

Assignment – 2
CSE 535: - Mobile Computing
Group 68

Group Members: Dinesh Koushik Deshapathi (1225344913), Sai Nikhit Gulla (1225356379), Devanand Vadlapudi (1225530722), Srinivas Dhanunjay Narava (1225406143), Chaitanya Kumar Gujjala (1225684694).

Objective: To develop an Android application that lets users click any picture and to identify the digit present in the image using a deep learning model.

Approach:

1. First we trained a deep learning model which achieved an accuracy of 99.44%. We used MNIST dataset which is available in the 'tensorflow.keras.datasets' library. The dataset consists of 60000 tuples. The CNN model has been run on 20 epochs and the Optimizer and loss functions used are 'Adam' and 'Categorical Loss Entropy' along with 'relu' as activation function for each layer.
2. The app consists of two screens. The first screen asks the user whether they would like to take the picture for classifying the image and on the second screen the user has the option to retake the picture or move ahead for predicting the digit in the image. The first screen's logic is handled in "MainActivity.java", the second screen's logic is handled in "ImageUploaderActivity.java" and "ResponseResult.java" is used to display the predicted value on the 2nd screen.
3. We built our API using Flask in Python and defined an endpoint "/v1/classify" which takes a POST/GET request that uses a json object to classify the image accordingly based on the digit inside image and saves the image in the respective folder in the file system.
4. Once the user clicks on "YES, LET'S DO IT" in the first page, the user will be taken to the camera to take a picture and then user can change the image by clicking on "RE-TAKE IMAGE" button present on the second screen, or the user can click on "CLASSIFY IMAGE" which classifies the image into one of the digits as desired.
5. After clicking on "CLASSIFY IMAGE" our backend takes the image and calls the predict method on that image which uses the trained model to predict the number present in the image and then saves it in the respective folder.
6. The server will return 200 if the image is saved successfully else will return the respective error code.

Technologies/Libraries Used:

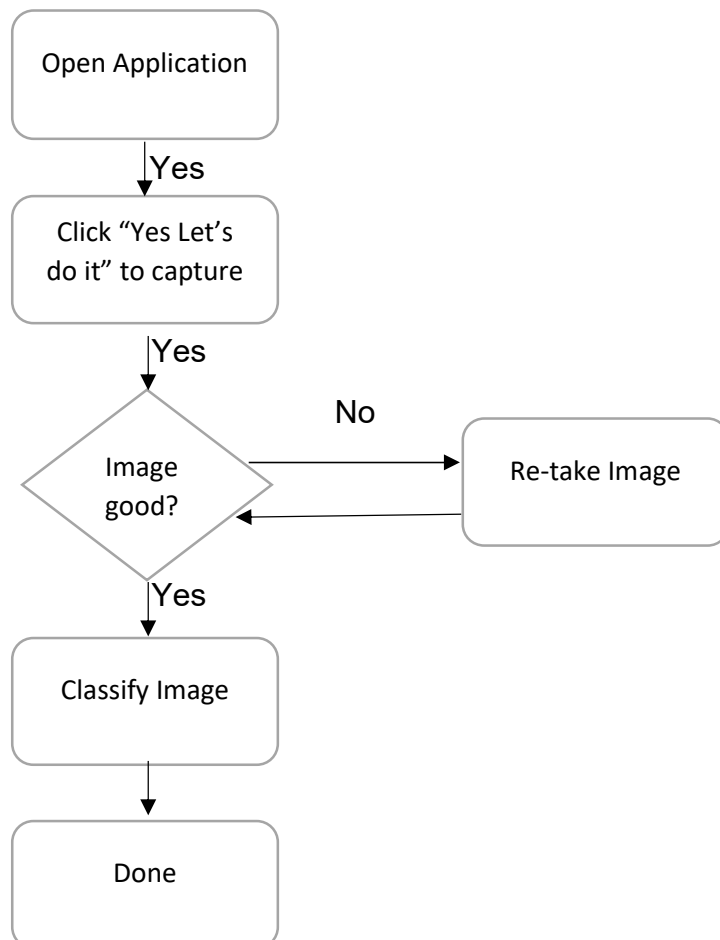
1. Android Studio
2. Visual Studio Code
3. Flask
4. Jupyter notebook
5. Tensorflow
6. Keras

Video demonstration: App Demo Video - [GDrive](#)

Backend Endpoints:

1. "v1/classify" (POST/GET):
 - a. "value": predicted_class obtained as json object.

Flow Chart:



References:

1. Camera Permission: <https://www.geeksforgeeks.org/how-to-open-camera-through-intent-and-display-captured-image-in-android/>
2. Connecting to localhost: <https://stackoverflow.com/questions/5528850/how-doyou-connect-localhost-in-the-android-emulator>

Github Link: [Github Repo for Backend and Frontend](#)