

Clasificación no supervisada del tipo de piel facial usando embeddings de CNN y mapas autoorganizados SOM

Jairo Daniel Mendoza Torres¹ ; Elías Samuel Talledo Vega² ; Raúl Eduardo Huarote Zegarra³

^{1,2,3}Universidad Nacional Tecnológica de Lima Sur, Lima, Perú

¹20b3010069@untels.edu.pe, ²2113110283@untels.edu.pe, ³rhuarote@untels.edu.pe

Resumen — El presente trabajo describe el diseño, desarrollo e implementación de una aplicación móvil capaz de identificar el tipo de piel facial (normal, grasa, seca) a partir de una imagen proporcionada por el usuario. La solución se fundamenta en un sistema híbrido que integra una red neuronal convolucional (CNN), utilizada como extractor de características, y una red autoorganizada (SOM), encargada de clasificar los vectores latentes en clusters representativos del tipo de piel. La arquitectura de la aplicación combina procesamiento local (captura y selección de imagen, selección de tono de piel) con servicios remotos alojados en Hugging Face Spaces, accesibles a través de una API REST. El modelo alcanzó niveles de precisión superiores al 90 % en pruebas controladas. La implementación móvil en Android Studio con Kotlin permitió lograr una interfaz amigable y funcional, compatible con dispositivos modernos. Este enfoque demuestra ser una alternativa eficiente, accesible y ampliable para la evaluación dermatológica automatizada.

Palabras clave — Redes neuronales, tipo de piel, CNN, SOM, clasificación no supervisada.

I. INTRODUCCIÓN

La piel facial constituye el órgano más expuesto y visible del cuerpo humano. Su condición no solo es reflejo de la salud y estética individual, sino que desempeña un papel crucial en la selección de productos cosméticos apropiados y en la prevención de afecciones dermatológicas como el acné, la dermatitis y la sequedad severa [1]. Factores ambientales y fisiológicos —como la radiación solar, el clima, la contaminación, el estrés o las variaciones hormonales— inciden en las características de la piel facial, favoreciendo que ésta se manifieste como grasa, seca o equilibrada [2]. La identificación precisa del tipo de piel resulta fundamental para la personalización de rutinas de cuidado y para la prevención de daños cutáneos en distintos segmentos poblacionales.

En la actualidad, la investigación ha demostrado que la IA puede aplicarse eficazmente al análisis de imágenes faciales para la detección y clasificación automatizada de tipos de piel [3], [2], [4]. Las soluciones supervisadas, especialmente aquellas basadas en CNN, han alcanzado elevadas precisiones; no obstante, suelen requerir grandes volúmenes de datos etiquetados para su entrenamiento adecuado.

Según la literatura reciente, las redes SOM permiten efectuar agrupamientos no supervisados, lo que resulta

especialmente eficiente en contextos donde la anotación de datos es limitada [5]. El investigador [6] introdujo esta arquitectura como un método efectivo para mapear entradas multidimensionales preservando la topología de los datos. Además, se ha reportado la eficacia de las SOM en aplicaciones de clasificación dérmica implementadas en dispositivos móviles [5].

Por otra parte, la presencia de maquillaje puede alterar significativamente las características visuales reales de la piel y, en consecuencia, dificultar su clasificación automática mediante IA. Para abordar este desafío, se han desarrollado enfoques que emplean autoencoders, arquitecturas VGG16 y métodos de análisis de textura, logrando detectar el maquillaje con precisiones elevadas [7], [8], [9], [10].

Diversas revisiones sistemáticas han evidenciado tanto el potencial como los límites actuales de las técnicas basadas en aprendizaje profundo en el ámbito dermatológico [1]. Asimismo, se han implementado aplicaciones móviles orientadas al diagnóstico facial y a la recomendación personalizada de productos o rutinas [11]. Otros trabajos exploran el uso de machine learning para la clasificación automática de fototipos cutáneos y enfermedades de la piel [12], [13].

En este contexto, se plantea una solución dual compuesta por una CNN, empleada como método de extracción de características, y una red SOM para la clasificación del estado de la piel facial. La propuesta incluye el desarrollo de una aplicación en *Android Studio* utilizando *Kotlin*, capaz de capturar imágenes, ejecutar preprocesamiento visual y clasificar automáticamente el tipo de piel mediante el modelo entrenado.

II. MATERIALES Y MÉTODOS

El sistema utiliza dos modelos: (i) una CNN EfficientNetB0 como extractor de características y (ii) una red SOM como clasificador no supervisado, ambos expuestos como microservicios RESTful mediante FastAPI. El procedimiento completo de entrenamiento se resume en el algoritmo 1.

A. Recursos computacionales

Se emplearon Torchvision, PyTorch, scikit-learn, OpenCV y matplotlib. El SOM se entrenó en GPU (NVIDIA GeForce RTX 3050) usando JupyterNotebook, Visual Studio Code y se guardó en formatos .pt y .npy. EfficientNetB0 solo se usó en modo inferencia. Para la inferencia en producción,

se desplegaron microservicios REST mediante FastAPI (en Hugging Face Spaces), segmentados por tono de piel.

B. Conjunto de datos

El conjunto de datos se creó manualmente a partir de imágenes faciales clasificadas en tres categorías (piel grasa, normal y seca), siguiendo criterios dermatológicos establecidos por expertos y características fotográficas como textura, brillo y porosidad. Las capturas se realizaron en entornos controlados y sin maquillaje para evitar sesgos, durante la extracción de características. El dataset incluye diversidad étnica con subconjuntos separados para tonos de piel oscura, morena y blanca, con aproximadamente 100 imágenes por clase y tono, totalizando alrededor de 900 imágenes. Se dividió en 80 % para entrenamiento y 20 % para validación, estratificado por clase.

C. Preprocesamiento, arquitectura y entrenamiento

El siguiente algoritmo brinda una visualización de como se procedió con el entrenamiento.

