
ATTENDANCE SYSTEM BY FACE RECOGNITION USING DEEP LEARNING

Prof. C.D. Sawarkar^{*1}, Mr. Sanket. S. Alane^{*2}

^{*1}Professor, Computer Science And Engineering, Shri Shankarprasad Agnihotri College Of Engineering Ramnagar, Wardha, Maharashtra, India.

^{*2}Students, Computer Science And Engineering, Shri Shankarprasad Agnihotri College Of Engineering Ramnagar, Wardha, Maharashtra, India.

DOI : <https://www.doi.org/10.56726/IRJMETS48994>

ABSTRACT

In colleges, universities, organizations, schools, and offices, taking attendance is one of the most important tasks that must be done on a daily basis. The majority of the time, it is done manually, such as by calling by name or by roll number. The main goal of this project is to create a Face Recognition-based attendance system that will turn this manual process into an automated one. This project meets the requirements for bringing modernization to the way attendance is handled, as well as the criteria for time management. This device is installed in the classroom, where and student's information, such as name, roll number, class, sec, and photographs, is trained. The images are extracted using Open CV. Before the start of the corresponding class, the student can approach the machine, which will begin taking pictures and comparing them to the qualified dataset. Logitech C270 web camera and NVIDIA Jetson Nano Developer kit were used in this project as the camera and processing board. The image is processed as follows: first, faces are identified using a Haarcascade classifier, then faces are recognized using the LBPH (Local Binary Pattern Histogram) Algorithm, histogram data is checked against an established dataset, and the device automatically labels attendance. An Excel sheet is developed, and it is updated every hour with the information from the respective class instructor. Due to the fact that CNNs achieve the best results for larger datasets, which is not the case in production environment, the main challenge was applying these methods on smaller datasets. A new approach for image augmentation for face recognition tasks is proposed. The overall accuracy was 95.02% on a small dataset of the original face images of employees in the real-time environment. The proposed face recognition model could be integrated in another system with or without some minor alternations as a supporting or a main component for monitoring purposes.

Keywords: Face Detection, Face Recognition, Haar Cascade Classifier, NVIDIA Jetson Nano.

I. INTRODUCTION

Attendance system is a crucial task in any organization, including educational institutions, businesses, and government agencies. Traditional attendance system systems, such as paper-based systems, are often time-consuming and error-prone, leading to inaccurate attendance records and a loss of productivity. With the advent of deep learning techniques, there has been a growing interest in developing automated attendance system systems that can provide accurate and efficient attendance records. In recent years, face recognition-based attendance systems have emerged as a promising solution to the challenges of attendance system. These systems leverage the power of deep learning algorithms to extract features from facial images, thereby achieving higher accuracy and robustness in recognizing faces. Moreover, face recognition-based attendance systems eliminate the need for manual input, which saves time and reduces errors. In this study, we suggest a deep learning based facial recognition attendance system built on the Dlib framework and HOG. The proposed system where facial features are extracted using the HOG algorithm, and facial recognition is performed using a deep learning-based neural network and of total six phases namely Capturing student image, Face Detection , Face Alignments, Feature Extraction, Face Recognition and providing attendance on .CSV file. The system has been designed to be efficient and accurate, even when there are differences in lighting, position, and attitude. The suggested system has the power to transform attendance system across a range of industries, including education, healthcare, and security. The system can reduce the work load of teachers and administrators and ensure accurate attendance records, leading to improved productivity and accountability. Moreover, the system can be deployed in a variety of settings, from small classrooms to large organizations, making it a versatile and scalable solution. The overview of previous approaches to face recognition and attendance control is presented

in the next section. The next section provides a description of the suggested approach, along with the stages involved in data collection and preprocessing, and the design of the deep learning model that will be utilised for face recognition.

