1. INTRODUCTION

In today's conventional surveillance system recording is done continuously, thus occupying huge storage space and power consumption even when there is no motion in area monitored. Aforesaid parameters decrease the efficiency of traditional CCTV surveillance system.

1.1 Objectives and Motivation

In today's digital world, surveillance is everything. Surveillance using microcontroller is most efficient way of detection of theft through various sensors. Authentication by password allows known users with valid authenticated credentials to enter the house and this system is also secured by user notification. Raspberry Pi is connected to a USB and converts analog to digital signal, the process is initiated by capturing image and saving the images in the cloud server. Images are sent to the android users as a trigger alert to deal with intruders.

Home security is one of our responsibility, but though at times there are possibility of intrusions. Nowadays, Wireless Monitoring is among the cutting-edge researches in the field of International Intelligent Building. For real-time surveillance of the home security, the intelligent remote monitoring system was developed based on ZigBee technology and GSM / GPRS network. The system can notify user with abnormal images and warning messages through MMS and SMS and receive remote instruction. The exploration of a variety of sensors and the improvement of system reliability guaranteed efficient home security. The experimental result shows that the intelligent home system can attain remote surveillance with high reliability.

Collecting information about the movement and position of specific targets in the sensing environment is one of the most important goals of this system.

1.2 Internet of Things

The “Internet of things” (IoT) is a rapidly growing topic grabbing attention both in the workplace and outside of it. IOT is a concept with potential to impact how we live and work. There are a lot of complexities around the “Internet of things”, related to technical and policy information.

With the rapid reduction in cost of connecting, more devices are being created with Wi-Fi capabilities and sensors built into them to connect to Broadband Internet and more technology costs are going down. All of these things are creating a “perfect storm” for the IoT.

This is the concept of basically connecting any device to the Internet (and/or to each other). This includes everything from cell phones, lamps, washing machines, coffee makers, earphones, wearable devices and almost anything else. This also applies to components of machines, for example a jet engine of an airplane or the drill of an oil rig. The analyst firm Gartner says that by 2020 there will be over 26 billion connected devices… That’s a lot of connections (some even estimate this number to be much higher, over 100 billion). The IoT is a giant network of connected “things” (which also includes people). The relationship will be between people-people, people-things, and things-things.

The new rule for the future is going to be, “Anything that can be connected, will be connected.” Say for example you are on your way to a hospital for regular check-up; your car could have access to your calendar and already know the best route to take. Due to heavy traffic your car might change your appointment timing. Another example is that your alarm clock wakes you up at 5 a.m. and then notifies your coffee maker to start brewing coffee for you.

On a broader scale, the IoT can be applied to things like transportation networks: “smart cities” which can help us reduce waste and improve efficiency for things such as energy use; thus, helping us understand and improve how we work and live.

IoT will allow virtually boundless connections to take place, many of which we cannot even think of. It’s not hard to see how and why the IoT is such a hot topic today, but it also has many challenges. Security is the most important among them. Imagine billions of devices being connected together and someone being able to hack into your toaster and thereby get access to your entire network. The IoT also opens up companies all over the world to more security threats. Privacy and data sharing is another issue. As billions of devices are connected, they will produce tons of data, and storage and analysis of this data is another big issue. Figure 1 shows how the components of our project combine together to form Internet of Things.

