WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Face Recognition for National Security

Dhonushree Banerjee, Swapnil Ingole, Shritika Mandal, Aman Shrivas, Nischay Dekate, Smit Patil

Dhonushree Banerjee,Gajanan societ,y dattawadi, Nagpur 440023, India
Swapnil Ingole,Midc area, Balrampur,New Nanded 431603, India

A B S T R A C T

Face recognition is a biometric software application adapted to identify individuals via tracking and detecting. The main purpose of this paper is to recognize the faces of people. This approach can be operated practically in crowded areas like airports, railway stations, universities, and malls for security. The main target of this paper is to enhance the recognition rate. After the event of 26/11, developing security systems has become more concerned importance to provide safety to the citizens, particularly in crowded areas like airports, railway stations, in borders, organizations where detection and recognition are imperative. Also, we can use this system for covid security, to recognize a person's face and checktheir previous medical records or if the person is covid positive then it will show us or the record will transfer to medical department and quick actions can be taken.

This System we can use for detect criminal or terrorist faces once it get detected it will transfer information to the nearby police station or other security forces through internet of things and after that security will check same record because persons face can be mistakenly change or wrong face can be detect, then it will check record and if matches the informationand face then security system will get activate and then action can be taken.

1. Introduction

As the below figurethe proposed technique deploys two progresses of images such as the input images and the image captured through live streaming. Both these process undergoes four common procedures namely, face acquisition, pre-processing, face detection using Haar- cascade classifier and feature extraction using Linear Binary Pattern algorithm to compute LBP values. These values are keep within the information solely just in case of process associate degree input image. Finally, comparison of the values in the database with the values computedvia live streaming takes place which recognizes the human face as known or unknown based on the matching.

