



School of Computing and Augmented Intelligence (SCAI)

CSE 535 Mobile Computing

Teaching Assistant: Aranyak Maity

Instructor: Ayan Banerjee

ASSIGNMENT 3

Android application to connect master mobile with four different devices running four different models (one for each quadrant), click images of handwritten digits, and return the classified digit while also uploading it to the master mobile.

Project Group 20

- Purna Venkatesh Peddireddy (1223526954)
- Vijay Maddineni (1222982943)
- Prasidh Aggarwal (1225362125)
- Mohit Suresh Ahuja (1225915823)
- Phanindra Pabba (1226585219)

Date: December 6, 2022

Project Overview

This project consists of 3 major parts. Application, Servers, and the TensorFlow Models.

- Application part is developed using Android Studio and Java. It comprises all the major work being done in the application, comprising connecting to four different flask servers, gathering their results, and then uploading the image on the master mobile.
- Servers are developed using VScode and Flask. They preprocess the image and are loaded with four models trained on four quadrants of the MNIST dataset.
- ML model is created using TensorFlow and Keras on Google collab notebooks.

Environment Setup

The environment involved the use of two IDE's:

- Android Studio (for Java)
- VsCode (for Flask)
- Installed Flask along with dependencies on zsh.
- Gradle was used for easily packaging the dependencies in Android Studio.
- Google Colab for building, testing, and training the model. (Using Keras, TensorFlow, Matplotlib, etc.)
- Four devices are running four different flask servers.

Tasks

Major tasks Included:

- Changed the app layout to add four IP addresses entering text boxes.
- Validating the IP addresses after the user has entered them.
- When the upload button is clicked, four URLs are constructed using those IPs and the image captured is broken down into four different parts. Four post calls are made to four flask servers using the four URLs and four payloads of images converted to strings. (Also encoded to base64)

- Flask servers preprocess every image (grayscale, sizing, etc.) and feed the image into the model to return the value of the digit classified and its confidence.
- All four such results are then gathered into a string array in our application, processed, and then using a HashMap, the most frequently occurring digit is recognized using the highest confidence.
- Finally, the result is shown as a toast notification on the master mobile, and the image is saved under the digit's directory on the master mobile.