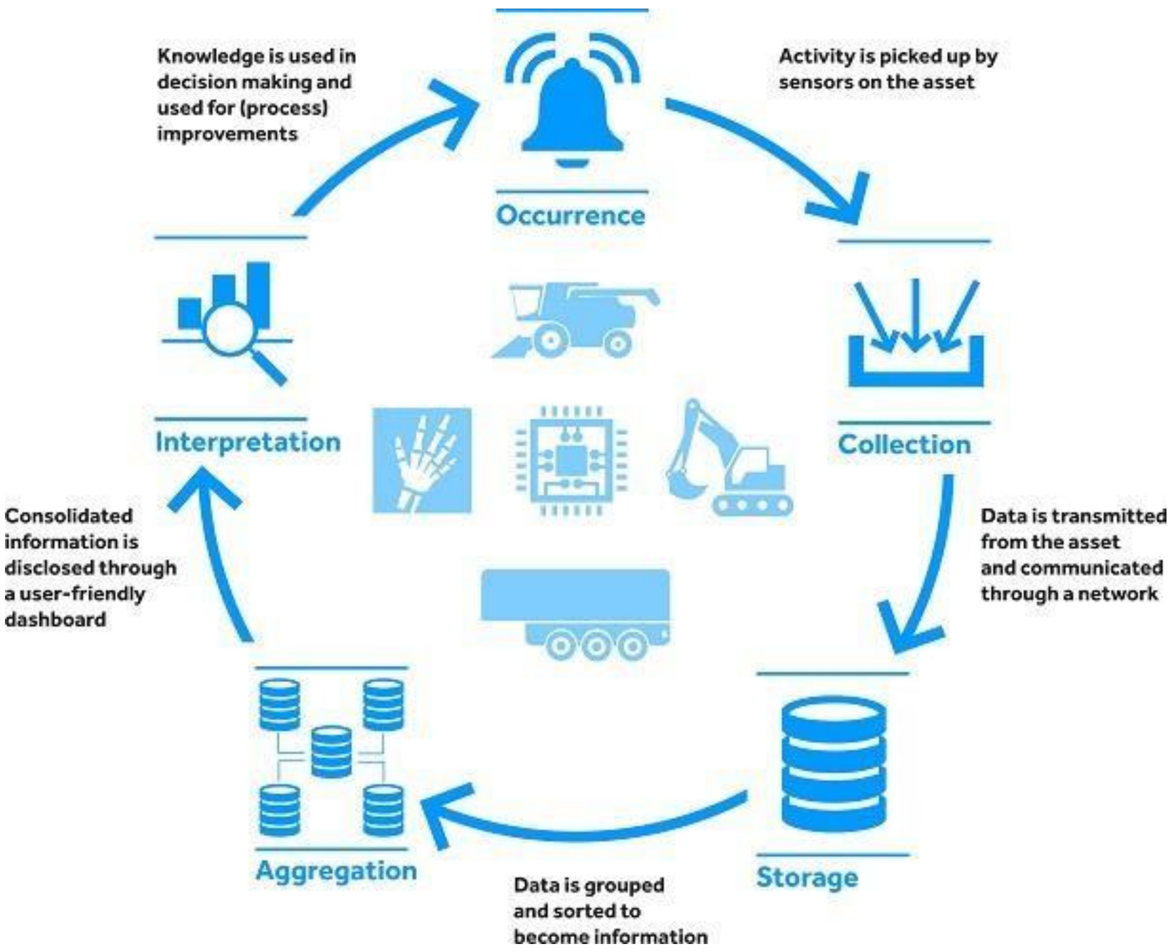


Figure 1: Components of our project form an IoT

1.3 Closed-Circuit Television (CCTV)

Video surveillance is the use of video cameras to transmit a signal to a specific place, on a limited set of monitors. The signal is not openly transmitted like in broadcast television, though it may employ point to point (P2P), point to multipoint, or mesh wireless links. Almost all video cameras fit this definition but the term is most often applied to those used for surveillance in areas that may need monitoring such as banks, casinos, airports, military installations, and convenience stores.

In industries, CCTV system is used to observe parts of a process from a central control room, also under inaccessible conditions. CCTV systems may operate continuously or only as required to monitor a particular event. A more advanced form of CCTV, utilizing digital video recorders (DVRs), helps recording for a long time, with a variety of quality and performance options and extra features (such as motion detection and email alerts). More recently, decentralized IP cameras, some equipped with megapixel sensors, support recording directly to network-attached storage devices, or internal flash for completely stand-alone operation. Most CCTV systems may record and store digital video and images to a digital video recorder (DVR) or, in the case of IP cameras, directly to a server, either on-site or offsite.

Figure 2: CCTV Camera

The retention of the images produced by CCTV systems depends on cost factor. The amount and quality of data stored on storage media is subject to compression ratios, images stored per second, image size and is affected by the retention period of the videos or images. DVRs store images in a variety of proprietary file formats. Recordings may be retained for a preset amount of time and then automatically archived, overwritten or deleted.

Closed-circuit television (CCTV) cameras can produce images or recordings for surveillance purposes, and can be either video cameras, or digital still cameras.

