Manage Artworks

Android App Documentation

By: Ayush Tiwari

Date: 30 December 2021

Project Objective

Creating a Native Android Application that can provide the following functionalities for accessible artwork management:

1) Pdf Viewer from local storage

2) Image Comparision using SSIM
3) Image Object Detection using Tensorflow
4) Python Support for Android

Tech Stack Used

- Java Version 11
- Gradle Version 7.0.2
- Android Studio Arctic Fox 2020.3.1
- Chaquopy Gradle Version 10.0.1
- Python 3.9.5
- PDFTron SDK Version 9.1.1
- Facebook Fresco Version 2.6.0
- Tensorflow Lite Version 0.1.0

Setting up Project

Github Repository: LINK

Prerequisites: Android Studio 2020.3.1 Patch 4 +, git, Python 3.9 +, pip 21.3.1 +

Step 1) Open the Terminal and clone the project from the above repository.

```
git clone https://github.com/ayush-tiwari26/ManageArtworks.git
```

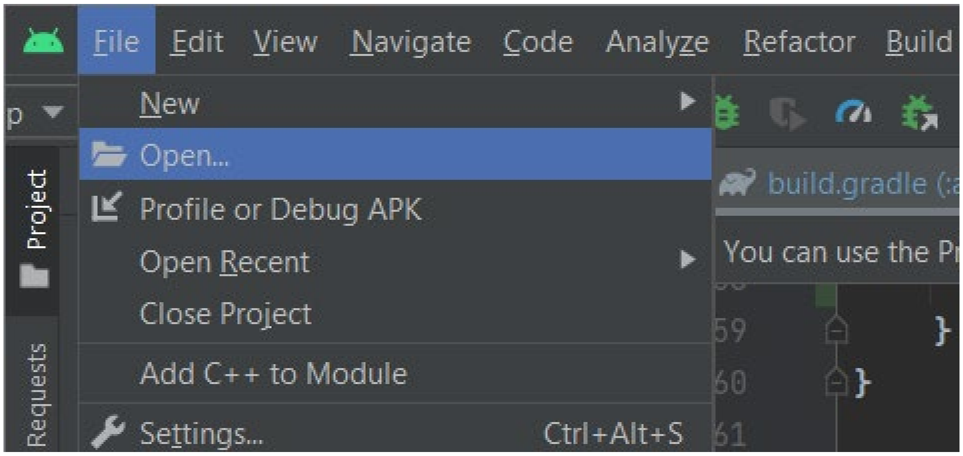
Note (Optional): Gradle sync can be done at this step by running the command below in the root directory (Usually takes lesser time than done using GUI)

```
./gradlew
```

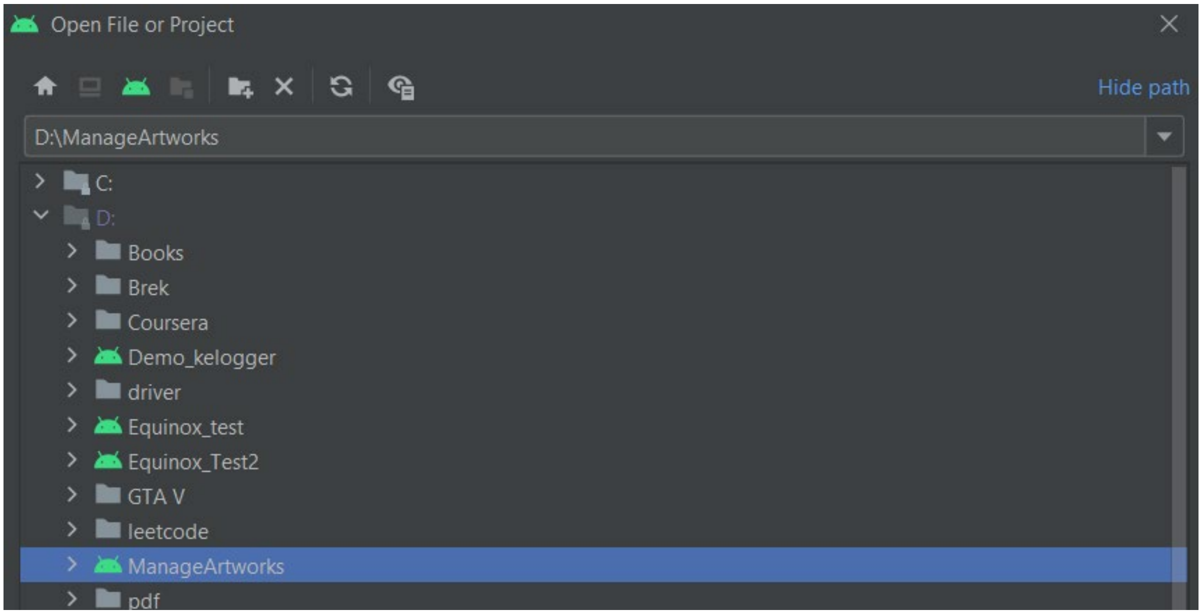
Otherwise, Gradle sync will be prompted by Android Studio as soon as you complete Step 4.

