FACIAL RECOGNITION ATTENDANCE SYSTEM USING PYTHON, OPENCV AND HAARCASCADE ALGORITHM

Roselinmary S*,Maheshwari M*,AnandJosephDaniel*,Nivetha.M*,Minu.K*

Computer Science And Engineering

Anand Institute of Higher Technology

ABSTRACT

This project's major goal is to develop a facial recognition-based attendance monitoring system for educational institutions in order to improve and modernise the current attendance system and make it more effective and efficient than it was previously. The existing outdated system is very ambiguous, which results in erroneous and ineffective attendance taking. When the authorities cannot enforce the rules that are present under the previous system, many issues occurThe key component will be face recognition technology. The face is one of the physical traits that can be utilised to specifically identify a person. A face is used to trace identification because it is unlikely that it would change or be duplicated. In this project, face databases will be created to feed information into the recognizer algorithm. Then, during the period for recording attendance, faces will be checked against the database to try to identify anyone. When a person is recognised, their attendance will be recorded automatically, recording the essential data into an excel sheet. The excel document with all students' attendance data is mailed to the appropriate faculty at the end of the day.

Keywords :Face recognition, attendance system ,haar cascade algorithm

1.INTRODUCTIONS

An image is a column- and row-organized array, or matrix, of square pixels (picture elements). Based on the quantity of pixels, its dimensions (height and breadth) serve as a representation. The point on the image known as a pixel assumes a particular shade, level of transparency, or colour. Each pixel of an image must undergo a fixed series of operations during image processing. The initial series of actions are carried out pixel-by-pixel by the image processor on the image. The second operation will start once this is finished in full, and so forth. Any pixel in the image can be used to determine the output value of these procedures. The process of converting an image into a digital format and carrying out specific procedures to extract some usable information from it is known as image processing. When using specific specified signal processing techniques, the image processing system typically interprets all images as 2D signals. One of the computer vision technologies that has received the greatest research attention is face recognition, with new methods and promising findings being published every year. Features-based and holistic approaches to face recognition are the two main categories.

Faces are recognised utilising global features from faces in holistic based approaches, but in feature-based approaches, faces are recognised using local features from faces. The ideal solution to these issues is a recognition system. Face recognition is incredibly useful in today's digital world. Particularly for workplaces where attendance must be verified. Perhaps some regions are now using technology to confirm attendance. The main obstacles for effective face detection and identification systems include lighting conditions, scale, occlusion, position, background, emotion, and others Numerous strategies and methods have been developed to address these problems. Utilizing Convolutional Neural Networks, the complex wavelet transform (CWT), and Fisherface to address shift and rotation, illumination invariant face recognition is used to resolve lighting issues The Haar cascade and Local Binary Pattern histogram methods are proposed in this study. The use of LBPH increases detection. Haar cascades are quick and effective in real-time applications.

3.PROPOSED SYSTEM

To deal with these issues, numerous tactics and solutions have been created. Illumination invariant face recognition is utilised to overcome shift and rotation problems using Convolutional Neural Networks, the complex wavelet transform (CWT), and Fisherface.

2.1 METHODOLOGY

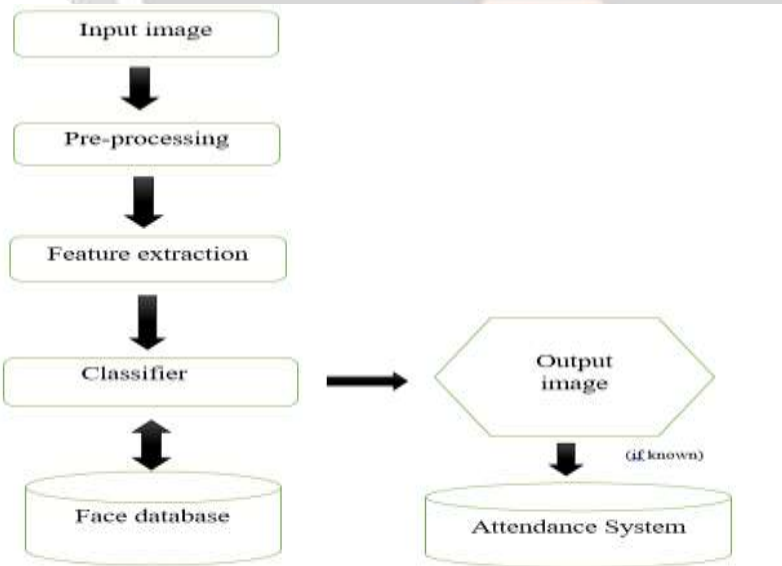
- Haar Cascade Classifier

OpenCV encompasses both of these approaches. A traine and a detector are included with OpenCV. So, if you want to train your classifier to recognise any item, you can use the Haar Cascade Classifier.

