Diseño de una Herramienta para la Detección de Imágenes de Cáncer de Piel Usando Redes Neuronales Profundas (DNN).

Andrew Dalai Rueda Rivera

Director

Jeyson Arley Castillo Bohorquez

Magister en Ingeniería Electrónica

Codirector

Jaime Guillermo Barrero Pérez

Doctor en Ingeniería Electrónica

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Bucaramanga

2023

Dedicatoria

A Cecilia Rivera y German Rueda por los cuales daría mi vida completa.

Andrew Dalai Rueda Rivera

Agradecimientos

En primer lugar, como todo en mi vida, agradezco a mis padres Cecilia Rivera y German Rueda, por ser el motor que me impulsa a seguir adelante cada día, quienes desde niño me han motivado e impulsado a conseguir mis sueños. Mil gracias por su amor y compresión a lo largo de estos años, por nunca defraudar y porque gracias a ustedes no hubiera llegado nunca a donde estoy ahora. Seguirán siendo el motivo de mis sonrisas y que se llenen de orgullo será mi gasolina para los años venideros. Gracias a mi director de proyecto Jeyson Castillo y al subdirector Jaime Barrero por la paciencia presentada con el desarrollo del proyecto y por los consejos que construyen mejores profesionales cada día. Gracias a mis compañeros de universidad y eternos amigos: Silvia, Karen, Juan, Jose, Tatiana, Geral, Muñoz, Daniela, Vivian que compartieron esta inolvidable experiencia a mi lado y solo queda mi enorme cariño hacia ustedes. Gaby gracias por haberme brindado tu apoyo cuando más lo necesitaba, tu amistad ilumina mis días haciéndolos menos pesados. Laura, mi otra mitad, te admiro tanto por la gran mujer e ingeniera en la que te convertiste, como te dije alguna vez, estás destinada para grandes cosas. Laura M Diaz, gracias por ser maestra y amiga, tú sabes el momento exacto que te ganaste un espacio en mi corazón. Ramón gracias por tu colaboración y amistad. Por último, pero no menos importante, gracias, Andrés Fernando, por ser mi compañero de vida, mi hombro para llorar, mi distracción, mi confidente, fuiste, eres y serás parte de mi familia para siempre, tu amistad vale más que cualquier otra cosa.

Tabla de Contenido

Lista de Figuras **7**

Lista de Tablas **8**

Introducción **12**

1 Objetivos **14**

1.1 Objetivo General 14

1.2 Objetivos Específicos 14

2 Marco De Referencia **15**

2.1 Contexto Histórico 15

2.1.1 Cáncer de Piel 15

2.1.1.1 Melanoma(mel) 15

2.1.1.2 Nevo Melanocitico(nv) 16

2.1.1.3 Carcinoma de Células Basales(bcc) 16

2.1.1.4 Queratosis Benigna(bkl) 16

2.2 Inteligencia Artificial 17

2.2.1 Procesamiento de Imágenes 17

2.2.1.1 Redes Neuronales Profundas DNN 17

2.3 Entorno de Trabajo 18

2.3.1 Colab 18

2.3.2 Thonny 18

3 Selección de una Base de Datos Adecuada para el Entrenamiento y Validación de la DNN Final 19

3.1 The HAM10000 19

3.2 Skin Cancer ISIC 20

3.3 Entrenamiento, Validación y Test 20

4 Selección de una Topología Base para Entrenar la Red Neuronal y Validar el Entrenamiento Realizado 22

4.1 Implementación de Modelo 23

4.2 Validación Cruzada de K Iteraciones 26

4.3 Análisis de Resultados 26

4.3.1 Clasificación en siete Categorías 27

4.3.2 Clasificación en cuatro Categorías 28

4.3.3 Aplicación de la validación cruzada 29

4.3.3.1 Clasificación en siete categorías 29

4.3.3.2 Clasificación en cuatro categorías 31

4.3.4 Resultados de la Mejor Topología Base 32

5 Aplicaciones de Técnicas de Aprendizaje Contrastivo para Mejorar la Precisión del Modelo Base **36**

5.1 Análisis de Resultados Aplicando Aprendizaje Constrastivo 38

6 Implementación de la Red Neuronal Final en un Sistema Embebido Prototipo **41**

6.1 Interfaz del modelo clasificador en la Raspberry PI 42

6.2 Análisis de Resultados 44

7 Conclusiones **47**

Referencias Bibliográficas **50**

Lista de Figuras

Figura 1 Modelos disponibles en Keras 22

Figura 2 Resumen de capas del segundo modelo en Colab 24

Figura 3 Comparativa de rendimiento vs costo de entrenamiento de optimizadores (Kingma & Ba, 2014) 25

Figura 4 Evaluación del modelo con los datos del test. 32

Figura 5 Imágenes del test clasificadas con la topología base seleccionada. 33

Figura 6 Matriz de confusión y métricas de eficiencia del test. 34

Figura 7 Evaluación de la técnica con los datos del test. 38

Figura 8 Imágenes del test clasificadas con aprendizaje contrastivo. 39

Figura 9 Matriz de confusión y métricas de eficiencia del test. 40

Figura 10 Interfaz del sistema embebido. 43

Figura 11 Prueba del prototipo clasificando una imagen del test, ver Vídeo 44

Figura 12 Matriz de confusión de la prueba del sistema prototipo. 45

Figura 13 Métricas de eficiencia de la prueba del sistema prototipo. 45

Lista de Tablas

Tabla 1 Contenido de las Bases de datos. 21

Tabla 2 Contenido de la base de datos modificada a cuatro categorías. 21

Tabla 3 Resultados del F1-Score en el test en diferentes iteraciones para los tres modelos de prueba. 27

Tabla 4 Resultados del F1-Score y las pérdidas con validación normal. 28

Tabla 5 Resultados del F1-Score y las pérdidas con validación normal. 28

Tabla 6 Resultados del F1-Score y las pérdidas con validación normal. 29

Tabla 7 Resultados del F1-Score y las pérdidas con validación normal. 29

Tabla 8 Resultados del F1-Score con validación cruzada para distintos valores del dropout. 30

Tabla 9 Resultados del F1-Score y la pérdida con validación cruzada en k iteraciones. 31

Tabla 10 Resultados del F1-Score con validación cruzada para distintos valores del dropout. 31

Tabla 11 Resultados del F1-Score y la pérdida con validación cruzada en k iteraciones. 32

Tabla 12 Costos del sistema embebido prototipo. 41

Tabla 13 Resumen final del rendimiento alcanzado en cada fase del proyecto. 46

Lista de Apéndices

Figura 5.1 Arquitectura del Codificador. 37

Figura 5.2 Arquitectura del modelo clasificador. 38

Figura 6.1 Montaje del sistema embebido prototipo. 42

Resumen

Titulo: Diseño de una herramienta para la deteccion de imagenes de cancer de piel utilizando redes neuronales profundas (DNN).

Autor: Andrew Dalai Rueda Rivera

Palabras Claves: Redes neuronales, Cancer, Deeep Learning, Deteccion, Imagenes, Piel, Contrastive.

En el presente proyecto se realizo la implementación de un sistema embebido prototipo capaz de clasificar imágenes. En primer lugar, se seleccionó la base de datos que se iba a trabajar teniendo en cuenta que existieran los suficientes datos para que no se presente overfitting ni underfitting, a grosso modo la base de datos más completa es la HAM10000. Para cumplir con el objetivo principal, se analizaron varias topologías base como la Resnet50, la MobileNetV2, la InceptionV2 y la EfficientnetV2, en el modelo clasificador se modificó la cantidad y el tipo de capas, las dimensiones de salida, la cantidad de filtros, el valor del dropout, la función de activación, los regularizadores, la cantidad de clases a clasificar, el tipo de validación, entre otros. Después de obtener los resultados se seleccionó la EfficientnetV2 porque tenía el mayor F1Score. A esta topología se le aplicó la técnica del aprendizaje contrastivo, que busca crear representaciones para discriminar las características que presentan similitudes y diferencias. El modelo final obtuvo un valor de F1-Score superior al 68 %. Se implementó la topología base seleccionada en la Raspberry pi 4 y se utiliza una cámara Logitech Brio para realizar la detección de imágenes. El sistema cuenta con una interfaz sencilla y amigable con el usuario, que permite interactuar fácilmente con el modelo clasificador. Puedes encontrar el código fuente en GitHub: Repositorio en GitHub.

Abstract

Title: Design of a tool for the detection of skin cancer images using deep neural networks (DNN).

Author: Andrew Dalai Rueda Rivera

Keywords: Neural networks, Cancer, Deep Learning, Detection, Images, Skin, Contrastive.

