

Diseño de una Herramienta para la Clasificación de Imágenes de Cáncer de Piel  
en un Sistema Android

Duván Felipe González Sandoval

Sebastián Rivera Muñoz

Trabajo de Grado para Optar el Título de Ingeniero Electrónico.

Director

Ing. Jeison Arley Castillo Bohorquez

Co-Directores

MSc. Jaime Guillermo Barrero Pérez.

Ing. Diana Paola Merchán Vargas.

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Bucaramanga

2022

### **Dedicatoria**

En primer lugar a Dios quién guió mi camino y me dio todas las herramientas necesarias para culminar con éxito esta gran etapa en mi vida.

A mis padres Elsa Sandoval García y Rafael Antonio González Almeyda quienes han sido pilar fundamental en mi vida. A mi madre por enseñarme a través de su ejemplo la disciplina y perseverancia que se debe tener para lograr los objetivos, por seguir adelante cuando el camino se complica y siempre confiar en mí. A mi querido padre por su entrega y ayuda incondicional, por todo el amor que me demostró con sus acciones.

A cada miembro de mi familia que me brindó su ayuda.

A mis amigos de la universidad por su constante apoyo durante la vida universitaria, quienes a través de su amor se convirtieron en pieza fundamental de este logro.

A Danilo Ramirez por su amistad incondicional, por estar ahí en aquella etapa tan retadora de mi vida y también por disfrutar conmigo de nuestros logros.

A mis amigos del ciclismo donde el ejercicio se convirtió en terapia y el grupo en una familia.

A todos ustedes dedico esto.

**Duván Felipe González Sandoval**

### **Dedicatoria**

Esta tesis esta dedicada:

A Dios, que me demuestra su amor incondicional mediante cada bendición que me da  
diariamente.

A mis padres, hermano y demás familiares que me ayudaron a cumplir este objetivo. Su ayuda, paciencia, sacrificio y esfuerzo fueron claves para ser la persona y profesional que soy hoy en día. A todos mis amigos de la carrera y de la vida, gracias por esos consejos, por esa ayuda brindada y por esas tardes de risas. Toda su incondicionalidad fue clave para lograr este objetivo.

**Sebastián Rivera Muñoz**

### **Agradecimientos**

A nuestro director, Jeison Arley Castillo Bohorquez, por la confianza depositada, por su constante guía y motivarnos en cada etapa de nuestro proyecto.

A nuestra codirectora, Diana Paola Merchan Vargas, por compartir con nosotros su experiencia y sabiduría.

A mi alma máter Universidad Industrial de Santander la cuál me brindo la oportunidad de formarme como profesional.

A la escuela de ingeniería eléctrica, electrónica y de Telecomunicaciones E3T, por permitirme conocer a grandes personas que se convirtieron en familia.

A mi familia por su constante apoyo y consejos, por la confianza en mi durante todo este proceso de formación.

A mis compañeros y amigos que fueron fundamentales en los momentos mas retadores para continuar por este camino, a mis profesores, tutores y demás personas que aportaron a mi formación profesional y personal durante mi formación académica.

**Duván Felipe González Sandoval**

Primero agradecer a Dios por darme la oportunidad de estudiar en tan excelente Alma mater y permitirme crecer como persona y profesional.

Mi profundo agradecimiento a mis padres, hermano y demás familiares que con su acompañamiento, consejos y ayuda permitieron comenzar y terminar este camino satisfactoriamente, sin ustedes no hubiese sido posible este logro.

Mi más grande agradecimiento al director, codirectores, profesores y tutores que me acompañaron en el camino. Con su sabiduría, paciencia, palabras de aliento y motivación permiten formar un profesional íntegro que aportará su grano de arena para mejorar su entorno.

Por último pero no menos importante, a todos mis compañeros y amigos que me acompañaron en la carrera. Sin su amistad y acompañamiento esto no se hubiese logrado. Soy un completo convencido que Dios me puso las personas indicadas en el camino para poder terminar la carrera exitosamente. Sin ellos, esto no se hubiera logrado.

**Sebastián Rivera Muñoz**

## Tabla de Contenido

<b>Introducción</b>	<b>14</b>
<b>1 Objetivos</b>	<b>18</b>
1.1 Objetivo General	18
1.2 Objetivos Específicos	18
<b>2 Base de Datos</b>	<b>19</b>
2.1 Benigno vs. Maligno	19
2.1.1 Elementos Repetidos en la Base	19
2.1.2 Clasificación de las Imágenes	20
2.1.3 Subclasificación de las Imágenes para Entrenamiento y Validación	20
2.2 Tres Clases	21
<b>3 Redes Neuronales Residuales (ResNet)</b>	<b>22</b>
3.1 El Vanishing Gradient o Desvanecimiento de Gradiente	23
3.2 Definición	26
<b>4 Knowledge Distillation</b>	<b>28</b>
4.1 Función de Pérdida	29
4.2 Tipos de Destilación	30

4.2.1	Destilación Basada en Respuestas	30
4.2.1.1	Función de Pérdida por Destilación	30
4.2.1.2	Predicciones Blandas o Soft Predictions	30
4.2.2	Destilación Basada en Características	32
4.2.3	Destilación Basada en Relaciones	33
4.3	Modos de Destilación	33
4.3.1	Destilación Fuera de Línea	33
4.3.2	Destilación en Línea	34
4.3.3	Autodestilación	34
4.4	F1 Score	35
4.4.1	Precisión o Accuracy	35
4.4.2	Recuperación o Recall	36
<b>5</b>	<b>Análisis Previo de Resultados</b>	<b>37</b>
5.1	Salida de Dos Clases	37
5.2	Salida de tres clases	38
<b>6</b>	<b>Creación del Modelo Estudiante para el Knowledge Distillation</b>	<b>38</b>
<b>7</b>	<b>Búsqueda de Hiperparámetros para la Aplicación de Knowledge Distillation</b>	<b>42</b>
7.1	Salida de Dos Clases	42
7.2	Salida de Tres Clases	44

<b>8 Implementación del Modelo en un Sistema Android</b>	<b>48</b>
8.1 Desarrollo y Montaje del Modelo	48
8.2 Implementación del Modelo en Sistemas Android	49
8.2.1 Desarrollo y Montaje del Modelo	49
8.3 Diseño y Funcionalidad de la Aplicación	50
8.3.1 Procesamiento de las Imágenes	51
8.4 Evaluación de Resultados	52
8.4.1 Salida de Dos Clases	52
8.5 Salida de Tres Clases	53
<b>9 Conclusiones</b>	<b>54</b>
<b>10 Trabajo Futuro</b>	<b>55</b>
<b>Referencias Bibliográficas</b>	<b>57</b>

### Lista de Figuras

Figura 1	Configuración de una red neuronal	24
Figura 2	Gráfica función sigmoide y su derivada	26
Figura 3	Diagrama básico de una red neuronal residual ResNet	27
Figura 4	Bloque residual	28
Figura 5	Esquema de los tipos de destilación	34
Figura 6	Matriz de confusión.	36
Figura 7	Arquitectura de la red neuronal propuesta para el modelo <i>Student</i>	40
Figura 8	F1-score y parámetros vs. bloques residuales	42
Figura 9	Matriz de confusión para una salida de dos clases	44
Figura 10	Matriz de confusión para una salida de tres clases	46
Figura 11	Código usado para generar el archivo .tflite	49
Figura 12	Interfaz de usuario de la aplicación Android	51
Figura 13	Matriz de confusión de dos clases	53
Figura 14	Matriz de confusión de tres clases	53

**Lista de Tablas**

Tabla 1	Distribución de clases en la base de datos.	20
Tabla 2	Distribución de clases entre entrenamiento y test en la base de datos.	21
Tabla 3	Comparación entre las bases de datos	22
Tabla 4	Distribución de la base de datos de los tres tipos de cáncer de piel con mayor incidencia en Colombia.	22
Tabla 5	Comparación de resultados de la red MobileNet V1 con diferentes bases de datos.	38
Tabla 6	Comparación de resultados de la red Inception V3 con diferentes bases de datos.	38
Tabla 7	Comparación de parámetros y consumo de memoria entre la red MobileNet V1 y la red Inception V3.	39
Tabla 8	Puntaje $F1 - score$ para una clasificación de tres clases	39
Tabla 9	Resultados obtenidos para cada valor de $K$	41
Tabla 10	Mejores modelos obtenidos a partir de los experimentos.	43
Tabla 11	Comparación de puntuación F1-score entre el modelo <i>Teacher</i> y <i>Student</i> para una salida de dos clases	45
Tabla 12	Comparación de número de parámetros y memoria entre el modelo <i>Teacher</i> y <i>Student</i> para una salida de dos clases y su respectivo nivel de compresión.	45
Tabla 13	Mejores modelos del estudiante destilado para una salida de tres clases	47

Tabla 14	Comparación de número de parámetros y memoria entre el modelo <i>Teacher</i> y <i>Student</i> para una salida de tres clases y su respectivo nivel de compresión.	48
Tabla 15	Comparación de resultados entre el modelo <i>Teacher</i> y <i>Student</i> para una salida de tres clases	48
Tabla 16	Comparación de resultados entre la red implementada en Python y la aplicación en Android para una salida de dos clases.	52
Tabla 17	Comparación de resultados entre la red implementada en Python y la aplicación en Android para una salida de tres clases.	54

## Resumen

**Título:** Diseño de una herramienta para la clasificación de imágenes de cáncer de piel en un sistema android. \*

**Autores:** Duván Felipe González Sandoval y Sebastián Rivera Muñoz. \*\*

**Palabras Clave:** Cáncer de Piel, Deep Neural Networks (DNN), Knowledge Distillation, Redes Neuronales Residuales, Android.