Lessons Learnt

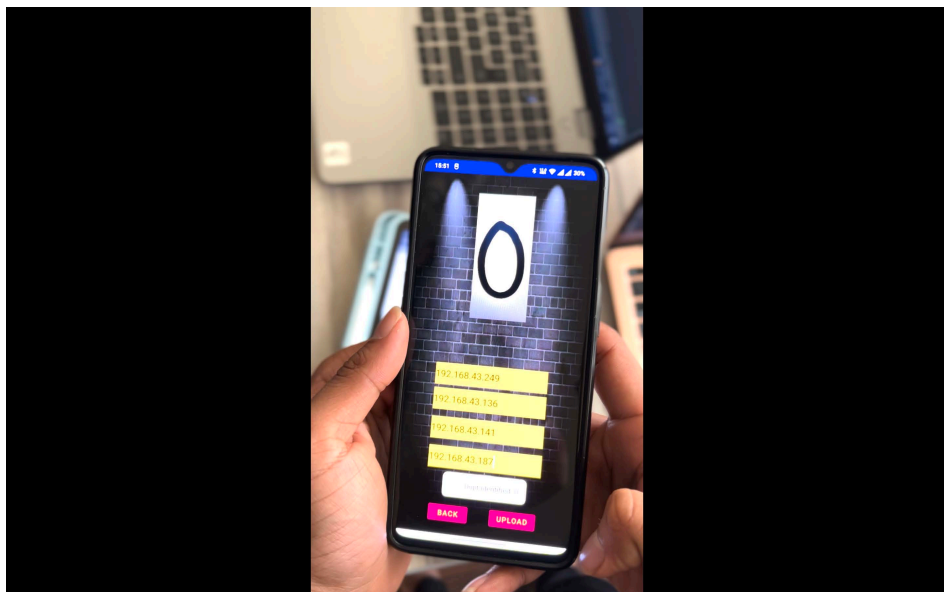
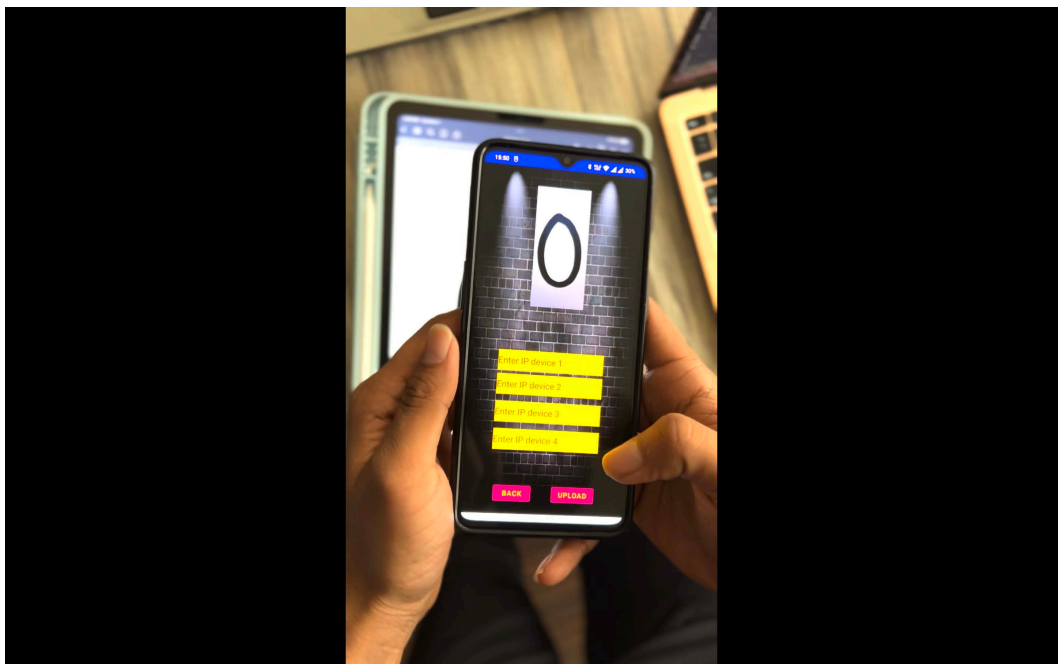
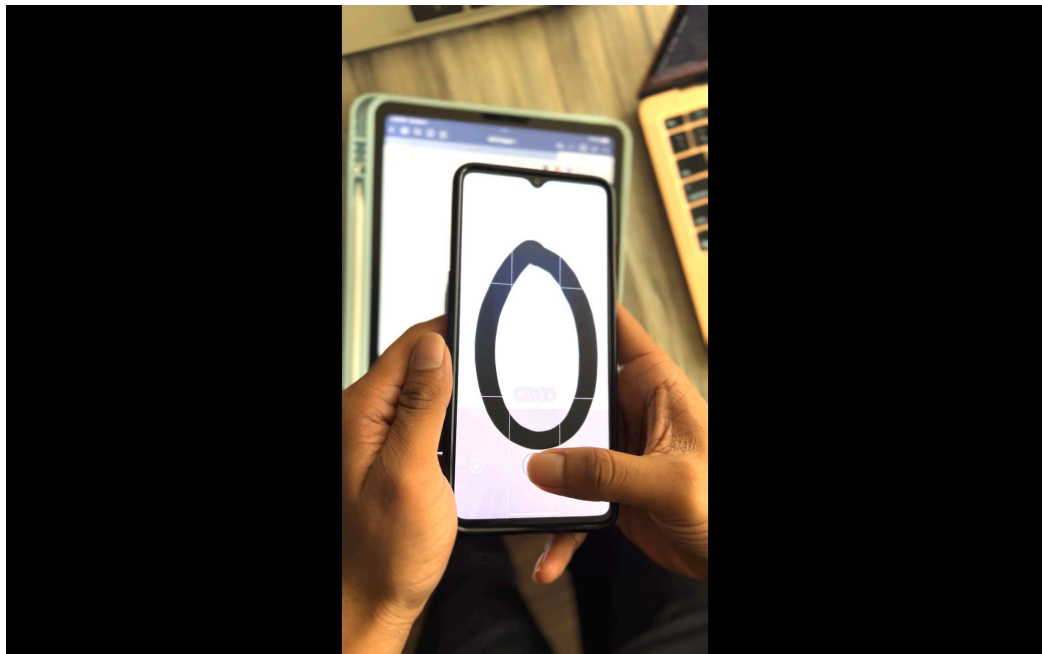
- Understood how to break an image into four equal parts and train the MNIST dataset separately on those four parts.
- Learnt how to deploy flask servers integrated with TensorFlow.
- Worked in a team and built the application using a SCRUM model to distribute tasks equally.
- Learnt how to troubleshoot various connectivity issues along the way.

Individual Member Work

These are the ballpark task numbers, even though everyone worked together to get the application running and helped wherever possible.

- Purna Venkatesh Peddireddy(6 tasks)
- Vijay Maddineni(6 tasks)
- Prasidh Aggarwal(6 tasks)
- Mohit Suresh Ahuja(6 tasks)
- Phanindra Pabba(1 task)

Application Working



+ Code + Text

```
39 # plt.xlabel('train_labels[1]')  
40 plt.show()
```

✓
1s