Algorithm 1 Entrenamiento no supervisado del modelo CNN-SOM

Require: Dataset de imágenes faciales etiquetadas en {*grasa*, *normal*, *seca*}, EfficientNetB0 preentrenada, tamaño de SOM 20×20 , σ_0 , α_0 , $T = 20000$ iteraciones

Ensure: Pesos entrenados del SOM y matriz de etiquetas dominantes

```

1: Inicializar lista  $E \leftarrow []$ 
2: for all imagen  $\in$  Dataset do
3:   Preprocesar: convertir a RGB, redimensionar a  $224 \times 224$ , normalizar
4:    $e \leftarrow \text{EfficientNetB0}(\text{imagen}) \triangleright$  Embedding de 1280 dimensiones
5:   Añadir  $e$  a  $E$ 
6: end for
7: Inicializar SOM( $20 \times 20$ ) con pesos aleatorios
8: for  $t \leftarrow 1$  to  $T$  do
9:    $\alpha \leftarrow \alpha_0(1 - t/T)$ ,  $\sigma \leftarrow \sigma_0(1 - t/T)$ 
10:  Elegir aleatoriamente embedding  $e \in E$ 
11:  Encontrar BMU  $u = \arg \min_n \|e - w_n\|$ 
12:  for all neurona  $n$  en vecindad de  $u$  (radio  $\sigma$ ) do
13:     $w_n \leftarrow w_n + \alpha(e - w_n)$ 
14:  end for
15: end for
16: Asignar etiquetas:
17: for all neurona  $n$  do
18:    $\ell_n \leftarrow$  mayoría de clases de embeddings mapeados en  $n$ 
19: end for
20: Guardar pesos SOM en .pt y matriz  $\{\ell_n\}$  en .npy

```

Todas las imágenes se convirtieron a formato RGB, se redimensionaron a 224×224 px y se normalizaron según los requerimientos de EfficientNetB0. Este modelo, preentrenado en ImageNet, se empleó como extractor congelado para generar *embeddings* de 1280 dimensiones, los cuales fueron estandarizados y utilizados como entrada a una SOM de tamaño 20×20 . El entrenamiento de la SOM se

realizó durante 20000 iteraciones mediante un esquema no supervisado, donde las etiquetas reales se emplearon únicamente en la etapa de evaluación para medir la calidad de la agrupación. La implementación completa se resume en el Algoritmo 1, y los resultados se almacenaron en formato .pt (pesos) y .npy (matriz de etiquetas dominantes).

III. ANÁLISIS Y MATRIZ DE CONFUSIÓN PARA LA RED SOM

El modelo SOM se entrena de forma no supervisada, asignando a cada neurona la clase mayoritaria de los *embeddings* mapeados durante el proceso de entrenamiento, como se detalla en el algoritmo 1. En este esquema, las etiquetas reales se utilizan exclusivamente durante la evaluación, sin intervenir en la fase de aprendizaje.

Para evaluar el rendimiento del modelo y cuantificar la calidad de la agrupación, se empleó la **matriz de confusión**, la cual compara las etiquetas verdaderas con las predicciones obtenidas. Cada elemento $M_{i,j}$ de la matriz representa el número de muestras cuya clase real es i y que fueron clasificadas como j :

$$M_{i,j} = \sum_{k=1}^N \mathbb{I}(y_k = i \wedge \hat{y}_k = j), \quad (1)$$

donde:

- y_k es la etiqueta real de la muestra k .
- $\hat{y}_k$ es la etiqueta predicha por el modelo para la muestra k .
- N es el número total de muestras evaluadas.
- $\mathbb{I}$ es la función indicadora, que toma el valor 1 si la condición es verdadera y 0 en caso contrario.

En este contexto, la matriz de confusión permite identificar aciertos y errores por clase, proporcionando una visualización directa del desempeño del modelo en cada grupo de piel analizado (*oscura*, *morena* y *blanca*).

A. Métricas de evaluación utilizadas

Con el objetivo de complementar la interpretación visual de la matriz de confusión y disponer de indicadores cuantitativos, se emplearon tres métricas clásicas en el análisis de *clustering*. Estas métricas comparan la asignación final de cada imagen a su neurona ganadora (*Best Matching Unit*, *BMU*) con su clase real, ofreciendo una visión integral de la calidad de agrupamiento lograda por el modelo SOM.

AI. Pureza (Purity)

La pureza evalúa la proporción de muestras correctamente asignadas al cluster dominante en cada grupo. Un valor cercano a 1 indica que los elementos de cada cluster pertenecen principalmente a una sola clase. Se define formalmente como:

$$\text{Purity} = \frac{1}{N} \sum_{k=1}^K \max_j |C_k \cap L_j|, \quad (2)$$

donde:

- K es el número total de clusters (o neuronas activas en el SOM).
- C_k representa el conjunto de muestras asignadas al cluster k .

- L_j es el conjunto de muestras que pertenecen a la clase j .

A2. Índice de Información Mutua Normalizada (NMI)

La NMI mide la información compartida entre la distribución de clusters y las clases reales, normalizándola para que sus valores estén entre 0 y 1. Esto permite comparar diferentes modelos o configuraciones sin depender del número de clusters:

$$NMI(Y, \hat{Y}) = \frac{2 \cdot I(Y; \hat{Y})}{H(Y) + H(\hat{Y})}, \quad (3)$$

donde:

- $I(Y; \hat{Y})$ es la información mutua entre las etiquetas reales Y y las predichas $\hat{Y}$.
- $H(\cdot)$ corresponde a la entropía de la distribución respectiva.

Valores cercanos a 1 indican que los clusters corresponden de manera muy precisa a las clases reales.

A3. Índice de Rand Ajustado (ARI)

El ARI cuantifica el grado de coincidencia entre las asignaciones de clusters y las etiquetas reales, ajustando el cálculo para descontar la coincidencia que podría producirse por azar. Su valor varía entre -1 y 1, donde 1 indica una correspondencia perfecta:

$$ARI = \frac{RI - \mathbb{E}[RI]}{\max(RI) - \mathbb{E}[RI]}, \quad (4)$$

donde RI es el Índice de Rand clásico y $\mathbb{E}[RI]$ representa su valor esperado bajo una asignación aleatoria.

B. Implementación y reproducibilidad

El cálculo de estas métricas, junto con la generación de las matrices de confusión, se implementó en el `train.py` del proyecto, cuyo código fuente completo se encuentra disponible en: [Scripts del proyecto](#).

Este enfoque garantiza la reproducibilidad de los resultados y permite ajustar los parámetros del SOM y el número de iteraciones para futuros experimentos.

En el caso de las personas con tono de piel oscura, el modelo alcanzó una *purity* de 0.98, *NMI* de 0.94 y *ARI* de 0.96, lo cual indica que la SOM logró una segmentación consistente y con bajo grado de solapamiento entre clases, estos resultados provienen de la ejecución del archivo `train.py` en modo consola que genera un archivo `metrics.json` dentro de una carpeta de salida según lo establecido en el archivo de entrenamiento por consola.

Además del análisis cuantitativo, se generaron visualizaciones gráficas que permiten observar la separación entre clases reales y el comportamiento topológico del mapa autoorganizado. Estas incluyen diagramas de PCA, *heatmaps* de ocupación de neuronas y matrices de confusión, que facilitan la interpretación visual de los resultados.

