International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 10 | Oct 2019 www.irjet.net p-ISSN: 2395-0072

Automated Criminal Identification System using Face Detection and Recognition

Piyush Chhoriya

UG Scholar, Computer Engineering Department, K. K. Wagh Institute of Engineering Education & Research, Nashik-422003, Maharashtra, India

---

Abstract - *As the world has seen exponential advancement over the last decade, there is an abnormal increase in the crime rate and also the number of criminals are increasing at an alarming rate, this leads toward a great concern about the security issues. various causes of theft, stealing crimes, burglary, kidnapping, human trafficking etc. are left unsolved because the availability of police personnel is limited, many times there is no identification of the person who was involved in criminal activities.*

To avoid this situation an automated facial recognition system for criminal identification is proposed using Haar feature-based cascade classifier. This paper presents a real-time face recognition using an automated surveillance camera. This system will be able to detect and recognize face automatically in real-time.

***Key Words*: Criminal Identification, Facial recognition, Haar Classifier, Real-time, Viola-jones.**

1. INTRODUCTION

The face is crucial for human identity. It is the feature which best distinguishes a person. Face detection and recognition is the technology which is used to identify a person from a video or image. The whole Face Detection and Recognition process can be divided into four major steps:

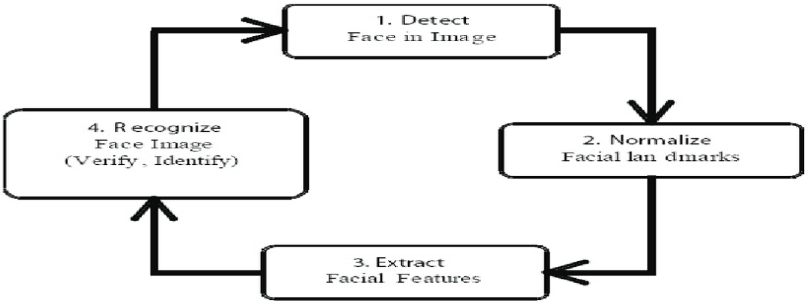


Fig.1 Face Detection and Recognition process

In this paper, we propose a face detection and recognition system for criminal identification using python along with OpenCV package. The proposed system consists of 4 steps: (1) Training of images (2) Face detection using Haar cascade classifier (3) comparison of trained images with images captured from the surveillance camera (4)result based on the comparison.

2. METHODOLOGY:

In this project, we can detect and recognize the faces of the criminals in a video stream obtained from a camera in real-time. The system consists of three databases. First is the citizen database, which will contain the images and unique-id of all the citizens living in that country. Second is local watch list database, which will have the images(min 10) and details(Unique-id, Name, Gender, Religion, Crimes done, etc) of each criminal who belongs to that country. Third is International watch list database, which will have the images (min 10) and details(Unique-id, Name, Gender, Religion, Crimes done, etc) of the criminals who are not the citizens of that country. All the images are first preprocessed. Then it goes through feature extraction where Haar cascade is used.

The video is captured from the surveillance camera which are converted into frames. When a face is detected in a frame, it is preprocessed. Then it goes through feature extraction where Haar cascade is used. The features of the processed real-time image is compared with the features of processed images which are stored in the citizen database. If a match is found, it is further compared with the features of images stored in a local watch list database to identify if the person is criminal or not. If

he is criminal a notification is sent to the police personnel with all the details and the time for which he was under the surveillance of the camera. If he is not a citizen of that country, it is then compared with the features of images stored in the international watch list database. If a match is found, a notification is sent to the police personnel with all the details and the time for which he was under the surveillance of the camera. If a match is not found in both the watch lists, he is innocent.