**Descripción:** El cáncer de piel pertenece a los diez tipos de cáncer más comunes que se registran en todo el mundo. Según la OMS, en 2020 registraron más de un millón de casos nuevos de Melanoma (OMS, 2022). Esta es una preocupación latente, ya que su crecimiento continúa y su diagnóstico puede ser demorado, o en algunos casos, inexistente para personas carentes de un sistema de salud. En la actualidad, las redes neuronales permiten una clasificación de imágenes con gran probabilidad de éxito. Por esta razón, esta investigación se centra en diseñar una herramienta que permita clasificar imágenes dermatoscópicas a través de una aplicación en un sistema Android. Tomando como punto de partida el trabajo realizado por Diana Merchán y Helis Navarro (Merchán Vargas *et al.*, 2021) se aplicó una técnica conocida como destilación del conocimiento (*Knowledge Distillation*) donde un modelo más pequeño (Student) aprende de uno más grande (Teacher). Esta técnica permitió comprimir el modelo y mantener su rendimiento. Se pasó de tener un rendimiento de 0.9332 a 0.9557 (puntuación F1-score) para la clasificación entre imágenes cancerígenas de tipo Benigno v.s Maligno. Además, se logró una compresión del 30.82x en términos de parámetros con respecto al modelo original. Estos resultados fueron obtenidos haciendo un ajuste fino en los hiperparámetros y seleccionando los mejores. De igual manera se replicó la metodología para un modelo de tres clases, pasando de un rendimiento de 0.7815 para el modelo *Teacher* a 0.8995 para el modelo *Student*. A su vez, se logró una compresión del 23.35x con respecto al modelo original. Finalmente, los dos mejores modelos correspondientes a cada tipo de entrada, fueron implementados en un sistema Android donde se puede tomar o cargar una foto y hacer su respectiva inferencia.

---

\* Trabajo de Grado.

\*\* Facultad de Ingeniería Fisicomecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones. Director: Jeison Arley Castillo Bohórquez; Co-Directores: Jaime Guillermo Barrero Pérez y Diana Paola Merchán Vargas

## Abstract

**Title:** Design of a tool for the classification of skin cancer images in an Android system. \*

**Authors:** Duván Felipe González Sandoval and Sebastián Rivera Muñoz. \*\*

**Keywords:** Skin Cancer, Deep Neural Networks (DNN), Knowledge Distillation, residual neural networks, Android.

**Description:** Skin cancer is the ten most common types of cancer in the world. According to WHO, in 2020 there were more than one million new cases of melanoma (OMS, 2022). This is worrying because is on the rise and its diagnosis can be late, even, non-existent for people who don't have access to a health center. Currently, neural networks allow classifying images successfully. For this reason, this investigation designed a tool to classify dermatoscopic images through an application in an android system. The article by Diana Merchan and Helis Navarro (Merchán Vargas *et al.*, 2021) was the starting point for applying the Knowledge distillation technique where a big model teaches (teacher) to a small model (student). This technique allows to compress a model and it keeps its efficiency. It allowed increasing from 0.9332 to 0.9557 (F1-Score parameter) to classify images of benign and malignant cancer. Also, the two classes model is compressed by a factor of 30.82x compared with the original model. In the three classes model, the efficiency increase was from 0.7815 to 0.8995, from the teacher model to the student model. Also, the three classes model is compressed by a factor of 23.35x compared with the original model. These results were gotten by setting hyperparameters and selecting the best. Finally, the best model for each type of input was implemented in an android system that can take or load a photo for its inference.

---

\* Undergrade Work

\*\* Faculty of Engineering Physicomechanics. School of Electrical, Electronic and Telecommunications Engineering. Advisor: Jeison Arley Castillo Bohórquez; Co-Advisors: Jaime Guillermo Barrero Pérez and Diana Paola Merchán Vargas

## Introducción

El cáncer de piel es una enfermedad donde las células tienen un crecimiento anormal y descontrolado. Se puede dividir en dos grandes grupos: no melanoma y melanoma maligno (Benigno y Maligno), siendo este último el de más gravedad, ya que representa la mayor cantidad de muertes por cáncer de piel en el mundo (Mayo, 2020). Una de las causas principales es el exceso de exposición solar, en particular, a los rayos ultravioleta (UV). La mayor parte de esta exposición viene en los primeros años de vida, específicamente, en las actividades de recreación y esparcimiento, pero también puede venir de actividades laborales tales como la construcción, ventas ambulantes y el agro. También los deportistas están ampliamente expuestos a los rayos UV (Social, 2015).

A nivel mundial, los pacientes con cáncer de piel han aumentado. En el año 2020 se presentaron más de 1.500.000 casos nuevos de cáncer de piel, reportando así 324.635 nuevos casos de cáncer tipo Melanoma y 1.198.073 nuevos casos de tipo no Melanoma. Según la Liga contra el Cáncer, esto podría relacionarse con cambios en el clima, y aumento en la expectativa de vida (Social, 2020a).

Ante esta problemática, el gobierno nacional de Colombia tiene normativas contempladas en la Ley 1384 de 2010 (Bogotá, 2010) que contribuyen al posicionamiento en la agenda política de la prevención del cáncer en general y del cáncer de piel en particular (Social, 2020b). En Colombia se duplicaron los casos reportados de cáncer de piel desde el 2015 al 2019, pasando en 2015 de

53.622 pacientes con cáncer de piel a 128.542 en 2019, siendo las personas mayores de 50 años las más afectadas (Social, 2020b). Es importante destacar que Colombia se ubica sobre la línea del ecuador y se convierte en uno de los países con mayor índice de radiación solar, y así mismo, esto los hace más vulnerables al cambio climático. Por consiguiente, se tienen mayores probabilidades de desarrollar alguna enfermedad cutánea (Instituto de Hidrología, 2015).

Según datos de la asociación colombiana de dermatología y cirugía dermatológica (la cual incluye a un 90 % de los dermatólogos de todo el país) se estima que por cada 100.000 habitantes existen 1.25 dermatólogos, los cuales a su vez están mal distribuidos, concentrándose una cifra de 3 dermatólogos por cada 100.000 pacientes en las ciudades y una oferta rural pobre. Comparando estas cifras con un país como Estados Unidos donde existen 2.8 a 3 dermatólogos por cada 100.000 pacientes, deja ver que el sistema de salud colombiano delega que un médico general deba hacerse cargo de muchos pacientes con enfermedades dermatológicas, como el cáncer de piel (Ramírez *et al.*, 2012). La importancia de que exista la cantidad de dermatólogos suficientes para atender la demanda de pacientes con cáncer de piel es poder detectar esta enfermedad en sus primeras fases para acudir a una mayor cantidad de tratamientos que poseen mayor eficacia y menor agresividad en los pacientes (Hoshyar *et al.*, 2011).

Por otra parte, la inteligencia artificial ha demostrado una gran capacidad en la clasificación de imágenes y el uso del Deep Learning con implementaciones de redes neuronales ha servido como herramienta de apoyo a los profesionales de la salud para dar diagnósticos clínicos en lapsos de tiempo menores, con un mayor índice de confiabilidad como lo es en el caso de la detección

temprana de cáncer de seno, de piel, entre otros (Haglin *et al.*, 2019).

En los últimos años, los algoritmos de aprendizaje han aportado a la clasificación de imágenes médicas para la detección de cáncer de piel. En Esteva *et al.* (2017) se informa que por medio de redes neuronales convolucionales (CNN) se puede tener una precisión comparable a la de un dermatólogo. En Alquran *et al.* (2017), por medio de un sistema de máquina vectorial (SMV) se logró alcanzar una precisión del 92.1% para clasificación de melanoma. En Brinker *et al.* (2018) realizaron una revisión sistemática de las investigaciones más avanzadas sobre la clasificación de lesiones cutáneas con redes neuronales convolucionales y manifiestan la necesidad de realizar implementaciones de estas redes neuronales en dispositivos móviles, con el fin de dar diagnósticos rápidos, incluso fuera de un hospital. Merchán Vargas *et al.* (2021) usó las ventajas de las redes neuronales profundas (DNN) para construir una red neuronal convolucional (CNN), la red permite la clasificación de lesiones cutáneas benignas y malignas, adicionalmente realizan una implementación en un sistema embebido con cámara y se implementa la CNN realizada. Google (2021) logra crear una aplicación para la detección de cáncer de piel, la cual surge ante la identificación de miles de millones de búsquedas realizadas al año en el buscador de Google, a su vez esta aplicación permite detectar también afecciones en cabello y uñas.

Los anteriores trabajos mencionados proveen una fuerte evidencia hacia el desarrollo de aplicaciones médicas para el diagnóstico temprano del cáncer de piel. Por lo tanto, en este trabajo se propone un esquema de deep learning basado en redes neuronales residuales para la clasificación de cáncer piel en un dispositivo de uso final (celular con plataforma Android). Es importante aclarar

que este trabajo se realizará partiendo de las investigaciones y resultados previos realizados en (Merchán Vargas *et al.*, 2021).

La primera parte de este trabajo trata sobre la base de datos, cómo se procesaron y seleccionaron los datos para el entrenamiento y validación. El siguiente título presenta conceptos relacionados con la técnica de Knowledge Distillation para el entrenamiento de redes neuronales con destilación de conocimiento. Posteriormente, se presenta la serie de experimentos y ajuste fino de parámetros para la búsqueda del mejor modelo, teniendo en cuenta la preservación de la precisión del modelo y así mismo, buscando la mayor compresión del mismo. Por último se presentan los modelos finales obtenidos para clasificar tanto imágenes de cáncer de piel maligno y benigno como de los 3 tipos de cáncer con mayor incidencia en Colombia (Carcinoma, Melanoma y Queratosis). Finalmente, las redes se implementaron en un sistema Android XOS (Basado en Android 11) con una aplicación que permite tomar y cargar fotografías para la clasificación.

## **1. Objetivos**

### **1.1. Objetivo General**

Diseñar una herramienta basada en CNNs la cual se implementará en una aplicación móvil para la clasificación de imágenes de cáncer de piel mediante el uso de redes neuronales profundas preentrenadas.

### **1.2. Objetivos Específicos**

Aplicar técnicas de compresión sobre una red neuronal profunda preentrenada para la clasificación de cáncer de piel.

Evaluar el desempeño de la red neuronal comprimida.

Implementar la red neuronal profunda en un dispositivo Android.

Evaluar el funcionamiento de la herramienta teniendo en cuenta la precisión del modelo con respecto a la red original.

## 2. Base de Datos

### 2.1. Benigno vs. Maligno

Para la selección de la base de datos, se tomaron las imágenes de benigno y maligno de *ISIC – Archive* con un total de 7826 imágenes (Katanskiy, 2022). Estas imágenes son de carácter público y fueron escogidas con el fin de garantizar que cada imagen correspondiera a un paciente diferente. Esta identificación se hizo por medio de metadatos contenidos en un archivo proporcionado por la organización ISIC, que presenta la información clínica de cada caso. Entre estos datos se observa la edad, el sexo del paciente, su respectiva clasificación en la categoría de cáncer y su ID de identificación única para cada caso. Antes de ser usada la base de datos, se hicieron una serie de preparaciones que se detallan a continuación:

#### 2.1.1. Elementos Repetidos en la Base

Entre las imágenes de benigno y maligno se cuenta con un total de 7826 imágenes, de las cuales existen 8 imágenes que tenían el mismo nombre o identificación, es decir, estaban repetidas. Para esto se hizo un programa en Python que leyera cada imagen, verificara existencia de imágenes repetidas y las eliminara. Al finalizar esta parte, se cuenta con un total de 7818 imágenes entre las clases de Benigno y Maligno (González y Rivera, 2022).

