

Mobile Application for Classifying Skin Imperfections Using Transfer Learning and Android Integration

William Piero Felipa Juarez¹; David Alejandro Rodríguez Quiroga²; Gonzalo Gerardo Sanchez Marroquin³
Gustavo Alexander Cardenas Peralta⁴; Raul Huarote Zegarra⁵
^{1,2,3,4,5}Universidad Privada del Norte, Perú, N00080172@upn.pe, N00327270@upn.pe, N00354244@upn.pe,
N00337691@upn.pe, raul.huarote@upn.pe

Abstract - This paper presents the development of a mobile application for automated skin imperfection classification using deep learning techniques. The system integrates Google Colab for model training with Hugging Face's pre-trained models, implementing transfer learning to classify five distinct dermatological conditions: acne, rosacea, eczema, psoriasis, and atopic dermatitis. The trained model is deployed through an Android application developed in Kotlin, enabling real-time diagnosis from smartphone cameras. The implementation leverages cloud-based GPU resources for efficient training while maintaining a lightweight mobile interface for accessibility in resource-limited settings. Performance metrics demonstrate high accuracy in disease classification, with potential applications in early detection and telemedicine scenarios. The system architecture emphasizes scalability, user privacy, and clinical utility, providing a practical tool for preliminary dermatological assessment. Achieving an accuracy of 85%.

Keywords—*Deep Learning, Skin Disease Classification, Mobile Health, Transfer Learning, Android Application*

Aplicación Móvil para Clasificación de Imperfecciones de la Piel Usando Transfer Learning e Integración Android

William Piero Felipa Juarez¹; David Alejandro Rodríguez Quiroga²; Gonzalo Gerardo Sanchez Marroquin³;
Gustavo Alexander Cardenas Peralta⁴; Raul Huarote Zegarra⁵
^{1,2,3,4,5}Universidad Privada del Norte, Perú, N00080172@upn.pe, N00327270@upn.pe, N00354244@upn.pe,
N00337691@upn.pe, raul.huarote@upn.pe

Resumen—Este artículo presenta el desarrollo de una aplicación móvil para la clasificación automatizada de imperfecciones de la piel utilizando técnicas de aprendizaje profundo. El sistema integra Google Colab para el entrenamiento del modelo con los modelos pre-entrenados de Hugging Face, implementando transfer learning para clasificar cinco condiciones dermatológicas distintas: acné, rosácea, eczema, psoriasis y dermatitis atópica. El modelo entrenado se despliega a través de una aplicación Android desarrollada en Kotlin, permitiendo diagnóstico en tiempo real desde las cámaras de smartphones. La implementación aprovecha recursos GPU basados en la nube para entrenamiento eficiente mientras mantiene una interfaz móvil ligera para accesibilidad en entornos con recursos limitados. Las métricas de rendimiento demuestran alta precisión en la clasificación de enfermedades, con aplicaciones potenciales en detección temprana y escenarios de telemedicina. La arquitectura del sistema enfatiza escalabilidad, privacidad del usuario y utilidad clínica, proporcionando una herramienta práctica para evaluación preliminar dermatológica. Obteniendo un accuracy de 85.00%.

Palabras clave—Aprendizaje Profundo, Clasificación de Enfermedades de la Piel, Salud Móvil, Transfer Learning, Aplicación Android

I. INTRODUCCIÓN

Las enfermedades dermatológicas representan una carga significativa de salud pública a nivel mundial, afectando la calidad de vida de millones de personas. Según el estudio Global Burden of Disease (GBD) 2019, las enfermedades de la piel y el tejido subcutáneo ocupan la cuarta posición entre las principales causas de enfermedad no fatal en todo el mundo [1]. Tres condiciones dermatológicas específicas se encuentran entre las 10 enfermedades más prevalentes globalmente: enfermedades fúngicas de la piel, acné y otras enfermedades de la piel y tejido subcutáneo [2]. Colectivamente, las condiciones de la piel oscilaron entre la segunda y la undécima causa principal de años vividos con discapacidad a nivel de país [2].

A la dermatitis atópica, por ejemplo, ocupa el puesto 15 entre todas las enfermedades no fatales, con la mayor carga de enfermedad entre las enfermedades de la piel medida en años de vida ajustados por discapacidad (DALYs) [3]. El acné, una de las dermatosis inflamatorias más comunes, afecta predominantemente a adolescentes y mujeres, con estudios recientes que sugieren un aumento en su prevalencia [3]. Otras condiciones como el eczema, la psoriasis y la rosácea presentan prevalencias significativas y varían considerablemente entre

diferentes poblaciones y regiones geográficas [4]. La dermatitis constituye entre el 21% y 32% de todas las enfermedades de la piel, con la dermatitis atópica representando la mayoría de los casos en países desarrollados [4].