In this project, the implementation of a prototype embedded system capable of classifying images was carried out. Firstly, the database that was going to be worked on was selected taking into account that there was enough data so that no overfitting or underfitting occurred. Roughly speaking, the most complete database is the HAM10000. To meet the main objective, several base topologies were analyzed such as Resnet50, MobileNetV2, InceptionV2 and EfficientnetV2, in the classifier model the number and type of layers, the output dimensions, the number of filters, the dropout value, activation function, regularizers, number of classes to classify, validation type, among others. After obtaining the results, EfficientnetV2 was selected because it had the highest F1Score. The contrastive learning technique was applied to this topology, which seeks to create representations to discriminate characteristics that present similarities and differences. The final model obtained an F1-Score value greater than 68 %. The selected base topology was implemented on the Raspberry pi 4 and a Logitech Brio camera is used to perform image detection. The system has a simple and user-friendly interface, which allows easy interaction with the classifier model. You can find the source code on GitHub:Repository on GitHub.

Introducción

Según cifras del Instituto Nacional de Cancerología el cáncer de piel es una prioridad en la salud pública, ya que en el 2018 se reportó un aumento del 4.4 % con relación a los casos totales documentados en 2017. Actualmente los subtipos histológicos que predominan son: el carcinoma baso-celular (52.7 %), el escamo-celular (22.6 %) y el melanoma (16.1 %), siendo un factor agravante la gran probabilidad de contraer melanoma y posterior metástasis en población mayor a 60 años. Teniendo en cuenta esta característica de nuestro territorio, es necesario mejorar la prevención, diagnóstico y detección temprana de cualquier tipo de cáncer de piel.

En el presente proyecto, vamos a mencionar tres aspectos importantes a analizar: En primer lugar el producto a entregar que consiste en un prototipo de un sistema embebido, compuesto de un computador compacto con una cámara, que permitirá adquirir las imágenes a evaluar. Es importante recordar que el proyecto busca, a partir de redes neuronales y deep learning, determinar la posibilidad de tener cáncer de piel, en sus principales variantes, de acuerdo a una foto adquirida con el microcontrolador. En el segundo aspecto se analiza las personas o entidades interesadas en el producto, de manera directa se tienen como interesados principales los habitantes de países tropicales, en América y África, donde la intensidad de luz UV es mayor en relación a países de otras latitudes a lo largo del mundo. Asi mismo se enucentran las entidades relacionadas con la UIS,

como los grupos de Investigación: CPS, CEMOS - E3T. Por ultimo se tuvo en cuenta el impacto que tiene una herramienta como esta para la sociedad y el ambito cientifico, que alcance se pretende tener con el proyecto para influir en futuras investigaciones relacionadas con la optimización de productos en relación a la medicina moderna, los alcances del presente proyecto son:

Para la identificación del tipo red se realizo una revisión del estado del arte donde se realice reconocimiento de patrones de cáncer en imágenes utilizando las redes neuronales profundas. En la selección de la base de datos para el entrenamiento de las redes neuronales profundas, se tuvo en cuenta aquellas bases de datos de uso libre proporcionadas por el repositorio digital Kaggle en colaboración en Google LLC. La descripción de la arquitectura y el entrenamiento se realizo usando el lenguaje de programación Python, por medio de la plataforma Google Colab, apoyándose en librerías como Keras y Pandas. Como plataforma de cálculo se uso el backend de Tensor Flow. Como sistema de desarrollo embebido se uso un computador compacto con cámara que a futuro sea de uso portátil, el cual permita clasificar imágenes de los tipos más frecuentes de cáncer de piel, esto con el fin de facilitar su previo diagnóstico.

1. Objetivos

1.1. Objetivo General

Diseñar una herramienta para la detección de cáncer de piel en imágenes dermatoscópicas, mediante la clasificación por redes neuronales profundas.

1.2. Objetivos Específicos

- Seleccionar una topología base para entrenar la red neuronal y validar el entrenamiento realizado.
- Utilizar una base de datos para el entrenamiento y la validación de la red.
- Aplicar técnicas de aprendizaje contrastivo para mejorar la precisión del modelo base.
- Implementar la red neuronal final en un sistema embebido prototipo.

2. Marco De Referencia

2.1. Contexto Histórico

2.1.1. Cáncer de Piel

El cáncer es la segunda causa de muerte a nivel mundial y su estudio es relevante debido al aumento reciente de su incidencia. Esta enfermedad implica un crecimiento descontrolado de células, en este caso de la piel, debido a múltiples alteraciones genéticas que generan una neoplasia. Las causas se dividen en factores constitucionales, relacionados con la constitución genética y fenotípica de la persona, y factores ambientales, como la exposición a rayos UV o productos químicos(CDC: Centros para el Control y la Prevención de Enfermedades, 2023). Este libro aborda siete tipos de lesiones cutáneas, destacando las más importantes para el estudio, mientras que el resto se detalla en un anexo**??**.

2.1.1.1. Melanoma(mel). La mayoría de muertes por cáncer de piel se deben a este tipo específico de lesión cutánea, aunque solo representa el 1 % de los diagnósticos. Este cáncer surge de las células que producen pigmento, conocidas como melanocitos, que se encuentran en distintas partes del cuerpo. Puede originarse a partir de nevos ya existentes. Las causas incluyen tener piel clara, haber sufrido quemaduras solares graves en la infancia, vivir en áreas con altitud elevada o

cerca del ecuador, tener más de 50 nevos en el cuerpo, antecedentes familiares de esta enfermedad o simplemente la edad(Society, 2019).

2.1.1.2. Nevo Melanocitico(nv). Casi todas las personas tienen traumas cutáneos como manchas, pecas o lunares, que son lesiones comunes del sistema melanocitario debido a la reproducción benigna de melanocitos. Al igual que el melanoma, se originan a partir de células productoras de melanina y pigmentación de la piel, pero no representan un riesgo para quienes las tienen. Estas lesiones pueden variar y cambiar con el tiempo en forma, color o incluso desaparecer.

2.1.1.3. Carcinoma de Células Basales(bcc). El carcinoma de células basales (BCC) es el tipo de cáncer de piel más común, ubicado en la capa basal de la epidermis. Se caracteriza por su lento crecimiento y capacidad para destruir tejido local, como piel, tejido subcutáneo, cartílago y hueso, pero tiene baja probabilidad de metástasis. La incidencia de BCC varía entre 0.0028 % y 0.1 % (Castañeda GP, 2016). Las causas comunes incluyen exposición a rayos UV, arsénico, radioterapia, inmunosupresores, traumatismos mecánicos o cicatrices persistentes (Vargas et al., 2021). El BCC aumenta la sensibilidad de la piel, lo que puede causar lesiones adicionales tras el tratamiento. El 60 % de los casos afectan la cara, con nódulos rosados perlados o translúcidos. El 30 % de los casos se encuentra en el tronco y extremidades, y el 5-10 % restante es morfeiforme (Lobos & Lobos, 2011).

2.1.1.4. Queratosis Benigna(bkl). Esta patología se caracteriza por un aumento de ortoqueratina o paraqueratina sin displasia en el epitelio. Incluye el lentigo solar, conocido como

manchas por la edad debido a la exposición prolongada al sol (De Cinfa, 2022); la queratosis seborreica, una neoplasia benigna que suele aparecer en la vejez y se manifiesta como manchas marrones o negras ("Queratosis seborreica - síntomas y causas - Mayo Clinic", 2022); y la queratosis tipo liquen plano, descrita por primera vez en 1966, es más común en mujeres de 50 a 70 años con piel clara y generalmente no presenta síntomas (L et al., 2010).

2.2. Inteligencia Artificial

2.2.1. Procesamiento de Imágenes

El procesamiento de imágenes es un método clave que involucra procesos de inspección y tratamiento de imágenes, proporcionando resultados más precisos y rápidos que los métodos manuales. Es esencial para aplicaciones de Deep Learning como clasificación de imágenes, inspección visual, conducción autónoma y robótica ("¿Qué es el reconocimiento de imágenes?", s.f.). En salud, el procesamiento de imágenes es crucial para el desarrollo de tratamientos oportunos de diversas enfermedades.

2.2.1.1. Redes Neuronales Profundas DNN. Las redes neuronales profundas son un tipo de red artificial con numerosas capas ocultas y aprendizaje automático. Pueden aprender de experiencias anteriores y transformar la información de entrada, minimizando la intervención del programador. Su lógica interna es compleja y difícil de comprender, pero representan un gran avance en análisis de datos y detección de imágenes. Estas redes tienen el potencial de impulsar futuras innovaciones que beneficiarán a los humanos en tareas específicas.

2.3. Entorno de Trabajo

2.3.1. Colab

Google Collaboratory, o Colab, es una plataforma en la nube que ofrece un entorno gratuito para codificar en Python. Proporciona almacenamiento, memoria, capacidad de procesamiento, acceso a GPU y TPU, y la ventaja de no sobrecargar la computadora del usuario al realizar todo el trabajo en línea. Además de permitir la escritura de código, Colab admite texto enriquecido, enlaces, imágenes, tablas y gráficos. Su capacidad para compartir proyectos y permitir que varios colaboradores modifiquen el código en tiempo real la convierte en una herramienta útil para el trabajo en equipo.