2. Concepts Review

The concepts being considered in the development of the proposed system have been discussed in the following section.

2.1 Motion Detection

Motion detection is the process of detecting a change in the position of an object relative to its surroundings or a change in the surroundings relative to an object.

- Background Subtraction Algorithms can be used to identify moving objects.
- A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. GMMs are commonly used as a parametric model of the probability distribution of continuous measurements or features in a biometric system. GMM parameters are estimated from training/learning data using the iterative Expectation-Maximization (EM) algorithm from a well-trained prior model. [3]

2.2 Background Subtraction

Background subtraction, or a synonym for foreground detection, is the method in the domains of computer vision and image processing in which an image's foreground extraction takes place for future processing applications (object recognition etc.). Normally regions of interest in an image are objects (text, trees, cars etc.) present in the foreground. Post the image pre-processing stage (that may comprise of post processing, image denoising etc.), this technique may be used for object localisation.

Detection of moving objects in the videos recorded from immobile cameras generally uses background subtraction. The main principle of this approach is to use the difference of the current frame and the background image (which acts as a reference frame) to detect the moving objects. Background subtraction is carried out if the image under consideration is present in the video stream. This technique sets the base for numerous other applications in the field of computer vision, like traffic monitoring or human action recognition.

Generally, basis of background subtraction is a static background hypothesis, but this is often not useful in real conditions. For indoor scenes, background variations are caused by the animated images or the reflections formed on screens. Similarly, when considering outdoor scenes difficulties are faced due to precipitation, wind or brightness variations caused by weather.

To overcome the recurring motions due to clutter, handle illumination changes and scene changes over long time we need a robust background subtraction algorithm. [5]

The following analysis considers the function of video sequence,

$$V(x,y,t)$$

where x and y variables depict the pixel location

t is the time dimension

Ex.- V(2,4,6) is the pixel intensity at (2,4) pixel location of the image at t = 6 in the video sequence.

- Using frame differencing

The first step in a motion detection algorithm is the segmentation part in which the moving objects (or foreground) are demarcated from the background. The most basic way for its implementation is to consider an image as background, denoted by B, and then consider the frames captured at time t, denoted by I(t), for comparison with the background. Here image subtraction method of computer vision enables to find the moving objects by simple arithematic calculations i.e. for every pixel in the frame I(t), subtract the pixel value, P[I(t)], with the pixel value of the equivalent pixel in the background image, P[B].[5]

The corresponding mathematical representation is

$$P[F(t)] = P[I(t)] - P[B] \tag{2.1}$$

This difference image, as shown in equation (2.1) will depict change of intensity for the pixel locations that have undergone a changed between the two time frames with a time difference of t. In spite of the fact that we have supposedly eliminated the background, the method is applicable only for the case when we have all static background pixels and mobile foreground pixels.

In order to obtain an improvement in subtraction, we introduce a threshold "Threshold" on this difference image.

$$|P[F(t)] - P[F(t+1)]| > \text{Threshold} \tag{2.2}$$

Equation (2.2) tells that intensities of the pixels of the difference image are 'threshold' i.e. the value of the Threshold filters the pixels of the difference image. The precision of this approach relies on how fast the movement is in the scene. Higher thresholds may be required for faster movements.

- Mean filter

An array of prior images is averaged, for obtaining the image comprising of only background. If the background image is computed at any given instant t and N represents the number of prior images in the array then this averaging would take the average value of the corresponding pixels for all the preceding images in the array. Here, N has dependency on the amount of motion in the video and the video speed (i.e. the number of images per second in the video). After driving the background, B(x,y,t), it is then subtracted from the image at time t, V(x,y,t) and the magnitude of the result is compared with the threshold value. Therefore, the equation (2.3) represents foreground.

$$|V(x,y,t) - B(x,y,t)| > Th \tag{2.3}$$

where Th is the threshold.

In the same manner, median can be used in the calculation instead of mean.

NOTE: Use of global and time-invariant Thresholds (i.e. same Th value for every pixels in the image) may limit the accuracy of the above two approaches.

2.3 Face Detection

Face detection refers to the computer technology with the help of which identification of human faces within digital images takes place. The term is also used to refer to the psychological process with which humans are able to locate the faces in a visual scene.

