CSE 535: Mobile Computing

Project 3 Report - DigitClassifier

Keenan Rahman	Ashutosh Garg	Akshay Malhotra	Himanshu Pahadia	Ninad Bharat Gund
1222316745	1222340795	1220233742	1222425139	1222336947
krahman2@asu.edu	agarg69@asu.edu	amalho23@asu.edu	hpahadia@asu.edu	ngund@asu.edu

Responsibilities

Name	Responsibility
Ashutosh Garg	Read about OpenCV and required functionality for preprocessing images. Contributed to project report.
Akshay Malhotra	Worked on client-side of the project, responsible for HTTP server development. Worked on server side of the project, explored OpenCV SDK & contributed to dev. Responsible for recording a demo video.
Himanshu Pahadia	Worked on client-side (flask server) of the project. Prepared, preprocessed and trained models that can be used for classification. Responsible for recording a demo video.
Keenan Rahman	Read about minified, light models that can be deployed on mobile devices. Contributed to project report.
Ninad Bharat Gund	Worked on the server-side (flask server) of the project. Responsible for ensuring that the image is split into 4 parts and sent to the client. Responsible for recording a demo video.

Git URL

All source code is mentioned in the below-mentioned GIT URL. It contains source code with respect to server-side (Android Application), client-side (Flask Server) and Model Training (TensorFlow). Please refer to the README for more details.

GIT URL: https://github.com/pahadiahimanshu/Flaskdroid/tree/assignment3

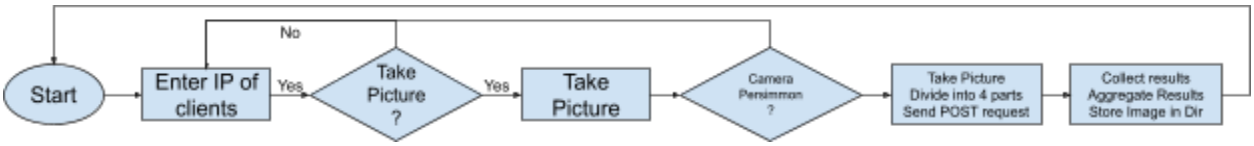
Technologies & Data Used

1. Flask - For developing a client-side application that accepts REST requests.
2. Postman - For testing and integration purposes.
3. Android Studio - Software used for developing the Android Application (server-side).
4. Kotlin - The language used for developing the Android Application.
5. Volley - The HTTP library used for sending REST requests to the client application.
6. CameraX - A camera-based library used in the android application.
7. Custom Neural Network - A neural network to classify the uploaded image quarters w.r.t digit.
8. MNIST Dataset - To classify and train the neural network.

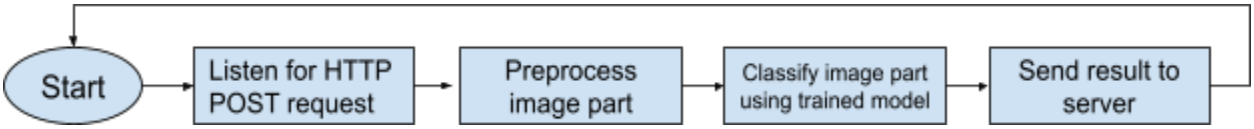
Introduction

In this project, we were responsible for developing an Android Application that lets users take a picture of digits, divide it into 4 parts and send it to 4 client applications for classification. The client uses a neural network to identify the digit and send their classification result to the Android Application. The Android Application then accumulates the results of classification and places the captured image in the respective folder. There are two components to the project - Android Application (server-side) and Flask Server (client-side).

Application Flow



Flow Chart of Server Application (Android Mobile Application)

