Global Journal of Enterprise Information System

DOI: 10.18311/gjeis/2018/19836

Real Time Face Recognition in Raspberry Pi: A Guide to Proper Usage of the Available Resources

Bhabesh Rath*

Mechanical Engineering, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India; bhabeshrath@yahoo.co.in

Abstract

The use of facial recognition technology is gaining rapid popularity due to its appealing nature and possible use in various fields of life. Its integration into security systems as well as other aspects of technology such as robots has caused researchers around the globe to come up with numerous methods of recognition using different concepts. This paper is a part of a project aiming to develop a robot to be able to identify people in the daily environment. Hence, it makes use of the most readily available and student-friendly development board, the Raspberry Pi, for image processing. The goal of this paper is to compare few widely used methods of face detection and conclude as to which is better at the task.

Keywords: Cascades, Convolutional Neural Networks, Face Detection, Face Recognition, Raspberry Pi, Webcam

Paper Code (DOI): 19836**; Originality Test Ratio:** 02%**; Submission Online:** 25-Jan-2018**; Manuscript Accepted:** 31-Jan-2018; **Originality Check:** 02-Feb-2018**; Peer Reviewers Comment:** 12-Feb-2018**; Double Blind Reviewers Comment:** 15-Feb-2018**; Author Revert:** 22-Feb-2018**; Camera-Ready-Copy:** 24-Feb-2018**; Editorial Board Excerpt:** 26-Feb-2018.

Editorial Board Excerpt: *At the Time of Submission (ToS) submitted paper had a 02% plagiarism which is an excellent clue as far as originality report is concerned and falls under an accepted percentage for publication. The editorial board is of an inspection that manuscript had a subsequent surveillance by the blind reviewer's which at a in a while stages had been rectified and amended by an author(bhabesh rath) in a range of phases as and when required to do as a result. The reviewer's had in an initial stages comment with minor revision with a following remark which at a short span restructured by an author. The comments related to references, abstract and body text is noticeable both subject-wise and research wise by the reviewers during evaluation and further at blind review process too. All the comments had been collective at a variety of dates by the authors' in due course of time and same had been integrated by the author in addition. By and large all the editorial and reviewer's comments had been incorporated in a paper at the end and further the paper had been earmarked and decided under "**View Point**" class as its things to see and underline the work in relation to Real Time Face Recognition.*

1. Introduction

Face recognition is an interesting and important user identification method. In this world where technology makes huge advances every moment, use of face recognition technology has become quite popular. This field has increasingly attracted the attention of researchers throughout the years, witnessing a continuous rise in the number of people taking interest in this technology. Literature survey statistics show that research work in face recognition system is in its booming era, and in the past forty years, the research in this field has increased exponentially[1] (Figure 1).

Face recognition technology makes an attempt to simulate the ability of the human eye to be able to recognize faces. This is made possible by use of feature extraction method which allows the computer to identify and extract points of interest from a given image and compare it with a template to provide results. This project runs a few programs for face recognition using the Raspberry Pi and a webcam. The programs use different approaches, as mentioned above, one being the HAAR cascade classifier and the second being the Convolutional Neural Networks[2]. The cascade method is tested only for face detection as using it for recognition separately becomes quite impractical. Later on, it is combined with machine learning to allow the recognition tasks to be performed.

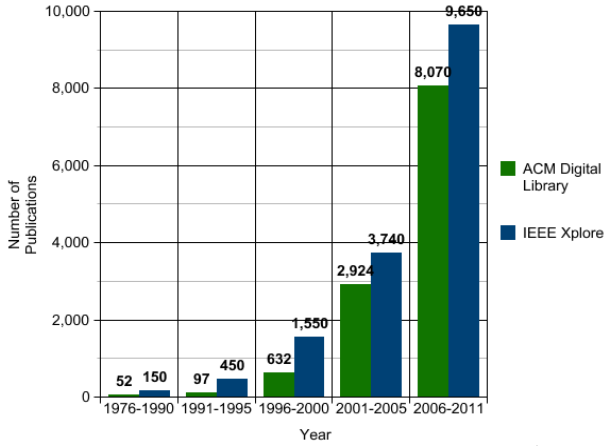

Figure 1. Research Statistics of Face Recognition[1].

