UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

MelCa: una aplicación para la clasificación del cáncer del melanoma.

Jianhao Wei

Ingeniería en Ciencias de la Computación

Trabajo de fin de carrera presentado como
requisito para la obtención del título de
Ingeniería en Ciencias de la Computación

Quito, 15 de Mayo de 2023

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

MelCa: una aplicación para la clasificación del cáncer delmelanoma.

Jianhao WEi

Nombre del profesor, Título académico **Noel Pérez Pérez, Ph.D**

Quito, 15 de Mayo de 2023

© DERECHOS DE AUTOR

Nombres y apellidos:	Jianhao Wei
Código:	00203027
Cédula de i dentidad:	1757823701
Lugar y fecha:	Quito, 15 de Mayo de 2023

ACLARACIÓN PARA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en http://bit.ly/COPETheses.

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on http://bit.ly/COPETheses.

RESUMEN

El melanoma es un cáncer agresivo de la piel que puede propagarse rápidamente a otras partes del cuerpo si no se diagnostica y trata de manera oportuna. Los métodos de diagnóstico actuales incluyen la evaluación visual, la biopsia y el análisis histopatológico, pero estos métodos pueden ser subjetivos y requerir tiempo y recursos significativos. En este trabajo, desarrollamos dos soluciones de aplicación, una aplicación nativa y una aplicación web, utilizando MobileNetV3 y transfer learning para abordar el problema de la clasificación del melanoma. La mejor arquitectura formada por MobileNetV3 Large junto con una capa GlobalAveragePooling2D, dos capas de Dropout y cuatro capas Dense se validó en un conjunto de datos experimental de HAM10000 (conjunto de datos de 10000 imágenes de entrenamiento para detectar lesiones cutáneas pigmentadas) utilizando un método de validación cruzada estratificado de 10 pliegues. Logró un promedio exitoso de área bajo la curva característica del receptor de 0.906 y 0.917 durante las fases de entrenamiento y prueba, respectivamente. El mejor modelo se implementó en las aplicaciones propuestas, y su viabilidad se validó midiendo el tiempo promedio y las métricas de consumo de batería en dos emuladores distintos. Las puntuaciones obtenidas de 8 y 7 segundos para los emuladores de teléfono y tableta, respectivamente, junto con los valores de consumo de batería de 0.32 y 0.21 mAh, se consideran satisfactorios en el desarrollo d eaplicaciones móviles. Al mismo tiempo, utilizamos las herramientas de desarrollo proporcionadas por el navegador Chrome para evaluar el rendimiento de la aplicación web. El tiempo de respuesta promedio para la clasificación del melanoma se midió en 115.4 4 milisegundos, con un consumo de tráfico promedio de 15 kilobytes. Estos resultados establecen las aplicaciones propuestas como una herramienta esencial para analizar imágenes de lesionescutáneas.

Palabras clave: Redes neuronales convolucionales, procesamiento de imágenes, clasificación de cáncer de melanoma, aplicación móvil, aplicación web, mobilenet V3.

ABSTRACT

Melanoma is an aggressive form of skin cancer that can rapidly metastasize if not diagnosed and treated promptly. Current diagnostic methods such as visual evaluation, biopsy, and histopathological analysis are subjective and resource-intensive. In this study, we developed two application solutions, a native app and a web app, using MobileNet V3 and transfer learning to address the challenge of melanoma classification. The best architecture, consisting of MobileNetV3 Large with a GlobalAveragePooling2D layer, two Dropout layers, and four Dense layers, was validated on an experimental dataset from HAM10000, which comprises 10,000 training images of pigmented skin lesions. Stratified 10-fold cross-validation was performed, resulting in a mean area under the receiver characteristic curve (AUC) score of 0.906 during the training phase and 0.917 during the test phase. The best model was implemented in the proposed apps, and its feasibility was assessed by measuring average execution time and battery consumption on two different emulators. The achieved execution times of 8 seconds for the phone emulator and 7 seconds for the tablet emulator, along with battery consumption values of 0.32 mAh and 0.21 mAh, respectively, are considered satisfactory for mobile application development. Additionally, the web app's performance was evaluated using development tools provided by the Chrome browser. The average response time for melanoma classification was measured at 115.44 milliseconds, with an average network traffic consumption of 15 kilobytes per image. These results establish the proposed applications as essential tools for analyzing skin lesion images.

Keywords: Convolutional neural networks, image processing, melanoma, cancer classification, mobile application, web application, MobileNet V3.

TABLA DE CONTENIDO

1. INTRODUCCIÓN 10
2. MATERIALES Y MÉTODOS 12
2.1 Base de datos 12
2.2 Modelos de aprendizaje profundo 13
2.3 Modelos propuestos para la clasificación del melanoma 14
2.4 Aplicación MelCa propuesta 18
2.4.1 Aplicación Nativa 18
2.4.2 Aplicación Web 20
2.5 Configuración experimental 22
2.5.1 Procesamiento de datos 22
2.5.2 Training, validation, and test sets 22
2.5.3 Configuración e implementación del modelo 22
2.5.4 Métricas de evaluación 23
2.5.5 Criterio de selección 23
2.5.6 Dispositivos móviles experimentales 24
2.5.7 Web app experimentale 24
2.5.8 Protocolo de medida de energía 25
3. RESULTADOS Y DISCUSIÓN 25
3.1 Evaluación del rendimiento en el conjunto de entrenamiento. 25
3.2 Evaluación del rendimiento en el conjunto de PRUEBA. 28
3.3 Viabilidad de la aplicación MelCa 29
4. CONCLUSIONES Y TRABAJO FUTURO 29
5. REFERENCIAS BIBLIOGRÁFICAS 31

ÍNDICE DE TABLAS

Tabla 1: Arquitectura de MobilenetV3 Large..... ..15

Tabla 2. Arquitectura de MobilenetV3 Small..... .16

Tabla 3: Resumen de emuladores de dispositivos móviles experimentales..... .24

Tabla 4: Resultados de rendimiento de diferentes arquitecturas..... 26

ÍNDICE DE FIGURAS

Figura 1: Transferencia de aprendizaje del modelo basado en MobileNetV3 y esquema 1 de fine-tuning.....17

Figura 2: Transferencia de aprendizaje del modelo basado en MobileNetV3 y esquema 2 de fine-tuning.....18

Figura 3: Flujo de trabajo de la aplicación MelCa propuesta implementada en dispositivos móviles.....20

Figura 4: Flujo de trabajo de la aplicación MelCa propuesta implementada en web -app.....21

Figura 5: Mean of Loss vs. Epochs performance.....28

Figura 6: ROC-AUC Curve.....28

Figura 7: Precision vs. Recall Curve.....28

1. INTRODUCCIÓN

El melanoma es un tipo de cáncer de piel relativamente raro pero grave. Según la Organización Mundial de la Salud, se estima que cada año se producirán aproximadamente 132,000 casos de melanoma (World Health Organization, 2019). Actualmente, la Sociedad Americana del Cáncer estima que habrá 97,610 nuevos casos de melanoma en 2023, y 7,990 personas morirán a causa de este tipo de cáncer de piel (American Cancer Society, 2023). E l melanoma puede afectar a cualquier persona, pero los caucásicos tienen un mayor riesgo que otras razas debido a que su piel es más sensible a los rayos ultravioleta (World Health Organization, 2019). Por lo tanto, el melanoma es un área de investigación constante en términos de su cuidado y prevención.

