Smart Attendance Management System Using Raspberry Pi and Deep Learning Technique

L Ashok Kumar[1], V.Indragandhi[2], Chitra.A[3], Rajat Paul[4], Saswata Banerjee[5]
{lak.eee@psgtech.ac.in[1], indragandhi.v@vit.ac.in[2], chitra.a@vit.ac.in[3],
rajat.paul2016@vitstudent.ac.in[4], sashwata.banerjee2016@vitstudent.ac.in[5]}

[1]Professor, Dept. of EEE, PSG College of Technology, coimbatore, [2,3,4,5]School of Electrical Engineering, VIT, Vellore

Abstract: Smart attendance system using face-recognition in real-time is an essential real-world solution which comes with day to day activities like handling employees/students. This task indeed is a challenge, as performing background subtraction in an image in real-time is still a highly complex operation. This works attempts to develop a smart attendance management system using deep learning technique. To detect real-time human face Haar Cascade classifier is used and a simple fast Local Binary Patterns Histograms (LBPH).Face recognizer are used to accurately recognize the faces that are identified. The employees' attendance is then recorded using the matching face.After marking the attendance, the names are stored in a file which is then sent automatically to the authorised person. This work employs raspberry pi for real time implementation of the proposed smart attendance scheme.

Keywords: Deep Learning, Machine Learning, OpenCV, Computer Vision, Raspberry pi, NVIDIA Jetson Nano.

1. Introduction

In today's dynamic world, taking attendance in a classroom or a multinational corporation can be time-consuming and sometimes stressful. The method of attendance keeping is a complicated process if performed manually. The smart and integrated attendance management framework can be introduced using diverse biometrics methods. Another of them is facial recognition. The problem of fake participation and proxies can be solved by using this program.Image processing has evolved over the last decade with blooming technologies at economical reach. So, the state of the art tools to process images very easily with a smaller code footprint is easily available. There is a massive research is in progress in this domain.

We have Haar Cascade classifier to detect faces, nose, lips, eyes or even emotions. Using OpenCV DNN module (Deep Neural Network) we can classify images pretty easily. With these and other libraries combined, we can develop software for surveillance, identification, segregation of objects etc. It can also be used to simplify our daily tasks. There were some drawbacks in the previous face recognition-based attendance program, such as the strength of the light problem and head pose question. Without the involvement of employees, our application employs the facial recognition approach to automate employee attendance in the office room area. Face recognition is divided into two stages: first, faces are recognised in the image, and then these discovered faces are compared to the database for verification.

ICCAP 2021, December 07-08, Chennai, India

DOI 10.4108/eai.7-12-2021.2315121

The main motivation of the work is to develop a system which can be implemented anywhere, it may be a classroom or in an auditorium. The system is a low-cost module and mobile which can be implemented everywhere. It should replace the conventional method of calling out names or signing papers, which takes a lot of time. The system should be reliable. We were motivated to develop a system by combining the power of deep learning and raspberry pi to help the society and something which can be implemented in the near future instead of providing a proof of concept.

2. Literature Review

This section consists of the previous literature works available in the domain of the attendance system.

2.1 An automated device for attending face-recognition:

Automated Face Recognition attendance program indicates that the system is based on face finding and recognition algorithms that are used to routinelyclassify the student's face when he/she enters the class and that the system is able to identify the attendance.

2.2 Attendance of students using Iris Detection System:

For this proposed process, the student is asked to stand before the camera to detect and identify the iris for the computer to mark the attendance of the students. Skin Pixel Detection is used to detect the iris using several algorithms such as GrayScale Conversion, Six Segment Rectangular Filter. This helps to prevent proxy issues, which guarantees the involvement of the student in an effective way, but in one of the time-consuming process for a student or staff to wait until the previous members are done.

2.3 Face Recognition based lecture attendance programme:

This paper suggests the program should take automatically consideration of the attendance obtained through continuous observation. Continuous observation helps to estimate and improve the performance of those attending. In taking attendance, the locations and facial images of the students present in the classroom are collected. Each student's seating position and location for attendance marking are determined by continuous observation and system recording. The work focuses on the method by which each oriented seat's different weights are obtained according to its location.