*Author for correspondence

2. Face Detection

The first stage of this paper concentrates on face detection using Haar Cascade and Convolutional Neural Networks. The Haar cascade method is used for the detection of faces using a pretrained file available in.xml format. Suggested by Viola and Jones, the Haar is a feature based object classifier. This is a machine learning based approach that trains cascade functions from positive and negative images. In this case, the images with a face are positive and the ones without a face are negative. It extracts features from an image that are termed as Haar features (Figure 2). Each feature here forms a single value that is the difference between the sum of the pixels in the different regions within the enclosing window. This operation gives out a large number of features amounting up to 1,60,000+ for a detector with a base resolution of 24 × 24[3].

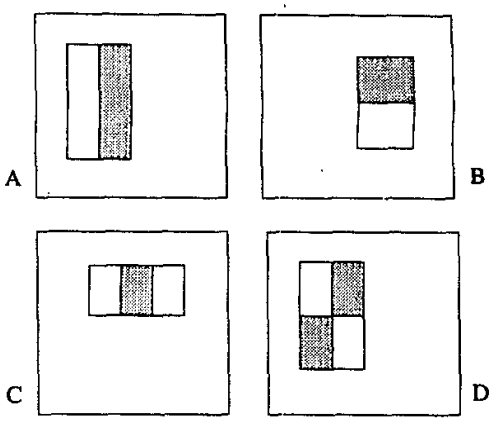

Figure 2. For example, the above squares featuring rectangles show features relative to the detection window. The sum of the pixels which lie within the white rectangles are subtracted from the sum of pixels in the grey rectangles. Nivo-rectangle features are shown in (A) and (B). Figure (1) shows a three-rectangle feature, and (D) a four-rectangle feature[3].

Hence, the cascade uses integral images in order to reduce the features that simplify calculations to operations involving just four pixels. This method when used for faces takes two major things into consideration, one being that, the region of eyes is darker than the region of nose and cheeks. The second being the fact that the eyes are darker than the nose bridge. The use of Adaboost helps with selection of the best features that are based on the error rate of each feature. Hence the final classifier is a weighted sum of each weak classifier picked out separately. The process is sped up by separating the 'face' and 'no-face' region to reduce calculations by allowing the computer to focus on the region that has higher chances of a face being present and ignoring the regions without a face.

A Convolutional neural network works differently compared to the Haar cascade. The CNN (Convolutional Neural Network) consists of three major parts. The convolution, pooling and the fully connected layer. Convolution is a filter applied like a sliding window to extract features from the given input. The filter shifts after each operation by an amount termed as strides. The filter, also known as the Kernel, covers the whole input by the given strides[4]. At each operation, the matrix multiplication of kernel and the current region of input are calculated.

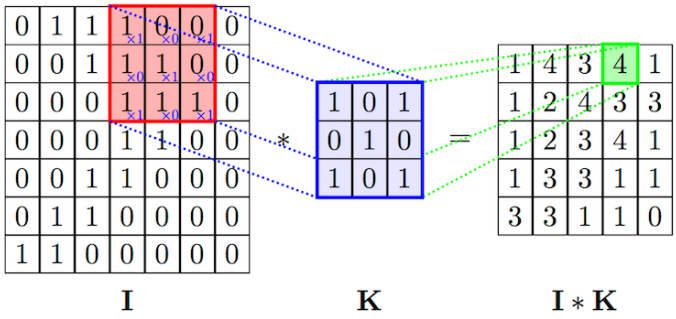

Figure 3. One single operation involving the Kernel (K) and Input (I) at a stage of the convolution to extract features.

When there is a discrepancy in the size of the input and kernel, same padding is used to maintain the size of the input. The ReLU (Rectifier Linear Unit) is used as an activator to increase the efficiency of the network. The next part is pooling, that reduces the dimensionality. The fully connected layer, as suggested by the name, makes sure that each neuron at input is connected to each neuron at output, strengthening the network and allowing better results.

3. Face Recognition

As the Haar cascades will have to be separately trained for each face to be able to recognize faces, their use for face recognition is quite unfeasible. Therefore, they are combined with a machine learning code to allow face recognition. The Haar method is used here to detect the face and then the algorithm teaches the device to recognize the face by extracting various features and marking them as unique[5].