2.3.2. Thonny

El entorno integrado de desarrollo Thonny Este software gratuito de codificación en Python es fácil de usar y adecuado para principiantes en programación. Incluye un intérprete para ejecutar el código, un editor para crear archivos de código fuente y un depurador que permite analizar paso a paso la ejecución del código y examinar las variables para resolver errores de forma efectiva.

3. Selección de una Base de Datos Adecuada para el Entrenamiento y Validación de la DNN Final

Para la selección de la base de datos se tuvo en cuenta aquellas bases de datos de uso libre proporcionadas por el repositorio digital Kaggle en colaboración con Google LLC. Para el presente proyecto se tuvo en cuenta dos bases de datos libres: “The HAM 10000” y “Skin cancer ISIC”.

3.1. The HAM10000

Human Against Machine (HAM) corresponde a una base de datos en la cual se almacenaron de forma organizada, las imágenes resultantes de un proceso de estandarización y limpieza de la información de imágenes dermatoscopicas. Entre su repertorio se encuentran 10015 imágenes de diferentes poblaciones, su fin es netamente académico y es de dominio público a través del archivo ISIC. La base de datos cuenta con siete clases genéricas, eliminando las clasificaciones ambiguas. Siendo elegidas por simplicidad, cubrieron más del 95 % de todas las lesiones pigmentadas examinadas previamente en casos de estudio, que corresponden a: Queratosis actínica y carcinoma intraepitelial/enfermedad de Bowen (akiec), Carcinoma de células basales (bcc), Lesiones benignas tipo queratosis (lentigos solares/queratosis seborreicas y queratosis tipo liquen plano) (bkl), Dermatofibroma (df), Melanoma (mel), Nevos melanocíticos (nv) y Lesiones vasculares (angiomas, angioqueratomas, granulomas piógenos y hemorragia) (vasc).

Las imágenes no están organizadas por etiquetas de lesiones cutáneas, sino por su image$_id.Porello, serealiz$

3.2. Skin Cancer ISIC

Se utilizó la herramienta digital Kaggle para acceder a una base de datos de 2357 imágenes de enfermedades oncológicas malignas y benignas, derivadas de la colaboración Skin Imaging Collaboration. Estas imágenes fueron organizadas en las categorías definidas por ISIC y divididas en subconjuntos equilibrados, excepto en el caso de las imágenes más predominantes como melanomas o lunares. La base de datos ya tiene una carpeta dedicada para pruebas.

3.3. Entrenamiento, Validación y Test

Ambas bases de datos se dividieron en tres subconjuntos: el 20 % se destinó a pruebas (test), ya que la base de datos ISIC tenía su propia división para este fin. Luego, se separó el 30 % de la base de datos *HAM10000* y el 20 % de la otra base de datos, según el tamaño de cada conjunto, para su validación. La validación es crucial para evitar que el algoritmo memorice las predicciones de los datos de entrada y logre alta precisión durante el entrenamiento, pero luego disminuya considerablemente al evaluar la red, conocido como *Overfitting*. El conjunto restante se utilizó para el entrenamiento del modelo.

La Tabla 1 muestra un resumen de la cantidad de imágenes en cada etapa de la simulación después de aplicar el algoritmo de división. Se eliminaron las categorías con una cantidad de datos insignificante, como se muestra en la Tabla 2. Para un análisis comparativo, se implementará el

modelo con ambas bases de datos, tanto con las siete categorías originales como con la reducción realizada.

Tabla 1

Contenido de las Bases de datos.

Clase	Cantidad Original		Entrenamiento		Validación		Test	
	HAM10000	ISIC	HAM10000	ISIC	HAM10000	ISIC	HAM10000	ISIC
Akiec	327	114	184	92	78	22	65	16
Bcc	514	376	289	301	123	75	102	16
Bkl	1099	462	616	370	264	92	219	16
Df	115	95	65	76	27	19	23	16
Mel	1113	438	624	351	267	87	222	16
Nv	6705	357	3755	286	1609	71	1341	16
Vasc	142	139	80	112	34	27	28	3
Total	10015	1981	5613	1588	2402	393	2000	99

Tabla 2

Contenido de la base de datos modificada a cuatro categorías.

Clase	Numero de datos	
	HAM10000	ISIC
Bcc	514	376
Bkl	1099	462
Mel	1113	438
Nv	6705	357
Total	9431	1633

4. Selección de una Topología Base para Entrenar la Red Neuronal y Validar el Entrenamiento Realizado

Uno de los primeros factores considerados en la selección de la red neuronal fue su implementación en un sistema embebido dentro de una Raspberry Pi, lo que presenta desafíos en términos de memoria y capacidad de cómputo. Por esta razón, se verificó el tamaño de la red neuronal para asegurarse de que el hardware pudiera manejar las conexiones y el flujo de datos sin exceder los límites operativos del sistema. La red seleccionada no puede ser demasiado grande para evitar sobrecargar el sistema.

Figura 1.
Modelos disponibles en Keras

MODELO	TAMAÑO (MB)	RENDIMIENTO TOP-1	RENDIMIENTO TOP-5	PARAMETRPS	TIEMPO (ms) POR PASO DE INFERENCIA (GPU)
ResNet50	98	74.9%	92.1%	25.6M	4.6
InceptionV3	92	77.9%	93.7%	23.9M	6.9
MobileNetV2	14	71.3%	90.1%	3.5M	3.8
EfficientNetV2B0	29	78.7%	94.3%	7.2M	4.9

Por estas razones, se optó por trabajar con modelos preentrenados que tuvieran un tamaño menor o igual a 100 MB. Luego, se examinó el tiempo de inferencia por paso en una CPU, buscando que no superara los 60 milisegundos, un tiempo de espera razonable. Las redes neuronales

profundas preentrenadas seleccionadas para su estudio fueron: **ResNet50**, **InceptionV3**, **MobileNetV2** y **EfficientNetV2B0**. La Figura **??** muestra un resumen de los atributos de las DNN en base a Keras.

4.1. Implementación de Modelo

Se implementan los modelos teniendo en cuenta los siguientes aspectos:

- Se cargó la red neuronal profunda preentrenada para trabajar en el modelo. Las opciones seleccionadas previamente incluyen **EfficientNetV2B0**, **MobileNetV2**, **ResNet50** e **InceptionV3**.Los pesos previos de cada red se cargaron según el entrenamiento con ImageNet.

- Inicialmente, se trabajó con la base de datos **HAM10000** debido a su mayor cantidad de imágenes. Se probaron tres modelos diferentes para el algoritmo de clasificación, repitiendo cada uno tres veces.

- Se configuró la red para permitir modificar solo los pesos de las últimas cuatro capas durante la fase de entrenamiento.

- Para evitar el overfitting y underfitting, se aplicó aumento de datos, que implica modificar características de los datos existentes para darles diversidad. Se realizaron transformaciones a las imágenes de entrada, incluyendo aumento de tamaño en un factor de 0.2, rotación en un factor de 0.1 e inversión tanto horizontal como verticalmente.

- Se definió la estructura del modelo clasificador y se evaluaron tres algoritmos diferentes. Se eligió la arquitectura mostrada en la Figura 2.

Figura 2.
Resumen de capas del segundo modelo en Colab

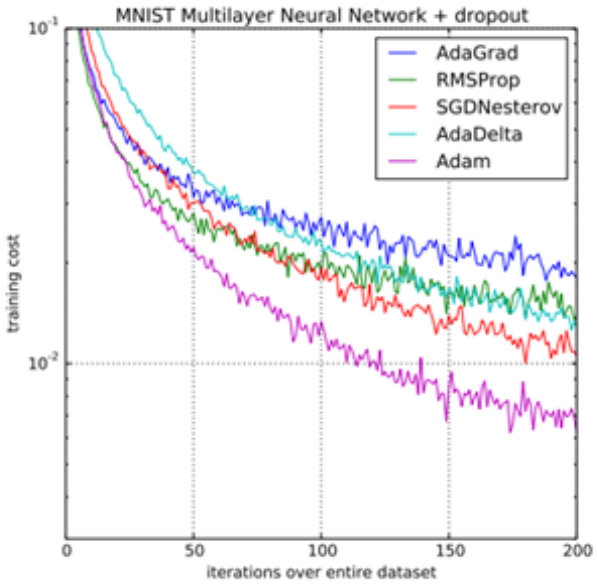
Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 180, 180, 3)]	0
sequential (Sequential)	(None, 180, 180, 3)	7
efficientnetv2-b0 (Functional)	(None, None, None, 1280)	5919312
flatten (Flatten)	(None, 46080)	0
dense (Dense)	(None, 385)	17741185
batch_normalization (BatchNormalization)	(None, 385)	1540
dropout (Dropout)	(None, 385)	0
dense_1 (Dense)	(None, 7)	2702

Total params: 23664746 (90.27 MB)
Trainable params: 17992977 (68.64 MB)
Non-trainable params: 5671769 (21.64 MB)

- Se utilizó la función de pérdida **“categorical_consentropy”** ya que es esencial cuando se trabaja con dos o más clases de etiquetas codificadas como **one_hot**.

