Employee Attendance Management System Using OpenCV

A, Siva Narayana
Department of Electronics and Communication Engineering
Madanapalle Institute of Technology & Science
Kadiri Road, Madanapalle 517325
Chithoor Dist,Andhrapradesh

T. Ramanjineyulu
Department of Electronics and Communication Engineering
Madanapalle Institute of Technology & Science
Kadiri Road, Madanapalle 517325
Chithoor Dist,Andhrapradesh

K. Raj Kumar
Department of Electronics and Communication Engineering
Madanapalle Institute of Technology & Science
Kadiri Road, Madanapalle 517325
Chithoor Dist,Andhrapradesh

Abstract**—** It's proven difficult to track attendance using traditional ways. In the field of face recognition, the growing demand for efficient and automatic ways for registering attendance is posing a significant difficulty. In recent years, common biometrics like as fingerprints and Radio Frequency Identification tags have been routinely used to overcome the problem of automatic attendance marking.

However, these methods lack the element of trustworthiness. Face detection and identification algorithms are used in this suggested project to create an automated attendance marking and management system. Instead, then using traditional methods, the suggested solution intends to create an automated system that uses facial recognition technology to track employee's attendance. The major goal of this project is to make the attendance tracking and management system more efficient, timesaving, simple, and straightforward. Face recognition techniques will be used here to recognize faces. The processed image will then be compared to the existing stored record, and attendance will be recorded accordingly in the database. This technique minimizes people's burden when compared to the current system's traditional attendance marking system. This suggested system will be executed in four phases: image capture, group image segmentation and face detection, face comparison and recognition, and attendance updating in an excel sheet.

Introduction :

The twenty-first century is a successful modern and scientific period. Computer technology has now become an integral component of daily life. In this fast moving and modernized World, the usage of manpower has been reduced by digitalized systems. In day to day needs the human is changing in to machine based digital systems. These changes have been making simplify our tasks and reduce our mistakes. Every company wants a reliable and consistent method for keeping track of its pupils' attendance. Each organisation has its own method of doing so; some use a sheet of paper to take attendance by calling their names during work hours, while others use biometrics such as fingerprints, RFID card readers, and Iris systems to mark attendance. The traditional practise of manually calling pupils' names is a time-consuming event. Each person is assigned a card with their associated identity under the RFID card system, but there is a risk of card loss or unauthorised use of the card for fraudulent attendance. Other biometrics, such as fingerprint, iris, and voice recognition, also have limitations and are not 100% accurate. The use of facial recognition for attendance marking is a sensible approach of implementing an attendance management system. Face recognition, among other techniques, is more precise and faster, and it decreases the risk of proxy attendance. Face recognition allows for passive identification, which means that the individual being recognised does not have to take any action in order to be identified. Face detection and identification methods have been introduced in a variety of ways. Face recognition can be done in two ways: appearance-based (which covers the entire face) or feature-based (which covers geometric characteristics such as the eyes, nose, eyebrows, and cheeks). We are using the face detection and face recognition methods to make the attendance making system. The process of detecting and locating a single or series of pictures and identifying them is known as facial recognition. As human beings, we do very well in identifying faces; Computers, on the other hand, have a hard time recognising faces. Face recognition has a variety of applications, including video surveillance systems, security, access control, law, general identity verification, gender recognition, missing person identification, etc. Facial 2 recognition can be divided into two categories, namely, facial recognition and facial detection. Face detection refers to the process by which the system identifies human faces in the images and video streaming, whereas Face Recognition process refers to identifying a known face which is already trained to the system. As we know marking attendance is very important aspect in any type of organization. Maintenance of employees' attendance is much more difficult task for the Organization. Every organization has its own approach to take attendance such as using attendance sheet or by using some biometric methods. But these methods are time consuming. Mostly employees attendance is taken with the help of RFID tags and fingerprint methods. This requires a lot of time and work. Calculation of consolidated attendance is another major task that can result in manual errors. To overcome such problems, we are developing automated attendance management system. The main objective of our project is to provide a attendance management system by which we can able reduce time and cost. Here we are trying to develop a automated attendance management system using OpenCV that helps organizations to take attendance with more effective and efficient way. The advantage over the previous manual system to increase the efficiency. It is to create a real-time automatic facial detection system utilising a normal PC camera. This project is to provide the better security application. It is very important to review the work of the researchers done previously to deduce out the problem statement and solution for it. This leads to the development of device which overcomes the problems posed by the earlier ones. Automated attendance management system has be proposed to reduce the time and cost of manual attendance taking system, but the limitations are that it can works better in the good lighting conditions.

