

Criminal Identification System Using Face Detection and Recognition

Piyush Kakkar¹, Mr. Vibhor Sharma²

Information Technology Department, Maharaja Agrasen Institute of Technology, Delhi¹

Assistant Professor, Information Technology Department, Maharaja Agrasen Institute of Technology, Delhi²

Abstract: There is an abnormal increase in the crime rate and also the number of criminals is increasing, this leads towards a great concern about the security issues. Crime preventions and criminal identification are the primary issues before the police personnel, since property and lives protection are the basic concerns of the police but to combat the crime, the availability of police personnel is limited. With the advent of security technology, cameras especially CCTV have been installed in many public and private areas to provide surveillance activities. The footage of the CCTV can be used to identify suspects on scene. In this paper, an automated facial recognition system for criminal database was proposed using known Haar feature-based cascade classifier. This system will be able to detect face and recognize face automatically in real time. An accurate location of the face is still a challenging task. Viola-Jones framework has been widely used by researchers in order to detect the location of faces and objects in a given image. Face detection classifiers are shared by public communities, such as OpenCV.

Keywords: Criminal Identification; CCTV; facial recognition; Haar classifier; real-time; Viola-Jones; OpenCV.

I. INTRODUCTION

The face is crucial for human identity. It is the feature which best distinguishes a person. Face recognition is an interesting and challenging problem and impacts important applications in many areas such as identification for law enforcement, authentication for banking and security system access [8], and personal identification among others. Face recognition is an easy task for humans but it's entirely different task for a computer. A very little is known about human recognition to date on How do we analyze an image and how does the brain encode it and Are inner features (eyes, nose, mouth) or outer features (head shape, hairline) used for a successful face recognition? Neurophysiologist David Hubel and Torsten Wiesel has shown that our brain has specialized nerve cells responding to specific local features of a scene, such as lines, edges, angles or movement. Since we don't see the world as scattered pieces, our visual cortex must somehow combine the different sources of information into useful patterns.

Automatic face recognition is all about extracting those meaningful features from an image, putting them into a useful representation and performing some classifications on them. Face recognition based on the geometric features of a face is probably the most instinctive approach for Human identification. The whole process can be divided in three major steps where the first step is to find a good database of faces with multiple images for each individual. The next step is to detect faces in the database images and use them to train the face recognizer and the final step is to test the face recognizer to recognize faces it was trained for.

Nowadays, face detection is used in many places especially the websites hosting images like Picassa, Photobucket and Facebook. The automatically tagging feature adds a new dimension to sharing pictures among the people who are in the picture and also gives the idea to other people about who the person is in the image. In our project, we have studied and implemented a pretty simple but very effective face detection algorithm which takes human skin color into account.

Our aim, which we believe we have reached, was to develop a system that can be used by police or investigation department to recognize criminal from their faces. The method of face recognition used is fast, robust, reasonably simple and accurate with a relatively simple and easy to understand algorithms and technique.

II. FACE DETECTION USING HAAR CLASSIFIER ALGORITHM

The face detection algorithm proposed by Viola and Jones is used as the basis of our design [4]. The face detection algorithm looks for specific Haar features and not pixels of a human face [5]. When one of these features is found, the algorithm allows the face candidate to pass to the next stage of detection. A face candidate is a rectangular section of the original image which is called as a sub-window. Generally, these sub windows have a fixed size (typically 24x24 pixels). This sub-window is often scaled in order to obtain a variety of different size faces. The algorithm scans the entire image with this window and denotes each respective section a face candidate [4].

A. Integral Image

The integral image is defined as the summation of the pixel values of the original image. The value at any location (X, Y) of the integral image is the sum of the image’s pixels above and to the left of location (X, Y). “Figure 1” illustrates the integral image generation.