- Además, se utilizó el optimizador **Adam**, seleccionado por su alta eficiencia y los beneficios que ofrece cuando se maneja una gran cantidad de parámetros. La decisión se basó en la Figura 3, que compara el rendimiento de varios optimizadores y su costo de entrenamiento para el conjunto de datos MNIST con regularización por Dropout. Los autores sugieren una tasa de aprendizaje de 0.001 (Kingma & Ba, 2014).

Figura 3.
Comparativa de rendimiento vs costo de entrenamiento de optimizadores (Kingma & Ba, 2014)



- Para mejorar la precisión y el tiempo de entrenamiento, se implementaron varias devoluciones de llamada (callbacks), que permiten realizar tareas en diferentes etapas del entrenamiento. Se seleccionaron tres métodos específicos para la ejecución del modelo: **EarlyStopping**, **ReduceLROnPlateau** y **ModelCheckpoint**.

- Para el entrenamiento se utilizó un tamaño de lote **(Batch size)** de 64 muestras. Además, se estableció un límite máximo de 1000 épocas para el entrenamiento del modelo, controlando así el número de iteraciones a realizar durante el proceso.

4.2. Validación Cruzada de K Iteraciones

El K-fold cross-validation es una técnica que ayuda a reducir el riesgo de overfitting en el entrenamiento de un modelo.Esta metodología mezcla aleatoriamente los datos de entrenamiento y validación y los divide en k partes. Durante cada iteración, k-1 partes se usan para entrenar el modelo, mientras que una se reserva para validarlo. El desempeño final del modelo se calcula promediando los resultados de las k iteraciones. Inicialmente se eligió k=10, pero se redujo a 8 debido a los altos costos computacionales. Se recomienda un valor de k entre 5 y 10 según el tamaño del conjunto de datos (“3.1. Cross validation: evaluating estimator performance”, s.f.).

4.3. Análisis de Resultados

El resumen de los resultados se realizó mediante una matriz de confusión. Además, el algoritmo muestra métricas clave para evaluar el rendimiento del modelo. Con la función **metrics.classification_report()** se obtienen varias métricas: **Precisión**, que mide la exactitud del algoritmo para clasificar correctamente las muestras; **Recall**, que evalúa la capacidad del algoritmo para identificar muestras positivas; **F1-score**, que es la media armónica entre precisión y recall; y **Suppot**, que indica el número de muestras clasificadas correctamente en cada categoría.

Los resultados se obtuvieron de la siguiente manera: primero, se trabajó con la base de datos HAM10000 para identificar siete categorías de lesiones cutáneas (akiec, bcc, bkl, df, mel, nv, vasc) en las imágenes de entrada, aplicando las tres arquitecturas propuestas para el modelo.

Luego, se simplificó el análisis a cuatro categorías (bcc, bkl, mel, nv), las más representativas en términos de cantidad de datos, para reducir el riesgo de overfitting. Después, se trabajó con la base de datos ISIC de forma similar, comenzando con las siete categorías y luego reduciendo a las mismas cuatro clases para realizar una comparación. Los cuatro modelos de redes neuronales profundas preentrenadas se implementaron para identificar cuál ofrece las mejores características.

4.3.1. Clasificación en siete Categorías

A continuación se presentan los resultados obtenidos al aplicar redes neuronales profundas preentrenadas, configuradas con un Dropout de 0.2 y una tasa de aprendizaje de 0.001, durante 1000 épocas con un tamaño de lote predeterminado. Los datos de entrada provienen del conjunto de datos HAM10000. Se realizaron tres iteraciones para cada uno de los tres modelos planteados para calcular un promedio de los F1-Score. La Tabla 3 destaca el mayor F1-Score de cada red neuronal profunda para identificar el modelo de prueba más efectivo en comparación con los demás.

Tabla 3

Resultados del F1-Score en el test en diferentes iteraciones para los tres modelos de prueba.

Modelo	Prueba 1			Prueba 2			Prueba 3			Totales		
	I1	I2	I3	I1	I2	I3	I1	I2	I3	P1	P2	P3
Mobilenet V2	0,5157	0,5100	0,5100	0,5557	0,4885	0,5242	0,5328	0,5128	0,5114	0,5119	0,5228	0,5190
Resnet 50	0,5742	0,5557	0,5742	0,5628	0,5700	0,5485	0,5814	0,5414	0,5185	0,5680	0,5604	0,5471
EfficientNet V2	0,5800	0,5714	0,5985	0,5785	0,6057	0,6057	0,5728	0,6000	0,6000	0,5833	0,5966	0,5909
Inception V3	0,4528	0,4542	0,4542	0,5271	0,4885	0,4542	0,4771	0,4757	0,5000	0,4537	0,4899	0,4843
Promedio Total										0,5292	0,5425	0,5353

Se eligió el segundo modelo porque, tras analizar el promedio de las iteraciones, mostró

el mayor F1-Score en comparación con los otros modelos. La Tabla 4 resume los resultados obtenidos para el segundo modelo, mostrando la cantidad de parámetros, el F1-Score y las pérdidas. El número de parámetros es constante, sin importar las iteraciones realizadas. Luego, se utilizó el conjunto de datos de la segunda base de datos derivada de ISIC.

Tabla 4

Resultados del F1-Score y las pérdidas con validación normal.

Modelo	Parametros	F1 Score	Perdidas
Mobilenet V2	14582618	0,5228	11,1358
Resnet 50	51977626	0,5604	12,7396
EfficientNet V2	23664746	0,5966	5,0652
Inception	34423098	0,4899	8,0861

Tabla 5

Resultados del F1-Score y las pérdidas con validación normal.

Modelo	Parametros	F1 Score	Perdidas
Mobilenet V2	14582618	0,5257	2,17691
Resnet 50	51977626	0,5671	29,67917
EfficientNet V2	23664746	0,5628	2,82604
Inception	34423098	0,5157	18,36415

4.3.2. Clasificación en cuatro Categorías

Se mantiene la configuración de los hiper parámetros, el conjunto de datos modificado de HAM10000 se limitó a solo cuatro categorías. La Tabla 6 resume los resultados obtenidos: **EfficientNetV2** presenta el mayor F1-Score, la menor perdida la presenta la red neuronal **InceptionV3** y **MobilenetV2** tiene la menor cantidad de parámetros.

Tabla 6

Resultados del F1-Score y las pérdidas con validación normal.

Modelo	Parametros	F1 Score	Perdidas
Mobilenet V2	14581460	0,6150	0,8359
Resnet 50	51976468	0,6675	17,1086
EfficientNet V2	23663588	0,6750	7,3582
Inception	34421940	0,6200	0,7051

Usando la segunda base de datos derivada de ISIC y limitándola a solo cuatro categorías, los resultados obtenidos se resumen en la Tabla 7. Se observa que la red neuronal **EfficientNetV2** tuvo el mejor F1-Score y la menor pérdida, mientras que **MobilenetV2** presentó la menor cantidad de parámetros.

Tabla 7

Resultados del F1-Score y las pérdidas con validación normal.

Modelo	Parametros	F1 Score	Perdidas
Mobilenet V2	14581460	0,5900	1,89366
Resnet 50	51976468	0,5825	2,01147
EfficientNet V2	23663588	0,6675	1,3185
Inception	34421940	0,6075	1,7871

4.3.3. Aplicación de la validación cruzada

4.3.3.1. Clasificación en siete categorías. Se mantiene la configuración de los hiper parámetros, a excepción del batch_size de 64, se utilizo como datos de entrada el dataset obtenido de la base de datos HAM10000.

La Tabla 8 examina el impacto del dropout en el F1 score del modelo de clasificación, probando valores de dropout entre 0.2 y 0.5. Se resaltaron los mejores rendimientos de cada red neuronal preentrenada para identificar el valor de dropout que maximiza la eficiencia del modelo. El valor de dropout que ofrece el mejor F1-Score promedio es 0.3. Además, la red neuronal profunda más eficiente para clasificar siete categorías utilizando validación cruzada en k iteraciones es InceptionV3 con un dropout de 0.5.

Tabla 8

Resultados del F1-Score con validación cruzada para distintos valores del dropout.

Modelo	Dropout			
	0,2	0,3	0,4	0,5
Mobilenet V2	0,6157 ± 0,0258	0,6294 ± 0,0314	0,6007 ± 0,0819	0,6230 ± 0,0599
Resnet 50	0,6236 ± 0,0538	0,5793 ± 0,0564	0,5609 ± 0,0928	0,6184 ± 0,0599
EfficientNet V2	0,6222 ± 0,0288	0,6209 ± 0,0512	0,5677 ± 0,1873	0,5356 ± 0,2102
Inception V3	0,6226 ± 0,0656	0,6566 ± 0,0189	0,6688 ± 0,0025	0,6705 ± 0,0000
Total	0,6210	0,6216	0,5995	0,6119

Se aplicó la técnica de validación cruzada a la segunda base de datos, que contiene una menor cantidad de imágenes. En la Tabla 9, se observa que la red neuronal profunda **Resnet50** mostró el mejor rendimiento entre las implementadas. **InceptionV3** fue la opción con la menor pérdida. Dado que todas las redes neuronales mostraron bajos niveles de eficiencia, se decidió no explorar el efecto de cambiar el valor del dropout, ya que no se esperaba obtener resultados que cumplieran con los requisitos del proyecto.

