EDGE-BASED VIDEO PROCESSING ON RASPBERRY PI

Mr. Nagesh Nam, M.Tech Student, Dept. of ECE Jayamukhi Institute of Technological Sciences
Mr. Sukumar N, Assistant Professor, Dept. of ECE Jayamukhi Institute of Technological Sciences

Abstract:
The rapid advancement of embedded vision technologies has opened new opportunities for deploying intelligent video analytics at the edge. This paper focuses on the study and implementation of various methods for motion detection, face recognition, and hand detection using state-of-the-art computer vision algorithms, optimized for the Raspberry Pi platform. By leveraging OpenCV functions and custom-trained classifiers, the system is designed to operate as a low-power, portable, and real-time surveillance solution embedded within a compact computing device. The integration of hand detection using a trained custom classifier further enhances its applicability in gesture-based control, healthcare monitoring, and human–computer interaction scenarios. In recent years, edge-based video processing has emerged as a promising solution to address the limitations of traditional cloud-dependent frameworks, which often suffer from latency, bandwidth consumption, and privacy concerns. To overcome these challenges, this work proposes an energy-efficient edge framework implemented on Raspberry Pi, enabling localized video capture, preprocessing, object detection, and decision-making without continuous reliance on remote servers. Such a design ensures faster response times, reduced network dependency, and enhanced data security. The system employs a combination of lightweight and computationally efficient algorithms, including background subtraction, edge detection, and embedded deep learning models such as MobileNet and Tiny-YOLO, tailored to the constraints of Raspberry Pi hardware. Through algorithmic optimization, the framework achieves real-time performance with acceptable frame rates and reliable detection accuracy, despite the limited processing power of the device. Experimental results validate the feasibility of the proposed approach, demonstrating its ability to effectively process live video streams for multiple tasks such as motion tracking, facial recognition, and hand gesture detection. The findings confirm that Raspberry Pi can serve as a cost-effective and scalable embedded vision platform suitable for diverse applications, including smart surveillance, traffic monitoring, smart healthcare, and IoT-enabled vision systems. Overall, the proposed edge-based video processing framework provides a practical balance between performance, privacy, and affordability, establishing it as a viable solution for next-generation intelligent embedded systems.

Keywords: Edge Computing, Raspberry Pi, Embedded Video Processing, Motion Detection, Face Recognition, Hand Detection, OpenCV, Lightweight Deep Learning, Smart Surveillance, IoT-enabled Vision Systems

I. INTRODUCTION

In the past decade, the need for security has been increased for various reasons. The next evolution of security cameras is to annotate video and local coordinates of the objects that are tracked by multiplexing many video streams together in real-time. Video surveillance is playing a vital role in maintaining security in privately or socially like banking, in house monitoring, finance etc. This can be easily monitored by using our regular personal computer. Over the years back all of us have used analog cameras that are connected to coaxial cables. As the years passed to improve the performance and increase the compatibility of the user we have switched to digital systems, now to IP- based systems. In practice it is not easy to detect and track any moving object. The omni directional cameras are generally used for this purpose. Mobile cameras can also be used for this purpose. In this implementation, we use the Raspberry pi. The recorded video data captured is collected and compressed into MPEG format then transferred to the network. This signal is processed by the client monitoring the system; it will reframe, restructure and recompose the video images. Using a coaxial cable cost more than this wireless video monitoring system.

II. RELATED WORKS

In existing system, there are no methods for detecting the human face. In case of any confidential matters the authorities must know the person. For example any person who is trying to harm the nation, then compulsory authorities need to know that person's face and his movement. But there are no specific

Doi:10.48047/ijiee.2022.12.4.10

technologies to know all these. Existing system disadvantages are we cannot know the person, We are not able to get the movement of the person. A few literatures checked various probabilities of acquiring the required data from concept of localization. This is made a bit easier with the IOT platform to get the details. This system is also provided with IOT to get a localized environment values. The previous literature review also has some disadvantages that this type of system cannot be used where there is no GSM facility. It also has some more problems like the signals cannot be properly transferred nor received properly in case of cloudy atmospheric conditions. There may be a delay in communication networks which is presently being used as a portable black box that detects the blast items. Similar few more applications like accident detection in case of blast of mines, station alarm system, mine tracking system using GPS. This helped in developing the paper to design a module that can easily identify the particular object and analyze its local parameters.

