VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590018, Karnataka

DST Report

on

"CFIS-CRIMINAL FACE IDENTIFICATION SYSTEM"

Submitted in partial fulfillment of the requirements for the award Bachelor Of Engineering Degree

of

CSE-Artificial Intelligence & Machine Learning

Submitted By

DHANYASHREE.M

4AD21CI020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(Artificial Intelligence & Machine Learning)

13th Kilometer, Mysore – Kanakapura – Bangalore Road,

Mysore – 570 028, Karnataka

Phone: +91-821-25 93 335

TABLE OF CONTENTS

INTRODUCTION

Criminal Face Detection project aims to build a automated Criminal Face Detection system by levering the human ability to recall minute facial details. Identification of criminals at the scene of a crime can be achieved in many ways like fingerprinting, DNA matching or eye witness accounts. Out of these methods eye witness accounts are preferred because it stands scrutiny in court and it is a cost effective method. It is possible that witnesses to a crime have seen the criminal though in most cases it may not be possible to completely see the face of the perpetrator. The Criminal Face Detection System will be built of an existing criminal database. Input would be provided in the form of sketch or an image and matched against the existing database and results would be provided. Criminal record generally contains personal information about particular person along with photograph. This project is aimed to identify the criminals in any investigation department. Perhaps one of the most well-known applications of facial recognition technology is law enforcement, where we can use it to find missing people, aid in solving crimes and help monitor large crowds of people.

The idea of the project has been taken from the TV Series named PERSON OF INTEREST which is an American science fiction crime drama television series in which a computer system were developed for the U.S. government named "THE MACHINE" that is capable of collating all sources of information to predict and identify— in advance—people planning terrorist acts. As an artificial super intelligence, its objective is to predict and prevent imminent terrorist attacks and does so by analysing immense amounts of surveillance data.

It uses various machine learning techniques to determine the identity, location, and intentions of monitored persons by infiltrating domestic organizations such as the National Security Agency and foreign agencies including Interpol ("No Good Deed") to analyse their databases and data from various sources such as video footage, phone calls (landline, VOIP, mobile), GPS, electronic transactions, e-mail, social media, etc. The Machine was created with the ability to simulate the outcomes of different scenarios to aid it in making choices and to better fulfill its purpose. Finch taught it to how to play chess and the importance of making good decisions. It can evaluate the outcomes of different strategies by way of simulating them.

Nowadays criminal activities are growing at an exponential rate. Crime prevention by effective identification of criminals is the main issue before the police and on the other hand, the availability of police officers is inadequate. There are various technological solutions for detecting criminals but they are not up to the mark. In this project, a face detection and recognition system for criminal identification is developed using the multi-task cascade neural network. This system will be able to detect faces and recognize faces of criminals automatically in real-time. This system would also just require a single image of the criminal to recognize him, also known as one-shot learning. The purpose is to identify the criminal face, retrieve the information stored in the database for the identified criminal and a notification is sent to the police personnel with all the details and the location at which he was under the surveillance the camera.

SOFTWARE AND HARDWARE REQUIREMENTS

SOFTWARE REQUIREMENTS

- PC, Mac or laptop with **x86-64 (64-bit)** compatible processors.
    - 2 GHz or better processor is recommended.
    - **AVX2 support is highly recommended.** Processors that do not support AVX2 will still run Neurotechnology Face Verification algorithm, but in a mode, which will not provide the specified performance. Most modern processors support this instruction set, but please check if a particular processor model supports it.
- At least **512 MB of free RAM** should be available for the application.
- A **camera or webcam** which is accessible using:
    - **DirectShow** interface on Microsoft Windows.
    - **GStreamer** interface on macOS or Linux.
- Internet connection required for managing the Face Verification transaction licenses
- **Microsoft Windows specific requirements:**
    - Microsoft Windows **7 / 8 / 10 / 11**.
    - Microsoft **.NET framework 4.5** or newer (for .NET components usage).
    - One of following **development environments** for application development:
        - Microsoft Visual Studio 2012 or newer (for application development under C/C++, C#, Visual Basic .Net)
        - Java SE JDK 8 or newer.

HARDWARE REQUIREMENTS

- **macOS specific requirements:**
    - **macOS 10.13** or newer.
    - XCode 9.3 or newer (for application development)
    - GStreamer 1.10.x or newer with gst-plugin-base and gst-plugin-good is required for face capture using camera/webcam or rtsp video (for application development)
    - GNU Make 3.81 or newer (to build samples and tutorials development)
    - Java SE JDK 8 or newer (for application development)
- **Linux specific requirements:**
    - **Linux 4.9 kernel** or newer
    - glibc 2.24 library or newer
    - GStreamer 1.10.x or newer with gst-plugin-base and gst-plugin-good is required for face capture using camera/webcam or rtsp video
    - gcc 6.3 or newer (for application development)
    - GNU Make 3.81 or newer (for application development)
    - Java SE JDK 8 or newer (for application development)

PROJECT EXPLANATION

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 situations.