Tabla 9

Resultados del F1-Score y la pérdida con validación cruzada en k iteraciones.

Modelo	F1-Score	Losses
Mobilenet V2	0,1566 ± 0,0502	4,6451 ± 0,7068
Resnet 50	0,1894 ± 0,0334	5,0238 ± 0,6436
EfficientNet V2	0,1452 ± 0,0306	4,3078 ± 0,7198
Inception V3	0,1402 ± 0,0198	3,1877 ± 0,1845

4.3.3.2. Clasificación en cuatro categorías. La Tabla 10 muestra el impacto de variar el dropout en el modelo. El dropout que generó el mejor F1-Score promedio fue 0.4. Además, la red neuronal profunda con mayor eficiencia en la clasificación de cuatro categorías, utilizando validación cruzada en k iteraciones, fue MobileNetV2 con dropout de 0.4.

Tabla 10

Resultados del F1-Score con validación cruzada para distintos valores del dropout.

Modelo	Dropout			
	0,2	0,3	0,4	0,5
Mobilenet V2	0,7071 ± 0,0117	0,7118 ± 0,0000	0,7118 ± 0,0000	0,7034 ± 0,0147
Resnet 50	0,6996 ± 0,0117	0,6936 ± 0,0364	0,7060 ± 0,0091	0,7038 ± 0,0115
EfficientNet V2	0,7057 ± 0,0117	0,6920 ± 0,0369	0,7070 ± 0,0056	0,6857 ± 0,0596
Inception V3	0,6831 ± 0,0519	0,6758 ± 0,0598	0,7076 ± 0,0071	0,6510 ± 0,0537
Total	0,6989	0,6933	0,7081	0,6860

Se aplicó la técnica de validación cruzada a la segunda base de datos con menos imágenes y etiquetas. La Tabla 11 muestra que los valores mejoraron notablemente respecto al análisis con siete categorías, aunque la eficiencia sigue siendo baja para todas las redes neuronales.

Tabla 11

Resultados del F1-Score y la pérdida con validación cruzada en k iteraciones.

Modelo	F1-Score	Losses
Mobilenet V2	0,3164 ± 0,0662	2,8985 ± 0,5449
Resnet 50	0,2363 ± 0,0344	4,2363 ± 0,8546
EfficientNet V2	0,2865 ± 0,0523	3,3235 ± 0,4033
Inception V3	0,2637 ± 0,0896	2,2822 ± 0,2380

Después de analizar los resultados obtenidos al modificar diversos atributos del modelo, incluyendo el uso de capas de BatchNormalization, diferentes configuraciones de capas densas y convolucionales, regularizadores, optimizadores, funciones de activación, dropout, redes neuronales profundas, número de etiquetas, técnicas de validación, datos de entrada, normalización de imágenes, tamaño de lote, y callbacks, se concluyó que la mejor topología base para entrenar el modelo es **EfficientNetV2**. Esta red neuronal se utilizó para clasificar cuatro categorías utilizando la base de datos HAM10000 y validación cruzada durante el entrenamiento. Se alcanzó un F1-Score del 70.7 % con 23,663,588 parámetros.

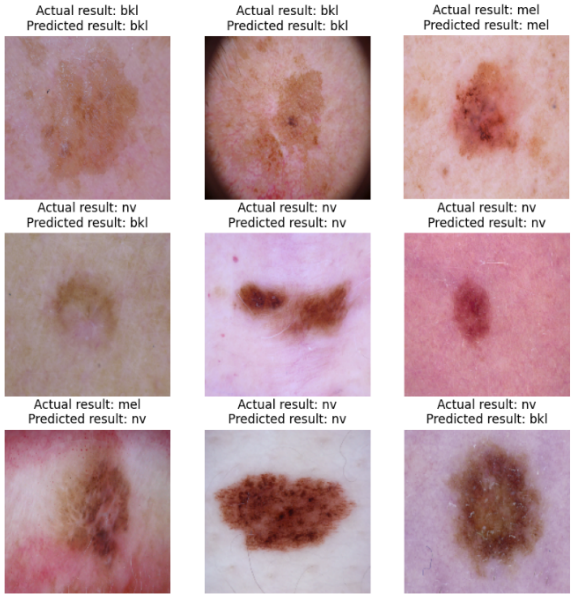
4.3.4. Resultados de la Mejor Topología Base

Figura 4.
Evaluación del modelo con los datos del test.

```
-------------------------Test Results-----------------------
59/59 - 3s - loss: 7.3582 - accuracy: 0.8084 - 3s/epoch - 56ms/step
Test Loss: 7.3582353591918945
Test Accuracy: 0.8083863854408264
```

Al evaluar el modelo con los datos del test se obtuvo un rendimiento de 80,838 % y una pérdida de 7,3582, el testeo completo 59 épocas durante un tiempo total de 3 segundos, en el cual el tiempo promedio por paso, que corresponde a la actualización de los pesos con base en un lote de datos, fue de 56 milisegundos como se observa en la Figura 4.

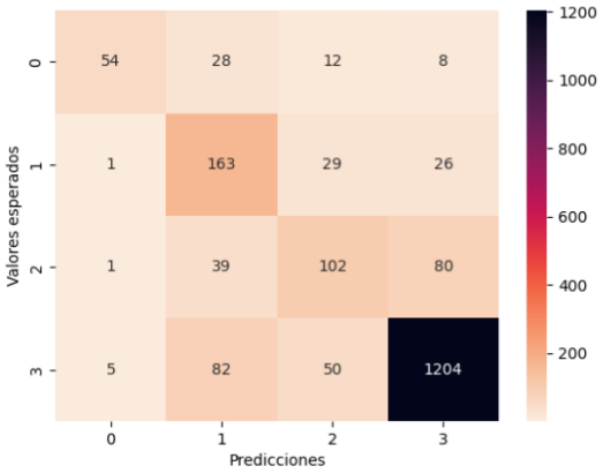
Figura 5.
Imágenes del test clasificadas con la topología base seleccionada.



Se observa en la Figura 5 nueve imágenes aleatorias del lote del test, el titulo de cada una se compone de la etiqueta real y la predicción realizada por el modelo. Seis de las nueve imágenes clasificadas tienen la etiqueta correcta, además que la categoría nevus presenta un mayor grado de acierto en la clasificación.

A continuación, se presenta la matriz de confusión y las métricas de eficiencia para el test:

Figura 6.
Matriz de confusión y métricas de eficiencia del test.



	precision	recall	f1-score	support
0	0.89	0.53	0.66	102
1	0.52	0.74	0.61	219
2	0.53	0.46	0.49	222
3	0.91	0.90	0.91	1341
accuracy			0.81	1884
macro avg	0.71	0.66	0.67	1884
weighted avg	0.82	0.81	0.81	1884

- El F1-Score final fue de 66.75 %.

- La precisión supera el 50 % en cada una de las clases.

- El promedio no ponderado para las tres métricas supera el 66 % mientras que cuando se pondera con la cantidad de muestras aumenta a más del 81 %, pues cada clase tiene un peso dependiendo de la cantidad de muestras que posea, por lo tanto aumenta debido a que la clase tres tiene más peso y presenta mayor eficiencia que el resto.

- Hay un gran desequilibrio en la cantidad de muestras de cada clase perteneciente a la base de datos, con respecto al total la clase 0 corresponde al 5,5 %, la clase 1 corresponde al

11,6 %, la clase 2 corresponde al 11,8 % y la última categoría posee el 71.1 % del total de las muestras.

- Los nevos y los carcinomas de células basales, cuando son clasificados incorrectamente, tienden a ser confundidos por el algoritmo con queratosis benigna pigmentada. Por otro lado, los melanomas y las lesiones queratósicas benignas de tipo bkl, si se clasifican de forma errónea, suelen asociarse con nevos.

- La clase 3 posee la mayor eficiencia en la clasificación, mientras que la clase 2 posee la menor eficiencia. El modelo tiene mejores resultados clasificando aquellas lesiones cutáneas que no son cancerígenas, como se observa en la métrica de recall.