Varios métodos se han propuesto para detectar el cáncer de piel de forma automática. La mayoría de ellos incluyen etapas de segmentación de lesiones y limpieza de ruido como una tarea previa a la clasificación debido al ruido (artefactos) presente en las imágenes dermatoscópicas. Por ejemplo, Patil y Bellary propusieron un enfoque de aprendizaje conjunto para detectar tipos de melanoma. En este contexto, se ha demostrado que un nuevo método de eliminación de artefactos combinado con segmentación por Expectation Maximization ofrece resultados prometedores (Patil & Bellary, 2021). La segmentación de lesiones cutáneas en imágenes dermatoscópicas es esencial para el diagnóstico asistido por computadora del melanoma. El desarrollo de una arquitectura de red más profunda con núcleos más pequeños e incluyendo información de color de múltiples espacios de color puede mejorar el rendimiento de la segmentación. El método logró un Índice de Jaccard promedio de 0.765 y se ubicó en el primer lugar entre las 21 presentaciones finales en el desafío de segmentación de lesiones cutáneas del ISBI (International Symposium on Biomedical Imaging) 2017 (Yuan & Lo, 2019). Los algoritmos de aprendizaje profundo muestran

grandes promesas en la clasificación del melanoma para el diagnóstico del cáncerde piel. Las funciones de activación desempeñan u n papel crucial en el rendimiento de las redes neuronales profundas. En otro estudio, Namozov y Cho propusieron un algoritmo de red neuronal convolucional con una función de activación parametrizada que supera a otras funciones de activación en la clasificación del melanoma (Namozov & Cho, 2018).

El melanoma es un cáncer de piel altamente agresivo, y la detección temprana es crucial. Los algoritmos de aprendizaje profundo como las redes neuronales convolucionales (CNN) han mostrado promesa en la clasificación de enfermedades basadas en imágenes. En un estudio previo (Rokhana, Herulambang, & Indraswari, 2020), se propuso una CNN profunda para la clasificación del melanoma, logrando una precisión del 84.76%, una sensibilidad del 91.97% y una especificidad del 78.71% en 352 imágenes de prueba. El modelo construido tiene potencial para aplicaciones en el mundo real que ayuden a los expertos en el diagnóstico y tratamiento.

En los últimos años, la mayoría de los enfoques han aplicado CNN entrenadas en imágenes de rmatoscópicas. Sin embargo, como se menciona en otro estudio (Feng, Berk - Krauss, Feng, & Stein, 2018), los dermatólogos son escasos en regiones en desarrollo, y este hallazgo no es una excepción en países emergentes como Ecuador. Sin embargo, debido a la popularidad de los teléfonos inteligentes, existe una oportunidad para posibles soluciones a este fenómeno. Por ejemplo, en Ecuador, hay 15.9 millones de teléfonos móviles conectados, lo que representa el 88% de su población (Medina, 2021). Por lo tanto, de sarrollar una aplicación móvil para clasificar lesiones cutáneas como melanoma o no podría ser útil y una forma fácil de auto diagnóstico.

En un estudio reciente (Dai, Spasić, Meyer, Chapman, & Andres, 2019), se propuso una aplicación iOS para inferencia e n el dispositivo que utiliza un modelo de CNN pre-entrenado con 10,015 imágenes de cáncer de piel. Sin embargo, esta aplicación solo se ha

desarrollado para el sistema operativo iOS, lo que dificulta su implementación específica en áreas empobrecidas o rem otas. Por otro lado, en otro estudio (Sadiq, Sankalpa, Ahfid, Sagahyroon, & Dhou, 2020), se enfocaron más en desarrollar una aplicación Android para la detección de melanoma utilizando una máquina de vectores de soporte (SVM), que utiliza el conjunto de datos ISIC (International Skin Imaging Collaboration) de 2019. Decidieron utilizar un servidor para realizar las tareas principales del modelo, como la detección de lesiones. Sin embargo, realizar cualquier tarea fuera del teléfono compromete la detección en tiempo real.

En este trabajo, nuestro objetivo es implementar un protocolo automático basado en técnicas de aprendizaje profundo para clasificar el cáncer de melanoma en dispositivos de bajo costo a través de una aplicación móvil. La principal contribució n detrás de este objetivo es utilizar transfer learning (aprendizaje por transferencia) desde un detector pre-entrenado sólido hacia nuestro protocolo de clasificación de bajo costo.

2. MATERIALES Y MÉTODOS

2.1 Base de datos

El HAM10000 es un conjunto de datos abierto que contiene 10,015 imágenes de casos de enfermedades de la piel proporcionadas por la Colaboración Internacional de Imágenes de la Piel (ISIC) (Tschandl, 2018). Este conjunto de datos incluye una variedad de enfermedades cutáneas comunes, como nevus , melanoma, carcinoma de células basales y queratosis actínica. Este conjunto de datos puede ser utilizado para impulsar el desarrollo e investigación de tecnología de aprendizaje automático e inteligencia artificial para mejorar la precisión del diagnósti co automatizado de enfermedades de la piel y mejorar las capacidadesde diagnóstico de los profesionales médicos en dermatología.

Dentro de este conjunto de datos, se han identificado 1,113 imágenes como casos de melanoma. Aunque el número de casos de mela noma es bajo en comparación con otras lesiones cutáneas, generar un conjunto de datos equilibrado ayudaráa asegurar que el modelo no estésesgado hacia las lesiones no melanoma, por lo que el conjunto de datos final contiene 2,226 imágenes, con melanoma y no melanoma representados por igual.