2.4 Fingerprint Recognition system based on:

In the current attendance system based on Fingerprint a portable fingerprint app has to be configured earlier with the student's Fingerprint. The student needs to record the Fingerprint on the installed computer later to ensure their attendance for the day, either during lecture hours or before.

2.5 RFID (Radio Frequency Identification) Reconnaissance system based on:

In the current RFID-based system, students must bring a Radio Frequency Identity Card with them and put the ID on the card reader to record their presence for the day. The computer connects to RS232 and records the attendance to the database that is saved.

3. Methodology

The entire work can be divided into three main divisions. The first part is data gathering where we create a dataset of images for various users. The second part is training the classifier

using the dataset we had created. The next part is to see the code in action. For creating dataset we may not use raspberry pi. After training we transfer the trained classifier into raspberry pi. We take video input in real-time using a webcam(Figure.1. to Figure.4).

Data Generation

Data generation is a vital part of any machine learning work. Bad data and a bad algorithm are a disaster for any work. A complex work needs a ton of examples to learn from. If the training data is full of errors, missing data, outliers and noise, the algorithm may find it very difficult to find a pattern. In facial recognition, we need clear images for training an algorithm and a bit of pre-processing, to begin with. In our work, we used Haar Cascade Classifier to detect faces first and then we cropped out the detected face and stored them in a folder. We trained the classifier later with these images.

3.1 Face Detection

On Facial Recognition the most important function is facial detection. Before anything, to recognize it, we must capture a face when compared to a new face captured in the future.

The "Haar Cascade Classifier" is the most common way to detect a face or any object. Object detection using Haar feature-based cascade classifiers is an efficient method of detection of objects proposed in their paper by Paul Viola and Michael Jones in 2001. OpenCV comes with a detector and a guide. Using OpenCV, anyone can train their classifier for any objects such as cars, planes etc.

Using OpenCV's Video Capture class we took live stream of video from the webcam of a user. We set the user id beforehand and also specified the path of the classifier which is trained to detect faces in an image. The script reads images frame by frame using cam.read() function. We converted the input RGB or coloured image into a grayscale image. After that, we set the hyperparameters in the detectMultiScale() function as per our requirement. The detector gave us the bounding box coordinates of the area in which a face is detected i.e., top left corner point (x,y), width (w) and height (h). Using OpenCV's rectangle function we cropped out the region of interest and saved the image in a folder. Each image's name consists of user id which is nothing but the class id(Figure.1). This entire process is done for all the images in the stream. The number of images to be captured can be controlled in code. We set it to 50 images per user. This code has to be run individually for all the users to generate images for as many classes.

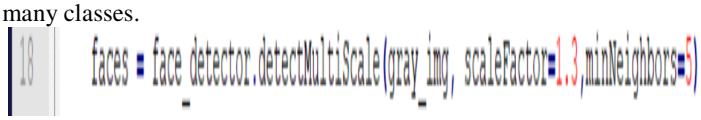

Fig.1 Classifier function to detect faces in an image

Gray: The input grayscale image.

ScaleFactor: the parameter specifying how much the image size is reduced at each image scale.

MinNeighbors: Parameter specifying how many neighbours, in which higher number gives lesserwrong positives.