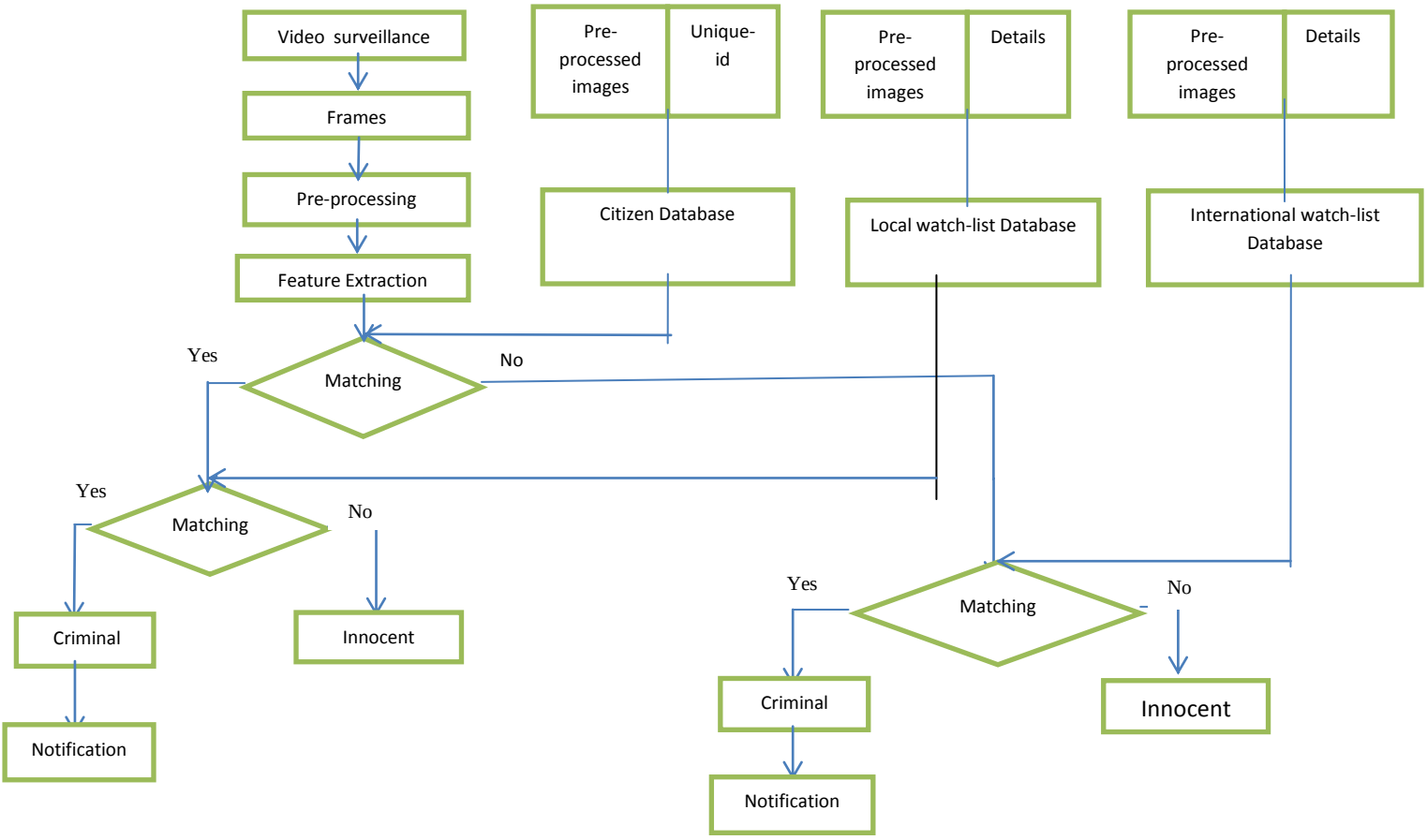


Fig -2: Flowchart of proposed methodology

3. IMPLEMENTATION:

3.1 Modules required

The Modules required to perform the facial recognition are cv2, image module and numpy. cv2 is the OpenCV module and contains various functions for face detection and recognition. The dataset images are in gif format and, OpenCV does not support gif format, Image module from PIL is used to read the image in grayscale format. Numpy is a python library used for scientific computing. NumPy arrays are used to store the images.