II. LITERATURE SURVEY

Jyoti D et al. [1] highlights the effectiveness of their proposed attendance system using a deep neural network. They conclude that their system achieves an accuracy rate of 96.5% on a dataset of 500 images, which is significantly higher than traditional attendance systems. The authors also note that their system can be easily integrated with existing school or college management systems, making it an efficient and practical solution for attendance management. The three main steps in the authors proposed deep learning-based attendance system are face detection, face alignment, and face identification. For face detection, the authors use the Haar Cascade classifier, while for face alignment, they use the facial landmark detection algorithm. For face recognition, they use the VGG-16 deep neural network. The authors use a dataset of 500 images collected from 50 students to train and test their system. The authors report an accuracy rate of 96.5% for their system on the dataset of 500 images. The authors further suggest that future research can focus on improving the accuracy of deep learning-based attendance systems, exploring the potential of combining multiple deep learning techniques, and investigating the system's performance on larger datasets. A comprehensive review of the existing deep learning techniques for face recognition is done by Siqi Deng et al. [2] The authors arrive to the conclusion that deep learning-based face recognition is now the most advanced technique in terms of accuracy and robustness. The authors also note that Convolutional Neural Networks (CNNs) have emerged as the most effective deep learning technique for face recognition, primarily due to their ability to learn hierarchical representations of facial features. In addition, the authors discuss various CNN architecture that have been used for face recognition, such as VGG-Net, ResNet, and Inception. The authors further highlight the importance of large-scale datasets for training deep learning models and discuss the availability of publicly available datasets for face recognition, such as LFW, YTF, and IJB-A. According to Xiaoyuan Jing [3] the following algorithms were used CNN for feature extraction and it is trained on a large dataset of facial images to learn the most important features of the facial images, KNN algorithm used for face recognition by comparing the features of the test image with those of the enrolled students in the database. The system utilized a similarity score to determine if the test image matched any of the enrolled students. The system achieved an accuracy rate of 94.4% in recognizing enrolled students and 90.4% in rejecting non-enrolled individuals. These results were obtained by testing the system on a dataset of 200 images consisting of 100 enrolled students and 100 non-enrolled individuals. The system was also tested on a real-time attendance scenario with 30 enrolled students and achieved an accuracy rate of 93.3%. B. Kavinmathi et al. [4] proposed an automated system based on CNNs. Here the GSM module is used to send the generated attendance report. The two normalisation processes are added to two of the layers to produce the enhanced convolutional neural network that the author proposes. This approach accelerates batch normalisation on the network. The system was developed using the SIFT technique and attendance will be taken by using MATLAB. An SMS will be sent to the specified number after the image has been taken and compared to the database. The primary procedures used here is producing the scale-space extreme detection, key point localisation, orientation assignment, and key point descriptor are features. The Arduino board's LED will start to blink when the system recognises a face. Shubhobrata Bhattacharya et al. [5] highlights that in order to obtain features with minimal dimensions the author employed a convolutional neural network because the pre-processed images are too high dimensional to be used as direct input by a classifier. They tracked the face from frame to frame using the correlation tracker after using the Viola and Jones method for face detection. The author of this study has concentrated on a number of characteristics, including posture estimation, sharpness, resolution, and brightness. The three angle roll, yaw, and pitch are used to compute the head position. The following method entails giving weights to each of the normalised criteria in order to generate a final score for the evaluation of face quality. In Mayank Yadav et al. [6] The authors suggest a motion-based attendance system that automatically tracks and records attendance in real-time using computer vision and machine learning approaches. The suggested method employs a camera to record live footage of a conference room. In order to identify people in the video and detect human motion, computer vision algorithms are used to process the footage in real-time. After detecting individuals, the system utilises machine learning algorithms to identify them and compare them to a database of registered attendance.

The attendance data is then saved and accessible to authorised staff via a web based interface. The paper by Yanhua Yang et al.[7] provides a comprehensive survey of deep learning-based face recognition techniques. Deep CNNs, recurrent neural networks(RNN), and other cutting-edge techniques for face recognition using deep learning have all been studied and analysed by the authors (RNNs), auto encoders and deep belief networks (DBNs).The paper highlights the benefits and limitations of these techniques and provides insights into their applications and research directions. The authors have also discussed the challenges associated with deep learning-based face recognition, including the need for large amounts of labeled data, computational complexity, and generalization performance. Khan et al.[8] highlights the Face API and OpenCV are combined to create a face-recognition face-based real-time automatic attendance system. The system was put to the test in a classroom setting, where it recognised and recorded student attendance with success. Because it does away with the necessity for manual attendance taking and lowers the possibility of mistakes, the authors emphasise the significance of such a system in enhancing attendance management in educational institutions. Teachers and administrators can also receive real-time attendance statistics from the system, enabling them to make prompt decisions based on attendance data. The proposed approach, according to the paper's conclusion, is a practical and cost-effective method for managing attendance in educational institutions and has the potential to be used in other contexts, such as workplaces and public areas.

III. METHODOLOGY

The FRBA System is implemented in six phases:

Phase -1:Capturing Image Of Student:

Phase-2: Face Detection: Locates the face, draws a bounding box around it and records the bounding box coordinates.

Phase-3: Face Alignments: Normalize the face to be consistent with the training image.