In other words, it can be said face detection is a special case under the object class detection. In case of the object-class detection, the aim is to locate all the objects belonging to a class and find their sizes in a given image. For examples: cars, pedestrians and trees.

Algorithms of face-detection focus on finding the front view of human faces. Face detection is similar to image detection, which involves bit by bit matching of image of a person and if there's any change in the facial features leads to invalidation of the matching process. [1]

2.4 Applications of Face Detection

Facial detection has variety of applications spread across various fields. A few of them have been discussed below to understand the usefulness of the technique.

Facial recognition

Perhaps the most common use of face detection algorithms is in the field biometrics. Face detection is often used as a part of a facial recognition system. Facial recognition is used extensively image database management, human-computer interface and video surveillance.

Marketing

Marketers are also making the use of face detection. A webcam is fitted in an advertising television and using this the face that passes by is detected. The system then finds the age range and gender of the face. Based on the information that is computed, advertisements can be shown, specific to age range/gender.

Photography

Face detection has enabled to incorporate the feature of autofocus in recent digital cameras. Also, another recent development has been to click a photograph after smile detection.[1]

2.5 Haar Cascade

Haar-like features are a type of digital image features employed in object recognition. They have named such due to the intuitive similarity they share with Haar wavelets. They were useful in the first real-time face detector.

Previously, working with image intensities (i.e., values of the RGB pixel for every individual pixel in image) alone lead to computationally expensive calculation. Viola and Jones adapted the idea of using Haar wavelets and developed the so-called Haar-like features.[4] A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. For example, let us say we have an image database with human faces. It is a common observation that among all faces the region of the eyes is darker than the region of the cheeks. Therefore, a common haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object (the face in this case).

In the detection phase of the Viola–Jones object detection framework, a window of the target size is moved over the input image, and for each subsection of the image the Haarlike feature is calculated. This difference is then compared to a learned threshold that separates non-objects from objects. In order to describe an object with sufficient accuracy a large number of Haar-like features is necessary because a single Haar-like feature is only a weak learner or classifier (its detection quality is slightly better than random guessing). In the Viola– Jones object detection framework, the Haar-like features are therefore organized in something called a *classifier cascade* to form a strong learner or classifier. [2][4]

Calculation speed of a Haar-like feature acts as the key advantage over most other features. By using integral images, calculation of a Haar-like feature of any size can be done in constant time (approximately 60 microprocessor instructions for a 2-rectangle feature).

- Rectangular Haar-like features

A general rectangular Haar-like feature is determined from the difference of the sum of pixels of areas inside the rectangle, which may be at any location and with any scale within the original image. *2-rectangle feature* is the name given to such a modified feature.

3-rectangle features and 4-rectangle features have also been defined by Viola and Jones. Certain characteristics of a particular area of the image are indicated by their values. Each feature type suggests the existence (or non-existence) of particular characteristics in the image, which could be edges or changes in texture. Border between a dark region and a light region is indicated by a 2-rectangle feature.

- Fast computation of Haar-like features

Integral images is one of the contributions of Viola and Jones, which uses summed area tables. Matrix with the same size of the original image are used to form two-dimensional lookup tables, and these matrices are known as Integral Images. The sum of all pixels located on the upper-left region of the original image (in relation to current element's position) is contained in each element of the integral image. Hence, Sum of rectangular areas in the image can be computed, at any desired position or scale, using just four lookups: If points A, B, C, D are a part of the integral image I, as shown in the Figure 3, then

$$\text{Sum} = I(C) + I(A) - I(B) - I(D) \tag{2.4}$$

Here Equation (2.4) represents the sum of rectangular area. Each Haar-like feature may require more than four lookups, depending on how it is defined. Six lookups are required by Viola and Jones's 2-rectangle features, 3-rectangle features need eight lookups, and nine lookups are required for 4-rectangle features. [4]

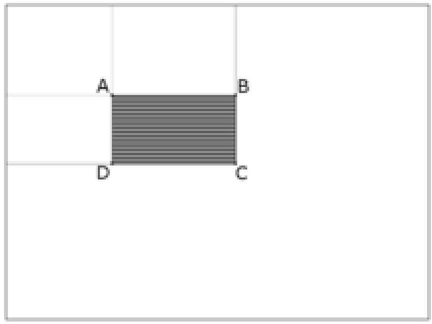


Figure 3: Finding the sum of the shaded rectangular area

- <u>Tilted Haar-like features</u>

The concept of a tilted (45°) Haar-like feature was introduced by Lienhart and Maydt. It attempts to enhance the detection of objects in images by increasing the dimensionality of the set of features. This was beneficial, as some of these features can define the object in a improved way. For example, the existence of an edge at 45° can be indicated by a 2-rectangle tilted Haarlike feature.