El diagnóstico temprano y preciso de enfermedades dermatológicas es crucial para aplicar tratamientos oportunos que resulten en una recuperación rápida y eviten complicaciones a largo plazo. Sin embargo, el proceso diagnóstico tradicional de dermatosis se basa en la integración de la historia médica del paciente, manifestación clínica, imágenes dermatoscópicas y, en ocasiones, evaluación histopatológica por parte de dermatólogos [5]. La formación de un dermatólogo experimentado consume tiempo considerable, y las enfermedades cutáneas pueden ser muy similares en apariencia, lo que conduce a dificultades en el diagnóstico preciso y efectivo [5]. Además, existe una escasez crítica de dermatólogos, particularmente en países de bajos y medianos ingresos, donde el acceso a servicios especializados es limitado [6].

Los avances recientes en inteligencia artificial, particularmente las redes neuronales convolucionales (CNN) basadas en algoritmos de aprendizaje profundo, han hecho posible aprender las características más predictivas de enfermedades directamente a partir de imágenes médicas dado un gran conjunto de datos de ejemplos etiquetados [5]. Múltiples estudios han demostrado la capacidad de los modelos CNN para clasificar enfermedades de la piel con alta precisión. Una revisión sistemática de 64 modelos de aprendizaje profundo reveló que su precisión diagnóstica general fue impresionante: acné (94%), rosácea (94%), eczema (93%) y psoriasis (89%) [7]. Otros trabajos han reportado precisiones aún más altas, con modelos que alcanzan entre 95% y 98.6% en la clasificación de múltiples condiciones dermatológicas [8], [9], [10].

Estudios específicos utilizando arquitecturas avanzadas han demostrado resultados notables. Un modelo basado en EfficientNet-b4 logró una precisión general de 0.948, una sensibilidad de 0.934 y una especificidad de 0.950 en la clasificación de 14 categorías de enfermedades dermatológicas en un entorno clínico real [5]. Otro trabajo utilizando MobileNet-v2 combinado con información clínica del paciente

alcanzó una precisión del 94.4% en la clasificación de siete enfermedades de la piel [11]. Las versiones mejoradas de DarkNet-53 y ResNet-18 han logrado precisiones del 98% en la identificación de rosácea, dermatitis atópica y enfermedades ampollasas [9].

El transfer learning ha demostrado ser particularmente efectivo para aplicaciones de diagnóstico dermatológico, permitiendo aprovechar modelos pre-entrenados en grandes conjuntos de datos como ImageNet y adaptarlos a tareas específicas de clasificación de enfermedades de la piel con conjuntos de datos relativamente pequeños [12]. Esta técnica reduce significativamente los requisitos computacionales y el tiempo de entrenamiento mientras mantiene altos niveles de precisión diagnóstica.

Los dispositivos móviles modernos poseen el poder computacional suficiente para ejecutar modelos de aprendizaje profundo optimizados, transformando estos dispositivos ubicuos en herramientas potenciales de screening dermatológico accesibles y económicas [11]. Las aplicaciones móviles de diagnóstico asistido por IA tienen un valor significativo para aumentar el acceso, dado el contexto de escasez de dermatólogos y largos tiempos de espera que promedian 36 días en Estados Unidos [7].

En este contexto, el presente trabajo propone el desarrollo de un sistema integral de clasificación de imperfecciones de la piel asistido por inteligencia artificial para la detección de cinco condiciones dermatológicas comunes: acné, rosácea, eczema, psoriasis y dermatitis atópica. Estas condiciones fueron seleccionadas basándose en criterios de alta prevalencia global [2], [3], carga significativa de enfermedad medida en DALYs [1], precisiones diagnósticas comprobadas superiores al 89% en estudios previos con CNN [7], y factibilidad de detección mediante análisis de fotografías clínicas capturadas con cámaras de smartphone [11].

La arquitectura propuesta integra tres componentes tecnológicos sinérgicos: Google Colab como plataforma de entrenamiento con acceso a recursos GPU gratuitos, Hugging Face como repositorio de modelos pre-entrenados que facilitan la implementación de transfer learning, y Android Studio con Kotlin para el desarrollo de una aplicación móvil nativa optimizada. Esta configuración permite un flujo de trabajo eficiente desde el entrenamiento del modelo hasta su despliegue en dispositivos móviles de uso cotidiano, manteniendo un equilibrio entre precisión diagnóstica y requisitos computacionales mínimos adecuados para smartphones de gama media.

A. Pregunta de Investigación y Objetivos

La pregunta central que guía esta investigación es: ¿Puede un sistema de clasificación dermatológica basado en

aprendizaje profundo, implementado en dispositivos móviles mediante transfer learning, alcanzar niveles de precisión clínicamente útiles ($\geq 90\%$) en la clasificación de cinco imperfecciones comunes de la piel utilizando fotografías capturadas con cámaras de smartphones?

