Automated Attendance System using Multiple Face Detection and Recognition

Shashwat Dobhal[1], Avinash Kumar Singh[2], Ajay Kumar[3]

[1,2]Student, Dept. of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani, Chennai, India
[3]Assistant Professor, Dept. of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani, Chennai, India

---***---

Abstract - *In today's world, everybody has a unique identity, which is their face. The face or as in facial features cannot be copied or replicated. In schools and colleges, time is of the essence. Teachers and professors cannot waste their time in taking attendance as they can be doing something productive in that time. One period is of usually 50 minutes to 1 hour, where 10 to 15 minutes are wasted in the process of taking attendance. In the traditional method, a teacher manually takes attendance, which takes up a lot of time as human interaction is required at both ends. For every tutor, this is a wastage of time. So to avoid these drawbacks, an automatic process will be used in this project, which is based on image processing. In this project, face detection and recognition are used in a group to save time. Face detection is used to locate human faces in a group, and face recognition is used to recognize their faces. The attendance of all the students in the class is stored in a database. When the front face of the individual student matches one of the faces stored in the database, then the attendance is marked, present for all available students in the class, and absent in other cases.*

***Key Words*: LBPH, OpenCV, HOG, Multiple Face Detection, Haar Cascade.**

1. INTRODUCTION

In today's world, maintaining attendance is a critical point in ensuring that a particular student is present in a school or an institution. A lot of methods are currently present, which takes attendance through various means. The conventional way of taking attendance is a manual method where human interaction is required. Generally, one period is of about 60 to 50 minutes, where 10 to 15 minutes are wasted on taking attendance. This traditional method is time-consuming and, at times, inefficient due to reasons like a proxy. Due to these reasons, many institutions are using biometric systems like fingerprint scanners, facial recognition, etc. But, in all these methods, time consumption is the main issue because each student has to go and give their attendance, be it a fingerprint scanner or through facial recognition. These methods can cause a lot of wastage of time. In order to automate this process, multiple face recognition is used. Multiple faces are detected and recognized at a time to mark a student absent or present. This process eliminates human interaction, which ultimately saves time while taking attendance in the class.

2. RELATED WORK

In the present day scenario, we have many attendance systems available with us which use different types of technologies. The most basic method of taking attendance is the manual method, but in this method, time consumption and human dependency are way too high. Also, social interaction is required at both ends. Engr. Imran Anwar Ujan and Dr. Imdad Ali Ismaili came up with a Biometric Attendance System [1], which uses a fingerprint to mark anyone present or absent. Their system is accurate, but the time consumption is high because every individual has to go and register their attendance. This system is not very efficient in a teaching environment where time is of the essence. Aamir Nizam Ansari, Arundhati Navada, Sanchit Agarwal, Siddharth Patil, and Balwant A Sonkamble came up with a system for automated attendance using RFID, Biometrics, [2]. This system used RFID and fingerprint both to mark absent or present. Their system is secure, but the time taken by the system to mark everyone present or absent is more as each individual has to go and mark their attendance manually. This system is not very efficient in a teaching environment as every individual student has to go and mark them present hence increasing the time taken for attendance. E.Varadharajan, R.Dharani, S.Jeevitha, B.Kavinmathi, and S.Hemalatha created an attendance system using facial recognition using face detection [3]. In their software, they used the Eigen Face method for facial recognition. Since we have seen that the LBPH algorithm performs better [4] than the Eigenface method in certain conditions with confidence factor ranging between 2-5, and it also has minimum noise interference. Therefore, we came up with a system that uses multiple face detection and recognition to mark an individual present or absent efficiently and with less time consumption.

3. PROPOSED SYSTEM

To overcome these problems with the existing system, we came up with a novel approach, which is an automated attendance system that uses multiple face detection and recognition to mark a particular individual present or absent.

Modules which are present in our proposed system are:

- Image Capturing
- Face Detection
- Face Recognition
- Attendance Management

Fig -1: Architecture Diagram

3.1 Image Capturing

In this project, we will be capturing test images from a video feed on an interval of 10 minutes, at least three times. This is to ensure that every student present in the class gets his or her face profile detected for marking attendance. This process can be repeated for any given interval of time as per the user's requirement. Capturing images several times, allows us to send every picture of every student that might have been left out the first time.

3.2 Face Detection

Face detection is achieved in this project using Haar Cascade classifier [5], it can be used for object detection, but here we will be using it for detecting faces.

Paul Viola and Michael Jones came up with an idea for object detection using Haar feature-based cascade classifiers, which is a very efficient object detection method[7]. It's a machine learning-based approach where a cascade function is trained from plenty of positive and negative images [5] so it can get accustomed to detect objects in other models.

Initially, the algorithm needs a great deal of positive-negative images to train the classifier. Then features are extracted from them. For this, Haar features shown in Fig-2 are used. They're similar to our convolutional kernel. Each single value is represented by a feature which is obtained by subtracting the sum of pixels under a white rectangle from the sum of pixels under a black rectangle.