Step 2) Open Android Studio

Step 3) Go To **File -> Open**



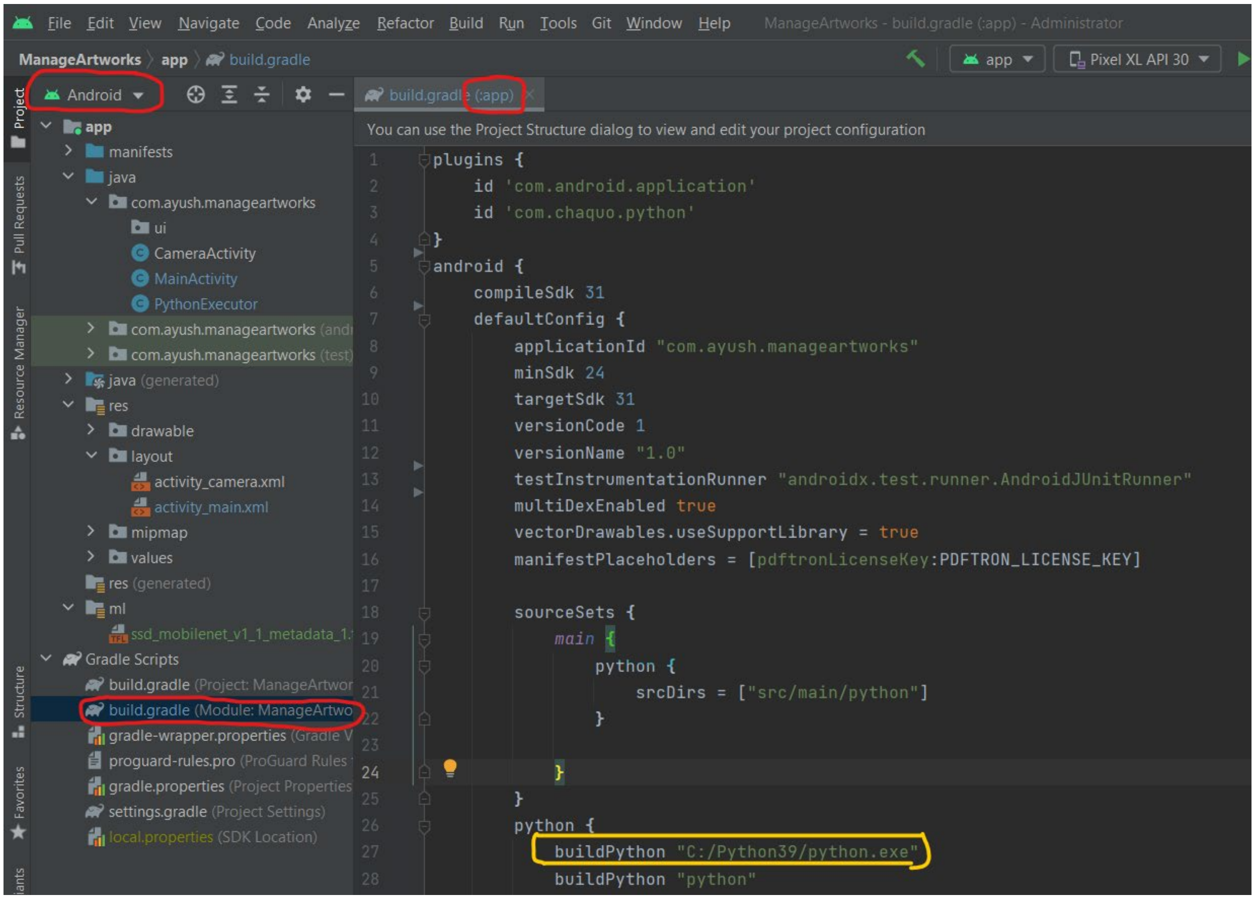
Step 4) Search for the Destination selected while cloning in Step 1.


Open File or Project
Hide path
D:\ManageArtworks
C:
D:
Books
Brek
Coursera
Demo_kelogger
driver
Equinox_test
Equinox_Test2
GTA V
leetcode
ManageArtworks
pdf

Setup Chaquopy

Within the **App-level build.gradle,** update the android { python { buildPython: "...' } } to your python.exe path in your local system.

→ Path to App-level build.gradle : ManageArtwork/app/build.gradle



Note: The project uses an unlicensed version of chaquopy, a commercial license can be purchased in the future. Additionally, an open-source license is available for open-sourced projects at no cost.

SSIM - Structural Similarity Index

SSIM is used to measure the similarity between two given images.

