國立東華大學資訊工程系

National Dong Hwa University

Image Processing Term Project

Date: 30th June, 2018

\*\*

**Facial Recognition and Detection**

\*\*

**Group No#:** 18

**Group members:**

Andre Freeman – 410521320

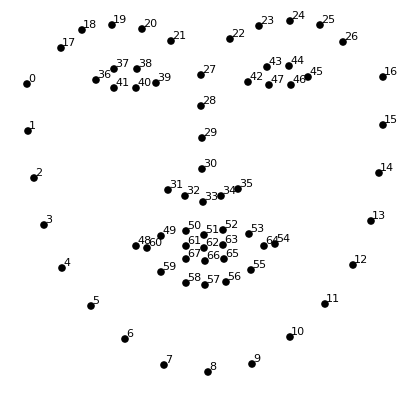
黃嵩育 - 410321306

**Abstract**

Facial recognition has many used in the present technological advance world. It is the ability for a machine or algorithm to detect and identify a person by a facial image. Facial recognition is achieved by a machine through a landmark point system. To increase accuracy machine learning is introduce to ‘train’ or ‘learn’ what each person is through multiple samples. This Project will utilize a 84 (x,y) landmark point system to identify and train it. A support vector machine (SVM) then it turns train and predict the results.

**Introduction**

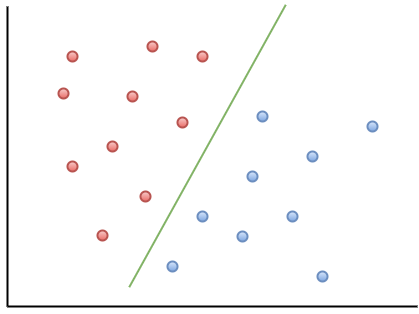
Human being can instinctively identify human and objects due to a highly cognitive brain. Machine on the hand is not capable (yet) for computation and analyzing at that level. To achieve facial recognition we must utilizes an approach that takes advantage for varies image processing techniques to come up with what we call landmarks. A landmark basically is a particular point on the face that we can and always detect and fine on every face. N.B. Each point is a composition of a x and y value which which acts as a coordinate value. Figure 1.1 tries to depict the landmark point system that we will attempt to use to achieve facial recognition.



Drawing 1: Figure1.1 - showing example of the point system

The first goal would be to identify a face in a image with some sort detector (that is expounded on below in section ). After facial detection we can locate landmarks. Since we will want to allow this project to be able to detect faces at a skewed, off to a side and/or with emotions, we first try to do a facial detection then facial alignment. Align the facial image will reduce and minimize the noise input and increase accuracy. It also help to create a basis in terms of angle and position of each face.

After comprising the landmarks we will treat the each landmark x and y values as a feature. Therefore, for our support vector machine (SVM) it will have 168 features. The SVM is used for its classification functionalities. It works in a way where it tries to generate a line (function) that can divide two classes as evenly possible. Figure 1.2 gives an example of how a SVM would try to classify two classes.

  
Illustration 1: Figure1.2 - illustrating how SVM works between 2 classes

This project we will utilize the SVM one vs one classification technique. That is, the SVM will generate these classification line function for one vs one round robin collection of all the classes. For example: If there were 4 classes there would be 6 classification line function (1 to 2, 1 to 3, 1 to 4, 2 to 3, 2 to 4 and 3 to 4).

In the very end the SVM can easily predict who a face belong to by comparing it to all the line functions generate from the training data set.

**Motivation and Research Question**

This group, group 18, wishes that they can create a mechanism that can identify face from a collection of faces of everyone in the class/group. N. B. These faces are not all (direct) frontal faces but are sometimes skewed, off to a side and with facial emotions. The group was inspirit into this direction after seeing how different image processing techniques can be applied to detect faces in a image or video. Just out of curious we extended the facial detection idea not only to detect a face but to identify it.

For this project though we will be utilizing a collection of 13 images obtained from 50 persons from an external source. One (1) picture is then randomly removed for each person thus obtain 50 images for testing purposes.

**Research Method**

The first step is to detect a face in an image. For this project we will utilize dlib facial detector. From the list we can load each image one by one. Also a mechanism is set in place so that we can detect if an image is a next epoch of the same person or if it is a new persons image. Example s01\_01.jpg and s01\_02.jpg is images for the same person where s02\_01.jpg is for a next person.

