

ANOMALY DETECTION USING OPENCV

DR.M.SENTHIL KUMARAN ¹, Suraj S ², Naga Dheeraj N ³

¹Associate professor &HOD, SCSVMV University, Kanchipuram, Tamilnadu, India.

²CSE, SCSVMV University, Kanchipuram, Tamilnadu, India.

³CSE, SCSVMV University, Kanchipuram, Tamilnadu, India.

Abstract - In recent days identification of Motion and Image capturing is big task to the researchers and Information Technology (IT) Engineers. This leads to abnormal events in crowded areas, ATM centers, Server rooms and Intelligent security control rooms etc. In this paper new techniques has been used to detect motion and image of the particular object clearly. These two techniques are for image recognition haarcascade algorithm and for motion opencv algorithms has been used. The two techniques has high in accuracy and speed of operation is very high. This has been analyzed with the help of RASPBERRY PI module and ARM1176JZF-S processor.

Key Words: Raspberry Pi,ARM1176JZF-S,SD/MMC Card, python language.

1. INTRODUCTION

Video Anomaly Detection is the process of detecting abnormal events in the Video. Video event detection is the process of detecting an event in the video. Video Anomaly detection is the subclass of Video Event Detection. This process of video Anomaly Detection produces vast amounts of data and display. There is always a necessity for this kind of software that can detect movement and record in a video. There are 3 different types of video Anomaly Detection techniques, they are 1.Manual 2.semi automated 3.Fully automated. Manual Anomaly Detection is the method in which a person is specially designated for the process of monitoring the live video. Data collected by this process nevertheless is hard to monitor. Indeed this is not conceivable to record all along because the hard disk will be full as the software is to be run for a significant number of days. The Semi-automated process is the one in which there is no requirement of an employed guy. When the machine detects face or motion in a video the machine starts recording from that instance and sends the recorded video for analyzing. Fully automated is similar to semi-automated but there is no involvement of video analyzer to analyze the anomaly video. The machine, however, processes the video and detects the anomaly in the motion and triggers the alarm. This can be done in two ways 1. Supervised 2.Unsupervised, Supervised have a training phase, whereas Unsupervised doesn't have any explicit training phase. The algorithm detects an abnormal event in the Supervised method if it finds an event that has no similarity with the trained events. There are different applications of video surveillance like access control, object-identification, anomaly detection, Intelligent visual surveillance (IVS) refers to an automated visual monitoring process that involves analysis and interpretation of object behaviors, as well as to object detection and tracking, to understand the visual events of the scene. The underlying principles used for all these methods of surveillance are based on detection, segmentation, tracking the objects in the video. The Embedded system is using for Anomaly Detection. Embedded systems are used in different areas like medical engineering [1], aeronautics [2], water and energy supply [3][4], transportation [5][6], automotive industry [7], and information and communication technology [8]. However, in everyday life embedded systems are omnipresent, e.g., in smart phones or tablets. These devices contain usually an integrated camera and deal with challenging tasks, like the calculation of high dynamic range images [9] or the detection of faces in real-time [10]. Although, the performance of the embedded systems is limited because of size and mobility, the systems are used in many applications for image and video signal processing. In this paper, the basic understanding in processing of image and motion detection, face recognition, are explained.

The handling of the system should be possible with display, mouse, and keyboard, and a user-friendly OS with a graphical user interface (GUI) should also be used. The needed connection for the additional hardware should exist and a connection to a network should also be possible. For the usage of Python, a Python interpreter should be present. Finally, the embedded system should be inexpensive, but capable for the experiments.

Different embedded systems exist, but not every of those fulfills these requirements. So in this paper Raspberry Pi is using, The Raspberry Pi Foundation provides six different embedded systems [11]. The Raspberry Pi Model A has been introduced in February 2012 as first embedded system and has been replaced by the successor Raspberry Pi 1 Model A in November 2014 [11]. The latest model is the Raspberry Pi 3 Model B. It is a cost-effective, up-to-- date and widespread embedded system. The Raspberry Pi 3 Model B fulfill all the requirements (so the Raspberry Pi has a better CPU and RAM. Thus, the Raspberry Pi is used in the Image and Video Signal Processing on Embedded Systems).

