Automatic Attendance system using Facial Recognition

Aayush Upadhyaya,Aditya Agarwal, Aniruddha Deshpande,
Jeevesh Kataria, P.V. Haridatta , R.V. Sricharan

1 Abstract

The project deals with the automatic face-recognition attendance system. The camera is placed outside the classroom. For every person who enters the class , the camera captures the image and sends the image to the Load Balancer. As there are multiple classes running in parallel, the load balancer is getting multiple requests at the same time. It sends the image to server which is having less load.The server extracts the face from the image.After extracting the face it detects the fact to which person it belongs to.After getting the person name , it marks the attendance of that student in attendance repository.

2 Solution Overview

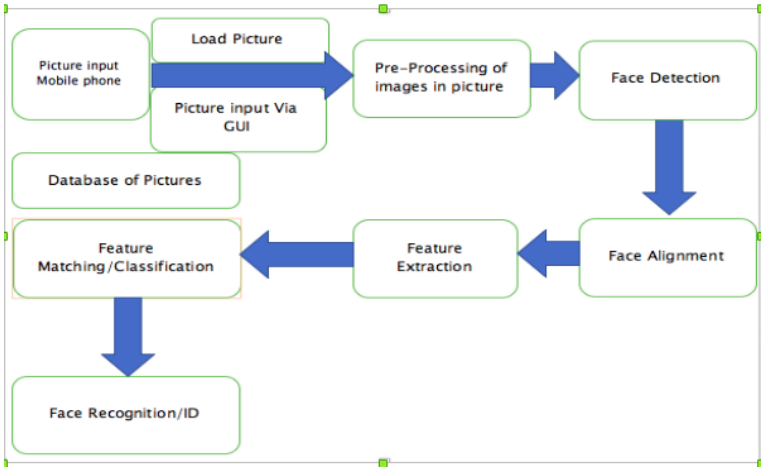

Figure 1: Face detection and recognition flow diagram

From the figure, above, Face Detection or face detector will detect any given face in the given image or input video. Face localization, will detect where the faces are located in the given image/video, by use of bounding boxes. Face Alignment is when the system will find a face and align landmarks such as nose, eyes, chin, mouth for feature extraction. Feature extraction, extracts key

features such as the eyes, nose, mouth to undergo tracking. Feature matching and classification. matches a face based on a trained data set of pictures from a database of pictures. Face recognition, gives a positive or negative output of a recognized face based on feature matching and classification from a referenced facial image.

3 Rumour Detection System

3.1 Introduction

Our ML project would be the Rumour Detection System wherein our main objectives include determining rumour veracity and support for rumours. Here we focus on stance classification of tweets towards the truthfulness of rumours circulating in Twitter conversations in the context of breaking news. Each conversation is defined by a tweet that initiates the conversation and a set of nested replies to it that form a conversation thread. The goal will be divided into following two subtasks:

Subtask 1 : We will be addressing the challenge of rumour stance classification, which involves identifying the attitude of Twitter users towards the truthfulness of the rumour they are discussing. Stance classification is considered to be an important step towards rumour verification, therefore performing well in this task is expected to be useful in debunking false rumours. **In this work we will be classifying a set of Twitter posts discussing rumours into either supporting, denying, questioning or commenting on the underlying rumours.**

They are described as

- **Support** : The author of the response supports the veracity of the rumour they are responding to.
- **Deny** : The author of the response denies the veracity of the rumour they are responding to.
- **Query**: The author of the response asks for additional evidence in relation to the veracity of the rumour they are responding to.
- **Comment**: The author of the response makes their own comment without a clear contribution to assessing the veracity of the rumour they are responding to.

Subtask 2 : (Verification) Verify it to classify the accuracy of the rumor as true, false or unverified.

3.2 Dataset to be used

We will be using the following dataset to be trained and tested. Dataset Used : Full English Twitter dataset
`https://figshare.com/articles/RumourEval_2019_data/8845580`

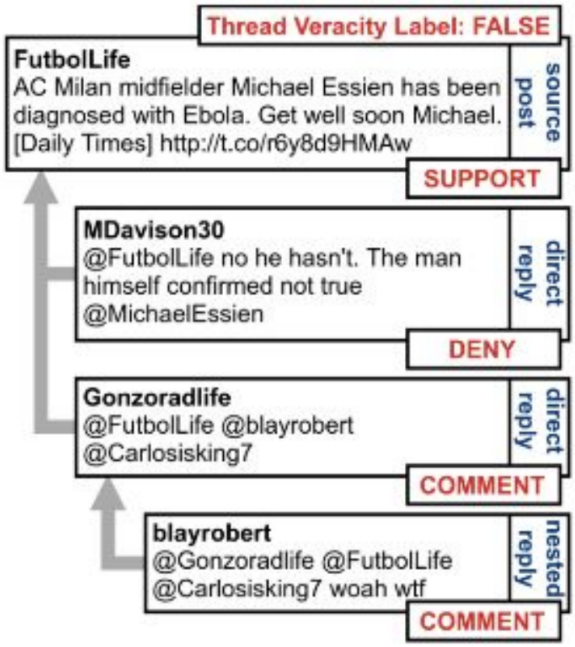

Figure 2: Tweet Labelling

3.3 Plan on doing and implementing Milestones :

Data Preprocessing - Representation of tweet in the format that can be fed to model.

Subtask 1 - Neural networks based model is proposed to classify each post of thread as support, deny, question or comment. (Tentative 21-October) To train our Model we will using Data parallelism .We will be having five servers, a parameter server and four other servers on which our data is distributed,while we train our data after each iteration each node send its weight to parameter server ,the parameter server aggregates the weight function and send the updated weights to each server which is used by next iteration , it will go on until we reach the convergence.

Subtask 2 - output from first subtask along with other features are again fed into RNN based model that estimates source post as true, false or unverified. (Tentative 11- November)