The idea to a generic rotated Haar-like feature was extended by Messom and Barczak. Practical problems hamper the use of Haar-like features at any angle, even though the idea is mathematically sound. In order to increase speed, low resolution images are used by detection algorithms which increase rounding errors. Rotated Haar-like features are not commonly used for this reason.

2.6 Face Detection using Haar Cascade

Object Detection using Haar feature-based cascade classifiers is a competent object detection approach proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001.[4] A cascade function is trained from a lot of positive and negative images in this machine learning based method. Objects in other images are then detected using this cascade function.

Here we work with face detection. Initially, a lot of positive images (images of faces) and negative images (images without faces) are required by the algorithm to train the classifier. Then we extract features from it. For this, Haar features shown in Figure 4 below are used. They are similar to our convolutional kernel. Subtracting sum of pixels under white rectangle from sum of pixels under black rectangle gives us each feature, a single value.

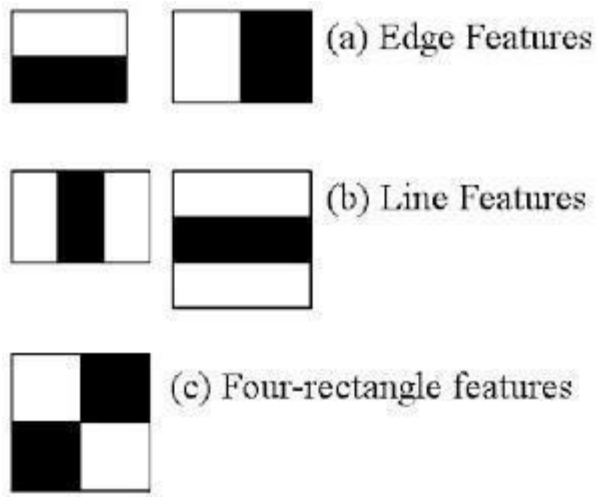


Figure 4: Haar Features

Now all possible sizes and positions of each kernel are used to calculate many features. (Which needs a lot of computation. Even a 24x24 window results over 160000 features). We need to get the sum of pixels under white and black rectangles for each feature calculation. To solve

this, integral images was introduced. It simplifies calculation of sum of pixels, however extensive the number of pixels may be, to an operation involving just four pixels.
It makes things super-fast.

For this, we apply each and every feature on all the training images. For each feature, the best threshold which classifies all the faces to positive and negative is found. But obviously, errors or misclassifications will be there. Features with minimum error rate are chosen, which means they are the features that best classify the face and non-face images. (The procedure is not as simple as it seems. An equal weight is given to each image in the beginning. Weights of misclassified images are increased after each classification. Then same process is repeated. New error rates are then calculated. Further, new weights are calculated. The process is repeated until desired accuracy and error rate is achieved or required number of features are identified).

Weighted sum of these weak classifiers gives us the final classifier. It cannot classify the image alone, hence it is called weak classifier, but together with others it forms a strong classifier. Even 200 features are able to provide detection with 95% accuracy, according to the paper. Their final setup had around 6000 features. (A reduction from 160000+ features to 6000 features is a big gain).

So now if an image is taken. Consider each 24x24 window. Apply 6000 features to it. Finally Check if it is a face or not. Most of the image region in an image is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, it may be discarded in a single shot. It can be left out while processing again. Instead target the region where there can be a face. This way, we can have more time to check a possible face region.

For this the concept of Cascade of Classifiers was introduced. Instead of applying all the 6000 features on a window, the features are to be grouped into different stages of classifiers and applied consecutively. (Normally very less number of features will be contained in first few stages). Discard the window if it fails the first stage. The remaining features are not considered on it. If it passes, the second stage of features are applied and the process is continued. The window which passes all stages is a face region.

The detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in first five stages. 10 features out of 6000+ are evaluated per sub-window on an average, according to the authors.

2.7 Face Detection Results

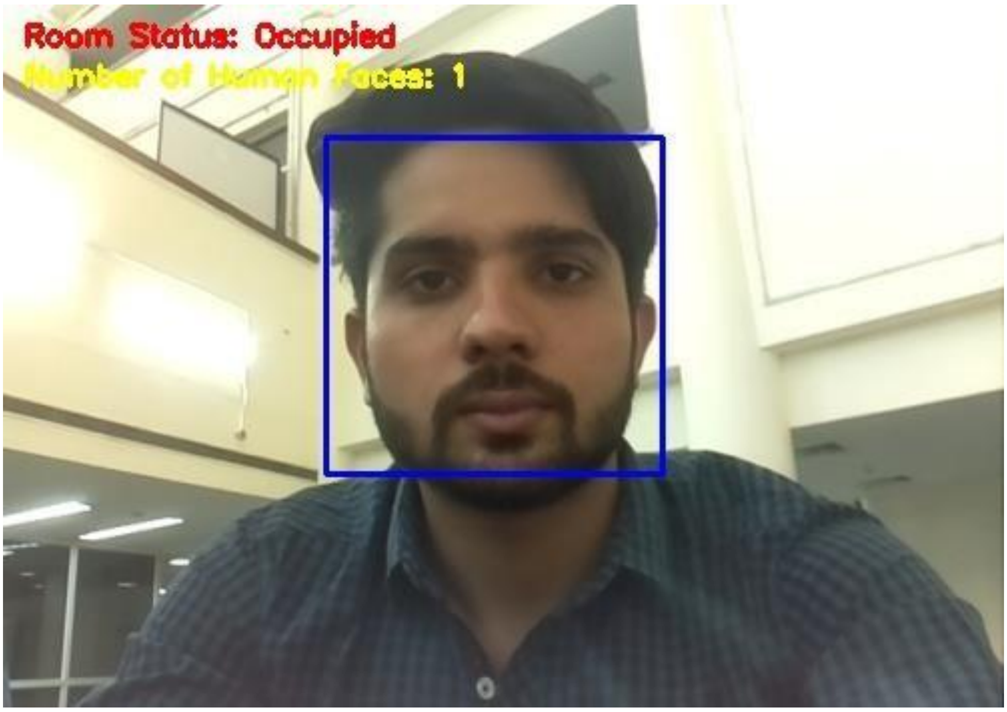


Figure 5: Face Detection Example 1

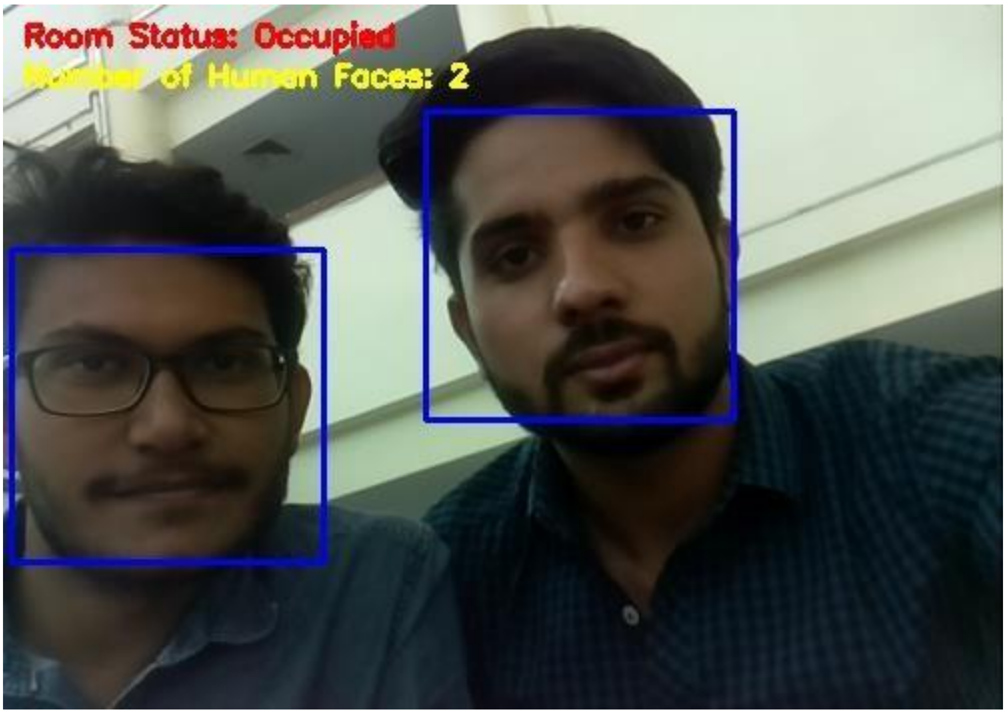


Figure 6: Face Detection example 2

3. EXPERIMENTAL PROCEDURES

The procedure of setting up Memory Efficient Surveillance System includes Raspberry Pi setup, organisation of New Out Of the Box(NOOBS) software and using it to install OpenCV on the Raspbian Jessie and finally installation of Dropbox. We begin by understanding the basics of Raspberry Pi.

3.1 Raspberry Pi

- The Raspberry Pi is a fully featured micro-computer squashed onto a circuit board measuring approximately 9cm x 5.5cm.
- It can be plugged into your TV and a keyboard, and can be used for many of the things that your average desktop does - spreadsheets, word-processing, games and it also plays high definition video.

<u>The Basics and the requirements</u>

1. Either Model A or Model B may be used.
2. Cost is $25 for model A and $35 for model B
3. HDMI (or DVI), or Composite
4. Needs an SD Card to Boot.