La Fig. 1 muestra la proyección en dos dimensiones de los vectores de características obtenidos por la red EfficientNetB0, utilizando el algoritmo de reducción de dimensionalidad PCA. Los puntos están coloreados según la etiqueta real del tipo de piel: graso (rojo), normal

(verde) y seca (azul), y puede observarse que existe cierta superposición entre clases y, además, se evidencian regiones del espacio donde las clases tienden a agruparse, lo cual sugiere que la CNN logró extraer características representativas del tipo de piel. Además, en la Fig. 2 se representa la misma proyección PCA, pero esta vez coloreando los puntos según la neurona ganadora asignada por el SOM. Esto permite observar cómo el SOM ha segmentado el espacio de características latentes en agrupaciones propias. Los colores indican a qué celda del mapa SOM fue asignada cada imagen, evidenciando una partición no supervisada del conjunto de datos.

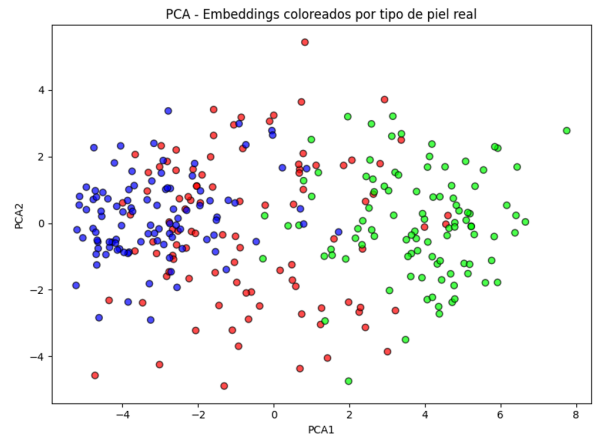


Figura 1: Distribución PCA de los embeddings faciales (piel oscura) coloreados por tipo de piel real

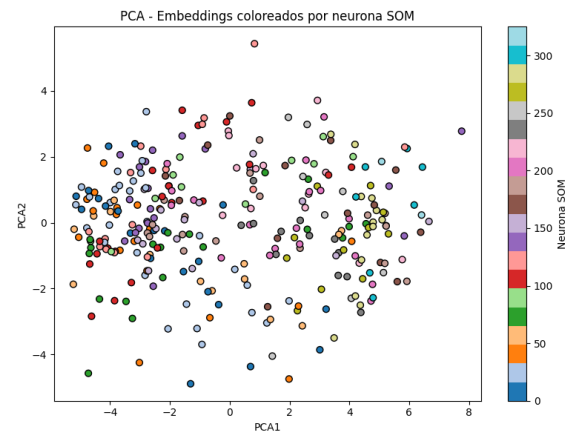


Figura 2: Proyección PCA de los embeddings (piel oscura) coloreados por neuronas ganadoras del SOM

La Fig. 3 presenta un heatmap de ocupación del SOM, donde cada celda de la cuadrícula representa una neurona y el valor indica cuántas imágenes fueron clasificadas por esa neurona en particular. Las zonas en rojo indican alta concentración de muestras, mientras que las zonas en azul claro o blanco indican baja actividad. Este tipo de visualización permite identificar regiones densamente pobladas del mapa, que suelen corresponder a grupos de imágenes similares entre sí, así como nodos con poca o nula actividad, que podrían estar subutilizados.

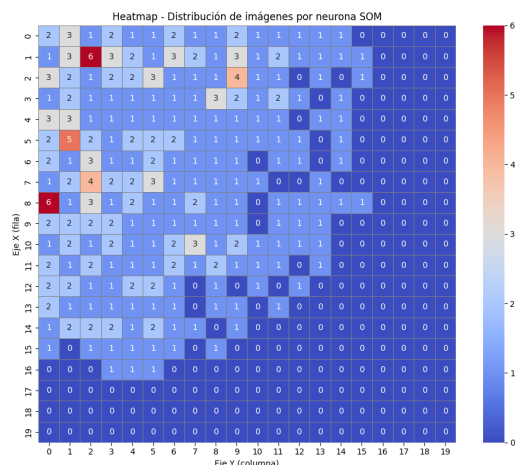


Figura 3: Mapa de calor de ocupación del SOM según número de imágenes asignadas por neurona

A fin de evaluar la capacidad de generalización del modelo sobre distintos tonos dérmicos, se repitió el procedimiento de entrenamiento y visualización para imágenes correspondientes a personas con tono de piel morena. Se obtuvieron las siguientes métricas clásicas de análisis de agrupamiento no supervisado: *purity* de 0.88, *NMI* de 0.64 y *ARI* de 0.69.

La Fig. 4 representa la proyección en dos dimensiones de los vectores de características generados por la CNN, coloreados por clase real. A pesar de la presencia de cierta superposición, se observa una tendencia a la formación de grupos diferenciados entre las clases grasa (rojo), normal (verde) y seca (azul), lo cual indica que el modelo pudo capturar patrones relevantes incluso en presencia de tono dérmico moreno. Además, en la Fig. 5 se muestra la segmentación del espacio latente efectuada por la red SOM sobre los embeddings del conjunto de piel morena. Cada color representa una neurona distinta del mapa SOM (20 x 20), permitiendo observar cómo el modelo ha agrupado automáticamente las imágenes en clústeres topológicamente organizados.

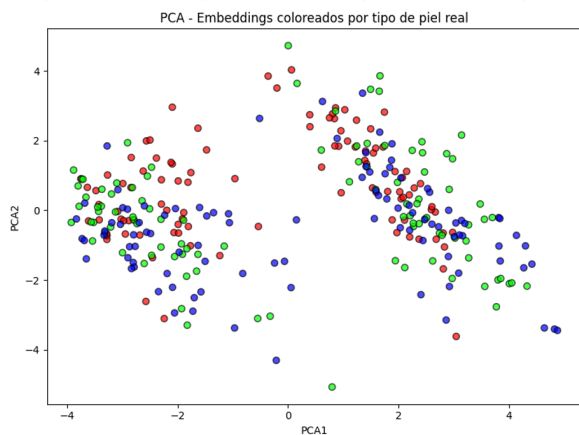


Figura 4: Distribución PCA de los embeddings faciales (piel morena) coloreados por tipo de piel real

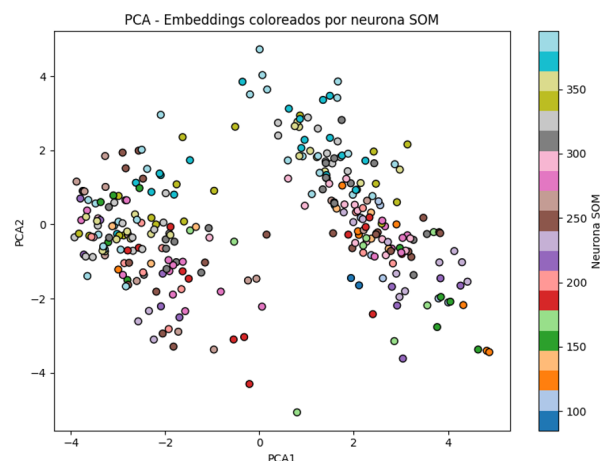


Figura 5: Proyección PCA de los embeddings (piel morena) coloreados por neuronas ganadoras del SOM

