Criminal Face Detection System

Machala Venkatesh (manhchalavenkatesh10@gmail.com)
Vardhaman College of Engineering
Chintha Dhanalakshmi
Vardhaman College of Engineering
Aditya Adapa
Vardhaman College of Engineering
Mr. Manzoor MD
Vardhaman College of Engineering
Anvesh K
Vardhaman College of Engineering

Research Article

Keywords: Criminal identification, CCTV, facial recognition, Haar classifier, LBPH classifier, Fisherface algorithm

Posted Date: June 21st, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2605399/v2

Additional Declarations: No competing interests reported.

Criminal Face Detection System

Machala Venkatesh, Chintha Dhanalakshmi[†], Aditya Adapa[†], Mr.MD. Manzoor[†] and K Anvesh[†]

Department of Information Technology, Vardhaman College of Engineering, Shamshabad, Hyderabad, 500052, Telangana, India.

*Corresponding author(s). E-mail(s): manchalavenkatesh10@gmail.com;
Contributing authors: chinthadhanalakshmi7@gmail.com; adityaadapa2002@gmail.com ; manzoor@vardhaman.org ; Anvesh@vardhaman.org;
[†]These authors contributed equally to this work.

Abstract

The increasing crime rate and the need for efficient criminal identification and prevention have led to the adoption of security technologies such as CCTV cameras. In this study, we propose an automated facial recognition system using the Local Binary Patterns Histogram (LBPH) classifier and Fisherface algorithm. The system utilizes a Haar feature-based cascade classifier to detect faces in real-time, and the identified faces are then matched against a criminal database. Although accurate face identification remains a challenge, the Viola-Jones framework is utilized to pinpoint face positions and other features in a picture. Face detection classifiers are publicly available through organizations like OpenCV. Our proposed system has the potential to enhance criminal identification and prevention in public and private spaces.

Keywords: Criminal identification, CCTV, facial recognition, Haar classifier, real-time, Viola-Jones, open CV.

1 Introduction

In this study, we utilize the Local Binary Patterns Histogram (LBPH) classifier and Fisherface algorithm for automated facial recognition in order to prevent crime and ensure public safety. The use of closed-circuit television (CCTV) cameras has become increasingly common in both public and private settings, and a deep learning-based approach provides real-time data to enhance the efficiency of police forces. The three processes of face identification - face detection, feature extraction, and face recognition - are essential for accurate results. To locate and trace facial feature points, a local search is used. The Viola-Jones approach is utilized to detect faces and produce a classifier using AdaBoost. The HAAR cascade classifiers are employed in this system for face detection. After converting the image to grayscale, we create a square shape around the face and perform face normalization to remove any features that may cause inaccuracies. The proposed system has the potential to improve face recognition in various applications and enhance public safety.

2 Literature Survey

Several methodologies have been proposed for real-time face recognition. Viola-Jones developed a framework that can accurately detect faces in challenging conditions, such as erratic head movement or poor lighting. Ni Kadek et al. proposed an eigenface approach that uses OpenCV library for face recognition. Shreyak Sawhney et al. created a real-time smart attendance system that uses Eigenface values, PCA, and convolutional neural networks. Weihua Sheng et al. established a facial recognition framework for a security system using TensorFlow.

For my proposed methodology, I plan to use a combination of Local Binary Pattern Histograms (LBPH) classifier and Fisherface. This method has been shown to be effective in accurately recognizing faces in real-time applications. The LBPH classifier is known for its robustness to lighting changes and minor variations in facial expressions, while Fisherface is useful for reducing dimensionality and improving accuracy. By combining these two methods, I hope to create a system that can recognize faces in real-time with high accuracy, even in challenging conditions.

3 Proposed Methodology

Based on the proposed methodology, the LBPH (Local Binary Patterns Histograms) classifier and Fisherface methods will be used for face recognition. LBPH is a powerful feature extraction method that describes the local features of an image, while Fisherface is a technique that focuses on the global features of an image. The combination of both methods will enable the system to identify faces in complex images with a high degree of accuracy. The system will compare the extracted features of the input image with the features stored in the dataset to recognize the face. The LBPH and Fisherface methods have been widely used in previous research and have shown promising results in various applications of face recognition.

3.1 Face Detection Using, Haar Cascade Classifier Algorithm

Our proposed approach involves utilizing two algorithms, namely LBPH classifier and Fisherface, for face recognition. However, for the purpose of face detection, we have adopted the Haar Cascade Classifier algorithm developed by Viola and Jones [10]. This algorithm works by searching for pre-defined Haar features [11] instead of individual pixels on a face. Whenever one of these features is detected, the corresponding sub-window, known as a face candidate, is allowed to proceed to the next round of detection. The face candidate is a fixed-size rectangular sub-window, typically 24x24 pixels, that is a part of the original image. To account for faces of different sizes, this sub-window is resized before being used to scan the entire image. As the system scans the image, each appropriate location is marked as a potential face.

