**Face Recognition Attendance Marking System**

Bibek Rawat(18BCS6728), Shreyansh Raut(18BCS1703), Subham Sah(18BCS6537), MD Tauhid Alam(18BCS6588)

Computer Science, University Institute of Engineering

Chandigarh University, Mohali, Punjab

**Abstract**— In this paper we deal to overcome the barriers that a traditional Attendance Management system holds and Introduce Deep Learning Technique to detect and recognise the faces. Face Recognition has been a long standing problem in computer vision. Recently Histogram of Oriented Gradients have been proven to an effective in descripting objects in general and faces in particular. The main focus of this paper is to recognise whether the face given as an input corresponds to a registered user in the database. Face Recognition is done using Histogram of Oriented Gradients (HOG) technique in the given image database. The features extracted from the real time subject are compared against the database. The proposed method checks whether a test image in different pose and lighting conditions is matched correctly with images of facial database. The results of the proposed approach show minimal false positives and improved detection accuracy.

**Keywords**— Face Recognition, Feature Extraction, Histogram of Oriented Gradients, face landmark estimation, gradients.

1. **INTRODUCTION**

The identification of a person mainly depends on an important feature face. It is so obvious that humans are experts at this task of recognising the faces of people in many odd conditions like. Different light conditions, different angles, variable distances. However for a computer system, it requires many complex parameters and tuning them until it reaches an optimal detection and many iterations are even to be considered to reach there.

There are many facial features that make up to form a human’s face which includes the eyes, nose and mouth are prime landmarks in marking one’s face. But, these features are just difficult in order to optimally recognise the face because, different expressions and pose variations makes it difficult.

Face Recognition is the first and foremost step in identifying the presence of faces in an image. Then face recognition is carried out which checks whether a given test input is matched with already available faces in image database. Facial Recognition tasks are performed in many areas of vision applications where security is more focused.

1. **RELATED WORK**

Face Recognition method mainly deal with image which are of large dimensions. To overcome this dimensionality reduction is introduced for the images in the database before extracting the facial features from the faces and a basic naïve flask application is wrapped around the main project as an interface between the user and the core application.

1. **SYSTEM ARCHITECTURE**

The core Application have the following modules as follows and are co performed as a whole [1].

1. **Face Detection:** The images are dimensionally reduced to be accommodated by the model. And the images are turned into grayscale as shown in Fig. 1. Next task is to figure out how dark the current pixel is compared to the surrounding pixels. Then an arrow is drawn showing in which direction the image is getting darker. The process is repeated for every pixel in the image, we end up with every pixel being replaced by an arrow. These arrows are called ***gradients.*** They show the flow from light to dark across the entire image. If each and every pixel is replaced with the gradient it may lead to the over fitting. So we consider a 16x16 pixel grid and find the gradient for the centred pixels as shown in Fig. 2.



Fig. 1 a part of converted Grayscale Image



Fig. 2 part of converted Gradient Image

The converted gradient images provide the layout of the face, we don’t use the traditional pixel values because they vary from image to image but gradients follow the brightness changes, both really dark images and really bright images will end up the same exact representation which makes it robust.

The image as a whole is known as the ***Histogram of Oriented Gradients (HOG).***

To find the faces in the HOG image, we have to find the part of the images that looks similar to known HOG pattern that was extracted from a bunch of other training faces.