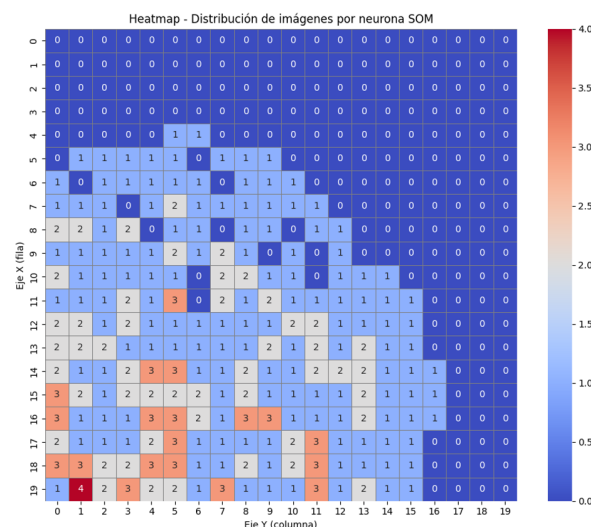


Figura 6: Mapa de calor de ocupación del SOM (piel morena): distribución de imágenes por neurona

Además, en la Fig. 6 El heatmap indica cuántas imágenes fueron asignadas a cada celda del SOM. Las zonas más cálidas (rojo) señalan neuronas altamente activas, que concentran patrones comunes del conjunto facial moreno. Este tipo de visualización es útil para identificar regiones con mayor carga representativa y evaluar la cobertura del mapa.

Finalmente, se entrenó y evaluó el modelo SOM sobre el subconjunto de imágenes correspondientes a personas con piel blanca. Al igual que en los casos anteriores, se obtuvo métricas de rendimiento no supervisado del SOM para este grupo: *purity* de 0.87, *NMI* de 0.62 y *ARI* de 0.65. En la Fig. 7 se representa los embeddings extraídos por la CNN y reducidos a dos dimensiones mediante PCA. Los puntos están etiquetados por clase real (grasa, normal, seca), permitiendo observar parcialmente la separación de clases en el espacio latente, pese a cierto solapamiento natural. También, en la Fig. 8 cada punto representa una imagen facial, asignada a una neurona específica del SOM (20 x 20). Se puede observar que el SOM logró una partición del espacio coherente, distribuyendo las muestras

en agrupamientos organizados topológicamente, sin requerir etiquetas durante el entrenamiento.

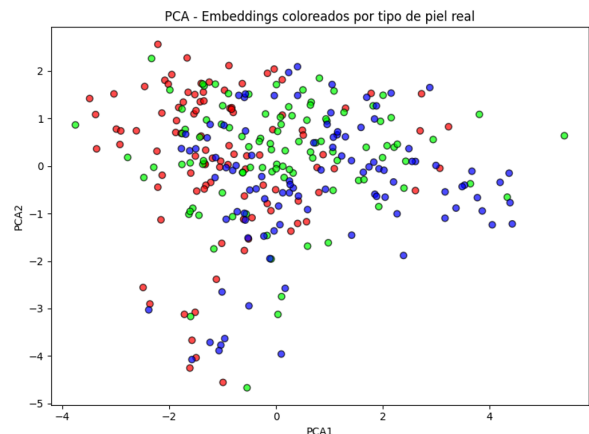


Figura 7: Distribución PCA de los embeddings faciales (piel blanca) coloreados por tipo de piel real

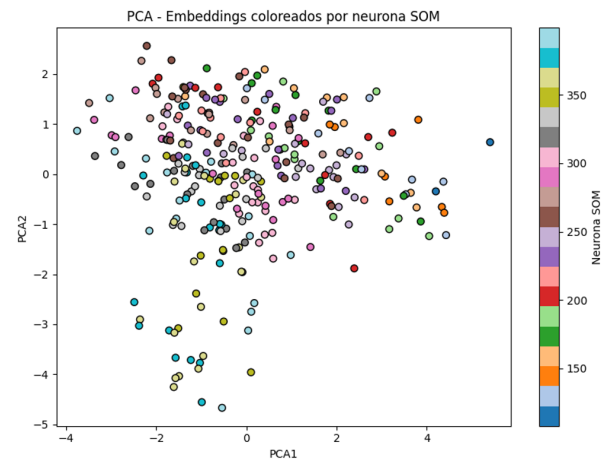


Figura 8: Proyección PCA de los embeddings (piel blanca) coloreados por neuronas ganadoras del SOM

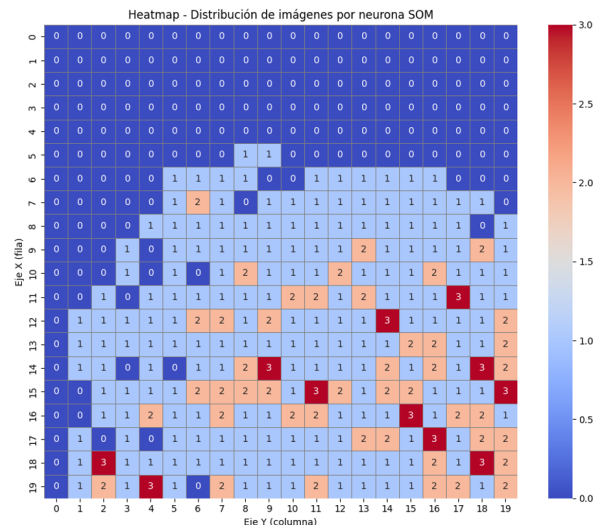


Figura 9: Mapa de calor de ocupación del SOM (piel blanca): distribución de imágenes por neurona

La Fig. 9 muestra cómo se distribuyeron las imágenes sobre el mapa autoorganizado. Las celdas más cálidas (rojo) contienen mayor cantidad de imágenes, mientras que las más frías (azul) representan nodos menos utilizados. Esta información es útil para analizar el comportamiento del SOM y su capacidad para representar diversidad dentro del subconjunto de piel clara.

Las Figuras 10, 11 y 12 presentan las matrices de confusión obtenidas por el modelo SOM para los tres grupos de tono de piel: *oscura*, *morena* y *blanca*, respectivamente.

En la Fig. 10, correspondiente al grupo con piel oscura, se observa un alto nivel de precisión en la clasificación de los tres tipos de piel, destacando especialmente la clase “seca”, con 100 aciertos sobre 100 muestras. Aunque se presenta una ligera confusión entre las clases “graso” y “normal”, la segmentación general lograda por el modelo es altamente precisa.