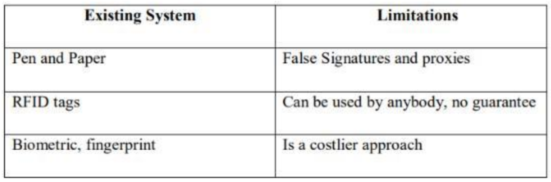
I. LITERATURE SURVEY

A literature survey is a proof of sorts. It is the study of relevant literature materials in relation to a topic we have been given. For developmen of the automated attendance management system using OpenCV, we need to go through each and every technical aspect related to it. This chapter gives a quick overview of the research field. A brief Study and Survey has been carried out to understand various aspects related to the project which involves providing an updated attendance taking system by reducing time and burden on faculty. Our project mainly focuses on the face detection and face recognition to maintain the attendance. Automated attendance management system mainly consists of detecting the faces and load them into a database and recognize the faces in the live video streaming. One can see a face and recognize them quickly without much effort. It is not a big deal for the human, but in the case of computer it is hard to recognize a face with more effectively. There are various difficulties, such as low clarity, light variation, etc. These factors greatly affect the accuracy of a computer to detect faces effectively. Our brain combines the different sources of information into the useful patterns, we don't see the visuals as scatters. If we define face recognition in the simple word, "Automatic face recognition is all about to take out those meaningful features from an image and putting them into a useful representation then perform some classification on them". The basic idea of face recognition is based on the geometric features of a face. It is the feasible and most intuitive approach for face recognition. The first automated face recognition system was described in the position of eyes, ears, nose. These positioning points are called features vector.

In recent decades, a numerous attendance management system based on face recognition has been introduced in order to improve the performance of students in different organization. 4 In Jomon Joseph, K.P. Zacharia proposed a system that uses image processing, PCA, Eigen faces, Microcontroller, based on MATLAB. Their program works only with front faces and there is a need for this an appropriate method that works in the form of system. A author proposed a method using artificial neural networks, and they employed PCA to extract facial images as well as testing and training. Neural networks were used to do this, and their method works well in many positions. A proposal for an attendance management system was made by VeeraMuthu, MuthuKalyani.K, MuthuKalyani.K A authors present an eigen face approach combined with a PCA algorithm for marking face recognition attendance systems, as well as a comparison of several face recognition algorithms. Overall, it was a good strategy for keeping track of attendance. The PCA method was used to develop an efficient attendance management system, which has achieved accuracy of up to 83 percent but suffers from system performance degradation owing to modest changes in light conditions. Another way to make an attendance system simple and secure is to employ artificial neural networks. The author suggested a system that uses PCA to extract facial images and neural networks for testing and training. Their system works in various orientations.

Disadvantages of Existing System :

- Low detection rates.
- Difficult to detect face in complex backgrounds or in presence of multiple faces.
- Application only to frontal faces. May not detect faces with beards.
- Convolution cost is high.
- It has High Dimensional Features Vector
- Sensitive to illumination, face sizes, poses and expressions
- Unable to detect the faces in occlusions and different poses.



Existing System	Limitations
Pen and Paper	False Signatures and proxies
RFID tags	Can be used by anybody, no guarantee
Biometric, fingerprint	Is a costlier approach

Proposed System :

To overcome some of the constraints of existing methods, the proposed approach has to meet specific criteria, such as the device's components having to be low-cost and accurate in order for the system to be economical and reliable. It reduces the flaws of the existing manual system. The system can calculate the attendance of each individual person, where we don't need to give any prior details about that person. It can take the name of the student into the excel sheet once that employee reaches the range of the camera. For face identification, we employed the Histogram of Oriented Gradient and deep learning algorithms to generate and compare 128-d face features. Once faces are spotted and recognised using the current database, the system calculates real-time attendance, and it is saved in the given excel sheet by the system automatically. Our system is divided into two types one is recognition based on the HOG (Histogram of Oriented Gradient) and second is LBP used to detect or identify a image or a real time video. Feature Descriptor for Histogram of Oriented Gradients (HOG), compute feature descriptors/vectors from an image. These characteristics serve as a numerical "fingerprint" for distinguishing one feature from another. The HOG (Histogram of Oriented Gradients) technique counts the number of times a gradient is oriented in a certain area of a picture. The HOG algorithm divides the image into small, connected sections called cells, and calculates the image gradient along the x- and y-axes for the pixels within each cell.