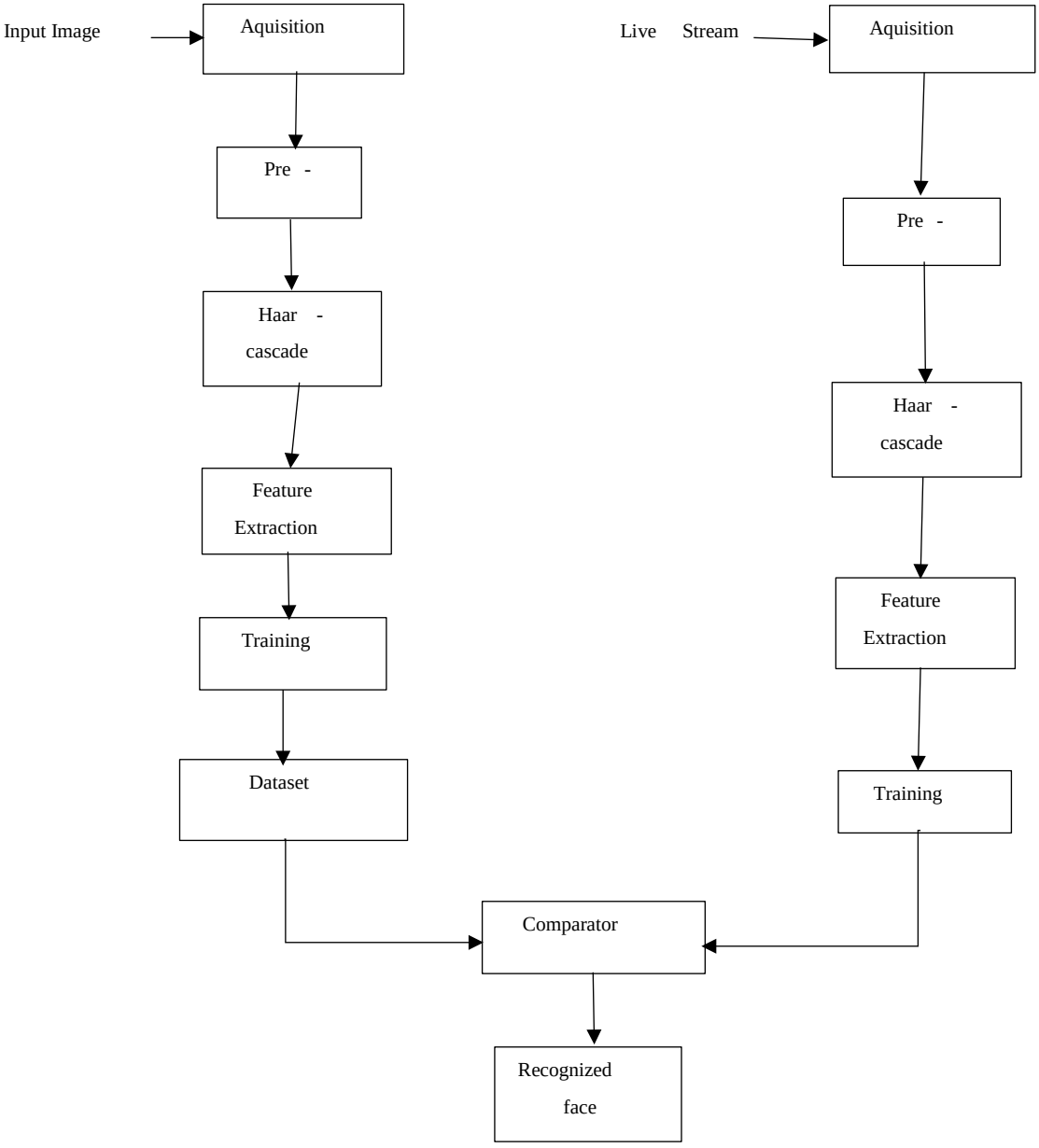


Figure 1. Flow Chart

The three main process followed in face recognition system are Detection, Feature extraction and Comparison.

1. Face detection :

For face detection, Viola Jones algorithmic program may be a useful methodology.In general, this formula isn't solely restricted for face detection however also can be utilised for several rigid structured object detection tasks. The Viola-Jones formula consists of 3 main ideas that build it potential to develop a true time face detector: Haar-like options, Image integral, Adaboost coaching and Cascading classifier. By applying these features, the system can determine the presence or the absence of a human face.

2. Haar-like features :

Haar-like features is used by Haar cascade classifier for human face detection. There are three forms of Haar- like features. From below fig, the first format is the edge feature, second type is the line feature and the last type is the four rectangle feature Using the integral image, Haar like principle can offer quick computation.It's called Haar-like features.

The Algorithm looks for specific haar feature of a face. This detection takes the image and converts it into 24X24 window and smears each Haar feature to that window pixel by pixel. Initially, the algorithm requires a lot of positive images (images of faces) and negative images (images without

faces) to train the classifier.