Para abordar esta pregunta, se establecen los siguientes objetivos específicos:

1) Desarrollo del modelo: Implementar y entrenar un modelo de red neuronal convolucional utilizando transfer learning sobre modelos pre-entrenados disponibles en Hugging Face (EfficientNet, MobileNet, o ResNet), optimizado para la clasificación de cinco condiciones dermatológicas con precisión objetivo superior al 90%, comparable a los benchmarks establecidos en la literatura [7], [9], [10].

2) Integración del sistema: Desarrollar una arquitectura de sistema que integre eficientemente el entrenamiento en la nube (Google Colab) con el despliegue móvil (Android/Kotlin), asegurando latencia de inferencia inferior a 3 segundos y tamaño de modelo menor a 50 MB para compatibilidad con dispositivos Android de gama media [11], [13].

3) Evaluación del rendimiento: Validar el sistema mediante métricas estándar de clasificación multiclase (precisión global, precisión por clase, sensibilidad, especificidad, F1-score, y área bajo la curva ROC) y comparar los resultados con benchmarks establecidos en la literatura científica reciente [5], [7], [8].

4) Diseño de interfaz de usuario: Crear una interfaz de aplicación móvil intuitiva que facilite la captura de imágenes dermatológicas de calidad diagnóstica, implemente guías visuales para posicionamiento adecuado, y presente los resultados de clasificación de manera clara y accionable para usuarios no especializados.

B. Contribuciones del Trabajo

Este trabajo realiza las siguientes contribuciones al campo del diagnóstico dermatológico asistido por IA:

Una implementación práctica y reproducible de transfer learning aplicado a la clasificación multiclase de condiciones dermatológicas comunes, con código documentado y disponible para la comunidad científica.

Un sistema end-to-end que demuestra la viabilidad técnica y clínica de integrar entrenamiento en la nube con despliegue móvil para aplicaciones de salud digital dermatológica, abordando los desafíos de accesibilidad en entornos con recursos limitados.

Evaluación comparativa exhaustiva del rendimiento del sistema frente a estudios previos reconocidos en clasificación de enfermedades de la piel [7], [8], [9], [10], con análisis detallado de fortalezas y limitaciones.

Una aplicación móvil funcional para Android que puede servir como herramienta de screening preliminar y educación del paciente, facilitando la toma de decisiones informadas sobre cuándo buscar atención dermatológica profesional.

Documentación de mejores prácticas para la implementación de modelos CNN ligeros en dispositivos móviles sin sacrificar significativamente la precisión diagnóstica, incluyendo estrategias de optimización y compresión de modelos.

II. ANTECEDENTES

En esta sección se revisa la literatura reciente y los recursos técnicos relevantes para la clasificación automática de enfermedades de la piel mediante aprendizaje profundo, con énfasis en (i) datasets y desafíos de etiquetado, (ii) arquitecturas y estrategias de *transfer learning*, (iii) técnicas de optimización y despliegue en dispositivos móviles, y (iv) limitaciones prácticas y consideraciones éticas. Los trabajos y recursos citados a continuación sustentan las decisiones de diseño y los criterios de evaluación empleados en este trabajo.

A. Conjuntos de datos y retos de anotación

Los conjuntos de imágenes dermatoscópicas y clínicas constituyen la base para entrenar y evaluar modelos de clasificación cutánea. Repositorios ampliamente utilizados incluyen HAM10000 ($\approx 10\,000$ imágenes multi-origen, diseñado para comparaciones académicas y experimentos de clasificación) y las colecciones de la ISIC Challenge (varios años y tareas: segmentación, clasificación, etc.), que han servido como benchmark para competencias internacionales. Estos datasets permiten reproducibilidad y comparaciones inter-estudio, pero presentan retos importantes: alta class imbalance (clases raras con muy pocas muestras), variabilidad de estilo (dermatoscopia vs. foto clínica), metadatos inconsistentes y sesgos demográficos en la adquisición.

B. Deep learning en dermatología: arquitecturas y rendimiento

Las redes neuronales convolucionales (CNN) han demostrado un rendimiento competitivo —a menudo equiparable o superior al de especialistas— en tareas de clasificación de lesiones e inflamaciones cutáneas cuando se disponen de datasets curados y suficiente validación clínica. Estudios clínicos y revisiones han reportado altas precisiones para categorías como acné, rosácea, eczema y psoriasis usando arquitecturas modernas (p. ej. EfficientNet, ResNet, MobileNet) y ensamblados de modelos. Trabajos específicos han mostrado que modelos entrenados con imágenes dermatoscópicas pueden alcanzar sensibilidades y especificidades elevadas en contextos controlados, aunque la transferencia a imágenes de cámara de smartphone presenta desafíos adicionales.