III. SYSTEM DESIGN

This paper deals with the Raspberry pi 3B minicomputer which is used to design an embedded surveillance system. This project mainly focuses on face detection using various motion detection methods and face detection algorithms. After selecting the specific method for testing and implementation, it is programmed using C/C++ to calculate and reduce the processing power of the embedded minicomputer. Using image processing algorithms, the image captures from the externally places web cam are captured and the all the passive scenes are processed. These algorithms analyze the images in real time and extract the information about the objects which are moving and if any motion has occurred it saves that video sequence. The captured images are processed using various computer based vision approaches. All these can be initially tested using open source software which contains over optimized algorithms for image and video analysis and manipulation. The open CV libraries contain all the functions that can be operated and optimized on the Raspberry Pi platform.

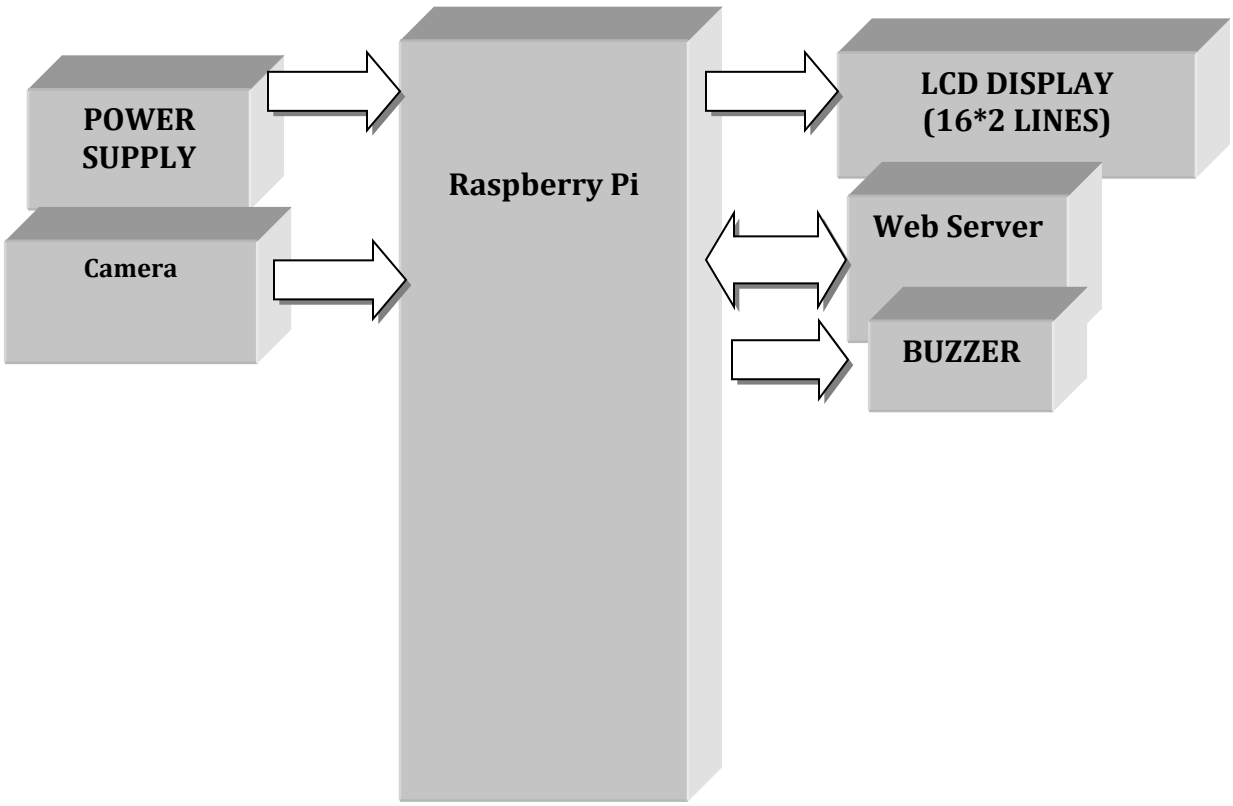

Fig 1 Block Diagram of the Proposed system

Doi:10.48047/ijiee.2022.12.4.10

A web camera is installed to record the live video. The proposed model uses Raspberry Pi Model B with a Logitech C300 which includes serial port (USB) webcam. An external hard drive is used to accumulate data. Here 16GB secure digital high limit card is used. Web association is given by means of CAT6Ethernet link. The camera is attached to the stepper engine so that it captures all pictures. Raspberry Pi is arranged with USB designed camera module and an outside screen to see the captured video. The Raspberry Pi comes in a variety of form factors for different use cases. Below is the board layout of the Raspberry Pi 3. While this layout is slightly different from previous models of the Raspberry Pi, most of the connections are the same.

