Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 5, No. 5, May : 2023

REALTIME ATTENDANCE SYSTEM USING OPENCV AND FACE-RECOGNITION

Hatim Batterywala, Kaustubh Pathak, Jayesh Dhakad, B.Tech Computer Science Engineering, Medi-caps University Indore.
Mrs. Hemlata Patel, Mr. Vivek Kumar Gupta, Professor, Dept. Of Computer Science Engineering, Medi-caps University Indore.

ABSTRACT
Face recognition is a crucial area of computer vision that focuses on detecting and recognizing faces in images or videos. The most challenging applications of face recognition is in live attendance systems, where accurate and fast recognition is important for error free attendance management systems.In this project, a system is proposed that leverages the python face-recognition module and open cv to perform live attendance with face recognition.The system uses a webcam or other camera to capture the faces of students and matches them with existing images stored in a database.The system's robustness and efficiency make it an ideal solution for schools, universities, and other organizations looking to automate their attendance systems.
Keywords: OpenCV, LBPH, CNN, HOG, PCA, MTCNN, SVM.

Introduction:
Real time attendance system is an extensively utilized application of image processing with numerous applications in domains such as security, biometrics, and education. Maintaining accurate and efficient attendance records of students is a significant challenge for educational institutions. Manual methods of attendance taking are prone to errors, manipulation, and time consumption. Contrary to this, biometric systems such as fingerprint or iris recognition are prone to spoofing attacks. Hence, there is a need for an automated attendance system that can recognize students by their faces and record the attendance without any human intervention.This paper presents a project that aims to develop an automated attendance system using Python modules such as OpenCV and face_recognition. The system uses a camera module to capture live video of students in a classroom and processes it using various image recognition algorithms. The algorithm observes and compares the recognized faces with a pre-stored database of student images and records their attendance on a .csv file.The system is designed to handle various challenges such as illumination, head movements, and varying lighting conditions.

Literature
This literature review delves into the application of facial recognition technology in student attendance systems. It starts by providing a brief history of the technology and the different algorithms that have been developed to enhance its performance. The review further explores the benefits of using facial recognition technology in student attendance systems, such as improving accuracy and reducing errors in manual recording processes, enhancing privacy and security, preventing fraudulent attendance, and generating regular attendance reports.
The review notes the influence of external factors, such as makeup, glasses, or changes in hairstyle, on facial recognition accuracy. There has been significant interest in using biometrics for tracking attendance in recent years. One of the approaches that has been explored is the use of fingerprints, which was discussed in a study referenced as [7]. This technique involves using a biometric sensor to capture a student's fingerprint, which is then processed using feature extraction. The database stores the student enrollment data or compared with existing data in the database for authentication purposes.In recent years, several methods have been proposed for attendance management in various settings. One such method is the use of face recognition techniques, as demonstrated in the work of

Kar [12]. This system has integrated features and facilitates the matching requests and new information is added to the database. Another approach that has been explored for attendance tracking is iris recognition, as described in papers [8] and [9]. However, this method is sensitive to environmental factors, which can affect the accuracy of the system.More recently, a real-time computer vision algorithm was implemented in an automated system that manages and records attendance, proposed by [11]. This system utilizes a non-intrusive camera which captures the images and detects faces in the classroom and compares the captured faces with those within the system using techniques of machine learning often employed in computer vision. Zacharia also proposed to develop a smart system to manage attendance by the funtcionalities of face recognition[13] that relies solely on principal component analysis (PCA).
Meanwhile, J.G. investigated an optimized method for marking attendance [14].
In order to improve the accuracy of biometric-based attendance tracking, authors in [10] proposed the idea of extending histograms into much spatial histograms that encode both the spatial relations and appearance of facial regions.
When facial recognition is discussed, there are several algorithms that have been proposed to improve accuracy and efficiency.
A literature review was conducted to compare some of these algorithms, including LBPH, CNN, and HOG (Histogram of Oriented Gradients) method.

- LBPH- is an algorithm that extracts facial features by computing histograms of binary patterns in small regions of an image. The resulting histograms are concatenated to create a feature vector of the face.[1]
- CNN-is a powerful deep learning algorithm for computer vision tasks, including face recognition, as it can learn features hierarchically from raw input data, making it adaptable to variations.[2]
- HOG-is an algorithm for feature extraction that computes gradients in an image and groups them into cells that are normalized to create histograms for forming a feature vector.[2]

OpenCV and face_recognition libraries are popular for facial recognition, using deep learning techniques to achieve high accuracy. Face detection is a crucial component, and OpenCV offers methods like the Viola-Jones algorithm [4]. Facial recognition researchhas made fascinating discoveries, such as experiments on Darwin's feelings for facial expressions.Facial recognition is compared to other biometric systems, and it is used in various applications, including security and thumb recognition systems. [5] presents a study on OpenCV and face detection technology. MTCNN is a popular face detection library that uses a multi task neural network model to generate candidate boxes and then fine-tunes classification and regression with a more complex model [6].This system would capture live video feed from a camera, process faces using these libraries to extract features and perform face recognition. The face_recognition library extracts features using face embeddings to compare similarities between faces. Many research works have utilized these libraries, and tutorials [3].
In conclusion, the use of face_recognition and OpenCV and MTCNN, accurate and efficient face detection can be achieved, making such systems more reliable and robust [6].