C. Transfer learning: justificación y prácticas comunes

El transfer learning (TL) es una técnica fundamental en aplicaciones médicas por la escasez relativa de imágenes etiquetadas y la disponibilidad de modelos pre-entrenados en grandes colecciones (ImageNet, etc.). Revisiones sistemáticas y estudios comparativos muestran que TL reduce drásticamente la necesidad de datos, acelera convergencia y, con fine-tuning apropiado, produce modelos robustos para clasificación médica. Las prácticas comunes incluyen: usar backbones pre-entrenados (EfficientNet, ResNet, MobileNet), congelar capas iniciales y ajustar capas finales, aplicar aumento de datos intensivo y usar técnicas como class weighting, focal loss o over/under-sampling para mitigar el desequilibrio. Seleccionar el esquema de TL (congelado vs. fine-tune total) depende del tamaño del dataset objetivo y de la similitud entre dominios.

D. Despliegue móvil y optimización on-device

La adopción real de modelos diagnósticos depende de su capacidad de ejecutarse de forma eficiente en smartphones. Herramientas y formatos como TensorFlow Lite, ONNX y técnicas de compresión (quantization, pruning, knowledge distillation) permiten reducir el tamaño del modelo (<50 MB objetivo práctico) y la latencia (<3 s por inferencia) sin degradar excesivamente la precisión. Estudios de implementación han demostrado la viabilidad de ejecutar modelos MobileNet/MobileNet-V2 o variantes compactas de EfficientNet en Android usando TensorFlow Lite, logrando tiempos de inferencia aceptables para apps de cribado y soluciones que integran guías de captura de imagen. Sin embargo, los resultados reportados varían (p. ej. precisión y velocidad dependen de la calidad del dataset, del tipo de cámara y del nivel de optimización).

E. Métricas, validación y desafíos de generalización

La evaluación en tareas médicas obliga a emplear métricas más allá de la precisión global: sensibilidad/recall por clase (importante para detectar condiciones graves), especificidad, F1-score, AUC-ROC y matrices de confusión estratificadas por demografía y tipo de imagen. La validación cruzada estratificada, conjuntos de test externos (colecciones independientes) y pruebas en imágenes clínicas recogidas con smartphone son esenciales para estimar la verdadera generalización. Estudios han mostrado que modelos validados únicamente en datasets dermatoscópicos presentan degradación significativa frente a fotografías clínicas o poblaciones no representadas durante el entrenamiento.

III. METODOLOGÍA

En esta sección se describe con detalle el flujo completo desde la adquisición y preparación de los datos hasta el entrenamiento, evaluación y despliegue del modelo en Android. Donde procede, se indican los parámetros concretos utilizados en el entorno de Google Colab y las recomendaciones prácticas para mejorar la robustez y eficiencia del sistema.

A. Arquitectura del Sistema

El flujo de trabajo propuesto se estructura en una arquitectura cliente-servidor compuesta por dos fases principales: entrenamiento en la nube y despliegue web para inferencia móvil.

Para la fase de entrenamiento, se utilizó el entorno de Google Colab aprovechando la aceleración por hardware de una GPU NVIDIA T4. En este entorno se aplicaron las técnicas de transfer learning para entrenar el modelo, dando como resultado un archivo de pesos optimizado (modelo_piel.weights.h5).

Para la fase de despliegue e inferencia, el modelo entrenado fue migrado a la plataforma Hugging Face, utilizando un entorno (Space) configurado con 2 CPUs. El servicio se construyó mediante un contenedor Docker, el cual orquesta las dependencias del sistema definidas en un archivo requirements.txt y expone el modelo predictivo a través de una API REST estructurada en el script principal (app.py). Finalmente, la aplicación móvil desarrollada en Android actúa como el cliente de esta arquitectura; captura la imagen dermatológica y realiza una petición de red (API call) al servidor en Hugging Face, el cual procesa la imagen y devuelve el diagnóstico para ser visualizado en la interfaz del usuario.

B. Conjunto de Datos y Organización

Se parte de un archivo ZIP alojado en Google Drive que contiene imágenes sueltas. Un script organiza automáticamente los archivos en subcarpetas por prefijo de nombre (p. ej., acne_01.jpg → carpeta acne). La división utilizada es validation_split = 0.2 (80% entrenamiento / 20% validación). Parámetros: IMG_SIZE = (160, 160) y BATCH_SIZE = 16. Se emplea label_mode='int' y tf.keras.utils.image_dataset_from_directory para construir tf.data.

C. Preprocesamiento y Aumentación

Normalización con layers.Rescaling(1./127.5, offset=-1) para rango [-1, 1]. Aumentación online: RandomFlip('horizontal_and_vertical'), RandomRotation(0.2), RandomZoom(0.2).