Advantages Over Existing System :

In our work we tried to overcome some of the disadvantage:

- We designed this system as a cost efficient and more reliable.
- We don't need to assign any prior information in the excel sheet.
- This system can easily detect multiple faces at a time.
- One can easily manage attendance with this system.

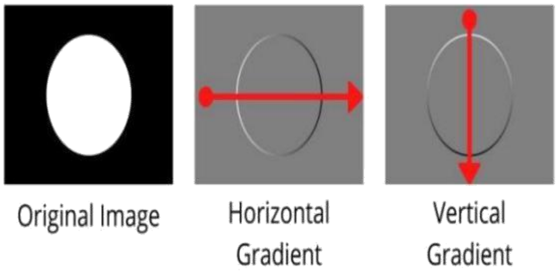
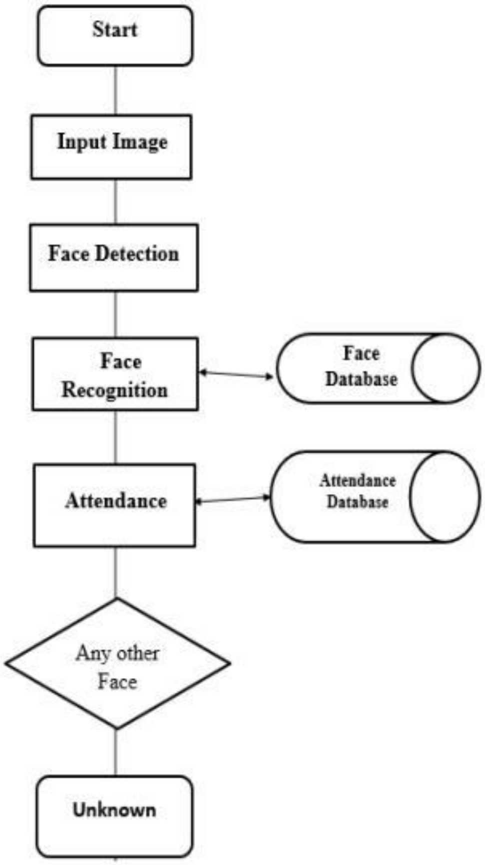


Figure 2.4.1 HOG image gradient

II. METHDOLOGY

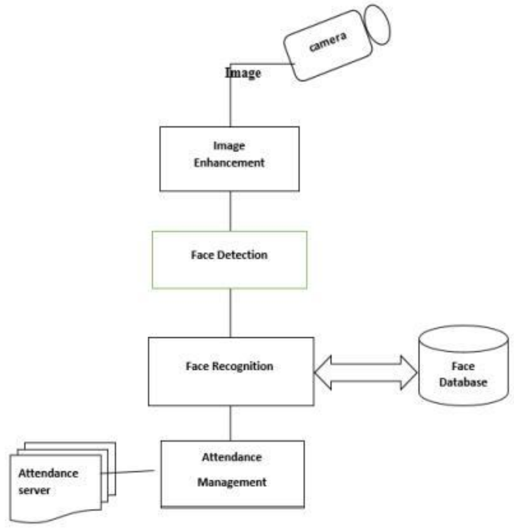
In order to assess employee's performance, all organizations require them to keep track of their attendance. In this sense, each institute has its unique technique. Some people take attendance manually using outdated paper or file-based systems, while others use biometric technology to take attendance automatically. However, with these approaches, peoples must wait a long time to form a line when they enter. There are a variety of biometric methods available, but the key authentications are the same across all modalities. Every biometric system begins with an enrolment procedure in which a person's unique characteristics are entered in a database, followed by identification and verification processes. These two processes compare a person's biometric feature to a template that was previously stored at the time of enrolment. Fingerprints, eye iris, face, hand geometry, signature, gait, and voice are all examples of biometric templates. Our technology employs a facial recognition approach to ensure that employees can automatically take their attendance for their participation. Face recognition consists of two steps: first, faces are recognized in the image, and then these found faces are verified by comparing them to a database. The Ada Boost algorithm, the Float Boost algorithm, the S-Ada Boost algorithm, Support Vector Machines (SVM), and the Bayes classifier have all been presented for face detection. The rapid face detection algorithm can improve the efficiency of the face recognition algorithm. In all the approaches listed above, SURF is the most efficient. The system comprises the employee's photos that are collected in the dataset and delivers them to an image enhancement module for processing. After the image has been enhanced, it is sent to the Face Detection and Recognition modules, where it is recognized, and the attendance is recorded on the database server. The Facial database stores templates of individual employee's face photos at the time of enrolment. The algorithm detects all the faces in the input image and compares them one by one to the face database. If a particular face is recognized, the attendance is recorded on a server that anyone can access and use for various purposes. OpenCV is a cross-platform library that may be used to create real-time computer vision apps. It focuses primarily on image processing, video recording, and analysis, with capabilities such as face detection and object detection.

