# Facial Detection and Recognition on RaspberryPi: Security can be Cheap and Smart

Foley, Patricia

Volkmann, Landon

Yadav, Hari

Zakari, Abdulmuhaymin

*UMKC*

May 05, 2020

**ABSTRACT**

The development of this project will be vertical in nature to demon- strate proof of concept rather than extensibility. In short, the Group 3 project team has developed a smart security system on the Raspber- ryPi at minimal cost. The system will use a motion sensor and camera to gather data, a convolutional neural network for facial detection and recognition, and finally a push notification based output. The targeted use case of this technology is a household security system. A home- owner will point the system at the door, and the system will notify the homeowner via push notification that they have a guest as well as the system’s best guess at the guest’s name. For this particular im- plementation, facial data will be manually created for demonstration purposes. Future implementations of the system may include but are not limited to: Facebook profile data, LinkedIn profile data, Google Photos image data, etc. The end goals of this project may be ana- lyzed on a few different planes. In the meta-educational plane, this project will serve to demonstrate Group 3’s competence in Machine Learning techniques, basic circuitry, and comfort with the Internet of Things. In terms of practical application, this project may serve to demonstrate the potential effectivity of machine learning with simple and cost effective components like the RaspberryPi.

**1** **Introduction**

The following document is a report on the mini project for Robotic visual perception and autonomy. It involved building a system for face detection and face recognition using several classifiers available in the open computer vision library(OpenCV). Face recognition is a non-invasive identification system and faster than other systems since multiple faces can be analysed at the same time. The difference between face detection and identification is, face detection is to identify a face from an image and locate the face. Face recognition is making the decision ”whose face is it ? ”, using an image database. In this project both are accomplished using different techniques and are described below. The report begins with a brief history of face recognition. This is followed by the explanation of HAAR-cascades, Eigenface, Fisherface and Local binary pattern histogram (LBPH) algorithms. Next, the methodology and the results of the project are described. A discussion regarding the challenges and the resolutions are described. Finally, a conclusion is provided on the pros and cons of each algorithm and possible implementations.

**2 The History of Face Recognition**

Face recognition began as early as 1977 with the first automated system being introduced By Kanade using a feature vector of human faces [1]. In 1983, Sirovich and Kirby introduced the principal component analysis(PCA) for feature extraction [2]. Using PCA, Turk and Pentland Eigenface was developed in 1991 and is considered a major milestone in technology [3]. Local binary pattern analysis for texture recognition was introduced in 1994 and is improved upon for facial recognition later by incorporating Histograms(LBPH) [4], [5]. In 1996 Fisherface was developed using Linear discriminant analysis (LDA) for dimensional reduction and can identify faces in different illumination conditions, which was an issue in Eigenface method [6]. Viola and Jones introduced a face detection technique using HAAR cascades and ADABoost [7]. In 2007, A face recognition technique was developed by Naruniec and Skarbek using Gabor Jets that are similar to mammalian eyes [8], [9]. In This project, HAAR cascades are used for face detection and Eigenface, Fisherface and LBPH are used for face recognition.

3 Face Detection using Haar-Cascades

A Haar wavelet is a mathematical fiction that produces square-shaped waves with a beginning and an end and used to create box shaped patterns to recognise signals with sudden transformations. An example is shown in figure 1. By combining several wavelets, a cascade can be created that can identify edges, lines and circles with different colour intensities. These sets are used in Viola Jones face detection technique in 2001 and since then more patterns are introduced [10] for object detection as shown in figure 1. To analyse an image using Haar cascades, a scale is selected smaller than the target image. It is then placed on the image, and the average of the values of pixels in each section is taken. If the difference between two values pass a given threshold, it is considered a match. Face detection on a human face is performed by matching a combination of different Haar-like-features. For example, forehead, eyebrows and eyes contrast as well as the nose with eyes as shown below in figure A single classifier is not accurate enough. Several classifiers are combined as to provide an accurate face detection system as shown in the block diagram below in figure 3.

# pic goes here

**6 Conclusion and Future Work**

**RÉFERENCES**

1. ITU­T (2012). Overview of the Internet of things. ITU­T. Retrieved from http://www.itu.int/rec/T­REC­ Y.2060­201206­I on 2016, Jan.
2. Cisco IBSG © 2011 Cisco et/ou ses filiales. Tous droits réservés.
3. Padmini Gaur, Mohit P. Tahiliani .Wireless Information Networking Group (WiNG)

1. A. Dunkels, B. Gronvall, T. Voigt Contiki ­ a Lightweight and Flexible Operating System for Tiny Networked Sensors In Proceedings of the First IEEE Workshop on Embedded Networked Sensors, Tampa, Florida, USA, 2004
2. Klues K, Liang CJM, Paek J, Musaloiu R, Levis P, Terzis A, Govindan R. TOSThread: Thread­Safe and Non­ Invasive Preemption in TinyOS. Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems; Berkeley, CA, USA. 4–6 November 2009; pp. 127–140.
3. PtolemyProject. Availableonline http://ptolemy.eecs.berkeley.edu/ (accessed on 17 April 2011)
4. Akyildiz IF, Melodia T, Chowdhury KR. A Survey on Wireless Multimedia Sensor Networks. Comput. Netw. 2007;51:921–960
5. Dunkels A, Gronvall B, Voigt T. Contiki a Lightweight and Flexible Operating System for Tiny Networked Sensors. Proceedings of the 9th Annual IEEE International Conference on Local Computer Networks; Washington, DC, USA. October 2004; pp. 455–462.
6. Eswaran A, Rowe A, Rajkumar R. Nano­RK: An Energy­Aware Resource­Centric RTOS for Sensor Networks. Proceedings of the 26th IEEE Real­Time Systems Symposium; Miami, FL, USA. 5–8 December 2005.
7. Cao Q, Abdelzaher T, Stankovic J, He T. The LiteOS Operating System: Towards Unix Like Abstraction for Wireless Sensor Networks. Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008); St Louis, MO, USA. 22–24 April 2008.
8. Julien Le Sech, thirsday 5 january 2012. FreeRTOS sur ATmega328 .
9. RIOT (2015). RIOT ­ The friendly Operating System for the Internet of Things. RIOT­OS.org. Retrieved from www.riot­os.org/ on 2016, Jan.
10. R. Morabito, R. Petrolo, V. Loscrì, and N. Mitton, “Reprint of : LEGIoT: A Lightweight Edge Gateway for the Internet of Things,” *Futur. Gener. Comput. Syst.*, vol. 92, pp. 1157–1171, 2019.

[14] Free Software Foundation https://www.gnu.org/licenses/lgpl­3.0.en.html .

[15] R. Morabito, “Virtualization on internet of things edge devices with container technologies: A performance evaluation,” IEEE Access, vol. 5, pp. 8835–8850, 2017.