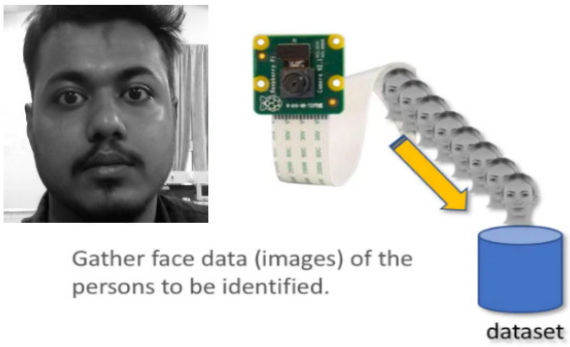

Fig.2. Data gathering

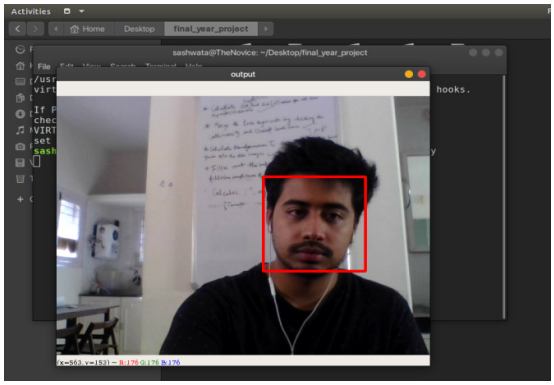

Fig.3. Face Detection in real-time

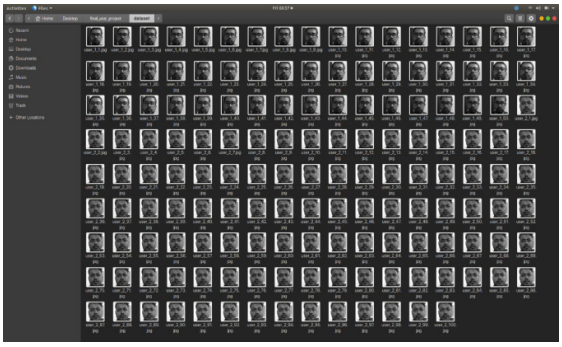

Fig.4. Dataset folder

3.2 LBPH

In the second step we took from the dataset folder all user data and from the OpenCVreco gnizer. It is achieved directly via a special feature in OpenCV. The result was a file

named trainer.yml that will be stored in a directory named “trainer.”We used the LBPH Face Recognizer as our recognizer. A file named “trainer.yml” will be saved in the trainer directory that was created previously(Figure.5.).

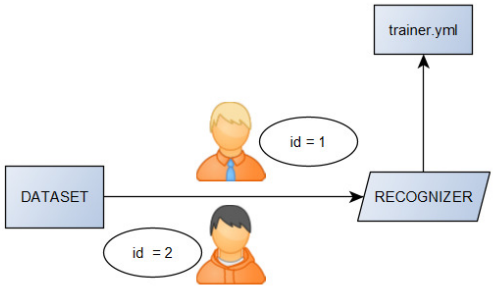

Fig.5. Block Diagram for Flow of Data

Face recognition in computer science is essentially the task of recognizing a person based on his / her facial image. During the last two decades, it has become very popular, mainly due to the new methods developed and the high quality of the current videos/cameras.We can represent the face pictures with a simple data vector using the Local Binary Pattern (LBP) and histograms. Because LBP is a visual descriptor, it may also be utilised for face recognition tasks, as seen in the following step-by-step description.

LBPH uses the following 4 parameters:Radius: It is typically set to 1. Neighbours: Number of sample points for constructing the local binary circular pattern. Grid X: cells number in the horizontal direction. Typically set to 8.Grid Y: cells number in the vertical direction. Typically set to 8.

3.3 LBP operation

The LBPH’s first computational step which uses a sliding window principle, based on the distance and neighbouring parameters(Figure.6).

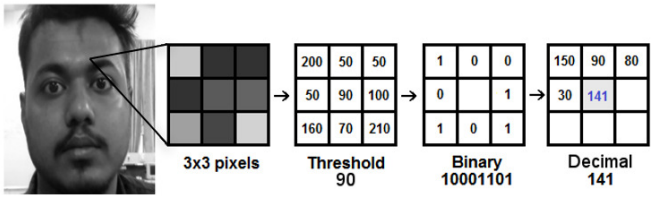

Fig.6. Applying LBP operation

Steps for LBP(Figure.7.):

- Get a facial image in Gray scale.
- A part of the image as a 3x3 pixels window is represented by containing the intensity of all the pixels
- Then, we have to take the central value of the matrix which is to be used as the threshold.