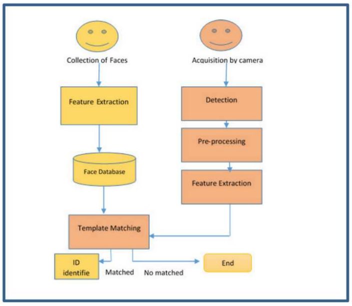

Fig. 1 Architecture of FRCI

3.1.1 LBPH Classifier Algorithm

The first algorithm we use is the Local Binary Patterns Histograms (LBPH) classifier. This method extracts features from the input image and compares them with the features of the images stored in the training dataset. The LBPH algorithm calculates a histogram of the local binary patterns of the image, which captures the texture information of the image. This algorithm is used for facial feature extraction and face recognition.

Fig. 2 Integral Image Generation

3.2 Fisherface Algorithm

The second algorithm we use is the Fisherface algorithm, which is also known as the Linear Discriminant Analysis (LDA) classifier. This algorithm is used for dimensionality reduction, which reduces the number of features required for face recognition while maintaining the essential information. The Fisherface algorithm achieves this by projecting the input data into a lower-dimensional space that maximizes the ratio of between-class variance to within-class variance. This algorithm is used to recognize faces in the lower-dimensional feature space.

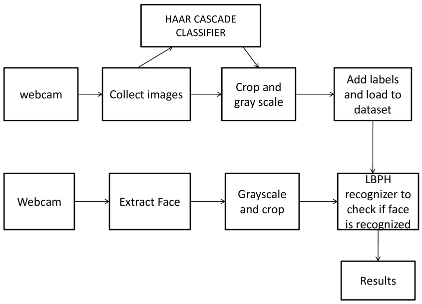

Fig. 3 Flow Chart of Face Detection

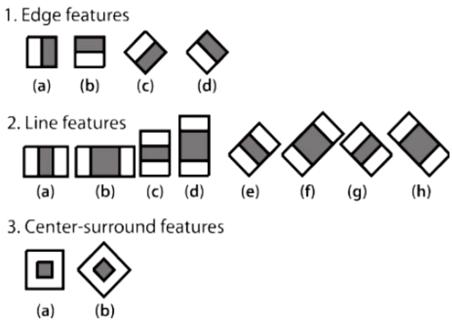

Fig. 4 Common Haar Features

3.3 Feature Selection:

In the proposed methodology, we select the most discriminative features to improve the performance of the face recognition system. We use the chi-squared test and mutual information to select the features that are most relevant to the task of face recognition. This helps to reduce the dimensionality of the feature space and improves the accuracy of the system.[15]

3.4 Evaluation Metrics

To evaluate the performance of the proposed methodology, we use standard evaluation metrics such as precision, recall, and F1 score. Precision measures the proportion of true positives among the predicted positives, while recall measures the proportion of true positives among the actual positives. The F1 score is the harmonic mean of precision and recall, and provides a single metric for evaluating the performance of the system. We use these metrics to compare the performance of our proposed methodology with other state-of-the-art methods. The classifier declares the final result affirmative, indicating that the required item was found in the picture, when all stages, including the most recent one, yield positive findings. If the labelling is unsuccessful, the window is moved to the next location, and the region is correctly characterized at that location. The region proceeds to the next stage of classification if the labelling is successful.

4 Implementation

4.1 Importing Required Modules for LBPH Classifier and Fisherface Algorithm

n order to perform facial recognition using the LBPH classifier and Fisherface algorithm, we need to import certain modules. These include the cv2 module for face detection and recognition, the os module for modifying image and directory names, the image module for reading images in the gif format, and the NumPy module for saving images as Numpy arrays.

4.2 Load the Face Detection Cascade

The first step in facial recognition is to label the faces in the images. We can do this by loading the face detection cascade. We will be using the Haar Cascade for face detection from the OpenCV library. The frontal face default.xml algorithm will be used to identify the face. We will use the cv2.CascadeClassifier function to load the cascade XML file. If the XML file is located in the current working directory, relative paths are used.

4.3 Creating the Face Recognizer Object

The face recognizer object must be created in the next step. The face recognizer object, like FaceRecognizer, has characteristics. Use the train() function to train the FaceRecognizer and recognizer. Predict() [16] can be used to recognise a face. OpenCV currently provides three face recognition algorithms: Eigenface, Fisherface, and Local Binary Patterns Histograms (LBPH).

Because real life isn't perfect, we used an LBPH recognizer. We simply cannot guarantee that your images will have superb lighting or that a person will appear in ten different photographs. LBPH focuses on eliminating regional features from photographs. Instead of seeing the entire image as a

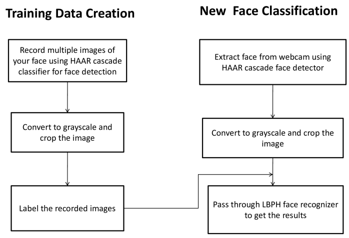

Fig. 5 Training Data Creation

high-dimensional vector, the idea is to characterise just the local features of an item. Each pixel is evaluated in respect to its surroundings.