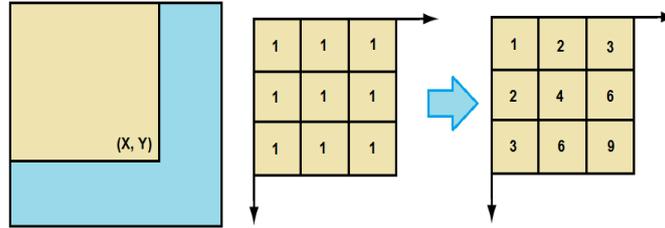


Figure 1: Integral image generation. The dark region represents the sum of the pixels up to position (X, Y) of the original image. It shows a 3×3 image and its corresponding integral image representation.

B. Haar Features

A simple rectangular Haar-like feature can be defined as the difference of the sum of pixels of areas inside the rectangle, which can be at any position and scale within the original image. This modified feature set is called 2-rectangle feature. Viola and Jones also defined 3-rectangle features and 4-rectangle features. Faces are scanned and searched for Haar features of the current stage. The weight and size of each feature and the features themselves are generated using a machine learning algorithm from AdaBoost [4][8]. The weights are constants generated by the learning algorithm. There are a variety of forms of features as seen below in “Figure 3”.

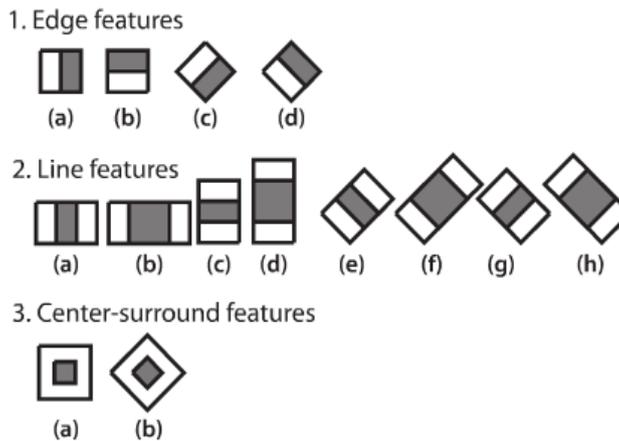


Figure 2: Common Haar Features

Each Haar feature has a value that is calculated by taking the area of each rectangle, multiplying each by their respective weights, and then summing the results [3]. The area of each rectangle is easily found using the integral image. The coordinate of the any corner of a rectangle can be used to get the sum of all the pixels above and to the left of that location using the integral image. By using each corner of a rectangle, the area can be computed quickly as denoted by “Figure 4”. Since A is subtracted off twice it must be added back on to get the correct area of the rectangle. The area of the rectangle R, denoted as the rectangle integral, can be computed as follows using the locations of the integral image: $C + A - B - D$.

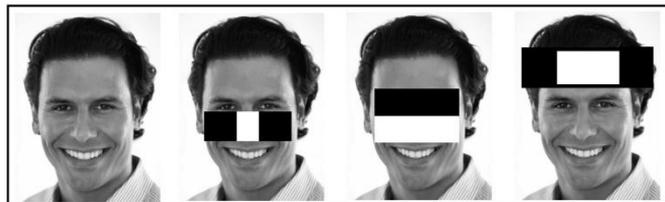


Figure 3: Examples of Haar features. Areas of white and black regions are multiplied by their respective weights and then summed up to get the Haar feature value.

C. Haar Feature Classifier

The cascade classifier contains a list of stages, where each stage consists of a list of weak learners. The system detects the required object by moving a window over the image. Each stage of the classifier labels the specific region defined

by the current location of the window as either positive or negative where positive means that an object was found and negative means that the specified object was not found in the image [6].

If the labelling yields a negative result, then the classification of that particular region is over and the location of the window is moved to the next location. If the labelling gives a positive result, then the region moves to the next stage of classification. The classifier yields a final result as positive, when all the stages, including the last one, yield a positive result, which implies that the required object is found in the image [6].