2.2 Modelos de aprendizaje profundo

El Aprendizaje Profundo es un subcampo del aprendizaje automático que utiliza redes neuronales artificiales con múltiples capas para modelar y resolver problemas complejos. La arquitectura de las redes neuronales profundas estáinspirada en la estructura y funcionamiento del cerebro humano, y las capas de la red se organizan jerárquicamente para procesar y extraer características de los datos de entrada. Los algoritmos de aprend izaje profundo se han utilizado en diversas aplicaciones, incluyendo visión por computadora (Charleen et al., 2021), procesamiento del lenguaje natural (Gupta et al., 2020), robótica (Guarneri et al., 2021) y biología computacional (Jiang et al., 2022). A diferencia del aprendizaje automático convencional, no requiere la extracción manual de características, lo que ahorra tiempo y esfuerzo. En cambio, el algoritmo aprende automáticamente las características relevantes de los datos. Algunas de las redes neur onales profundas más populares incluyen las CNN (redes neuronales convolucionales), RNN (redes neuronales recurrentes) y GAN (redes generativas adversariales). Las CNN generalmente consisten en capas convolucionales, capas de agrupación (o submuestreo) y u na capa completamente conectada al final. La capa convolucional consta de filtros aplicados a todas las áreas de entrada. La capa de agrupación se encarga de reducir el tamaño de las características del modelo para lograr una mayor eficiencia computacional (Raschka & Mirjalili, 2017). Sin embargo, los recursos informáticos y los pesos requeridos por las CNN profundas

tradicionales o sus variantes no son adecuados para su implementación en escenarios con recursos limitados, como dispositivos móviles. Por lo tanto, los modelos de redes neuronales convolucionales livianas, como MobileNet, se han vuelto indispensables para completar este tipo de proyectos.

MobileNet: es una arquitectura de CNN diseñada para dispositivos móviles y embebidos, desarrollada por Google en 2017 (Howard et al., 2017). Estáoptimizada para plataformas móviles con recursos computacionales limitados, como teléfonos inteligentes y dispositivos de IoT (Internet de las cosas), al reducir el número de parámetros y operaciones requeridas para la inferencia. En 2019, se optimizó MobileNetV3 para las CPUs de teléfonos móviles utilizando una búsqueda de arquitectura de red consciente del hardware y avances novedosos en el diseño de arquitectura. Incluye dos nuevos modelos, MobileNetV3 -Large y MobileNetV3-Small, dirigidos a casos de uso de alta y baja capacidad de recursos, que logran resultados de vanguardia en tareas de clasificación, detección y segmentación móvil (Howard et al., 2019).

2.3 Modelos propuestos para la clasificación del melanoma

Adoptamos tanto las redes neuronales MobileNetV3 Small como MobileNetV3 Large para construir el método propuesto, el cual añade diferentes capas para aplicar técnicas de fine-tuning, resultando en la siguiente arquitectura: MobileNetV3 Small + esquema 1 (fine-tuning 1), MobileNetV3 Small + esquema 2 (fine-tuning 2), MobileNetV3 Large + esquema 1 (fine-tuning 1), MobileNetV3 Large + esquema 2 (fine-tuning 2).

Para una mejor explicación del método propuesto, primero explicaremos la diferencia entre MobileNetV3 Small y MobileNetV3 Large. Según la Tabla.1 y Tabla.2 , podemos observar que la diferencia principal entre las dos redes neuronales es el número de "bnecks".

En comparación con la versión Large, MobileNetV3 Small tiene menos parámetros y una estructura de red más simple. Por lo general, MobileNetV3 Large tiene una precisión más alta. Aunque la versión Small puede sacrificar algo de precisión, sigue siendo eficiente computacionalmente y rápida.

Input	Operator	exp size	#out	SE	NL	s
224^2 x3	conv2d	-	16	-	HS	2
112^2x 16	bneck, 3x3	16	16	-	RE	1
112^2x 16	bneck, 3x3	64	24	-	RE	2
56^2 x 24	bneck, 3x3	72	24	-	RE	1
56^2 x 24	bneck, 5x5	72	40	V	RE	2
28^2 x 40	bneck, 5x5	120	40	V	RE	1
28^2 x 40	bneck, 5x5	120	40	V	RE	1
28^2 x 40	bneck, 3x3	240	80	-	HS	2
14^2 x 80	bneck, 3x3	200	80	-	HS	1
14^2 x 80	bneck, 3x3	184	80	-	HS	1
14^2 x80	bneck, 3x3	184	80	-	HS	1
14^2 x80	bneck, 3x3	480	112	V	HS	1
14^2 x 112	bneck, 3x3	672	112	V	HS	1
14^2 x 112	bneck, 5x5	672	160	V	HS	2
7^2x 160	bneck, 5x5	960	160	V	HS	1
7^2 x 160	bneck, 5x5	960	160	V	HS	1
7^2 x 160	conv2d, 1xl	-	960	-	HS	1
7^2 x 960	pool, 7x7	-	-	-	-	1
1^2 x 960	conv2d 1x1, NBN	-	1280	-	HS	1
1^2 x 1280	conv2d 1x1, NBN	-	k	-	-	1

Tabla 1: Arquitectura de MobilenetV3 Large, SE indica si hay un bloque Squeeze-And-Excite en ese bloque. NL indica el tipo de no linealidad utilizado. HS indica h-swish y RE indica ReLU. NBN indica no se utiliza normalización por lotes. s indica el paso o stride.

Input	Operator	exp size	#out	SE	NL	s
224^2 x3	conv2d	-	16	-	HS	2
112^2x 16	bneck, 3x3	16	16	V	RE	2
56^2 x 16	bneck, 3x3	72	24	-	RE	2
28^2 x 24	bneck, 3x3	88	24	-	RE	1
28^2 x 24	bneck, 5x5	96	40	V	HS	2
14^2 x 40	bneck, 5x5	240	40	V	HS	1
14^2 x 40	bneck, 5x5	240	40	V	HS	1

14^2 x 40	bneck, 5x5	120	48	V	HS	1
224^2 x3	conv2d	-	16	-	HS	2
112^2x 16	bneck, 3x3	16	16	V	RE	2
14^2 x 48	bneck, 5x5	144	48	V	HS	1
14^2 x 48	bneck, 5x5	288	96	V	HS	2
7^2x 96	bneck, 5x5	576	96	V	HS	1
7^2 x 96	bneck, 5x5	576	96	V	HS	1
7^2 x 96	conv2d, 1xl	-	576	V	HS	1
7^2 x 576	pool, 7x7	-	-	-	-	1
1^2 x 576	conv2d 1x1, NBN	-	1280	-	HS	1
1^2 x 1280	conv2d 1x1, NBN	-	k	-	-	1

Tabla 2: Arquitectura de MobilenetV3 Small. Consulta la tabla 1 para la notación.

En cuanto al fine-tuning, proponemos dos esquemas y a continuación se describe detalladamente cada uno de ellos:

Esquema 1: El esquema 1 implica agregar una capa de GlobalAveragePooling2D para comprimir la dimensión espacial del mapa de características de entrada en un vector de longitud fija. Esta adición resulta especialmente beneficiosa al utilizar MobileNetV3 Sma ll y MobileNetV3 Large como modelos base. Reduce eficazmente el número de parámetros y la complejidad computacional al tiempo que preserva información esencial de las características. A continuación, se agrega una capa completamente conectada con un solo nodo de salida. Para controlar la complejidad del modelo y mejorar su capacidad de generalización para predicciones de clasificación binaria, se aplica una función de activación sigmoide y una regularización L2 a esta capa. Este esquema se puede combinar co n MobileNetV3 Large y Small para generar nuestra arquitectura mencionada, como se muestra en la Figura 1.

