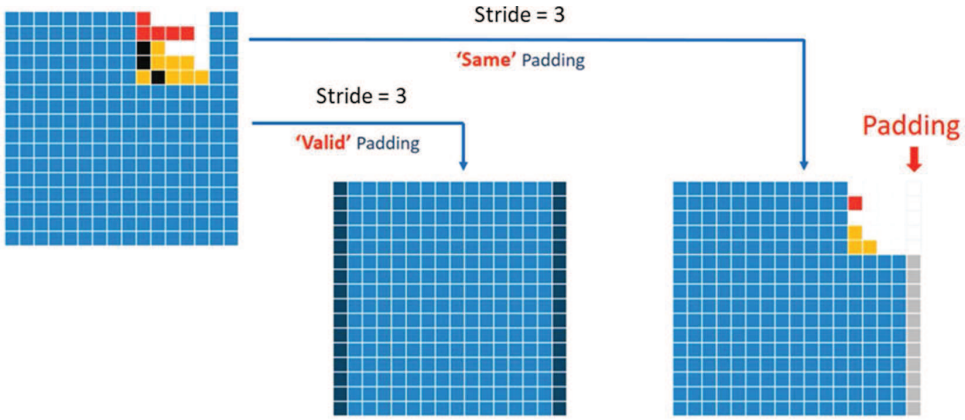


FIGURA 47: VISUALIZACIÓN DEL PADDING. ADAPTADO DE [54]

7.2.2. Filtros y Mapas de Características

Para poder trasladar de manera representativa la información de los píxeles dentro de un campo receptivo a una única neurona, se emplean filtros. Estos, también llamados *kernel*, son una matriz de las mismas dimensiones que la ventana del campo receptivo. Esta matriz contiene ciertos valores que son luego convolucionados con cada valor de píxel de la imagen. Es decir, se multiplica cada valor de píxel con el valor de filtro correspondiente y luego se suman todos los productos. El valor final obtenido logra entonces representar la información de los píxeles de todo el campo receptivo [38], [54].

Cada filtro es aplicado a lo largo de toda la capa anterior, donde cada posición da como resultado una activación de la neurona que da como salida el denominado mapa de características. De esta forma, termina obteniéndose varios mapas de características por capa. Los valores del filtro son autosuficientes, es decir, no es necesario definir los valores del filtro ya que la red los aprende a lo largo de su entrenamiento [38], [54].

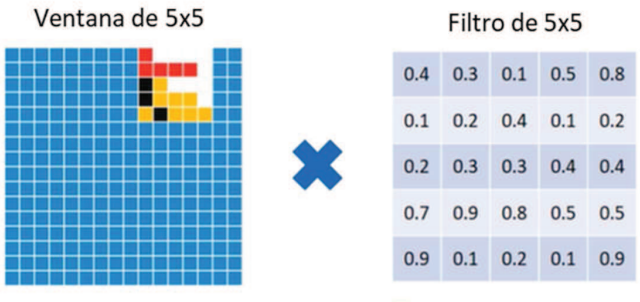


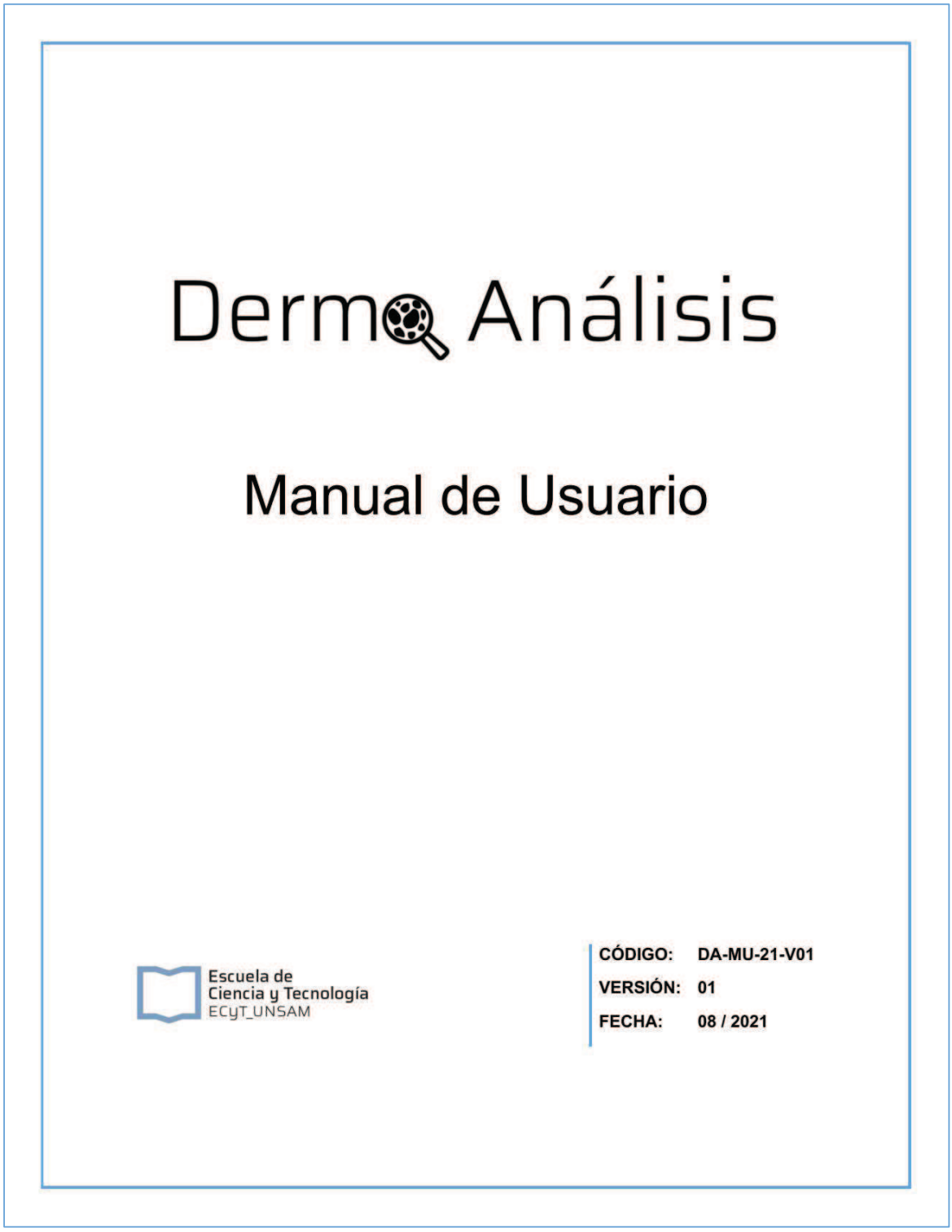
FIGURA 48: ILUSTRACIÓN DE FILTROS APLICADOS EN LAS CAPAS CONVOLUCIONALES. ADAPTADO DE [54].

7.2.3. Capa de *pooling*

El objetivo de las capas de *pooling* es submuestrear la imagen de entrada para reducir la carga computacional, limitando así el riesgo de sobreajuste. Al igual que en las capas convolucionales, cada neurona tiene un campo receptivo pequeño y rectangular, donde debe definirse su tamaño, el stride y el padding. El tamaño de la capa de pooling varía según el valor de stride seleccionado. Por ejemplo, al elegir un stride de 2, la capa de pooling termina por tener la mitad del ancho y la altura de la capa anterior. De esta manera, se logra reducir los cálculos y el uso de memoria [38], [54], [64].