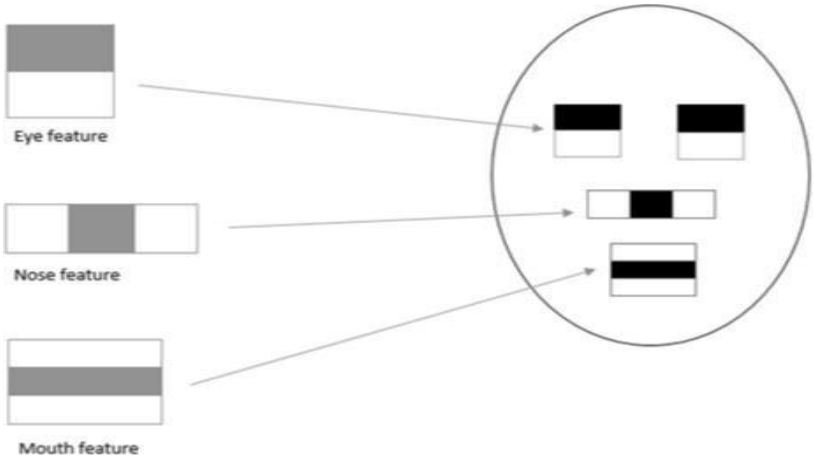


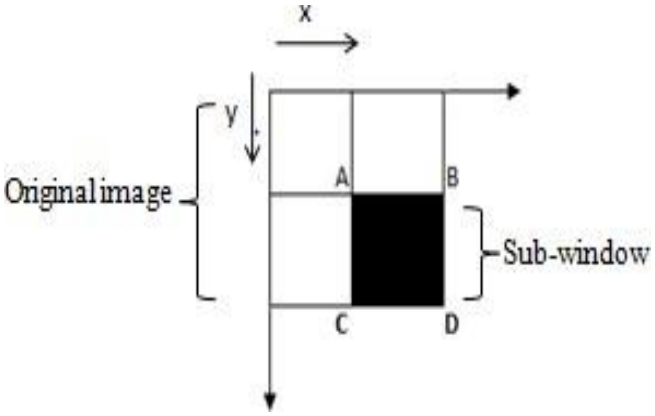
Fig.ure2 : Types of Haar- like features

Then, Features are numerical worths determined from pictures that ar accustomed distinguish one image from another every feature may be a single value noninheritable by subtracting the add of the pixels beneath the white rectangle from the sum of the pixels beneath the black rectangle.

Feature = Σ (pixels in black area) dark - Σ (pixels in white area) white (1)

3.Integral Image

Rectangle features can be determined rapidly via an intermediate representation of the image called the integral Image. The integral image includes of tiny units illustration of a given image.



SHADED REGION = D - (B+C) – A

Figure 3 : Integral image schematic diagram

For example, the value of this integral image at position1 is the sum of pixels in rectangular A. The value at position 2 is A + B and so on. So, the sum total of pixels in rectangular D is:

S (D) = ii (4) – (ii (3) + ii (2)) + ii (1) (2)

Where, S(D) is the sum of pixels in the rectangular D only - which is the sum of pixels in the rectangle A + B + C + D, represented by ii(4); ii(3) is the integral image of rectangle A+C ; ii(2) is the integral image of A+B and finally ii(1) is the integral image of the rectangle A (the addition is executed since the region A is subtracted twice in ii(3)and ii(2)). The integral image is outlined as:

ii [x , y] = i [x' , y'] (3)

Where, ii[x, y] represents integral image, and i [x', y'] represents original image.

The pixel value of integral images at any (x,y) location is the sum of all pixel values displayed before the current pixel. The integral value of an individual pixel is the sum of pixels on the top and the pixel towards the left.

4. AdaBoost Learning :

AdaBoost is an adequate boosting algorithm which combines weak classifiers while reducing significantly not only the training error but also the more elusive generalized error. The main idea of Boosting lies in connecting the simple classifiers which are known as weak classifiers. Since the weak classifiers do not expect even the best classification function to classify the data well, they are called as weak classifiers. Here a classifier is combined with a single feature to easily link the Haar features with weak classifier. Haar-like feature is used as a threshold in AdaBoost learning algorithm by Viola and Jones. The Haar- classifier is the strongest classifier since it uses the strongest specifications. The positive and negative samples are best differentiated by the feature. In order to build a strong final classifier AdaBoost is used . It reduces the features from 160000 to 6000, thus making the computation simpler and hence it is less in computational complexity.

5. Cascade Classifier:

Cascade classifier can be a cascading of weak classifiers accustomed boost the face detection method and scale back the procedure quality. every node within the series contains a weak classifier and filter for one Haar feature. AdaBoost provides weights to the nodes and therefore the highest weighted node primarily arrives. once a filter ignores to allow image regions, that specific sub window of the image is eliminated for more process. it's then thoughtabout as a non-face, which suggests that the image regions that ar processed don't contain the face to be detected. this can be terribly imperative to the performance of the classifier, since all or nearly all negative image sub-windows are going to be eliminated within the 1st stage.

On the contrary, when image regions successfully passed the filter, they go to the following stage, which contains a more complex filter. Only regions that with success passess all filters are considered to contain a match of the face. This means that regions of the image contain the facial subject for detection. The reason behind the multi-stage classifier is to eliminate efficiently and rapidly the non-face sub- windows. The classifier is employed to reject a lot of false positives (non-face regions) of the sub-windows. the amount of false positive rate is drastically reduced once many steps of process.