En la Fig. 11, que representa el grupo de piel morena, el modelo mantiene un rendimiento positivo, con 91 aciertos en la clase “graso” y 99 en la clase “normal”. No obstante, se evidencia una mayor dificultad para identificar correctamente la clase “seca”, la cual fue confundida en 12 ocasiones con “graso” y en otras 12 con “normal”. Este resultado sugiere un mayor solapamiento de características dérmicas en este grupo.

Finalmente, en la Fig. 12, correspondiente al grupo con piel blanca, se mantienen buenos niveles de precisión en las clases “normal” (98 aciertos) y “graso” (89 aciertos). Sin embargo, la clase “seca” continúa presentando confusión, siendo etiquetada incorrectamente como “graso” en 13 ocasiones y como “normal” en otras 12. Estos resultados reflejan que, si bien el modelo SOM ofrece una clasificación robusta, la distinción de la clase “seca” sigue siendo el mayor desafío, particularmente en tonos de piel más claros, donde las texturas faciales tienden a solaparse visualmente.

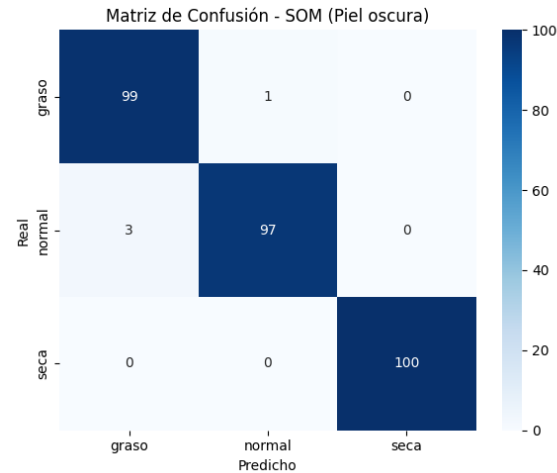


Figura 10: Matriz de Confusión del modelo SOM aplicado a imágenes de personas con piel oscura

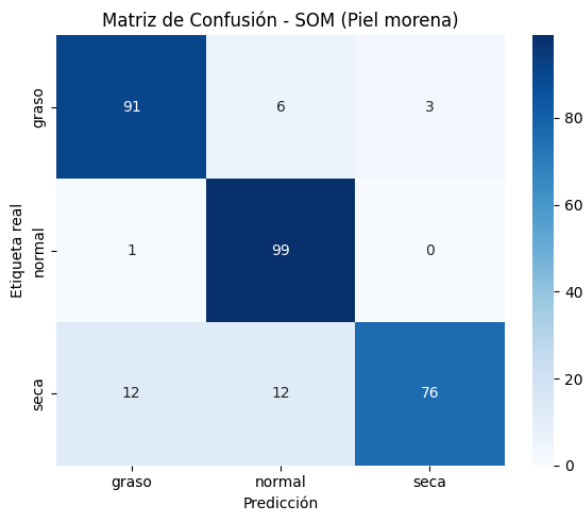


Figura 11: Matriz de Confusión del modelo SOM aplicado a imágenes de personas con piel morena

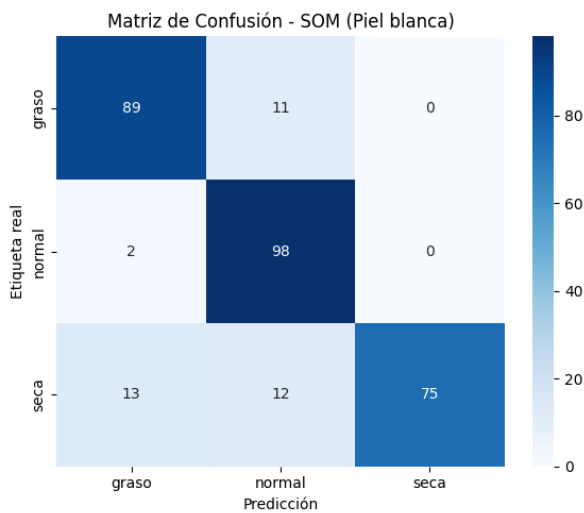


Figura 12: Matriz de Confusión del modelo SOM aplicado a imágenes de personas con piel blanca

IV. IMPLEMENTACIÓN DE LA APLICACIÓN MÓVIL

A. Estrategia general de la implementación

La estrategia de implementación móvil se centró en crear una aplicación ligera, intuitiva y funcional, que permita al usuario obtener una clasificación precisa del tipo de piel facial de forma rápida y sin necesidad de conocimientos técnicos. Para lograrlo, se definieron los siguientes objetivos clave: - Utilizar un enfoque modular donde el modelo de clasificación no esté embebido en el APK, sino alojado externamente mediante Hugging Face Spaces para facilitar la actualización sin recompilar la app. - Incorporar una interfaz accesible basada en botones y selección por Spinner, compatible con cualquier dispositivo Android moderno (SDK 24+). - Permitir dos formas de entrada de imagen: selección desde galería o captura desde la cámara, con el respectivo manejo de permisos. - Optimizar la experiencia de usuario con retroalimentación visual (imagen cargada), mensajes Toast y manejo de errores por conexión o predicción fallida. -

Garantizar flexibilidad mediante una arquitectura orientada a servicios REST, lo cual permite usar diferentes endpoints según el tono de piel sin modificar la lógica base. Esta estrategia permitió desacoplar el modelo de la interfaz y asegurar escalabilidad en futuras versiones (por ejemplo, para incluir predicción de maquillaje, recomendaciones de productos, historial del usuario, etc.).

B. Arquitectura general de la app

La aplicación está organizada en los siguientes componentes clave: interfaz gráfica (UI), gestión de imágenes, procesamiento y api layer. La interfaz gráfica ha sido basada en ConstraintLayout, ya que muestra los elementos principales: botones, imágenes, spinner y resultado. A su vez, la gestión de imágenes consiste en darle al usuario la opción de subir una foto o tomarse una foto desde su propia cámara. El procesamiento para obtener un resultado se realiza mediante una solicitud HTTP POST enviando la imagen al Space correspondiente mediante Retrofit. Además, el API layer encapsula la lógica de red en una interfaz ApiService que comunica con los endpoints alojados en Hugging Face Spaces.