Las neuronas de la capa de pooling no tienen peso. Solamente unifican la entrada usando una función de agregación de máximos (*max*) o de la media (*mean*). Utilizando la función de *max pooling* se calcula el valor máximo de los campos pertenecientes a cada ventana de campo receptivo. De esta forma se obtiene un único valor por ventana. Similar a la función de max pooling, en la función *average pooling* se obtiene la media de los valores y se guarda como salida para la capa de pooling. Por lo general, la función de max pooling funciona mejor que las opciones alternativas porque resalta las características principales en lugar de promediarlas [54], [64], [85].

7.3. Anexo 3 – Manual de Usuario de DermoAnálisis



Derm Análisis

CÓDIGO:	DA-MU-21-V01
VERSIÓN:	01
FECHA:	08 / 2021

1 TABLA DE CONTENIDO

Dermo Análisis

CÓDIGO:	DA-MU-21-V01
VERSIÓN:	01
FECHA:	08 / 2021

2 OBJETIVO

El objetivo de la aplicación **DermoAnálisis** es asistir a médicos dermatólogos al momento de analizar y seleccionar el tratamiento adecuado para una lesión dermatológica melanocítica.

Como la identificación de una lesión melanocítica maligna suele ser una tarea de alta complejidad, la aplicación permite ingresar una imagen de una lesión y calcular una probabilidad diagnóstica que indicará qué tan probable es que dicha lesión sea benigna o maligna. Realiza esta tarea gracias a un algoritmo de cálculo basado en inteligencia artificial con una exactitud del 77.00% y una exhaustividad (tasa de verdaderos positivos) del 91.5%. A partir de la información brindada por el algoritmo, sumada a la experiencia del médico, el dermatólogo podrá realizar un diagnóstico más preciso sobre el tipo de lesión en estudio del paciente.

A su vez, la aplicación permite gestionar los datos de los pacientes, sus informes clínicos y las imágenes de las lesiones dermatológicas.

3 INSTALACIÓN

Para acceder al programa es necesario descomprimir el archivo .zip en su directorio de interés. Una vez hecho esto, adentro de la carpeta correspondiente encontrará dos elementos fundamentales: el ejecutable (archivo .exe) y una carpeta llamada 'DirectorioDermoAnalisis'.

De no existir la carpeta al descomprimir el archivo .zip, abra la aplicación seleccionando el ejecutable. Automáticamente se generará una carpeta de nombre 'DirectorioDermoAnalisis' en el mismo directorio que el ejecutable.

Para facilitar el uso cotidiano del programa es recomendable generar un acceso directo al ejecutable de la aplicación y al directorio 'DirectorioDermoAnalisis'.

4 DERMOANÁLISIS

La aplicación permite almacenar los datos demográficos de los pacientes, generar una carpeta para cada uno de ellos y a su vez organizar los diferentes estudios en carpetas individuales dentro de la carpeta exclusiva de cada paciente.

Para poder utilizar la aplicación, primero es necesario ingresar al paciente al sistema y posteriormente cargar el registro de consulta. Los datos de los pacientes son modificables, mientras que los registros de consulta no lo son.

La aplicación contiene cuatro pestañas que interactúan entre sí:

1) Pacientes
2) Editar Pacientes
3) Historial de consultas
4) Nuevo análisis

En cada una de las cuatro pestañas es posible realizar diferentes acciones. A continuación, se enunciarán las mismas detallando cada paso.

4.1 PACIENTES

La pestaña *PACIENTES* cuenta con cinco zonas principales, tal como se observa en la Figura 1:

1. Tabla de pacientes
2. Buscador de paciente por número de DNI
3. Buscador de paciente por Apellido
4. Refrescar información
5. Campo de mensajes

TABLA DE PACIENTES

El objetivo de esta tabla es visualizar los datos básicos de los pacientes cargados en el sistema. A cada uno de ellos le corresponde un único registro modificable. La tabla cuenta con ocho columnas de información: DNI, Nombre, Apellido, Sexo, Fecha de Nacimiento (FDN), Edad, Celular e E-Mail. De esta manera el usuario logra acceder a los datos del paciente ágilmente, incluyendo los datos de contacto en caso de necesitar comunicarse con el mismo.

Dermo Análisis

CÓDIGO:	DA-MU-21-V01
VERSIÓN:	01
FECHA:	08 / 2021

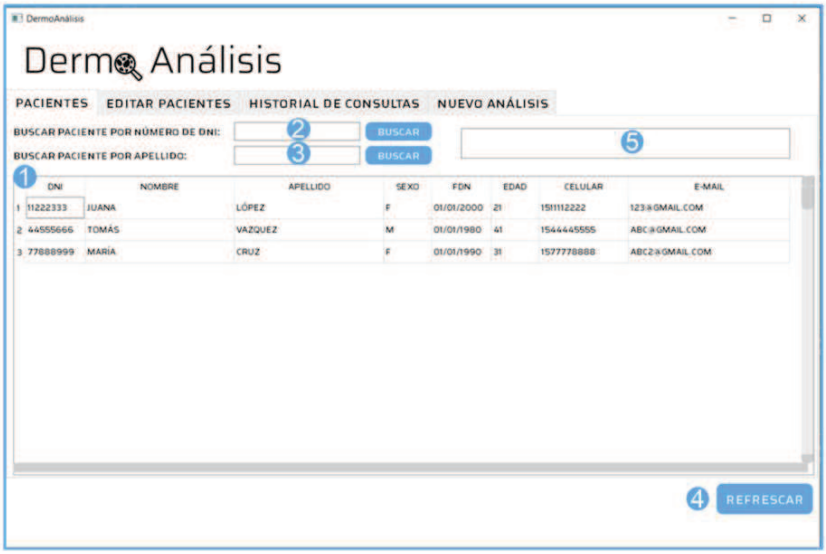


Figura 1: pestaña 1, PACIENTES

BUSCADOR DE PACIENTE POR NÚMERO DE DNI

Para acceder a los datos de un paciente en particular es posible acotar la búsqueda utilizando su número de DNI. Al ingresar en el campo 2 dicho valor y posteriormente hacer *click* en el botón BUSCAR, localizado a la derecha del campo de texto, la tabla automáticamente mostrará el registro que corresponda al valor ingresado. En simultáneo, en el campo de mensajes número 5 se visualizará el mensaje "REGISTRO ENCONTRADO POR DNI". De no existir ningún registro con el valor de DNI ingresado, en el mismo campo se visualizará el anuncio "[ERROR] REGISTRO NO ENCONTRADO" y la tabla mostrará la totalidad de sus registros.

BUSCADOR DE PACIENTE POR APELLIDO

Al igual que el buscador anterior, es posible acotar la búsqueda mediante el apellido del paciente. Este buscador permite buscar por apellido completo, además de permitir la búsqueda de un único Apellido en caso de poseer más de uno. Realizar la búsqueda con letras mayúsculas y/o minúsculas no afecta al resultado de la misma. En este caso, al

ingresar en el campo 3 el Apellido deseado y posteriormente hacer *click* en el botón BUSCAR que se encuentra a la derecha del campo de texto, la tabla automáticamente mostrará todos los registros que correspondan al valor ingresado. En simultáneo, en el campo de mensajes número 5 se visualizará "REGISTRO ENCONTRADO POR APELLIDO". De no existir ningún registro con el apellido ingresado, en el mismo campo se leerá "[ERROR] REGISTRO NO ENCONTRADO" y la tabla mostrará la totalidad de sus registros.