Computer vision is a discipline that discusses how to reconstruct, interrupt, and interpret a 3D scene from its 2D images in terms of the structure inherent in the scene. It is concerned with employing computer software and hardware to model and replicate human vision. Computer Vision has a lot of overlap with the fields. Image Processing is concerned with the alteration of images. It demonstrates how to classify patterns using a variety of strategies. Photogrammetry is the process of extracting precise measurements from photographs. The development of clear, meaningful descriptions of actual objects from their images is known as computer vision. A description or interpretation of structures in a 3D scene is the result of computer vision. Image processing is concerned with image-to-image conversion. Image processing uses images as both its input and output. The OpenCV library's primary library modules are listed below. Core Functionality: This module introduces the essential data structures used in OpenCV applications, such as Scalar, Point, and Range. It also includes the multidimensional array Mat, which is used to store the images, in addition to these. This module is included in the OpenCV Java library as a package named org.opencv.core. Image Processing: Image filtering, geometrical image transformations, colour space conversion, histograms, and other image processing processes are covered in this module. This module is included in the OpenCV Java library as a package named org.opencv.imgproc. Video: The fundamentals of motion estimation, background subtraction, and object tracking are covered in this module. This module is provided in the OpenCV Java library as a package named org.opencv.video. Video I/O: The OpenCV library is used to demonstrate video capturing and video codecs in this lesson. This module is provided in the OpenCV Java library as a package named org.opencv.videoio. calib3d: Basic multiple-view geometry methods, single and stereo camera calibration, object pose estimation, stereo correspondence, and aspects of 3D reconstruction are all covered in this topic. This module is included in the OpenCV Java library as a package named org.opencv.calib3d. features2d: The fundamentals of feature detection and description are covered in this module. This module is provided in the OpenCV Java library as a package named org.opencv.features2d. Objdetect: Objects and instances of the predefined classes, such as faces, eyes, mugs, people, cars, and so on, are detected in this module. This module is included in the OpenCV Java library as a package named org.opencv.objdetect. NumPy is the most important Python package for scientific computing. It's a Python library that includes a multidimensional array object, derived objects (such as masked arrays and matrices), and a variety of routines for performing fast array operations, such as mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and more. The ndarray object lies at the heart of the NumPy package. Many operations are performed in compiled code for performance, and this encapsulates n-dimensional arrays of homogeneous data types. Between NumPy arrays and normal Python sequences, there are a few key differences: 12 • Unlike Python lists, NumPy arrays have a fixed size when created (which can grow dynamically). When the size of a ndarray is changed, a new array is created and the old one is deleted. • The elements of a NumPy array must all be of the same data type and so have the same memory footprint. The exception is that arrays of (Python, including NumPy) objects can be used, allowing for arrays of various sizes. • NumPy arrays make it easier to do advanced mathematical and other operations on massive amounts of data. Such actions are typically performed more quickly and with less code than utilising Python's built-in sequences. • NumPy arrays are used by a growing number of scientific and mathematical Pythonbased packages; while they normally support Python-sequence input, they convert it to NumPy arrays before processing, and they frequently output NumPy arrays. To put it another way, knowing how to utilise Python's built-in sequence types isn't enough to efficiently use much (if not all) of today's scientific/mathematical Python-based software; you also need to know how to use NumPy arrays. NumPy completely supports an object-oriented approach, beginning with ndarray once more. For instance, ndarray is a class with many methods and attributes. Many of its methods are replicated by functions in the NumPy namespace's outermost namespace, allowing programmers to develop in their preferred paradigm. The NumPy array dialect and NumPy ndarray class have become the de-facto language for multi-dimensional data transfer in Python due to their flexibility. Face Recognition Module is a Python package that allows a computer to automatically detect all of the faces in an image or in a live video stream. Recognize and recognise people by their faces. Because it is not a built-in module in Python, we must load it into our computer. We get a basic commandline application named face recognition, once we install face recognition, which we can use to recognise faces in a photograph or folder full of photographs. OS Module An operating system (OS) is software that manages computer hardware and software resources while also providing common functions to computer programmes.Time-sharing operating systems plan tasks to make the most of the system's resources, and they may also 13 contain accounting software to track the cost of processing time, storage, printing, and other resources. This module allows you to use operating system dependent functions on the go. Many functions to interface with the file system are included in the *os* and *os.path* modules. Although application code is usually executed directly by the hardware and frequently makes system calls to an OS function or is interrupted by it, the operating system acts as an intermediary between programmers and the computer hardware for hardware functions such as input and output and memory allocation. From cellular phones and video game consoles to web servers and supercomputers, operating systems are present on many devices that incorporate a computer. An operating system's components are all designed to make the many components of a computer function together. To use any of the hardware, whether it's as simple as a mouse or keyboard or as complicated as an Internet component, all user software must pass via the operating system. Datetime module Although date and time aren't data types in Python, a module called datetime can be imported to work with both the date and the time. There is no need to install the Python Datetime module outside because it is included in Python. The Python Datetime package provides classes for manipulating dates and times. These classes offer a variety of capabilities for working with dates, times, and time intervals. In Python, date and datetime are objects, so when you manipulate them, you're altering objects rather than strings or timestamps. The DateTime module is divided into six categories: date — An idealised naive date based on the assumption that the present Gregorian calendar was and will always be in use. Year, month, and day are its characteristics. time — An idealised time that is independent of any given day and assumes that each day has exactly 24*60*60 seconds. It has the following properties: hour, minute, second, microsecond, and tzinfo. Datetime — which is a combination of date and time with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo. timedelta — A duration that expresses the difference in microseconds between two dates, times, or datetime instances. tzinfo — This item offers time zone information. timezone — A fixed offset from UTC implementation of the tzinfo abstract base class.