Phase-4: Feature Extraction: Extract features of face those will be used for training and recognition tasks.

Phase-5: Face Recognition: Matching of the face to pictures in the database that has been prepared.

Phase-6: Providing Attendance By .CSV File:After recognizing the face of the student then it enter the details of the student in the .CSV file.

Phase -1: Capturing Image Real-time facial images can be recorded using a camera. The camera can either be a separate device or integrated into a gadget like a smartphone or tablet.

Phase-2: Face Detection In a face recognition attendance system that uses Dlib, HOG, and does without SVM, the face detection phase entails pre-processing the image, extracting features using HOG, spotting potential face regions using a sliding window technique, classifying each region as a face or not a face using a binary classifier, removing overlapping face regions using non-maximum suppression, and drawing a bounding box around the detected face region. Image Pre-processing: In order to make the input image more efficient to process, it is first pre-processed by turning it into a grayscale image. Feature Extraction: The image is then put through the Histogram of Oriented Gradients feature extraction technique (HOG). HOG calculates the gradient orientation of each pixel by locating its edges and corners in the image. Steps To Calculate HOG Features: 1.HOG features should be calculated for the supplied image. The picture should be resized to 128 by 64 pixels (128 pixels height and 64 width). 2.The gradient of the image is calculated and it is produced by adding the image's magnitude and angle. In a block of 3x3 pixels, G_x and G_y are first calculated for each pixel. Formula used for calculating for each pixel value. where r stand for rows and c stand for columns.

The following formulas are used to determine the magnitude and angle of each pixel once G_x and G_y are determined. I. The gradient matrices (magnitude and angle matrix) are grouped into 8x8 cells to construct a block once the gradient of each pixel is determined. A 9-point histogram is computed for each block. A 9-point histogram produces a histogram with 9 bins, each with a 20- degree angle range. The graphic representation of each of these nine-point histograms is a histogram with bins that each output the gradient's intensity. Since a block might have up to 64 different values, the following calculation is performed for each of the 64 magnitude and gradient values. since we are utilising nine- point histograms. Boundaries for each j th bin, bin will come from: The value of each bin's centre will be: an image of a 9-bin histogram. For one 8x8 block with 64 cells, this particular histogram will be the only one of its sort. The j th and $(j+1)$ th positions of the array will be added to

by each of the 64 cells together with their respective V_j and V_{j+1} values. 4. Before deciding on the value that will be assigned to the j th and $(j+1)$ th bins, respectively, we will first find the j th bin for each cell in a block. The values of V_j and V_{j+1} are added to an array at the index of the j th and $(j+1)$ th bins calculated for each pixel. An array is used as a bin for a block. 5. Using the computations above, a $16 \times 8 \times 9$ matrix will be the outcome. 6. When the histogram computation is finished for each block, 4 blocks from the 9 point histogram matrix are joined to make a new block (2×2). This clubbing is done in an overlapping manner with an 8-pixel stride. To produce a 36 feature vector, we integrate the nine point histograms for each of the four cells in a block. In the graphic above, the computation process for 9 bin histograms is shown. 7. The fb values for each block are standardised using the L2norm. So, to avoid a division error by zero, a small number is ϵ added to the square of fb. The number entered into the code is $1e-05$. 8. The following formulas are used to calculate the value of k before normalization. 9. The impact of variations in contrast between photographs of the same object is decreased by normalising the photos. from each brick. A 36-point feature vector is created. 15 blocks are present vertically and 7 blocks are present horizontally. Thus, the HOG characteristics will be: $7 \times 15 \times 36 = 3780$. Sliding Window: In order to look for potential face regions in the image, multi-scale sliding window technique is used. With this method, a window that is a fixed size is moved across the image at various scales and places. To extract HOG characteristics, window positions and scales are examined. The multi-scale sliding window approach uses the sliding window to recognise faces of various sizes at various scales. To do this, the input image is resized to various scales and the sliding window approach is applied to each resized image. Detection: A binary classifier like Logistic Regression is then used to categorise each window as either a face or not using the retrieved attributes. Using a dataset of labelled faces and non-faces, the classifier is trained. Non-maximum Suppression: A non-maximum suppression approach is applied to the detected face regions in order to prevent the detection of many overlapping faces. The algorithm preserves the face region with the highest confidence score and discards the overlapping face regions. Bounding Box: The recognised face region is then enclosed by a bounding box, and its coordinates are noted.

