CSE535 Smart Home Hand Gesture Mobile application and Hand Gesture Classification Project Report

Junhui Liao
School of Computing and Augmented Intelligence
Arizona State University
699 S Mill Ave, Tempe, AZ, The United States
jliao28@asu.edu

***Abstract*— This project report is served as one of the two projects select to be included in the Project Portfolio to fulfill the graduation requirements.**

I. INTRODUCTION

A. Project Background

Using computers to recognize the hand gesture is not a new topic. Back in 1987, researchers have started to use the analog flex sensors on the gloves to detect the positions and orientations of the hand as well as the bending of fingers [1]. The computer connected with the gloves could provide real time 3-D model and allow the hand gesture to be the interactive interface. Later, more researchers were engaged in this topic and some Japanese researchers were able to use computer to recognize 32 out of 46 Japanese alphabets using the hand gesture capture glove in real time [2]. The limitations of such technology were obvious. The glove equipped with sensors was not available for everyone, any this equipment must be connected with the computer with cables, which limited the application scenario. Later, to overcome the limitation of using external glove sensor, researchers started to work on picture recognition via analyzing pictures took by camera. Normally, to achieve the hand feature recognition, machine learning algorithm or even deep learning algorithm should be applied to train the computer to distinguish the differences between the hand gestures. Thanks to the development of the computation capabilities, there algorithm can be easily running on personal computer and even on the smart phones. Nowadays, the smarts phones are equipped with multi-cameras, some even include depth camera that enable to capture distance information when taking photos, which enabling 3-D model build up, recognition and interaction with smart phone without external gloves and sensors [3].

B. Project Purpose

The main purpose of the project in CSE535 (Smart Home Gesture Control Application) is to give student a taste of how picture or video-based hand gesture recognition work. Despite that the voice control smart assistants, like Siri or Alex, are popular these days, there are still rooms for hand gesture control assistant for specific people or using scenario. For example, hand gesture can be very useful to deaf people or old people who can only use hand gesture to express themselves. Hand gesture can be also useful when working in quiet area without disturbing others. Also, the basic knowledge behind the hand gesture recognition is similar to picture or video feature recognition. With the idea that students can learn from this project, one can apply these knowledge when trying to have computers to recognize loads of pictures or videos in one go. In addition, this project provides students the chance to practice on how to develop an android mobile application that can be ran on smart phones by using the Android Studio development platform, which is a very practical and useful skills that can be later applied to real world tasks.

C. Project Description

The whole project is separated into two parts. In the first part, the students are required to use the Android Studio to develop a mobile application with three actives (interfaces). The first activity will have the drop down list that contain all 17 gesture options (including number 0 to 9 and lights on and off, fan on and off, fan speed increase or decrease fan speed and set thermostat). After choosing one of the 17 gestures, the second activity will pop up with the corresponding short video of how the gesture should be done. The video should be able to replay for at least 3 times. On second activity, another bottom, “Practice”, should be included. Clicking the bottom will open the camera in video mode and record the gesture practiced by the user for 5 seconds. After confirming the video, the third activity will pop up with a bottom “Upload” on it. The upload bottom will send the recorded video to the local server and also take the user back to the first activity. The expert videos are provided, and it is recommended to used Flask to build the local server. Student should use the application to practice three times for each of the 17 gestures and create 51 videos. The second part of the project is about gesture recognition through Convolutional neural network (CNN) deep learning algorithm via python platform. The 51 videos that recorded from part 1 will be used as the training videos and the base codes that can do frame extraction and feature extraction are provided. The CNN model that was built by recognizing alphabets is also provided. After training the model with the video, the provided test video will be used to validate to check the

accuracy rate of the gesture recognition.

II. SOLUTION

A. Layout and Activities Switch

The first challenge of using Android Studio is getting familiar with how the layout design of the application works on this development platform. The official tutorial is a great resources to learn basic ideas about the layout [4]. One can use the user-friendly way, called design mode, to use the mouse to choose the desired views like text, button, widgets, different layouts, containers, etc to be added on the activity layout. Similar to creating a PowerPoint slide, you can drag the component to the desired positions, but one need to define the constraints (distance to the edge, distance to other components or centering types) applied to the components. The text, color, size, fonts and so on can be altered in the attributes of each components. This can provide you the immediate impression of how the application will look like. When you are getting familiar with the design process, one can switch to the code mode and use the command line codes to define the attributes of each components. The user can switch to the design mode to check the outcome of the code whenever he or she wants to confirm. The code mode is actually better if the user is try to create activities with same layouts by simply copy and paste of the code. After a few tries on using design mode with drag and drop, I turned to the code mode which gives me more precise control on the attributes of different components.