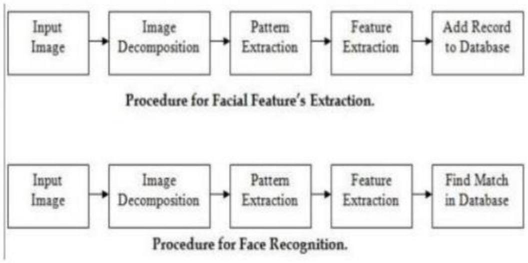


III. System Design and Analysis

The hardware of our system consists of Pc with pre-installed required modules, Highdefinition Cameras, Power supply. By using the high-definition camera, we can train the database with images taken with it and also used detect faces which are already trained in database. Pc should contain pre-installed modules like – pip package manager for Python packages, or modules face recognition, OpenCV, Visual studios.

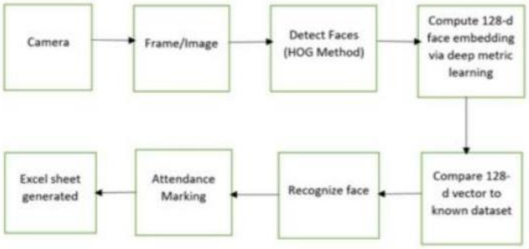


Block Diagram



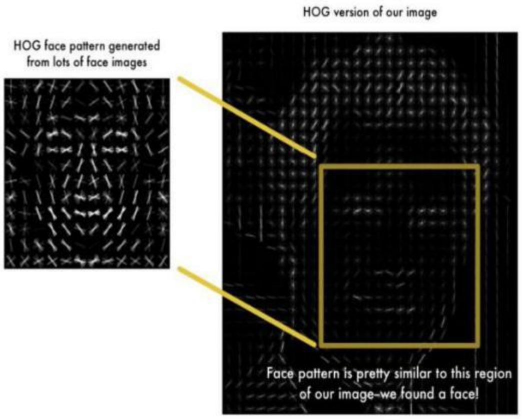
Procedure for Facial Feature's Extraction.

Procedure for Face Recognition.