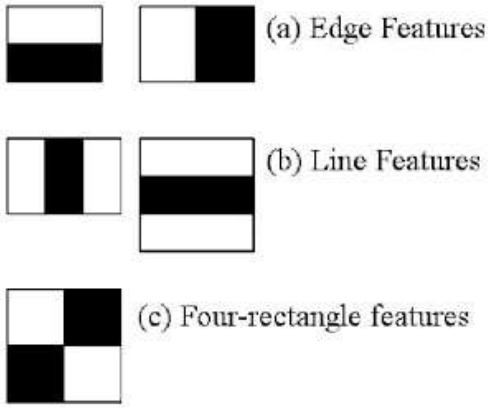

Fig -2: Cascade Classifiers

Now all possible sizes and locations of every kernel are used to calculate lots of features. But even a 24x24 window ends up giving over 160000 features [5]. For every feature calculation, we have to calculate the sum of pixels under white and black rectangles [7]. To resolve this, integral images were introduced. It simplifies the calculation of the number of pixels, however large may be the number of pixels, to an operation involving just four pixels.

In Fig-2, the top row shows two useful features. The first feature selected seems to select the property region of the nose and cheeks, which is usually lighter than the part of the eyes. The next feature relies on the property that the bridge of the nose is brighter than the eyes because of the difference in depth [7]. But identical windows, when applied on cheeks or the other place is irrelevant. So, to pick the suitable features out of 160000+ features, Adaboost is employed [7].

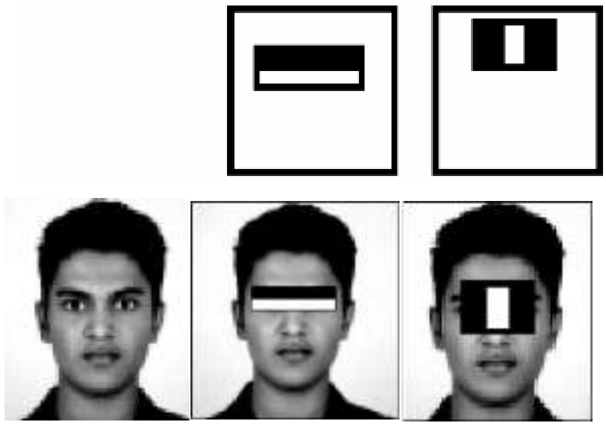

Fig -3: Edge Detection

For this, every feature is applied on all the training images. For every feature, best threshold is calculated which can classify the faces into negative and positive. But there will be errors so we select the features with a minimum error rate, which implies they're the features that best classifies the face and non-face images.

Some of these classifiers are called weak classifiers because they alone can't classify the image, but in conjunction with others they form a powerful classifier. The final classifier thus obtained is a weighted sum of these weak classifiers. In a picture, most of the image regions may be a non-face region. So, having a technique to ascertain if a window could be a face region or not is a better idea [5]. If it's a non-facial region, discard it don't process it again. Instead, concentrate on the region where there could be a face. This way, we are able to find save a lot more time to examine possible facial regions.

Fig -4: Several frontal faces detected

We can see in Fig-4 that only frontal faces are being detected. The concept of Cascade of Classifiers was introduced later on, which suggested that, rather than applying all the features on a window, group the features into different stages of classifiers and apply one-by-one [5]. Ranging from a smaller number of features which can be as low as a single feature in initial stages and increasing the number of features applied to pictures gradually at each stage. If a window fails the initial stage, discard it. We do not consider remaining features on it. If it clears the initial stage, apply the second stage of features, and continue the process. This explains the working of Viola-Jones face detection in layman's term [7].

3.3 Face Recognition

Local Binary Pattern is a very powerful texture operator, described in 1994 and has since been acknowledged as a dominant algorithm for texture classification [6]. The pixels of the picture are labeled by thresholding the neighborhood of every pixel, and the binary number was considered as the result. It was observed that the detection performance was considerably improved on some of the datasets when LBP was combined with histograms of oriented gradients descriptor. LBPH is one of the simplest face recognition algorithms out there. It can represent local features within the images and is robust against monotonic grayscale transformations [6]. It is provided by the OpenCV library [8]. The face images can be represented with a much simpler data vector when we use LBP combined with histograms.

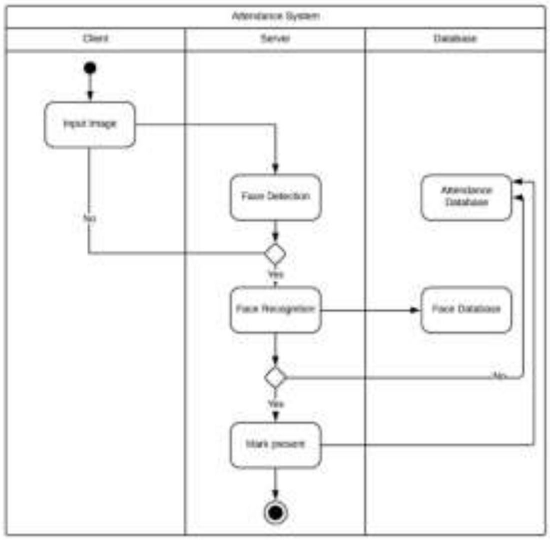

Fig -5: UML Diagram

Training the Algorithm: The first step would be to train the algorithm. To do so, we will use a dataset with the facial images of the people we want to be recognized. We also need to set an ID for each image, so the algorithm can use this information to identify an input image and give you an output.

