

Real Time Face Recognition using Raspberry Pi

Sanchit Dass

Department of Computer Science,
BMS College of Engineering,
Bangalore, India

Mohammed Sadruhlhuda Quadri

Department of Computer Science,
BMS College of Engineering,
Bangalore, India

Navaz Pasha

Department of Computer Science,
BMS College of Engineering,
Bangalore, India

Nishant Nayan

Department of Computer Science,
BMS College of Engineering,
Bangalore, India

Jyothi S Nayak

Department of Computer Science,
BMS College of Engineering,
Bangalore, India

ABSTRACT

Face recognition is a fast growing and challenging area in the field of computer vision and real time applications. A lot of techniques and algorithms are available with varying degrees of accuracy and speed. Face recognition has a lot of applications in the field of advertising, healthcare, security, accessibility, and even payments. Hence, there is a need for low cost, reliable and accurate face recognition systems in today's world [3]. The aim is to implement a face recognition system using a Raspberry Pi device. This system is part of an assistive device created by us for visually impaired people. The setup consists of a Raspberry Pi 3 Model B device with a camera module attached to it. The Raspberry Pi has a 1.2 GHz 64-bit CPU along with 1 GB RAM and the camera module has a resolution of 5 MP.

General Terms

Face Recognition

Keywords

Face Recognition, Raspberry Pi, Computer Vision, LBPH, Haar Cascade, Real Time

1. INTRODUCTION

Face recognition is a technique used to identify a person from an image or a video feed. It can emulate the ability of a human eye to recognize people. Face recognition has many applications in the real world [7]. There are multiple methods available to implement this technology. The first step in any face recognition system is to detect the face (from the source). After a face has been detected, certain information is extracted from the detected face and compared to a known database to identify the person [3].

For the system, a Raspberry Pi 3 Model B has been used along with a camera module attached to it. The face detection has been done using Haar cascade and the recognition using Local Binary Pattern Histogram (LBPH) algorithm. The aim is to achieve a low cost and reliable system which can be used for a variety of applications.

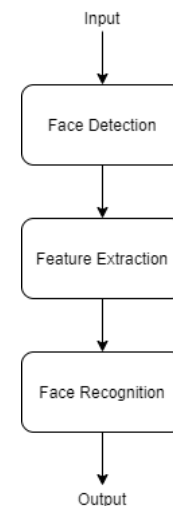


Fig. 1. Generic Face Detection Architecture [3]

2. LITERATURE SURVEY

Face recognition has been a fast growing and challenging area in the field of computer vision and real time applications. A lot of techniques have been developed over the years for face recognition. The following section gives an overview of some of the algorithms developed for this particular task.

Local Binary Pattern (LBP): This technique is used for feature extraction. The face is divided into several small regions from which histograms are extracted. These histograms are then concatenated into a single enhanced histogram which can then be used to match faces as demonstrated in [6]. This algorithm is more robust to face conditions but it is not feasible in real time environments.

Principal Component Analysis (PCA): Human recognition with the help of PCA was done by Turk and Pentland [8]. The recognition technique, called Eigen face technique defines an area

which reduces the depth of the authentic data space. This reduced knowledge area is used for recognition.

Genetic Algorithms: One of the main challenges in face recognition is feature selection. In fact, it is a global optimization problem in machine learning. It is used to remove the number of features and irrelevant, noisy, redundant data in order to improve efficiency and accuracy. Methods based on genetic algorithms have been proposed [8, 9] which help to optimize the search strategies for feature selection. This can be particularly useful in real time applications. They have been used in tandem with some other techniques like Principal Component Analysis and Discrete Cosine Transform to achieve up to 99% [8] accuracy in face detection.