Tabla 1

*Distribución de clases en la base de datos.*

Clase	Número de imágenes
Benigno	6705
Maligno	1113

### ***2.1.2. Clasificación de las Imágenes***

Dentro de las 7818 imágenes, se encuentran imágenes clasificadas como cáncer tipo benigno y cáncer tipo maligno de forma aleatoria. De esta manera, se lee cada imagen con su respectiva identificación y clasificación y se ubican en carpetas diferentes. Este código se desarrolló en Python y está disponible en el repositorio de GitHub (González y Rivera, 2022). Finalmente, se obtienen 6705 imágenes tipo benigno y 1113 imágenes tipo maligno (Ver Tabla 1).

### ***2.1.3. Subclasificación de las Imágenes para Entrenamiento y Validación***

Una vez realizada la clasificación de imágenes en benigno y maligno, se procede a separar los grupos de entrenamiento y test. Debido al tamaño de la base de datos, y a las recomendaciones proporcionadas en el estado del arte, se toma una proporción del 80% para entrenamiento y un 20% para test, esta separación se encuentra presente en la Tabla 2.

Tabla 2

*Distribución de clases entre entrenamiento y test en la base de datos.*

	Benigno	Maligno	Total	Porcentaje
Entrenamiento	5364	890	6254	80 %
Validación	1341	223	1564	20 %
<b>Total</b>	<b>6705</b>	<b>1113</b>	<b>7818</b>	<b>100 %</b>

En la tabla 3 se compara la base de imágenes resultante con la base de datos usada por Merchán Vargas *et al.* (2021) para Benigno y Maligno. Estas bases de datos serán usadas por separado para entrenar los dos mejores modelos obtenidos por el estudio de Merchán Vargas *et al.* (2021) y escoger la relación arquitectura/base de datos que obtenga el mejor valor de F1-score como el modelo **teacher**.

## 2.2. Tres Clases

La base de datos usada para la clasificación de las tres clases más comunes de cáncer de piel en Colombia es la usada en Merchán Vargas *et al.* (2021) y es diferente a la base de datos usada para las dos clases Benigno y Maligno (ver Tabla 2). La Tabla 4 muestra la cantidad de imágenes con su respectiva separación para entrenamiento y test. Cabe resaltar que el porcentaje de clasificación viene dado por el 86 % para entrenamiento - validación y 14 % para test.

Tabla 3

*Comparación entre las bases de datos*

Clase	Número de imágenes base resultante	Número de imágenes usada en Merchán Vargas <i>et al.</i> (2021)
Benigno	6705	2464
Maligno	1113	2463
Entrenamiento - Validación Benigno	5364	2200
Entrenamiento - Validación Maligno	890	2200
Test Benigno	1341	264
Test Maligno	223	263

### 3. Redes Neuronales Residuales (ResNet)

Una red neuronal clásica tiene entradas, salidas y un sistema conectado entre sí llamado capas. El número de capas puede variar dependiendo de la profundidad del aprendizaje realizado a la red y a esto se le conoce como profundidad: a mayor número de bloques o capas intermedias,

Tabla 4

*Distribución de la base de datos de los tres tipos de cáncer de piel con mayor incidencia en Colombia.*

	Carcinoma	Melanoma	Queratosis	Total
Entrenamiento - Validación	675	1328	1000	3003
Test	136	239	99	474
<b>TOTAL</b>	<b>811</b>	<b>1567</b>	<b>1099</b>	<b>3477</b>

mayor profundidad. Se ha demostrado que el incremento en la profundidad de las redes mejora notablemente los resultados de clasificación (He *et al.*, 2015). Pero a su vez incrementar demasiado el número de capas puede llevar al modelo a dificultar su aprendizaje. Esto debido a que:

- Mas profundidad implica una propagación del error mucho más amplia. Esto hace que el error de inferencia crezca llegando a crear conflicto según el número de capas y clases del modelo (He *et al.*, 2015).
- Mas profundidad puede significar más parámetros y el algoritmo incrementa su número de operaciones y MFLOPS (Tarrés Benet, 2019).
- El algoritmo converge más lento, ya que cada capa puede añadir un mayor nivel de abstracción (Tarrés Benet, 2019).
- En la retro-propagación, el gradiente se desvanece con la profundidad, lo que impide un buen ajuste de las capas más cercanas a las entradas. A este fenómeno se le describe como *Vanishing Gradient* (He *et al.*, 2015).

### 3.1. El Vanishing Gradient o Desvanecimiento de Gradiente

Para el entrenamiento de las redes neuronales existe un método muy conocido de optimización descenso de gradiente. Este método permite encontrar el punto del error mínimo en la función de coste donde se compara la salida generada por la red y la salida teórica o deseada. Esto permite ajustar los pesos sinápticos en cada capa de la red neuronal y reducir el error asociado a la salida.

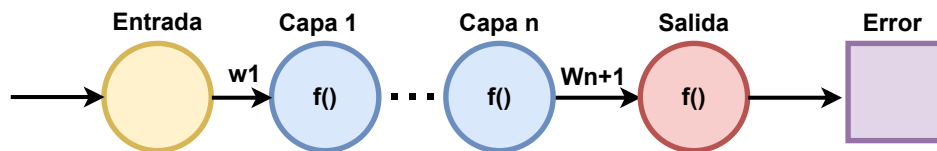
Muchos métodos como RMSprop y ADAM están basados en modificaciones del método pero en esencia todos son similares.

La metodología del proceso es la siguiente: El procedimiento hecho por la red permite que esta genere unos datos en la salida dada una entrada, se denomina propagación hacia adelante o forward propagation. Por el contrario, la propagación hacia atrás o Backpropagation tiene que ver con la propagación del error que se realiza desde las últimas capas hacia las primeras con el fin de reducirlo (Barrera, 2018). Sin embargo, cuando se realiza la retropropagación en redes que tienen múltiples capas se presenta el problema del desvanecimiento del gradiente, que consiste en la disminución del gradiente de la función que hace que el entrenamiento de la red no se realice de la mejor manera.

Para entender un poco mejor este problema, observemos la configuración de red neuronal de la Figura 1, en ella se observa la entrada, la salida, los pesos correspondientes a cada capa y el error producido por la salida de una red neuronal. Usando la regla de la cadena o Chain rule, la derivada respecto al peso de la primera capa sería:

*Figura 1.*

Configuración de una red neuronal



$$\frac{\partial J}{\partial W_1} = \frac{\partial J}{\partial Salida} * \frac{\partial Salida}{\partial Capa_N} \dots \frac{\partial Capa_2}{\partial Capa_1} * \frac{\partial Capa_1}{\partial W_1} \quad (1)$$

Tomando cualquier expresión de la derivada de la salida respecto a la  $Capa_N$ , se tiene que:

$$\frac{\partial Salida}{\partial Capa_N} = \frac{\partial f(z_1)}{\partial z_1} W_{n+1} \quad (2)$$

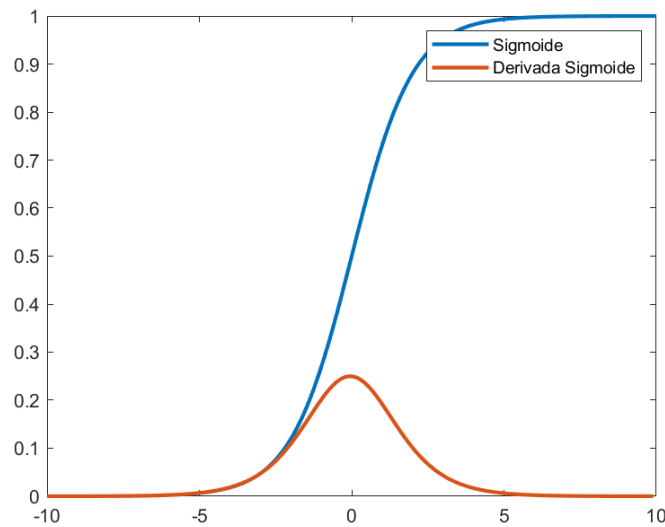
Donde:

$$Z_1 = Capa_N w_{n+1} \quad (3)$$

La función  $f(z)$  se conoce como la función de activación, siendo la función *sigmoide* una de las más usadas. Si se observa la Figura 2, el rango de la derivada de la función *sigmoide* está acotada entre 0 y 0.25. De esta manera, si se tiene una red con muchas capas, el valor del gradiente se vuelve cero o cercano a cero, ya que se multiplica varias veces por un valor pequeño (Ver fórmula 2). Debido a este problema relacionado con el *backpropagation*, las primeras capas son más difíciles de entrenar, ya que el valor del gradiente cada vez es más pequeño (Ligency, 2022). Al suceder estos problemas, en 2016 el equipo de Microsoft Research liderado por He *et al.* (2015), propuso en su artículo *Deep Residual Learning for Image Recognition* implementar las redes neuronales residuales o Resnet.

*Figura 2.*

Gráfica función sigmoide y su derivada



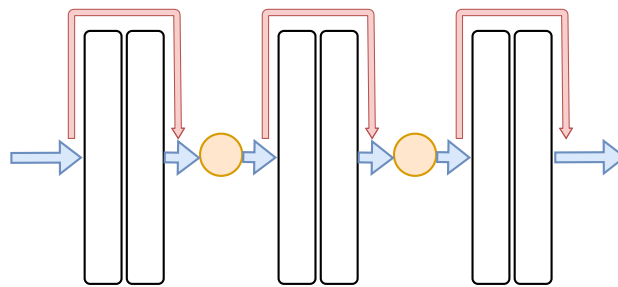
### 3.2. Definición

Las redes neuronales residuales permiten el entrenamiento de redes profundas, ya que utilizan conexiones de acceso directo de identidad o saltos entre capas, logrando así que las primeras capas del modelo obtengan información para la actualización de sus pesos. Esta información en una red neuronal no tan profunda sería dada por el descenso del gradiente, pero cuando la red es muy profunda, dicho gradiente tiende a desaparecer debido a la multiplicación repetida de un valor de gradiente infinitamente pequeño, como se observa en la fórmula (2). Por esta razón cuando la red se profundiza su rendimiento tiende a degradarse rápidamente (He *et al.*, 2015).