- This value is used to describe new standards from the 8 neighbours.
- We must create a new binary value for each of the core value's neighbours. We used a '1' for values that were equal to or higher than the threshold, and a '0' for those that were below.
- The matrix will only include binary values, with the centre value being ignored.
- We translated this binary value to a decimal value and assigned it to the matrix's centre v alue.
- Received a new image which represents better the appearances of the original image.

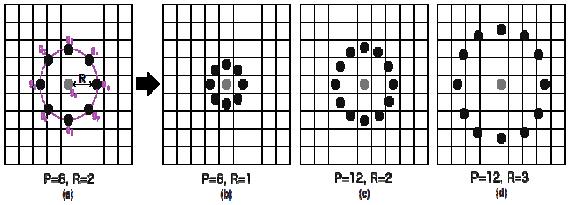

Fig.7. Circular LBP

Extracting the Histograms:

Using the picture created in the previous phase, we can now divide the image into numerous gr ids using the grid X and grid Y parameters, as seen in the accompanying image (Figure.8.).

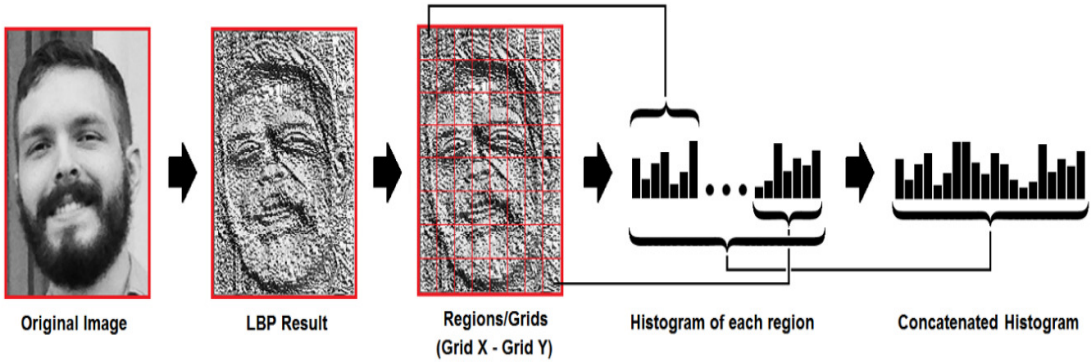

Fig.8. Generating histogram from an image

4. Design Approach And Results

4.1 ***Face Recognition***

The algorithm has been trained before. Every image is represented by a histogram created from the training sample. So, given an input image, we repeat the processes to generate a histogram that reflects the image for this new image.The Euclidean distance is calculated by using the equation.,

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

We also added a new series, so we will display "names," rather than numbered ids. Next, we'll detect a name, the same we did before with the classifier haarCascade. Having a face detected we can call recognizer. predict.It may assess a captured section of the face as a parameter and return the likely owner, along with the ID and how confident the recognizer is in the match. That the trust index returns "empty" when a perfect match is considered.

And finally, if the recognizer will predict a face, we'll put a text over the picture with the probable ID and how much the "probability" is in percentage that the match is right ("probability" = 100-trust index). If not, the face is placed on a "unknown" sticker(Figure.9.).

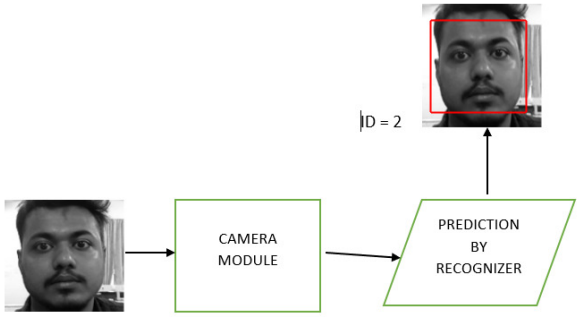

Fig.9.

4.2 Automating Attendance Sheet

After we have successfully detected faces and recognized them, it's time we store the data in the file. The outcome of face recognition is a list of classes with ids or names. We did not the names, instead, we stored the numerical ids to save memory. Attendance system has to be full-proof with much more precision score than accuracy score. That means the classifier should rather not detect a person rather than misclassify it. For our work, we did not train the classifier with many images, but in real-time application, much more images with variance should be used for better performance.