Face Recognition and OpenCV Algorithms

The research question was addressed through a quantitative approach, measuring the accuracy and efficiency of the system. OpenCV provides image processing and video analysis functions, while face_recognition builds on OpenCV, providing high-level functions for face detection, recognition, and manipulation.

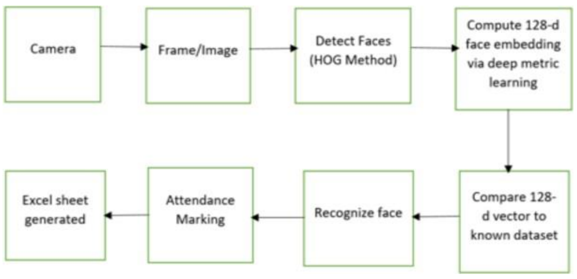
Camera
Frame/Image
Detect Faces (HOG Method)
Compute 128-d face embedding via deep metric learning
Compare 128-d vector to known dataset
Recognize face
Attendance Marking
Excel sheet generated

Figure 1: Image Processing

Data collection involved creating a dataset of 100 face images of 10 students. Each student provided 10 images of their face, stored in a "dataset" folder with subfolders named after each student's ID number.Image classification uses feature descriptors like HOG to represent an image as a point in higher dimensional space. An algorithm like SVM is implied to partition the space into hyperplanes to separate points representing different classes.

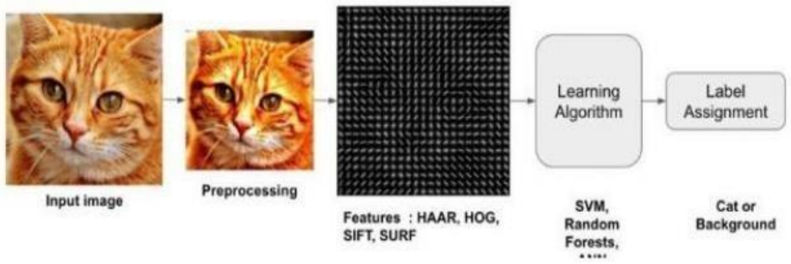

Figure 2: Algorithm Implementation

Although Deep Learning may seem very different from the traditional approach, there are actually some conceptual similarities between the two. For example, Dlib's Face Recognition module, shown in Figure 4, is based on a CNN architecture called ResNet.

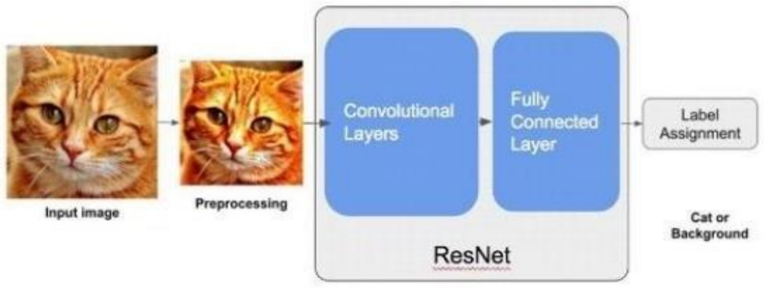

Figure 3: Resnet Implementation

ResNet uses a bank of learned convolutional layers and a fully connected layer to produce a feature vector for image classification. This method differs from HOG as ResNet is adaptable to the problem at hand and the fully connected layer performs the same job as the SVM classifier. The distance between two points in ResNet is typically calculated using the Euclidean distance.

Face Recognition and OpenCV Libraries and Modules

Import necessary libraries and modules:This step involves importing all the libraries and modules required to execute the code. In this code, you have to make sure that you import the following libraries: cv2, numpy, face_recognition, os, and datetime using pip.

