Integrated Smile Level Labeling Application Tutorial (1A)

EECS410, Mobile Health (mHealth) Technology, Fall 2016
Advisor: Prof. Ming-Chun Huang

Table of Contents

INTRODUCTION **1**

BODY **2**

SECTION 1: DATASET 2
Faces *2*
Saved Results *2*
SECTION 2: APP 2
Android Sample App *2*
Important Coding Notes *3*

REFERENCES **3**

PREVIEW 1B **3**

Introduction

In this tutorial, you will make a smile level indicator application on Android. The app UI will contain: an ImageView, smile level labeling buttons, browse folder button, and next/previous image buttons. The Android ImageView will display images (of people faces) that are located in a specified folder that is determined by the user click of the "Browse Folder" button. Once an image is viewed, the user should rate the smile of the image through the smile level labeling buttons. The click should save a log entity in a CSV file that contains "file_name.jpg" and "#". The "#" is an integer value (1 – 4) that specifies or rates the smile of the shown face. A value of (1) indicates that the person is natural and (4) indicates that the person is at the peak of smiling.

All teams will be provided with at least 3 folders. These folders include frames taken from a small video that captured volunteers going from ideal face to a smiling one. That is, folder may contain around 300 frames for users to rate the smile of the volunteer at the framed viewed on the ImageView. Once clicked on a label, the user may proceed to the next picture in the folder.

The result of this tutorial, in terms of saved logs, will be incorporated with your next tutorials and projects.

Body

Section 1: Dataset

Faces

All teams will be provided with at least 3 sets. A set contains all frames of captured footage for a smiling person. The user of the app will indicate and rate the smile of the person's face in the viewed frame. An example of a rating for a volunteer face is as the below image.

1 2 3 4

Saved Results

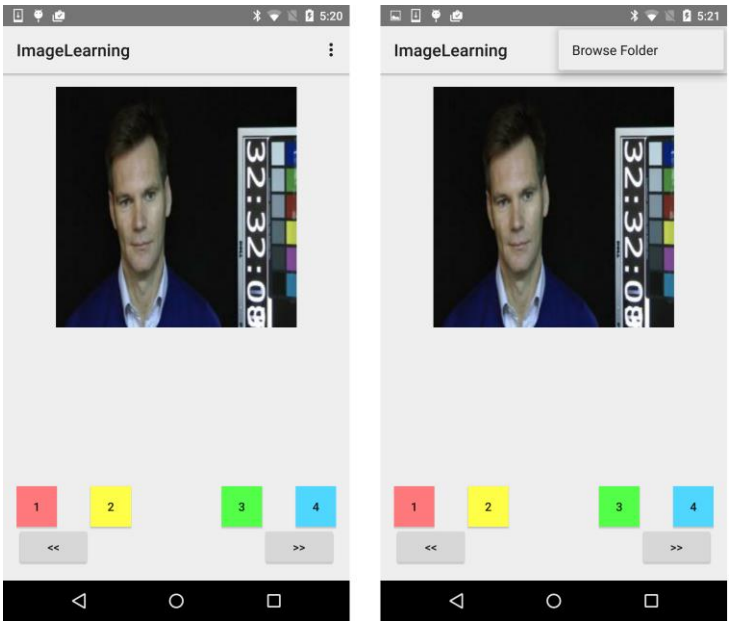
All results should be saved in a CSV file that contains only two columns, "THE_FILE_NAME" and "#_OF_LABEL". Please note that CSV files should pertain the same name as the folder that is being browsed for frames. We also advise that you implement the openCSV library to handle CSV files. An example CSV output file has been provided to you.

(FILE: 020_spontaneous_smile_4.mp4.csv)

Section 2: App

Android Sample App

All teams will be provided with a sample app that uses Picasso as a library to handle viewing images on the ImageView. Even though all UI items are visible in the UI, all buttons and menu items have not been implemented. Thus, all teams should follow the guideline for conditions of naming the CSV files and logs, as mentioned in the previous section. Below are screenshots of the sample app.



(CODE: ImageLearning)

Important Coding Notes

- The approaches used in the sample code is primitive and can easily be enhanced. Try to make your UI code as dynamic as possible, for instance, by adding other functions to browser for a folder, rather than coding it in onCreate().
- Don't forget to add memory read and write access permissions to your AndroidManifest.xml file.
- Please carefully review this document and other attachments, including the output template, dataset (photos) and code.
- All material of this class and tutorial, including example codes, volunteer photos and other templates are not to be shared to any external sources.

References

- http://square.github.io/picasso/
- http://opencsv.sourceforge.net/

Preview 1B

For tutorial 1B, all your collected data will be incorporated with Android sensor data. That is, your code will be retrieving multiple sensor data from Android phone simultaneously, and how to describe touch movements in terms of an action code and a set of axis values while logging your results.