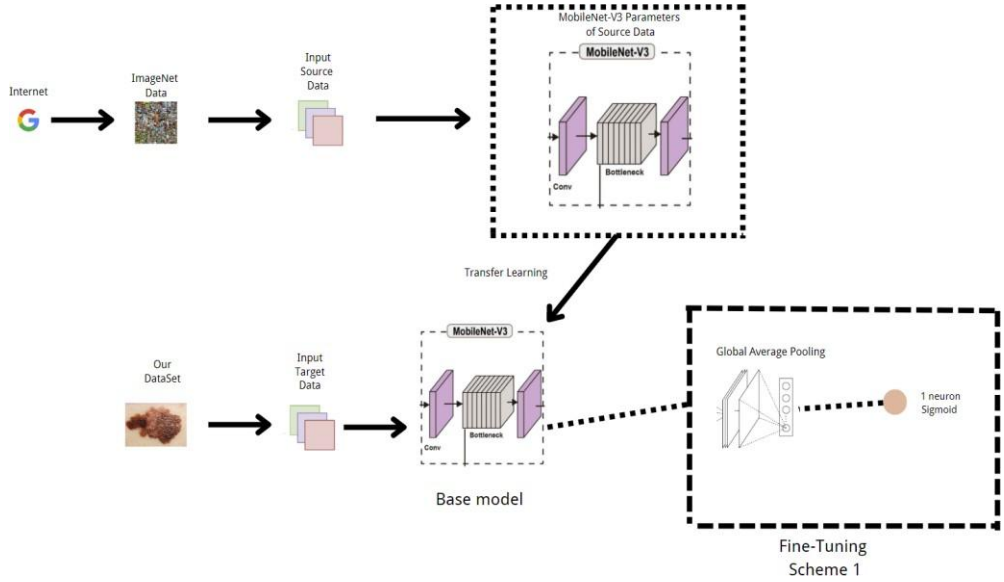

Figura 1: Transferencia de aprendizaje del modelo basado en MobileNetV3 y esquema 1 de fine-tuning.

Esquema 2: Sobre la base del esquema 1 de fine-tuning, creamos e insertamos cinco capas adicionales entre GlobalAveragePooling2D y la última capa Dense para construir nuestro esquema 2 de fine-tuning. Estas capas consisten en dos capas de Dropout con una probabilidad del 0.2 para evitar el sobreajuste, y tres capas Dense completamente conectadas con 512, 256 y 128 neuronas, respectivamente. Cada una de las capas Dense utiliza la función de activación Relu y se somete a una regularización L2. La Figura 2 muestra la combinación de nuestro modelo base (MobileNetV3) con el esquema 2 de fine-tuning.

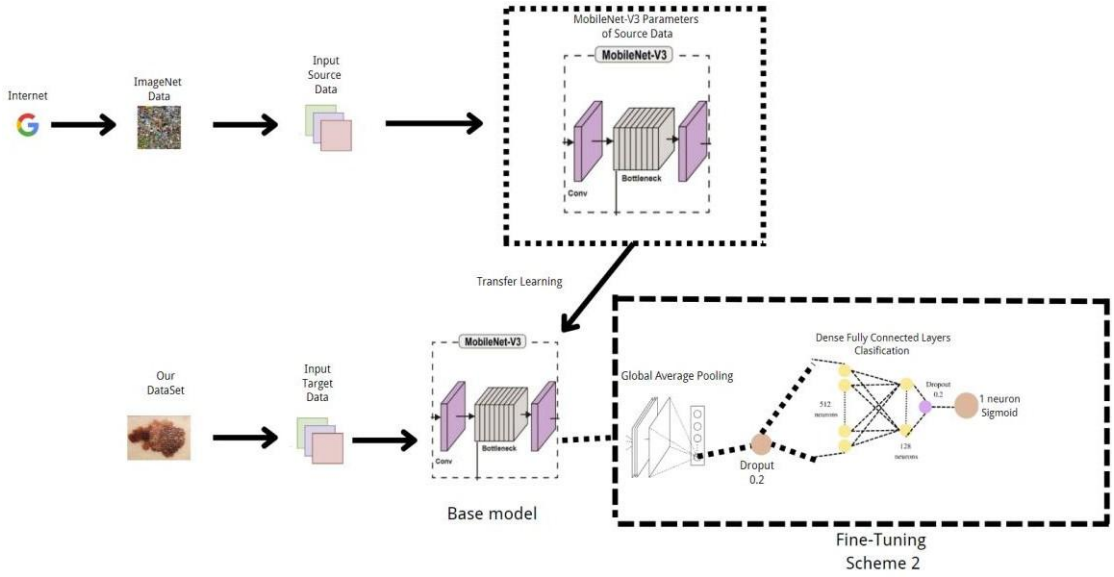

Figura 2: Transferencia de aprendizaje del modelo basado en MobileNetV3 y esquema 2 de fine-tuning.

2.4 Aplicación MelCa propuesta

2.4.1 Aplicación Nativa

Para lograr este objetivo, la aplicación debe obtener permisos del sistema al llamar a la cámara del dispositivo para tomar una foto o al acceder a la galería. Se deben solicitar los permisos correspondientes en el archivo AndroidManifest.xml y en el código que activa el botón "takePhoto". Además, se requiere un archivo .../res/xml/provider_paths.xml para ayudar a configurar la aplicación, el cual se debe declarar en AndroidManifest.xml. Luego, se usa un evento de Intent para iniciar la cámara y abrir la galería.

Una vez que se obtiene la imagen, es necesario guardarla como un objeto de clase Bitmap para su procesamiento previo. Para lograr la integración del modelo de TensorFlow, se deben preparar tres archivos: 1. Un modelo de red neuronal ya entrenado y guardado en formato .pb; 2. Un archivo jar de libandroid_tensorflow_inference_java.jar que estéen línea con la versión que hemos generado o que ha sido generada por alguien más; 3. Una librería de clase libtensorflow_inference.so. A continuación, se deben colocar estos tres archivos en

las carpetas correspondientes del proyecto.