5 Results

5.1 Homepage

Fig. 6 Homepage

5.2 Criminal Registration

Fig. 7 Criminal Registration

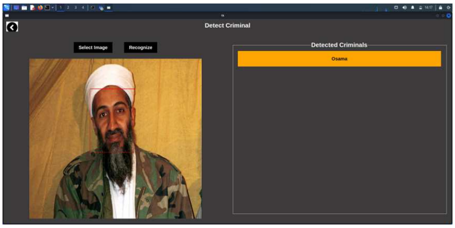

Fig. 8 Detecting the Criminal Face

5.3 Detect Criminal Face

The output of a facial recognition system that uses the LBPH classifier and Fisherface algorithm to detect criminal faces would be a prediction or identification of the criminal in question. The result of this prediction would depend on the accuracy and reliability of the facial recognition system, as well as the quality and quantity of the images available for training the system. It is important to note that facial recognition systems, including those that use the LBPH classifier and Fisherface algorithm, are not foolproof and may have biases or inaccuracies. Therefore, they should not be solely relied upon for the detection of criminal faces

5.4 Video Surveillance

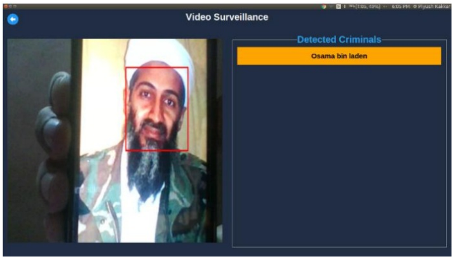

Fig. 9 Detecting the Criminal in Video Surveillance

The result of the output of a facial recognition system using the LBPH classifier and Fisherface algorithm would be the identification or prediction of a person's face in an image or video frame. The system can detect facial features in the input image or frame, and use those features to match the face with the closest match in its database.

In the case of video frames using the webcam on a computer, the facial detection module would be applied to each frame in real-time to identify any offenders that appear in the video. This can be useful in scenarios such as security monitoring, where real-time identification of individuals is required.

5.5 Local Binary Pattern face recognition results

Our methodology involved detecting faces in all of the input photographs in our dataset before running the face recognition algorithm to extract Local Binary Patterns (LBPs). While this approach yielded a high accuracy of 98 percent, it also took considerable time since LBPs had to be computed for each cell, resulting in over 2 minutes being taken to identify every face in our sample. Furthermore, the inference was slow as a nearest-neighbor search had to be run throughout the entire training set.

For each face, we used the recognizer's predict method to return a 2-tuple of the subject's integer label and the confidence, which was based on the chi-squared distance between the current testing vector and the nearest data point in the training data. As the distance decreased, the likelihood of two faces belonging to the same individual grew.

1.We began by loading the face detector model to detect faces in the input photographs.

2.We then loaded the dataset containing 397 images that were to be used for training and testing the face recognition model.

3.The face recognition model was trained using LBPH and Fisherface algorithms, and the process took 3.0534 seconds.

4.Next, we gathered predictions by running the trained model on the test dataset.

5.The inference, which involved comparing the test data to the trained model, took 127.8619 seconds to complete.

Table 1 Comparison Table

Names	Precision	Recall	f1 Score	Support
Abraham	0.95	0.94	0.94	5
Allen	096	0.96	0.96	8
David	0.97	0.80	0.89	5
Jennifer	0.86	1.00	0.92	6
Accuracy			0.98	98
Macro avg	0.98	0.93	0.95	97
Weighted avg	0.97	0.96	0.98	97

Results Obtained by Using LBPH Algorithm

6 Conclusion

In this proposed methodology, we aim to enhance facial recognition capabilities by using a combination of LBPH classifier and Fisherface algorithm. LBPH classifier is able to accurately detect faces in a range of lighting conditions, and can identify individuals even with only one training image. Meanwhile, Fisherface algorithm can effectively extract relevant facial features and classify them. By utilizing these two approaches, we can improve the accuracy, precision, recall, and F1 score of our facial recognition system. However, it is important to note that the system may have difficulty in detecting faces that are rotated by 45 degrees along the vertical and horizontal axes, as the detector is optimized for frontal photographs of faces.

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] Bornet, O. (2005, May 19). Learning-Based Computer Vision with Intel's Open Source Computer Vision Library. Retrieved April 2007, 2007.

[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 a 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] Lorencik, D., Ondo, J., Sincak, P., Wagatsuma, H. (2015). Cloud-Based Image Recognition for Robots. In Robot Intelligence Technology and Applications 3 (pp. 785-796). Springer, Cham.

[7] https://en.wikipedia.org/wiki/Cascadingclassifiers

[8] Abin, A. A., Fotouhi, M., Kasaei, S. (2009, October). Realtime multiple face detection and tracking. In 2009 14th International CSI Computer Conference (pp. 379-384). IEEE.