Cv2: provides a set of functions and tools that allow users to read, manipulate, and process images and videos, as well as perform various computer vision tasks such as object detection, face

recognition, and image segmentation. It also provides support for various image file formats and video codecs.
In face recognition, NumPy is often used to represent images as arrays of numbers, with each number representing the brightness or color of a pixel. NumPy allows for efficient manipulation and processing of these arrays, enabling tasks such as feature extraction, normalization, and classification.
The face_recognition module in Python uses deep learning algorithms to perform facial recognition and detection tasks by encoding facial features of each person in a particular database, it then compares these features with the features of the faces detected in a live video stream. It has imortant utilities necessary for attendance management, access control, and security systems. The system marks attendance of recognized individuals by comparing the faces detected in a video stream with a database of known faces.

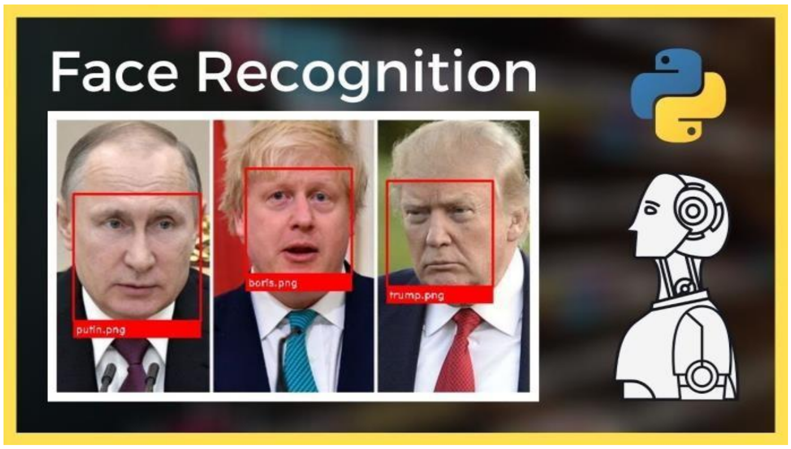

Image 1: Face Recognition

Procedure of Face Recognition and Attendance Recording

- Prepare the dataset: The code reads each image from a directory, generates face encodings using face_recognition.face_encodings() function, and appends them to a list. It also extracts the name of each file and associates it with a person's name using os.path.splitext() function. The resulting list of names is then printed to the console.
- Encode the faces: “The faceEncodings() function in this code uses a face_recognition library to encode faces in the images, while the start_attendance() function uses the face_recognition library to detect and recognize faces in the live video feed from a webcam.” Video is captured using cv2.VideoCapture() function and creates an infinite loop to continuously read frames from the video. The frames are then processed to detect faces using the face_recognition library, and if a match is found with the known face encodings, an indivisuals name is identified and recorded for attendance purposes.
- Detect faces and compare with encodings: In this step of the code, the face locations in each frame of the captured video are detected using the face_recognition.face_locations() function. Then, the face_recognition.face_encodings() function is used to encode the detected faces, which are then compared with the known face encodings using the face_recognition.compare_faces() function. The face_recognition.face_distance() function measures the distance between encodings for identifying the best match.
- Identify an indivisual: In this step, you need to identify the person in the current frame by comparing the encodings of the detected face with the encodings of the faces in the dataset. np.argmin() is an important function to find the index of the face with the smallest distance. Once you have the index, you can retrieve an indivisual’s name from the "personNames" list.

- Display the result: In this step, you need to display the result on the video frame. cv2.rectangle() is a function that draws a rectangle around the detected face. You can also use the cv2.putText() function to display that indivisual's name on the video frame.“cv2.rectangle() is a function provided by the OpenCV library used to draw a rectangle on an image.cv2.putText() is a function provided by the OpenCV library used to put text on an image”.
- Record the attendance: In this step, the attendance of the identified person is successfully recorded. You can use the attendance() function defined in the code to write the name, date, and time of the person in a CSV file called "Attendance.csv".
- Terminate the program: To terminate the program on "Enter" key press using cv2.waitKey(). The program waits for 1 millisecond using cv2.waitKey(1) for a key event. If the Enter key (ASCII 13) is pressed and as the loop exits the program finally terminates. Otherwise, the video frames are continuesly processed from the webcam. The project uses face recognition and computer vision to detect faces in a video, compares them with a dataset, identifies the person, and records their attendance in a CSV file.

Results
From this paper, we present the implementation and results of a face recognition-based attendance system. The project leverages OpenCV, face_recognition, and tkinter libraries to create a user-friendly application capable of recording attendance through a video stream that’s captured live.