IV. IMAGE PROCESSING WITH OPENCV:

Visual information is the most important type of information perceived, processed and interpreted by the human brain. Image processing is a method to perform some operations on an image, in order to extract some useful information from it. An image is nothing more than a two dimensional matrix (3-D in case of coloured images) which is defined by the mathematical function f(x,y) where x and y are the two co-ordinates horizontally and vertically. The value of f(x,y) at any point is gives the pixel value at that point of an image, the pixel value describes how bright that pixel is, and/or what color it should be.

For grayscale images the pixel value is a single number that represents the brightness of that pixel, the most common pixel format is the byte image, which is stored as an 8-bit integer giving a range of possible values from 0 to 255. As a convention is taken to be black, and 255 is taken to be white the values in between make up the different shades of gray.

To represent color images, separate red, green and blue components must be specified for each pixel (assuming a RGB color model), and so the pixel `value‘ becomes a vector of three numbers. Often the three different components are stored as three separate `grayscale‘ images known as color planes (one for each of red, green and blue), which have to be recombinedwhen displaying or processing.

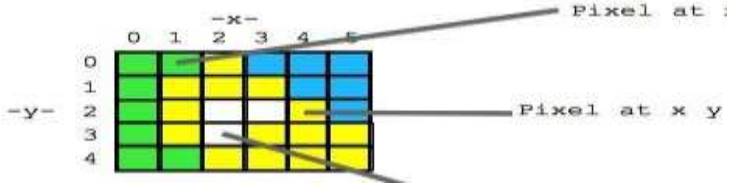

Fig 2. Representation of an image as a matrix

Now allow me to introduce the color models formally as follows, a color model is an abstract mathematical model describing the way colors can be represented as tuples of numbers, typically as three or four values or color components. When this model is associated with a precise description of how the components are to be interpreted (viewing conditions, etc.), the resulting set of colors is called color space.

Once known how the images could be represented, let‘s focus on the image processing side and specifically with OpenCV and python.

ALGORITHM:

Start
Read each and every frame from the camers i.e; live video
Capture each frame and detect the eyes in every framr
Convert the RGB image(actually it takes BGR)into grayscale image.
In gray scale we have different shades of black.
Using a threshold value convert the grayscale into binary image.
Divide the image into three sections.
Count the number of black pixels in every section.
If the number of black pixels is more in right section your logic must drive the drivertor to …right, similarly to the left.

Doi:10.48047/ijiee.2022.12.4.10

If the number of black pixels is more in middle section and verify the black pixles inupper ….and lower regions.
If the number of black pixels are more in top section your logic must drive the drivermotor …forward, similarly to the back.
If there are no black pixels detected continuously for five frames stop the motor.
Stop

V. Experimental Analysis

Here in this Detailed experimental view with flowchart has been described

Observation section includes a PC or any display and a Wi-Fi router which are shown in the fig 5.3. the use of PC/display is to execute the code in the terminal, to observe video live. The use of Wi-Fi router is to provide hotspot for the PC and the Raspberry Pi board.

On providing the power supply to the Raspberry Pi3 board the screen should look like above shown in figure. In some versions of OS to get the above screen we need to type **startx** command in the terminal shown on the screen then the above screen appears.

Terminal window of Raspberry Pi3

The fig 3 shows the terminal window used in Raspberry pi where the program can be written and executed by using appropriate commands into it. It is also used to configure any library files and can also be used to add/remove files from the directory.