REFRESCAR INFORMACIÓN

El botón REFRESCAR permite actualizar la tabla luego de haber realizado una búsqueda y volver a visualizarla.

4.2 EDITAR PACIENTES

Esta pestaña permite realizar tres acciones principales: agregar paciente, actualizar paciente y eliminar paciente. Cuenta con seis zonas principales observadas en la Figura 2:

1. Buscador de paciente por número de DNI
2. Campos con información de paciente a completar o a modificar
3. Botón para borrar registro paciente
4. Botón para agregar registro paciente
5. Botón para actualizar registro paciente
6. Campo de mensajes

Dermo Análisis

CÓDIGO:	DA-MU-21-V01
VERSIÓN:	01
FECHA:	08 / 2021

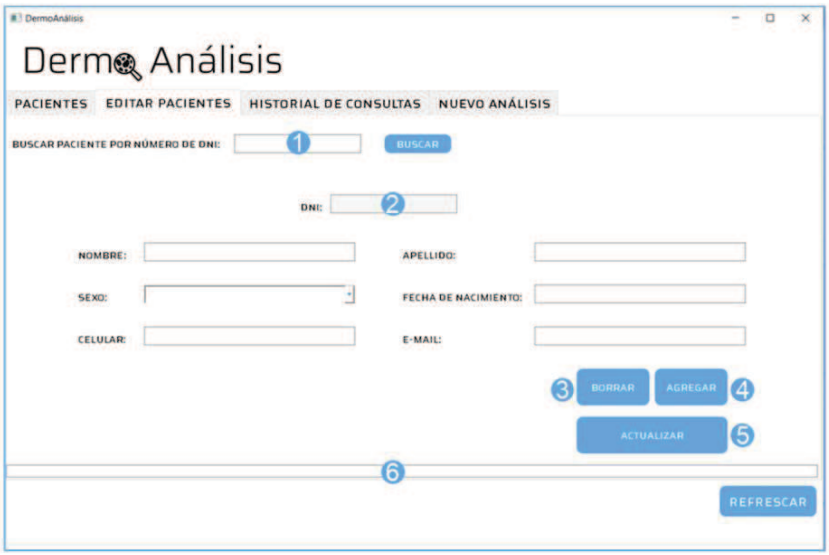


Figura 2: pestaña 2, EDITAR PACIENTES

Figura 3: pestaña 2, EDITAR PACIENTES. Paciente agregado con éxito.

AGREGAR PACIENTE

Para agregar un paciente es necesario completar los campos de texto ubicados en pantalla. Cada uno de ellos presenta características específicas sobre el dato a ingresar; siempre que estas no se cumplan, aparecerá en el campo 6 una advertencia para asistir al usuario en el correcto ingreso de datos. Las delimitaciones de cada campo son los siguientes:

- **DNI**: valor de nueve dígitos. Acepta únicamente caracteres numéricos.
- **NOMBRE**: acepta únicamente caracteres alfabéticos de extensión indefinida.
- **APELLIDO**: acepta únicamente caracteres alfabéticos de extensión indefinida.
- **SEXO**: en pantalla se desplegará una lista donde se puede elegir la opción F de femenino y M de masculino.
- **FECHA DE NACIMIENTO**: acepta únicamente caracteres numéricos, separados por los caracteres "/" o "-", ingresados en el formato DD/MM/AAAA o DD-MM-AAAA.
- **CELULAR**: acepta únicamente caracteres alfabéticos de extensión indefinida.
- **E-MAIL**: acepta caracteres alfanuméricos de extensión indefinida.

Una vez ingresados los valores, apretar el botón 4 llamado AGREGAR. Automáticamente los campos de texto se vaciarán y en pantalla se observará un anuncio de confirmación como en la Figura 3. Al navegar hacia la primera pestaña del programa logrará visualizar el nuevo registro ingresado. Todos los caracteres de tipo alfabético serán guardados en letras mayúsculas.

Al realizar esta acción se adicionarán los datos del contacto a la tabla de la primera pestaña de la interfaz y además se generará una carpeta en el directorio destino titulado con el número de DNI del paciente, como se observa en la Figura 4.

Para modificar los datos es necesario simplemente reemplazar en los campos de interés la información actualizada y posteriormente hacer *click* en el botón 5 llamado ACTUALIZAR.

El campo número 6 de la Figura 6 informará al usuario de haber algún inconveniente con la información ingresada, al igual que en el punto AGREGAR PACIENTE.

El único campo no modificable es el del número de DNI. De desear modificar dicho dato, es necesario eliminar el registro paciente y cargar uno nuevo con los datos correctos.

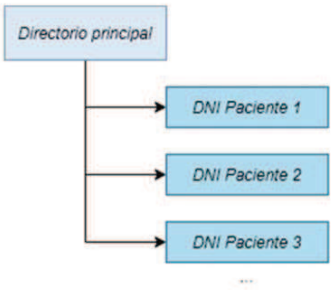


Figura 4: diagrama de estructura de carpetas paciente generadas.

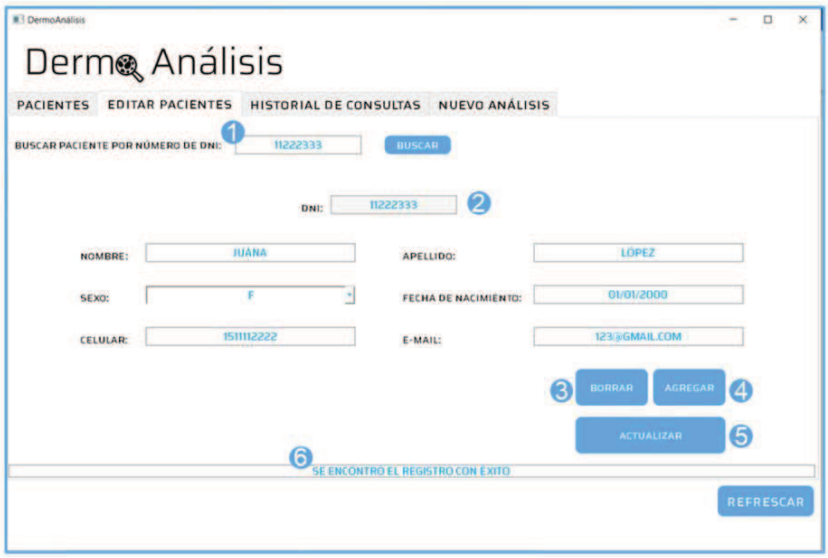


Figura 5: pestaña 2, EDITAR PACIENTES. Paciente con datos a actualizar encontrado con éxito.

Figura 6: pestaña 2: EDITAR PACIENTES. Registro de paciente actualizado con éxito.

ELIMINAR PACIENTE

Para eliminar un registro de un paciente se debe primero buscar el paciente por número de DNI mediante el elemento 1 de la Figura 5. Al ingresar el valor de DNI y posteriormente pulsar en el botón BUSCAR ubicado a la derecha del campo de texto, se visualizarán en pantalla los datos guardados asociados a dicho registro paciente.

Posteriormente deberá hacer *click* en el botón 4 de la Figura 6 llamado BORRAR. Al realizar esta acción, se abrirá una ventana de advertencia tipo verificación de dos pasos para confirmar el borrado del registro paciente como se observa en la Figura 7. Hacer *click* en "SI" para confirmar y en "NO" para desestimar la acción de borrado.