Applying the LBP operation: The initial step of the LBPH is to create a transitional image that describes the original picture in a better way, by outlining the facial region. This is achieved by the algorithm using a concept of a sliding window, supported by the parameter's **radius** and **neighbors**.

The Fig-6 shows the procedure:

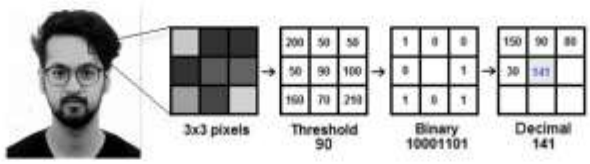

Fig -6: LBP Matrix

- After this stage is completed, we get a new image that represents better characteristics of the original image as shown in Fig-7.

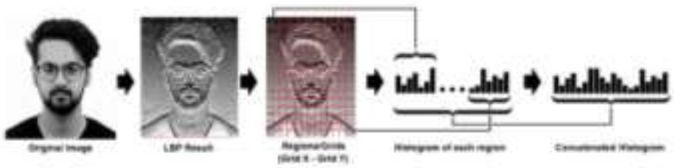

Fig -7: Characteristic Image

Based on the image obtained after applying the LBP operation, we can extract the histogram of each region as follows:

- As we have a picture in grayscale, each histogram will contain only values ranging from 0 to 255, representing the occurrences of every pixel intensity.
- Then, we need to combine each histogram to make a bigger histogram. Assuming we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram [6]. This last histogram will represent the characteristics of the original image.

Performing face recognition: Once the algorithm is trained, each histogram created is used to represent each image from the training dataset [6]. So, given an input image, we will perform the above steps again for the test image and create a histogram that represents the image.

- To find the picture that matches the test image, we have to compare two histograms and return the image with the closest histogram.
- There are various approaches to compare the histograms Euclidean distance, chi-square, absolute value, etc. In this project, we will be using, the Euclidean distance [6] based on the following formula given in Fig-8:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

Fig -8: Euclidean Distance Formula

- The algorithm output will be ID from the image with the closest histogram. The algorithm will also return the calculated distance, which will be used as a 'confidence' measurement. Lower the confidence better the result, because it means the distance between the two histograms is much closer when compared with histograms from other images.
- We will then set a threshold for 'confidence' and assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

3.4 Attendance Management

The teachers can manage attendance through this, and it is stored in the database. Students can access and view their attendance through an online portal, whereas teachers and professors can make necessary changes to it. By default, everybody is marked absent. If the algorithm recognizes the face of the student, he/she is marked present in the database else they are marked absent. Attendance of all the students who are present or absent, their record is maintained in these databases, which can be accessed by our faculty members at any point in time.

4. CONCLUSION AND FUTURE SCOPE

So, an automated system was developed by us, which helps to mark an individual absent or present. It is an excellent tool for schools and institutions where time cannot be wasted on taking attendance; instead, it can be focused on more productive things. The existing system lacks proper time management, and a lot of time is wasted in taking attendance. In our system, that time is saved as we are using facial recognition on multiple faces to mark present or absent. Our proposed method doesn't work in real-time, and it doesn't monitor the students, which includes problems like what to do if the student leaves the class. Since our system has a modular approach; therefore, any advancements in the future can be directly added to the system, ultimately increasing its efficiency.

REFERENCES

[1] Imran, Engr & Ujan, Imran Anwar & Imdad, Ali & Ismaili,. (2011). Biometric Attendance System. 10.1109/ICCME.2011.5876792.

[2] Aamir Nizam Ansari, A. Navada, S. Agarwal, S. Patil and B. A. Sonkamble, "Automation of attendance system using RFID, biometrics, GSM Modem with .Net framework," 2011 International Conference on Multimedia Technology, Hangzhou, 2011, pp. 2976-2979.

[3] E. Varadharajan, R. Dharani, S. Jeevitha, B. Kavinmathi and S. Hemalatha, "Automatic attendance management system using face detection," 2016 Online International Conference on Green Engineering and Technologies (IC-GET), Coimbatore, 2016, pp. 1-3.

[4] SudhaNarang, Kriti Jain, MeghaSaxena, AashnaArora (2018); Comparison of Face Recognition Algorithms Using Opencv for Attendance System; Int J Sci Res Publ 8(2) (ISSN: 2250-3153).

[5] L. Cuimei, Q. Zhiliang, J. Nan and W. Jianhua, "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers," 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), Yangzhou, 2017, pp. 483-487.

[6] A. Ahmed, J. Guo, F. Ali, F. Deeba and A. Ahmed, "LBPH based improved face recognition at low resolution," 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, 2018, pp. 144-147.

[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Kauai, HI, USA, 2001, pp. I-I.

[8] Bradski, G., 2000. The OpenCV Library. Dr. Dobb's Journal of Software Tools.