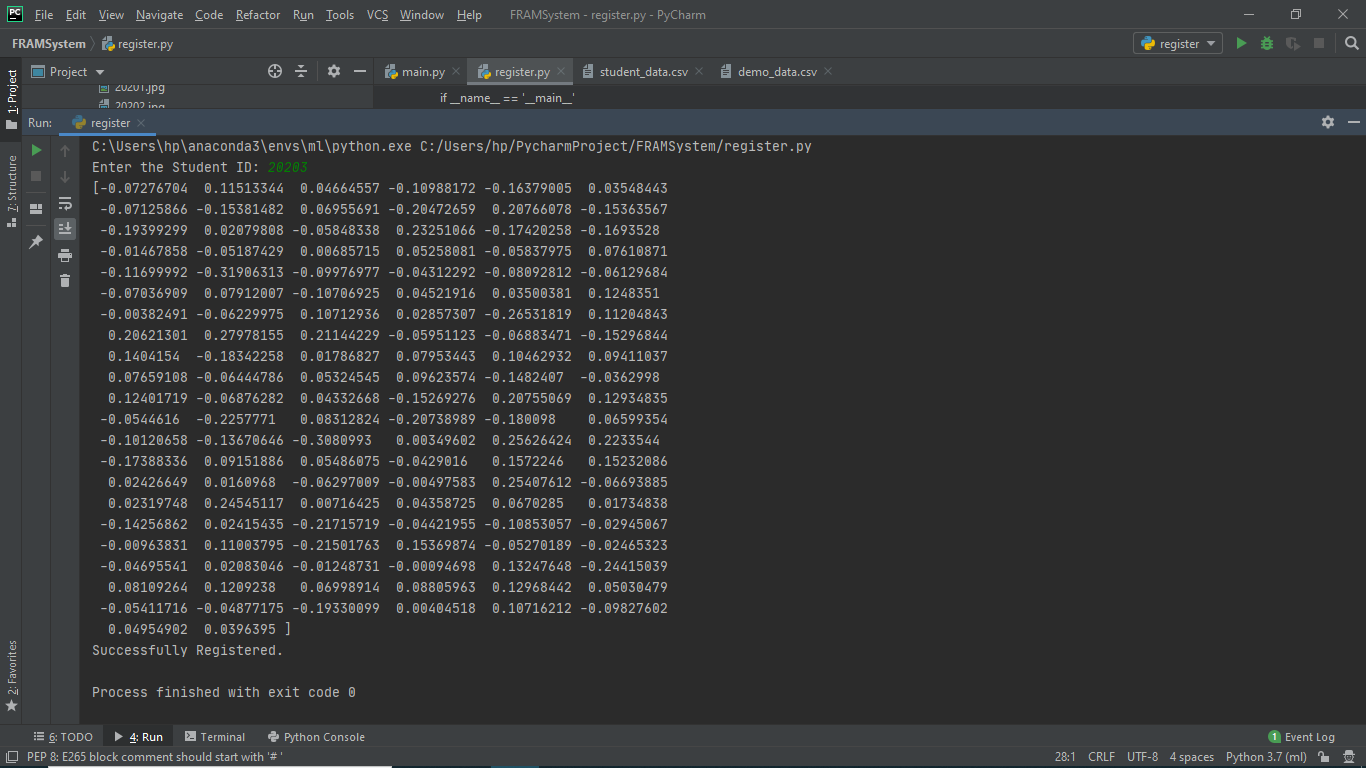
1. **Posing and Projecting Faces:** To do this we will try to wrap each picture so that the eyes and lips are always in the sample place in the image. This will make it a lot easier for us to compare faces in the next steps. For this ***face Landmark estimation*** algorithm has been used. The basic idea is we will come up with 68 specific points (called landmarks) that exist on every face. Then we have trained the algorithm to be able to find these 68 specific points on any face.

Based on this points simple transformations are performed in order to rotate and scale the point coordinates into the centre making it robust to distortions and help the model to compare the facial features of the images.

1. **Encoding faces:** In order to recognise the faces we need some measurements to compare against the other faces. Here we use a trained deep convolutionalNeural Network. It generate 128 measurements for each face which is also known as ***embedding*** as shown in Fig. 3.

The training process works by looking at 3 face images at a time.

