Face Detection and Tracking to Find the Missing Person

Roshin John[1*], Basil Kuriakose[2]
[1]*P.G. Student, Department of Electronics and Communication Engineering, Indira Gandhi Institute of Engineering and Technology, Nellikuzhy, India*
[2]*Associate Professor, Department of Electronics and Communication Engineering, Indira Gandhi Institute of Engineering and Technology, Nellikuzhy, India*
*Corresponding author: johnroshin@gmail.com

***Abstract*: This project proposes a system that would help the police and the public by accelerating the process of searching using face recognition. When a person goes missing, the people related to that person or the police can upload the picture of the person which will get stored in the database. When the public encounter a suspicious person, they can capture and upload the image of that person into our portal. The face recognition model in our system will try to find a match in the database with the help of face encodings. In our system, the image of the person given by the guardian at the time of missing is stored in the database. The captured image or video from various CCTV is compared with this and identifies the missing person. This project also support blind people, when a known persons come in front of the CCTV the system will give a voice message to them.**

***Keywords*: Track the missing, face detection, Haarcascade algorithm, openCV, Python.**

1. Introduction

In India a countless number of persons are reported missing every year. The deep learning methodology can be used for identifying the reported missing person. Face recognition can be used to find missing children and victims of human trafficking.

In our proposed system we will detect and recognize the faces by using Eigen object detector algorithm. This can be done with the help of OpenCV with haar cascades which are present in the OpenCV integral. We will design a face detection system for the students with the help of this OpenCV, for this the system will need a HD webcam to take the input images in a fixed area where the camera is located. The images which are taken from the camera are detected with haarcascade frontal faces and eyes then trained with Eigen algorithm, the trained faces are kept in a database first and equated to the trained images (the trained images are initially present in the database means the related persons) after comparing it will make a log of the system to the recognized persons. The system also produces a voice output when a person is detected. The program coding is done in Python software.

2. Algorithms

A. *Real-time face detection*

Real-time face detection contains detection of a face from a sequence of frames from a video- capturing device. While the hardware necessities for such a system are far more severe, from a computer vision stand point, real-time face detection is actually a far greener process than detecting a face in a stationary image. This is because unlike most of our surrounding environment, people are continually moving.

B. *Role of Haarcascade algorithm*

Very often in crowded scenes, people are only partially visible to the camera. Hence approaches, that attempt to detect full body, fail in most cases. We have adopted an method that uses the coarse-to-fine policy to divide the entire body space into smaller and smaller subspaces to block this problem. We learn several body poses of humans using AdaBoost separately, and obtain detectors for each of these body poses. Annotated data of these body poses are served, during training, separately to AdaBoost algorithm that use Haar features to generate reliable classifiers for the corresponding body poses. Throughout the training phase, we typically tune each classifier to obtain a high detection rate even at the cost of a higher false alarm rate. However, the proposed algorithm is able to reliably detect persons and reject false alarms despite the higher false alarm rates of the early classifiers.

C. *AdaBoost*

AdaBoost, short for Adaptive Boosting, is a machine learning algorithm formulated by Y. Freund and R. Schapire. It can be used in grouping with many other learning algorithms to improve their performance. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in help of those instances misclassified by previous classifiers. AdaBoost is sensitive to noisy data and outliers. Otherwise, it is less prone to the over fitting problem than most learning algorithm.

A variant of AdaBoost is used both to select a small set of features and train the classifier. In its original form, the AdaBoost learning algorithm is used to boost the classification performance of a simple (sometimes called weak) learning algorithm.

The key indication of boosting algorithms is to chain many simple and moderately accurate hypotheses (called weak classifiers) into a sole, extremely accurate classifier for the task at hand. The weak classifiers are trained sequentially and, conceptually, each of them is trained mostly on the examples which were most difficult to classify, by the previous weak classifiers.

D. *Implementation*

Phase 1: Training

Our first step is to detect human face from crowd using Haar cascade algorithm. The cascade classifier used in the project is a sliding window type binary detector that uses Haar-like features for weak classifiers (Viola, 2001). The basic idea here is that a small subsection of an image, the sliding window, is inspected using very simple weak classifiers which are combined first for cascade stages and the stages further for a strong binary classifier. Each cascading stage may reject the inspection window (Figure 1). Early rejection leads to improved detection speed as only a minority of features is tested. On the other hand, each passed stage increases the probability that the window contains positive detection.

Phase II: Testing

The testing phases consist of the same process as in the training phase. The difference is, here we check whether the input images are identified accurately. For example, if the first person is shown in front of the camera the detector detects the person and tells his name. Also it stores the time and name of the person in a text document.

Phase III: Results

Following figure shows the detection of a person. The system detects the person and his face. The system identifies his face by rectangular window and his name is marked in that window.

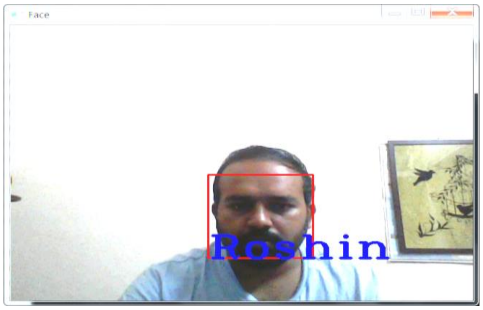
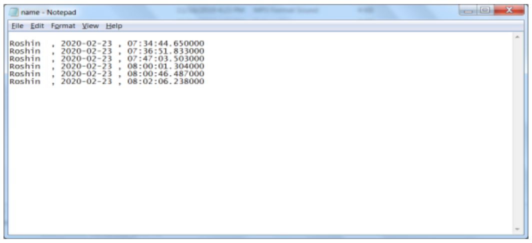


Fig. 1. Detecting the person



```
Roshin  , 2020-02-23 , 07:34:44.650000
Roshin  , 2020-02-23 , 07:36:51.833000
Roshin  , 2020-02-23 , 07:47:03.503000
Roshin  , 2020-02-23 , 08:00:01.304000
Roshin  , 2020-02-23 , 08:00:46.487000
Roshin  , 2020-02-23 , 08:02:06.238000
```

Fig. 2. The database created in text document

The text document stores the details of the detected person. This data can be shared to authorities and parents when a missing person is identified.

The system produces a voice output too. This will help the blind people to detect the persons in from there CCTV.

3. Application

- Enhanced security. The first thing to start with is surveillance
- Automation of identification in ATMS
- For online banking transactions.

4. Future Scope

- Health
- Security
- Retail, Marketing, and Advertising
- Banking

5. Conclusion

This paper presented an overview on face detection and tracking to find the missing person.

References

[1] Adelson, E. H., and Bergen, J. R. (1986) The Extraction of Spatio-Temporal Energy in Human and Machine Vision, Proceedings of Workshop on Motion: Representation and Analysis (pp. 151-155) Charleston, SC; May 7-9

[2] AAFPRS (1997). A newsletter from the American Academy of Facial Plastic and Reconstructive Surgery. Third Quarter 1997, Vol. 11, No. 3. Page 3.

[3] Baron, R. J. (1981). Mechanisms of human facial recognition. International Journal of Man Machine Studies, 15:137-178

[4] Beymer, D. and Poggio, T. (1995) Face Recognition from One Example View, A.I. Memo No. 1536, C.B.C.L. Paper No. 121. MIT

[5] Bichsel, M. (1991). Strategies of Robust Objects Recognition for Automatic Identification of Human Faces. PhD thesis, Eidgenossischen Technischen Hochschule, Zurich.

[6] Brennan, S. E. (1982) the caricature generator. M.S. Thesis. MIT.