

CSE 535 Mobile Computing - Project 2 - Group 14

Team Members

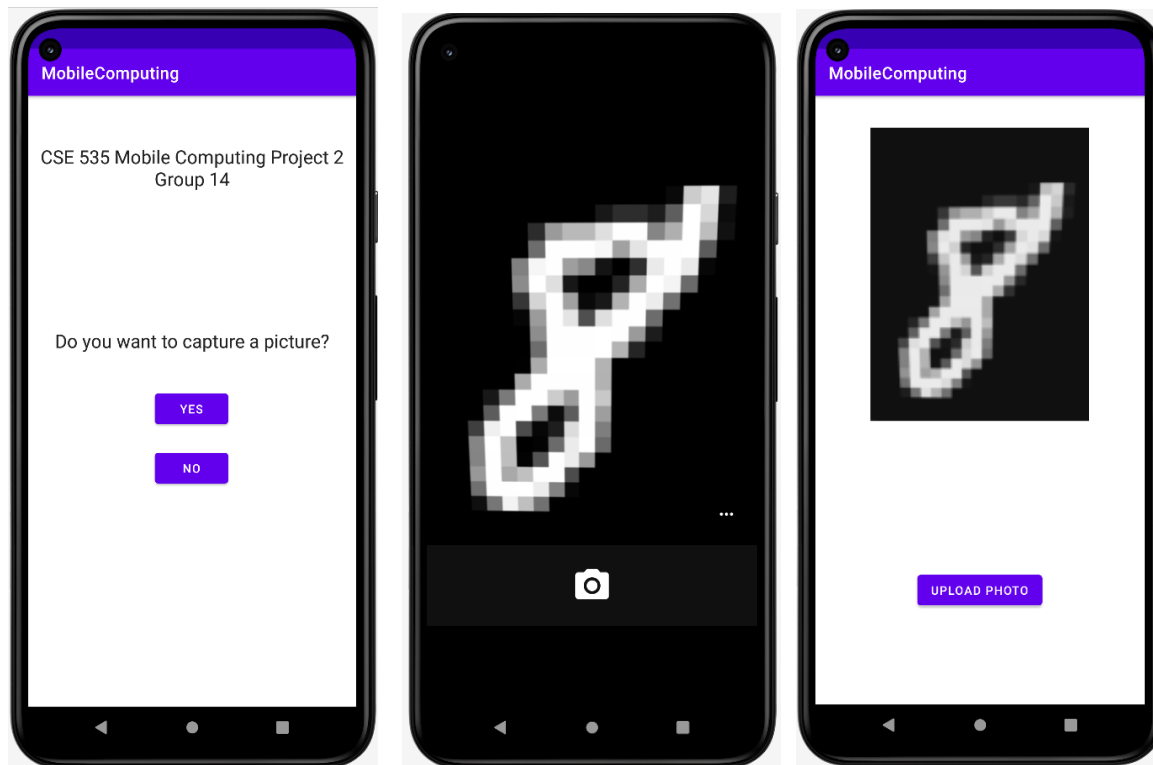
Sai Krishna Reddy Cheruku	1222300703	scheru13@asu.edu
Mahesh Chandra Yayi	1224141347	myayi@asu.edu
Fenny Zalavadia	1221443561	fzalavad@asu.edu
Shilpitha Gandla	1224631798	sgandla2@asu.edu
Ajay Kannan	1219387832	akanna14@asu.edu

Technical Details

The objective of our project was to create an android application that lets the user capture pictures of handwritten digits, upload it to a server and use the server to classify the digits. The digits will be classified into their respective folders. The server would in turn save this image. The project mainly consists of two parts, first is the android application and second is the flask server. The details of these have been explained further in the below sections.

Android Application

The functionality of the android application developed in the previous assignment has been extended to accommodate ML-based prediction for handwritten images of digits. Another change in addition to this is that the category dropdown has been removed. To reiterate the functionality of the previous version of the application, it consists of two pages. In the first page the user is asked whether they want to capture a photo or not. The user is provided with two buttons namely yes and no. If the user clicks on the yes button, they are navigated to the camera to capture a photo. If the user clicks the no button they are exited from the app. These activities are implemented with the help of listeners which are initialized when the app is started. So whenever the user clicks on these buttons the corresponding methods are triggered. After the user clicks on the yes button they are taken to the camera functionality.¹ Here they can capture a photo. Once they capture a photo they have the option to confirm or recapture. Once they confirm the photo they are taken to the second page of the application. The camera functionality is provided with the help of the underlying camera application which is in the operating system. This is done with the help of camera intent. This photo is initially loaded into a bitmap class variable. Later the bitmap photo is converted into a byte array which is sent to the server for prediction of the digit using an ANN pre-trained on the MNIST dataset.



In the second page the user can view the image that they just captured. This is done using image view class. Below the image the user can select a category for the image from a dropdown list.² After selecting the category the user can click on the upload button. When the user clicks on the upload button a post request is made to the server. The post request is made on the ip address and port number in which the server is running. The body of the post request also contains the image and category selected by the user. If the image is successfully saved the user is navigated back to the first page of the application. The dropdown functionality is achieved with the help of spinner class. The body of the post request is created using the multipartbody class which helps to add multiple data components.

Specifications of the ML Model

The model consists of 3 Hidden layers with 256 neurons in each layer. The first and second layers are connected with the activation function ReLu. Dropout has been implemented for 45% of neurons in these two layers during training. Softmax is applied on the third layer which is used for predicting the digit. Loss function used is categorical cross entropy and the optimizer used is ADAM. Model has been trained for 20 epochs with a batch size of 128.⁴

Flask Server

The backend was developed using Flask server and python. Flask server is defined as server software that is capable of running HTTP requests on the public world wide web, private LAN, and private WANs and comprises one or many computers bundled together and dedicatedly working for running the software application on the worldwide web. The server receives a post request from the App via internet - local connection.³ The server receives an image consisting of a handwritten digit, which is then converted into grayscale image. The conversion is important, because the dataset used to train the model was in grayscale image scale. Now, this image is normalized and converted into feature input on which the trained model is called to predict. Based on the prediction, the digit will be returned to the user.

References

1. <https://developer.android.com/training/camera/photobasics>
2. <https://developer.android.com/develop/ui/views/components/spinner>
3. <https://www.freecodecamp.org/news/how-to-build-a-web-application-using-flask-and-deploy-it-to-the-cloud-3551c985e492/>
4. <https://www.tensorflow.org/tutorials/keras/classification>