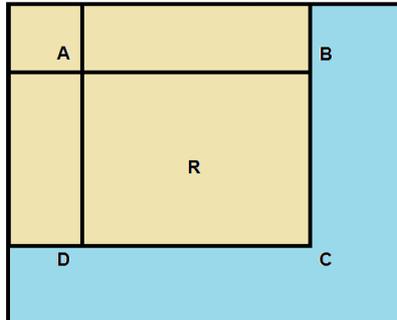


Figure 4: Calculating the area of a rectangle R is done using the corner of the rectangle: $C + A - B - D$.

D. Cascade

The Viola and Jones face detection algorithm eliminates face candidates quickly using a cascade of stages. The cascade eliminates candidates by making stricter requirements in each stage with later stages being much more difficult for a candidate to pass. Candidates exit the cascade if they pass all stages or fail any stage. A face is detected if a candidate passes all stages. This process is shown in “Figure 5”.

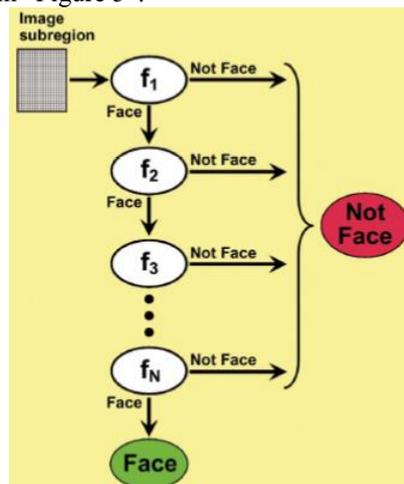


Figure 5: Cascade of stages. The face must pass all stages in the cascade to be conclude.

III. IMPLEMENTATION

A. Import the required modules

The Modules required to perform the facial recognition are cv2, os, image module and numpy. cv2 is the OpenCV module and contains the functions for face detection and recognition. OS will be used to maneuver with image and directory names. First, we use this module to extract the image names in the database directory and then from these names individual number is extracted, which is used as a label for the face in that image. Since, the dataset images are in gif format and as of now, OpenCV does not support gif format, Image module from PIL is used to read the image in grayscale format. Numpy arrays are used to store the images.

B. Load the face detection Cascade

To Load the face detection cascade the first step is to detect the face in each image. 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. haarcascade_frontalface_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 relative path is used.



C. Create the Face Recognizer Object

The next step involves creating the face recognizer object. The face recognizer object has functions like FaceRecognizer.train() to train the recognizer and FaceRecognizer.predict() to recognize a face [13]. OpenCV currently provides Eigenface Recognizer, Fisherface Recognizer and Local Binary Patterns Histograms(LBPH) Face Recognizer. 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. LBP operator is robust against monotonic gray scale transformations.

D. Prepare the training set and Perform the training

To create the function to prepare the training set, we will define a function that takes the absolute path to the image database as input argument and returns tuple of 2 list, one containing the detected faces and the other containing the corresponding label for that face. For example, if the i^{th} index in the list of faces represents the 4th individual in the database, then the corresponding i^{th} location in the list of labels has value equal to 4.

Now to perform the training using the Face Recognizer. Train function. It requires 2 arguments, the features which in this case are the images of faces and the corresponding labels assigned to these faces which in this case are the individual number that we extracted from the image names.

E. Testing

For testing the Face Recognizer, we check if the recognition was correct by seeing the predicted label when we bring the trained face in front of camera. The label is extracted using the os module and the string operations from the name of the sample images folder. Lower is the confidence score better is the prediction.

IV. RESULTS

A. Homepage

Homepage is the main page of Criminal Identification System application. It contains three buttons for: Register Criminal, Detect Criminal and Video Surveillance.



Figure 6: Homepage

B. Criminal Registration

Criminal Registration page will ask the user to select atleast 5 images of the criminal that needs to be registered and also provides input form for providing various details of the criminal like his Name, DOB, Identification mark, Profile picture etc. After selecting images and filling details, user will click register. The criminal will be successfully registered if any error doesn't occur.