3.2 Load the detection Cascade

To Load the face detection cascade the first step is to detect the face in each frame. Once we get the region of interest containing the face in the image, we use it for training the recognizer. For the purpose of face detection, we will use the Haar Cascade provided by OpenCV. The haar cascades that come with OpenCV are located in the directory of OpenCV installation. Haar cascade frontal face default.xml is used for detecting the face. Cascade is loaded using the cv2 CascadeClassifier function which takes the path to the cascade xml file. If the xml file is in the current working directory, then the relative path is used.

3.3 Create the Face Recognizer Object

The next step involves creating the face recognizer object [4]. The face recognizer object has functions like FaceRecognizer.train() to train the recognizer and FaceRecognizer.predict() to recognize a face. OpenCV currently provides

Eigenface Recognizer, Fisherface Recognizer and Local Binary Patterns Histograms(LBPH) Face Recognizer [4]. We have used LBPH recognizer because Real life isn't perfect. We simply can't guarantee perfect light settings in your images or 10 different images of a person. LBPH focus on extracting local features from images. The idea is to not look at the whole image as a high-dimensional vector but describe only local features of an object. The basic idea of Local Binary Patterns is to summarize the local structure in an image by comparing each pixel with its neighbourhood.

3.4 Perform the training

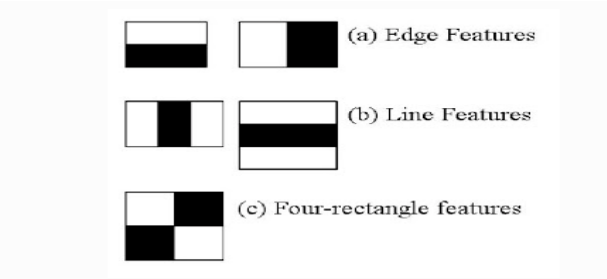
3.5 Perform Testing

4. HAAR CASCEDE CLASSIFIER

The face detection algorithm proposed by Viola and Jones is used as the basis of our design [1]. The face detection algorithm looks for specific Haar features and not pixels of a human face. This method consists of four steps-

4.1 Haar Feature Selection

After the tremendous amount of training data (in the form of images) is fed into the system, the classifier begins by extracting Haar features from each image. Haar features are kind of convolution kernels which primarily detect whether a suitable feature is present on an image or not. Some examples of Haar features are mentioned below:

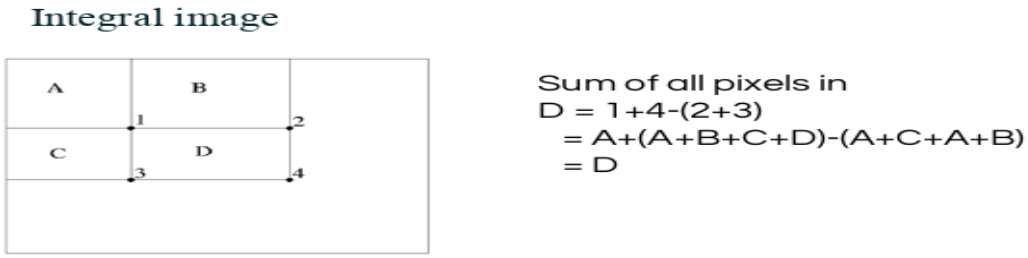


These Haar Features are like windows and are placed upon images to compute a single feature. The feature is essentially a single value obtained by subtracting the sum of the pixels under the white region and that under the black. The process can be easily visualized in the example below.

For demonstration purpose, let's say we are only extracting two features, hence we have only two windows here. The first feature relies on the point that the eye region is darker than the adjacent cheeks and nose region. The second feature focuses on the fact that eyes are kind of darker as compared to the bridge of the nose. Thus, when the feature window moves over the eyes, it will calculate a single value. This value will then be compared to some threshold and if it passes that it will conclude that there is an edge here or some positive feature.