Phase-3: Face Alignments To guarantee consistency and correctness of face photos across many samples, normalisation techniques are required for face alignments in face recognition systems. In order to correctly match faces, face recognition algorithms rely on recognising key facial landmarks and characteristics, such as the eyes, nose, and mouth. These facial features might seem differently depending on the stance, lighting, and emotion used. In addition to normalising the brightness, contrast, and colour of the image, this normalisation procedure may also entail changing the position, size, and orientation of the facial image.

Phase-4: Feature Extraction It entails removing essential and distinctive information from the facial photos that can be matched and identified. In order to extract information from certain areas or patches of the face image, local feature descriptors like the HOG are used in face recognition algorithms. Dlib is a well-liked open-source Python toolkit for feature extraction, face identification, and landmark detection. The pretrained HOG-based face detector it offers can identify faces in pictures and extract facial landmarks like the eyes, nose, and mouth. Dlib can employ HOG-based feature descriptors, which store the gradient's orientation and magnitudes of the image pixels inside particular blocks or regions, to extract features from the aligned face picture. The feature vectors that are produced can be utilised to match and identify faces. HOG-based feature descriptors are highly suited for face recognition applications since they are made to capture local texture and shape information that is invariant to variations in illumination and position. However, the effectiveness of HOG-based feature descriptors may be influenced by the accuracy of the face normalisation and alignment, as well as the size and complexity of training and testing datasets.

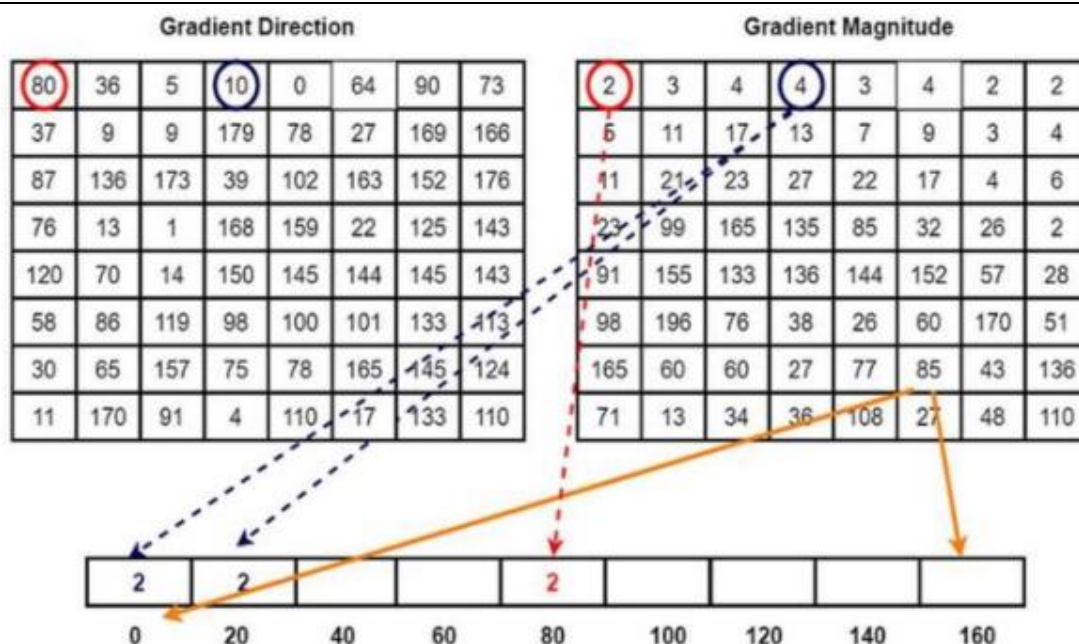


Fig 1: Histogram of Gradients

Phase-5: Face Recognition In face recognition systems, the matching stage is comparing the traits retrieved from the input face image with those in a database of reference photos to ascertain the person's identification. HOG and Dlib are helpful at this step because of the following. I. Feature Representation: The underlying patterns and structures in the face, such as the texture and contour of the face, can be captured through discriminative features that can be extracted from the face pictures using HOG and Dlib. The face photos can be represented using these attributes in a concise and informative manner that is appropriate for matching. II. Similarity Measurement: After the features have been retrieved, HOG and Dlib can be used to assess how closely the input face image resembles the database's reference images. This is commonly accomplished by measuring the distance or similarity between the feature vectors using a distance metric, such as the Euclidean distance or cosine similarity. III. Classification: A classification model, such as a support vector machine (SVM) or a neural network, can be trained using HOG and Dlib to divide the face photos into many categories depending on their attributes. This could be helpful for locating and classifying various database users.