The next method is the usage of dlib machine learning library to allow face detection and recognition. The method used here is the Histogram of Oriented Gradients method to detect faces (Figure 4). The program converts the given input into an image with arrows pointing from lighter to darker regions. It then proceeds to mark major points of interest to recognize the face[6]. The code itself uses very less memory and functions accurately when it comes to face detection.

The third method uses face recognition features that come as an additional library and run on the concepts of machine learning. It uses the eigen faces to train and recognize faces.

The fourth is a combination that uses Haar for face detection and fisher faces for face recognition.

The codes have been made according to the requirements and tweaked to best performance to the knowledge of the author.

Figure 4. Histogram of oriented gradients.

4. Setup

The setup consists of the Raspberry Pi 3 Model B, and a Logitech C270 webcam (Figure 5). The Raspberry Pi board having the following specs: inbuilt Wi-Fi and Bluetooth module, 1GB Ram, 1.2 GHZ quad-core ARM Cortex A53v. The setup is quite affordable and allows experiments to be carried out without worrying about the cost effectiveness of the project. The Logitech camera allows 720 p video capture and has an inbuilt mike allowing it to serve more than one purpose. Rest of the setup is mostly software that has to be installed to be able to carry out the tests. The following were installed on the Raspberry Pi:

- Raspbian Operating System.
- Python 2.7.13.
- OpenCV 3.1.0 along with extra modules.
- Other dependencies such as numpy, scikitlearn, dlib, face_recognition etc.

Figure 5. The setup made to run the experiments for facial detection and recognition. The first picture shows the Raspberry Pi board while the second picture shows the whole setup along with the webcam.

Three different codes were used for the experimentation process with a database of a hundred photos to train the system and preinstalled xml files for frontal face recognition. A few different lighting conditions were tested of which provide indifferent results.

5. Results

The results have been presented in terms of CPU usage by the system, accuracy of the method, time to load the program and the user experience. These factors play a major role when it comes to carrying out tasks in real time as the responsiveness of the system forms a criterion to determine the success of the system.

The frontal face detection using Haar cascade was able to detect faces quite smoothly in distance ranging from approximately 0.3 m to 3 m. The detection failed beyond that as it was unable to grab the features of the face beyond that point. The HOG detection algorithm was comparatively slow and took a lot of time to process the images giving about 15 seconds between the changes in each frame. This made the algorithm less responsive and quite the hassle to deal with. The fisherfaces algorithm was very responsive in detection but failed at a few places in identification, typically when more than one

Table 1. CPU Usage

Method	CPU usage in percentage	Accuracy percentage	Time to Load in seconds
Haar Cascade	75–80%	30%	2.34
HOG detection using PCA	25–30%	42%	15.62
Detection using Fisherfaces	60–70%	87%	12.54
Detection using Eigenfaces	60–70%	70%	31.85

person was to be identified. This was then improved by increasing the number of inputs provided during the training. The CPU usage and the other parameters during the programs are tabulated below:

Detection rates, as represented by the table are clearly highest for the fisherfaces detection. The accuracy increased to about 90% after increasing the number of inputs for training.

6. Conclusion

The experiment carried out gives an idea as to the method that would prove to be most effective for application in real world problems such as security systems, interactive environments or social robots. The fisherfaces method proves to be the most useful of all the four tested in the in process. Moreover, the equipment used is easily accessible and proves to be a good tool for experiment even with its own limitations.

There is further scope of improvement in the facial recognition techniques that are used here and can be brought in through code optimization or usage of various other facial recognition features available that have not been explored in this paper.