The application will contain three activities (three different user interfaces), therefore the second challenge will be figuring out how to make the transition between different activities via simple clicking on button or items. The normal way is simply creating a new java class and associated activity layout file separately and then add button or other components as the trigger point via onclick listener to trigger the transitions between activities. Then a new intent will need to be created within the onclick or on select listener to specify which activity should be presented after the click action. After reading some tutorials, I figured out that the best way is simply right click the package folder and simply add an empty activity to the project. The Android Studio will automatically create all the necessary items for you. And it should the most secure and easiest way, since it is likely that one would miss some parts by creating the whole thing manually.

B. First Activity

To create the drop-down list on the first activity, the spinner as one of the containers type components was selected. One thing should be kept in mind is that the names list of the items should be defined in the strings.xml file as string array and the entries source shoud be specified in the activity layout code section. The font attributes can also be specified in the layout. After creating the spinner, the onItemSelected will be used to trigger the activity switch with the new intent option.

C. Second Activity

On the second activity, there are three major challenges. The first one is to select the right video based on the selection made on the first activity and the key is to transfer the name of the gesture selected in the first activity to the second activity. One can simply create a public string to contain the name of the gesture selected in the first activity. Then, in the second activity, one can use FirstActivity.name to get the name of the gesture selected. Then, switch function can be applied to determine the corresponding video file (by assigning the corresponding url to the video file) to be played based on the gesture name. The second challenge is to have the video to play at least three times. The video view component can be easily added and resized in the layout. The video url will pass to the video view and a button is created to act as the play function so that the video can be played as many times the user wants to. Open the camera is third challenge. The first important task is to give the application permission to use the camera as well as writing to the external memory in the mainfest.xml [5]. It is also safe to add "ask permission to use camera" into the code, so that the application can ask the user to grand the permission again. To open the camera is similar to activity switch, a new intent should be created to switch to camera interface. The 5 seconds limitation on video duration can be set in this intent. Over ride the camera activity by sent the camera request code and save the recorded video path which will be sent to the third activity for video file upload. Also, another new intent to switch to the third activity should be included after confirming the recorded video by clicking on "ok" and the video path is sent along with the intent.

D. Third Activity and Flask

The key challenge in the third activity is to learn how to use Flask on python to build a local sever and connect the server with the application. For the Flask part on python, the official tutorial already provides a good example that provide all the necessary information [6]. The similar switch function will be used to assign the gesture name based on the selection on the first activity. A "upload" button is included in this activity and the click will trigger the function that can connect between application and the Flask server, and upload the file. The local server ip address should be specified here and after the upload finished, a new intent to switch the activity back to the first activity should be included, and a toast message with "upload successful" should be defined right after the activity switch. Okhttp3 package [7] is selected as the core package for the function that connect the application to the local server. A new Okhttp Client is fist created and then the video content is resolved to provide the information of the video to the request function. The name or class of the video will be passed to the server via request function as well. After the local server received name or the class of the video, the python code can go through the upload folder to check if there is any video file that belongs to the same class. If there is, then the video file will be renamed as the gesture name with the number of the video in the same

class plus one. If not, the video file will be renamed as the gesture name with number 1 indicating it is the first video of this type of gesture recorded.

E. Hand Feature Classification

Since the key python codes are given, the challenge of this task is not about developing the CNN deep learning model, but to understand the procedures and logic behind the deep learning. All we need to do is to understand what the provided codes do, and how should the codes be connected with each other. The first step is to get the middle frame of the video. The provided python code simply get the duration of the video and get the frame at the middle of the duration. This part of codes do not require any alteration, mainly set the video fold path to where the 51 video located. The gesture name will be extracted from the file name as the corresponding gesture class for each frame selected. Secondly, the feature extraction codes get the pixel information of the frame and then use the CNN model to calculate a feature matrix for each frame. The key point here is to add one procedure to reduce the pixel information of the frame from RGB model to gray scale model. These two steps should be done for both training videos (51 from part 1) and the testing video (provided by the project). After the feature matrix of the testing video obtained, cosine similarity function is used to compare the feature matrix of the frame from a testing video with all feature matrix extracted from the training video. Then the gesture class of the training video that is most similar to the testing video based on cosine similarity will be assigned to the testing video, and the accuracy of the prediction can be calculated by counts of accurate prediction (predicted class same as testing video original class) divided by the total number of testing videos.