Phase-6: Providing attendance on .CSV file After recognizing the face of the student the system will automatically enter the details of the student in the .CSV file and provides attendance.

IV. MODELING AND ANALYSIS

Face detection involves separating image windows into two classes; one containing faces (turning the background (clutter). It is difficult because although commonalities exist between faces, they can vary considerably in terms of age, skin color and facial expression. The problem is further complicated by differing lighting conditions, image qualities and geometries, as well as the possibility of partial occlusion and disguise. An ideal face detector would therefore be able to detect the presence of any face under any set of lighting conditions, upon any background. The face detection task can be broken down into two steps. The first step is a classification task that takes some arbitrary image as input and outputs a binary value of yes or no, indicating whether there are any faces present in the image. The second step is the face localization task that aims to take an image as input and output the location of any face or faces within that image as some bounding box with (x, y, width, height). After taking the picture the system will compare the equality of the pictures in its database and give the most related result. We will use NVIDIA Jetson Nano Developer kit, Logitech C270 HD Webcam, open CV platform and will do the coding in python language. The main components used in the implementation approach are open source computer vision library (OpenCV). One of OpenCV's goals is to provide a simple to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. OpenCV library contains over 500 functions that span many areas in vision. The primary technology behind Face recognition is OpenCV. The user stands in front of the camera keeping a minimum distance of 50cm and

his image is taken as an input. The frontal face is extracted from the image then converted to gray scale and stored. The Principal component Analysis (PCA) algorithm is performed on the images and the eigen values are