- Introduced in 2004, it was created to replicate Human Behaviour to analyze differences in images rather than quantifying errors between source and target images like in Mean Squared Error.
- SSIM is calculated between -1 to 1 or adjusted to 0 to 1. In which 1 means absolute similarity or same image. And, 0 or -1 (whatever be the case) means completely different Images.
- Structural Similarity Index Metric uses three key features extracted from an Image:

1) **Luminance**
2) **Contrast**
3) **Structure**

Calculating Metrics for SSIM

1) Luminance

Luminance is measured by averaging over all the pixels values. Denoted by mu(μ), the formula is :

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i. \tag{2}$$

The luminance comparison function $l(\mathbf{x}, \mathbf{y})$ is then a function of μ_x and μ_y.

2) Contrast

Image contrast is measured by taking the Standard Deviation (square root of variance) of all the pixel values. Denoted by sigma(σ), it is formulated as below :

$$\sigma_x = \left(\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu_x)^2 \right)^{\frac{1}{2}} . \qquad (4)$$

The contrast comparison $c(\mathbf{x}, \mathbf{y})$ is then the comparison of σ_x and σ_y.

3) Structure

For structure, we divide the input signal with its Standard Deviation, so the result has a unit standard deviation.

$$(\mathbf{x} - \mu_x)/\sigma_x$$

Here X is the input image, and mu and sigma are the Mean and Standard Deviations, respectively.

The Structural Comparision of the two images is done with a consolidated formula discussed later.

Comparison Functions for SSI metrics

Comparison Functions gives the index value after comparing both images for all 3 metrics discussed above.

1) Luminance Comparison Function

It is defined by a function shown below. Where mu represents the mean of the given image, and X and Y are the provided images.

$$l(\mathbf{x}, \mathbf{y}) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

C1 ensures that the denominator is never zero.

$$C_1 = (K_1 L)^2$$

L is the dynamic range for pixel values (255 while dealing with 8-bit images, i.e., uint8).

2) Contrast comparison function

The definition of the function is shown below. Sigma here denotes the standard deviation of a given image. X and Y are two images being compared.

$$c(\mathbf{x},\mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

C2 ensures the denominator is never zero and is given by: a

$$C_2 = (K_2 L)^2$$

L is the dynamic range for pixel values (255 while dealing with 8-bit images, i.e., uint8).

3) Structure Comparision Function

The function is shown below, where sigma denotes the Standard Deviation of a given image. X and Y are the two provided images.

$$s(\mathbf{x},\mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}.$$

Again, here C3 is a constant to ensure the denominator is never zero similar to C1 and C2.

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y).$$

SSIM Score

The SSIM score using all three metric scores is given by:

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = [l(\mathbf{x}, \mathbf{y})]^{\alpha} \cdot [c(\mathbf{x}, \mathbf{y})]^{\beta} \cdot [s(\mathbf{x}, \mathbf{y})]^{\gamma}$$

Here, α > 0, β > 0, γ > 0, denotes the importance given to each metric. If assumed α = β = γ = 1, i.e. equal importance to all three metrics, and C3 = C2/2, SSIM is simplified to

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)}.$$

Enhancing SSIM

The above-demonstrated procedure can directly be applied to images, but they won't give a good result. So, as suggested by the authors, the SSIM index should be applied locally rather than globally *(i.e. in small sections of the image and taking the mean overall)*.

We generally use an 11x11 circular-symmetric Gaussian Weighing function(basically, an 11x11 matrix whose values are derived from a gaussian distribution) which moves pixel by pixel over the entire image.

At each step, local SSIM is calculated within the local window. The formula is revised as:

$$\mu_x = \sum_{i=1}^{N} w_i x_i$$

$$\sigma_x = \left(\sum_{i=1}^{N} w_i (x_i - \mu_x)^2 \right)^{\frac{1}{2}}$$

$$\sigma_{xy} = \sum_{i=1}^{N} w_i (x_i - \mu_x)(y_i - \mu_y).$$

Here wi is the Gaussian Weighting Function.

Once all computations are performed over the image, we take the mean of all local SSIM values to find absolute SSIM.

$$\mathrm{MSSIM}(\mathbf{X}, \mathbf{Y}) = \frac{1}{M} \sum_{j=1}^{M} \mathrm{SSIM}(\mathbf{x}_j, \mathbf{y}_j)$$

TensorFlow Lite - ML for Mobile and IoT

TFLite for Native Android Apps

Android Devices and Edge Devices have restrictions such as computation power and limited energy source, which makes it impossible to run orthodox TensorFlow models. Using cloud computing is also not optimal because not always one has access to the internet with edge devices, and also latency becomes an issue. Thus TensorFlow Lite (TFLite) is an alternative to run your pre-trained models directly on the device.

Key Features:

1) Privacy: Data never leaves the edge/android device (No internet required)
2) Diverse Language Support: Java, C++, Swift, Objective-C, Python
3) High Performance and Low size
4) Reduced Power Consumption: Due to no network required and Efficient Interface

Android Quickstart Guide: **LINK**