To improve the detection quality, we first stored all the detected in a list. After the process of taking attendance is over, we iterate the list delete multiple instances for each class. A person may be detected multiple times and that's a reality. We set a threshold of 10 to eliminate classes that have been detected less than the threshold and simultaneously if any class is detected more than the threshold then just consider one instance of it.

So now, we have a list of ids which have been properly detected. We take iterate through the list of ids and pick the corresponding names from the name list. We now need to write the names in a file. For that, we create a text file using python and write the names in the text file.

4.3 Automating Email Using Python

Python comes with the built-in smtplib module to use the Basic Mail Transfer Protocol (SMTP) for sending emails. Smtplib uses SMTP using the RFC 821 protocol. We need to adjust the security settings for our Gmail account to allow access from your Python code.Sending an e-mail with mixed content requires setting a multipart/mixed content-type header. Then, you can define parts of text and attachment within boundaries.

We need to give authorization in Gmail so that we can log in to email using python. We need to provide our python code with login credentials for login. If the connection is successful, then our python code will be able to sign in. It will automatically fill up the sender's email addresses, subject, body and attach the text file as instructed. Upon sending the email we get a confirmation message from python.

We tested the system in real-time, with a good enough lighting condition. The system detected faces and classified them with very good accuracy. We kept the probability threshold for classification as 80 percent. There were a few miss-detections, but we filtered them out in the post processing part. Initially, we trained it for binary classification with 50 photos per class to test the algorithm.

Fig.10. Face Recognition in real-time

In Fig.10, we see the model has correctly classified a face and tagged an image which was not in the dataset as an "Unknown" face. The confidence score is 0.53 as the experiment was done in a low light condition.

5. CONCLUSION

Thus, the main goal of the work is to capture video of students or employees in a good lighting conditions, extract faces from the video, and make prediction of the student's names by comparing the features from the database. After prediction the system will automatically enter their names in a file and send an email to an authorized person. The entire process will be in real-time.

Since, we used a mobile device with a low processing power, we were unable to deploy complex algorithms. We had to choose a model or classifier which is both light-weight and moderately effective. We used Local Binary Pattern Histogram for our task. We could have used Convolution Neural Networks but Raspberry pi could not handle such processing, as a

result frame per second rate would have been very low (1FPS), making it impossible to run in real-time.

LBPH has some advantages, like:

In summary, the system has a fair accuracy and can be implemented in real-time with a few changes in the hardware. Using hardware with parallel processing power, we can deploy complex algorithms to make it accurate with almost zero miss-detections.

Future Scope: Currently according to the resources present for the work allows us to generate an excel(.xml) file of the attending students/employees and mail it to the controlling authority.

It can also track the present percentage for a particular student, for a particular section or whole institution. If the present percentage goes below a particular threshold, it can automatically send a mail/message to the student/parent/employee.If any person is absent for a pre-set period continuously, it gives a prompt to the authority about it so that the authority can enquire about it.If any person is late for a pre-set time continuously for some days, the system can send the person a warning mail/message to the late comer.

References

[1]. Kumar, N. A., Swarnalatha, P., Chowdary, P., Naidu, J. K., & Kumar, K. S. (2019). Smart Attendance Marking System using Facial Recognition. Research Journal of Science and Technology, 11(2), 101-108.

Jadhav, A., Jadhav, A., Ladhe, T., & Yeolekar, K. (2017). Automated attendance system using face recognition. International Research Journal of Engineering and Technology (IRJET), 4(01).

[2]. Kanti, J., & Sharm, S. (2012). Automated Attendance using Face Recognition based on PCA with Artificial Neural Network. International journal of science and research IJSR.

[3]. Pei, Z., Xu, H., Zhang, Y., Guo, M., & Yang, Y. H. (2019). Face Recognition via Deep Learning Using Data Augmentation Based on Orthogonal Experiments. Electronics, 8(10), 1088.