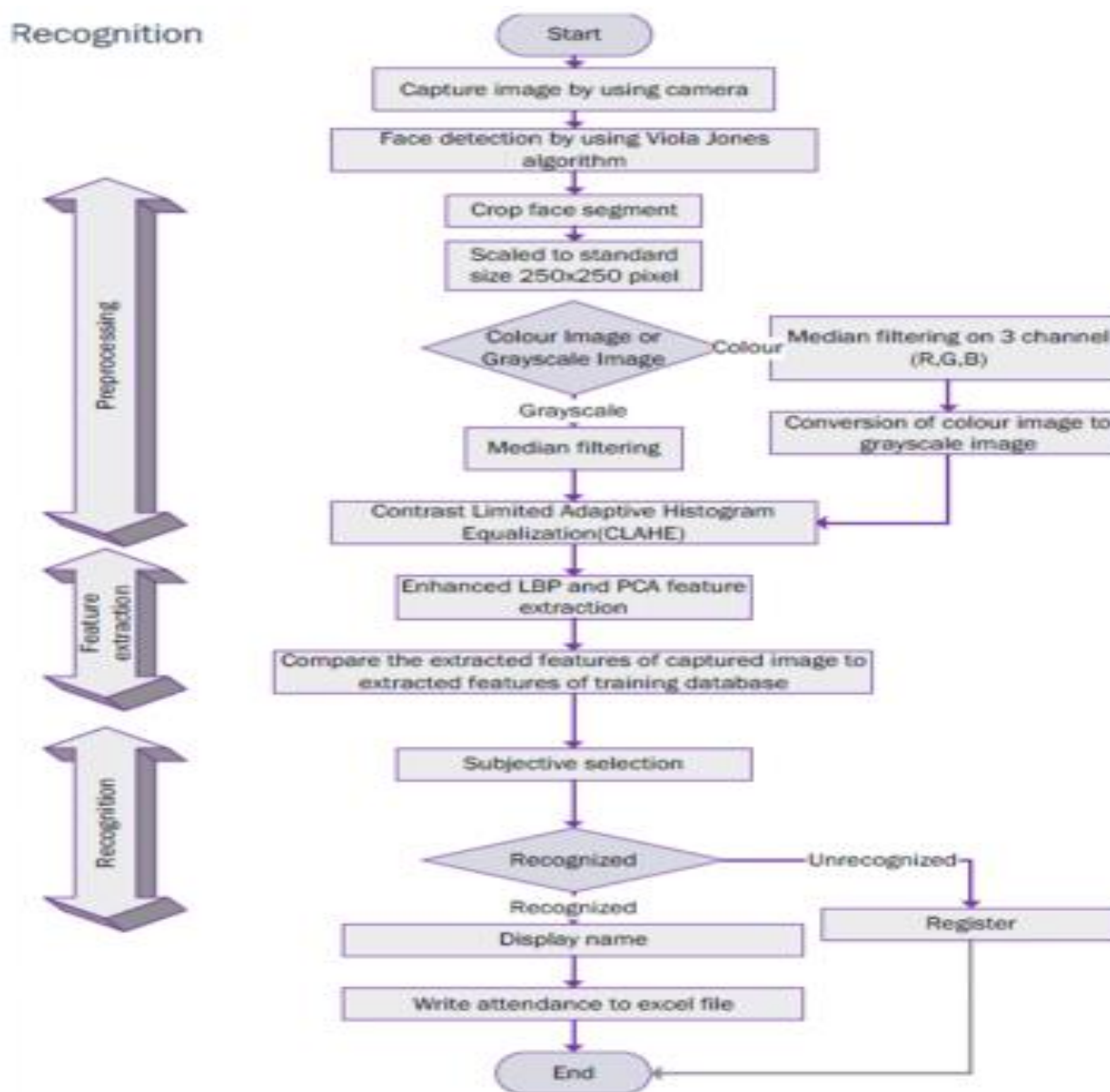


Fig 2. Flow diagram

Stored in an xml file. When a user requests for recognition the frontal face is extracted from the captured video frame through the camera. The eigen value is re-calculated for the test face and it is matched with the stored data for the closest neighbor

Software Implementation:

1. OpenCV: We used OpenCV 3 dependency for python 3. OpenCV is library where there are lots of image processing functions are available. This is very useful library for image processing. Even one can get expected outcome without writing a single code. The library is cross-platform and free for use under the open-source BSD license. Example of some supported functions are given below:

- Derivation: Gradient/Laplacian computing, contours delimitation
- Hough transforms: lines, segments, circles, and geometrical shapes detection
- Histograms: computing, equalization, and object localization with back projection algorithm
- Segmentation: thresholding, distance transform, foreground/background detection, watershed segmentation
- Filtering: linear and nonlinear filters, morphological operations
- Cascade detectors: detection of face, eye, car plates

- Interest points: detection and matching
- Video processing: optical flow, background subtraction, camshaft (object tracking)
- Photography: panoramas realization, high definition imaging (HDR), image inpainting So it was very important to install OpenCV. But installing OpenCV 3 is a complex process. How we did it is given below:

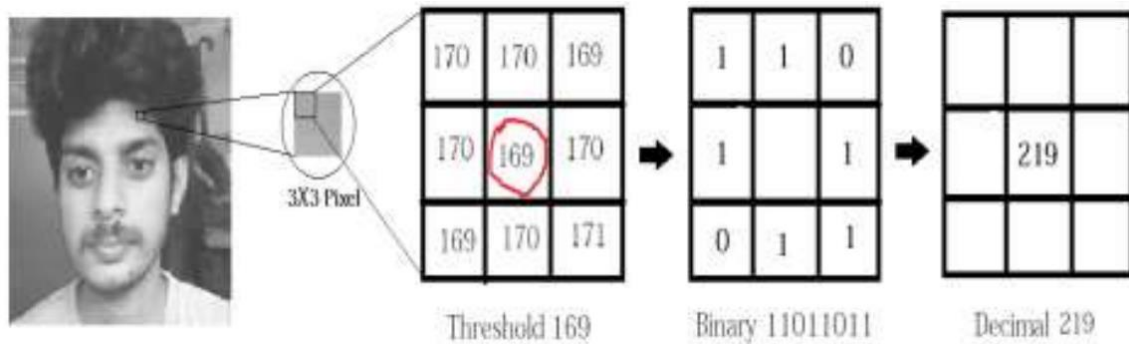


Fig 3. Photography

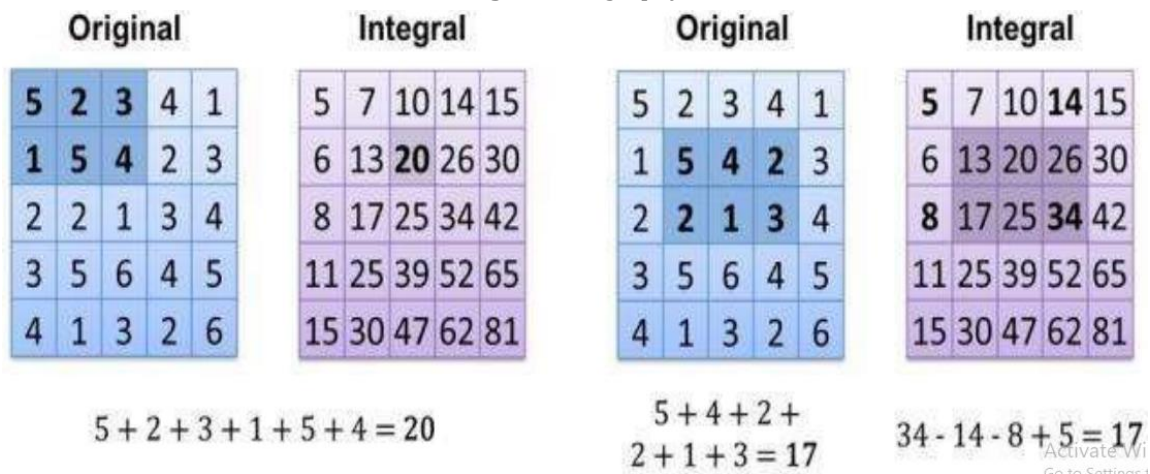
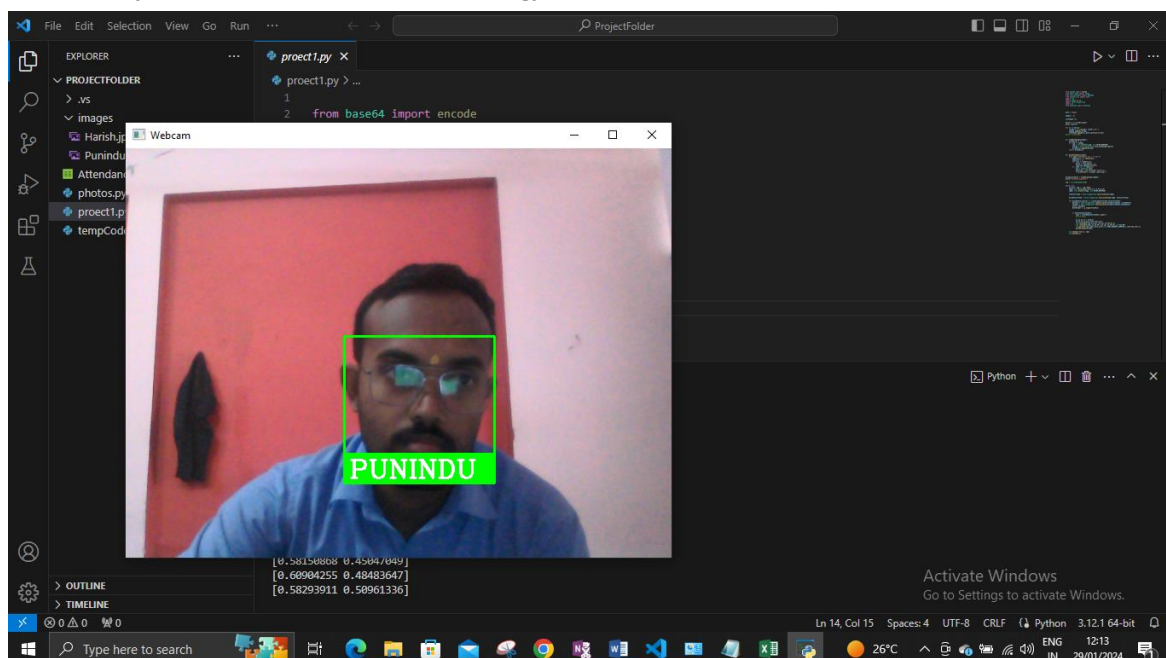