7. References

1. Gupta I, Patil V, Kadam C, Dumbre S. Face detection and recognition using Raspberry Pi. 2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE); 2016. https://doi.org/10.1109/WIECON-ECE.2016.8009092.
2. Hu G, Yang Y, Yi D, Kittler J, Christmas W, Li SZ, Hospedales T. When face recognition meets with deep learning: An evaluation of convolutional neural networks for face recognition. 2015 IEEE International Conference on Computer Vision Workshops; 2015. https://doi.org/10.1109/ICCVW.2015.58.
3. Viola PA, Jones MJ. Rapid object detection using a boosted cascade of simple features. Proceedings IEEE Conference on Computer Vision and Pattern Recognition; 2001. https://doi.org/10.1109/CVPR.2001.990517.
4. Lit H, Lin Z, Shen X, Brandt J, Huat G. A convolutional neural network cascade for face detection, 2015 IEEE Conference on Computer Vision and Pattern Recognition; 2015.
5. Ranjan R, Patel VM, Chellappa R. HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. 2017 IEEE Transactions on Pattern Analysis and Machine Intelligence; 2017. https://doi.org/10.1109/TPAMI.2017.2781233.
6. Agarwal A, Triggs B. Multilevel image coding with hyperfeatures. International Journal of Computer Vision. 2008; 15–27. https://doi.org/10.1007/s11263-007-0072-x.
7. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009; 2009. p. 248–255. https://doi.org/10.1109/CVPR.2009.5206848.
8. Chen D, Ren S, Wei Y, Cao X, Sun J. Joint cascade face detection and alignment. Computer Vision-ECCV; 2014. https://doi.org/10.1007/978-3-319-10599-4_8.
9. Felzenszwalb PF, Girshick RB, McAllester D, Ramanan D. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence; 2010. https://doi.org/10.1109/TPAMI.2009.167 PMid:20634557.
10. Schroff F, Kalenichenko D, Philbin J. FaceNet: A unified embedding for face recognition and clustering. 2015 IEEE Conference on Computer Vision and Pattern Recognition; 2015. https://doi.org/10.1109/CVPR.2015.7298682.
11. Yin X, Liu X. Multi-task convolutional neural network for pose-invariant face recognition. IEEE Transactions on Image Processing; 2018. https://doi.org/10.1109/TIP.2017.2765830.
12. Zhao W, Chellappa R, Phillips PJ, Rosenfeld A. Face recognition: A literature survey, ACM Computing Surveys. 2003 Dec; 35(4):399–458. https://doi.org/10.1145/954339.954342.
13. Lienhart R, Maydt J. An extended set of Haar-like features for rapid object detection. Proceedings International Conference on Image Processing; 2002. p. I-900–I-903. https://doi.org/10.1109/ICIP.2002.1038171.
14. Jiang H, Learned-Miller E. Face detection with the faster R-CNN. 2017 12th IEEE International Conference on Automatic Face and Gesture Recognition; 2017.
15. Ahonen T, Hadid A, Pietikainen M. Face description with local binary patterns: Application to face recognition. IEEE Trans Pattern Anal Mach Intell. 2006; 28(12):2037–41. https://doi.org/10.1109/TPAMI.2006.244 PMid:17108377.

Annexure-I

Real Time Face Recognition in Raspberry Pi: A Guide to Proper Usage of the Available Resources

ORIGINALITY REPORT

2%
SIMILARITY INDEX

PRIMARY SOURCES

1	documents.mx Internet	9 words — 1%
2	www.enggjournals.com Internet	8 words — < 1%
3	Zhang, Zhanpeng, Ping Luo, Chen Change Loy, and Xiaoou Tang. "Learning Deep Representation for Face Alignment with Auxiliary Attributes", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015. Crossref	8 words — < 1%
4	arxiv.org Internet	8 words — < 1%

EXCLUDE QUOTES ON
EXCLUDE BIBLIOGRAPHY ON
EXCLUDE MATCHES OFF

Source: http://www.ithenticate.com/

Prevent Plagiarism in Publication

The Editorial Board had used the ithenticate plagiarism [http://www.ithenticate.com] tool to check the originality and further affixed the similarity index which is 02 % in this case *(See Annexure-I)*. Thus the reviewers and editors are of view to discover it suitable to publish in this *Volume-10, Issue-1, January-March, 2018*.

Citation:
Bhabesh Rath
"Real Time Face Recognition in Raspberry Pi: A Guide to Proper Usage of the Available Resources",
Global Journal of Enterprise Information System. Volume-10, Issue-1, January–March, 2018. (http://informaticsjournals.com/index.php/gjeis)
DOI: 10.18311/gjeis/2018/19836
Conflict of Interest:
Author of a Paper had no conflict neither financially nor academically.