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

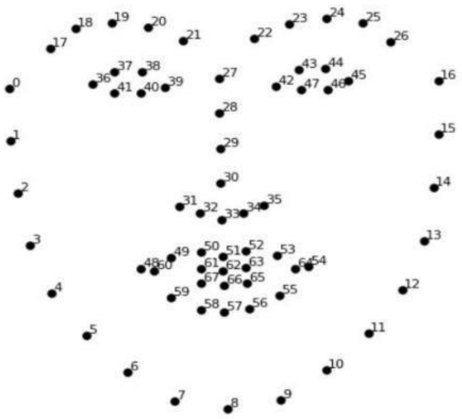
Implementation Steps

Histograms of Oriented Gradients are commonly used to detect and recognize visual objects in computer vision, pattern recognition, and image processing (i.e., faces recognition). To boost detector performance, they are generated on a dense grid of pixels that overlap local contrast histogram normalizations of image gradient orientations. As a result, this feature set performs exceptionally well for many form-based object categories (e.g., face detection) due to the distribution of local intensity gradients, even when the corresponding gradient is unknown. Count the occurrences of edge orientations during a native neighborhood of a picture to derive HOG descriptors. A video is used as an input, which was most likely provided by the user. The image vertical and horizontal gradients can be determined using gradient calculation and a median filter to conduct filtering by value [101] [-1 0-1]. The input video is transformed into an image frame sequence. This image is segmented into 256*256 pixels small cells on average. Each cell is then subdivided into four little blocks, each of which is 2*2 pixels in size. It is obtained using a histogram of an oriented gradient bar graph. The 13 direction channels created in step three are represented by the coordinates of the bar graph. Normalization is the process of representing and associating a vector with pixels. Block normalization and normalized histograms of each block cell are used to correct local contrast. The Histogram of Oriented Gradient is an algorithm that employs a local reference of picture coordinates and calculates the gradient's local direction. Currently, the HOG technique is used in image identification and has a high success rate in detecting human faces. The HOG feature is based on an oriented gradient histogram. It can characterize not only the feature of face contours, but also the fact that it is not light sensitive and has a little offset. By merging the features of all blocks in line, you may get the human alternatives countenanced facial expression. Using the 256 256-input image as an example, fig eleven depicts the technique for obtaining the depth image's HOG options. We calculate the HOG feature as follows: 1) A video is used as an input, which is most likely provided by the user. 2) The image vertical and horizontal gradients can be determined using gradient calculation and a median filter to perform filtering by value [101] [-1 0-1]. 3)The input video is transformed into an image frame sequence. This image is segmented into 256*256 pixels small cells on average. Each cell is then subdivided into four little blocks, each of which is 2*2 pixels in size. 4)An oriented gradient bar graph histogram is obtained. The 13 direction channels created in step three are represented by the coordinates of the bar graph. 5) Normalization is the technique of representing and associating a vector with pixels. Block normalisation and normalised histograms of each block cell are used to correct local contrast. Working of HOG algorithm: Finding all the faces. We don't need colour data to locate faces in an image, we'll start by converting it to black and white. Then we'll go over each and every pixel in our image one by one. We want to look at the pixels that are directly surrounding each pixel. Our goal is to determine how dark the current pixel is in comparison to the pixels in its immediate vicinity. Then we'll add an arrow indicating which direction the image is darkening. If you repeat this method for each and every pixel in the image, each pixel will be replaced by an arrow. Gradients are the arrows that depict the transition from bright to dark throughout the entire image. To accomplish this, we'll divide the image into 16x16 pixel squares. We'll count how many gradients point in each main direction (how many points up, up-right, right, etc...) in each square. Then we'll replace that square in the image with the strongest arrow directions. As a result, we've transformed the original image into a very simple representation that accurately captures the essential anatomy of a face. All we have to do to locate faces in this HOG image is find the region of our image that looks the most like a known HOG pattern that was taken from a bunch of other training faces.


HOG version of our image
HOG face pattern generated from lots of face images
Face pattern is pretty similar to this region of our image-we found a face!

HOG vision of the human face

We were able to isolate the faces in our shot, which was a relief. But now we have to cope with the fact that faces oriented in different orientations seem to a computer in completely different ways. To compensate for this, we'll try to warp each image so that the eyes and lips are always in the sample location. This will make comparing faces in the following steps a lot easier. The main concept is that we will identify 68 distinct places (known as landmarks) on every face, such as the top of the chin, the outside edge of each eye, the inside edge of each brow, and so on. Then, using a machine learning method, we'll teach it to locate these 68 specific locations on any face.



