

Reporte Individual 3

Background Removal

Manzano Estrada Vanessa Nataly*, Medina Cavazos Luis Alejandro* y Moreno Ledesma Ximena Abigail*
Ingeniería en Tecnologías de la Información
Universidad Politécnica de Victoria

Resumen—En este documento se presenta el desarrollo de una aplicación Android que realiza segmentación en tiempo real de personas en imágenes y video, utilizando CameraX para la captura de la cámara, ML Kit Selfie Segmentation para generar máscaras de segmentación, y Jetpack Compose para la interfaz de usuario. La aplicación permite seleccionar una imagen desde la galería y, posteriormente, combinarla con la persona detectada en la cámara frontal. Se realizaron pruebas para validar que la segmentación y el reemplazo de fondo funcionen correctamente, mostrando la persona en primer plano sobre el fondo seleccionado.

I. INTRODUCCIÓN

La segmentación de imágenes es una técnica fundamental en el campo de la visión por computadora, permitiendo la identificación y separación de objetos dentro de una imagen. En aplicaciones móviles, esta tecnología se utiliza para mejorar la experiencia del usuario mediante efectos visuales avanzados, como el reemplazo de fondos en tiempo real. En este contexto, Google ha desarrollado ML Kit, un conjunto de herramientas de aprendizaje automático que facilita la implementación de funcionalidades como la segmentación de selfies en dispositivos Android. Esta API permite separar al usuario del fondo en tiempo real, utilizando modelos optimizados para dispositivos móviles [1].

Por otro lado, Jetpack Compose es un moderno kit de herramientas de UI para Android que simplifica la creación de interfaces de usuario declarativas y reactivas. Su integración con CameraX, una biblioteca que facilita el uso de la cámara en aplicaciones Android, permite el desarrollo de aplicaciones con capacidades avanzadas de captura y procesamiento de imágenes [2].

Este proyecto tiene como objetivo el desarrollo de una aplicación Android que combine ML Kit, CameraX y Jetpack Compose para realizar segmentación en tiempo real de personas en imágenes y video. La aplicación permite seleccionar una imagen de fondo desde la galería y, posteriormente, mostrarla detrás de la persona detectada en la cámara frontal. Se implementa un flujo de procesamiento que convierte los frames de la cámara de formato YUV a Bitmap RGB, aplica la segmentación y renderiza el resultado en tiempo real [1]. Además, se utiliza Jetpack Compose para la construcción de la interfaz de usuario, aprovechando sus ventajas en términos de declaratividad y reactividad [3].

La herramienta desarrollada ofrece una experiencia interactiva, permitiendo al usuario visualizar en tiempo real cómo su imagen se combina con diferentes fondos seleccionados.

Se realizaron pruebas para validar que la segmentación y el reemplazo de fondo funcionen correctamente, mostrando la persona en primer plano sobre el fondo seleccionado sin errores.

II. DESARROLLO EXPERIMENTAL

Para la implementación del proyecto, se desarrolló una aplicación móvil en **Kotlin**, utilizando **Jetpack Compose** para la interfaz gráfica, junto con **CameraX** y **ML Kit de Google** para aplicar segmentación en tiempo real con efectos de fondo virtual.

Inicialmente se exploró el uso de **Fritz AI**, siguiendo el tutorial “Image Segmentation for Android — Smart Background Replacement with Fritz AI” [4]. Dicho recurso proponía el uso del Fritz SDK para generar máscaras segmentadas y reemplazar fondos en imágenes, sin embargo, se determinó que las librerías utilizadas estaban obsoletas y sin soporte, lo que imposibilitaba su uso en entornos actuales de Android.

Ante esta limitación, se consultó un tutorial en Python sobre segmentación en tiempo real [5], el cual sirvió de inspiración para la lógica general. Debido a que estaba enfocado en otro lenguaje, se realizó la adaptación completa hacia **Kotlin** y **Jetpack Compose**, logrando implementar el procesamiento en dispositivos móviles Android.

II-A. Fase 1 – Selección de Imagen de Fondo

Se implementó la selección de imagen mediante `ActivityResultContracts.GetContent()`, permitiendo al usuario elegir un archivo de tipo `image/*`. La imagen fue convertida a un objeto `Bitmap` en formato **ARGB_8888**, asegurando compatibilidad con los procesos posteriores.

Este procedimiento se apoyó en prácticas descritas en literatura especializada sobre programación móvil [6].

II-B. Fase 2 – Gestión de Permisos

Antes de iniciar la cámara, fue necesario implementar la gestión de permisos en tiempo de ejecución. Para ello se utilizó `checkSelfPermission()` y, en caso de ser necesario, se solicitó el permiso de cámara mediante `ActivityResultContracts.RequestPermission()`.

El correcto manejo de permisos se documenta ampliamente en textos de desarrollo Android [6], reforzando la importancia de la seguridad y la interacción usuario-sistema en aplicaciones móviles.