Fig 3 Terminal window of Raspberry pi3

Install usb camera drivers

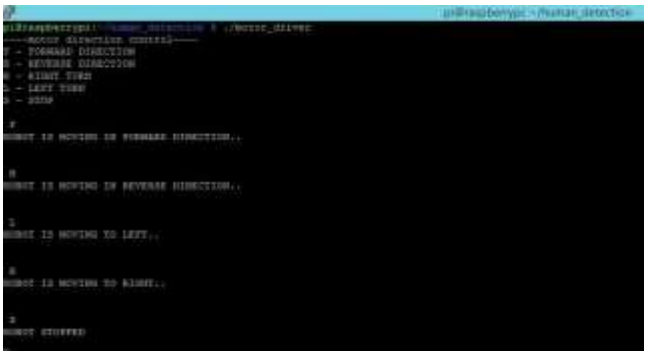

Fig 4. Output on the screen after USB cam driversinstallation

To execute the USB cam program the command should be **"./filename"**. In my case it is ./apt-get install_driver, thenthe screen looks as shown in fig 5.6 above. This program is used to run the USB cam in any direction.

To get the live streaming of the location through USB cam the instruction should be **"sudo motion"**. On successfulexecution we can see live video by entering the IP address inthe browser as shown above fig 5.

Doi:10.48047/ijiee.2022.12.4.10

Output screen of overall project operation

Fig 5. Output Screen along with video streaming

After successful operation the final output screen is shown in the fig is used to enter the commands to execute the code and also to operate the robot by giving the right directions. The window on the right side is used to monitoring the location using the USB cam by entering the IP address in any of the browser.

VI. CONCLUSION

The project has been successfully done with a designed embedded real-time video monitoring system. Linux operating system is used for streaming purpose. It is available at portable, low cost, easy to use and maintain. It can easily be upgraded to any high end application. Here the web browser is based on MJPG streamer for streaming captured video from camera placed in remote location. The MJPG streamer is cross-compiled and loaded in to the Raspberry pi board to act as a web streaming server. The regularly captured videos are transmitted to the server and these are being processed to obtain the actual data whenever it is needed.

REFERENCES

1. D.Jeevanand, K.Keerthivasan, J.MohamedRilwan, P.Murugan "Real Time Embedded Network Video Capture And SMS Alerting system" International Journal of Communication and Computer Technologies, 2014.
2. Yong-ik Yoon, Jee-ae Chun, Tracking System for mobile user Based on CCTV. Information Networking (ICOIN), 2014 International Conferenceon, Phuket, 10-12 Feb. 2014, pp. 374-378.
3. KavithaMamindla, Dr.V.Padmaja, CH.NagaDeepa, and "Embedded Real Time Video Monitoring System Using Arm", IOSR Journal of Engineering (IOSRJEN) Vol. 3, Issue 7 (July. 2013), ||V6 || Page(s) 14-18.
4. Viren Pereira, VandykAmsdemFernandes, JunietaSequeira, Low Cost Object Sorting Robotic Arm using Raspberry Pi. Global HumanitarianTechnology Conference - South Asia Satellite (GHTC-SAS), 2014 IEEE,Trivandrum, 26-27 Sept. 2014, pp. 1-6.
5. Zhou Zhe, "ARM-Based Embedded Linux System For Wireless Video Monitor applications", Department of Information Engineer, Beijing University of Post and Telecommunication, Beijing (100876), Page(s):1 -4.
6. G. SenthiKumar, S.Ragu, N. Siva Kumar, "Embedded Video Surveillance with Real time Monitoring on Web", International Journal of Mathematics Trends and Technology- May to June Issue 2011 Page(s):46-49.
7. Yimamuaishan.Abudoulikemu, Yuanming Huang, Changqing, A Scalable

Doi:10.48047/ijiee.2022.12.4.10

Intelligent Service Model for Video Surveillance System Based on RTCP .Signal Processing Systems (ICSPS), 2010 2nd International Conferenceon (Volume:3), Dalian, 5-7 July 2010, V3-346 - V3-349.

8. C. Bahlmann, Y. Zhu, Y. Ramesh, M. Pellkofer, T. Koehle, A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. IEEE Intelligent Vehicles Symposium, Proceedings,2005, pp. 255-260.
9. Adrienne Heinrich, Dmitry Znamenskiy, Jelte Peter Vink, Robust and Sensitive Video Motion Detection for Sleep Analysis. Biomedical andHealth Informatics, IEEE Journal of (Volume:18 , Issue: 3) , 2168-2194,20 September 2013, pp. 790-798.

Doi:10.48047/ijiee.2022.12.4.10