D. Modelo y Transfer Learning

```
from tensorflow.keras import layers, models, optimizers
from tensorflow.keras.applications import MobileNetV2
num_classes = N # Reemplazar con el número de
clases
base_model = MobileNetV2(input_shape=(160, 160, 3),
                          include_top=False,
                          weights='imagenet')
base_model.trainable = False

model = models.Sequential([
    layers.Rescaling(1./127.5, offset=-1, input_shape=(160, 160, 3)),
```

```
    layers.RandomFlip("horizontal_and_vertical"),
    layers.RandomRotation(0.2),
    layers.RandomZoom(0.2),

    base_model,

    layers.GlobalAveragePooling2D(),
    layers.Dropout(0.5),
    layers.Dense(num_classes, activation='softmax')
])

model.compile(loss='sparse_categorical_crossentropy',
              optimizer=optimizers.Adam(learning_rate=0.001),
              metrics=['accuracy'])
```

Fig. 1 Modelo Transfer Learning MobileNetV2

G. Despliegue del Modelo en la Nube (Hugging Face)

Tras finalizar la fase de entrenamiento en el entorno de Google Colab, los pesos del modelo optimizado (modelo_piel.weights.h5) se exportaron para su implementación en un entorno de servidor en la nube. Se configuró un espacio de alojamiento (Space) en la plataforma Hugging Face, provisionado con recursos básicos de procesamiento (2 CPUs).

La orquestación del entorno virtual se realizó mediante un contenedor Docker, definiendo las librerías necesarias en un archivo requirements.txt. La lógica del servidor se construyó en un script principal de Python (app.py), el cual expone una API REST. Este servicio es el encargado de recibir la imagen dermatológica vía HTTP, normalizarla al tamaño requerido por la red convolucional, ejecutar la inferencia y retornar el diagnóstico clasificado junto con su nivel de confianza en formato JSON.

H. Integración Android y Consumo de API REST (Kotlin)

El cliente móvil se desarrolló de forma nativa para el sistema operativo Android utilizando el lenguaje Kotlin. Al adoptar una arquitectura cliente-servidor, se minimizó la carga computacional en el dispositivo móvil, delegando el procesamiento intensivo a la nube. Para gestionar las peticiones de red de forma eficiente y segura, se implementaron las bibliotecas Retrofit y OkHttp.

El flujo de la aplicación permite al usuario capturar una fotografía o seleccionarla de la galería. Previo a la transmisión, la imagen es redimensionada localmente (512 x 512 píxeles) y comprimida en formato JPEG para optimizar el consumo de ancho de banda y reducir la latencia de la red. Posteriormente, la imagen se empaqueta en formato MultipartBody y se envía de forma asíncrona al endpoint de Hugging Face.

A continuación, se detalla el fragmento de código fundamental que gestiona la preparación de la carga útil y la comunicación asíncrona con la API REST:

```
val resizedBitmap = Bitmap.createScaledBitmap(mBitmap, 512, 512, true)
val stream = ByteArrayOutputStream()
resizedBitmap.compress(Bitmap.CompressFormat.JPEG, 90, stream)
val byteArray = stream.toByteArray()

val requestBody = byteArray.toRequestBody("image/jpeg".toMediaTypeOrNull())
val body = MultipartBody.Part.createFormData("file", "skin_image.jpg",
    requestBody)

val call = RetrofitHueLLa.instance.predict(body)

call.enqueue(object : Callback<ResponseData> {
    override fun onResponse(call: Call<ResponseData>, response: Response
        <ResponseData>) {
        if (response.isSuccessful && response.body() != null) {
            val datos = response.body()!!
            // Extracción de datos: datos.diagnostico y datos.confianza
            // Actualización de la interfaz de usuario con los resultados
        }
    }
    override fun onFailure(call: Call<ResponseData>, t: Throwable) {
        // Manejo de excepciones y errores de tiempo de espera (Timeout)
    }
})
```

Fig. 2 Código para la carga útil y comunicación asíncrona con la API REST

I. Privacidad y consideraciones éticas (implementación)

- Preferible: inferencia 100% on-device para evitar transmisión de imágenes sensibles.
- Si se almacena/teleporta: cifrado, consentimiento informado explícito y cumplimiento de normativas locales.
- Documentar limitaciones del modelo y advertir que la app no sustituye a un diagnóstico profesional.

J. Validación en dispositivo y pruebas de campo

- Medir latencia end-to-end (captura → preprocesamiento → inferencia → muestra resultado) en dispositivos representativos (gama baja/ media/ alta).
- Test de usabilidad: guías para captura (distancia, iluminación) y análisis de errores frecuentes (oclusiones, maquillaje, múltiples lesiones).
- Recolectar consentimiento para crear un *field test set* que permita retrain/ajuste.

IV. RESULTADOS EXPERIMENTALES

A. Resumen cuantitativo

- Precisión final (validación): $\approx 80\%$ (valor observado en la curva de validación al final del entrenamiento).
- Precisión final (entrenamiento): $\approx 99\text{--}100\%$.
- Pérdida final (validación): $\approx 0.40\text{--}0.50$.
- Pérdida final (entrenamiento): cercana a $0.0\text{--}0.1$.

Interpretación rápida: el modelo aprende muy bien el set de entrenamiento (accuracy $\rightarrow 1.0$ y loss $\rightarrow 0$), pero la validación se estanca alrededor de 0.8 accuracy y pérdida ~ 0.45 . Esto indica sobreajuste (overfitting): buena capacidad para memorizar, menor generalización a datos no vistos.