Criminal Identification for Low Resolution Surveillance S. P. Patil1 has proposed a model that uses the Tiny Face Detector for face recognition, which is a mobile and web-friendly model and is a prominent real-time face detector. It detects faces and facial landmarks on images or frames, takes a person's face as input, and gives a vector called embedding of 128 numbers as output that represents the most important features of the face. The model is trained on a dataset of photos of various criminals, and after training, live surveillance video feed can be given as input to identify any criminal. The frames undergo feature extraction of the detected faces, and the embeddings are compared to the features of the images from the dataset to find a match, which is judged using a threshold value. The recognized images are saved in PNG format in a folder, which can be accessed through the portal developed using the Django framework that allows the admin to view the results Criminal identification system using deep learning The criminal face identification system proposed by D.Nagamallika1 uses MTCNN, FaceNet, and OpenCV for detecting faces and facial landmarks, face embedding, and video/image processing. The system registers a new criminal and performs pre-processing, feature extraction, and matching to identify criminals from images/videos. If the person is a criminal, the system sends a notification via SMS. Overall, the system is effective in identifying criminals, and its accuracy depends on preprocessing and feature extraction.

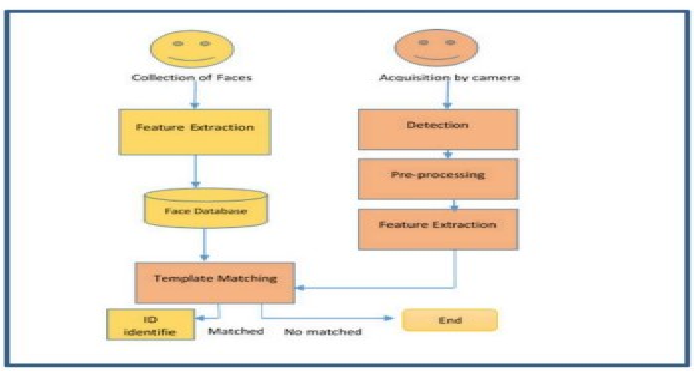
Real-Time Object Detection System with Multi-Path Neural Networks Identifying where an object exists and what the object is using DNN(Deep neural network) which provides superior accuracy over traditional algorithms. Real-time object detection systems need to satisfy environment-varying time constraints; however, the existing object detection systems which employ DNNs to achieve high accuracy fail to satisfy the real-time requirements detailed experiments using widely used driving datasets clearly show that the system successfully satisfies any time limit while maintaining high accuracy.

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.