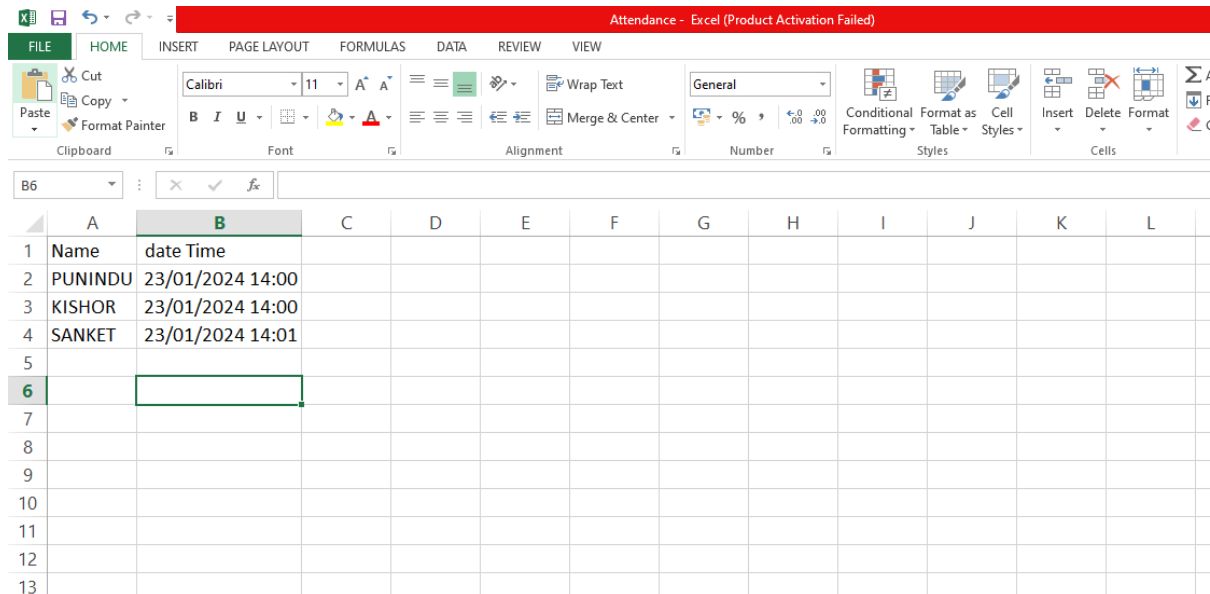
Fig 4. Histograms: computing

V. RESULTS AND DISCUSSION

The results of the experiments conducted for this project might be shown. The planned technique is demonstrated by the results, and the entire strategy is seen as an attachment to the successful result.



OUTPUT If a student's image matches one of the photos in the directory, their name will display on the image and information about them will be added to the Attendance. csvfile



	A	B	C	D	E	F	G	H	I	J	K	L
1	Name	date Time										
2	PUNINDU	23/01/2024 14:00										
3	KISHOR	23/01/2024 14:00										
4	SANKET	23/01/2024 14:01										
5												
6												
7												
8												
9												
10												
11												
12												
13												

VI. CONCLUSION

Nowadays, various attendance and monitoring tools are used in practice in industry. Regardless the fact that these solutions are mostly automatic, they are still prone to errors. In this paper, a new deep learning based face recognition attendance system is proposed. The entire procedure of developing a face recognition component by combining state-of-the-art methods and advances in deep learning is described. It is determined that with the smaller number of face images along with the proposed method of augmentation high accuracy can be achieved, 95.02% in overall. These results are enabling further research for the purpose of obtaining even higher accuracy on smaller datasets, which is crucial for making this solution production-ready. The future work could involve exploring new augmentation processes and exploiting newly gathered images in runtime for automatic retraining of the embedding CNN. One of the unexplored areas of this research is the analysis of additional solutions for classifying face embedding vectors. Developing a specialized classifying solution for this task could potentially lead to achieving higher accuracy on a smaller dataset. This deep learning based solution does not depend on GPU in runtime. Thus, it could be applicable in many other systems as a main or a side component that could run on a cheaper and low-capacity hardware, even as a general-purpose Internet of things (IoT) device.

VII. REFERENCES

- [1] "Face Recognition Based Attendance System using Deep Learning," by Jyoti D. Thorat and Vijay S. Gulhane, International Journal of Computer Applications, Vol. 180, No. 25, pp. 34-39, 2018.
- [2] "A Survey of Deep Learning Techniques for Face Recognition," by Siqi Deng and Yongdong Zhang, Cognitive Computation, Vol. 9, No. 4, pp. 561-575, 2017.
- [3] "Automatic Attendance System using Face Recognition based on Deep Learning," by Xiaoyuan Jing, International Conference on Information and Computer Technologies, pp. 112-116, 2019.
- [4] B. Kavinmathi, S. Hemalatha, "Attendance System for Face Recognition using GSM module", 4th International Conference on Signal Processing and Integrated Networks, 2018.
- [5] Shubhobrata Bhattacharya, Gowtham Sandeep Nainala, Prosenjit Das, Aurobinda Routray "Smart Attendance Monitoring System (SAMS): A Face Recognition based Attendance System for Classroom Environment", IEEE 18th International Conference on Advanced Learning Technologies, 2018.
- [6] Mayank Yadav, Anmol Aggarwal, "Motion based attendance system in real time environment for multimedia application", 2018.
- [7] "Face Recognition using Deep Learning: A survey," by Yanhua Yang, Zhaohui Wu, and Yuexian Hou, Journal of Computer Science and Technology, Vol. 32, No. 4, pp. 739-764, 2017.