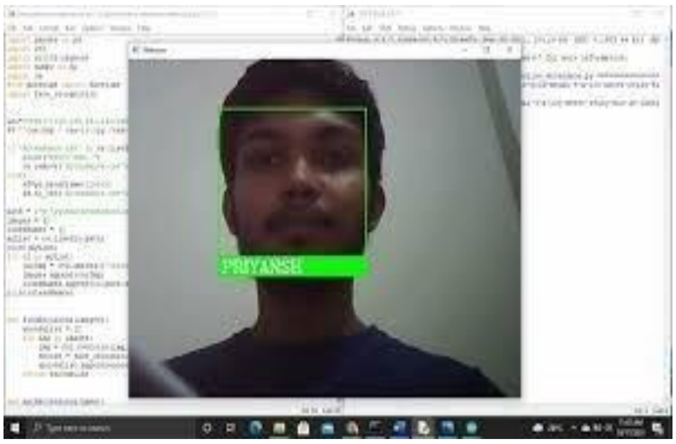

Image 2: Working Snippet

The experimental results show that the attendance system is capable of accurately recognizing individuals and logging their attendance in real-time. The use of the face_recognition library, along with OpenCV, enables the system to process video frames efficiently and detect faces accurately. The tkinter-based interface offers a user-friendly way to control the application.

Conclusion
In conclusion, the use of OpenCV and dlib in combination provides the best approach for extracting frames and recognizing multiple faces from a single image or video stream. The implementation of this method has demonstrated higher accuracy in recognition of faces with a significant reduction in response time. This combination of tools offers a robust solution for applications that require real-time or near-real-time face recognition .Overall, the use of OpenCV and dlib in tandem offers a highly effective and efficient solution for face recognition that can meet the demands of a broad spectrum of and has various applications complimenting it’s use.

References:

[1] S L Suma, Sarika Raga. “Real Time Face Recognition of Human Facesby using LBPH and Viola Jones Algorithm.” International Journal of Scientific Research in Computer Science and Engineering ,Vol.6, Issue.5, pp.01- 03, Oct. 2018.

[2] H. Ahamed, I. Alam and M. M. Islam, "HOG-CNN Based Real Time Face Recognition," 2018 International Conference on Advancement in Electrical and Electronic Engineering (ICAEEE), Gazipur, Bangladesh, 2018, pp. 1-4, doi: 10.1109/ICAEEE.2018.8642989.

[3] Facial Recognition Attendance System Using Python and OpenCv https://www.questjournals.org/jses/papers/Vol5-issue- 2/D05021829.pdf

[4] Parhi, Manoranjan, et al. "Ioats: an intelligent online attendance tracking system based on facial recognition and edge computing." International Journal of Intelligent Systems and Applications in Engineering 10.2 (2022): 252-259.

[5] M. Khan, S. Chakraborty, R. Astya and S. Khepra, "Face Detection and Recognition Using OpenCV," 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2019, pp. 116-119, doi: 10.1109/ICCCIS48478.2019.8974493.6.Research on Face Detection Technology Based on MTCNN | ieeexplore.ieee.org

[6] Dong Yachao, Bao Jun, Liu Hongzhe. Research progress of small target detection technology based on deep learning [C]. Network application branch of China Computer Users Association. Proceedings of the 23rd annual meeting of new network technology and application in 2019 of network application branch of China Computer Users Association. Network application branch of China Computer Users Association: Key Laboratory of information service engineering, Beijing United University, 2019: 172-176.

[7] O.Shoewu, PhD, O.A. Idowu, B.Sc., “Development of Attendance Management System using Biometrics.”, The Pacific Journal of Science and Technology(2012).

[8] A. Khatun, A. K. M. F. Haque, S. Ahmed and M. M. Rahman, ”Design and implementation of iris recognition based attendance management system”, 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, 2015, pp.1-6.

[9] Kadry, Seifedine Smaili, Mohamad. (2010). ”Wireless attendance management system based on iris recognition”. Scientific Research and Essays. 5. 1428-1435.

[10] Ahonen, Timo, Abdenour Hadid, and Matti Pietikainen. "Face description with local binary patterns: Application to face recognition." IEEE transactions on pattern analysis and machine intelligence 28.12 (2006): 2037-2041.

[11] K. S. Kumar, S. Prasad, V. BhaskarSemwal, and R. C. Tripathi, “Real time face recognition using adaboost improved fast pca algorithm,” Int. J. Artif. Intell.Appl, p. 45–58, 2011.

[12] N. Kar and D. M. K. D. Barma, “Study of implementing automated at-tendance system using face recognition technique,” International Journal of Computer and Communication Engineering, 2012.

[13] J. Joseph and K. Zacharia, “Automated attendance management system using face recognition,” International Journal of Science and research, vol. 2, 2013.

[14] R. Tharanga, Samarakoon, Karunarathne, Liyanage, and D. Parer, “Smart Attendance using real time face recognition (smart-fr),” Semantic Scholar, 2013.