uRe-touch

Se trata de una App Android que permitirá editar cualquier foto de una forma fácil e intuitiva. Esta aplicación será capaz de aplicar desde una sencilla corrección de luz y sombra hasta un filtro de lo más profesional para darle a cada foto un toque personal y por tanto único. Nos encargaremos de que no sea necesaria ninguna guía para utilizarla ya que cuenta con una interfaz muy sencilla y minimalista que hará que la edición sea completa, clara y concisa.

FUNCIONALIDADES DE LA APP

- **Funciones básicas:**
  - Exposición
  - Brillo
  - Sombras (incluyendo punto negro: es el valor de sombra más profundo en ella)
  - Contraste
  - Color (sombras y luces)
  - Saturación
  - Temperatura
  - Nitidez
  - Degradado
  - Granulado
    - Uniforme
    - Gaussiano
  - Recorte
    - Original
    - Libre
    - 9:16
    - 5:5
  - Rotación (enderezar la imágen)

- **Funciones avanzadas:**
  - Bordes (externo, interno, radio)
  - Doble exposición
  - Edición selectiva con radio ajustable (brillo y contraste)
  - Borrar un objeto que previamente se ha seleccionado de la imagen
  - Desenfoque
    - Filtros
      - Desenfoque de campo
      - Desenfocar iris
      - Cambio de inclinación
      - Desenfoque de trazado
      - Desenfoque de giro
      - Desenfoque gaussiano
    - Ajuste
      - Cantidad de filtro del 0.0 al 10.0
      - Restaurar el ruido en las áreas desenfocadas
        - Uniforme
        - Gaussiano
  - Color filtros
    - Filtros
    - Ajuste (cantidad de filtro)
  - Blanco y negro
    - Filtros
      - B1
      - B2
      - B3
      - B4
      - B5
      - B6
    - Ajuste
      - Cantidad de filtro de 0.0 a 10.0
  - Vintage
    - Filtros
      - AU1
      - AU5
      - AV4
      - AV8

        - B&N
    - Ajuste
        - Cantidad de filtro de 0.0 a 10.0
        - Luces
        - Texturas
- Ojo de pez
    - Ajustes
        - Cantidad de filtro de 0.0 a 10.0
        - Radio
- Eliminar el fondo de la imagen

LIBRERÍAS

- Para la selección de imágenes de la galería:
    - **Álbum**: https://github.com/yanzhenjie/Album
        - Permite seleccionar imágenes, videos o ambas.
        - Tomar una foto, grabar un video o usar la cámara en la lista de álbumes.
        - Obtener una vista previa de imágenes y videos en la galería
    - **Louvre**: https://github.com/andremion/Louvre
        - Biblioteca de soporte de diseño: El paquete de diseño proporciona API para admitir la adición de patrones y componentes de diseño de materiales a sus aplicaciones.
        - CounterFab: Una subclase FloatingActionButton que muestra una insignia de contador en la esquina superior derecha.
        - Glide: Una biblioteca de carga y almacenamiento en caché de imágenes para Android centrada en el desplazamiento suave
        - PhotoView: Implementación de ImageView para Android que admite el zoom mediante varios gestos táctiles.
    - **MultiType FilePicker**: https://github.com/fishwjy/MultiType-FilePicker
    - **Glide** (recomendada por google): https://github.com/bumptech/glide

- **Photo editor SDK**: https://github.com/imgly/pesdk-android-demo
    - Dibujar en la imagen con la opción de cambiar el color, el tamaño, la opacidad, el borrado y las formas básicas de su pincel.
    - Aplicar efecto de filtro en la imagen usando MediaEffect
    - Adición/edición de texto con la opción de cambiar su color con fuentes personalizables
    - Adición de emojis con fuentes de emojis personalizadas.
    - Adición de imágenes/pegatinas
    - Pellizcar para escalar y rotar ‘Views’.
    - Deshacer y Rehacer para Pincel y ‘Views’.
    - Eliminación de ‘Views’
    - Guardar foto después de editar.