Para obtener los resultados de predicción, se ha diseñado una función llamada "predict()". La imagen, después de convertirse en bitmap, se utiliza como parámetro de entrada de la función "predict()". En primer lugar, el bitmap se somet e a una preprocesamiento, se redimensiona a un tamaño de 224 x 224, y luego entra en el bucle "for" para normalizar y comprimir la matriz de píxeles de dos dimensiones en un array unidimensional llamado "pixels[]". A continuación, se introduce el bitmap pr eprocesado en el modelo de predicción mediante el método "inferenceInterface.fillNodeFloat(INPUT_NODE, INPUT_SIZE, pixels)", y luego se realiza la predicción y salida de datos a través del método "inferenceInterface.runInference(new String[] {OUTPUT_NODE})". El parámetro que se le proporciona a este método es la forma y el tamaño de los datos de salida. Finalmente, el resultado de la predicción se almacena en el array "prediction[]" a través del método "inferenceInterface.readNodeFloat(OUTPUT_NODE, prediction)". Se realiza una evaluación textual de los valores de "prediction[]" para determinar los resultados de la predicción. La Figura 3 muestra el flujo básico de diseño de la aplicación.

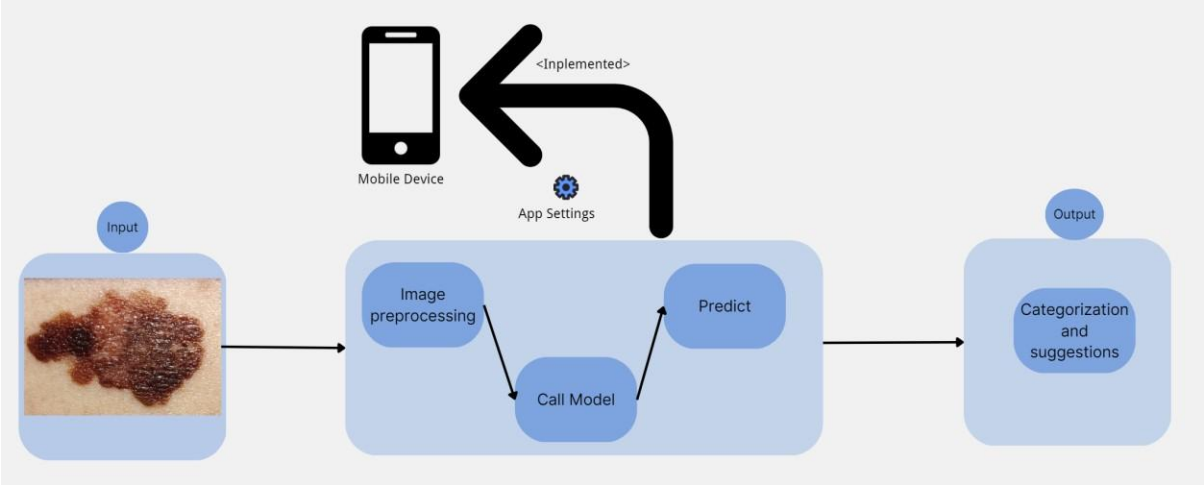

Figura 3: Flujo de trabajo de la aplicación MelCa propuesta implementada en dispositivos móviles.

2.4.2 Aplicación Web

Además, teniendo en cuenta que algunos dispositivos móviles son antiguos, la aplicación web también puede ser considerada como otra opción de desarrollo de aplicaciones. Las ventajas de la aplicación web en comp aración con la aplicación nativa son: En cuanto a la plataforma de soporte, las aplicaciones web pueden ejecutarse en cualquier dispositivo que admita un navegador web, mientras que las aplicaciones nativas deben desarrollarse y desplegarse en función de l os sistemas operativos específicos como iOS o Android, y solo pueden ejecutarse en dispositivos particulares.

Se utiliza el marco de trabajo Flask de Python. La aplicación permite a los usuarios cargar imágenes o especificar URL de imágenes para determina r si una lesión en la piel es benigna o maligna. Esta aplicación utiliza una red neuronal pre -entrenada de TensorFlow para realizar la clasificación. Las imágenes son preprocesadas, redimensionadas y luego convertidas a un formato base64 para su uso en la interfaz de usuario. La aplicación también incluye una función para decodificar las imágenes base64 en formato PNG para su procesamiento en la red neuronal.

La aplicación cuenta con tres rutas principales. La primera es la ruta de inicio (base.html), que permite a los usuarios cargar una imagen o especificar una URL de imagen para su procesamiento. La segunda ruta (index.html) se utiliza para procesar las imág enes cargadas por los usuarios, mientras que la tercera ruta procesa las imágenes a partir de las URL especificadas por los usuarios. La Figura 4 muestra el flujo de trabajo de la Web-app.

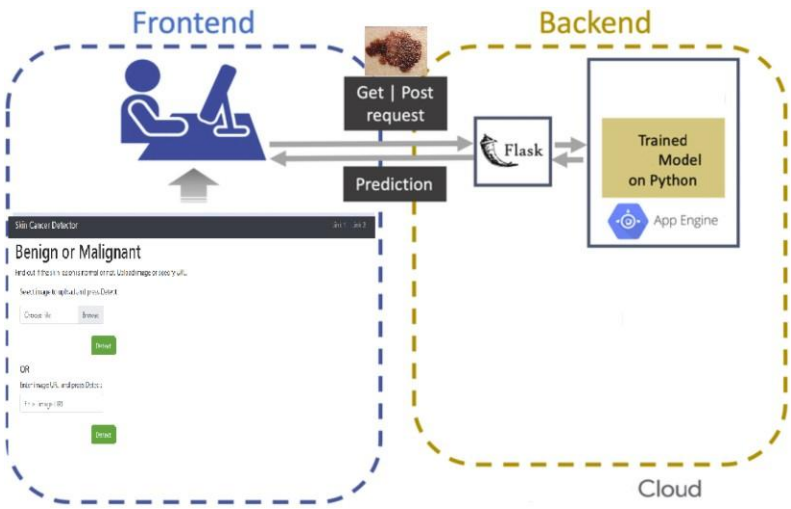

Figura 4: Flujo de trabajo de la aplicación M elCa propuesta implementada en web-app.

Ambos esquemas se proporciona recomendaciones médicas adecuadas según los resultados de la predicción es muy importante para garantizar la salud del usuario. Por ejemplo, si los resultados de la predicción muestran síntomas graves de una enfermedad de la piel, puede ser necesario recomendar al usuario que busque ayuda médica lo antes posible. Si los resultados de la predicción son benignos, se puede recomendar al usuario que se someta a revisiones periódicas para asegurar la salud de su piel. Independientemente de los resultados de la predicción, se debe recomendar al usuario que busque asesoramiento de profesionales médicos para asegurar un diagnóstico y tratamiento adecuados.

2.5 Configuración experimental

2.5.1 Procesamiento de datos

Las imágenes se redimensionan a un tamaño objetivo de 224x22 4, y en esta etapa se utiliza la clase ImageDataGenerator de Keras para generar imágenes aumentadas. Se especifican varios parámetros de aumento para definir las transformaciones aplicadas durante el entrenamiento. Estos parámetros incluyen rango de rotaci ón, cambios de ancho y alto, rango de cizallamiento, rango de zoom, volteo horizontal y modo de relleno para los píxeles recién creados. La razón por la cual se realiza esto es que la investigación muestra los beneficios de la aumentación de datos (Pham et al., 2018).