Esta acción eliminará la carpeta exclusiva del paciente correspondiente junto a los contenidos de la misma: carpetas de estudios con sus contenidos y otros documentos guardados en la misma.

Dermo Análisis

CÓDIGO:	DA-MU-21-V01
VERSIÓN:	01
FECHA:	08 / 2021

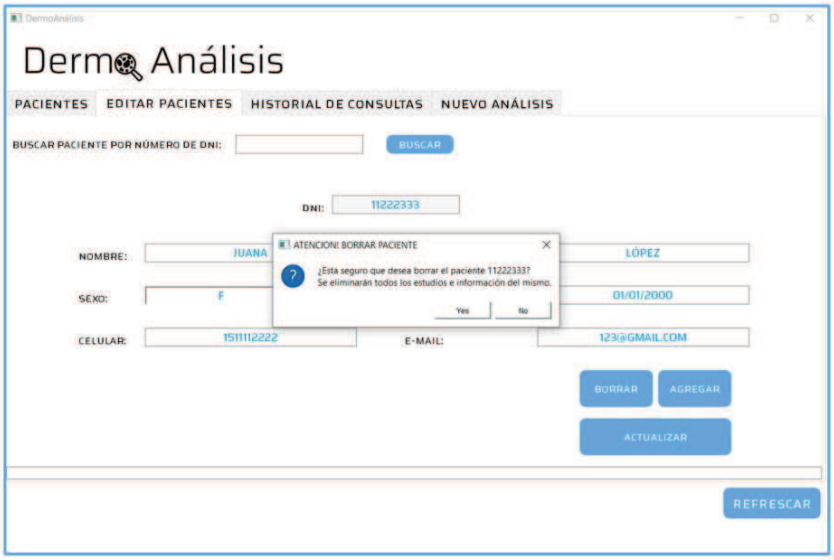


Figura 7: pestaña 2, EDITAR PACIENTES. Mensaje de confirmación de eliminación de registro paciente.

4.3 HISTORIAL DE CONSULTAS

La pestaña *HISTORIAL DE CONSULTAS* es parecida a la primera pestaña del programa (Figura 8). La principal diferencia es que la tabla en pantalla ofrece un listado de estudios o consultas de los pacientes en vez de su información.

El objetivo de esta tabla es visualizar todos los estudios realizados por el profesional. A su vez la misma permite comparar los datos entre diferentes análisis de un mismo paciente utilizando buscadores de DNI y de Apellido. A cada estudio realizado le corresponde un único registro no modificable. La tabla cuenta con nueve columnas de información: Número de registro, Fecha, Número de estudio del día, Nombre, Apellido, Reporte, Imagen y Borrar. De esta manera, el usuario logra acceder a los estudios ágilmente.

Los buscadores 2 y 3 funcionan exactamente de la misma manera que aquellos en la primera pestaña del programa, incluyendo los mensajes que aparecerán en el campo número 5.

Dermo Análisis

CÓDIGO:	DA-MU-21-V01
VERSIÓN:	01
FECHA:	08 / 2021

ID DE REGISTRO

La primera columna corresponde al identificador (ID) de registro asociado al estudio o análisis realizado. Este número se genera a partir del DNI del paciente, la fecha en la cual se realiza el estudio y el número de estudio de ese mismo día ya que en un único día es posible revisar y analizar más de una lesión de piel.

Se recomienda adicionar cada estudio el mismo día en que cita al paciente ya que a cada estudio cargado se le asocia la fecha del día en el que se lo incorporó al sistema.

REPORTE

La séptima columna corresponde al reporte generado por cada estudio realizado. Al hacer *click* sobre la palabra PDF, se abrirá en pantalla el documento con formato *PDF* asociado al número de registro correspondiente a esa fila en la tabla.

IMAGEN

La octava columna corresponde a la imagen asociada al estudio perteneciente a esa fila en la tabla. Al hacer *click* sobre la palabra IMAGEN, automáticamente se abrirá en pantalla el archivo de la imagen de la lesión de piel asociada a dicho registro.

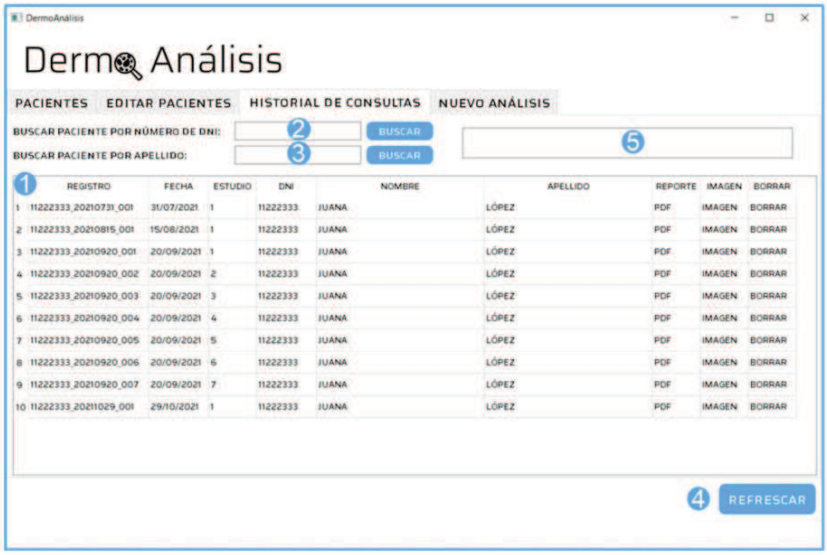
BORRAR

La novena y última columna de la tabla permite eliminar el registro asociado al estudio perteneciente a esa fila de la tabla. Esta acción es irremediable, razón por la cual al hacer *click* sobre la palabra BORRAR en la fila deseada, aparecerá en pantalla una advertencia solicitando la confirmación de la acción. De desear continuar con la misma, seleccionar SI, sino presionar NO y se desestimará la acción de borrado.

Esta acción elimina toda la carpeta asociada al registro correspondiente, incluyendo el reporte generado, la imagen y cualquier otro elemento que se encuentre en la carpeta del registro.

Dermo Análisis

CÓDIGO:	DA-MU-21-V01
VERSIÓN:	01
FECHA:	08 / 2021



REGISTRO	FECHA	ESTUDIO	DNI	NOMBRE	APELLIDO	REPORTE	IMAGEN	BORRAR
1 11222333_20210731_001	31/07/2021	1	11222333	JUANA	LÓPEZ	PDF	IMAGEN	BORRAR
2 11222333_20210815_001	15/08/2021	1	11222333	JUANA	LÓPEZ	PDF	IMAGEN	BORRAR
3 11222333_20210920_001	20/09/2021	1	11222333	JUANA	LÓPEZ	PDF	IMAGEN	BORRAR
4 11222333_20210920_002	20/09/2021	2	11222333	JUANA	LÓPEZ	PDF	IMAGEN	BORRAR
5 11222333_20210920_003	20/09/2021	3	11222333	JUANA	LÓPEZ	PDF	IMAGEN	BORRAR
6 11222333_20210920_004	20/09/2021	4	11222333	JUANA	LÓPEZ	PDF	IMAGEN	BORRAR
7 11222333_20210920_005	20/09/2021	5	11222333	JUANA	LÓPEZ	PDF	IMAGEN	BORRAR
8 11222333_20210920_006	20/09/2021	6	11222333	JUANA	LÓPEZ	PDF	IMAGEN	BORRAR
9 11222333_20210920_007	20/09/2021	7	11222333	JUANA	LÓPEZ	PDF	IMAGEN	BORRAR
10 11222333_20211029_001	29/10/2021	1	11222333	JUANA	LÓPEZ	PDF	IMAGEN	BORRAR

Figura 8: pestaña 3, HISTORIAL DE CONSULTAS.