- **GPUImage for Android**: https://github.com/cats-oss/android-gpuimage
    - El objetivo es tener algo tan similar a GPUImage como sea posible
    - Mirar funcionalidades en el enlace, tiene casi todos los filtros básicos y bastantes de las funciones extendidas que quiero

- **Photo Filter**: https://github.com/mukeshsolanki/photofilter

    - Funcionalidades en el enlace, tiene varios filtros pero la veo menos completa que la anterior, quizás podría complementar a la anterior para los filtros que faltan

- **OpenCV java for android**: https://opencv.org/android/
    - No encuentro del todo claras las funcionalidades que puede llegar a incluir por muy completa y potente que sepa que es (quizás está más enfocada a la detección de caras y objetos de las imágenes etc)

- **android-imaging-utils**: https://android-arsenal.com/details/1/6047
    - Una forma sencilla de usar opencv en la aplicación sin tener que preocuparse por las bibliotecas nativas y ndk. Esta biblioteca empaqueta las bibliotecas opencv nativas y expone la API de Java a los usuarios. La biblioteca también incluye clases de ayuda simples para la cámara y la grabación de video.

- **Cekrek**: https://android-arsenal.com/details/1/8141
    - Exportar o generar un 'Bitmap' desde una 'View' sin necesidad de mostrarlo.
    - Exportar o generar una 'Image File' desde una 'View' sin necesidad de mostrarlo.
    - Generador de 'Image File' y 'Bitmap' configurable.
    - Método amigable para usuarios de Java.

- **ImageWorker**: https://android-arsenal.com/details/1/7370
    - Sirve para guardar, recuperar y convertir las imágenes que seleccionemos. Hacia y desde el almacenamiento externo.
    - Guardar 'Bitmap', 'Drawable' y Cadena codificada en 'Base64' directamente en el almacenamiento externo. En los siguientes formatos *.png, *.jpeg y *.webp.
    - Recuperar archivos guardados y utilizarlos como 'Bitmaps'.
    - Convierta 'Bitmap' a 'Drawable' o 'Base64' y viceversa.

- **auto-background-remover:** https://github.com/GhayasAhmad/auto-background-remover?ref=androidexample365.com

He destacado las que considero más importantes y útiles para este proyecto pero, en caso de necesitar alguna otra funcionalidad que no aparece en el listado de librerías anterior, aquí hay más: https://android-arsenal.com/tag/47

PROYECTO PARECIDO

https://github.com/DrMint/Litrato

Para las 4 primeras librerías (Álbum, Louvre, MultiType FilePicker) no haría falta tabla porque son librerías que incluyen solo la funcionalidad de seleccionar imágenes, buscar info de ellas y elegir una (justificar, pej: la más clara e intuitiva, minimalista)

Las últimas tres librerías (android-imaging-utils, Cekrek e ImageWorker) las usaré para la implementación de dichas funcionalidades (trabajar con BitMaps etc).

FUNCIONALIDADES BÁSICAS

	Photo editor (SDK)	GPUImage for Android	Photo Filter	OpenCV java for Android
Exposición	X	X	X	
Brillo		X	X	
Sombras		X		
Contraste	X	X	X	
Color (sombras, luces)				
Saturación	X	X	X	
Temperatura			X	
Nitidez		X	X	
Degradado				
Granulado uniforme			X	
Granulado gaussiano				
Recorte	X		X	
Rotación	X		X	X

FUNCIONALIDADES AVANZADAS

	Photo editor (SDK)	GPUImage for Android	Photo Filter	OpenCV java for Android
Bordes (externo, interno, radio)	X			
Borrar objetos de la imagen				
Filtros desenfoque		X (Desenfoque gaussiano, de caja, bilateral, de zoom)		
Filtros color	X (60 filtros disponibles y la opción de diseñar un filtro personalizado)	X		
Filtros blanco y negro	X	X	X	
Filtros vintage	X			
Ojo de pez			X	

En cuánto a la doble exposición (creo) que no es necesaria ninguna librería (se umbraliza una de las imágenes y se usa de máscara para la otra, añadiendo varias imágenes con distintos pesos que podremos ajustar mediante la interfaz).