4.2 Creating Integral Images

The algorithm proposed by Viola-Jones uses a 24X24 base window size, and that would result in more than 180,000 features being calculated in this window. Imagine calculating the pixel difference for all the features. The solution devised for this computationally intensive process is to go for the **Integral Image** concept. The integral image means that to find the sum of all pixels under any rectangle, we simply need the four corner values.



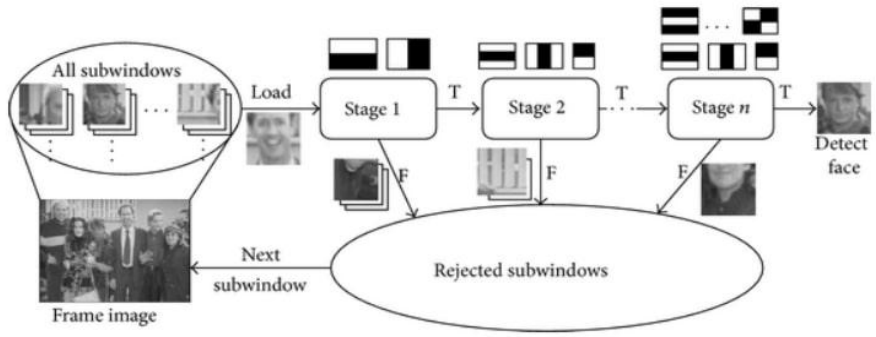
This means, to calculate the sum of pixels in any feature window, we do not need to sum them up individually. All we need is to calculate the integral image using the 4 corner values.

4.3 Adaboost training

As pointed out above, more than 180,000 features values result within a 24X24 window. However, not all features are useful for identifying a face. To only select the best feature out of the entire chunk, a machine learning algorithm called **Adaboost** is used [2]. What it essentially does is that it selects only those features that help to improve the classifier accuracy. It does so by constructing a strong classifier which is a linear combination of several weak classifiers. This reduces the amount of features drastically to around 6000 from around 180,000.

4.4 Cascading Classifier

Another way by which Viola Jones ensured that the algorithm performs fast is by employing a cascade of classifiers [3]. The cascade classifier essentially consists of stages where each stage consists of a strong classifier. This is beneficial since it eliminates the need to apply all features at once on a window. Rather, it groups the features into separate sub-windows and the classifier at each stage determines whether or not the sub-window is a face. In case it is not, the sub-window is discarded along with the features in that window. If the sub-window moves past the classifier, it continues to the next stage where the second stage of features is applied. The process can be understood with the help of the diagram below.



3. CONCLUSION

In this project, we can detect and recognize faces of the criminals in a video stream obtained from a camera in real time. We have used Haar feature-based cascade classifiers in OpenCV approach for face detection. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Also, we have used Local Binary Patterns Histograms (LBPH) for face recognition. Several advantages of this algorithm are: Efficient selection of features, Scale and location invariant detector, instead of scaling the image itself, we scale the features. LBPH recognizer can recognize faces in different lighting conditions with high accuracy. Also, LBPH can recognize efficiently even if single training image is used for each person. The real-time automated face detection and recognition system proposed would be ideal for crowd surveillance applications.

REFERENCES

[1] Viola, P. and Jones, M. Rapid object detection using boosted cascade of simple features. IEEE Conference on Computer Vision and Pattern Recognition, 2001.
[2] Hui-Xing, J., Yu-Jin, Z.: Fast Adaboost Training Algorithm by Dynamic Weight Trimming. Chinese Journal of Computers (2009)
[3] https://en.wikipedia.org/wiki/Cascading_classifiers
[4] https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms
[5] https://opencv.org/
[6] Apoorva.P, Impana.H.C, Siri.S.L, "Automated criminal identification by face recognition using open computer vision classifiers", 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)
[7] http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html
[8] Piyush Kakkar, Vibhor Sharma, "Criminal identification system using Face Detection and Recognition ", IJARCCE, Vol. 7,issue 3, March 2018