En la Figura 3 se muestra la arquitectura básica de una red neuronal residual, donde cada rectángulo representa las capas, las flechas azules representan el flujo desde la entrada hacia la salida y las flechas rojas muestran “atajos” entre capas con el fin de saltarse ciertos bloques.

*Figura 3.*

Diagrama básico de una red neuronal residual ResNet

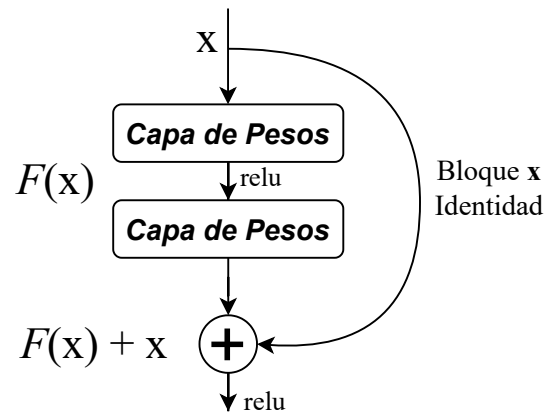


*Nota.* Elaboración propia

Finalmente, cada capa se ve influenciada por ambas direcciones de información.

Otra razón que demuestra la importancia de las redes neuronales residuales está relacionada con el aumento de capas. Es de esperarse que al aumentar el número de capas en una red, el modelo se vuelva más profundo y pueda aprender más características para su correcta clasificación, pero no sucede así. Cuando un modelo está lo suficientemente entrenado y tiene una convergencia, al agregar más capas lo que sucede es que aumentará el error a la salida. Como solución, en He *et al.* (2015) propusieron que se podían dar saltos entre capas con el fin de que al aumentar el número de capas, no aumentara el error sino disminuyera, para que la red tuviera un mejor aprendizaje. La Figura 4 muestra que una entrada llamada  $X$  se ve reflejada al final de la red como adición de  $F(x)$  ya que realiza un salto entre capas. Dicho esquema, permite que al agregar más capas, la entrada  $X$  no se vea afectada por las capas siguientes sino se pueda ver a la salida y este presente en la función de activación.

Figura 4.  
Bloque residual



Nota. Elaboración propia

#### 4. Knowledge Distillation

El *Knowledge Distillation* o destilación por conocimiento es un método que permite entrenar un modelo grande (conocido como el profesor o *Teacher*) para transferir sus conocimientos a un modelo más pequeño (conocido como el estudiante o *Student*) manteniendo su eficacia. En Buciluă *et al.* (2006) se propuso por primera vez este método y luego en el 2015 fue generalizado como se muestra en Hinton *et al.* (2015).

Es cierto que un modelo grande que tiene muchos parámetros o pesos tiene mayor capacidad de conocimiento, es posible que este modelo no use todo su potencial, lo que se traduce en una difícil implementación de dispositivos con menor capacidad de hardware. De esta forma, el método de *Knowledge Distillation* propone implementar modelos más pequeños con menos parámetros,

manteniendo la misma precisión y eficacia que el modelo más grande o *Teacher* (Keras, 2022).

Según el artículo Gou *et al.* (2021), existen tres tipos de *Knowledge Distillation* conocidos como destilación basada en respuestas, destilación basada en características y destilación basada en relaciones.

#### 4.1. Función de Pérdida

Esta función le permite a la red neuronal evaluar la desviación que existe entre la predicción realizada por ella y los valores reales que fueron utilizados durante el entrenamiento, un valor alto de desviación implica bajo rendimiento de la red y un bajo valor de desviación implica una red neuronal eficiente. La función de pérdida se muestra en la fórmula 4:

$$\mathcal{L}(x; W) = \alpha * \mathcal{H}(y, \sigma(Z_n; T = 1)) + \beta * \mathcal{H}(\sigma(Z_t; T = \tau), \sigma(Z_s, T = \tau)) \quad (4)$$

Donde  $x$  es la entrada,  $W$  los parámetros del modelo estudiante,  $y$  la etiqueta de verdad básica o *ground truth label*,  $\mathcal{H}$  es la función de pérdida de entropía cruzada,  $\sigma$  es la función softmax parametrizada por la temperatura  $T$  y los coeficientes  $\alpha$  y  $\beta$ ,  $Z_t$  y  $Z_s$  son los logits del profesor y del estudiante respectivamente.

## 4.2. Tipos de Destilación

### 4.2.1. Destilación Basada en Respuestas

Este tipo de destilación consiste en que el modelo del alumno aprende a imitar las predicciones que el modelo profesor realiza. Esto se logra usando la función de pérdida por destilación la cual consiste en calcular la diferencia entre las predicciones “blandas” hechas por el modelo del estudiante con las etiquetas “blandas” hechas por el profesor. Por lo tanto, entre más pequeña sea esta diferencia significa que el estudiante tiene mayor capacidad de hacer predicciones similares a las del maestro.

**4.2.1.1. Función de Pérdida por Destilación.** Esta función de pérdida es la encargada de capturar la diferencia que existe entre los logits de los modelos del maestro y el estudiante.

**4.2.1.2. Predicciones Blandas o Soft Predictions.** Las predicciones blandas brindan información acerca de las probabilidades sobre las clases de salida. Por ejemplo, una posible salida para este caso en el que la red tiene tres salidas sería [0.02 , 0.05 , 0.93]. A pesar de que muestra una clara salida que corresponde a la última clase, las predicciones suaves pueden mostrar información sobre que similitud tiene la entrada respecto a la primera y segunda clase.

Este tipo de predicciones o salidas entregadas en un rango de probabilidad se dan por medio de la softmax. La softmax es una función de activación que permite entregar una salida entre 0 y 1 para cada clase o salida. Esta salida es obtenida mediante la fórmula 5.

$$\sigma(Z)_i = \frac{e^{\frac{Z_i}{T}}}{\sum_j e^{\frac{Z_j}{T}}} \quad (5)$$

Donde  $Z_i$  son los *logits* y  $T$  la *temperatura*.

- **Logits:** corresponde a la salida de la última capa en la red neuronal, la cual contiene predicciones. Esta salida generalmente entrega datos sobre un modelo de clasificación sin procesar que luego son llevados a una función de normalización, por ejemplo, la función softmax (Overflow, 2017).
- **Temperatura:** el parámetro de temperatura puede dar información sobre los valores de probabilidad o las predicciones blandas de las clases o etiquetas. Por ejemplo, supongamos que una red neuronal entrega la siguiente salida con una temperatura de 1: [0.03, 0.01, 0.96]. En la lista anterior, el mayor valor de probabilidad representa que la entrada pertenece a la última clase y la red entrega poca probabilidad sobre las predicciones suaves. Esto se debe a que las primeras dos clases tienen un comportamiento muy similar, lo que hace que no sea un buen indicador para decir a que otra clase se parece. Por el contrario, si se tiene la siguiente salida con una temperatura de 3: [0.14, 0.01, 0.85], se observa que sigue mostrando que la entrada pertenece a la última clase, y a su vez muestra cierta dependencia con la primera clase.

Por lo tanto, un modelo que tenga bajas temperaturas suele tener buenos resultados en las

predicciones o etiquetas duras (*hard labels*). Mientras que un modelo con altas temperaturas se considera que tiene un buen conocimiento oscuro, es decir, puede distinguir a qué clase pertenece la entrada y además cuáles son las similitudes con el resto de clases.

En definitiva, la función *Softmax* permite mantener una probabilidad mayor para el valor de entrada más grande y “aplanar” o volver más pequeños los valores menos probables de la clase.

- Etiquetas duras o *Hard Labels*: corresponde a las salidas arrojadas por la red que muestra pertenencia a la clase en binario. Por ejemplo, una posible salida para el caso en el que la red tiene tres salidas sería [0 , 0 , 1]. A pesar de que muestra una clara salida, no muestra información sobre la similitud que encuentra la red neuronal respecto a la primera y segunda clase, ya que sus salidas son cero.

#### ***4.2.2. Destilación Basada en Características***

Las redes neuronales en sus capas intermedias logran abstraer características específicas las cuales permiten obtener información importante, esto lo logra por medio de las funciones de activación. La destilación basada en características busca que por medio del proceso de destilación el modelo de estudiante o *student* aprenda las mismas funciones de activación con las que cuenta el modelo del profesor o *teacher*.

### ***4.2.3. Destilación Basada en Relaciones***

Los anteriores tipos de destilación se basan en las características que entregan ciertas capas específicas del modelo del profesor. La destilación basada en relaciones busca la ampliación de conexiones entre las diferentes capas del modelo del profesor, estas relaciones de mapas de características son resumidas por medio de la matriz de Gram, la cual permite la descomposición de valores singulares entre capas para destilar el conocimiento. Estas relaciones se pueden resumir como la relación entre gráficos, matrices de similitud, acoplamiento de características y mapas de características.

## **4.3. Modos de Destilación**

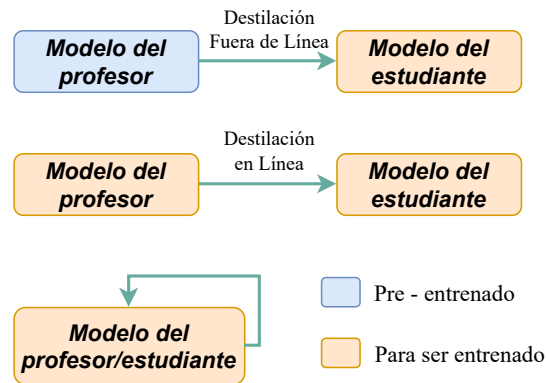
Existen principalmente tres formas en que el modelo del estudiante logra abstraer características del modelo del profesor y se conocen como: autodestilación, destilación fuera de línea y destilación en línea (Gou *et al.*, 2021). La Figura 5 muestra un esquema sobre estos tipos de destilación.