2.2 HAAR CASCADE CLASSIFIER

It is efficient to find items by using Haar cascade classifiers. A cascade function must be trained as part of the object detection machine learning technique. utilising a lot of both positive and negative pictures. A classifier that is trained using several examples of a particular object and uses boosted classifiers with haar-like features is a good example (e.g., a face or an automobile). Therefore, you will be able to detect anything you want if you train your classifier with those kinds of numbers. The image that lacks a face, for instance, is referred to as a negative image if we are trying to detect a face. The term "portrait" or "positive photos" is also used when an image has a face or a group of faces because it is deemed to be a portrait. After being trained, a classifier can be used to analyse a particular area of interest. The classifier in an input image returns 1 when a region is likely to contain the object and 0 when it is not. Here, we'll be focusing on facial recognition. The classifier needed to be trained at first. A lot of positive (face-containing) and negative (non-face-containing) photos are needed for the cascade function. a picture (images without faces). We must then extract features from it after that. They are similar to one another. Convolutional kernel is what we have. The total number of pixels under the white rectangle is subtracted from the total number of pixels under the black rectangle to provide a single value for each feature..

3.BLOCK DIAGRAM



3.1 BLOCK DIAGRAM PROPOSED METHOD

When we start the application, the computer's built-in camera opens and starts taking pictures of the user. The folder named Student Details has the file with the names Student Details that contains this Id. It stores a few images as an example in the Training Image folder. After completion, it notifies you that the photos have been saved. The ID and name of the student are displayed on the image if the camera recognises their face. Use Q to close this window (or q). After leaving, the name, ID, date, and time of the person who attended will be kept in the Attendance folder.

Algorithm steps

(1) It catches the input picture first and foremost.

(2) After capturing the image, it preprocesses it and converts it to a grayscale image.

(3) Face detection will be performed using the Haar Cascade Classifier, which collects features from the picture and stores them in a trained set database.

(4) Similarly, the Local Binary Patterns Histogram is used to recognise faces.

The retrieved features will then be compared to the trained data set in step 5.

(5) If the information is correct, the attendance folder will be updated.

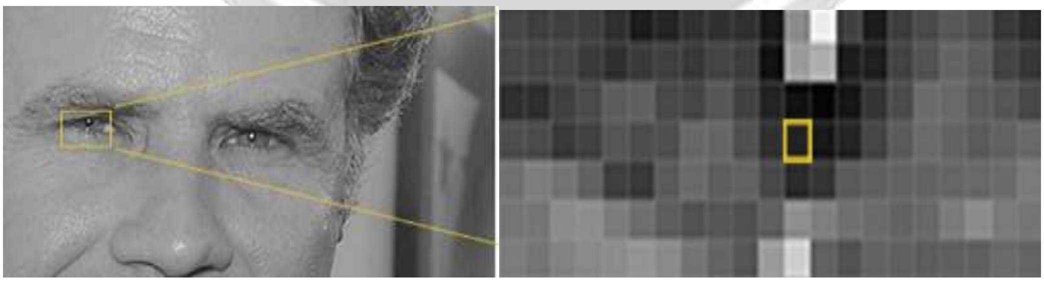
(6) If there are no matches, the attendance folder will not be updated.

3.2 IMPLEMENTATION RESULTS

Step 1: Locate all of the Faces

Face recognition is a fantastic camera function. When a camera can detect faces automatically, it can ensure that all of them are in focus before taking a photo. Because we don't require colour data to locate faces in an image, we'll start by converting it to black and white.