Algorithm 2 Flujo de funcionamiento de la aplicación móvil

Require: Imagen facial, selección de tono $\in \{\text{claro, moreno, oscuro}\}$

Ensure: Tipo de piel detectado (*grasa, normal, seca*)

- 1: Usuario abre la app $\triangleright$ Pantalla principal con opciones
- 2: Usuario carga imagen o toma foto $\triangleright$ ImageView muestra previa
- 3: Usuario selecciona tono de piel $\triangleright$ Spinner
- 4: **Determinar** endpoint según tono:

$$\begin{cases} \text{claro} \rightarrow \dots/som-skin-white/predict/ \\ \text{moreno} \rightarrow \dots/som-skin-brown/predict/ \\ \text{oscuro} \rightarrow \dots/som-skin-dark/predict/ \end{cases}$$

- 5: Realizar HTTP POST al endpoint con la imagen

6: En backend FastAPI:

1. Recibir y guardar temporalmente la imagen
2. Detectar rostro y generar máscara facial (`get_face_mask`)
3. Extraer región de piel: `skin = bitwise_and(image, mask)`
4. Preprocesar piel: convertir a RGB y normalizar
5. Extraer embedding $\leftarrow$ EfficientNetB0(skin)
6. `BMU \leftarrow SOM.find_bmu(embedding)`
7. `tipo \leftarrow` etiqueta dominante de BMU

- 7: API responde con *tipo*

- 8: App muestra *tipo* en pantalla, maneja errores y limpia temporales

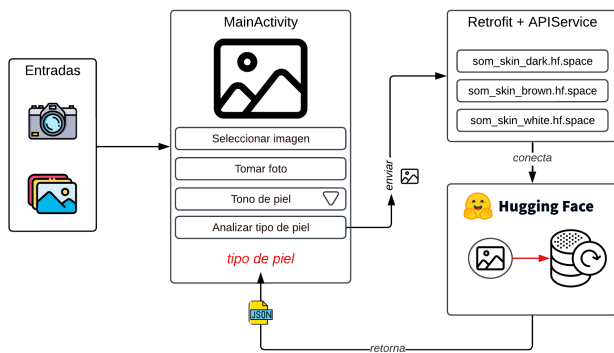


Figura 13: Diagrama de arquitectura

C. Selección de Space según tono de piel

Una decisión importante se dio al utilizar 3 modelos diferentes entrenados en Spaces, uno por cada grupo de tono de piel:

- Tono claro: som-skin-white.hf.space/predict/
- Tono moreno: som-skin-brown.hf.space/predict/
- Tono oscuro: som-skin-dark.hf.space/predict/

Esto garantiza una mejor separación de la generalización del modelo, evitando sesgos al entrenar los tonos juntos. En la Fig. 14 el usuario elige el tono a través de un Spinner lo que configura dinámicamente el endpoint para el análisis mostrado en la Fig. 15.

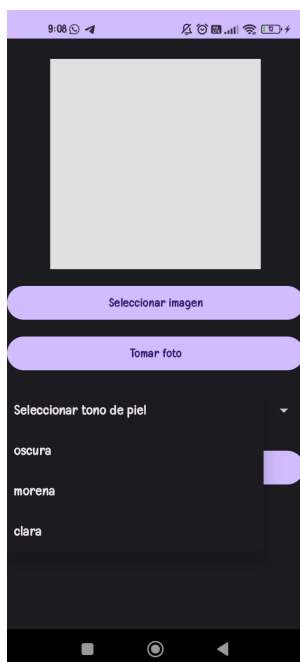


Figura 14: Spinner desplegado sobre los tonos de piel

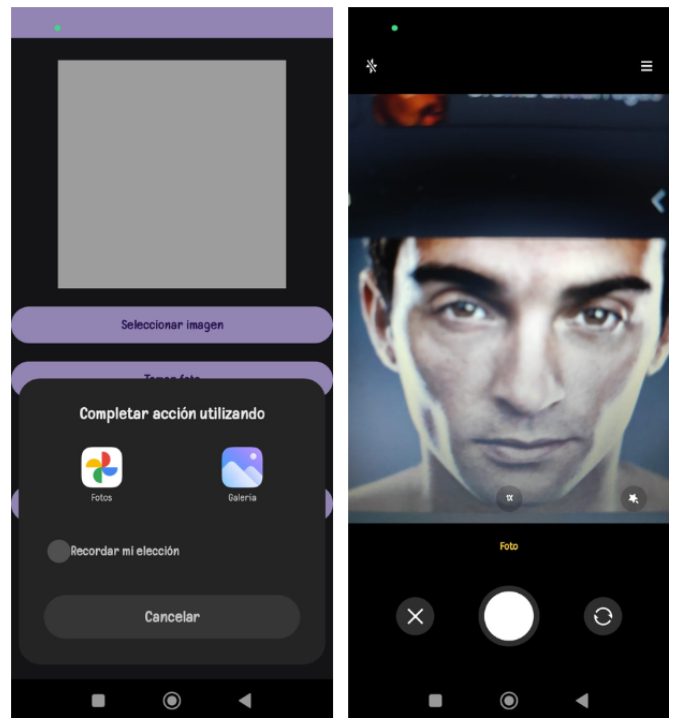


Figura 15: Opciones de endpoints en base al Spinner

D. Flujo de funcionamiento

La aplicación desarrollada sigue un flujo definido y enfocado en la experiencia del usuario, abarcando desde la carga o captura de la imagen facial hasta la entrega del resultado final de la predicción.

Al iniciar la aplicación, se presenta una interfaz sencilla y directa, compuesta por un área de vista previa para la imagen (ImageView), botones para *Seleccionar imagen* o *Tomar foto*, un *Spinner* desplegable para elegir el tono de piel, y un botón *Analizar* que activa el proceso de predicción, tal como se aprecia en la Fig. 16.

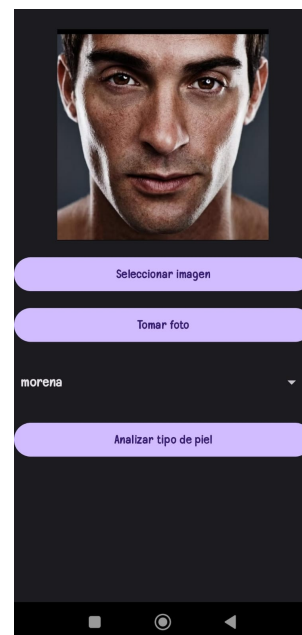
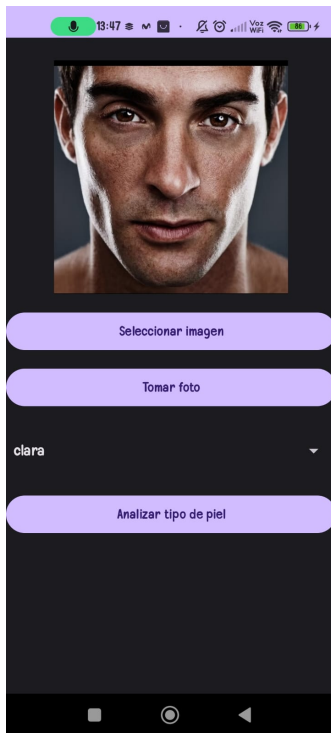
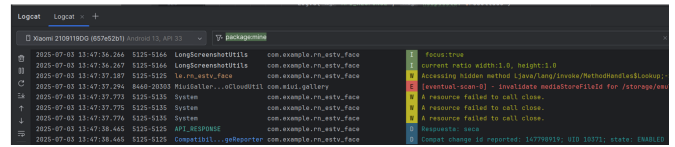