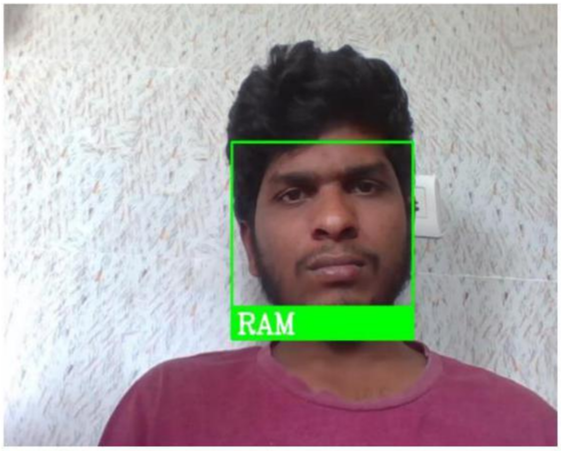
Landmarks on a human face

Now that we know where the eyes and mouth are, all we have to do is rotate, scale, and shear the image to get the eyes and mouth to be as centred as possible. We're not going to apply any sophisticated 3d warps because it would distort the image. Only basic image modifications such as rotation and scale that preserve parallel lines will be used (called affine transformations) To improve detection performance in this project, we used both the local binary pattern (LBP) and the Histogram of Oriented Gradients (HOG) descriptor. The number 4 is used as the central pixel in the graphic, and we may find its neighbourhood pixel values by using this value. If the value of the centre pixel is larger than that of its neighbouring pixel, it is counted as one (1); conversely, if the value of the central pixel is less than that of its neighbouring pixel, it is counted as zero (0).

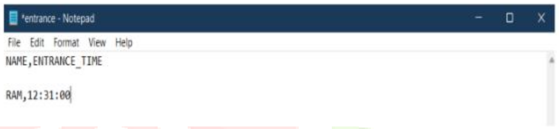
IV. Results and Discussion

IDE is an integrated development environment for Python. This is a software environment which usually consist of a software development package containing Code Editor, build Automation, tools and debugger. Visual Studio is a Microsoft Integrated Development Environment (IDE) that may be used to create graphical user interfaces (GUIs), consoles, Web applications, online apps, mobile apps, cloud, and web services, among other things. With the help of this IDE, you may write both managed and native code. It utilises a variety of Microsoft software development platforms, including Windows Store, Microsoft Silverlight, and Windows API, among others. Because it can be used to write code in C#, C++, VB(Visual Basic), Python, JavaScript, and a number of other languages, it is not a language-specific IDE. It is compatible with 36 different programming languages. It's compatible with both Windows and Mac OS. A high-definition camera is installed in the classroom to capture the image. This image is fed into the system as an input. Prior to the recognition procedure, a student data collection is established. This dataset was developed solely for the purpose of training this system. We produced a dataset with 5 students' names, various stances and variants. A minimum of 15 photos of each pupil should be recorded for improved accuracy. When we register student data and photographs in our system to establish a dataset, deep learning is applied to each face to compute 128-d facial features and save them in the student face data file so that we may recall that face during the recognition process. This procedure is followed for each image captured throughout the registration process.

Model testing for known single User:

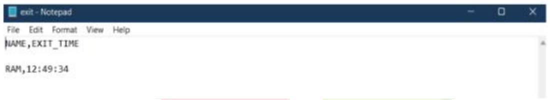


Detecting a single known face



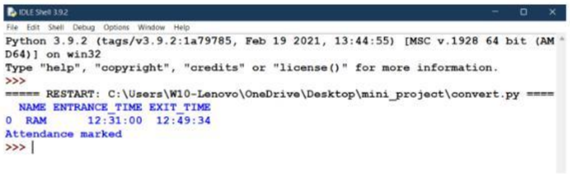
```
NAME,ENTRANCE_TIME

RAM,12:31:00
```

Note the time of entry in a text file of single person



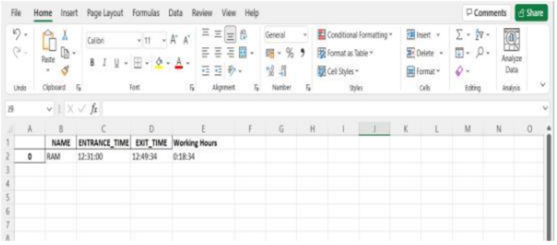
```
NAME,EXIT_TIME

RAM,12:49:34
```

Note the time of exit in a text file of single person



```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\W10-Lenovo\OneDrive\Desktop\mini_project\convert.py ====
  NAME ENTRANCE_TIME EXIT_TIME
0  RAM      12:31:00  12:49:34
Attendance marked
>>> |
```

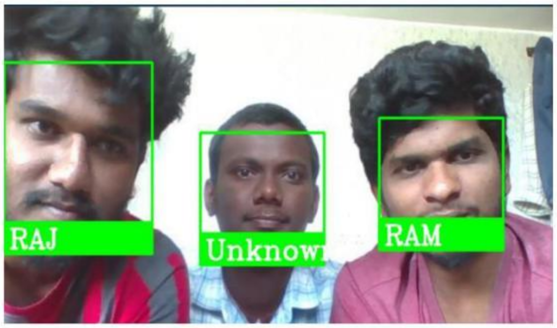
IDLE attendance tracking



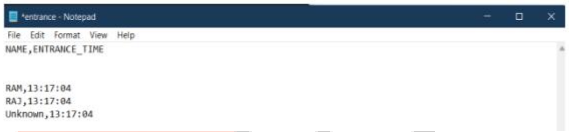
	NAME	ENTRANCE_TIME	EXIT_TIME	Working Hours
0	RAM	12:31:00	12:49:34	0:18:34

Attendance is recorded on an excel sheet.