Deep Learning and CNNs: Convolutional neural network is a class of deep neural networks, which has been successful for face recognition algorithms. Complete images can be provided to the network (for feature selection, extraction and training), but this can be a complex and time consuming task. Gupta, Priya, et al [10] have proposed a new way of using deep neural network for face recognition. This approach involves providing only the extracted facial features as input instead of providing raw pixels. This approach is less complex than the traditional method but still manages to achieve 97.05% accuracy. The neural network consists of four dense layers. Another framework called region-based CNN (RCNN) [11], which is a kind of CNN extension has been used for face detection. Sun, Xudong, et al have proposed a method that improves the existing RCNN method that incorporates strategies like feature concatenation [12], hard negative mining, multi-scale training, etc. Although the efficiency and scalability of this method hasnt been addressed as of yet, it still managed to achieve cutting-edge performance when evaluated on the Face Detection Dataset and Benchmark (FDDB) [13]. For high-accuracy real time face verification, MobileFaceNets [15], an extremely efficient CNN model has been proposed. It uses less than 1 million parameters and has an actual inference time of 18 milliseconds. It has been designed specifically for mobile devices due to their low computation power. It is very useful for real time applications. A real life application of CNNs for face detection has been demonstrated in [14]. It uses a CNN model that has been trained using an artificially augmented dataset to successfully recognize pigs at a farm. Over time the accuracy of deep CNNs has been known to increase steadily, with the most recent deep CNN scoring more than the median of the forensic facial examiners [16].

3. SYSTEM DESIGN

The hardware includes a Raspberry Pi 3 Model B device along with a camera module. It has a 1.2 GHz 64 bit CPU along with 1 GB RAM and the camera module has a resolution of 5 MP. Python language has been used along with the OpenCV library to create the face recognition program. OpenCV is an open-source library aimed at computer vision. It is highly optimized and offers excellent support for real time applications [1]. The input from the camera module is being processed into a grayscale image in order to reduce computation. Before anything, the face has to be detected i.e. captured by the system. The method used is the Haar Cascade Classifier. After detection, the LBPH algorithm identifies the face.

4. HAAR CASCADE

It is an object detection algorithm based on machine learning. It uses a lot of positive and negative examples to train the classifier.

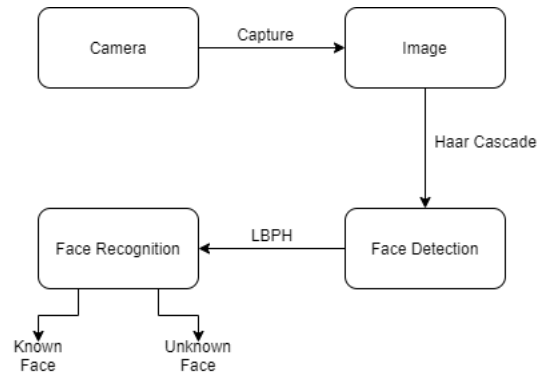


Fig. 2. Flow of Operation

Feature Selection is done first which involves collection of Haar features which are nothing but rectangular regions adjacent to each other at a particular location. To extract a feature, the difference of sum of pixel intensities of each region is calculated [5, 2]. The cascade classifier then makes the detection by moving a window over the image. Each location specified by the window is then labelled positive or negative. If a region is labelled as negative, then the window moves on to the next location, else another stage of classification is done. This stage yields a final verdict of positive [5, 2].

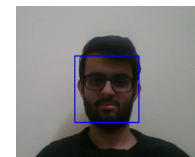


Fig. 3. Haar Cascade in Action

5. LBPH

It is a simple but effective algorithm which labels the pixels of an image. The labelling is done using a technique called thresholding. It is a simple form of image segmentation that converts a grayscale/color image to a binary format. This process basically separates the background from the foreground. [3].

It has four parameters:

- (1) radius
- (2) number of neighbours
- (3) grid_x
- (4) grid_y

The radius is used for building the circular pattern around the central pixel. The number of neighbours represents the sample points to build a circular binary pattern from. grid_X and grid_Y represent the number of cells in horizontal and vertical directions respectively [6].

The algorithm divides the input image into windows of pixels. Each window is then converted into a matrix. The matrix is populated with the intensity value of each pixel in the region. The central value of each matrix is considered as the threshold. Then the neighboring binary values are set according to this value. If the neighboring value is higher than the threshold then it is set as 1

and if it is less than the threshold , then it is set as 0 [6].

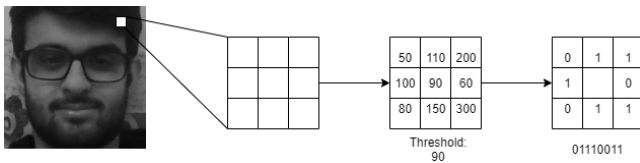


Fig. 4. Example of LBPH

Once the binary matrix is ready, each binary value from the matrix is concatenated to get the final value. This process is then repeated along the length and breadth of the image to divide it into multiple grids. These grids are then used to obtain a histogram containing the central value of each matrix. This histogram is unique for each image.