After the face is detected, next step is to extract features this is done using linear binary pattern algorithm. Initial step of this algorithm is to convert the test image into gray scale. This L x M pixel size image will get divided into regions. The same pixel size is used for the regions, producing n x n regions. Each region will goes through Linear binary pattern operator.

In this process, it will compare the center pixel with its neighbor pixels. If the pixel size is greater to center pixel it is '1' or it is '0'.

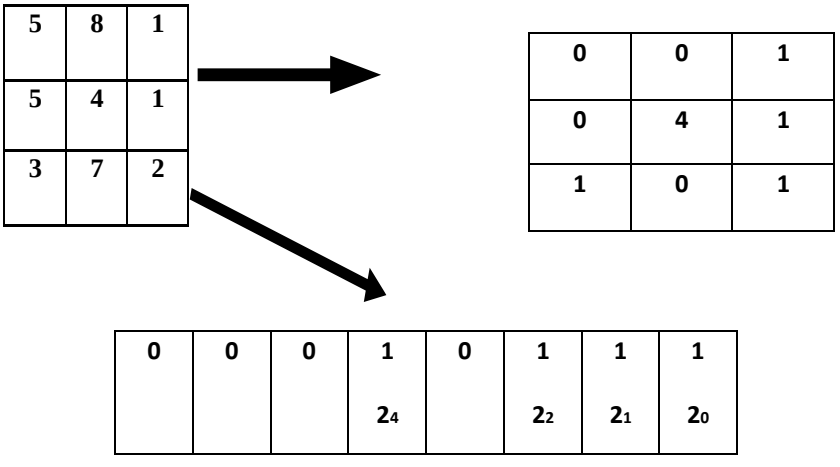


Figure 4. LBPH algorithm example

Executing this process will result in 8binary values. By linking the binary values it results in binary number. The LBP value is obtained by translating 8binary number into a decimal number, it will be in the range of 0-255. This algorithm implementation is shown in the above Fig. 7. The histogram for each region is drawn using the LBP Values of each region. Each region will contain 256 cases. This implementation is shown in the below equation:

$$NX = \sum_{i,j} X\{LBP?(Y(i,j)) = x\}, \quad x = (0,\ldots, 255) \quad (4)$$

Where, Nx is a case of value x, Y (i , j) is the (i , j) pixel of Image and X is the conditional operator, providing '1' when it is true or '0'. After finding the histogram for each region, the sole histogram is created by uniting each region histogram. The final histogram is in the form of 256 *n *n cases and it is determined as the image feature vector. The drawback of this algorithm is it has a fixed scale (3 x 3 scale). To overcome this, there is an extension of original LBP implementation to handle multiple neighborhoods. There are two parameters: first is 'p' which is the number of points in the symmetric circle neighborhood, second is 'r' the circle radius.

There is an important concept called LBP uniformity. A LBP is uniform if it has at most two 1-0 or 0-1 transitions, for example: consider pattern 10000000(1 transition) when 00100000(2 transitions) they both are uniform, the pattern 00100100(4 transitions) is considered as uniform. LBP uniformity completely depends on the 'p' value. When p increases resulting histogram dimensionality increases.

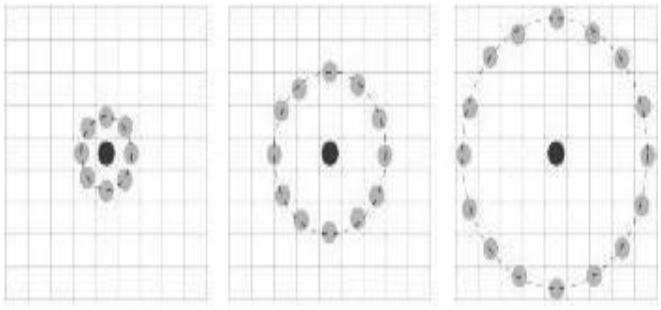


Figure 5. Represents varying p and r to form a Local Binary pattern.