4.4 NUEVO ANÁLISIS

El objetivo de la pestaña *NUEVO ANÁLISIS* es agregar un nuevo estudio o análisis al sistema. A continuación se observa la Figura 9 donde se visualizan indicadores numéricos para un mejor entendimiento del instructivo:

1. Buscador de paciente por DNI
2. Datos del paciente asociados al registro a guardar
3. Buscador de imagen a analizar
4. Sector de visualización de imagen
5. Nombre de imagen seleccionada
6. Botón para analizar imagen
7. Datos del algoritmo
8. Resultados de la imagen, obtenidos por el algoritmo
9. Campo de observaciones
10. Campo de mensajes
11. Botón para guardar el estudio

Dermo Análisis

CÓDIGO:	DA-MU-21-V01
VERSIÓN:	01
FECHA:	08 / 2021

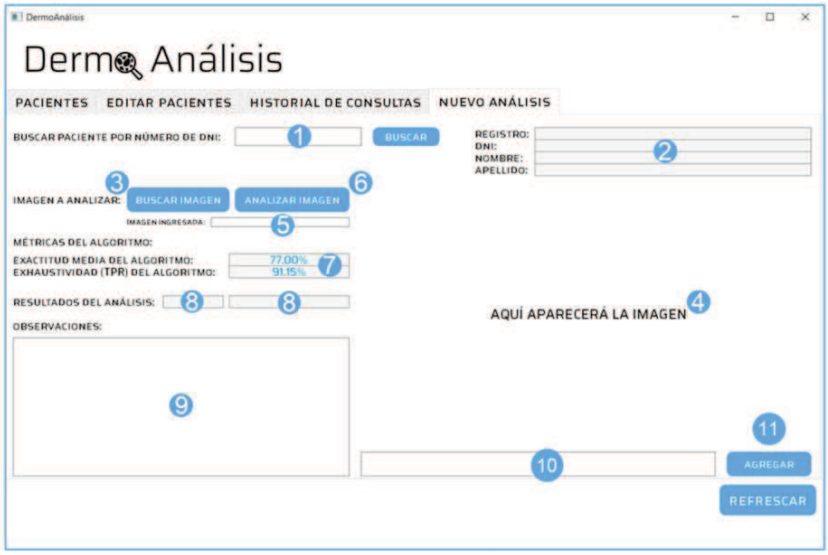


Figura 9: pestaña 4, NUEVO ANÁLISIS.

4.4.1 CÓMO ADICIONAR UN NUEVO ESTUDIO

1. ASOCIAR EL ESTUDIO CON UN PACIENTE

El primer paso es asociar el estudio a realizar con un paciente previamente cargado en la tabla de la pestaña PACIENTES (Figura 10). Al ingresar el DNI de dicho paciente en el campo de texto 1 y al presionar el botón BUSCAR que se encuentra a su derecha, se actualizará el campo número 2. En este campo se visualizará un nuevo número de registro asociado al estudio en cuestión – como se mencionó anteriormente, el número de registro posee el formato DNI_FECHA_NUMERO_DE_ESTUDIO –, el DNI, el Nombre y el Apellido del paciente. Esta información permite al profesional visualizar los datos básicos del paciente mientras se le realiza la consulta médica.

Figura 10: pestaña 4: NUEVO ANÁLISIS. 1. Asociar el estudio con un paciente

2. BUSCAR IMAGEN

Posteriormente deberá presionar el botón 3 denominado BUSCAR IMAGEN (Figura 10). Al hacerlo podrá navegar por las carpetas de la computadora y localizar la imagen de interés. Una vez seleccionada la imagen, ésta aparecerá en pantalla en el campo 4 y su nombre en el campo 5. Visualizar la imagen permite al profesional observarla de cerca mientras realiza anotaciones en el campo de observaciones número 9. Estas pueden anotarse tanto en esta instancia del instructivo como más adelante, posterior a la visualización de los resultados del algoritmo de análisis.

3. ANALIZAR IMAGEN

Para analizar la imagen previamente seleccionada deberá hacer *click* sobre el botón 6 de la Figura 10, denominado ANALIZAR IMAGEN. Una vez realizada esta acción, el resultado se visualizará en el campo 8.

El resultado del algoritmo es representado por un valor de probabilidad de clasificación de pertenencia a la clase maligna. Es importante tomar en consideración la información brindada en el campo 7 de la pestaña a la hora de tomar la decisión clínica de realizar o no la escisión de la lesión. Tal como se observa en pantalla (Figura 11), el algoritmo tiene un valor de exactitud del 77.00% y una exhaustividad (tasa de verdaderos positivos) del 91.15%. La información mostrada en pantalla tiene como objetivo ayudar al médico dermatólogo a tomar una decisión más acertada al momento de diagnosticar la lesión.

4. COMPLETAR CAMPO DE OBSERVACIONES Y GUARDAR

El último paso es terminar de completar el campo de observaciones teniendo en cuenta toda la información brindada y posteriormente hacer *click* sobre el botón 11 de la Figura 12, denominado AGREGAR.
Antes de confirmar la acción de guardado, corroborar que la información volcada haya sido la correcta y que ésta se encuentre completa.

El botón 11 permite guardar el registro en la tabla de la pestaña HISTORIAL DE CONSULTAS y además genera una carpeta con el número de registro de estudio adentro de la carpeta del paciente en cuestión, junto a una copia de la imagen seleccionada previamente y un reporte con extensión PDF, como se observa en la Figura 13. El campo 10 de la Figura 12 confirmará la acción en caso de ser positiva, o de lo contrario brindará un mensaje.

En el reporte generado puede observarse la misma información mostrada en la pestaña NUEVO ANÁLISIS, junto a la imagen seleccionada.

Dermo Análisis

CÓDIGO:	DA-MU-21-V01
VERSIÓN:	01
FECHA:	08 / 2021

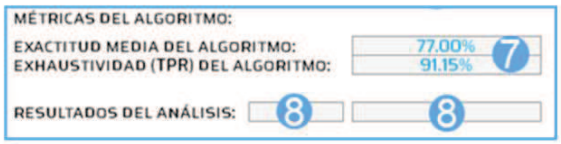


Figura 11: NUEVO ANÁLISIS. Datos del algoritmo y resultados del análisis.

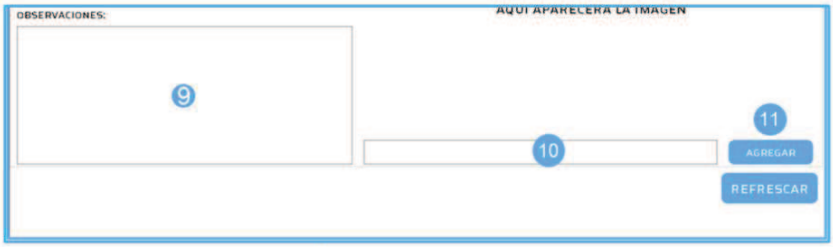


Figura 12: pestaña 4: NUEVO ANÁLISIS. Campo de observaciones y guardado de análisis.