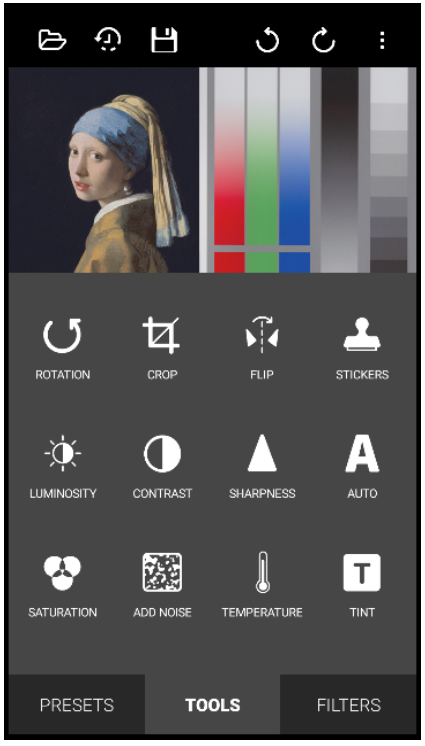
Para la edición selectiva con radio ajustable no encuentro nada aún, creo que habrá que encontrar la manera de aplicar las herramientas que proporcionan las librerías (brillo, contraste) solamente en las zonas que el usuario seleccione.

Para borrar objetos de la imagen tampoco he encontrado nada claro aún, necesito buscar más adelante información más detallada

PROYECTO PARECIDO

	Proyecto “Litrato”
Exposición	
Brillo	X
Sombras	
Contraste	X
Color (sombras, luces)	
Saturación	X
Temperatura	X
Nitidez	X
Degradado	
Granulado uniforme	X (con las opciones de granulado a color o en byn)
Granulado gaussiano	
Recorte	X
Rotación	X
Bordes (externo, interno, radio)	
Borrar objetos de la imagen	

Filtros desenfoque	X (Average blur, gaussian blur, directional blur)
Filtros color	X (2 strip, invert, bleach bypass, candle light, crisp warm. crisp winter, drop blues, tension green, edgy amber, night from day, late sunset, futuristic bleak, soft warming, colorize, change hue, selective coloring, hue shift, threshold, posterize, laplacian, sobel, sketch, cartoon)
Filtros blanco y negro	X (solamente un filtro para byn)
Filtros vintage	X (solamente un filtro: old analog)
Ojo de pez	



La interfaz es bastante sencilla e intuitiva, separa ‘Presets’, ‘Tools’ y ‘Filters’. Las herramientas de gestión de la app se encuentran en la parte superior y son:

- Elegir imagen de la galería o realizar una nueva
- Ver el historial de ediciones y poder volver a cualquier versión anterior
- Guardar imagen
- Deshacer cambio
- Rehacer cambio
- Ajustes, con las opciones de:
    - Establecer o no el modo oscuro
    - Elegir la resolución de la imagen cargada (200, 350, 500, 750, 1000,1200, 1500, 2000).
    - Marcar si queremos o no guardar la imagen con la resolución original (en vez de con la resolución de la imagen cargada del punto anterior)
    - Marcar si queremos o no que se muestre automáticamente el histograma

Considero que contiene todas las herramientas de gestión que inicialmente quiero que tenga mi aplicación, siendo muy importante e interesante la última (mostrar el histograma de la imagen mientras se está editando). Además, al aplicar un filtro puede elegirse (a mano, con una “brocha”) en qué partes de la imagen aplicarlo (pudiendo ajustar el radio y opacidad de dicha brocha).

12 - 1 - 2023

He implementado un menú en ‘MainActivity.java’ en el que existen dos opciones:

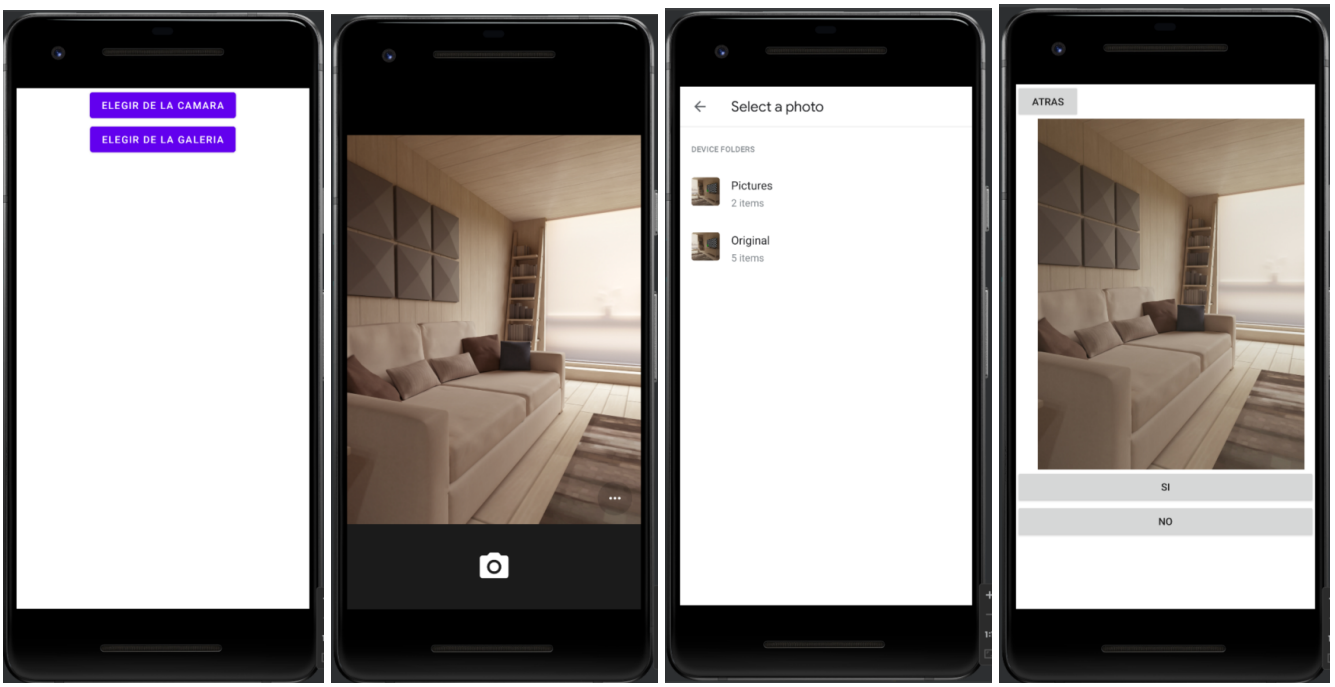
- **Elegir una instantánea que podemos capturar en el momento.**

  Para ello, he implementado una función ‘abrirCamara()’ que abrirá la cámara a través de un Intent y generará la ruta de la imagen capturada almacenándola en la variable ‘imagenCamara’.
  En el resultado de esta actividad (‘requestCode = 1’) enviaremos dicha ruta a través de un Intent y ‘bundle’ (herramienta necesaria para que funcione correctamente) a la siguiente actividad, ‘ShowImage’

- **Elegir una imagen de la galería realizada anteriormente.**

  Para ello, he implementado una función ‘abrirGaleria()’ que abrirá la galería dándonos la posibilidad de elegir una imagen de esta.
  En el resultado de esta actividad (‘requestCode = 2’) recogeremos la imagen seleccionada en formato Uri (almacenada en ‘data’ recibido como argumento). Obtenemos en formato Bitmap (almacenado en ‘imagenGaleria’) dicha imagen y la guardamos en disco. Por último, enviaremos a ‘ShowImage’ imagenGaleria a través de un Intent y ‘bundle’.

Cuando ya tenemos la imagen que deseamos editar, podemos confirmar que será la elegida o volver atrás.



14 - 1 - 2023

He incorporado al proyecto dos librerías que pueden ser útiles para empezar con las funcionalidades básicas.

- **FilterLibrary** (https://github.com/hgayan7/FilterLibrary)
  Tiene varios filtros (numerados del 1 al 16), no sé si lo usaré para las funcionalidades básicas o para las avanzadas (cuando ya empiece a aplicar filtros). Para incorporarla, he añadido:

  En build.gradle (TFG):

```
allprojects
{
        repositories
        {
                maven { url 'https://jitpack.io' }
        }
}
```

  En build:gradle (app):

```
dependencies
{
   implementation 'com.github.hgayan7:FilterLibrary:0.1.0'
}
```

  La forma de aplicarlo luego en el .java sería:

```
imgView.setImageBitmap(photoFilter.three(getApplicationContext(),imgBitmap));
```

- **GPUImage for Android** (https://github.com/cats-oss/android-gpuimage)
  Va a ser con la librería con la que creo que voy a empezar a desarrollar las funcionalidades básicas de la app (se ve en la tabla que realicé al principio que puede darme bastantes funcionalidades que quiero para mi app).
  No me ha funcionado como lo especifican en github, he tenido que cambiar la versión (2.x.x por 2.1.0) para que funcione correctamente.

  En build.gradle (TFG) añadimos a repositorios:

```
mavenCentral()
jcenter()
```

  En build:gradle (app):

```
implementation 'jp.co.cyberagent.android:gpuimage:2.1.0'
```

  Por último, ejecutar: gradle clean assemble