### ***4.3.1. Destilación Fuera de Línea***

Este modo de destilación o aprendizaje por parte del estudiante consiste en que el modelo del profesor es completamente entrenado previo a la destilación del estudiante, es decir, finalizado el entrenamiento del profesor, este modelo empieza su destilación o entrenamiento del estudiante. Esta técnica de destilación es la más usada en el aprendizaje profundo ya que es la más fácil de implementar.

*Figura 5.*

Esquema de los tipos de destilación

*Nota.* Elaboración propia

### 4.3.2. Destilación en Línea

Cuando se tiene un modelo del profesor que no tiene gran rendimiento ni capacidad, surgen algunos problemas al momento de la destilación. De esta manera, la destilación en línea pretende mejorar el rendimiento del modelo del estudiante, donde ambos modelos se entrenan y actualizan al tiempo.

### 4.3.3. Autodestilación

En este tipo de destilación, se utilizan las mismas redes para crear el modelo del profesor y del alumno. Es un tipo de destilación en línea donde se forma al modelo del alumno con los aprendizajes de las primeras épocas que son trasladadas a las siguientes épocas.

#### 4.4. F1 Score

El *F1 Score* es un parámetro o medida que permite conocer la exactitud de una prueba. Existen muchos parámetros, entre ellos el de precisión o *Accuracy*. Este parámetro es útil cuando la base de datos se encuentra nivelada, ya que solo tiene en cuenta los resultados que están correctamente clasificados. El problema surge cuando la base de datos usada no tiene la misma cantidad entre clases, lo que hace que el parámetro de *Accuracy* no sea fiable en sus resultados. Por lo tanto, el parámetro que entrega un buen resultado cuando la base de datos está desnivelada es el *F1 Score*. Esto se debe a que este parámetro mezcla la precisión con la recuperación o *recall* (Korstanje, 2021).

##### 4.4.1. Precisión o Accuracy

La precisión tiene en cuenta los resultados que pronostica como positivos, es decir, puede que clasifique correctamente los casos positivos en su gran mayoría, pero a su vez, detecte casos como positivos cuando en realidad no lo son. A este último caso se le conoce como falsos positivos (*False Positives*).

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

Donde TP corresponde a los *True Positive* o los positivos verdaderos y FP corresponde a los *False Positive* o falsos positivos.

Figura 6.  
Matriz de confusión.

		<b>PREDICCIÓN</b>	
		Predicción Positiva	Predicción Negativa
<b>CLASIFICACIÓN</b>	Clasificación Positiva	TP	FP
	Clasificación Negativa	FN	TN

Nota. Adaptado de Merchán Vargas *et al.* (2021)

#### 4.4.2. Recuperación o Recall

El parámetro de recuperación o *Recall*, también llamado recuerdo, puede encontrar los casos positivos y a su vez, identificar erróneamente casos negativos como casos positivos. A este último caso se le conoce como falsos negativos (*False Negatives*).

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

Donde TP corresponde a los *True Positive* o los positivos verdaderos y FN corresponde a los *False Negatives* o falsos negativos.

Para entender un poco mejor estos dos conceptos, la Figura 6 ilustra el esquema básico de una matriz de confusión donde se observa el uso de estos conceptos de manera más clara.

El objetivo del parámetro de F1-score es mezclar la precisión con la recuperación y obtener la media armónica entre ellos, es decir, la fórmula del F1 Score se muestra en la ecuación 8 ó 9.

$$F1Score = 2 * \frac{Precision * recall}{Precision + recall} \quad (8)$$

$$F1Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \quad (9)$$

## 5. Análisis Previo de Resultados

### 5.1. Salida de Dos Clases

En Merchán Vargas *et al.* (2021) se detalla el proceso para la detección de cáncer de piel usando redes neuronales profundas. En su investigación aplicaron un procedimiento de experimentación factorial para obtener una arquitectura final. Los investigadores encontraron que la red MobileNet V1 e Inception V3 obtuvieron la mejor precisión respecto a las otras redes preentrenadas disponibles en los repositorios. En las tablas 5 y 6 se muestran los resultados de cada una de las redes mencionadas anteriormente usando la base de datos empleada en Merchán Vargas *et al.* (2021) y la explicada en la sección 2.

Finalmente, la Tabla 7 muestra el número de parámetros usados por cada red y el consumo computacional de cada modelo usado por los investigadores en Merchán Vargas *et al.* (2021).

Tabla 5

Comparación de resultados de la red MobileNet V1 con diferentes bases de datos.

F1 score				
Base de datos	Red	Benigno	Maligno	Total
Usada por Merchán Vargas <i>et al.</i> (2021)	MobileNet V1	0.9109	0.9123	0.9104
Nuestra (Sección 2)	MobileNet V1	0.9603	0.7807	0.9332

Tabla 6

Comparación de resultados de la red Inception V3 con diferentes bases de datos.

F1 score				
Base de datos	Red	Benigno	Maligno	Total
Usada Merchán Vargas <i>et al.</i> (2021)	Inception V3	0.8308	0.8276	0.8301
Nuestra (Sección 2)	Inception V3	0.9284	0.5974	0.8811

## 5.2. Salida de tres clases

Los investigadores en Merchán Vargas *et al.* (2021) realizan el proceso similar de ajuste fino que se hizo en la red para detección de lesiones malignas y benignas, pero usando las 3 clases de cáncer con más incidencia en el país. En la tabla 8 se puede apreciar los valores de F1-score obtenidos por los autores. Los resultados replicados por los autores de este trabajo coinciden con los contenidos en Merchán Vargas *et al.* (2021).

## 6. Creación del Modelo Estudiante para el Knowledge Distillation

Para aplicar la técnica de Knowledge Distillation es necesario definir el modelo *Teacher* y el modelo *Student*. Para el modelo *Teacher* se tomó la red que demostró un mejor F1-score, en este caso, la red MobileNet V1. Para la creación del modelo *Student* se utilizó la estrategia de conformación de capas presentada por Fajardo *et al.* (sf). La estructura del modelo *Student* tienen

Tabla 7

*Comparación de parámetros y consumo de memoria entre la red MobileNet V1 y la red Inception V3.*

Número de Parámetros y Consumo de memoria		
Red	Parám.	Memoria (MB)
MobileNet V1	3.23M	17
Inception V3	21.91M	95.5

Tabla 8

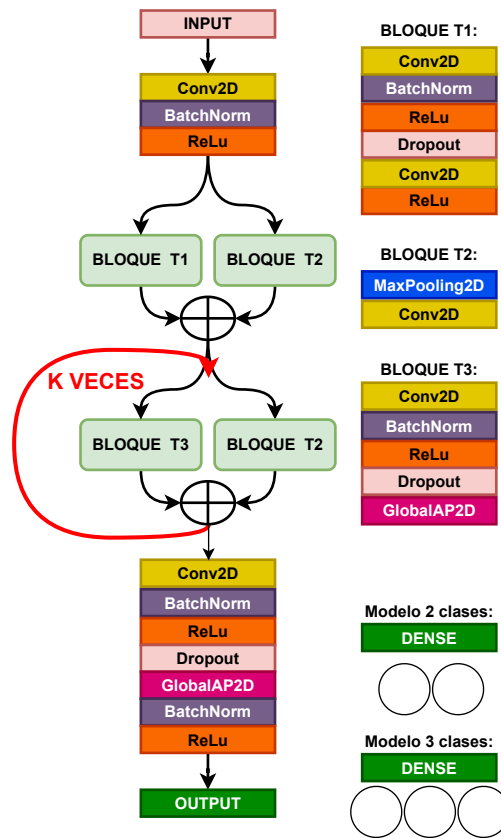
*Puntaje F1 – score para una clasificación de tres clases*

F1 score				
Red	Carcinoma	Melanoma	Queratosis	Total
MobileNet V1	0.85	0.84	0.68	0.7815
Inception V3	0.77	0.75	0.51	0.6617

un parámetro llamado  $K$  que indica la profundidad de la red y la cantidad de bloques residuales. La figura 7 presenta la arquitectura básica del modelo desarrollado. El modelo tiene dos bloques que permiten realizar las conexiones residuales que son T1 y T2. T1 es el bloque de convolución y extracción y T2 es la conexión de atajo. La red entera inicia con convoluciones de 32 filtros en cada convolución y se aumentan gradualmente. Cabe resaltar que para los bloques residuales T1 se varió el número de filtros dando un valor inicial de 32 filtros para valores de  $K < 4$  y duplicándolos cada 4 bloques residuales adicionales. Las convoluciones de T2 son convoluciones  $1 \times 1$  con stride 1. La misión de este bloque es adaptar la cantidad de filtros de la capa anterior con la siguiente. El código se puede apreciar mejor en el repositorio (González y Rivera, 2022).

Para la búsqueda del grupo de estudiantes opcionados, se varió  $K$  de 1 hasta 8. Los criterios establecidos para la búsqueda de los estudiantes opcionados fueron los siguientes:

Figura 7.

Arquitectura de la red neuronal propuesta para el modelo *Student*

Nota. Elaboración propia

- Un F1-score no inferior al 10% (o cercano) con respecto al modelo *Teacher*. El F1-score del modelo del *Teacher* es de 0.9332 (Ver Tabla 5).
- Los cinco modelos que tengan menos parámetros. El modelo *Teacher* tiene 3'230.914 parámetros. El nivel de compresión del modelo *Student* se calcula con los parámetros obtenidos por la red *Teacher* sobre la cantidad de parámetros de la red *Student*.

La Tabla 9 muestra resultados de F1-score y su nivel de compresión al variar  $K$ . Es de

Tabla 9

*Resultados obtenidos para cada valor de K*

<b>K</b>	<b>Número de Parámetros</b>	<b>F1 Score</b>	<b>Compresión</b>	<b>Porcentaje de Compresión</b>
1	37730	0.8227	85.6324	98.83 %
2	71266	0.8778	45.3359	97.79 %
3	104802	0.8905	30.8287	96.75 %
4	138338	0.9000	23.3552	95.71 %
5	627682	0.8944	5.1473	80.57 %
6	760290	0.8909	4.2495	76.46 %
7	892898	0.8729	3.6184	72.36 %
8	1025506	0.8784	3.1505	68.25 %

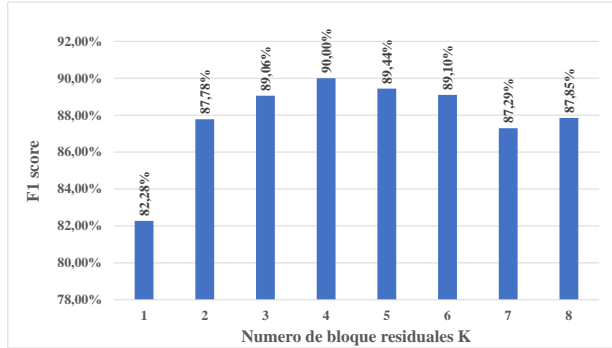
esperarse que al aumentar el valor de  $K$  aumente el número de parámetros, ya que el modelo se vuelve más profundo, es decir, aumenta el número de capas (un comportamiento exponencial). Además, la tabla 9 muestra que los valores de  $K$  que cumplen con las condiciones anteriormente nombradas son:  $K=1$ ,  $K=2$ ,  $K=3$ ,  $K=4$  y  $K=5$ .