Face detection:

This paper presents a process of using local binary pattern (LBP) texture features to provide a facial image representation efficiently. The Facial image is partitioned into local regions to extract texture descriptions from each region independently. The global description of the face is then formed by coalescing descriptors which are extracted from texture descriptions. The texture description of a specific region describes the appearance of a region and the combination of all the region descriptions helps in encoding the global geometry of the face. This local-feature based and hybrid method seems to be more favorable against different geometries of face and illumination. This process of texture extraction is considered from this paper to extract and detect the face based on Haarclassifiers technique [13].

Motion Detection:

This paper proposes a combination of Adaptive background modeling system and the cascade classifiers based object detection system for the application of Video surveillance. The system proposed is resistant to temporal illumination changes and tracks moving objects precisely. It also adapts itself to long-lasting changes in the background over time. After learning of the statistical model in the background and by subtracting it from the original frame the foreground is extracted. The main job of the background model is to learn only the stationary objects of the scene and to ignore all the moving objects. Gaussian Mixture Model is used for modeling the background adaptively, in this way the regions in motion are tracked. The moving objects can be a human, an animal or a vehicle [14].

METHODOLOGY: This consists mouse, keyboard, HDMI to DVI adapter cable, power supply, Raspberry Pi 3 Model B, USB camera, and micro SD card components are used. In the first experiment and before every experiment all the components are connected together. The setup is very easy to perform on embedded systems.

First, the micro SD card with the pre-installed OS Raspbian Jessie (Raspberry Pi Foundation, 2018) is inserted in the SD card slot. After that, all the other hardware is connected by USB or HDMI with the Raspberry Pi as depicted in Fig. 1. The OS boots after connecting the power supply. The whole hardware is recognized instantly.