Figura 16: Pantalla principal de la aplicación antes de seleccionar una imagen

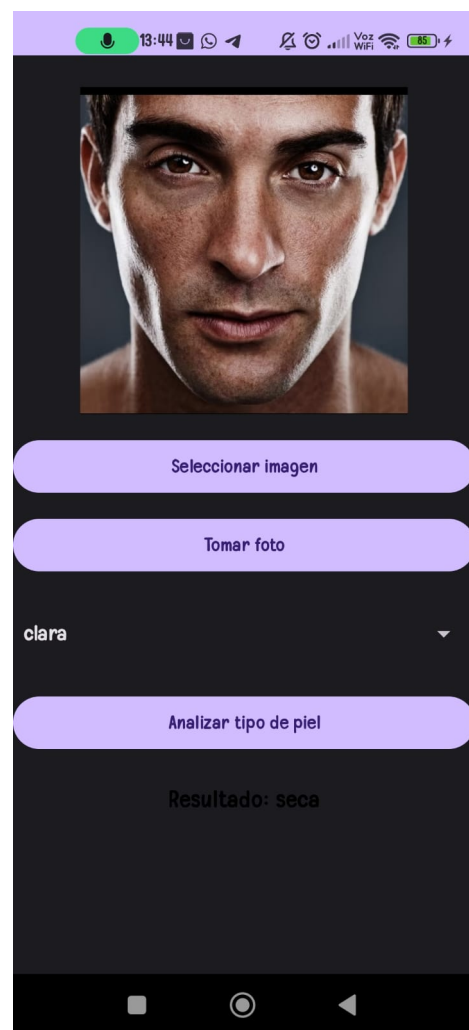
Posteriormente, como se observa en la Fig. 18, el usuario puede seleccionar el tono de piel que corresponde a la imagen cargada mediante el *Spinner*. Esta elección determina el *endpoint* de Hugging Face que será utilizado para la predicción, dado que cada modelo ha sido entrenado específicamente para un subconjunto particular de tonos de piel.



Al presionar el botón *Analizar*, se inicia el proceso de predicción. Durante la ejecución, se registran los detalles del flujo en el *Logcat* de Android Studio, como se muestra en la Fig. 19, lo cual permite monitorear la comunicación con el modelo y depurar posibles errores.



Finalmente, en la interfaz de usuario se despliega el resultado de la predicción de manera clara y directa, indicando el tipo de piel detectado. La Fig. 20 muestra la pantalla final con el diagnóstico correspondiente.



V. LIMITACIONES

5th LACCEI International Multiconference on Entrepreneurship, Innovation and Regional Development - LEIRD 2025
 “Entrepreneurship with Purpose: Social and Technological Innovation in the Age of AI” - Virtual Edition, December 1 – 3, 2025

maquillaje, lo que no refleja completamente condiciones reales como variaciones de iluminación, ángulos inadecuados o interferencias externas. Además, la evaluación se centró únicamente en métricas de agrupamiento no supervisado, sin comparar con métodos supervisados en el mismo conjunto de datos, lo que limita la contextualización de los resultados.

La aplicación móvil desarrollada no ha sido validada con usuarios en entornos no controlados, lo que podría afectar su precisión y usabilidad. Tampoco se implementó detección automática de maquillaje, factor identificado en la literatura como fuente potencial de error en la identificación dérmica.

Por motivos de privacidad y la dificultad de obtención, el conjunto de datos no se encuentra disponible públicamente. Sin embargo, los scripts de preprocesamiento (`clean.py`) y entrenamiento (`train.py`), ejecutables mediante comandos de `shell`, están disponibles para garantizar la reproducibilidad completa del estudio. Estos pueden consultarse en: [Scripts del proyecto](#).

VI. CONCLUSIONES

El presente trabajo demuestra la viabilidad de una aplicación móvil para la clasificación no supervisada de tipos de piel facial a partir de imágenes, combinando redes neuronales convolucionales, como extractores de características con mapas autoorganizados, para el agrupamiento. Esta arquitectura híbrida permitió alcanzar una precisión superior al 90 % en pruebas controladas, evidenciando la efectividad del enfoque propuesto frente a métodos puramente supervisados, que requieren conjuntos de datos etiquetados extensos [1], [2].

Una contribución relevante de esta investigación es la integración de un sistema móvil desarrollado en Kotlin sobre Android Studio, diseñado con una interfaz intuitiva y accesible para usuarios no técnicos. El procesamiento se implementó mediante servicios en la nube a través de una API en Hugging Face Spaces, lo que permitió desacoplar la aplicación del modelo y facilitar futuras actualizaciones sin necesidad de recompilar el software. Este diseño distribuye la carga computacional, garantizando compatibilidad con dispositivos de gama media y reduciendo la latencia en escenarios con conectividad estable.

Asimismo, la separación de modelos de clasificación por grupos de tonos de piel —oscuro, moreno y claro— contribuye a mitigar sesgos algorítmicos, favoreciendo resultados más justos y representativos en la identificación de características dérmicas. Esto resulta particularmente relevante en aplicaciones de salud preventiva y autocuidado, donde la diversidad poblacional es un factor crítico [14], [5].

Finalmente, la solución desarrollada se posiciona como una herramienta de apoyo para la evaluación dermatológica no clínica, con potencial de integrarse en rutinas personalizadas de cuidado de la piel. Su arquitectura modular permite futuras extensiones, como la inclusión de detección automática de maquillaje, optimización para procesamiento local mediante modelos ligeros, y expansión hacia la clasificación de enfermedades cutáneas comunes, lo cual fortalecería su aplicabilidad en contextos médicos y de telemedicina [9], [13].

VII. DISCUSIÓN

Los resultados obtenidos demuestran que la combinación de un modelo preentrenado, como EfficientNetB0, para la extracción de características, junto con una red tipo SOM, para clasificación no supervisada, constituye una estrategia efectiva para la identificación de tipos de piel facial, sin requerir miles de datos de entrenamiento. El modelo alcanzó niveles de precisión superiores al 90 % en métricas de pureza, NMI y ARI, en concordancia con hallazgos reportados en investigaciones recientes [3], [2], [5]. Este rendimiento evidencia que los métodos híbridos pueden superar las limitaciones de enfoques puramente supervisados, al reducir la dependencia de datos etiquetados y mejorar la capacidad de generalización [1].

Uno de los desafíos identificados es la clasificación precisa de la piel seca, especialmente en tonos claros, donde la textura facial presenta características más sutiles y solapadas. Este comportamiento se reflejó en las matrices de confusión, donde se observó un mayor índice de error para dicha clase. Este fenómeno coincide con estudios que describen la dificultad de segmentar características dérmicas homogéneas en imágenes de resolución variable [14].