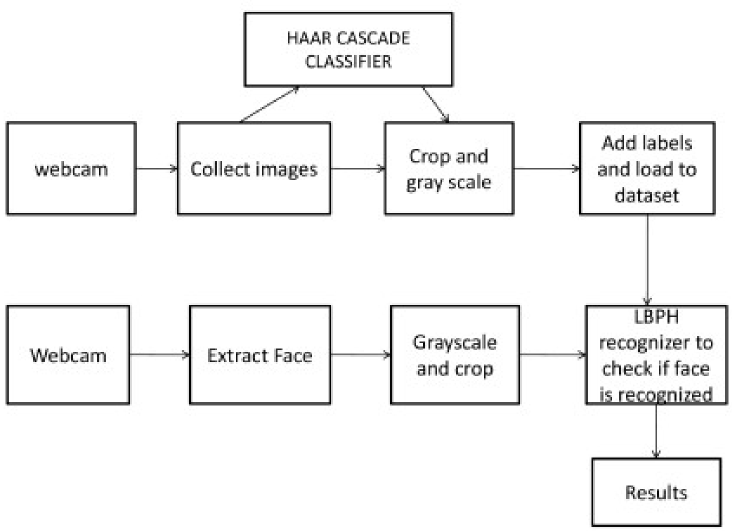


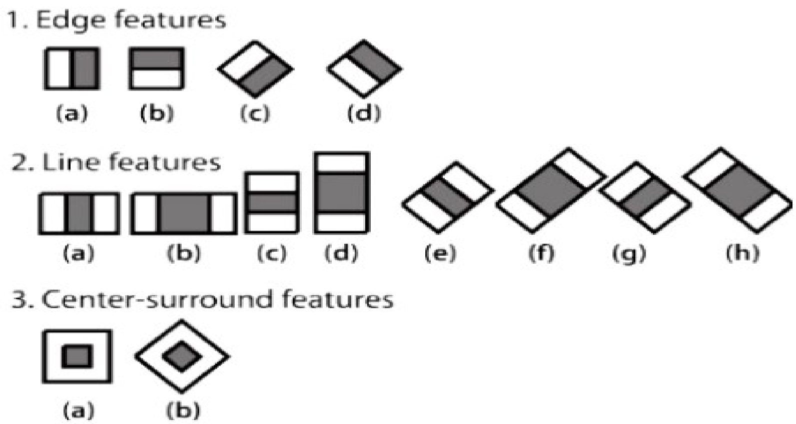
3.1.1 LBPH Classifier Algorithm The first algorithm

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.

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.





3.3 Feature Selection

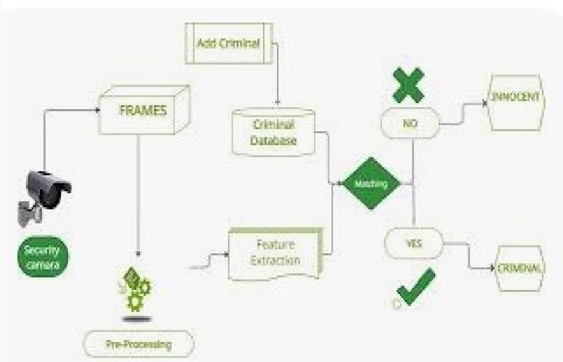
In the proposed methodology, we select the most discriminative features to improve the performance of the face recognition system. We use the chi-squared test and mutual information to select the features that are most relevant to the task of face recognition. This helps to reduce the dimensionality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

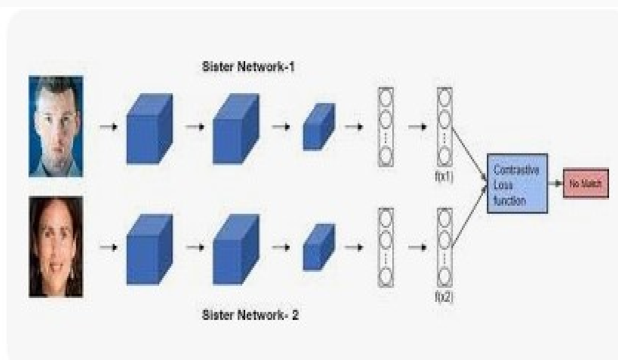
3.4 Evaluation Metrics

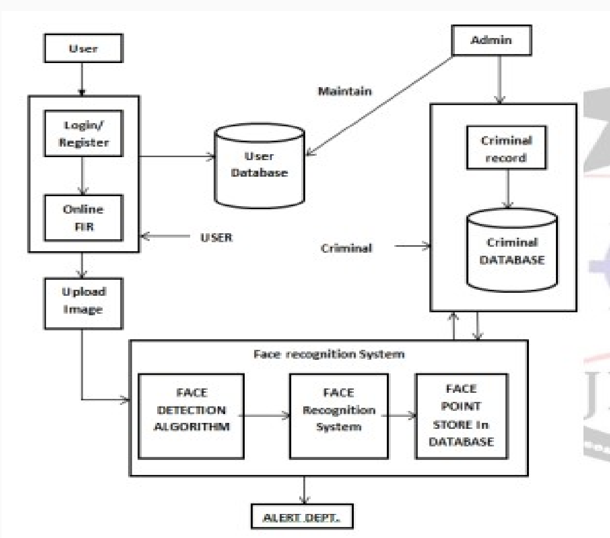
To evaluate the performance of the proposed methodology, we use standard evaluation metrics such as precision, recall, and F1 score. Precision measures the proportion of true positives among the predicted positives, while recall measures the proportion of true positives among the actual positives. The 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-ofthe-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.

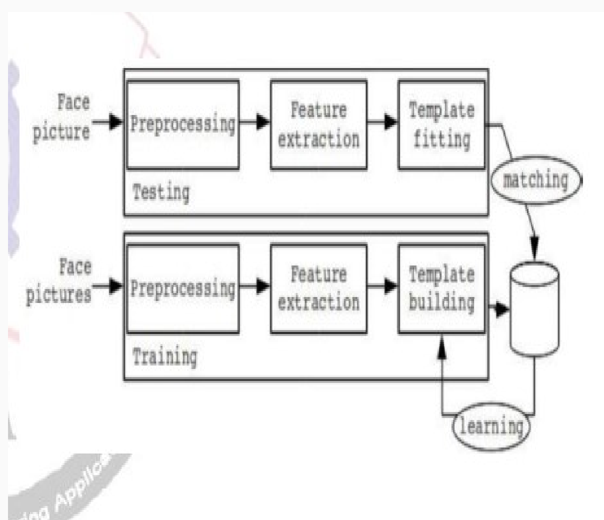
DIAGRAMS


Criminal Identification System
Register Criminal
Detect Criminal
Video Surveillance