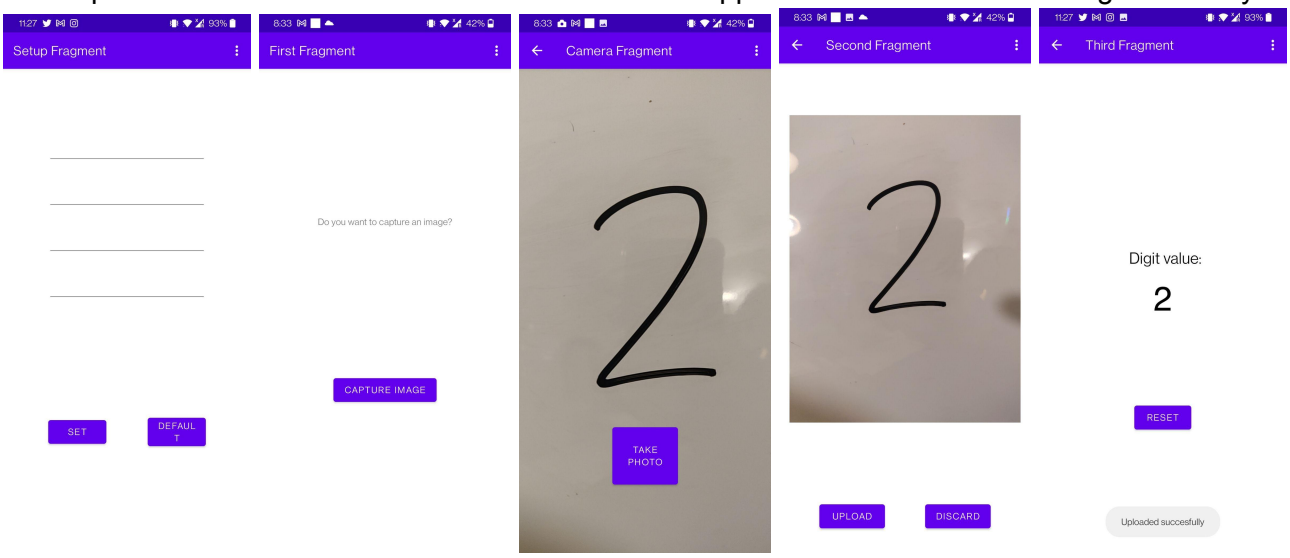


Flow Chart of Client Application (Flask Server)

Android Application

Android applications are developed using fragments. Fragments are the reusable components in the application UI and it helps us in modular application design. The application asks the user to take a picture of a digit. The input image is then split into 4 different parts and sent to the client-side application for classification purposes. A deep-learning model is used to classify the images on the client side and the classification results are sent back to the server. The image is stored in the respective directory based on the aggregated classification result on the server. Android application is developed using various frameworks -

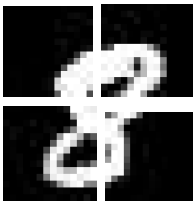
- **Camera System -** A Jetpack library called **CameraX** is used to make the use of cameras in the application simpler. CameraX gives us the ability to create these use cases, add listeners to them, and then connect them to the activity lifecycle since it is a use-case-based approach.
- **Network System -** An HTTP library called **Volley** makes networking for Android apps quicker and easier. We have implemented our HTTP communication of mobile apps to the flask backend using this library.



Screen capture of the Android application (L:R IP addres form, camera activity, send fragments to clients, classified)

Dataset

MNIST dataset has been augmented by fragmenting each image into four segments. Below is a sample fragmented image of the digit 8 -



Flask Server

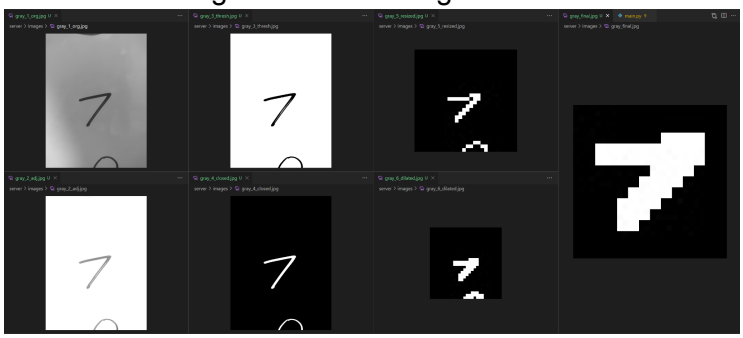
This backend Flask application contains one POST API. This API takes an image as a multipart form request. Once the request is received the image is sent to the neural network for classification and based on the digit classified the image gets saved locally into the folder.

Image processing - on individual cropped part

Currently we are expecting the captured images to be handwritten digits on a clear background without any major noise. We have tested with white on black background and black on white background. The MNIST dataset uses white digits on black background images. We are preprocessing the captured images to make the input of the classification network better.

Steps taken to process the image (can be seen in demo video) are -

- Conversion to grayscale
- Adjusting contrast and brightness
- Thresholding into binary pixels
- Closing (morphology) operator
- Resizing into 28x28
- Dilating thinner lines
- Detecting digit contour
- Removing rest of the background



Neural Network

The neural network is built using Tensorflow and Keras Library. We have used 4 different heads to our custom hierarchical neural network in this task in order to improve accuracy on an ambiguous dataset (where top left of 8 could be same as top left of 9).

Our model showed significance improvement over imagenet pretrained model that we tested. As the other model was failing on classifying the parts.

- Model Architecture -