2.5.2 Training, validation, and test sets

El 90% del conjunto de datos se utilizarápara un esquema de validación cruzada estratificada de 10 pliegues. Esta técnica evalúa de manera efectiva el rendimiento del modelo en diferentes divisiones de datos, al tiempo que garantiza que se mantenga la proporción entre melanoma y no melanoma, evitando asíel sobreajuste (Muralidhar, 2021). El 10% restante de los datos se utilizarácomo conjunto de prueba para observar la capacidad de generalización del model o.

2.5.3 Configuración e implementación del modelo

Durante el proceso de entrenamiento, los hiperparámetros del tamaño del lote (batch size) y las épocas se establecieron en 128 y 1000, respectivamente, y se aplicaron a todos los modelos. Cabe destacar que se utilizó una técnica de detención temprana (early stopping) para prevenir el sobreajuste del modelo (Li & Tan, 2009). La tasa de aprendizaje se inicializó en 1 × 10^-4, y se utilizó una tasa de aprendizaje de 1 × 10^-5 para la arquitectura MobileNetV3 Large+ con el esquema 2 (fine-tuning 2) para verificar si se obtienen mejores resultados. En la arquitectura MobileNetV3 Large+ con el esquema 2 (fine-tuning 2), se intentó descongelar o congelar los pesos preentrenados para observar si se podrían

lograr mejores resultados, mientras que en otras arquitecturas se mantuvieron congelados los pesos preentrenados. Se utilizó el optimizador Adam, que almacena de forma adaptativa el promedio decaído exponencialmente de los gradientes anteriores, similar al momento (Giordano,2018). El entrenamiento se llevó a cabo en la plataforma Colab desarrollada por Google, utilizando el lenguaje de programación Python.

2.5.4 Métricas de evaluación

La evaluación del rendimiento del modelo se realizaráutilizando las siguientes métricas: p recisión (ACC), recall(REC), precisión (PRE), área bajo la curva (AUC), F1 - score y pérdida. Además, se generarán la curva ROC (característica operativa del receptor), la curva de precisión - recall y la curva de pérdida en función de las épocas para evaluar los cambios en el rendimiento del modelo a lo largo del tiempo. El ROC-AUC se considera el indicador más importante entre todas las métricas, ya que representa la capacidad del modelo para distinguir entre dos clases (Narkhede,2021).

2.5.5 Criterio de selección

El modelo fue seleccionado finalmente en base a los siguientes criterios: (1) El modelo con el puntaje ROC-AUC más alto, con significancia estadística por arquitectura; (2) En caso de tener un rendimiento similar en ROC-AUC, se prefirió el modelo con meno s épocas de entrenamiento. Esto implica que se consumieron menos recursos computacionales durante el entrenamiento; y (3) Si, después de las dos rondas anteriores de comparación, los resultados seguían siendo similares, se elegía el modelo con menos paráme tros. Esto indica que cada parámetro en el modelo desempeñaba un pape significativo. También significa que el modelo resultante era ligero, lo que lo hace más adecuado para implementarse en dispositivos móviles.

2.5.6 Dispositivos móviles experimentales

Se eligieron dos emuladores de dispositivos móviles con especificaciones más bajas para instalar y probar la aplicación nativa, considerando que podrían cumplir con los requisitos de la aplicación incluso con un rendimiento más bajo. Las especificaciones de los dispositivos seleccionados se resumen en la Tabla 3.

Dispositivo	Especificación	Valor
Tablet	Model	Pixel C.
	CPU/ABI	Google Play intel Atom (x86)
	API	28 (Android 9.0 Pie)
	Target	google_apis [Google APIs] (API level 28)
	Skin	pixel_c
	Resolution	2560x1800 xhdpi
	Camera	Host Camera (Acer Nitro 7 Laptop)
	Internal Storage	2048 MB
	RAM	2048 MB
	VM heap	127 MB
Phone	Model	Pixel 2.
	CPU/ABI	Googel Play Intel Atom (x86)
	API	29 (Android 10.0 Q)
	Target	google_apis_playstore[Google Play] (API level 29)
	Skin	pixel_2
	Resolution	1080x1920 xxhdpi
	Camera	Host Camera (Acer Nitro 7 Laptop)
	Internal Storage	6144 MB
	RAM	1536 MB
	VM heap	256 MB

Tabla 3: Resumen de emuladores de dispositivos móviles experimentales.

2.5.7 Web app experimentale

Para realizar pruebas de rendimiento de una aplicación web, la mayoría de los navegadores modernos, como Chrome, Firefox y Safari, proporcionan herramientas de desarrollo incorporadas que se pueden utilizar para analizar las solicitudes de red y los tiempos de respuesta. Estas herramientas permiten a los desarrolladores evaluar el rendimiento de sus aplicaciones web.

2.5.8 Protocolo de medida de energía

Los registros de estadísticas de la batería y los registros de procesos nos ayudan a comprender el rendimiento de la aplicación según el servicio de Android Debug Bridge (herramienta de línea de comandos adb)

3. RESULTADOS Y DISCUSIÓN

3.1 Evaluación del rendimiento en el conjunto de entrenamiento.

Según la configuración experimental, se evaluó el rendimiento de diferentes modelos en un conjunto de datos de 2006 imágenes de entrenamiento. Los resultados obtenidos basados en las métricas ROC -AUC, pérdida de entropía cruzada binaria y precisión se muestran en la Tabla 4.

Modelo base	Tasa de aprendizaje	Esquema Fine-tuning 1 o 2	Congelar sío no	Batch Size	Epocas (u)	AUC (u)	Loss B.C (u)	ACC (%)	PRE (u)	REC (u)	F1-score (u)
Mobilenet V3 Large	**0.0001**	**2**	**Sí**	**128**	**113**	**0.906 ± 0.02**	**1.188 ± 0.13**	**84.8 ± 0.02**	**0.926 ± 0.02**	**0.765 ± 0.02**	**0.814 ± 0.03**
Mobilenet V3 Large	0.0001	2	No	128	75	0.878 ± 0.03	1.982 ± 0.62	76.8 ± 0.02	0.985 ± 0.02	0.557 ± 0.02	0.800 ± 0.03
Mobilenet V3 Large	0.00001	2	Sí	128	768	0.898 ± 0.04	2.152 ± 0.80	82.0 ± 0.02	0.916 ± 0.02	0.783 ± 0.02	0.811 ± 0.03
Mobilenet V3 Large	0.0001	1	Sí	128	81	0.889 ± 0.02	0.508 ± 0.03	83.0 ± 0.02	0.835 ± 0.02	0.835 ± 0.02	0.802 ± 0.03
Mobilenet V3 Small	0.0001	1	Sí	128	181	0.873 ± 0.02	0.525 ± 0.04	78.6 ± 0.02	0.791 ± 0.02	0.791 ± 0.02	0.802 ± 0.07
Mobilenet V3 Small	0.0001	2	Sí	128	143	0.894 ± 0.02	0.845 ± 0.07	83.0 ± 0.02	0.777 ± 0.02	0.919 ± 0.02	0.802 ± 0.05

Tabla 4: Resultados de rendimiento de diferentes arquitecturas.