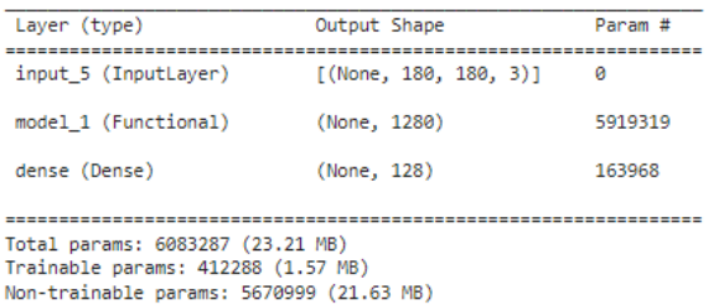
5. Aplicaciones de Técnicas de Aprendizaje Contrastivo para Mejorar la Precisión del Modelo Base

El aprendizaje contrastivo es una técnica de auto-supervisión que se utiliza en tareas de clasificación y detección cuando los datos disponibles son limitados. Esta metodología contrasta las muestras entre sí para identificar características compartidas y diferenciadas. Los pares positivos son muestras con representaciones similares, mientras que los pares negativos son aquellas que difieren (Salama, 2020).

El procedimiento para aplicar el aprendizaje contrastivo es el siguiente (Fonseca et al., 2022):

- Se toma el conjunto de datos y se crean lotes.
- Se aplica una transformación a las imágenes del lote con la finalidad de obtener un par de imágenes.
- Se entrena un codificador para que aprenda a generar representaciones vectoriales, con el fin de que los datos visuales pertenecientes a la misma clase tengan similitudes y entre clases se diferencien.

Figura 5.1.
Arquitectura del Codificador.



- Se utiliza la función de la similitud del coseno con el objetivo de calcular las características semejantes.

- La pérdida de entropía cruzada se calcula utilizando una escala de temperatura normalizada, que por recomendación de Keras, tiene un valor de 0.05. Esta forma de pérdida también se conoce como pérdida de contraste.

- El encoder se utiliza con sus pesos congelados en el modelo clasificador durante el entrenamiento, lo que permite mantener las características aprendidas previamente sin ajustarlas. Después del entrenamiento, se calcula la pérdida de entropía cruzada para evaluar el modelo.

Figura 5.2.
Arquitectura del modelo clasificador.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_6 (InputLayer)        [(None, 180, 180, 3)]     0

 model_1 (Functional)        (None, 1280)              5919319

 flatten (Flatten)           (None, 1280)              0

 dense_1 (Dense)             (None, 385)               493185

 batch_normalization (Batch  (None, 385)               1540
 Normalization)

 dropout (Dropout)           (None, 385)               0

 dense_2 (Dense)             (None, 4)                 1544

=================================================================
Total params: 6415588 (24.47 MB)
Trainable params: 495499 (1.89 MB)
Non-trainable params: 5920089 (22.58 MB)
_________________________________________________________________
```

Se utilizó como base la librería de keras para obtener un código funcional que aplique la técnica de aprendizaje contrastivo.

5.1. Análisis de Resultados Aplicando Aprendizaje Constrastivo

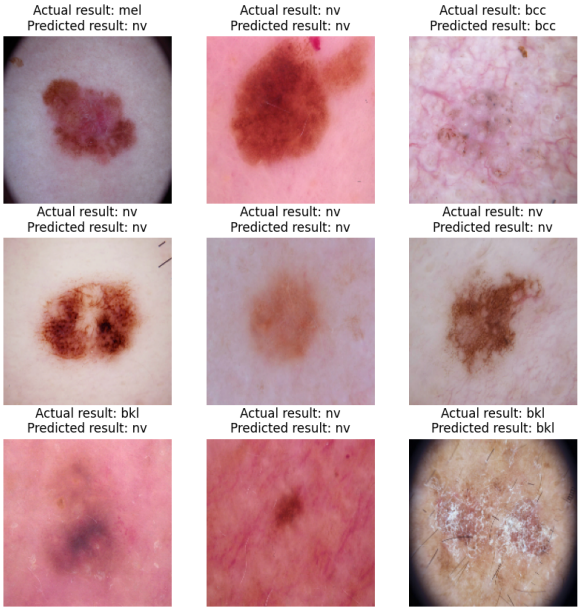
Figura 7.
Evaluación de la técnica con los datos del test.