[4].Pranav, K. B., & Manikandan, J. (2020). Design and Evaluation of a Real-Time Face Recognition System using Convolutional Neural Networks. Procedia Computer Science, 171, 1651-1659.

[5]. Waingankar, A., Upadhyay, A., Shah, R., Pooniwala, N., &Kasambe, P. (2018). Face Recognition based Attendance Management System using Machine Learning. International Research Journal of Engineering and Technology (IRJET), 5(06), 1979-1985.

[6]. Kanti, J., & Papola, A. (2014). Smart attendance using face recognition with percentage analyzer.International Journal of Advanced Research in Computer and Communication Engineering, 3(6.2014).

[7]. Zeng, W., Meng, Q., & Li, R. (2019, March). Design of intelligent classroom attendance system based on face recognition.In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 611-615).IEEE.

[8]. Sarker, D. K., Hossain, N. I., & Jamil, I. A. (2016, December). Design and implementation of smart attendance management system using multiple step authentication. In 2016 International Workshop on Computational Intelligence (IWCI) (pp. 91-95).IEEE.

[9]. ChayaDevi, S. K., &Agnihotram, V. (2020). Smart attendance system using Convolution Neural Network and Image Processing. European Journal of Engineering Research and Science, 5(5), 611-616.

[10].Chauhan, R. K., Pandey, V., & Lokanath, M. (2018). Smart Attendance System Using CNN. International Journal of Pure and Applied Mathematics, 119(15), 675-680.

[11]. Pooja, I., Gaurav, J., Devi, C. Y., Aravindha, H. L., & Sowmya, M. (2019, February). Smart Attendance System Using Deep Learning Convolutional Neural Network.In International Conference on Remote Engineering and Virtual Instrumentation (pp. 343-356).Springer, Cham.

[12]. Sawhney, S., Kacker, K., Jain, S., Singh, S. N., & Garg, R. (2019, January). Real-Time Smart Attendance System using Face Recognition Techniques. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 522-525).IEEE.

[13]. Rathod, H., Ware, Y., Sane, S., Raulo, S., Pakhare, V., & Rizvi, I. A. (2017, January). Automated attendance system using machine learning approach. In 2017 International Conference on Nascent Technologies in Engineering (ICNTE) (pp. 1-5). IEEE.

[14]. Kar, N., Debbarma, M. K., Saha, A., & Pal, D. R. (2012). Study of implementing automated attendance system using face recognition technique. International Journal of computer and communication engineering, 1(2), 100.

[15]. Rekha, E., & Ramaprasad, P. (2017, January). An efficient automated attendance management system based on Eigen Face recognition.In 2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence (pp. 605-608).IEEE.

[16].Jha, A. (2007). Class room attendance system using facial recognition system. The International Journal of Mathematics, Science, Technology and Management, 2(3), 4-7.

[17]. Wagh, P., Thakare, R., Chaudhari, J., & Patil, S. (2015, October). Attendance system based on face recognition using eigen face and PCA algorithms. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT) (pp. 303-308).IEEE.

[18]. Krishnan, M. G., & Balaji, S. B. (2015). Implementation of Automated Attendance System using Face Recognition.International Journal of Scientific & Engineering Research, 6(3).

[19]. Lukas, S., Mitra, A. R., Desanti, R. I., &Krisnadi, D. (2016, October). Student attendance system in classroom using face recognition technique.In 2016 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 1032-1035).IEEE.

[20] Jijina, G.O., Ranganathan, V., "Low power architecture for reconfigurable fir filter for SDR",World Applied Sciences Journal, 2014, 32(2), pp. 164–168

[21] Jijina, G.O., Ranganathan, V., Kalavathy, R."Design of low power and area efficient new reconfigurable FIR filter using PSM and shift and add method", Research Journal of Applied Sciences, Engineering and Technology, 2014, 8(24), pp. 2416–2421

[22] Jijina, G.O., Ranganathan, V.;"Design of a novel reconfigurable FIR filter by Constant Shift Method Based on 4-Bit BCSE technique", International Journal of Applied Engineering Research, 2014, 9(22), pp. 12689–12697.