De la Tabla 4, se puede observar que MobileNetV3 Small tiene un puntaje AUC más bajo en comparación con MobileNetV3 Large con los mismos parámetros en la mayoría de los casos. Por lo tanto, aunque tiene un tamaño de parámetro y un tamaño de modelo ligeramente más pequeños en comparación con MobileNetV3 Large, su rendimiento no es suficiente para una clasificación binaria de alta precisión. En Mobilenetv3 large+ esquema 2 (fine tuning 2), una tasa de aprendizaje de 0.0001 tiene un mejor rendimiento que 0.00001.

Esto se debe a que con una tasa de aprendizaje de 0.00001, la velocidad de convergencia es demasiado lenta, lo que resulta en una pérdida promedio mucho más alta durante el proceso de entrenamiento. Por otro lado, después de descongelar los pesos pre -entrenados, las épocas de entrenamiento para Mobilenetv3 Large+ con esquema 2 (fine tuning 2) se volvieron significativamente mayores, pero no mostraron una mejora en el rendimiento.

Además, al comparar el rendimiento de MobileNetV3 Large con diferentes esq uemas de ajuste fino, el mejor rendimiento se obtiene con Mobilenetv3 large+ esquema 2, con una tasa de aprendizaje de 0.0001 y los pesos pre-entrenados congelados (ver Tabla 4, línea en negrita).

En cuanto al entrenamiento de la mejor arquitectura, el valor de pérdida disminuye gradualmente en forma de escalera, lo que indica la efectividad del transfer learning, como se muestra en la Figura 5. Como el modelo ya ha convergido en los datos originales, se recomienda utilizar una tasa de aprendizaje pequeña, aproximadamente 10^(-4), para el ajuste fino de los nuevos datos. En este caso, el valor de 0.0001 se considera una tasa de aprendizaje adecuada. Además, la detención temprana permite que el modelo detenga el entrenamiento antes de que ocurra el sobreajuste en el conjunto de validación. La curva promedio de ROC - AUC del modelo seleccionado después de diez iteraciones de entrenamiento es 0.906, con un buen rendimiento en diferentes conjuntos de datos, como se muestra en la Figura 6. Además, el equilibrio entre la precisión y la recall para el mejor modelo resalta resultados exitosos. La Figura 7 muestra el equilibrio entre la precisión y la recall durante el entrenamiento del modelo. El modelo óptimo logra puntuaciones de

precisión y recall de 0.926 y 0.765, respectivamente, lo que indica su destacada capacidad para distinguir entre casos demelanoma y casos no melanoma.

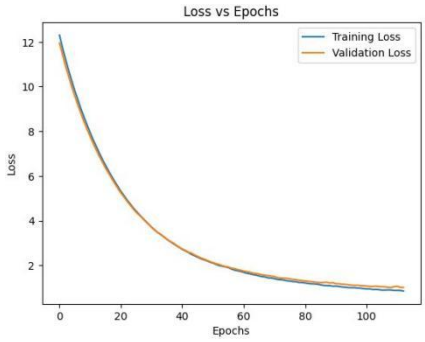

Figura 5: Mean of Loss vs. Epochs performance.

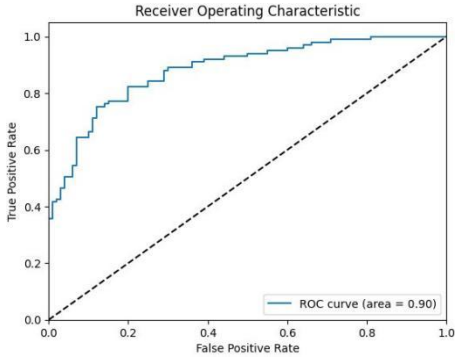

Figura 6: ROC-AUC Curve

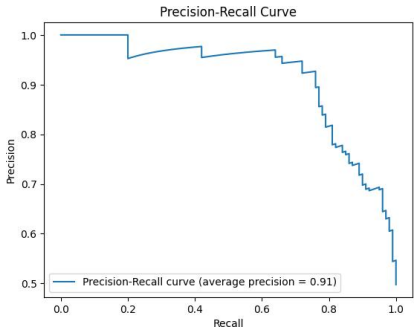

Figura 7: Precision vs. Recall Curve

3.2 Evaluación del rendimiento en el conjunto de PRUEBA.

Durante la fase de pruebas, se seleccionó el modelo óptimo con 113 épocas de MobileNet V3 Large + (fine tuning 2) basado en sus puntuaciones en las métricas descritas.

Para evaluar su capacidad de generalización, se utilizó un conjunto de pruebas que contenía imágenes de lesiones cutáneas clasificadas como no melanoma y melanoma en proporciones iguales. Se obtuvieron resultados satisfactorios utilizando este conjunto de datos, con puntuaciones de ACC, REC, PRE y AUC de 77.8%, 0.934, 0.943 y 0.917, respectivamente. Estos resultados demuestran que el modelo es competitivo y tiene excelentes capacidades de generalización.

3.3 Viabilidad de la aplicación MelCa

La viabilidad de la solución de la aplicación nativa se validó preliminarmente midiendo el tiempo de ejecución promedio y el consumo de batería de dos dispositivos simulados. El proceso involucró el uso del comando " adb shell dumpsys batterystats" y buscar "Estimated power use (mAh)" para observar el consumo de energía de la aplicación durante las pruebas. Se encontró que ambos dispositivos pudieron mantener el tiempo de ejecución del diagnóstico dentro de los 8 segun dos, con un consumo promedio de energía de alrededor de 0.45mAh por imagen. En cuanto a la evaluación del rendimiento de la aplicación web, se midió observando el tiempo de espera en la sección de red del navegador Google. El tiempo de respuesta promedio de la aplicación fue de 115.44ms, con un promedio de 15kb transferidos a través de la red por imagen. Estos resultados demuestran que el diseño ligero del modelo propuesto de clasificación de melanoma permite la utilización eficiente de redes neuronales profundas en entornos con recursos limitados, brindando servicios con alta velocidad y una utilización óptima de los recursos computacionales.