La figura 8 muestra la variación del rendimiento en la clasificación y el número de parámetros según los valores de  $K$ . En la Figura 8a se puede observar como para  $K>5$  hay una leve afectación del rendimiento. En la Figura 8b también se aprecia como se incrementa considerablemente el número de parámetros para  $K>5$ .

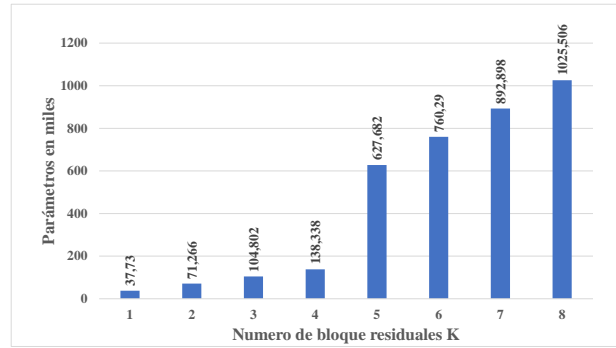
Figura 8.

F1-score y parámetros vs. bloques residuales

(a) F1-score



(b) Número de parámetros en miles vs. número de bloques residuales



## 7. Búsqueda de Hiperparámetros para la Aplicación de Knowledge Distillation

### 7.1. Salida de Dos Clases

Para la aplicación de esta técnica es necesario definir los hiperparámetros de alpha y temperatura para lograr una destilación exitosa, es decir, que el estudiante logre capturar los conocimientos del profesor. Se realizó una búsqueda en malla variando los valores de  $\alpha$  y Temperatura de la siguiente manera:

- $\alpha = 0.1, 0.2, 0.3, 0.4$
- $T = 1, 3, 5, 10, 15$

La Tabla 10 reúne los 10 mejores resultados según su puntaje de F1-score para un estu-

Tabla 10

*Mejores modelos obtenidos a partir de los experimentos.*

<b>K</b>	<b>Alpha</b>	<b>Temperatura</b>	<b>F1-score Destilado</b>	<b>F1-score Sin Destilar</b>
1	0.4	3	0.9464	0.8227
2	0.1	15	0.9308	0.8778
3	0.2	1	0.9557	0.8905
	0.3	1	0.9210	
		3	0.9466	
		5	0.9119	
	0.4	1	0.9496	
4	0.2	1	0.9249	0.9000
		3	0.9418	
	0.4	5	0.9388	

Nota. La tabla muestra los 10 mejores modelos a partir de una búsqueda en malla

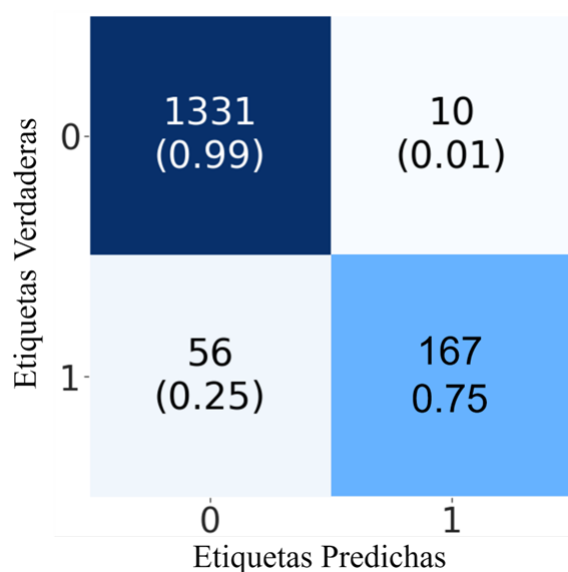
dante destilado, además se compara este resultado con el puntaje F1-score de un estudiante sin destilación. Nótese que hay un incremento en el valor de F1-score, del modelo destilado con respecto a su contraparte. Esto indica evidencia que el modelo *Student* ha logrado extraer parte del conocimiento oscuro contenido por el *Teacher*.

La Figura 9 muestra la matriz de confusión correspondiente a los resultados del estudiante con una salida de dos clases. Nótese que la red tiene una predisposición mayor a la clasificación de benigno. Esto puede deberse al desbalance de la base de datos.

La Tabla 11, muestra los resultados del modelo *Student* comparado con el modelo *Teacher* en puntuación F1-score. Nótese que la técnica logró incrementar en 2 puntos porcentuales el F1-score, y a su vez, con menor cantidad de parámetros. También cabe resaltar que mejoró el desem-

*Figura 9.*

Matriz de confusión para una salida de dos clases



*Nota:Elaboración propia*

peño para la clasificación de imágenes malignas.

La tabla 12 muestra una comparación del número de parámetros y los requerimientos de memoria en una salida de dos clases por cada caso. Nótese que los niveles de compresión son diferentes en términos de parámetros y memoria.

## 7.2. Salida de Tres Clases

Para el modelo de tres clases se sigue una metodología similar a la contemplada para las dos clases. El cambio principal está relacionado con la base de datos, ya que en este caso, se trabajó con imágenes pertenecientes a los tres tipos de cáncer más comunes en Colombia.

Así como se necesitó un modelo de *Teacher* y *Student* para la aplicación de la técnica del

Tabla 11

Comparación de puntuación F1-score entre el modelo Teacher y Student para una salida de dos clases

Modelo	Red	Benigno	Maligno	F1-score
Modelo Teacher	MobileNet V1	0.9603	0.7807	0.9332
Modelo Student	Explicada en la sección 6	0.9758	0.835	0.9557

Tabla 12

Comparación de número de parámetros y memoria entre el modelo Teacher y Student para una salida de dos clases y su respectivo nivel de compresión.

Modelo	Red	Parám.	Memoria (MB)
Modelo Teacher	MobileNet V1	3230914	17
Modelo Student	Explicada en la sección	104802	0.4003
<b>Nivel de Compresión</b>		<b>30.82X</b>	<b>42.46X</b>

*Knowledge Distillation* en una salida de dos clases, para una salida de tres clases se necesita lo mismo. El modelo del *Teacher* será el mejor modelo implementado por Merchán Vargas *et al.* (2021) que se muestra en la Tabla 8. En este caso se trata de la red MobileNet V1 para tres clases. Además, el modelo *Student* tendrá la misma arquitectura que el modelo implementado en una salida de dos clases, la diferencia radica que la capa de salida ahora tiene 3 neuronas (Ver sección 6).

Finalmente se realizan las pruebas usando los mismos hiperparámetros con los mejores cinco valores de  $K$ . La Tabla 13 reúne los 10 mejores resultados según su puntaje de F1-score. El valor de F1-score para una salida de tres clases con la configuración de  $K=4$ ,  $\alpha=0.3$  y Temperatu-

Figura 10.

Matriz de confusión para una salida de tres clases

0	98 (0.72)	17 (0.12)	21 (0.15)
1	7 (0.03)	232 (0.97)	0 (0.00)
2	0 (0.00)	1 (0.01)	98 (0.99)
	0	1	2
	Etiquetas predichas		

Nota. Elaboración propia

ra=5, estos valores representa el mejor resultado respecto a las demás configuraciones. Nótese que varió el valor de K,  $\alpha$  y T. En este caso, al tener menor cantidad de datos se evita que el modelo se sobre-entrene. Es por esto que el valor de temperatura con mejor resultado es superior al obtenido para 2 clases.

La Figura 10 muestra la matriz de confusión correspondiente a los resultados del estudiante con una salida de tres clases. Se puede observar que a pesar de que la base de datos esta desbalanceada hay una buena tendencia de entrenamiento para las 3 clases. También se evidencia como la clase carcinoma parece ser la más confusa para la red.

La Tabla 15 muestra los resultados del modelo *Student* comparado con el modelo *Teacher*.

Tabla 13

*Mejores modelos del estudiante destilado para una salida de tres clases*

<b>K</b>	<b>Alpha</b>	<b>Temperatura</b>	<b>F1-score Destilado</b>
1	0.1	15	0.8689
2	0.3	3	0.8712
3	0.1	15	0.8689
	0.2	15	0.8736
4	0.1	10	0.8993
	0.3	5	0.8995
5	0.2	10	0.8755
		3	0.8706
	0.3	15	0.8870
		0.4	1

*Nota. La tabla evidencia los mejores 10 modelos a partir de un la búsqueda en malla*

Se puede evidenciar en este modelo un incremento general del F1-score respecto al modelo destilado con respecto a MobilenetV1. Nótese también que el valor de F1-score para la clase carcinoma se ha reducido en el modelo destilado.

La tabla 14 muestra una comparación de el número de parámetros y los requerimientos de memoria en una salida de tres clases por cada caso. Nótese que los niveles de compresión son diferentes en términos de parámetros y memoria.

Tabla 14

*Comparación de número de parámetros y memoria entre el modelo Teacher y Student para una salida de tres clases y su respectivo nivel de compresión.*

<b>Modelo</b>	<b>Red</b>	<b>Parám.</b>	<b>Memoria (MB)</b>
<i>Modelo Teacher</i>	MobileNet V1	3230914	17
<i>Modelo Student</i>	Explicada en la sección	138338	0.541
<b>Nivel de Compresión</b>		<b>23.35X</b>	<b>31.42X</b>

Tabla 15

*Comparación de resultados entre el modelo Teacher y Student para una salida de tres clases*

<b>Modelo</b>	<b>Carcinoma</b>	<b>Melanoma</b>	<b>Keratosis</b>	<b>F1-score</b>
<i>Modelo Teacher</i>	0.85	0.84	0.68	0.7815
<i>Modelo Student</i>	0.8132	0.9488	0.8990	0.8995

## 8. Implementación del Modelo en un Sistema Android

### 8.1. Desarrollo y Montaje del Modelo

Para implementar los modelos de mejor rendimiento experimentados, tanto para dos y tres clases en un sistema Android, es necesario un archivo adaptado para tal fin. TensorFlow Lite es una biblioteca móvil que permite implementar modelos en dispositivos móviles, microcontroladores, microcomputadoras, etc (Lite, 2022). Para esto se debe crear el modelo definitivo, con su entrenamiento y validación respectiva, lista para implementar. Finalmente, se agregan las líneas de código que se muestran en la Figura 11 para generar el archivo con extensión “.tflite” a partir del modelo.