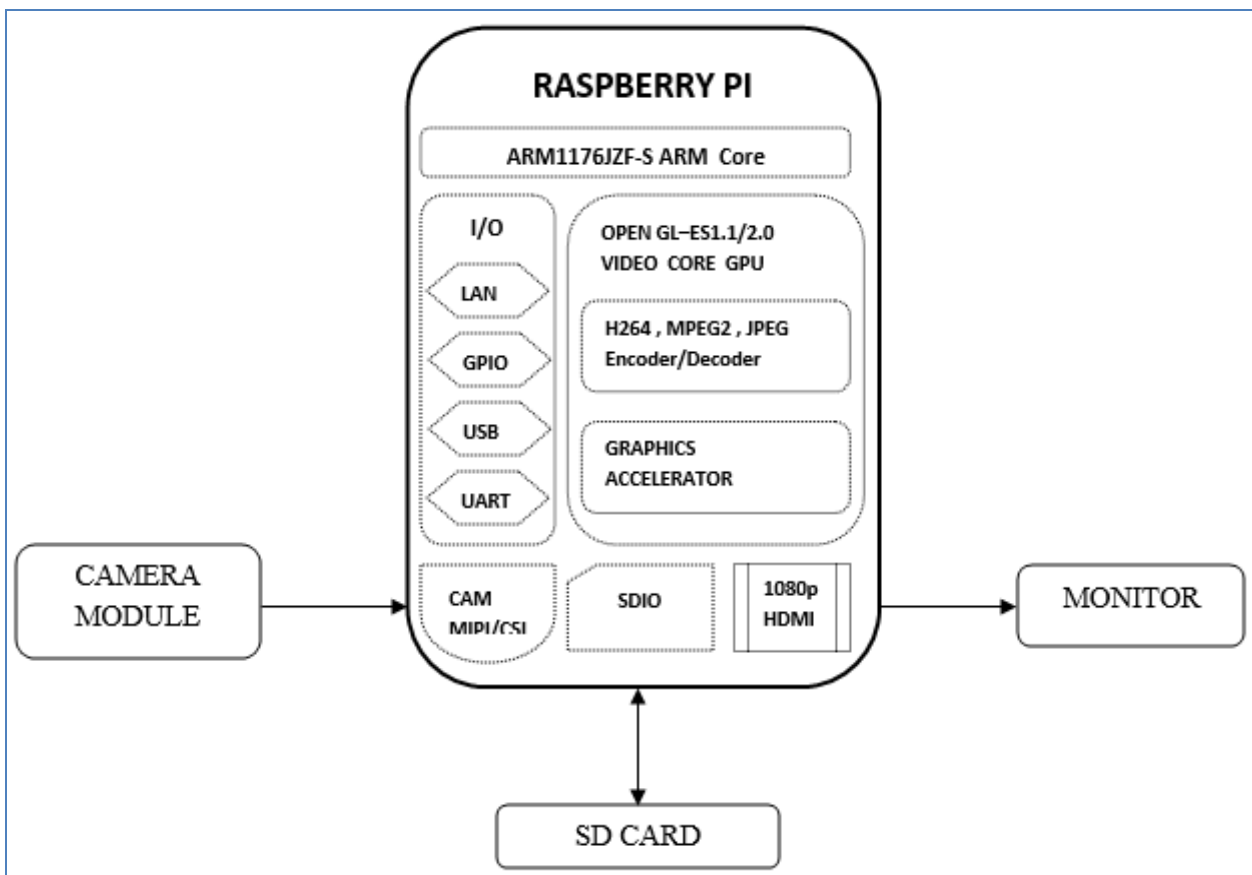


Fig. 1. Block diagram of the connected components of the workstation.

The camera is connected to Raspberry Pi camera module. In this paper an USB camera used instead of the Raspberry Pi camera module (Raspberry Pi Foundation, 2018), so the experiments can easily be transferred to other embedded systems. Furthermore, USB cameras are used in many industrial computer vision applications, e.g., Basler, an internationally leading manufacturer of high-quality cameras mostly uses USB as interface.

Software: The experiments are prepared with the open source programming language Python (Python Software Foundation, 2018). The readability of a program has been the most important part during the development of Python. The language can easily be learned and supports the most common programming paradigms, e.g., object, aspect, and functional orientation. Python only needs a few keywords and the syntax is clear and easy to read. Python 2, Python 3, and the integrated development environment IDLE for Python are already installed on the OS.

Python was chosen because it is simple to read and to learn, it is open and free. Moreover, it consists of an extensive standard library, which aims at programming in general and contains specific modules for threading, databases, etc. Furthermore, there are a lot of additional packages, e.g., for plotting and handling images, which are also used in the laboratory. Python serves as a good basis for future jobs, because many companies use Python as primary programming language [12].

PROPOSED FLOW CHART:

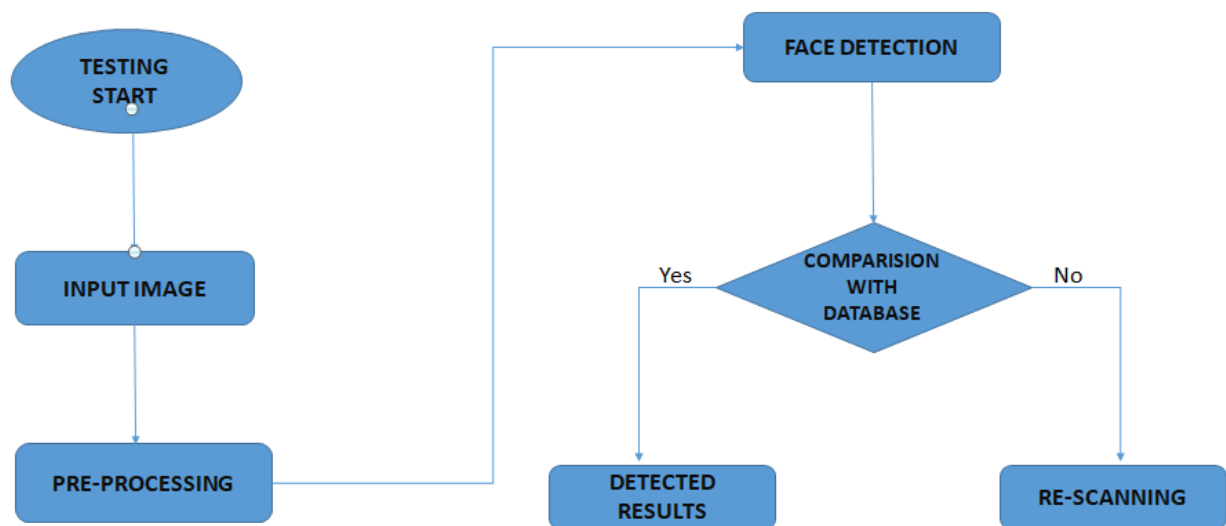


Figure.2 Flow chart for facial recognition

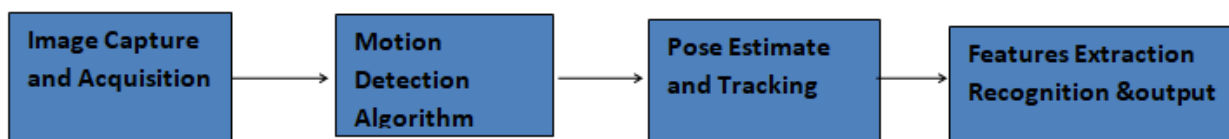


Figure.3 Flow chart for motion detection

IMPLEMENTATION OF FACIAL RECOGNIZATION PROCESS:

Initially, The input image is acquired through an inbuilt or external camera and the process of Face detection is carried out using Haar Classifiers and Ada Boosting Technique to detect the face granules using Gaussian filters to obtain a Gaussian Pyramid, The difference of Gaussian (DoG), $D(x, y, \sigma)$, is calculated as the difference between two filtered images, one with k multiplied by scale of the other. These images are then transferred to Weber Local Descriptors to analyze the spatial feature changes between the images and the mean value is calculated. The fraction $\Delta I/I$ is known as the Weber fraction. The calculated mean value in the previous process is given to Gabor Filter Bank. Depending on this mean value a single image is extracted consisting of all the spatial features and matching is done by using Euclidean distance measures the similarity between two different feature vectors.

$$Ed = \sqrt{\sum (Q - D_i)^2}$$

Where,

Q – Input image features, D – Data base features

i – Number of samples in database 1 to N

If the threshold value reaches 1.5 then the faces are said to be matched.

IMPLEMENTATION OF MOTION DETECTION:

Here, we will thus follow the flow mentioned in the flowchart of motion detection

- Image Acquisition

First in this the program checks for the input from camera devices .If no camera devices are detected ,the user is given option to use the “avi” version where input will be captured from recorded video file.

- Image Subtraction

In this two images are compared and processed using arithmetic operations of subtraction to subtract the pixel values in the image by the implementation of absolute subtraction technique there is no difference in choosing which image to subtract from which because both would result in a same image. As the most commonly used images are coloured ones the algorithm separates the images in to three planes. Then performs arithmetic subtraction to each planes .After that, the results are combined to form a colour image.

- Image processing

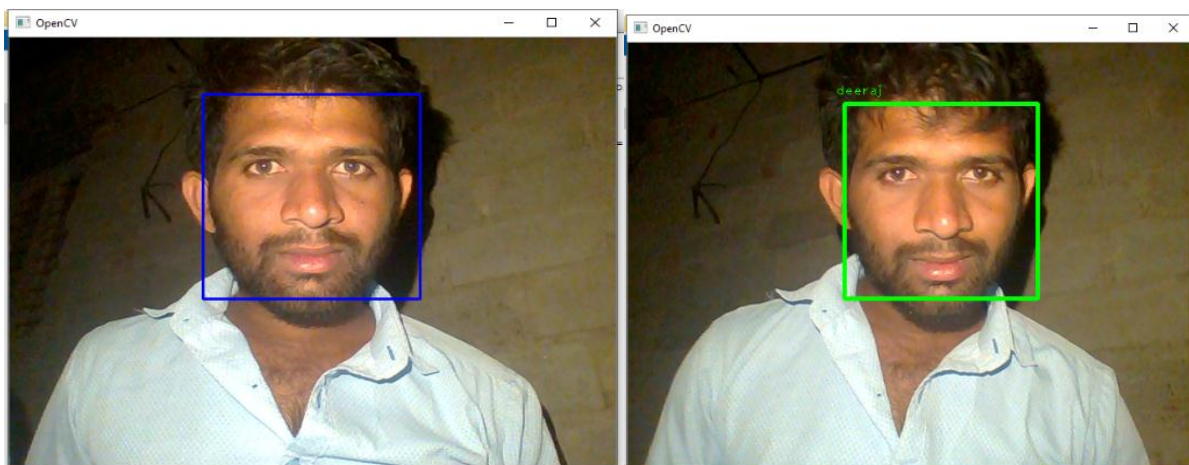
The results obtained from previous process is used in this processing state as it filters human body shapes better than the other output. An Adaptive threshold function called as Otsu threshold is used here. This function simply deletes the noise in the image that is caused by camera signal noise or change in minute pixels. The reason for using this instead of threshold for contour is because the to draw the bounding box so it does not loses any body parts outside the box.