From each image we utilized the following:

Face predictor:

shape\_predictor\_68\_face\_landmarks.dat

facenet model: external code to loacate landmarks

Face detectors:

dlib.get\_frontal\_face\_detector

pose\_predictor:dlib.shape\_predicictor

Face aligner:

face\_aligner: openface.AlignDlib

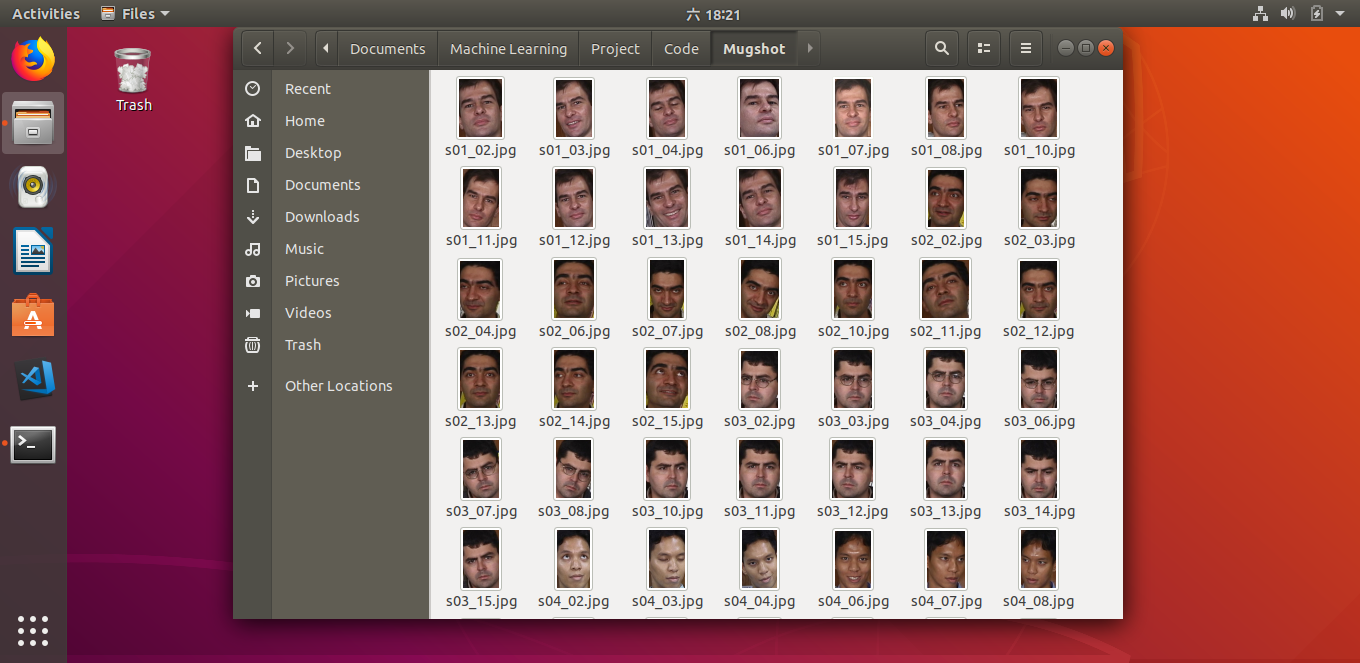
From the predictor and the face detector after apply face alignment and the aid of some external code we can generate 84 (x,y) points for each face.

The code then tries to aggregate the landmarks for the entire training data set. The sklearn svm package is then utilized for data training and predicting. For the sklearn SVM an another data structure is needed to label/name each group of landmarks. To predict and identify a image, a sklearn svm predict function with the landmarks of the face trying to be predicted as parameters can then be simply implemented.

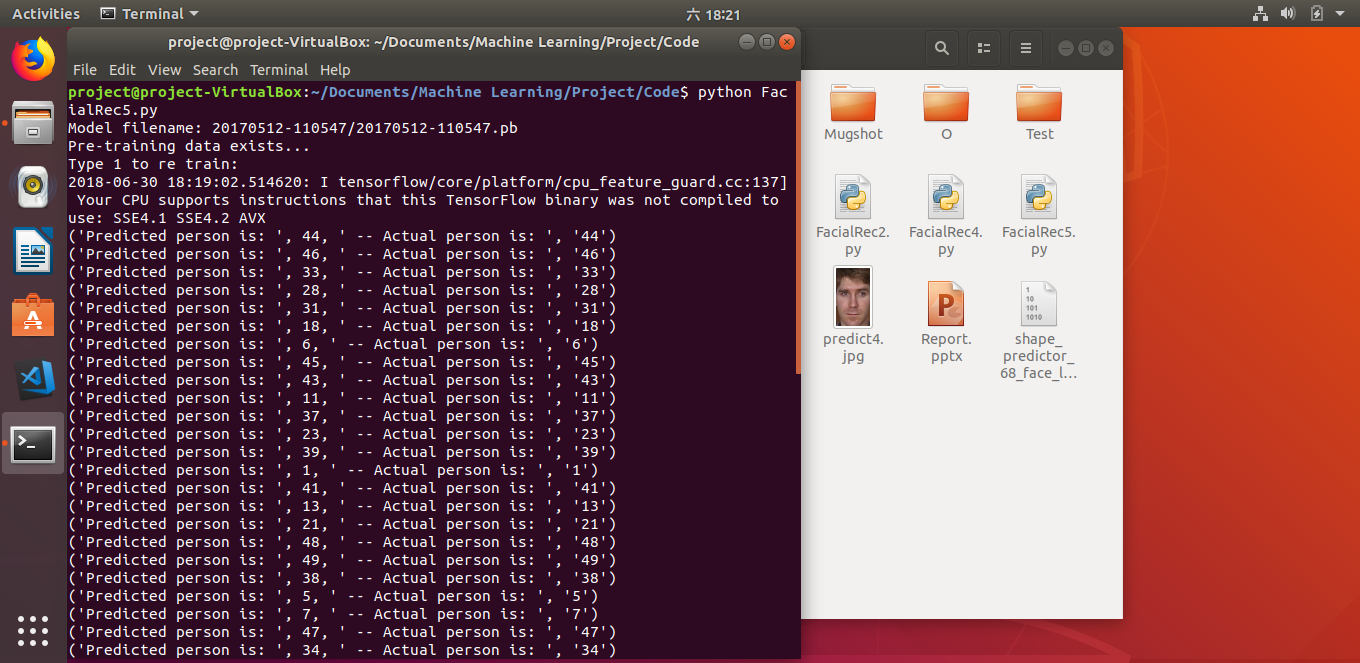
In efforts for error checking more functionalities are incorporated to load images from the ‘Mugshot’ folder as training data and ‘Test’ as test data. The accuracy is also calculated after all the test images has been predicted. Generate landmarks for each photo is a very time consuming process. Therefore to save time a read to file and write to file functionality is added to improve computation speed by allowing you only having to generate landmarks once for each test image once the data-set have previously been save to a file.