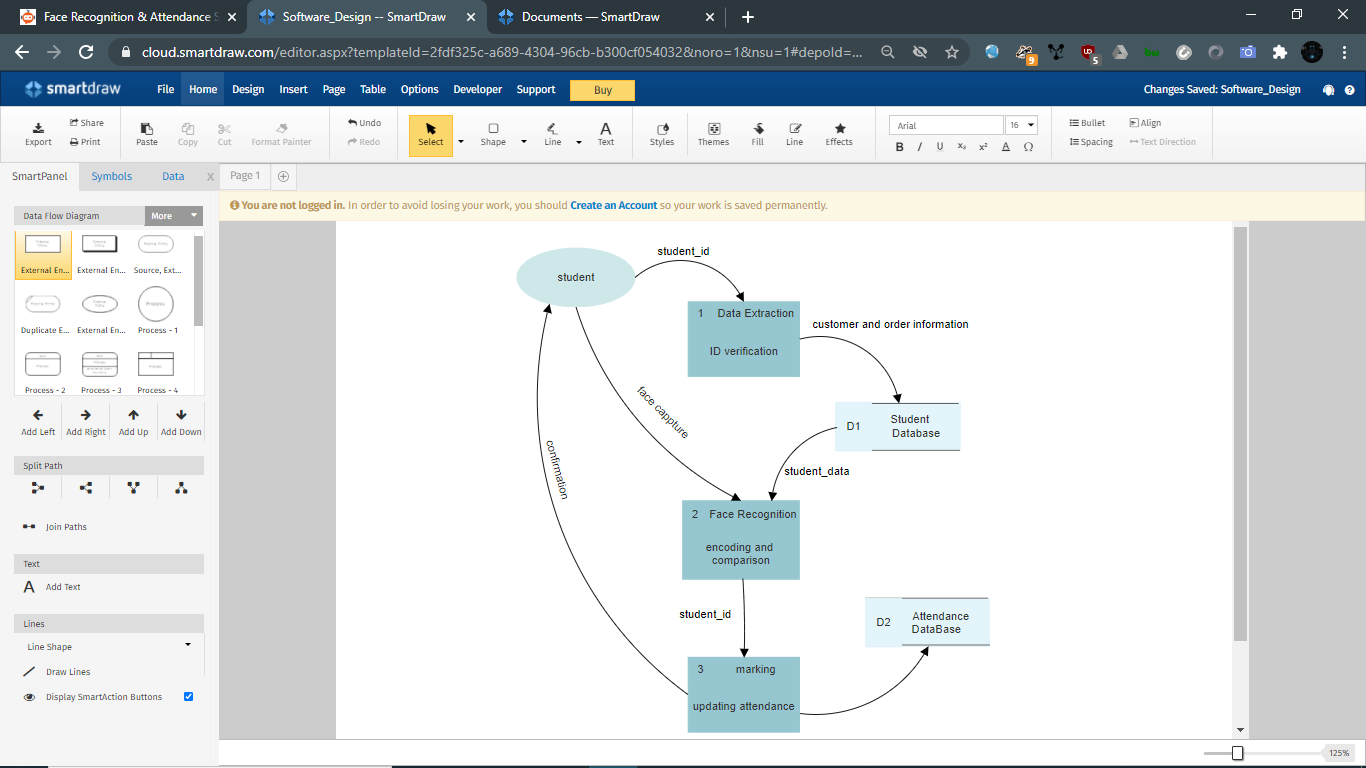
1. Load a training face image of a known person.
2. Load another picture of the same known person.
3. Load a picture of a totally different person.

Fig. 3 measurements generated by the neural network

The algorithm makes sure that for each of those three images. #1 and #2 are slightly closer while making sure #2 and #3 are slightly farther apart. For this part we use an already trained model from OpenFace [2] since it’s computationally so expensive to bring up.

The measurements are since randomly and precisely generated by Neural Network we have no idea what these measurement values represent.

1. **Face Recognition:** Since we have got the facial features as 128 measurement values. We finally compare these values using basic machine learning classification algorithm. Simple linear SVM classifier. Using this we find the closest measurements to our test image.

­­­

1. **RESULTS AND DISCUSSION**

In this experiment, the model is already trained and uses a very less amount of data to distinguish between the people. For every person we have we used only one image of the person to encode the measurements and with real time feed of the data the faces are compared against each other and later the accuracy points are outputted.

Method Comparison

|  |  |  |  |
| --- | --- | --- | --- |
|  | **LBP** | **HAAR** | **HOG** |
| **TPR%** | 60.37 | 78.23 | 92.68 |
| **FNR%** | 39.65 | 21.76 | 7.31 |

The above table is based on [5], here we compare LBP (Local Binary Pattern), HAAR (Haar-like cascade), HOG (Histogram of Oriented Gradients) base on two factors TPR (True Positive Rate) and FNR (False Negative Rate) and from the table we can easily say that HOG method way more accurate than others.

In my dataset, all the people were exactly classified giving a total accuracy score of 100% out of a dataset of 4 people.

The model is working with high Accuracy during optimal and medium light conditions and in different angles. While, during minimal light conditions the model is a bit struggling.

Input and output formats, the model has already given a set of images for training and the student data to check over the names as shown in Fig. 5.