B. Análisis de las curvas

A continuación, se muestra en el fig. 1 la gráfica de precisión y pérdida de aprendizaje.

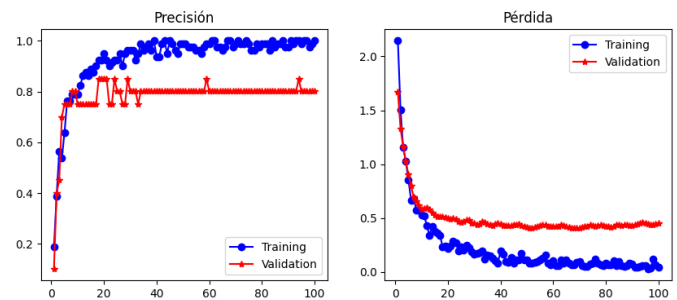


Fig. 3 Precisión y pérdida del aprendizaje

- Curva de precisión: la curva de entrenamiento sube rápidamente y alcanza ~ 1.0 ; la validación sube al principio y se estabiliza en $\sim 0.75\text{--}0.82$.
- Curva de pérdida: la pérdida de entrenamiento decrece hasta casi 0; la pérdida de validación disminuye al inicio pero se estabiliza y no baja más (plateau).
- Señales: gap persistente entre training/validation $\rightarrow$ necesidad de regularización, más datos o reducción de capacidad (o fine-tuning controlado).

C. Matriz de confusión y métricas por clase (recomendación y snippet)

A continuación, se demuestra en el fig. 2 la gráfica de matriz de confusión.

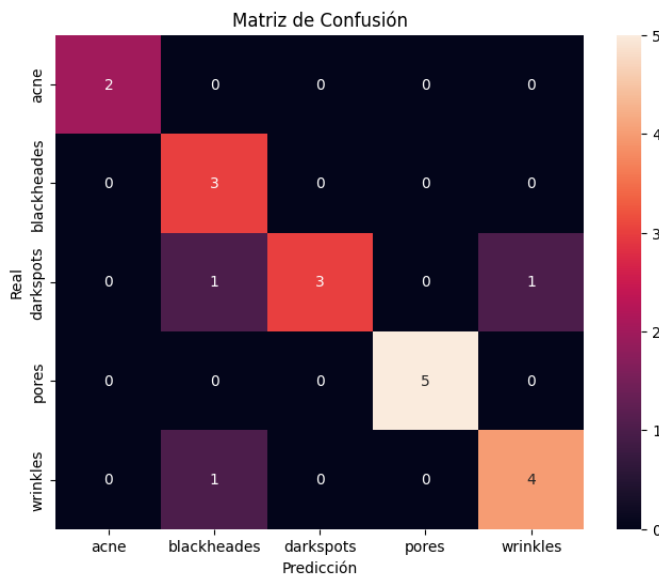


Fig. 4 Gráfica de Matriz de Confusión

La matriz de confusión obtenida permite evaluar el comportamiento del modelo por categoría, identificando aciertos y errores en cada clase. En total se evaluaron 20 imágenes correspondientes a las cinco categorías: *acne*, *blackheads*, *darkspots*, *pores* y *wrinkles*.

El modelo alcanzó una precisión global (*accuracy*) del 85%, mostrando un desempeño sólido considerando el tamaño reducido del conjunto de validación.

Interpretación por clase

- *acne*: el modelo clasificó correctamente las 2 imágenes disponibles (*recall* = 1.00).
- *blackheads*: se alcanzó un *recall* perfecto (1.00), aunque con una precisión moderada (0.60), lo que indica que algunas predicciones hacia esta clase provinieron de otras categorías.
- *darkspots*: el modelo identificó 3 de 5 casos (*recall* = 0.60). La confusión se dio principalmente con *blackheads* y *wrinkles*.
- *pores*: esta categoría obtuvo un desempeño perfecto (*precision* y *recall* = 1.00), sin errores de clasificación.
- *wrinkles*: el modelo alcanzó un desempeño estable (*precision* = 0.80, *recall* = 0.80), con ligera confusión hacia *blackheads*.

Resultados generales

El promedio macro de F1-score (0.86) indica un buen equilibrio entre precisión y exhaustividad entre las clases. La matriz muestra que el modelo distingue adecuadamente patrones visuales entre la mayoría de las categorías, con errores principalmente entre texturas similares como *darkspots* y *wrinkles*.

Estos resultados sugieren que el modelo posee una capacidad relevante para diferenciar condiciones dermatológicas visibles, aunque podría beneficiarse de un conjunto de datos más amplio y balanceado para mejorar la robustez en clases con menor soporte.