Figure 7: Output: Criminal Registration Page

C. Detect Criminal Page

This page allows the user to browse an image from the system and helps in detecting one or more criminals in it. User can also see the profile of the criminal by clicking on detected criminal names.



Figure 8: Output: Detect Criminal Page

D. Criminal Profile Page

This page will show criminal profile after clicking criminal name from detect criminal or video surveillance page.



Figure 9: Output: Criminal Profile Page

E. Detecting Unknown Criminal

Our system is able to identify a non-criminal face.



Figure 10: Output: Non-Criminal Face

F. Video Surveillance

This page will use the pc webcam to capture the video frames in real time. After this it will use face detection module on each frame to detect and recognize criminals in the video in real time. User can also see the profile of the criminal by clicking on detected criminal names.

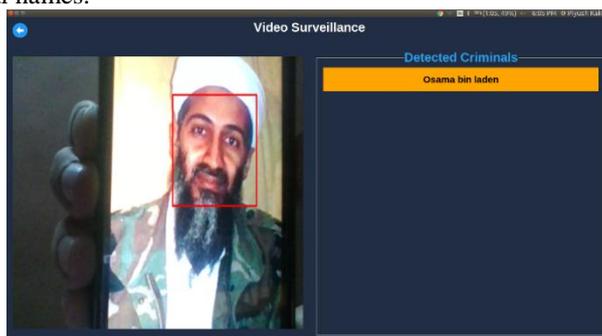


Figure 11: Output: Video Surveillance

V. CONCLUSION

In this project, we are able to detect and recognize faces of the criminals in an image and 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 feature selection, Scale and location invariant detector, instead of scaling the image itself, we scale the features. Such a generic detection scheme can be trained for detection of other types of objects (e.g. cars, sign boards, number plates etc). 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. Our application has some disadvantages like: Detector is most effective only on frontal images of faces, it can hardly cope with 45° face rotation both around the vertical and horizontal axis.

REFERENCES

- [1] Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, pp. 711-720. IEEE Computer Society
- [2] Borneo, O. (2005, May 19). Learning Based Computer Vision with Intel's Open Source Computer Vision Library. Retrieved April 2007, 2007, from Intel.com Website: http://www.intel.com/technology/itj/2005/volume09issue02/art03_learning_vision/p04_face_detection.htm
- [3] Brunelli, R., & Poggio, T. (1993). Face Recognition: Features versus templates. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 15 (10), 1042-1052.
- [4] Viola, P. and Jones, M. Rapid object detection using boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.
- [5] P. Viola and M. Jones. Robust Real-time Object Detection. *International Journal of Computer Vision*, 57(2):137–154,2002.
- [6] https://en.wikipedia.org/wiki/Cascading_classifiers
- [7] Open Computer Vision Library Reference Manual. Intel Corporation, USA, 2001.
- [8] Hui-Xing, J., Yu-Jin, Z.: Fast Adaboost Training Algorithm by Dynamic Weight Trimming. *Chinese Journal of Computers* (2009)
- [9] Bureau of Justice Statistics, U.S. Department of Justice, April 1990), pp. 43-66; SEARCH Group, "Legal and Policy Issues Relating to Biometric Identification Technologies".
- [10] J. Cox, J. Ghosn, and P.N. Yianilos, "Feature-Based Face Recognition Using Mixture Distance," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1996, pp. 209-216.
- [11] G. Wang, H. Chen, and H. Atabakhsh, "Automatically Detecting Deceptive Criminal Identities," *Comm. ACM*, Mar. 2004, pp. 70-76.
- [12] Cristinacce, D. and Cootes, T. Facial feature detection using AdaBoost with shape constraints. *British Machine Vision Conference*, 2003.
- [13] https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html#local-binary-patterns-histograms
- [14] http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html
- [15] <https://aip.scitation.org/doi/pdf/10.1063/1.5005335>