BLACK & WHITE IMAGE

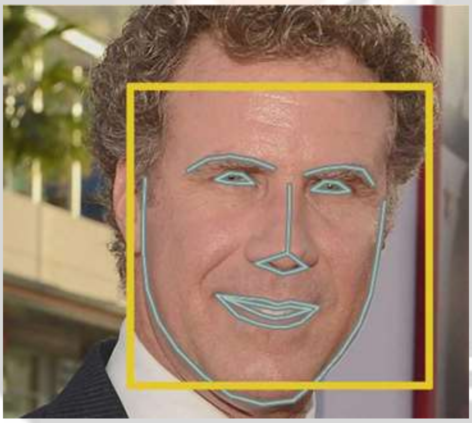


SINGLE PIXEL

Then we'll go over each and every pixel in our image one by one. The pixels immediately surrounding each pixel are what we're interested in. Our goal is to see how dark the current pixel is in contrast to the pixels around it. After that, we'll add an arrow to show which direction the image is darkening. Each pixel in the image will be replaced by an arrow if you repeat this approach for each pixel. These arrows are referred to as gradients. The picture will be broken into little 16x16 pixel squares. In each square, we'll count how many gradients point in each primary direction (up, up-right, right, etc.). Then we'll use the strongest arrow directions to replace that square in the image. As a result, the original image has been turned into a relatively basic depiction that properly depicts the core structure of a face.Using this method, we can now readily find faces in any image.

Step 2: Projecting Faces and Posing

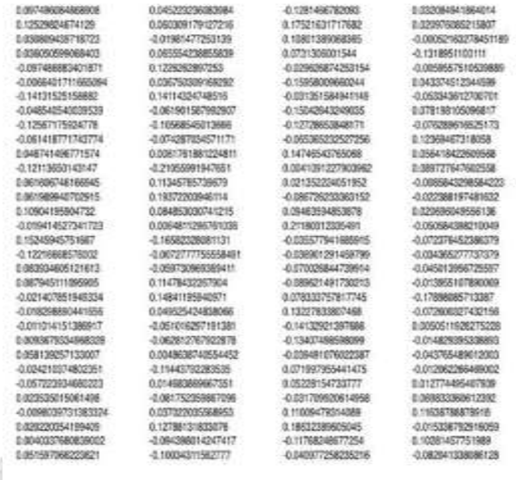
Humans can determine right away that the two images are of the same person. Computers, on the other hand, would see these two photos as two separate people. To compensate, we'll modify each image so that the eyes and mouth are always in the sample area.To accomplish so, we'll use a technique called facial landmark estimation. We'll look for 68 different points on each face (known as landmarks), such as the top of the chin, the outside edge of each eye, the inner edge of each brow, and so forth. Then we'll train a machine learning system to find these 68 unique places on any face.



LANDMARKS ON TEST IMAGE

Step 3: Encoding our picture of our face

It takes a lot of data and computing resources to train a convolutional neural network to output face embedding. However, once trained, the network can provide measures for any face, including those it has never seen before. We only need a technique to get a few basic measures from each face. Then we might use the same method to measure our unknown face and locate the known face with the closest measurements. We may, for example, take measurements of the size of each ear, the distance between the eyes, the length of the nose, and so on. To collect the 128 metrics for each face, we just need to put our face photos through their pre-trained network.



0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.0072777755558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

MEASUREMENT OF INPUT IMAGE

Step 4: Deciphering the encoding for the person's name

This final step is actually the simplest of the entire procedure. All we have to do now is look through our database of known persons to discover the person with the most similar dimensions to our test image. All we have to do now is train a classifier to take the measurements from a fresh test image and determine which known individual is the most similar.

4.CONCLUSION

Prior to the creation of this project. The traditional system of taking attendance has a lot of flaws, which has produced a lot of problems for most institutions. Since the previous method had weaknesses, the facial recognition feature built into the attendance monitoring system can not only guarantee that attendance is taken accurately but also fix those flaws. Utilizing technology to eliminate flaws not only conserves resources but also minimises the need for human involvement by delegating all the difficult tasks to machines. The sole expense associated with this method is having enough room in the database storage to accommodate all of the faces. Thankfully, there are devices like micro SD that can make up for the data volume The face database is successfully constructed in this project. In addition, the face recognition system is operating effectively. By delivering the attendance sheet to the respected professors, the system ultimately not only fixes issues with the old model but also makes it convenient for users to get the information.

REFERENCES

- "Attendance System Using NFC Technology with Embedded Camera on Mobile Device" (Bhise, Khichi,Korde,Lokare, 2015)
- K.SenthamilSelvi,
- P.Chitrakala,A.AntonyJenitha, "Face Recognition Based Attendance Marking System", IJCSMC, Vol. 3, Issue. 2, February 2014.
- "Fingerprint Based Attendance System Using Microcontroller and LabView" (Kumar Yadav, Singh, Pujari, Mishra, 2015)
- "RFID based Student Attendance System" (Hussain, Dugar, Deka, Hannan, 2014)
- Open Cv Documentation -https://opencv.org
- Numpy - https://numpy.org