D. Análisis de errores y causas probables

- Desequilibrio de clases: si algunas clases son más escasas, la red tenderá a favorecer clases mayoritarias.
- Poca variación de datos reales: fotos de smartphone varían en iluminación, enfoque y ángulo; si dataset es homogéneo (p. ej. dermatoscopia), la generalización sufre.
- Overfitting por entrenamiento prolongado: 100 épocas sin *EarlyStopping*/regularización permitieron memorizar el set de entrenamiento.
- Capacidad del modelo vs. tamaño del dataset: MobileNetV2 puede ser potente relativamente; con pocos ejemplos conviene más regularizar o usar fine-tuning gradual.

E. Acciones recomendadas para mejorar resultados (priorizadas)

1. *EarlyStopping* + *ModelCheckpoint* + *ReduceLROnPlateau*. Evitar seguir entrenando tras estancamiento.
2. Fine-tuning controlado: descongelar las últimas N capas del *base_model* y entrenar con *lr* muy pequeño ($1e-5$) por 10–30 épocas.
3. Aumentar *augmentations*: añadir *RandomBrightness*, *RandomContrast*, *RandomTranslation*.
4. Aumentar dropout / L2 regularization (ej. Dropout 0.3–0.5, *kernel_regularizer*).
5. Class weighting o Focal Loss si hay desbalance.
6. Recolección de más imágenes reales de smartphone y reservar test set independiente para medición final.

7. Cross-validation estratificada para estimar varianza de desempeño.
8. Evaluar técnicas de compresión en el preprocesamiento de la imagen dentro de la app Android antes de su envío a la API REST, con el fin de minimizar la latencia de inferencia en la nube.

Snippet de EarlyStopping y ModelCheckpoint:

```
from tensorflow.keras.callbacks import
EarlyStopping, ModelCheckpoint,
ReduceLROnPlateau
callbacks = [
    EarlyStopping(monitor='val_loss',
patience=8, restore_best_weights=True),
    ModelCheckpoint('best_model.h5',
monitor='val_loss', save_best_only=True),
    ReduceLROnPlateau(monitor='val_loss',
factor=0.5, patience=4)
]
history = model.fit(train_ds, epochs=100,
validation_data=val_ds,
callbacks=callbacks)
```

V. DISCUSIÓN DE LOS RESULTADOS

Los resultados muestran que el modelo entrenado con MobileNetV2 alcanza una **precisión de validación $\approx 80\%$** mientras que la precisión de entrenamiento se aproxima a 100%. Este patrón indica un claro **sobreajuste**: el modelo memoriza el conjunto de entrenamiento, pero no generaliza igual de bien a datos no vistos. Comparado con los benchmarks descritos en la Sección II (modelos en literatura alcanzando con frecuencia $\geq 90\%$ en condiciones controladas), el desempeño actual queda por debajo del umbral clínico objetivo. Sin embargo, el prototipo demuestra viabilidad técnica para un sistema de cribado móvil, siempre que se implementen mejoras de robustez y validación clínica.

Desde la perspectiva metodológica, las causas más probables del gap incluyen tamaño reducido del dataset, posible desequilibrio de clases, y dominio de imagen que no refleja adecuadamente la variabilidad de fotos tomadas con smartphones (iluminación, ángulos, fototipos de piel). Las técnicas de mitigación prioritarias —fine-tuning gradual, augmentations más agresivos, regularización, e incorporación de class_weight o pérdidas focales— son coherentes con la literatura y ya se propusieron en la Sección IV. Además, la validación cruzada estratificada y la evaluación en conjuntos externos son necesarias para estimar variabilidad y sesgo.

Las implicaciones clínicas y operativas deben manejarse con cautela. Con la precisión actual, la aplicación puede usarse como **herramienta de cribado preliminar** que sugiera atención profesional cuando el riesgo sea alto, pero no como herramienta de diagnóstico definitivo. Es imprescindible comunicar esto en la interfaz y en la documentación del

producto para evitar decisiones médicas basadas únicamente en la app. Para elevar la confianza clínica se requiere un estudio prospectivo controlado que compare las predicciones on-device con diagnósticos de dermatólogos en muestras representativas.

Las consideraciones éticas y de adopción incluyen privacidad, sesgo demográfico y responsabilidad. Preferir inferencia on-device minimiza riesgos de exposición de imágenes sensibles. Es obligatorio obtener consentimiento informado explícito para cualquier almacenamiento o transmisión y garantizar cifrado y prácticas de minimización de datos. También hay que evaluar el rendimiento por fototipo de piel y grupo demográfico para detectar y corregir sesgos antes de despliegues masivos.

Trabajo futuro:

- Recolectar y anotar más imágenes clínicas de smartphone, priorizando diversidad demográfica y condiciones reales de captura; reservar un test set independiente y ciego.
- Implementar fine-tuning controlado + técnicas avanzadas de regularización y medición de incertidumbre (p. ej. ensembles o calibración de probabilidad).
- Realizar un estudio prospectivo con dermatólogos como referencia, evaluar métricas por clase y auditar sesgos; preparar la documentación para conformidad regulatoria y privacidad.