Hence, the recognition can be done by comparing this histogram with the histogram of an unlabelled image [6].

6. DATASET

A self created dataset was used to train the model. The pictures were taken using the Raspberry Pi camera module. We trained the model to recognize three people. The dataset consists of 90 images - 30 for each person. Each image has the dimension of 640x480.



Fig. 5. Images from the Dataset

7. EXPERIMENTAL SETUP

The experimental setup consists of a Raspberry Pi device and a camera module. The specifications of the device are:

Processor: 1.4 GHz, 64 bit

RAM: 1 GB

Camera Resolution: 5 Megapixel

The device is running a linux operating system (Raspbian Buster) specifically designed for the device. The programming language used is Python (version 3.7).

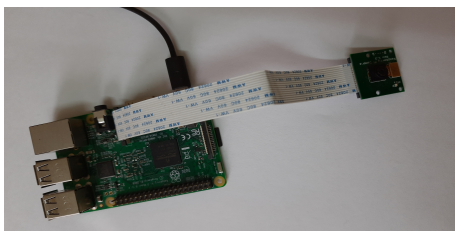


Fig. 6. Setup

8. RESULT

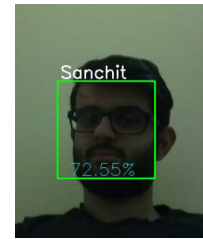


Fig. 7. The system recognizing the face correctly

The system recognized all the trained faces. The average accuracy obtained over a duration of 60 seconds was 72.01% with a maximum value of 98.53% at a frame rate of 3 FPS. These numbers indicate that the Raspberry Pi can be used for simple real time applications despite its modest hardware capability.

9. CONCLUSION

A low cost and reliable face recognition system can have many benefits. It can be used for implementing biometric security systems which can be used in schools and offices. Other applications may include real time crowd monitoring systems, assistive devices for visually impaired people which can help in navigation, surveillance systems, etc. The field of face recognition is growing at a tremendous pace right now and the need for a low cost and reliable system is imperative.

10. REFERENCES

- [1] Bradski, Gary, and Adrian Kaehler. "Opencv." *Dr. Dobbs journal of software tools* 3 (2000).
- [2] Lienhart, Rainer, and Jochen Maydt. "An extended set of haar-like features for rapid object detection." *Proceedings. international conference on image processing*. Vol. 1. IEEE, 2002.
- [3] Zhao, Wenyi, et al. "Face recognition: A literature survey." *ACM computing surveys (CSUR)* 35.4 (2003): 399-458.
- [4] Viola, P., and M. Jones. "Face detection." *IJCV* 57 (2004): 2.
- [5] Soo, Sander. "Object detection using Haar-cascade Classifier." *Institute of Computer Science, University of Tartu I*(2014): 1-12.
- [6] Gondole, Devendra, and P. A. Salunkhe. "Face Recognition Based On Local Binary Pattern." (2018).
- [7] Abate, Andrea F., et al. "2D and 3D face recognition: A survey." *Pattern recognition letters* 28.14 (2007): 1885-1906.
- [8] Zhi, Hui, and Sanyang Liu. "Face recognition based on genetic algorithm." *Journal of Visual Communication and Image Representation* 58 (2019): 495-502.
- [9] Moussa, Mourad, Maha Hmila, and Ali Douik. "A novel face recognition approach based on genetic algorithm optimization." *Stud. Inform. Control* 27.1 (2018): 127-134.
- [10] Gupta, Priya, et al. "Deep neural network for human face recognition." *International Journal of Engineering and Manufacturing (IJEM)* 8.1 (2018): 63-71.
- [11] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

- [12] Bell, Sean, et al. "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- [13] Jain, Vidit, and Erik Learned-Miller. *Fddb: A benchmark for face detection in unconstrained settings*. Vol. 2. No. 6. UMass Amherst technical report, 2010.
- [14] Hansen, Mark F., et al. "Towards on-farm pig face recognition using convolutional neural networks." *Computers in Industry* 98 (2018): 145-152.
- [15] Chen, Sheng, et al. "Mobilefacenet: Efficient cnns for accurate real-time face verification on mobile devices." *Chinese Conference on Biometric Recognition*. Springer, Cham, 2018.
- [16] Phillips, P. Jonathon, et al. "Face recognition accuracy of forensic examiners, superrecognizers, and face recognition algorithms." *Proceedings of the National Academy of Sciences* 115.24 (2018): 6171-6176.