4. CONCLUSIONES Y TRABAJO FUTURO

Este estudio se divide en dos partes principales. En la primera parte, se analizaron diferentes arquitecturas basadas en el modelo MobileNetV3 y transfer learning, considerando

las métricas ROC -AUC, ACC, PRE, REC y F1-score. El mejor rendimiento se obtuvo con Mobilenetv3 large+ esquema 2, con una tasa de aprendizaje de 0.0001 y pesos pre-entrenados congelados. Se obtuvieron resultados de AUC, ACC, PRE, REC y F1-score de 0.906 ± 0.02, 84.8% ± 0.02, 0.926 ± 0.02, 0.765 ± 0.02 y 0.814 ± 0.03, respectivamente, en una validación cruzada estratificada de 10 pliegues. Además, el rendimiento de la arquitectura óptima se evaluó en un conjunto de pruebas externo, logrando un puntaje AUC de 0.917. Este análisis de rendimiento confirmó la capacidad de generalización del modelo y demostró su exitosa clasificación del melanoma. En el futuro, se planea exp lorar nuevas variantes de modelos de redes neuronales convolucionales (CNN) mediante la experimentación con diferentes arquitecturas e hiperparámetros, y utilizando conjuntos de datos más grandes para el entrenamiento del modelo. Por otro lado, se desarrollaron dos tipos de aplicaciones basadas en la arquitectura óptima: una aplicación nativa y una aplicación web. La aplicación nativa alcanzó valores medios de ejecución satisfactorios de 8 y 7 segundos, y valores medios de consumo de batería de 0.32 y 0.21 mAh para los emuladores de Pixel 2 y Pixel C, respectivamente. La aplicación web mostró un tiempo de respuesta promedio de 115.44 ms, con un promedio de 15 kb transferidos a través de la red por imagen. Estos resultados indicaron que nuestra arquitectura óptima, basada en MobilenetV3 Large, es adecuada para su implementación en dispositivos móviles. Los planes futuros para la parte de la aplicación incluyen el desarrollo de una aplicación nativa para el sistema operativo MacOS. Además, se planea optimizar iterativamente la aplicación web utilizando herramientas de diagnóstico de aplicaciones web como Google Lighthouse. Este proceso de optimización se llevaráa cabo una vez que la aplicación web se implemente en la nube, lo que permitirámejoras continuas para mejorar su rendimiento y experiencia de usuario.

5. REFERENCIAS BIBLIOGRÁFICAS

American Cancer Society. (2023). Cancer facts and figures 2023. Retrieved from https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/annual-cancer-facts-and-figures/2023/2023-cancer-facts-and-figures.pdf

Charleen, C., Angelica, H., Purnama, H., & Purnomo, F. (2021). Impact of computer vision with deep learning approach in medical imaging diagnosis. In 2021 1st InternationalConference on Computer Science and Artificial Intelligence (ICCSAI) (pp. 37-41).

Dai, X., Spasić, I., Meyer, B., Chapman, S., & Andres, F. (2019). Machine learning on mobile: An on-device inference app for skin cancer detection. In 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC) (pp. 301-305).

Feng, H., Berk-Krauss, J., Feng, P. W., & Stein, J. A. (2018). Comparison of dermatologist density between urban and rural counties in the United States. JAMA Dermatology, 154(11), 1265-1271.

Gupta, M., Verma, S. K., & Jain, P. (2020). Detailed study of deep learning models for natural language processing. In 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN) (pp. 249-253).

Guarneri, I., Messina, G., Bruna, A., & Giacalone, D. (2021). A deep learning short commands recognition for MCU in robotics applications. In 2021 7th International Conference on Automation, Robotics and Applications (ICARA) (pp. 43-47).

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

Howard, A., et al. (2019). Searching for mobilenetv3. In 2019 IEEE/CVF InternationalConference on Computer Vision (ICCV) (pp. 1314-1324).

Jiang, T., et al. (2022). G protein-coupled receptor interaction prediction based on deep transfer learning. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 19(6), 3126-3134.

Li, Y., & Tan, X. (2009). An anti-photo spoof method in face recognition based on the analysis of Fourier spectra with sparse logistic regression. In 2009 Chinese Conference on Pattern Recognition (pp. 1-5).

Medina, K. R. (2021). Estadísticas de la situación digital en Ecuador 2021 -2022. Retrieved from https://branch.com.co/marketing-digital/estadisticas-de-la-situacion-digital-en- ecuador-2021-

2022/#:~:text=Uso%20de%20celulares%20en%20Ecuador,el%2088%25%20de%20la%20misma

Muralidhar, K. (2021, February). What is stratified cross-validation in machine learning? Retrieved from https://towardsdatascience.com/what-is-stratified-cross-validation-in-machine-learning-8844f3e7ae8e

Narkhede, S. (n.d.). Understanding AUC-ROC curve - towards data science. Retrieved fromhttps://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Nasr-Esfahani, E., Samavi, S., Karimi, N., Soroushmehr, S. M. R., Jafari, M. H., Ward, K., & Najarian, K. (2016). Melanoma detection by analysis of clinical images using convolutional neural network. In 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 1373-1376).

Patil, R., & Bellary, S. (2021). Ensemble learning for detection of types of melanoma. In 2021 International Conference on Computing, Communication and Green Engineering (CCGE) (pp. 1-6).

Pham, T.-C., Luong, C.-M., Visani, M., & Hoang, V.-D. (2018). Deep CNN and data augmentation for skin lesion classification. In Asian Conference on Intelligent Information and Database Systems (pp. 573-582).

Raschka, S., & Mirjalili, V. (2017). Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow (2nd ed.). Packt Publishing.

Rivera, D., Grijalva, F., Acurio, B. A. A., & Álvarez, R. (2019). Towards a mobile and fastmelanoma detection system. In 2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI) (pp. 1-6).

Rokhana, R., Herulambang, W., & Indraswari, R. (2020). Deep convolutional neural network for melanoma image classification. In 2020 International Electronics Symposium (IES) (pp. 481-486).

Sadiq, M. U., Sankalpa, D., Ahfid, K., Sagahyroon, A., & Dhou, S. (2020). Preliminary melanoma detection mobile application using support vector machine classification. In 2020 International Conference on Computing, Electronics Communications Engineering (iCCECE) (pp. 115-118).

Tschandl, P. M. U. o. V. (2018). The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions [Data set]. Retrieved from https://doi.org/10.7910/DVN/DBW86T

World Health Organization. (2019). How common is skin cancer? Retrieved fromhttps://www.who.int/uv/faq/skincancer/en/index1.html

World Health Organization. (2019). Who is most at risk of getting skin cancer? Retrievedfrom https://www.who.int/uv/faq/skincancer/en/index2.html

Yuan, Y., & Lo, Y.-C. (2019). Improving dermoscopic image segmentation with e nhanced convolutional-deconvolutional networks. IEEE Journal of Biomedical and Health Informatics, 23(2), 519-526.