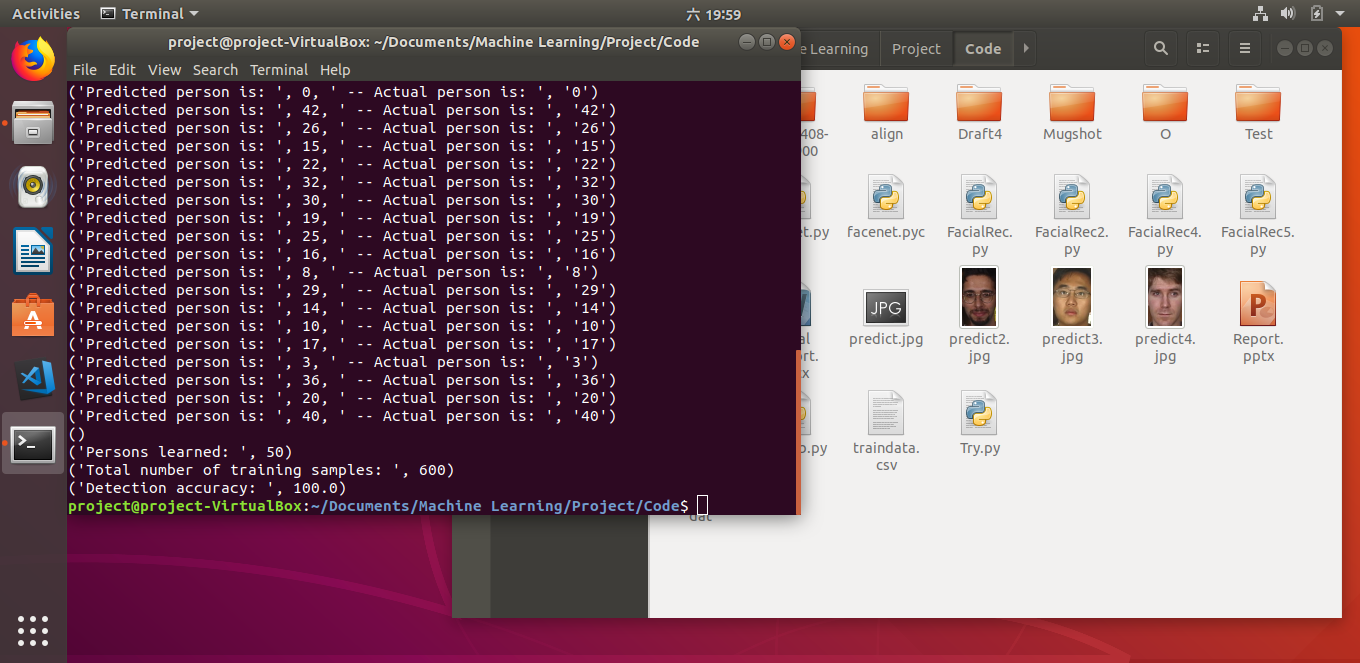
**Results**

To provide the ability to check accuracy, 50 photos is utilized during testing to be contrasted against 600 photos as training data. (N.B. 12 photos is for each 50 persons). A sample of the test data is shown below in figure 1.5.

  
Illustration 2: Figure 1.5 showing testing data preview

The algorithm predicting each photo is seen below in figure 1.3 and figure 1.4



  
Illustration 3: Figure 1.3 (above) and Figure 1.4 (below) - showing the program resuults.

As seen in figure 1.3 and figure 1.4 the algorithm was capable on all counts to correctly identify each person correctly.

**References**