*Figura 11.*

Código usado para generar el archivo .tflite

```
1 student_new = distiller.student
2 print(student_new)
3 converter = tf.lite.TFLiteConverter.
  from_keras_model(student_new)
4 tflite_student_new = converter.convert()
5 open("/content/drive/MyDrive/"+str(k)+
  '.tflite', "wb").write(
  tflite_student_new)
```

El valor de F1 Score para una salida de tres clases para  $K=4$ ,  $\alpha=0.3$  Y Temperatura=5 se considera el mejor resultado para la red de *Student* aplicando la técnica de *Knowledge Distillation* con el entrenamiento y conocimiento proveniente del modelo del *teacher*. La Tabla 15 muestra los resultados del modelo *student* comparado con el modelo *teacher*.

## 8.2. Implementación del Modelo en Sistemas Android

### 8.2.1. Desarrollo y Montaje del Modelo

Para implementar el mejor modelo desarrollado para dos y tres clases en un sistema Android, se debe tener un archivo con extensión “.tflite”. TensorFlow Lite es una biblioteca móvil que permite implementar modelos en dispositivos móviles, microcontroladores, etc (Lite, 2022). Para esto se debe crear el modelo definitivo, con su entrenamiento y validación respectivo listo para implementar. Finalmente, se agregan las líneas de código que se muestran en la Figura 11 para generar el archivo “.tflite” a partir del modelo.

Una vez se tiene el archivo generado, se realiza la implementación en Android Studio. Android Studio es un entorno de desarrollo de aplicaciones integrado oficial de Android (Studio,

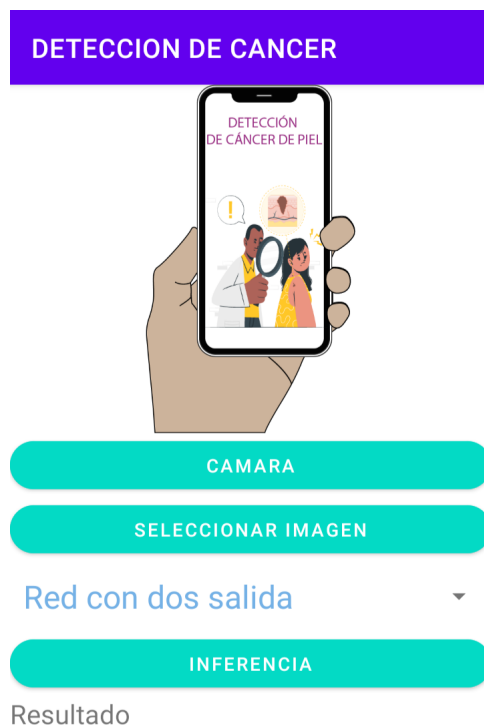
2022). Para la instalación de la App desarrollada se trabajó con la versión de Android Studio Chipmunk | 2021.2.1 Patch 1 y con lenguaje de programación Java.

### **8.3. Diseño y Funcionalidad de la Aplicación**

La aplicación cuenta con una interfaz sencilla e intuitiva para el usuario que cuenta con las siguientes características, ver Figura 12. En la Figura 12 lo primero que se observa es el título alusivo al tema de interés, que en este caso es “DETECCIÓN DE CÁNCER”. Luego viene una imagen relacionada a una aplicación instalada en un teléfono. Adicionalmente, se observan tres botones: el botón “CÁMARA”, el botón “SELECCIONAR IMAGEN” y el botón de “INFERENCIA”. El primer botón tiene como objetivo abrir la cámara y tomar una foto para cargarla en la aplicación. La finalidad del segundo botón es abrir la galería del teléfono y cargar una imagen que este en el celular. Debajo del botón de “SELECCIONAR IMAGEN”, se selecciona la salida de la red de dos o tres clases, donde la salida de la red de dos clases clasifica entre Benigno V.S Maligno y la salida de la red de tres clasifica entre Carcinoma, Melanoma o Queratosis. Finalmente, se oprime el tercer y último botón que permite hacer la inferencia para obtener los resultados de clasificación para dos clases o tres clases. Estos resultados se muestran de forma porcentual.

Las imágenes tomadas para hacer el ícono y el fondo de la aplicación fueron tomadas de Canva (Perkins, sf) y de Freepik respectivamente. Esta última con autoría de storyset (Storyset, sf) y raw pixel (RawPixel, sf) respectivamente.

Figura 12. Interfaz de usuario de la aplicación Android



### 8.3.1. Procesamiento de las Imágenes

Para la creación del modelo, se tiene que hacer un procesamiento previo a las imágenes que ingresan a la red. Este procesamiento viene dado por un redimensionamiento a cada imagen con el fin de que cada Figura que ingresa a la red tenga un tamaño de 224x224px. Este mismo procesamiento se debe hacer en la aplicación de Android ya que cada fotografía tomada tiene una resolución particular dada por las especificaciones del celular y debe ser convertida a 224x224px para que pueda ingresar a la red. Esto se logró a través de un modulo de interpolación integrado a las librerías de Android.

Tabla 16

*Comparación de resultados entre la red implementada en Python y la aplicación en Android para una salida de dos clases.*

F1-score			
Resultados	Benigno	Maligno	Total
Resultados en la red	0.9758	0.835	0.9557
Resultados en la App Android	0.9497	0.560	0.9497

#### 8.4. Evaluación de Resultados

Para evaluar el funcionamiento de la herramienta en Android, se tomaron todas las imágenes de test de tipo Benigno v.s Maligno y de tipo Carcinoma, Melanoma, Queratosis. Luego se cargó cada imagen de manera individual a la aplicación de Android con el fin de registrar cada resultado y sacar la matriz de confusión correspondiente. Finalmente, estos resultados son comparados con los obtenidos en Python tomando las mismas imágenes de test de tipo Benigno v.s Maligno y de tipo Carcinoma, Melanoma, Queratosis.

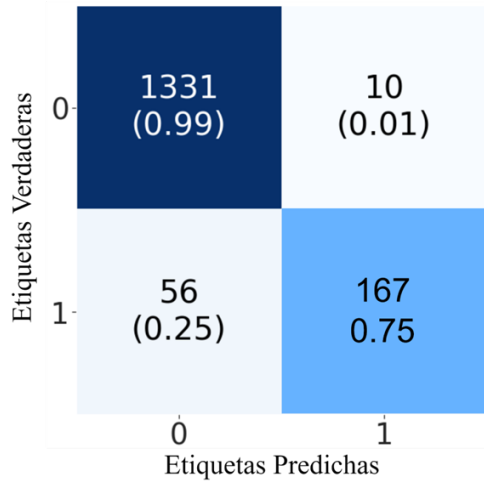
##### 8.4.1. Salida de Dos Clases

La Figura 13 muestra las matrices de confusión para una salida de dos clases, siendo la Figura 13a la matriz de confusión obtenida en el entrenamiento en Python y la Figura 13b la matriz de confusión obtenida con los resultados de las pruebas en la aplicación Android. Finalmente la tabla 16 muestra la puntuación F1-score para cada clase según los resultados de la red y la App Android.

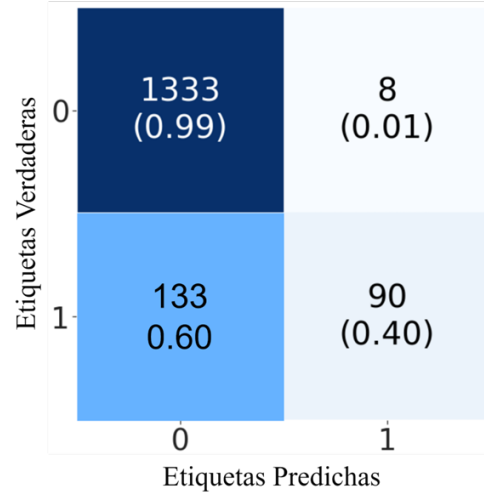
Figura 13.

Matriz de confusión de dos clases

(a) Matriz en Python



(b) Matriz en App Android



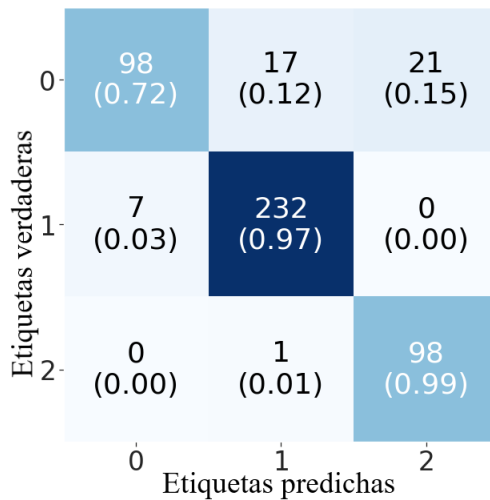
### 8.5. Salida de Tres Clases

la Figura 14 muestra las matrices de confusión para una salida de tres clases, siendo la Figura 14a la matriz de confusión obtenida en el entrenamiento en Python y la Figura 14b la

Figura 14.

Matriz de confusión de tres clases

(a) Matriz en Python



(b) Matriz en App Android

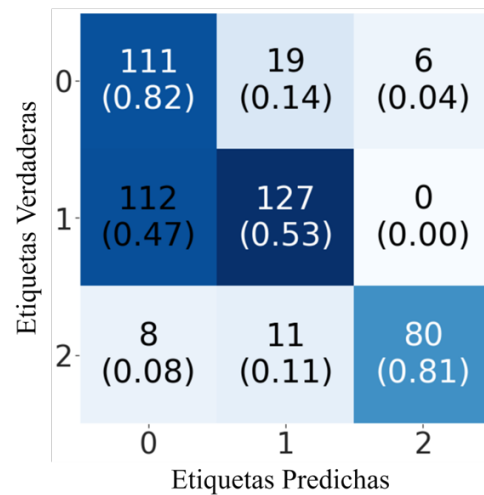


Tabla 17

*Comparación de resultados entre la red implementada en Python y la aplicación en Android para una salida de tres clases.*

F1-score				
Red	Carcinoma	Melanoma	Queratosis	Total
Resultados en la red	0.8132	0.9488	0.899	0.8995
Resultados en la App Android	0.6049	0.6414	0.8648	0.7290

matriz de confusión obtenida con los resultados de las pruebas en la aplicación Android.