```
-------------------------Test Results-----------------------
59/59 - 3s - loss: 0.7947 - accuracy: 0.8248 - 3s/epoch - 45ms/step
Test Loss: 0.7947413325309753
Test Accuracy: 0.824840784072876
```

Al evaluar el modelo con los datos del test se obtuvo un rendimiento de 82,484 % y una pérdida de 0.79474, el testeo completo 59 épocas durante un tiempo total de 3 segundos, en el cual el tiempo promedio por paso, fue de 45 milisegundos. Se predijo la etiqueta de nueve imágenes

aleatorias obtenidas de la base de datos del test.

Figura 8.
Imágenes del test clasificadas con aprendizaje contrastivo.



A continuación, se presenta la matriz de confusión y las métricas de eficiencia para el test.

- El objetivo principal de aplicar esta técnica era buscar incrementar el F1-Score del modelo seleccionado en el capítulo anterior, logrando un F1-Score final de 68.25 %.

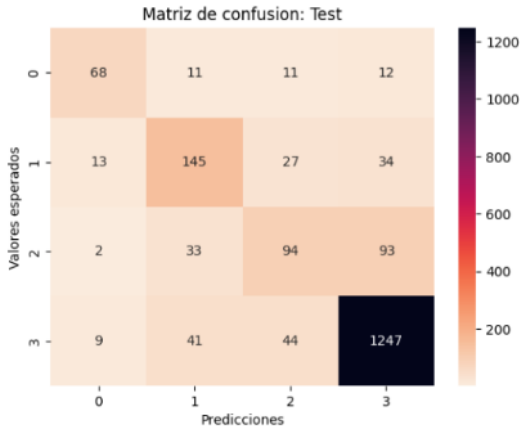
- La precisión supera el 50 % en cada una de las categorías.

- El promedio no ponderado para las tres métricas supera el 67 % mientras que cuando se pondera con la cantidad de muestras aumenta a más del 82 %, debido a que la categoría tres tiene más peso y presenta mayor eficiencia que el resto de las clases.

- Hay un gran desequilibrio en la cantidad de muestras de cada clase, la clase 0 corresponde al 5,5 %, la clase 1 corresponde al 11,6 %, la clase 2 corresponde al 11,8 % y la última categoría posee el 71.1 % del total de las muestras.

- Para los diferentes tipos de lesiones cutáneas, cuando el algoritmo los clasifica de forma errónea, la mayoría tienden a asociarse al nevo (Clase 3). La clase 3 posee la mayor eficiencia en la clasificación, mientras que la clase 2 posee la menor eficiencia.

Figura 9.
Matriz de confusión y métricas de eficiencia del test.



	precision	recall	f1-score	support
0	0.74	0.67	0.70	102
1	0.63	0.66	0.65	219
2	0.53	0.42	0.47	222
3	0.90	0.93	0.91	1341
accuracy			0.82	1884
macro avg	0.70	0.67	0.68	1884
weighted avg	0.82	0.82	0.82	1884

Se decidió mantener la base de datos desequilibrada, pues el impacto que genera la cantidad de datos pertenecientes a la clase 3 permiten que la eficiencia del modelo clasificador aumente, además de disminuir el impacto del overfitting en los resultados.

6. Implementación de la Red Neuronal Final en un Sistema Embebido Prototipo

Para el modelo embebido, se eligió una Raspberry Pi 4 modelo B con 4 GB de RAM, que es una computadora pequeña y económica, adecuada para diversas tareas digitales (Electronics, s.f.). También se seleccionó una cámara web Logitech Brio Pro 4K a 30 fps con HDR, que ofrece una excelente calidad de imagen, frecuencia de cuadro y encuadre automático para obtener imágenes claras, fluidas y detalladas (Logitech, s.f.). La Tabla 19 resume los costos del montaje del prototipo.

Tabla 12

Costos del sistema embebido prototipo.

Dispositivo	Costo
Raspberry Pi 4 Model B	$380.000
Logitech Brio Pro	$689.000
Fuente 5V	$65.000
Adaptador	$8.800
Total	$1.142.800

En la página web de Mercado Libre se realizó la cotización de los implementos para montar el sistema embebido que consiste en una Raspberry Pi 4 Model B, una Logitech Brio Pro, Una fuente de 5V y un adaptador HDMI. El montaje del prototipo final se puede observar en la figura 6.1

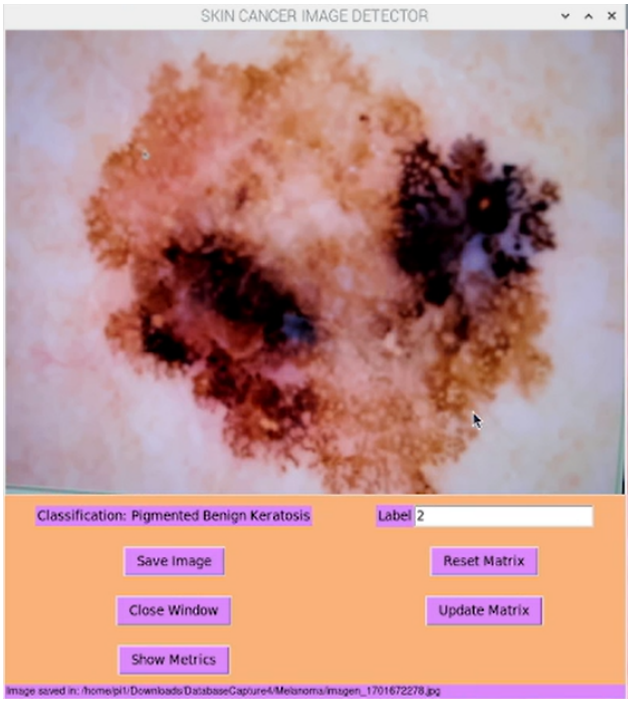
Figura 6.1.
Montaje del sistema embebido prototipo.

El algoritmo para realizar la clasificación en la Raspberry Pi se implementó en el lenguaje de programación Python. Se usó la herramienta OpenCV para capturar imágenes de la cámara y realizar la clasificación. Sin embargo, hubo un problema con esta utilidad, ya que la calidad de la imagen disminuía, afectando la resolución. Como solución a este problema, se cambió la forma en que el modelo final carga la base de datos. Anteriormente, se utilizaba el comando **image_dataset_from_directory**, pero se reemplazó por **cv2** para que el modelo pudiera detectar nuevos patrones y características de las imágenes con mayor calidad al ser leídas directamente por esta herramienta.

6.1. Interfaz del modelo clasificador en la Raspberry PI

El sistema captura contenido audiovisual de la cámara utilizando la herramienta **OpenCV**, tras configurar previamente la resolución de la cámara a 640 x 480. Posteriormente, el sistema clasifica el contenido utilizando el modelo final seleccionado con la topología base.

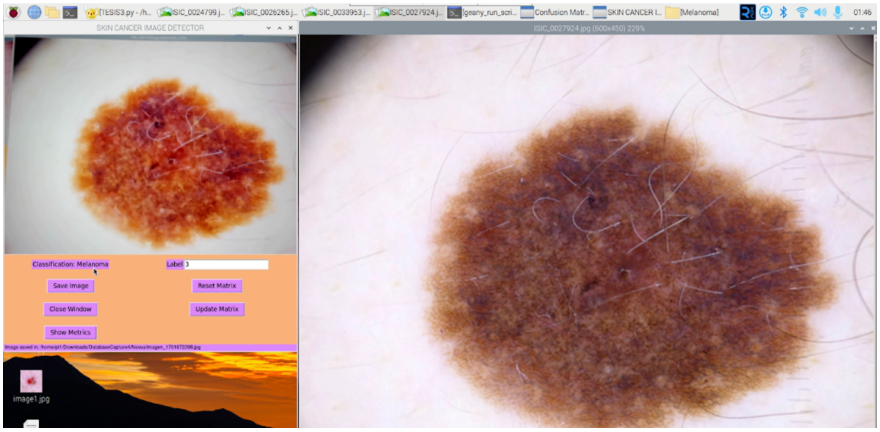
Figura 10.
Interfaz del sistema embebido.



Se desarrolló una interfaz sencilla con una barra de título que dice "SKIN CANCER IMAGE DETECTOR", un panel para visualizar la imagen, un cuadro de texto para mostrar la predicción, otro cuadro de texto con un marco para ingresar la etiqueta real, y cinco botones para guardar la imagen, cerrar la ventana, mostrar métricas, actualizar y reiniciar la matriz de confusión. La interfaz fue personalizada con colores de fondo, estilo y diseño de botones, y una disposición espaciosa de sus elementos. También se añadió una barra de estado que muestra las acciones realizadas. Cuando el usuario interactúa con la interfaz y selecciona el botón de actualización de la matriz, se abre una ventana adicional que presenta la gráfica con los valores actuales.

6.2. Análisis de Resultados

Figura 11.
Prueba del prototipo clasificando una imagen del test, ver Vídeo



Dado que la obtención de imágenes dermatoscópicas requiere la aprobación de un comité de ética, se decidió extraer aleatoriamente diez imágenes por cada clase de la base de datos HAM10000 del subconjunto de prueba. De esta manera, se asegura que el modelo no haya sido entrenado con ellas ni aprendido de memoria su contenido, se cumplen las normativas éticas y se puede evaluar el desempeño del prototipo con etiquetas reales verificadas para comparar las predicciones. Cada una de las imágenes seleccionadas se abre en el dispositivo y se detectan con la cámara. A medida que el sistema realiza la clasificación, las imágenes se almacenan en una carpeta del dispositivo. En total, se llevaron a cabo cuarenta experimentos, la prueba final arrojó un desempeño estimado de 57.74 %. Como se muestra en la Figura 11, la imagen original experimenta cambios cuando es detectada por la cámara, transformada y mostrada en la pantalla. Estos cambios afectan sus características, lo que a su vez reduce el rendimiento del modelo durante la

clasificación, ya que tiende a confundir patrones y atributos entre clases.

A continuación, se presenta la matriz de confusión y las métricas de eficiencia para la prueba del funcionamiento del prototipo clasificador, donde se puede observar que:

- El F1-Score final del sistema embebido fue de 57.74 %.
- La precisión, el recall y el rendimiento superan el 46 % en cada una de las categorías.

Figura 12.
Matriz de confusión de la prueba del sistema prototipo.

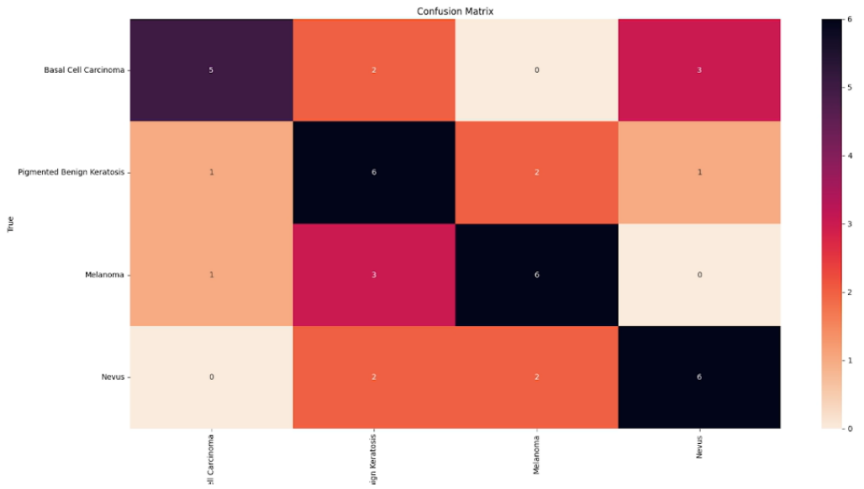


Figura 13.
Métricas de eficiencia de la prueba del sistema prototipo.

	precision	recall	f1-score	support
Basal Cell Carcinoma	0.714286	0.500	0.588235	10.000
Melanoma	0.600000	0.600	0.600000	10.000
Nevus	0.600000	0.600	0.600000	10.000
Pigmented Benign Keratosis	0.461538	0.600	0.521739	10.000
accuracy	0.575000	0.575	0.575000	0.575
macro avg	0.593956	0.575	0.577494	40.000
weighted avg	0.593956	0.575	0.577494	40.000

A continuación, en la Tabla 13 se muestra un resumen del F1-Score en los diferentes procesos llevados a cabo en el presente proyecto:

Tabla 13

Resumen final del rendimiento alcanzado en cada fase del proyecto.

Fase	EfficientnetV2	Aprendizaje Contrastivo	Sistema Embebido
F1-Score	0,6675	0,6825	0,5774

7. Conclusiones

En virtud del objetivo principal se logró construir un prototipo para la clasificación de 4 tipos de lesiones cutáneas, el cual dispone de una red neuronal EfficientnetV2 con un índice de eficiencia 68.25 % en el software y un 57.74 % en la validación. Este prototipo fue elaborado sobre una Raspberry pi 4+ y una cámara(modelo de la cámara).

Una de las primeras observaciones que se tienen es que debido a la forma como se decidió validar el modelo, se presento una disminución de la exactitud del mismo. Recordando que la validación se dio a través de la toma de imágenes desde la cámara a una pantalla que proyectaba las imágenes. Uno de los problemas principales es la introducción de artefactos, bordes y otros elementos que no son parte de la imagen original y que inducen el error en la clasificación de la red. Adicionalmente, la variación en el foco y la iluminación producen alteraciones en la captura en cuanto al color y la distribución de los píxeles, lo cual también induce a una mala clasificación. La razón por la que se optó por este método fue por que no se dispone de pacientes de control para la prueba .