Fig. 4. 6588.jpg image

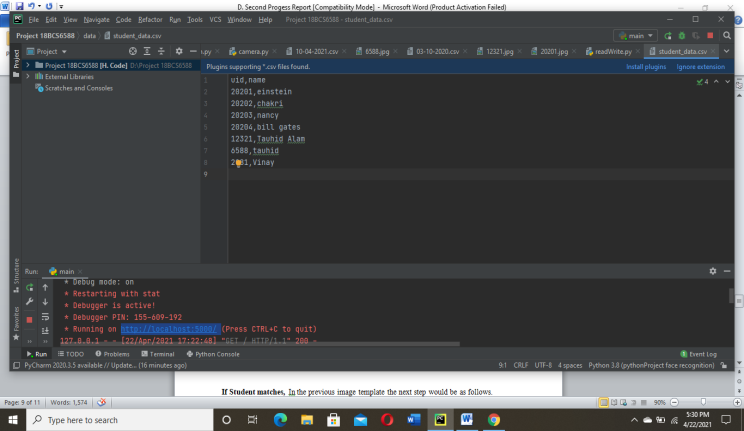
****

Fig. 5. Student dataset

Although there is a lot of noise in the background of image as shown in the Fig. 4. The model id robust to such noise and even gives us the exact results.

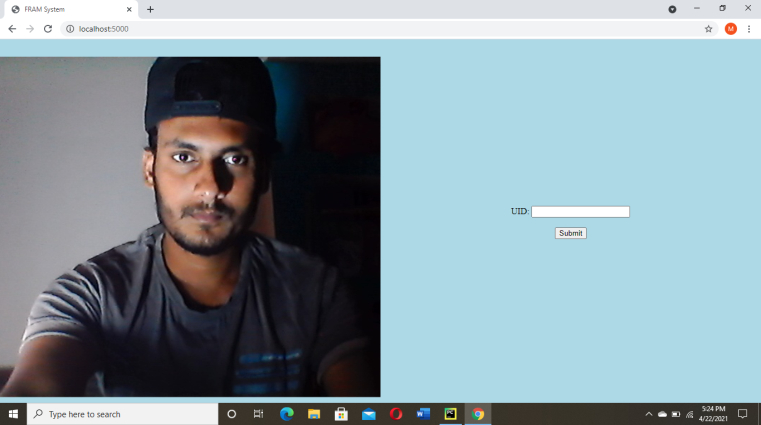


Fig. 6. Footage from camera while marking attendance (compares this)

On a successful recognition, marks the student present, else doesn’t and waits for the others to get themselves marked.

1. **FUTURE SCOPE**

As we know, Facial recognition is not a new topic. It has been a part of our natural instinct for any living being from the very past in order to classify between the friends and foes. By automating the whole system, fixing the misconceptions and by introducing some security measures we will end up saving a lot of time which is very precious. As of my knowledge we use the HOG (Histogram of Oriented Gradients) as a feature extractor as it uses the gradients which is new boom in this facial recognition task and to classify the face after the feature extraction we used the SVM (Support Vector Machines) as a classifier for our classification purpose, we can use many other classifiers like that where each qualifies its own pre-set of features.

By introducing the fraud and mischief detection with means of these classifier we can easily get to know about the candidate, who is responsible and also in many other ways by introducing many other feature providing the good representation.

1. **CONCLUSION**

This paper explores the use of HOG features for face recognition. The contributions are

1. To provide robustness to facial feature detection, we propose to uniformly sampling the HOG features.
2. To remove redundancy in the data, improve computational efficiency and avoid over fitting.
3. To encode the facial features i.e., feature extraction been done automatically using the Neural Network.

Taking these into consideration we were able to obtain a significant increase in recognition performance on dataset.

**ACKNOWLEDGEMENTS**

Portions of the research in this paper use open sourced facial recognition library [3], which was authored by Adam Geitgey, I would sincerely like to thank him for the effort, he made to do bring the HOG method accessible to everyone.

**REFERENCES**

[1] <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>

[2] <https://cmusatyalab.github.io/openface/>

[3] <https://pypi.org/project/face-recognition/#data>

[4] <https://www.sciencedirect.com/science/article/abs/pii/S0167865508001104>

[5] <https://www.researchgate.net/profile/Pg_Mohan/publication/305709603_Improved_Face_Recognition_Rate_Using_HOG_Features_and_SVM_Classifier/links/57aedbba08ae95f9d8f11b57/Improved-Face-Recognition-Rate-Using-HOG-Features-and-SVM-Classifier.pdf>

[6] <https://www.sciencedirect.com/science/article/abs/pii/S0167865511000122>

[7] <https://ieeexplore.ieee.org/abstract/document/7944082>