Finalmente la tabla 17 muestra la puntuación F1-score para cada clase según los resultados de la red y la App Android.

## 9. Conclusiones

A través de la investigación de la teoría de las redes residuales y del Knowledge Distillation como estrategia de compresión, y su posterior aplicación sobre un modelo base, se logró obtener dos modelos para la clasificación de dos casos de cáncer de piel. El primero de imágenes malignas y benignas y el segundo modelo para clasificar las tres clases de cáncer con más incidencia en Colombia (carcinoma basocelular, queratosis, melanoma). La inferencia para el modelo de dos clases fue un F1-score de 95.57 %, precisión de 95.96 %, recall de 99.25 %, cantidad de parámetros de 104.802 (compresión de un factor de 30.82x) y un espacio en memoria de 400.3kB (tasa de compresión de un factor de 40.46x) con respecto al modelo base.

Para el modelo de tres clases se obtiene un F1-score de 89.95 %, precisión de 89.49 %,

recall de 89.37 %, cantidad de parámetros de 138.338 (compresión de un factor de 23.35x) y un espacio en memoria de 541 KB (tasa de compresión de un factor de 31.42x) con respecto al modelo base.

Adicionalmente se logró una implementación satisfactoria en Android. Usando Android estudio se creó una aplicación que posee los dos modelos obtenidos. La inferencia para el modelo de dos clases fue un F1-score de 94.97 %, precisión de 90.92 %, recall de 99.40 % y para el modelo de tres clases un F1-score de 72.90 %, precisión de 73.98 % y recall de 71.85 %.

Se obtiene una fuerte evidencia con respecto al uso de las redes neuronales residuales en conjunto con la técnica del knowledge distillation y como estas permiten comprimir los modelos sin sacrificar considerablemente la precisión.

La calidad de los sistemas Android puede afectar los valores de la inferencia, ya que de un dispositivo a otro puede variar el resultado cuando la red le cuesta clasificar y muestra valores muy similares entre clases. También se recomienda ver la afectabilidad que puede tener la calidad de la cámara y de la captura en el desempeño del sistema.

## **10. Trabajo Futuro**

Teniendo en cuenta que el cáncer de piel en Colombia viene en continuo crecimiento (Velasquez, 2020), es importante tener una base de datos con más imágenes e información sobre cada una de ellas, que le permita a la red un mayor aprendizaje para que los resultados puedan ser mejores. Se propone para trabajos futuros buscar el apoyo de instituciones médicas y organizaciones

que luchen contra el cáncer que puedan aportar a la construcción de bases de datos más amplias y confiables.

Para trabajos futuros también se recomienda probar con otras técnicas de compresión como lo son el pruning y la descomposición tensorial para averiguar si es posible reducir aún más los modelos.

Para investigaciones futuras se recomienda encontrar resultados con el parámetro de AUC como medida de desempeño, debido a su interpretabilidad estadística y epidemiológica. Posterior a ello, se puede comparar resultados con los obtenidos usando el parámetro de F1-Score.

Además, se puede ampliar el alcance de la aplicación en un sistema iOS para que más usuarios puedan tener acceso a este beneficio para tomar acciones preventivas y disminuir el crecimiento de pacientes con cáncer de piel.

De igual manera se recomienda considerar los diferentes efectos que pueden surgir alrededor de la calidad de la cámara y la captura del dispositivo móvil y como repercute esto en el desempeño del sistema.

También se recomienda que para el uso del botón cámara se pueda tener acceso a bases de datos con imágenes cancerígenas de dos y tres clases tomadas con la cámara de un celular con el fin de que la red pueda ser entrenada con estas imágenes y se pueda habilitar este botón en la aplicación.

### Referencias Bibliográficas

- Alquran, H., Qasmieh, I. A., Alqudah, A. M., Alhammouri, S., Alawneh, E., Abughazaleh, A., & Hasayen, F. (2017). The melanoma skin cancer detection and classification using support vector machine. En *2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pp. 1–5. IEEE. doi: 10.1109/AEECT.2017.8257738.
- Barrera, J. A. (2018). Redes neuronales. [http://www.cucei.udg.mx/sites/default/files/pdf/toral\\_barrera\\_jamie\\_areli.pdf](http://www.cucei.udg.mx/sites/default/files/pdf/toral_barrera_jamie_areli.pdf).
- Bogotá, A. M. (2010). Ley 1384 de 2010 nivel nacional. <https://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=39368>.
- Brinker, T. J., Hekler, A., Utikal, J. S., Grabe, N., Schadendorf, D., Klode, J., Berking, C., Steeb, T., Enk, A. H., & Von Kalle, C. (2018). Skin cancer classification using convolutional neural networks: systematic review. *Journal of medical Internet research*, 20(10):e11936. doi: 10.2196/11936.
- Buciluă, C., Caruana, R., & Niculescu-Mizil, A. (2006). Model compression. En *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541. <https://doi.org/10.1145/1150402.1150464>.
- Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S.

- (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118. <https://doi.org/10.1038/nature21056>.
- Fajardo, C. A., Parra, A. S., Castellanos Parada, T. V., & Roy, K. (sf). Towards atrial fibrillation detection in wearable devices using deep learning.
- González, D. & Rivera, S. (2022). Skincancerproject/skincancer. <https://github.com/SkinCancerProject/SkinCancer>.
- Google (2021). Using ai to help find answers to common skin conditions. <https://blog.google/technology/health/ai-dermatology-preview-io-2021/>.
- Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819. <https://doi.org/10.1007/s11263-021-01453-z>.
- Haglin, J. M., Jimenez, G., & Eltorai, A. E. (2019). Artificial neural networks in medicine. *Health and Technology*, 9(1):1–6. <https://doi.org/10.1007/s12553-018-0244-4>.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385. <http://arxiv.org/abs/1512.03385>.
- Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. <https://arxiv.org/abs/1503.02531>.
- Hoshyar, A. N., Al-Jumaily, A., & Sulaiman, R. (2011). Review on automatic early skin cancer

detection. En *2011 International Conference on Computer Science and Service System (CSSS)*, pp. 4036–4039. IEEE. doi: 10.1109/CSSS.2011.5974581.

Instituto de Hidrología, M. y. E. A. (2015). Escenarios de cambio climático. <http://www.ideam.gov.co/web/tiempo-y-clima/escenarios-cambio-climatico>.

Katanskiy, A. (2022). Skin cancer isic. <https://www.isic-archive.com/#!/topWithHeader/wideContentTop/main>.

Keras (2022). Knowledge distillation. <https://keras.io/examples/vision/knowledgedistillation/>.

Korstanje, J. (2021). The f1 score. <https://towardsdatascience.com/the-f1-score-bec2bbc38aa6>.

Ligency, T. (2022). Deep learning de la a - z: Redes neuronales en python desde cero. <https://www.udemy.com/course/deep-learning-a-z/>.

Lite, T. (2022). Deploy machine learning models on mobile and edge devices.

Mayo, C. (2020). Cáncer de piel. <https://www.mayoclinic.org/es-es/diseases-conditions/skin-cancer/symptoms-causes/syc-20377605>.

Merchán Vargas, D. P., Navarro Báez, H., Barrero Pérez, J. G., & Castillo Bohórquez, J. A. (2021). Design of a tool for the classification of skin cancer images using deep neural networks (dnn). *Ciencia y Tecnología*, pp. 65–80. <https://doi.org/10.18682/cyt.vi21.4612>.

OMS (2022). Cáncer. <https://www.who.int/es/news-room/fact-sheets/detail/cancer>.

Overflow, S. (2017). What is the meaning of the word logits in tensorflow? [duplicate].

Perkins, M. (sf). Canva. [https://www.canva.com/es\\_us/](https://www.canva.com/es_us/).

Ramírez, A. F., Guillermo Chalela, J., & Ramírez, J. (2012). ¿cuántos dermatólogos hay en Colombia? análisis de los datos de la asociación colombiana de dermatología y cirugía dermatológica. <https://revistasocolderma.org/sites/default/files/cuantosdermatologoshayencolombia.pdf>.

RawPixel (sf). Maqueta de pantalla de teléfono inteligente, ilustración de vector de dispositivo digital. [https://www.freepik.es/vector-gratis/maqueta-pantalla-telefono-inteligente-ilustracion-vector-dispositivo-digital\\_19085884.htm#query=celular%20en%20la%20mano&position=31&from\\_view=search](https://www.freepik.es/vector-gratis/maqueta-pantalla-telefono-inteligente-ilustracion-vector-dispositivo-digital_19085884.htm#query=celular%20en%20la%20mano&from_view=search).

Social, M. S. P. (2015). Manual para la prevención del cáncer de piel no melanoma dirigido al entorno educativo. <https://www.minsalud.gov.co/sites/rid/Lists/BibliotecaDigital/RIDE/VS/PP/ENT/manual-prevencion-cancer-piel-entorno-educativo.pdf>.

Social, M. S. P. (2020a). Datos cáncer. <https://www.minsalud.gov.co/Paginas/Vicesalud-destacaciones-de-Colombia-frente-al-cancer-de-piel.aspx>.

Social, M. S. P. (2020b). Vicesalud destacó acciones de Colombia frente al cáncer de piel. <https://www.minsalud.gov.co/Paginas/Vicesalud-destacaciones-de-Colombia-frente-al-cancer-de-piel.aspx>.

Storyset (sf). Ilustración del concepto de melanoma. <https://www.freepik.es>.

Studio, A. (2022). Android for developers. <https://developer.android.com/>.

Tarrés Benet, L. (2019). Clasificación de lesiones en la piel con un ensemble de redes neuronales residuales. B.S. thesis, Universitat Politècnica de Catalunya. <http://hdl.handle.net/2117/165984>.

Velasquez, P. (2020). <https://consultorsalud.com/cancer-de-piel-como-esta-colombia-en-esta-materia/>. <https://consultorsalud.com/cancer-de-piel-como-esta-colombia-en-esta-materia/>.