- Bounding Rectangle Calculations

To remove the overlaying boxes and rectangles drawn to the source few operations are done. In this the rectangles which are nearer and those almost cross one another edges would be combined to form a bigger rectangle. This help in removing the problem of only a part of human being is recognized.

RESULTS:

FOR FACE DETECTION AND RECOGNITION:



FOR MOTION DETECTION:



3. CONCLUSION

In this paper the newly developed technique for finding face recognition and motion detection had presented and verified. The Raspberry Pi Model has chosen as embedded system for storing datasets and preprocessing data. The tasks have been solved on Raspberry Pi using python programming language. Python is an open source programming language, it is very easy to learning, and efficiency of the programming is high. The requirements and different experiments that have to developed and presented in this paper.

REFERENCES

1. Mastinu, E., Doguet, P., Botquin, Y., Håkansson, B., Ortiz-Catalan, M., 2017. Embedded system for prosthetic control using implanted neuromuscular interfaces accessed via an osseointegrated implant. *IEEE Trans. Biomed. Circuits.Syst.* 11 (4), 867–877.
2. Sharp, D.C., Bell, A.E., Gold, J.J., Gibbar, K.W., Gvillo, D.W., Knight, V.M., Weismuller, S.P., 2010. Challenges and solutions for embedded and networked aerospace software systems. *Proc. IEEE* 621–634.
3. Raghunathan, V., Kansal, A., Hsu, J., Friedman, J., Srivastava, M., 2005. Design considerations for solar energy harvesting wireless embedded systems. In: *Proc. Fourth International Symposium on Information Processing in Sensor Networks*, pp. 457–462.
4. Stoianov, I., Nachman, L., Madden, S., Tokmouline, T., 2007. PIPENET: a wireless sensor network for pipeline monitoring. In: *Proc. 6th International Symposium on Information Processing in Sensor Networks*, pp. 264–273.
5. Bernini, N., Bombini, L., Buzzoni, M., Cerri, P., Grisleri, P., 2014. An embedded system for counting passengers in public transportation vehicles. In: *Proc. IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications*, pp. 1–6.
6. Castellanos, J.C., Susin, A.A., Fruett, F., 2011. Embedded sensor system and techniques to evaluate the comfort in public transportation. In: *Proc. 14th International IEEE Conference on Intelligent Transportation Systems*, pp. 1858–1863.
7. Bhat, A., Samii, S., Rajkumar, R., 2017. Practical task allocation for software fault- tolerance and its implementation in embedded automotive systems. In: *Proc. IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 87–98.
8. Rupniewski, M., Mazurek, G., Gambrych, J., Nałęcz, M., Karolewski, R., 2016. A real-time embedded heterogeneous GPU/FPGA parallel system for radar signal processing. In: *Proc. Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress*, pp. 1189–1197.
9. Tsai, W.K., Lai, C.J., Sheu, M.H., Chen, T.H., 2014. High dynamic range image based on block-based edge strength for embedded system design. In: *Proc. Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 329–332.
10. Mao, H., Yao, S., Tang, T., Li, B., Yao, J., Wang, Y., 2017. Towards real-time object detection on embedded systems. *IEEE Trans. Emerg. Top. Comput.* 1-1.
11. Raspberry Pi Foundation, 2018. *Raspberry Pi - Teach, Learn, and Make with Raspberry Pi*.
12. Cass, S., 2018. The 2017 Top Programming Languages. Retrieved from. <https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>.
13. Face Description with Local Binary Patterns: Application to Face Recognition", Timo Ahonen, Abdenour Hadid, and Matti Pietikainen, 2006.
14. "Object Motion Detection and Tracking for Video Surveillance", M.sahasri, C.Gireesh, 2017.