II-C. Fase 3 – Captura de Imágenes con CameraX

Para la captura de imágenes se integró **CameraX**, utilizando `ImageAnalysis` con la estrategia `STRATEGY_KEEP_ONLY_LATEST` a fin de procesar únicamente el frame más reciente y optimizar rendimiento. El análisis de cámara se ejecutó en un `Executor` independiente, evitando bloqueos en la interfaz.

La arquitectura recomendada para este tipo de integración se encuentra descrita en guías de desarrollo con Kotlin [7].

II-D. Fase 4 – Conversión YUV a RGB

Las imágenes capturadas por la cámara se reciben en formato **YUV**, el cual no es adecuado para procesamiento directo en **ML Kit**. Para resolverlo, se implementó una clase utilitaria `YuvToRgbConverter`, que transforma YUV a `Bitmap` RGB mediante `YuvImage` y `compressToJpeg`.

Este proceso de conversión se apoyó en fundamentos descritos en bibliografía de visión por computadora aplicada a Android [8].

II-E. Fase 5 – Segmentación y Composición de Imágenes

La segmentación en tiempo real se realizó con **ML Kit Selfie Segmentation** en modo `STREAM_MODE`. El modelo generó una máscara probabilística, la cual fue utilizada para combinar dinámicamente el primer plano (usuario) y el fondo elegido:

- Si el valor de la máscara era mayor a 0.5, se mantenía el píxel de la cámara.
- En caso contrario, se utilizaba el píxel del fondo.

Este tipo de integración práctica entre **ML Kit** y Kotlin ha sido explorada en distintos textos [9].

II-F. Fase 6 – Visualización con Jetpack Compose

El resultado de la segmentación se desplegó en la interfaz de usuario mediante el componente `Image` de **Jetpack Compose**. Se utilizaron mecanismos de estado (`remember`, `mutableStateOf`) para refrescar dinámicamente la vista con cada nuevo frame procesado.

El uso de **Jetpack Compose** como motor de interfaz declarativa se fundamentó en guías modernas de desarrollo UI en Android [7, 10].

III. RESULTADOS

Tras integrar cada uno de los módulos principales del sistema, incluyendo la captura de imágenes mediante la cámara del dispositivo, la selección de fondos personalizados desde la galería y la aplicación de segmentación de personas, se obtuvo una aplicación funcional que cumple con los objetivos planteados.

En la **Figura 1** se observa la primera interacción con la aplicación, donde el sistema solicita al usuario los permisos necesarios para el uso de la cámara y el acceso a los archivos multimedia del dispositivo. Este paso asegura el correcto funcionamiento de las funciones de captura y almacenamiento.

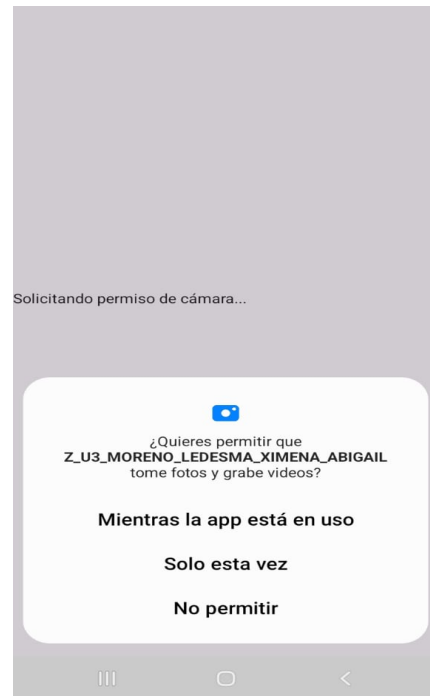


Figura 1: Solicitud de permisos para el uso de cámara y galería.

Una vez concedidos los permisos, la aplicación presenta la opción de personalizar el entorno visual mediante un botón para seleccionar un fondo **Figura 2**. Al activarlo, se despliega la galería del dispositivo, donde la persona puede elegir entre diferentes imágenes disponibles **Figura 3**.

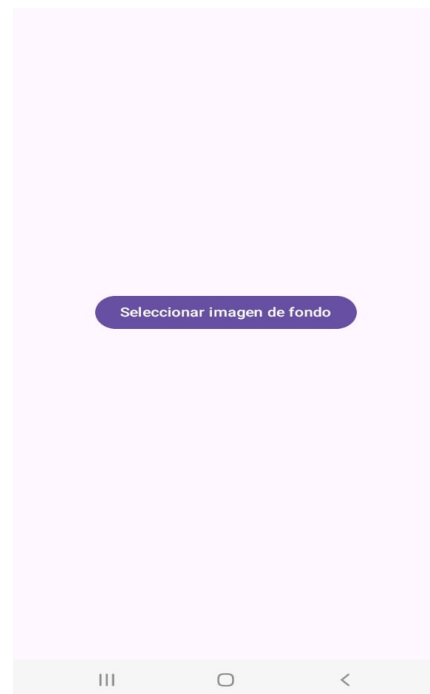


Figura 2: Botón para seleccionar imagen de fondo.

