**CHAPTER III**

**HARDWARE AND SOFTWARE COMPONENTS**

**3.1 Resources**

For an embedded real-time surveillance system to be utilized for effective monitoring and alerting, the system has to have at least three functions. These functions are: detection, image processing and alert mechanism. This Raspberry Pi based security system is thus composed of mainly two parts. These are design hardware and design software

**3.2 Design Hardware (System Modules Set Up and Configuration).**

* Personal computer,
* webcam
* Microphone
* USB powered speakers.

The above-mentioned hardware components are essential to setup the system configuration for the desired process

**3.2.1 Personal Computer**

The Pc with Windows 10 was chosen to implement the project. It has been chosen to reduce the cost of surveillance system. A surveillance can also be built using the personal computer. This may be a desktop or a laptop.

**3.2.2 Programming for the surveillance system**

To enable communication with the outside world, the system has to be programmed with a suitable programming language. These languages include Java, FOTRAN, Pascal, Python, C, C++ etc. Each language has its own syntax and semantics. The Pc can be programmed using any of these languages but for purposes of this project, Python will be of great importance to study. It is provided by default through and thus optimum operation of the Pi can be achieved.

**3.2.3 The webcam**

Some laptops has its own camera module that can be used. Rather than using this camera, which has low qualities, we can just use a USB camera which is might be available everywhere at less cost. The configuration of such a camera is very simple. Just connect the USB camera to one of the USB ports of the Pc. After doing that, we can check whether the camera works well or not by capturing images using the camera option in the Pc. At first, we need to install the camera driver software.

****

**Figure 3.3: Pi Camera module**

**3.3 Enabling the WebCam**

After these configuration settings, the system was rebooted. This was done to ensure that the camera was allocated enough space in memory. The camera takes 5MP image and has a resolution of 342 by 342. It stores the images in pictures as image.jpg.

**3.4 Interfacing I/O Devices**

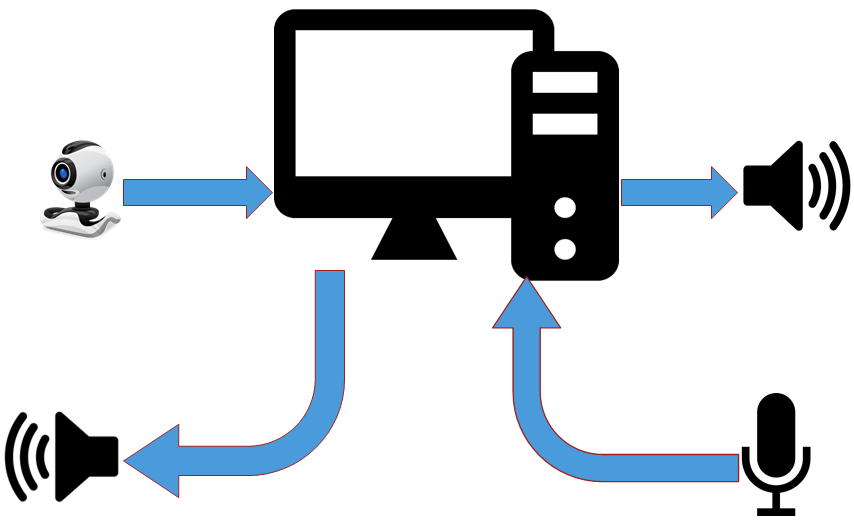
As far as our system is concerned Keyboard mouse Speakers microphone and Monitors are the input and output devices. The above mentioned devices are interfaced Pc by means of USB interface and HDMI ports.The webcam also connected through the USB ports

**3.5 Hardware Architecture**

The main ingredient is going to be the Personal Computer which is provided with facilities to give audio outputs to an sophisticated authority in case of any alert to be issued. The pc has been connected with camera modules to import images from real world as well as a display monitor connected with the help of a HDMI port in order to view the ongoing processes inside the pc.

The keyboard and mouse are connected through the USB available ports so that one can efficiently communicate with the system, the pc must be provided with a valid internet connection in order to give the predefines audio outputs through the speakers.

The entire system modules were interfaced together as shown below.



**Figure 3.7: Overall hardware architecture**

**3.6 Software design**

The whole of the software part has been developed with the help of python programming language and many supporting modules were included to support the execution of the project. Some of the modules installed through the **pip install** command on command window whereas others are already inbuilt python IDLE interpreter. The needy modules are imported inside the coding to ensure no errors are stated by the interpreter.

**3.6.1 Open CV**

OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCV, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform.

Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

OpenCV has been used to training of the face datasets as well as the recognition of face later on. By importing the CV module from open cv software image processing becomes possible in python.

**3.6.2 GTTS LIBRARY**

The Google Text To Speech (GTTS) is used by many companies and organizations for converting a text to an audio. Python provides an Google Text To Speech module in its standard library called **GTTS** that implements the text to speech conversion process. You can learn all about the Google Text To Speech by reading the gtts documentation on the Internet. However, the full specification is outside the scope of this article.

Gtts provide the audio output to the authority about the information of the individuals who appeared Infront of the webcam.

**3.6.3 OpenCV – Python Video Processing**

OpenCV is a very powerful tool used to analyse images and video files. The basic processing procedure to be followed is detailed in the flowchart below. Thresholding as a technique of image processing was chosen for the implementation of motion detection and tracking in video streams. has paved a way for the transfer of files between the clients and the server by the means of internet medium making it possible to communicate between the different clients to transmit images and information.

The choice to script using OpenCV – Python was because Python on its own does not support video processing. There is so far no video processing library in Python. OpenCV thus provided the necessary platform to achieve image processing.

The following flowchart was used for this implementation 

**Figure 3.8: Video processing in Python**

**3.6.4 Pseudocode**

i. Start the camera and set to capture video stream.

ii. Grab a frame from the video stream. If frame is grabbed, continue with the process. Else stop. Initialize the frame as the current frame.

iii. Convert the captured frame to grayscale. Then do Gaussian blurring to remove noise in the grey image.

iv. Capture another frame and repeat step two above.

v. Check for pixel threshold if enough to detect the face.

vi. Draw a rectangle around the region where the face is detected.

**3.6.5 Speech Recognition Library**

The Speech Recognition library is downloaded from the module name speech Recognition. This help the authority to converts his/her speech to text. The converted Text are the commands through which the system process is accessed. The detailed processes of the Speech Recognition module can be learned through the Speech Recognition documentation.

**3.6.6 Twilio Library**

The Twilio library is downloaded from the module name twilio. This helps to send the information of the known individuals to the authority in short message service. This uses a twilio number to send a short message service to the authority’s mobile number which is predefined for the message receiving number.