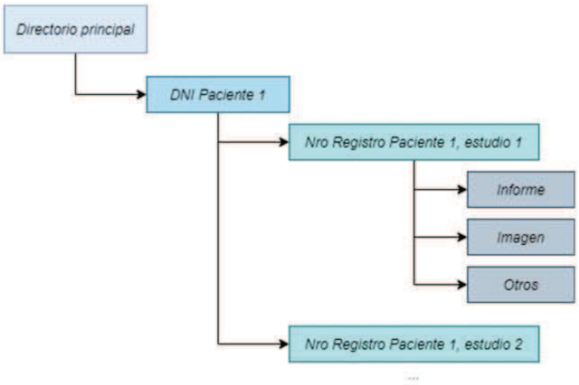


Figura 13: diagrama de estructura de carpetas de consultas generadas, junto a su contenido.

8. Referencias

[1] G. J. Tortora and B. Derrickson, *Principios de anatomía y fisiología*. México; Madrid: Editorial Médica Panamericana, 2006.

[2] C. Pelufo, *Estructura y función de la piel*. Departamento de Ciencias Biomedicas de la Universidad Cardenal Herrera (CEU).

[3] Instituto del Cáncer, 'Cáncer de piel: Recomiendan cuidarse durante todo el año y en especial a los niño y niñas', *Gobierno de la Provincia de Buenos Aires*. https://www.gba.gob.ar/saludprovincia/noticias/c%C3%A1ncer_de_piel_recomiendan_cuidarse_durante_todo_el_a%C3%B1o_y_en_especial_los (accessed Oct. 05, 2021).

[4] American Cancer Society, '¿Qué es la radiación ultravioleta (UV)?', *American Cancer Society | Information and Resources*, Apr. 19, 2017. https://www.cancer.org/content/cancer/es/cancer/cancer-de-piel/prevencion-y-deteccion-temprana/que-es-la-radiacion-de-luz-ultravioleta.html (accessed Oct. 30, 2021).

[5] World Health Organization, *Artificial tanning sunbeds: risks and guidance*. Geneva: World Health Organization, 2003. [Online]. Available: https://www.who.int/uv/publications/en/sunbeds.pdf

[6] IDEAM - Instituto de Hidrología, Meteorología y Estudios Ambientales, 'Generalidades de la radiación ultravioleta', *IDEAM*. http://www.ideam.gov.co/web/tiempo-y-clima/generalidades-de-la-radiacion-ultravioleta (accessed Oct. 30, 2021).

[7] American Cancer Society, '¿Qué es el cáncer?', *American Cancer Society | Information and Resources*. https://www.cancer.org/es/cancer/aspectos-basicos-sobre-el-cancer/que-es-el-cancer.html (accessed Aug. 30, 2020).

[8] Instituto Nacional del Cáncer (NIH), 'Cualquier persona puede padecer cáncer de piel', *Instituto Nacional del Cáncer (NIH)*. https://www.cancer.gov/espanol/tipos/piel/cualquiera-padecer-cancer-piel (accessed Sep. 10, 2020).

[9] F. N. Díaz, C. Mosquera, and M. A. Ricci Lara, *Inteligencia Artificial en Imágenes Médicas de la teoría a la aplicación*. Buenos Aires, Argentina: Hospital Italiano de Buenos Aires, 2020.

[10] WORLD HEALTH ORGANIZATION: REGIONAL OFFICE FOR EUROPE, *WORLD CANCER REPORT: cancer research for cancer development*. S.l.: IARC, 2020.

[11] C. Garbe *et al.*, 'Time trends in incidence and mortality of cutaneous melanoma in Germany', *J. Eur. Acad. Dermatol. Venereol.*, vol. 33, no. 7, pp. 1272–1280, Jul. 2019, doi: 10.1111/jdv.15322.

[12] A. R. Rinflerch, V. I. Volonteri, M. C. Roude, L. V. Pagotto, M. Pol, and L. D. Mazzuoccolo, 'Trends in melanoma incidence at Hospital Italiano de Buenos Aires, 2007–2016', *An. Bras. Dermatol.*, p. S0365059621002646, Nov. 2021, doi: 10.1016/j.abd.2020.10.013.

[13] Department of Dermatology, The Warren Alpert Medical School, Brown University, Providence, RI, USA *et al.*, 'Epidemiology of Melanoma', in *Cutaneous Melanoma: Etiology*

and Therapy, Department of Surgical Oncology, Fox Chase Cancer Center, Philadelphia,PA, USA, W. H. Ward, J. M. Farma, and Department of Surgical Oncology, Fox Chase Cancer Center, Philadelphia,PA, USA, Eds. Codon Publications, 2017, pp. 3–22. doi: 10.15586/codon.cutaneousmelanoma.2017.ch1.

[14] CAEME, 'Prevención y tratamiento del cáncer de piel', *CAEME Innovación para la salud*. https://www.caeme.org.ar/prevencion-y-tratamiento-del-cancer-de-piel/ (accessed Apr. 15, 2020).

[15] Organización Médica Colegial de España (OMC) and Consejo General de Colegios Oficiales de Médicos, 'Más del 90% de los cánceres de piel se podrían evitar', Jun. 12, 2019. http://www.medicosypacientes.com/articulo/mas-del-90-de-los-canceres-de-piel-se-podrian-evitar (accessed Apr. 15, 2020).

[16] S. I. Ao and International Association of Engineers, Eds., *International Conference on Computational Biology 2013; International Conference on Chemical Engineering 2013; International Conference on Circuits and Systems 2013; International Conference on Communications Systems and Technologies 2013; International Conference on Machine Learning and Data Analysis 2013; International Conference on Modeling Health Advances 2013; International Conference on Modeling, Simulation and Control 2013*. Hong Kong: IAENG, International Association of Engineers, 2013.

[17] W, Stewart B, B. W. Stewart, C. Wild, International Agency for Research on Cancer, and World Health Organization, *World cancer report 2014*. 2014.

[18] World Cancer Research Fund, 'Diet, nutrition, physical activity and skin cancer - 2019'. Accessed: Sep. 09, 2020. [Online]. Available: https://www.wcrf.org/sites/default/files/Skin-cancer.pdf

[19] World Cancer Research Fund, 'Skin cancer statistics - Melanoma of the skin is the 19th most comon cancer worldwide', *World Cancer Research Fund*. https://www.wcrf.org/dietandcancer/cancer-trends/skin-cancer-statistics (accessed Sep. 09, 2020).

[20] Skin Cancer Foundation, 'Basal Cell Carcinoma, Warning Signs, Early Detection Best Practices', *Skin Cancer Foundation*. https://www.skincancer.org/skin-cancer-information/basal-cell-carcinoma/bcc-warning-signs-images/ (accessed Sep. 10, 2020).

[21] A. Hendi and J.-C. Martinez, Eds., *Atlas of skin cancers: practical guide to diagnosis and treatment*. Heidelberg: Springer, 2011.

[22] CDC: Centros para el Control y la Prevención de Enfermedades, 'Cáncer de piel', *CDC: Centros para el Control y la Prevención de Enfermedades*. https://www.cdc.gov/spanish/cancer/skin/index.htm (accessed Apr. 15, 2020).

[23] Ministerio de Salud, 'Estadísticas - Incidencia'. https://www.argentina.gob.ar/salud/instituto-nacional-del-cancer/estadisticas/incidencia (accessed Oct. 30, 2021).