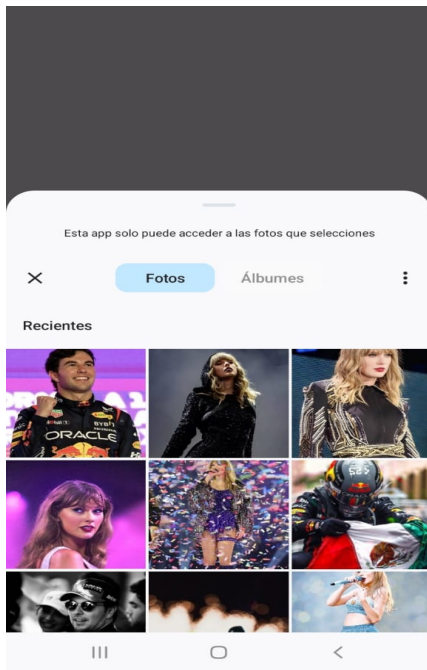


Figura 3: Vista de la galería de imágenes para elegir fondo.

Posteriormente, el sistema procesa la fotografía capturada, aplicando segmentación para aislar a la persona del fondo original. En la **Figura 4** se muestra el resultado final, donde la persona permanece en primer plano mientras se sustituye el fondo por la imagen seleccionada de la galería. Este procesamiento se realiza en tiempo real, garantizando una experiencia fluida.

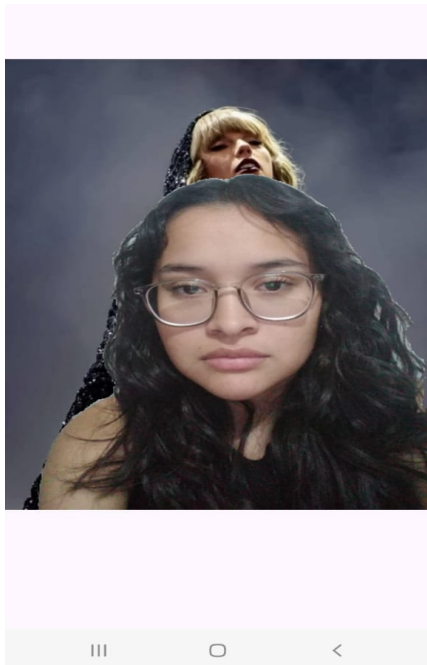


Figura 4: Resultado de segmentación con la persona sobre el fondo seleccionado.

IV. CONCLUSIÓN

En este trabajo se desarrolló una aplicación Android capaz de realizar segmentación de personas en tiempo real y reemplazo dinámico de fondo usando CameraX, ML Kit Selfie Segmentation y Jetpack Compose. En un inicio, se intentó usar el tutorial de Fritz AI, pero presentó dificultades por SDK desactualizado, dependencias incompatibles y falta de compatibilidad con interfaces modernas. Por ello, se tomó como referencia un video de Python sobre remoción de fondo con OpenCV, adaptando su lógica a Android y Compose.

La aplicación resultante permite seleccionar un fondo desde la galería y combinarlo en tiempo real con la persona capturada por la cámara. Cada frame se convierte de YUV a Bitmap RGB, se segmenta la persona y se renderiza sobre el fondo elegido. Haber trabajado con Jetpack Compose facilitó la creación de la interfaz, reduciendo código repetitivo, mejorando la modularidad y reutilización de componentes.

Las pruebas confirmaron que la segmentación y el reemplazo de fondo funcionan correctamente, mostrando el potencial de esta solución para efectos visuales, videollamadas y aplicaciones de realidad aumentada.

REFERENCIAS

- [1] *ML Kit Selfie Segmentation on Android*. <https://developers.google.com/ml-kit/vision/selfie-segmentation/android>. Accedido: 2025-08-15.
- [2] *CameraX Overview*. <https://developer.android.com/jetpack/androidx/releases/camera>. Accedido: 2025-08-15.
- [3] *Jetpack Compose Overview*. <https://developer.android.com/jetpack/compose>. Accedido: 2025-08-15.
- [4] Fritz AI. *Image Segmentation for Android — Smart Background Replacement with Fritz AI*. <https://fritz.ai/image-segmentation-for-android/>. Accedido: 2025-08-15. 2020.
- [5] YouTube. *Image Segmentation Tutorial (Python)*. <https://www.youtube.com/watch?v=dt2L8x1J7Ps>. Accedido: 2025-08-15. 2021.
- [6] Bill Phillips, Chris Stewart y Kristin Marsicano. *Android Programming: The Big Nerd Ranch Guide*. 5.^a ed. Big Nerd Ranch, 2019. ISBN: 9780135245125.
- [7] Josh Skeen y David Greenhalgh. *Kotlin Programming: The Big Nerd Ranch Guide*. Big Nerd Ranch, 2019. ISBN: 9780135161630.
- [8] Joseph A. Singh. *Hands-On Computer Vision on Android with ML Kit*. Packt Publishing, 2021. ISBN: 9781800568124.
- [9] Earl Brown y Chintan K. *Machine Learning with Kotlin for Android Developers*. Packt Publishing, 2020. ISBN: 9781838647930.
- [10] Ray Wenderlich Tutorial Team. *Jetpack Compose by Tutorials*. Razeware LLC, 2021. ISBN: 9781950325425.