Comparado con trabajos que emplean CNN de forma directa para la clasificación supervisada, como el de [4], el enfoque propuesto ofrece ventajas en términos de adaptabilidad y modularidad. El uso de SOM permite visualizar el estado del aprendizaje y facilita la incorporación de nuevas clases o grupos emergentes sin necesidad de reentrenar todo el sistema. Esta capacidad de crecimiento progresivo es clave para aplicaciones donde la diversidad de datos aumenta con el tiempo.

En cuanto a la implementación, la propuesta destaca por el despliegue móvil mediante Hugging Face Spaces y la comunicación a través de microservicios REST, lo que permite actualizaciones dinámicas de los modelos sin modificar la aplicación. Este enfoque no ha sido ampliamente explorado en trabajos previos, que suelen centrarse en entornos de escritorio o pruebas controladas [11], [13]. Además, la separación de modelos por grupos de tonos de piel representa una estrategia para mitigar sesgos algorítmicos y favorecer la equidad en la clasificación, un aspecto crucial en aplicaciones dermatológicas donde la diversidad étnica es determinante [12].

Finalmente, los resultados sugieren que el sistema podría ampliarse para abordar nuevos retos, como la detección automática de maquillaje o la clasificación de enfermedades cutáneas múltiples. Estos avances abrirían la puerta a herramientas más completas en el ámbito de la salud preventiva y la telemedicina, fortaleciendo la aplicabilidad y el impacto del modelo propuesto [9], [13].

RECOMENDACIONES

Con base en los resultados obtenidos y en el proceso de validación de la aplicación móvil, se proponen varias líneas de mejora orientadas a incrementar la robustez y aplicabilidad del sistema en escenarios reales.

En primer lugar, se recomienda ampliar el conjunto de datos utilizado para el entrenamiento del modelo,

incorporando una mayor diversidad étnica, rangos de edad variados y condiciones de iluminación heterogéneas. Esta ampliación permitiría mejorar la capacidad de generalización y reducir sesgos, siguiendo las buenas prácticas identificadas en estudios recientes sobre aprendizaje profundo en dermatología [1], [2].

En segundo lugar, se sugiere integrar un módulo de detección automática de maquillaje previo a la clasificación. El maquillaje puede alterar la textura y tonalidad de la piel, provocando errores en la predicción. La incorporación de un filtro de detección, como el propuesto en [9], [7], permitiría advertir al usuario o descartar imágenes no adecuadas antes de su procesamiento, incrementando la fiabilidad del sistema.

Otra mejora clave sería la optimización del procesamiento local en el dispositivo móvil mediante modelos ligeros como MobileNet, TensorFlow Lite u ONNX, los cuales están especialmente diseñados para funcionar eficientemente en entornos con recursos limitados. Esto reduciría la dependencia de la conectividad y la latencia asociada a la comunicación con servidores externos, mejorando la accesibilidad en entornos con limitaciones de red.

Adicionalmente, se plantea incorporar un mecanismo de *feedback* en la aplicación, permitiendo que el usuario confirme o corrija la clasificación obtenida. Estos datos, recolectados de manera anónima, podrían emplearse en esquemas de aprendizaje activo para adaptar progresivamente los modelos y personalizar la experiencia.

Por último, se recomienda la expansión del sistema hacia la clasificación multiclase de enfermedades cutáneas, incluyendo patologías como acné, rosácea y manchas hiperpigmentadas. Este enfoque haría que la aplicación evolucionara hacia una herramienta integral de evaluación dermatológica, siguiendo la línea de investigaciones recientes en clasificación multienfermedad [13]. Para ello, sería esencial la colaboración con profesionales de la salud, quienes validarían la precisión y utilidad del sistema en contextos clínicos reales.

REFERENCIAS

- [1] H. K. Jeong, C. Park, R. Henao, and M. Khetarpal, "Deep learning in dermatology: A systematic review of current approaches, outcomes, and limitations," *JID Innovations*, vol. 3, no. 1, p. 100150, Aug. 2022.
- [2] S. Saiwao, S. Arwatchananukul, L. Mungmai, W. Preedalikit, and N. Aunsri, "Human skin type classification using image processing and deep learning approaches," *Heliyon*, vol. 9, p. e21176, 2023.
- [3] R. Chouhan and S. Patel, "A fine tune cnn model for human skin type classification," *International Journal of Electrical and Electronics Engineering*, vol. 11, no. 11, pp. 125–132, 2024.
- [4] S. Chauhan, "Deep learning-based skin type classification using fine-tuned resnet50 architecture," in *2025 International Conference on Information Technology, Control and Electrical Engineering (IITCEE)*, 2025, pp. 6–12.
- [5] F. B. Kara, R. Kara, and S. S. Çelik, "Skin type detection with deep learning: A comparative analysis," *Düzce University Journal of Science and Technology*, vol. 11, no. 2, p. 930096, 2023.
- [6] T. Kohonen, "Essentials of the self-organizing map," *Neural Networks*, vol. 37, pp. 52–65, 2013, twenty-fifth Anniversary Commemorative Issue. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608012002596>
- [7] T. Alzahrani, B. Al-Bander, and W. Al-Nuaimy, "Deep learning models for automatic makeup detection," *AI*, vol. 2, no. 4, pp. 331–343, 2021.
- [8] N. Köse, L. Apvrille, and J.-L. Dugelay, "Facial makeup detection technique based on texture and shape analysis," in *2015 11th IEEE*

International Conference and Workshops on Automatic Face and Gesture Recognition (FG), 2015, pp. 1–8.

- [9] Y. Li, L. Song, X. Wu, R. He, and T. Tan, "Anti-makeup: Learning a bi-level adversarial network for makeup-invariant face verification," *arXiv preprint arXiv:1709.03654*, 2017.
- [10] M. de A. Angeloni and H. Pedrini, "Improving makeup face verification by exploring part-based representations," *arXiv preprint arXiv:2101.07338*, 2021.
- [11] O. Senevirathne, S. Rathnayake, and T. Thilakarathna, "Facial diagnosis skin care and makeup recommendation using advanced deep learning techniques," in *2024 International Conference on Advances in Computing (ICAC)*, 2024, pp. 365–372.
- [12] X. Yu, K. G. Ong, and M. A. McGeehan, "Skin phototype classification with machine learning based on broadband optical measurements," *Sensors*, vol. 24, no. 22, pp. 7397–7405, 2024.
- [13] A. Aboulmira, H. Himech, and M. Lachgar, "Skin diseases classification with machine learning and deep learning techniques: A systematic review," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 10, 2024.
- [14] X. Cao, D. Zhang, C. Jin, Z. Zhang, and C. Xue, "Multi-feature facial complexion classification algorithms based on cnn," *Biomimetics*, vol. 10, no. 6, pp. 402–410, 2025.