2. Illustrations

In the first module, the Camera will open using OpenCV (Computer Vision) library. The camera will detect the face from the whole picture and blur the background(using the frontal face algorithm). It will capture a minimum of 100 photos of a particular person at a time.

In the second module, it will save in the path. It will provide a specific ID to every person. Everyone has a different face structure. It will recognize different face structures. It will do calculations of face structure using NumPy libraries. After thousands of calculations, an XML file will be created and it will save in that file folder. A training folder will be created and all information will save in that folder.

In our last module, When a known person is captured it will show the accuracy and when an unknown person is captured through the camera / CCTV it will generate a message and send an email to the police station. If any quarantine person is captured then it will give a voice message. Captured pictures of the unknown person will save in a police folder and then we can send that person's information to the police via mail.

3. Equations

This implementation is shown in the below equation:

$$NX = \sum_{i,j} X\{LBP?(Y(i,j)) = x\}, \quad x = (0,\ldots, 255)$$

4. Acknowledgements

At the time of forming the dataset, each person will get designated using an id number. While recognition, when the test person image matches with the dataset then a message will get send like an unauthorized person symbolizes criminal, terrorist or Sleeper cells Or infected by virus like Covid to nearby police station or other security center through the internet of things, if the test person image does not get matched with the dataset then no message will get send symbolizes a normal human being.

In this work, we have discussed the concept of face detection using OpenCV in Python using Haar Cascade. We used the rich library set of OpenCV for a robust face detection from a sample image. For training the model with the feature set of a face, we used the "Haar frontal face" XML file. Security is an imperative part of any industry. This work is most particularly for criminal identification. The algorithms carried out in this paper were Viola-Jones algorithm and Linear binary pattern algorithm. The presented system will get implemented using Open CV and Raspberry pi. The recognition rate attained by this process is 90%-98%. There will be deviation in the result on account of the distance, camera resolution and lightning. Advanced processors can be put to use to reduce the processing time.By affixing more number of recognition servers to attenuate the processing time for collection of images.

REFERENCES

1. KushsairyKadir ,MohdKhairiKamaruddin .Haidawati Nasir, Sairul I Safie,Zulkifli Abdul KadirBakti." A comparative study between LBP and Haarlike features for Face Detection using OpenCV", 4th International Conference on Engineering Technology and Technopreneuship (ICE2T), 2014.

2. SouhailGuennouni, AnassMansouri."Face Detection: Comparing Haar-like combined with Cascade Classifiers and Edge Orientation Matching", International Conference onWireless Technologies, Embedded and Intelligent Systems(WITS), pp. 02-04, 2017.

3. Jiwen Lu, Jie Zhou. "Learning CBFD for Face Recognition."IEEE Transactions on Pattern Analysis and MachineIntelligence, Vol:37 , Issue: 10 , pp.10-12, Oct. 2015.

4. Ayman A. Wazwaz, Amir O. Herbawi, Mohammad J. Teeti, Sajed Y.Hmeed. "Raspberry-Pi and Computers- Based Face Detection and Recognition System", 4th International Conference on Computer and Technology Applications, pp. 01- 03, 2018.

5. Viola, P..& Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR, 2001), December 8-14, 2001, Kauai, HI, USA. 2.

6. Kirby, M., Sirovich, L. (1990) Application of the Karhunen-Loeve procedure for the characterization of human faces. IEEE Transaction of Pattern Analysis and Machine Intelligence, Vol 12, No 1, January 1990., pp. 103 – 108. 3.

7. Liao, S., Jain, A.K., Li, S. Z. (2016). A fast and accurate unconstrained face detector. IEEE Transaction of Pattern Analysis and Machine Intelligence, Vol 38, No 2, pp. 211 – 123. 4.

8. Luo, D., Wen, G., Li, D., Hu, Y., and Huna, E. (2018). Deep learning-based face detection using iterative bounding-box regression. Multimedia Tools Applications. DOI: https://doi.or/10.1007/s11042-018- 56585