Model Testing for multiple faces:

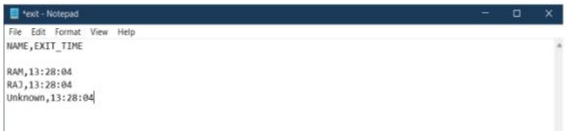


Detection of multiple faces at a time



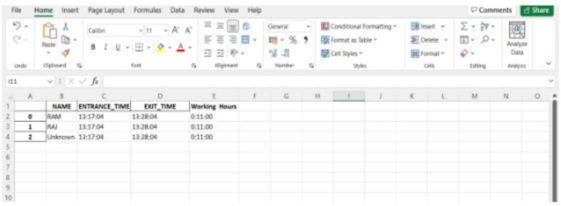
```
NAME,ENTRANCE_TIME

RAM,13:17:04
RAJ,13:17:04
Unknown,13:17:04
```

Note the time of entry in a text file for multiple persons



```
NAME,EXIT_TIME

RAM,13:28:04
RAJ,13:28:04
Unknown,13:28:04
```

Note the time of exit in a text file for multiple persons



	NAME	ENTRANCE_TIME	EXIT_TIME	Working Hours
0	RAM	13:17:04	13:28:04	0:11:00
1	RAJ	13:17:04	13:28:04	0:11:00
2	Unknown	13:17:04	13:28:04	0:11:00

Attendance being marked using excel sheet

V. Conclusion

Automatic attendance management system using OpenCV, HOG and LBPH algorithms is developed successfully. This Project was created to address the shortcomings of manual systems. We employed the face recognition technique to track student attendance and improve the system. The method works well in a variety of stances and variations. Using facial recognition algorithms, this research proposes a simple yet effective method for calculating class attendance. This system's output can be summarised as follows: As shown in Fig. 6, the system not only recognises a single employee face, but it also recognises several employees or faces. In addition, the system correctly recognises and records the attendance of the detected faces.This system will need to be enhanced in the future because it sometimes fails to recognise faces from a distance, and we also have certain processing limitations. • Working with a high-processing system may result in even better performance of this system.

References

1. Kar, Nirmalya, et al. "Study of implementing automated attendance system using face recognition technique." International Journal of computer and communication engineering 1.2 (2012)

2. Selvi, K. Senthamil, P. Chitrakala, and A. Antony Jenitha. "Face recognition-based attendance marking system." (2014)

3. Bhattacharya, Shubhobrata, et al. "Smart Attendance Monitoring System (SAMS): A Face Recognition Based Attendance System for Classroom Environment." 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT). IEEE, 2018.

4. Karnalim, Oscar, et al. "Face-face at classroom environment: Dataset and exploration." 2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA). IEEE, 2018.

5. Abdul, Rhman Salim, Rashidah Funke Olanrewaju, Wasiu Adebayo Balogun. “Class Attendance Management System Using Face Recognition." 2018 7th International Conference on Computer and Communication Engineering (ICCCE) IEEE 2018.

6. Jinsu Kim, Usman Cheema, Seungbin Moon. “,Face Recognition Enhancement by Employing Facial Component Classification and Reducing the Candidate Gallery Set. Department of Computer Engineering, Sejong University, Seoul, 143-747, Korea (sbmoon@sejong.ac.kr).

7. M. Joe Minichino, Joseph Howse. Learning OpenCV 3 Computer Vision with Python. 2016: 16-82.

8. Face Detection and Tracking using OpenCV. S.V.Viraktamath, Mukund Katti, Aditya Khatawkar, Pavan Kulkarni. 3, s.l.: SIJ, July-August 2013, The Standard International Journals (The SIJ) , Vol. 1, pp. 45-50. ISSN: 2321 – 2403

9. Parmar, D.N. and Mehta, Face Recognition Methods & Applications.arXiv:14030485

10. Face Detection in Real Time Based on HOG. N. J. Wang,S. C. Chang and P. J. Chou. Taipei, Taiwan: IEEE, DOI:10.1109/ISPACS.2012.6473506, 2012. International Symposium on Intelligent Signal Processing and Communications Systems. pp. 333- 337. ISBN: 978-1-4673-5081-5.