Add Criminal
FRAMES
Criminal Database
NO
INNOCENT
Matching
Security camera
Feature Extraction
YES
CRIMINAL
Pre-Processing


Sister Network-1
f(x1)
Contrastive Loss function
No Match
f(x2)
Sister Network- 2


User
Admin
Maintain
Login/ Register
User Database
Criminal record
Online FIR
USER
Criminal
Criminal DATABASE
Upload Image
Face recognition System
FACE DETECTION ALGORITHM
FACE Recognition System
FACE POINT STORE In DATABASE
ALERT DEPT.


Face picture
Preprocessing
Feature extraction
Template fitting
Testing
matching
Face pictures
Preprocessing
Feature extraction
Template building
Training
learning

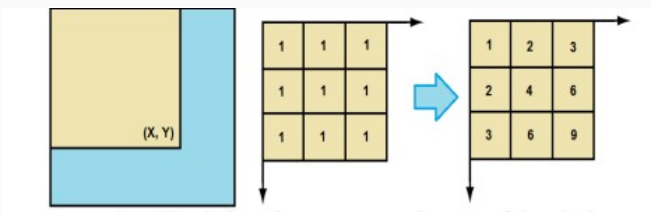
ALGORITHM

FACE DETECTION USING HAAR CLASSIFIER ALGORITHM

The face detection algorithm proposed by Viola and Jones is used as the basis of our design. The face detection algorithm looks for specific Haar features and not pixels of a human face. 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 24×24 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.

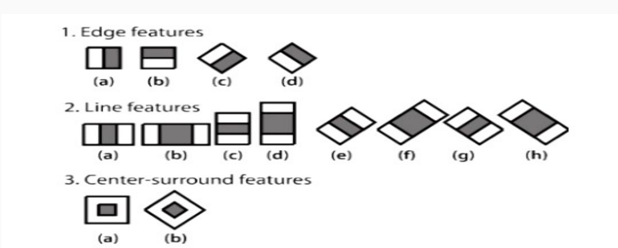
1.Integral Image

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).



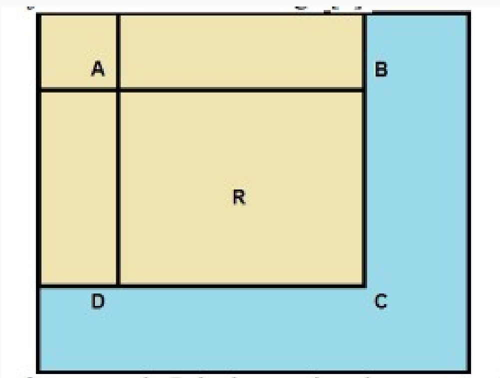
2. Haar Features

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.



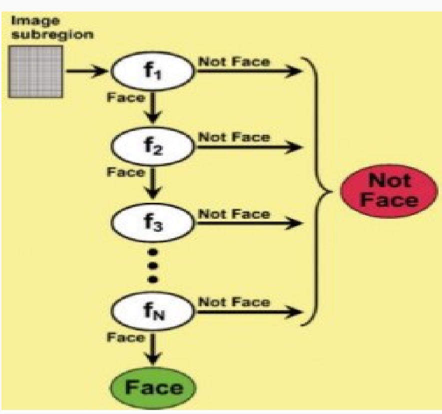
3. Haar Feature Classifier

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.



4. 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.



WORKFLOW

1.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.

2. 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.

3. 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.

4. 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 ith index in the list of faces represents the 4th individual in the database, then the corresponding ith 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.

5. 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.

FUTURE ENHANCEMENT AND REFERENCES

FUTURE ENHANCEMENT

- **Integration with surveillance systems:** The system can be integrated with surveillance systems allowing law enforcement to monitor public areas and identify criminals in real time.

- **Deployment in other fields:** The technology used in the proposed system can be applied in other fields such as healthcare, finance and retail, for applications such as facial recognition based access control and customer identification.

- **Multi modal identification:** The system can be extended to perform multi-modal identification by integrating other biometric data such as fingerprints, voice and iris scans to enhance identification accuracy.

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 tem plates. 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 multi ple face detection and tracking. In 2009 14th International CSI Computer Conference (pp. 379-384). IEEE.

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 clas sify them. By utilizing these two approaches, we can improve the accuracyprecision, 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.