III. RESULTS

For part 1, the mobile application can be run on both emulator on the Android Studio as well as on the actual Android mobile phones, and all the required functions from the project can be achieved. A video demo was made to to demonstrate how this application works and uploaded to the YouTube (https://www.youtube.com/watch?v=hf-GH2J3iUs). For part 2, the final accuracy I got is about 11%. There are many reasons that may lead to the low accuracy. Firstly, the provided CNN model was built for identify hand gesture for alphabets, which is different from this project. A new model built specifically for this project may increase the prediction accuracy. Secondly, the hand gestures are completed within 5 seconds but the tempo of the gesture actions can be very different from one person to another. The start and end pose of the hand is a fist, therefore, if a person do the gesture in a quick tempo the gesture may finish at 2.5 seconds (middle of the video). This will result in getting multiple fist gestures but assigned with different classes since they come from different gesture videos. To avoid such issue, static hand gesture poses are highly recommended. Thirdly, the same gesture done by right and left hand can be very different, so it is important to use the same hand in both training and testing video. Lastly, the background of the video should be clean and same color of both training and testing video is preferred.

IV. CONTRIBUTIONS

The CSE535 Smart Home Gesture Control Application is a individual project. I went through all project materials and tried to digest by myself. For the code parts (Both Android parts and python parts), I went through many tutorial websites and video available online, and learn from the concepts and ideas behind those, then tried to write my own codes. Eventually, I managed to build the application, local server, record 51 gesture videos and hand gesture classification all by myself.

V. REFLECTION

As the first project in my MCS program, this project is definitely a very interesting but difficult task for me. Before this project, my coding experience was mainly focused on python and I had no experience on Android development. This project forced me to learn some basics related to Android application development, which is a crucial and practical skill that required for real world occupations like software development engineer. Even though the skills that we can learn from this project are only a small fraction of the skills required in actual application development, this project is served as a good start for the students and we can have certain directions to dig into different parts as needed. The flask part is not difficult as it is based on python, but it is the first time I learnt that we can use the python code to create a local server under the same network, which expend my skills on python coding. Using the camera to identify the hand gestures is a hot and interesting deep learning topic nowadays. The concept we learnt from this project can easily be extended to other topics, such as using computer to classify pictures or videos, or even facial recognition. Even though we are not required to build the model from scratch, but we gained a big picture of how one way of deep learning should be done. In addition, since there are so many great open resources, working on the project educates me how to search smartly and ask question efficiently, which will be critical to facilitate my study in the future.

REFERENCES

[1] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, Y. Harvill "A hand gesture interface device," CHI '87: Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface, May 1987, pp. 189–192.

[2] T. Takahashi, F. Kishino, "Hand gesture coding based on experiments using a hand gesture interface device," ACM SIGCHI Bulletin, Volume 23, Issue 2, Apr. 1991 pp. 67–74.

[3] Z. Lu, S. Rehman, "Touch-less interaction smartphone on go!," SA '13: SIGGRAPH Asia 2013 Posters, Nov. 2013, Article No.: 28.

[4] "Build your first Android app in Java [Online]," Accessed on: Feb. 10, 2022. [Online]. Available: https://developer.android.com/codelabs/build-your-first-android-app0.

[5] "Camera API," Accessed on: Feb. 10, 2022. [Online]. Available: https://developer.android.com/guide/topics/media/camera.

[6] "Uploading Files," Accessed on: Feb. 10, 2022. [Online]. Available: https://flask.palletsprojects.com/en/2.0.x/patterns/fileuploads/.

[7] "Upload a File with OkHttp," Accessed on: Feb. 10, 2022. [Online]. Available: https://howtoprogram.xyz/2016/11/21/upload-file-okhttp/.