[24] A. Esteva *et al.*, 'Dermatologist-level classification of skin cancer with deep neural networks', *Nature*, vol. 542, no. 7639, pp. 115–118, Feb. 2017, doi: 10.1038/nature21056.

[25] K. M. Hosny, M. A. Kassem, and M. M. Foaud, 'Skin Cancer Classification using Deep Learning and Transfer Learning', in *2018 9th Cairo International Biomedical Engineering Conference (CIBEC)*, Cairo, Egypt, Dec. 2018, pp. 90–93. doi: 10.1109/CIBEC.2018.8641762.

[26] Vijayalakshmi M M, 'Melanoma Skin Cancer Detection using Image Processing and Machine Learning', *Int. J. Trend Sci. Res. Dev. IJTSRD*, vol. 3, no. 4, Jun. 2019.

[27] S. Kalouche, 'Vision-Based Classification of Skin Cancer using Deep Learning'.

[28] Diagnóstico por Imágenes Junín, 'Dermatoscopía Digital: Diagnósticos por imágenes de alta resolución en Dermatología DIAR-D.', *Diagnóstico por Imágenes Junín*. https://diagporimagenes.com/video-dermatoscopia-digital/ (accessed Apr. 23, 2020).

[29] DermNet NZ, 'Teledermatology for suspected skin cancers', *DermNet NZ*. https://dermnetnz.org/cme/teledermatology-skin-cancer/dermatoscopes-and-cameras/ (accessed Aug. 21, 2021).

[30] E-Derma, 'Centro dermatológico dedicado específicamente al cáncer de piel', *E-Derma*. https://www.e-derma.com.ar/ (accessed Aug. 21, 2021).

[31] Elite Laser, 'Bodystudio', *Elite Laser*. https://www.elitelaser.es/tecnologia/fotofinder/bodystudio/ (accessed Aug. 21, 2021).

[32] J. F. Pillou, 'Vitropresión - Definición', Jan. 05, 2015. https://salud.ccm.net/faq/21300-vitropresion-definicion (accessed Oct. 30, 2021).

[33] Hospital Fuensanta, 'Dermatoscopía', *Hospital Fuensanta*. https://hospitalfuensanta.com/especialidades/dermatologia/dermatoscopia/ (accessed Oct. 30, 2021).

[34] Hospital Italiano de Buenos Aires, 'Sector DIAR-D (Diagnóstico por imágenes de alta resolución en dermatología)', *Hospital Italiano de Buenos Aires*. https://www1.hospitalitaliano.org.ar/#!/home/dermatologia/seccion/36640 (accessed Apr. 23, 2020).

[35] Hospital Alemán, 'Servicios Médicos: Dermatología', *Hospital Alemán*. https://www.hospitalaleman.org.ar/nuestro-hospital/servicios-medicos/dermatologia/?subpage=practicas (accessed Apr. 23, 2020).

[36] K. Munir, H. Elahi, A. Ayub, F. Frezza, and A. Rizzi, 'Cancer Diagnosis Using Deep Learning: A Bibliographic Review', *Cancers*, vol. 11, no. 9, p. 1235, Aug. 2019, doi: 10.3390/cancers11091235.

[37] Skin Cancer Foundation, 'Melanoma, Warning Signs, Early Detection Best Practices', *Skin Cancer Foundation*. https://www.skincancer.org/skin-cancer-information/melanoma/melanoma-warning-signs-and-images/ (accessed Sep. 10, 2020).

[38] A. Rosebrock, *Deep Learning for Computer Vision with Python: Starter Bundle*, 1st ed. PyImageSearch. [Online]. Available: https://www.pyimagesearch.com/deep-learning-computer-vision-python-book/

[39] C. Cao *et al.*, 'Deep Learning and Its Applications in Biomedicine', *Genomics Proteomics Bioinformatics*, vol. 16, no. 1, pp. 17–32, Feb. 2018, doi: 10.1016/j.gpb.2017.07.003.

[40] Y. Fujisawa, S. Inoue, and Y. Nakamura, 'The Possibility of Deep Learning-Based, Computer-Aided Skin Tumor Classifiers', *Front. Med.*, vol. 6, p. 191, Aug. 2019, doi: 10.3389/fmed.2019.00191.

[41] R. A. el-Azhary, 'The inevitability of change', *Clin. Dermatol.*, vol. 37, no. 1, pp. 4–11, Jan. 2019, doi: 10.1016/j.clindermatol.2018.09.003.

[42] A. Esteva *et al.*, 'A guide to deep learning in healthcare', *Nat. Med.*, vol. 25, no. 1, pp. 24–29, Jan. 2019, doi: 10.1038/s41591-018-0316-z.

[43] D. A. Okuboyejo, O. O. Olugbara, and S. A. Odunaike, 'Automating Skin Disease Diagnosis Using Image Classification', Hong Kong, 2013.

[44] H. A. Haenssle *et al.*, 'Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists', *Ann. Oncol.*, vol. 29, no. 8, pp. 1836–1842, Aug. 2018, doi: 10.1093/annonc/mdy166.

[45] Á. Iglesias-Puzas and P. Boixeda, 'Deep learning y DerMATología', *Actas Dermo-Sifiliográficas*, vol. 111, no. 3, pp. 192–195, Apr. 2020, doi: 10.1016/j.ad.2019.01.014.

[46] M. A. Marchetti *et al.*, 'Results of the 2016 International Skin Imaging Collaboration International Symposium on Biomedical Imaging challenge: Comparison of the accuracy of computer algorithms to dermatologists for the diagnosis of melanoma from dermoscopic images', *J. Am. Acad. Dermatol.*, vol. 78, no. 2, pp. 270-277.e1, Feb. 2018, doi: 10.1016/j.jaad.2017.08.016.

[47] A. Hekler *et al.*, 'Superior skin cancer classification by the combination of human and artificial intelligence', *Eur. J. Cancer*, vol. 120, pp. 114–121, Oct. 2019, doi: 10.1016/j.ejca.2019.07.019.

[48] N. Gessert, M. Nielsen, M. Shaikh, R. Werner, and A. Schlaefer, 'Skin Lesion Classification Using Ensembles of Multi-Resolution EfficientNets with Meta Data', *ArXiv191003910 Cs*, Oct. 2019, Accessed: Oct. 31, 2021. [Online]. Available: http://arxiv.org/abs/1910.03910

[49] S. S. Han, M. S. Kim, W. Lim, G. H. Park, I. Park, and S. E. Chang, 'Classification of the Clinical Images for Benign and Malignant Cutaneous Tumors Using a Deep Learning Algorithm', *J. Invest. Dermatol.*, vol. 138, no. 7, pp. 1529–1538, Jul. 2018, doi: 10.1016/j.jid.2018.01.028.

[50] T. J. Brinker *et al.*, 'Skin Cancer Classification Using Convolutional Neural Networks: Systematic Review', *J. Med. Internet Res.*, vol. 20, no. 10, p. e11936, 17 2018, doi: 10.2196/11936.

[51] T. J. Brinker *et al.*, 'Deep learning outperformed 136 of 157 dermatologists in a head-to-head dermoscopic melanoma image classification task', *Eur. J. Cancer*, vol. 113, pp. 47–54, May 2019, doi: 10.1016/j.ejca.2019.04.001.

[52] T. J. Brinker *et al.*, 'Deep neural networks are superior to dermatologists in melanoma image classification', *Eur. J. Cancer*, vol. 119, pp. 11–17, Sep. 2019, doi: 10.1016/j.ejca.2019.05.023.