En cuanto a las dos bases de datos planteadas como fuente de información, la HAM10000 fue la elegida por su cantidad de imágenes y por ser uno de los pocos conjuntos de datos de lesiones cutáneas existentes, debido al riguroso proceso legal que se debe llevar a cabo para la

extracción de material dermatoscópico. No obstante, se encuentra desequilibrada pues la mayoría de sus elementos se concentran en una clase lo que dificulta el potencial de clasificación de las clases menos predominantes. Es necesario analizar la posibilidad de mejorar el modelo a traves de un modelo de perdidas ponderado o a través de la inclusión de más imágenes en las clases menos predominantes, con el fin de proporcionar un mejor desemepeño en el futuro.

De los resultados obtenidos después de realizar cada uno de los entrenamientos, se puede deducir que el modelo clasifica de mejor forma, a aquellas imágenes de lesiones cutáneas benignas. Esto se debe a que las diferencias de forma, color y tamaño en cuanto a las lesiones clasificadas como malignas o benignas son mas amplias. En cambio cuando las clases se dividen en lesiones particulares, las similitudes son mayores y eso hace más difícil la tarea de clasificación.

Aplicar el aprendizaje contrastivo al modelo final permitió aumentar el rendimiento en 2 puntos porcentuales aunque se considera que la mejora no es significativa. Una de las causas se debe a que el modelo ya está preentrenado y sus kernel ya han sido modificados por un conjunto muy amplio de imágenes que otorgo una generalización mayor de las características. Es necesario encontrar la combinación perfecta de atributos y parámetros en el modelo clasificador permite explotar la ventaja del codificador con pesos favorables. pro ejemplo: modificar la temperatura de la pérdida de contraste funciona como calibrador pues permite obtener una mayor sensibilidad en las probabilidades de clasificación, acentuando la opción predicha. Durante el proyecto se realizó una búsqueda manual de todos estos atributos lo que no permitió obtener un modelo óptimo, pero se sugiere implementar una búsqueda parametrizada en el futuro.

Se recomienda para futuros estudios en la temática del presente proyecto, trabajar en coordinación con dermatólogos para realizar una recolección, autorizada por un comité de ética, de imágenes dermatoscopicas para ampliar las bases de datos existentes. Adicionalmente tener en cuenta los factores externos para realizar la clasificación, de esta forma se minimiza el margen de error. Finalmente, esto es un primer paso, a medida que surjan nuevo conceptos o topologías de red, se sugiere tomar como base el código y realizar mejorías que considere oportunas.

Bibliografía

(s.f.). https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html#sklearn.metrics.precision_recall_fscore_support

¿Qué es el reconocimiento de imágenes? (s.f.). https://la.mathworks.com/discovery/image-recognition-matlab.html

¿Qué es una red neuronal profunda? Más información sobre las redes neuronales profundas y cómo funciona el aprendizaje profundo | BotPress Blog. (s.f.). https://botpress.com/es/blog/deep-neural-network

3.1. Cross validation: evaluating estimator performance [[Accessed 24-01-2024]]. (s.f.).

Aaron, D. M. (2023, octubre). Lipomas. https://www.msdmanuals.com/es/hogar/trastornos-de-la-piel/crecimientos-cut%C3%A1neos-benignos/lipomas

ATICO34, G. (2019). Overfitting. Qué es, causas, consecuencias y cómo solucionarlo. https://protecciondatos-lopd.com/empresas/overfitting/

Baume, G. L. (2021). Breve introducción a Google Colab. http://fcaglp.unlp.edu.ar/~gbaume/grupo/Publicaciones/Apuntes/GoogleColab.pdf

Carmena-Ramón, R., Mateu-Puchades, A., Santos-Alarcón, S., & Lucas-Truyols, S. (2017). Queratosis actínica: nuevo concepto y actualización terapéutica. *Atención Primaria*. https://doi.org/10.1016/j.aprim.2017.01.004

Castañeda GP, E. T. (2016). El cáncer de piel, un problema actual. *Rev Fac Med UNAM*, *2*, 6-14.

CDC: Centros para el Control y la Prevención de Enfermedades. (2023). Cáncer de piel.

Das, T. (2023). Google Colab: todo lo que necesita saber. *Geekflare*. https://geekflare.com/es/google-colab/

De Cinfa, D. J. M. M. (2022, octubre). Léntigos solares. https://cinfasalud.cinfa.com/p/lentigos-solares/

Electronics, M. (s.f.). ¿Que es Raspberry Pi? [[Accessed 26-01-2024]].

EV, M., GA, R., & ÁM, V. (2011). Enfermedad de Bowen. *Rev Cent Dermatol Pascua*, *3*, 104-109.

Fonseca, K., Osorio, S., Castillo, J., & Fajardo, C. (2022). Contrastive learning for atrial fibrillation detection in challenging scenarios. *2022 30th European Signal Processing Conference (EUSIPCO)*, 1218-1222. https://doi.org/10.23919/EUSIPCO55093.2022.9909842

Fps, R. (2021). Dermatofibroma. *Fundación Piel Sana AEDV*. https://aedv.fundacionpielsana.es/wikiderma/dermatofibroma/

Guía avanzada de Inception V3. (s.f.). *Google Cloud*. https://cloud.google.com/tpu/docs/inception-v3-advanced?hl=es-419

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1512.03385

Hospital del Mar Barcelona. (2023). Nevus melanocíticos.

IBM documentation. (s.f.). https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model

Inbound_MAS. (2023, agosto). Lesiones vasculares en la piel: qué son y tratamiento. https://laserluz.com/blog/lesiones-vasculares

Jesús. (2022). Fundamentos de la clasificación de imágenes. *DataSmarts*. https://www.datasmarts.net/fundamentos-de-la-clasificacion-de-imagenes/

Jhoward. (2023). Which image models are best? *Kaggle*. https://www.kaggle.com/code/jhoward/which-image-models-are-best

Kanstren, T. (2023, agosto). A look at precision, recall, and F1-score. https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1412.6980

L, J. M., G, A. L. B., R, S. N., & A, M. M. (2010). Enfermedad de Bowen. *Rev Chilena Dermatol*, 280-283.

Lobos, B. P., & Lobos, S. A. (2011). Cáncer de piel no-melanoma. *Revista Médica Clínica Las Condes*, *22*(6), 737-748. https://doi.org/10.1016/s0716-8640(11)70486-2

Logitech. (s.f.). Camara web Logitech BRIO con video Ultra HD 4K y HDR [[Accessed 26-01-2024]].

Lucena, R. (2018). Lesiones vasculares: indentificación y tratamiento. *Clínica Ferran Sola*. https://www.clinicaferransola.com/lesiones-vasculares-indentificacion-y-tratamiento/

Manuel Olvera. (2023). Queratosis actínica vs Enfermedad de Bowen.

Morales, J. R. R. (2013). Enfermedad de Bowen. https://revactamedicacentro.sld.cu/index.php/amc/article/view/17/252

Queratosis seborreica - síntomas y causas - Mayo Clinic. (2022, enero). https://www.mayoclinic.org/es/diseases-conditions/seborrheic-keratosis/symptoms-causes/syc-20353878

ResNet-50: the basics and a quick tutorial. (2023, mayo). https://datagen.tech/guides/computer-vision/resnet-50/

Salama, K. (2020). Keras documentation: Supervised Contrastive Learning keras.io [[Accessed 25-01-2024]].

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1801.04381

Servicio de Dermatología del Hospital del Mar. (2022). Nevos melanocíticos.

Shaip-Admin. (2023). AI Image Recognition : Top 4 use cases and best practices. *Shaip*. https://es.shaip.com/blog/what-is-ai-image-recognition-and-how-does-it-work/

Skin Cancer ISIC. (2019, agosto). https://www.kaggle.com/datasets/nodoubttome/skin-cancer9-classesisic?resource=download

sklearn.modelselection.StratifiedKFold [[Accessed 24-01-2024]]. (s.f.).

Society, A. C. (2019). ¿Qué es el cáncer de piel tipo melanoma?

Society, A. C. (2023). Estadísticas importantes sobre el cáncer de piel tipo melanoma.

Svoboda, E. (2020). Artificial intelligence is improving the detection of lung cancer. *Nature*, *587*(7834), S20-S22. https://doi.org/10.1038/d41586-020-03157-9

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). Rethinking the inception architecture for computer vision. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1512.00567

Tan, M., & Le, Q. V. (2021). EfficientNetV2: Smaller models and faster training. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2104.00298

TensorFlow. (s.f.). TensorFlow. https://www.tensorflow.org/

Thonny, Python IDE for beginners. (s.f.). https://thonny.org/

Tschandl, P. (2018). *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions*. https://doi.org/10.7910/DVN/DBW86T

Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, *5*(1). https://doi.org/10.1038/sdata.2018.161

Vargas, M., Carballo, R. S., Bruno, G. K., Soto, D. C., & Angulo, V. (2021). Cáncer de piel: revisión bibliográfica. *Ciencia & Salud*, *5*(5). https://doi.org/10.34192/cienciaysalud.v5i5.347

Welcome to Python.org. (2023, octubre). https://www.python.org/