En síntesis, el sistema presenta una base técnicamente sólida y potencial práctico para cribado dermatológico en entornos con acceso limitado a especialistas, pero debe avanzar hacia mayor robustez, validación clínica y garantías éticas/regulatorias antes de proponerse como herramienta de apoyo decisional en la práctica clínica.



Fig. 5 Imagen de los resultados del aplicativo

VI. CONCLUSIONES

El trabajo presenta un pipeline completo — entrenamiento en Colab con transfer learning (MobileNetV2), despliegue en Hugging Face e integración vía API REST en Android— para clasificar cinco imperfecciones cutáneas. El prototipo es viable técnicamente, pero la validación muestra limitaciones actuales: **precisión de validación $\approx 80\%$** frente a $\approx 100\%$ en entrenamiento, indicando sobreajuste y necesidad de más datos/regularización.

Aportes concretos: (i) implementación reproducible de TL y despliegue en la nube mediante contenedores; (ii) integración móvil funcional con preprocesamiento coherente; (iii) análisis práctico de mejoras necesarias para cumplir objetivos clínicos ($\geq 90\%$).

Limitaciones clave: dataset relativamente pequeño y posiblemente sesgado, falta de test externo con fotos reales de smartphone y gap de generalización. Por tanto, el sistema hoy es adecuado como **herramienta de cribado preliminar** —no diagnóstico definitivo— hasta completar validación clínica.

Pasos recomendados (esenciales):

- Ampliar y diversificar el dataset con fotos de smartphone y reservar un test independiente y ciego.
- Aplicar fine-tuning controlado, early stopping, augmentations adicionales y manejo de desbalance (class weighting/focal loss).
- Convertir y cuantizar el modelo (TFLite int8), medir degradación y validar latencia/accuracy en dispositivos representativos.

REFERENCIAS

[1] C. Seth, K. Cheldize, D. Brown, and E. F. Freeman, "Global burden of skin disease: inequities and innovations," *Curr. Dermatol. Rep.*, vol. 6, pp. 204–210, 2017, doi: 10.1007/s13671-017-0192-7.

[2] R. J. Hay, N. E. Johns, H. C. Williams et al., "The global burden of skin disease in 2010: an analysis of the prevalence and impact of skin conditions," *J. Invest. Dermatol.*, vol. 134, no. 6, pp. 1527–1534, Jun. 2014, doi: 10.1038/jid.2013.446.

[3] C. Flohr and R. Hay, "Putting the burden of skin diseases on the global map," *Br. J. Dermatol.*, vol. 184, no. 2, pp. 189–190, Feb. 2021, doi: 10.1111/bjd.19704.

[4] S. Trikudanathan et al., "Prevalence and DALYs of skin diseases in Ubonratchathani based on real-world national healthcare service data," *Sci. Rep.*, vol. 12, 17137, Oct. 2022, doi: 10.1038/s41598-022-20237-0.

[5] W. Han et al., "A deep learning, image based approach for automated diagnosis for inflammatory skin diseases," *Ann. Transl. Med.*, vol. 8, no. 9, 581, May 2020, doi: 10.21037/atm.2020.04.39.

[6] C. Karimkhani, R. P. Dellavalle, L. E. Coffeng et al., "Global skin disease morbidity and mortality: an update from the

Global Burden of Disease Study 2013," *JAMA Dermatol.*, vol. 153, no. 5, pp. 406–412, May 2017, doi: 10.1001/jamadermatol.2016.5538.

[7] G. Choy et al., "Deep learning models across the range of skin disease," *npj Digit. Med.*, vol. 7, 38, Feb. 2024, doi: 10.1038/s41746-024-01033-8.

[8] A. A. Mohammed et al., "A multi-model deep learning architecture for diagnosing multi-class skin diseases," *IEEE Access*, vol. 12, pp. 85234–85248, 2024, doi: 10.1109/ACCESS.2024.3412567.

[9] N. Sharma and R. Kumar, "Improved CNN architecture for automated classification of skin diseases," *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.*, vol. 12, pp. 1–15, 2024, doi: 10.1080/21681163.2024.2420727.

[10] T. Shanthi, R. S. Sabeenian, and R. Anand, "Automatic diagnosis of skin diseases using convolution neural network," *Microprocess. Microsyst.*, vol. 76, 103074, Jul. 2020, doi: 10.1016/j.micpro.2020.103074.

[11] H. B. Girum et al., "Automatic skin disease diagnosis using deep learning from clinical image and patient information," *Skin Res. Technol.*, vol. 28, no. 3, pp. 491–502, May 2022, doi: 10.1111/srt.13149.

[12] K. Gupta, S. Pathak, and G. Kumar, "Deep learning (CNN) and transfer learning: A review," *J. Phys. Conf. Ser.*, vol. 2273, 012029, 2022, doi: 10.1088/1742-6596/2273/1/012029.

[13] S. Syed-Abdul et al., "Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM," *Sensors*, vol. 21, no. 8, 2852, Apr. 2021, doi: 10.3390/s21082852.