[53] A. Lallas and G. Argenziano, 'Artificial intelligence and melanoma diagnosis: ignoring human nature may lead to false predictions', *Dermatol. Pract. Concept.*, pp. 249–251, Oct. 2018, doi: 10.5826/dpc.0804a01.

[54] Start-Tech Academy, *Machine Learning & Deep Learning in Python & R*. Udemy Academy. Accessed: Apr. 09, 2021. [Online]. Available: https://www.udemy.com/course/data_science_a_to_z/

[55] NVIDIA, '¿QUÉ ES LA COMPUTACIÓN ACELERADA POR GPU?' https://www.nvidia.com/es-la/drivers/what-is-gpu-computing/ (accessed Sep. 04, 2021).

[56] Google, 'Preguntas frecuentes de Colaboratory'. https://research.google.com/colaboratory/intl/es/faq.html (accessed Sep. 05, 2021).

[57] D. Shiffman, *The nature of code: simulating natural systems with processing*, Version 1.0, Generated December 6, 2012. s.l.: Selbstverl., 2012.

[58] S. Raschka and V. Mirjalili, *Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow*, Second edition, Fourth release,[fully revised and Updated]. Birmingham Mumbai: Packt Publishing, 04.

[59] 'Diagram of a Perceptron'. https://tex.stackexchange.com/questions/104334/tikz-diagram-of-a-perceptron

[60] C. M. Bishop, *Pattern recognition and machine learning*, Corrected at 8th printing 2009. New York, NY: Springer, 2009.

[61] G. E. Liberman, 'Clasificación de imágenes de melanomas mediante redes profundas y vectores de Fisher', Tésis de Magíster en Explotación de Datos y Descubrimiento del Conocimiento, UBA - Facultad de Ciencias Exactas y Naturales, Departamento de Ciencias de la Computación, Buenos Aires, Argentina, 2019.

[62] K. Cortés Aguas, 'A guide to transfer learning with Keras using ResNet50', *Medium*, Jul. 03, 2020. https://medium.com/@kenneth.ca95/a-guide-to-transfer-learning-with-keras-using-resnet50-a81a4a28084b

[63] A. Ye, 'A Guide to Neural Network Layers with Applications in Keras', *Medium*, Mar. 03, 2020. https://medium.com/analytics-vidhya/a-guide-to-neural-network-layers-with-applications-in-keras-40ccb7ebb57a

[64] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*, Second edition. Beijing [China]; Sebastopol, CA: O'Reilly Media, Inc, 2019.

[65] A. Rezvantalab, H. Safigholi, and S. Karimijeshni, 'Dermatologist Level Dermoscopy Skin Cancer Classification Using Different Deep Learning Convolutional Neural Networks Algorithms', *ArXiv181010348 Cs Stat*, Oct. 2018, Accessed: Oct. 26, 2021. [Online]. Available: http://arxiv.org/abs/1810.10348

[66] F. Chollet, 'Building powerful image classification models using very little data', *The Keras Blog*, Jun. 05, 2016. https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html (accessed Oct. 23, 2021).

[67] ISIC Archive, 'The International Skin Imaging Collaboration'. https://www.isic-archive.com/#!/topWithHeader/wideContentTop/main (accessed Oct. 23, 2020).

[68] A. Rosebrock, *Deep Learning for Computer Vision with Python: Practitioner Bundle*, 1st ed. PyImageSearch. [Online]. Available: https://www.pyimagesearch.com/deep-learning-computer-vision-python-book/

[69] Stack Overflow user, 'Does ImageDataGenerator add more images to my dataset?', *Stack Overflow*. https://stackoverflow.com/questions/51748514/does-imagedatagenerator-add-more-images-to-my-dataset (accessed Oct. 08, 2021).

[70] G. Boesch, 'Deep Residual Networks (ResNet, ResNet50) – Guide in 2021', *29 august 2021*. https://viso.ai/deep-learning/resnet-residual-neural-network/ (accessed Oct. 10, 2021).

[71] Opengenus, 'Understanding ResNet50 architecture'. https://iq.opengenus.org/resnet50-architecture/ (accessed Oct. 10, 2021).

[72] K. Gak, 'When to use Dense, Conv1/2D, Dropout, Flatten, and all the other layers?', *Data Science Stack Exchange*, Jan. 17, 2019. https://datascience.stackexchange.com/questions/44124/when-to-use-dense-conv1-2d-dropout-flatten-and-all-the-other-layers

[73] D. Godoy, 'Understanding binary cross-entropy / log loss: a visual explanation', *Towards Data Science*, Nov. 21, 2018. https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a (accessed Oct. 24, 2021).

[74] Keras API reference, 'Adam', *Keras*. https://keras.io/api/optimizers/adam/ (accessed Nov. 01, 2021).

[75] D. P. Kingma and J. Ba, 'Adam: A Method for Stochastic Optimization', *ArXiv14126980 Cs*, Jan. 2017, Accessed: Nov. 01, 2021. [Online]. Available: http://arxiv.org/abs/1412.6980

[76] J. Brownlee, 'Gentle Introduction to the Adam Optimization Algorithm for Deep Learning', *Machine Learning Mastery*, Jul. 03, 2017. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/ (accessed Nov. 01, 2021).

[77] D. Yastremsky, 'Exploit Your Hyperparameters: Batch Size and Learning Rate as Regularization', *Towards Data Science*, Mar. 04, 2021. https://towardsdatascience.com/exploit-your-hyperparameters-batch-size-and-learning-rate-as-regularization-9094c1c99b55 (accessed Nov. 01, 2021).

[78] J. Brownlee, 'How to Configure the Learning Rate When Training Deep Learning Neural Networks', *Machine Learning Mastery*, Jan. 23, 2019. https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/ (accessed Nov. 01, 2021).

[79] Hao, 'Why are BatchNorm layers set to trainable in a frozen model'. https://forums.fast.ai/t/why-are-batchnorm-layers-set-to-trainable-in-a-frozen-model/46560/2

[80] J. Brownlee, 'How to Use ROC Curves and Precision-Recall Curves for Classification in Python', Aug. 31, 2018. https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/ (accessed Oct. 24, 2021).

[81] Digital Guide IONOS, '¿Qué es una interfaz gráfica de usuario (GUI)?', *Digital Guide IONOS*. https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-una-gui/ (accessed Aug. 14, 2021).

[82] J. Brownlee, 'How to Make Predictions with Keras', *Machine Learning Mastery*, Aug. 27, 2020. https://machinelearningmastery.com/how-to-make-classification-and-regression-predictions-for-deep-learning-models-in-keras/ (accessed Sep. 23, 2021).

[83] J. Mongan, L. Moy, and C. E. Kahn, 'Checklist for Artificial Intelligence in Medical Imaging (CLAIM): A Guide for Authors and Reviewers', *Radiol. Artif. Intell.*, vol. 2, no. 2, p. e200029, Mar. 2020, doi: 10.1148/ryai.2020200029.

[84] J. Eng, 'Sample Size Estimation: How Many Individuals Should Be Studied?', *Radiology*, vol. 227, no. 2, pp. 309–313, May 2003, doi: 10.1148/radiol.2272012051.

[85] A. Rosebrock, *Deep Learning for Computer Vision with Python: ImageNet Bundle*, 1st ed. PyImageSearch. [Online]. Available: https://www.pyimagesearch.com/deep-learning-computer-vision-python-book/