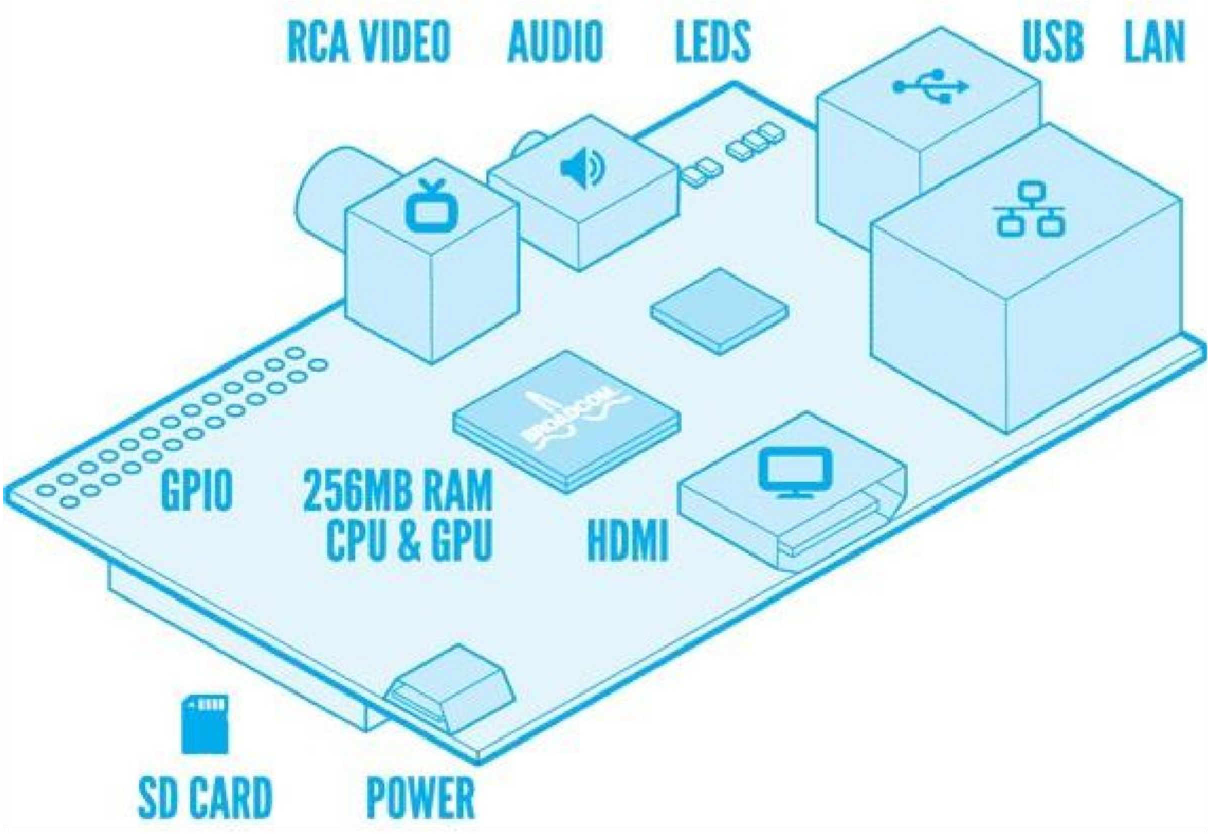


Figure 7: Raspberry Pi Board

Raspberry Pi 3, Model B

- Raspberry Pi 3 delivers more speed and 6 times the processing capacity as compared to previous models.
- The second generation Raspberry Pi is available with an upgraded Broadcom BCM2836 processor, which is a very capable ARM Cortex-A7 based quad-core processor that operates at 900MHz.
- The board can provide an increase in memory capacity to 1Gbyte.

Operating System

The operating system used is Linux on a bootable SD card. Any of the following variants may be used.

1. Fedora
2. Raspbian
3. Debian
4. ArchLinux ARM

Programming

- By default, programming language is Python.
- Any language which can compile ARMv7 may be used with the Raspberry Pi.

Major Processor Segments

All the processor segments (figure 8) are displayed below.

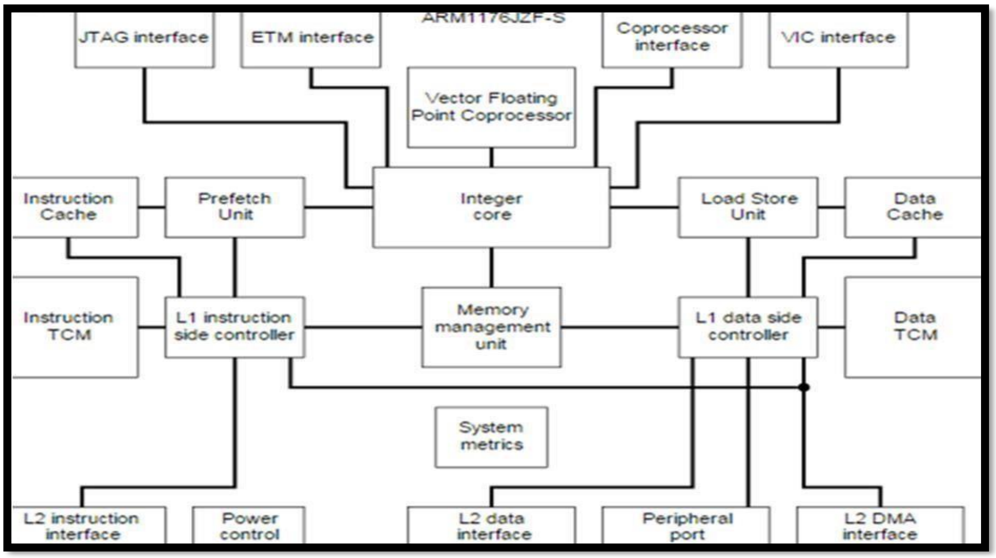


Figure 8: Raspberry Pi Processor Segments

- Integer Core can process integer values

  40 total 32-bit registers

  - Three Pipelines
  - ALU can handle arithmetic, logic, shift, as well as saturation operations
  - MAC(Memory Access Unit) can handle all Multiply operations

- Load/Store Unit is there to handle load and store operations from the Integer Core, and decouples such instructions from the MAC and ALU pipelines.

- Prefetch Unit :- to handle instruction calls

  - Utilizes both types of branch prediction
  - Combined with the Branch Target Address Cache (BTAC) results in nearly zero wasted cycles (Dynamic)
  - Also handles branches not in BTAC with normal branch predictor (Static)

- Memory Management Unit: - To organize all memory calls, to make them more efficient, so that system delays are lowered.

- Vector Floating Point Coprocessor (VFPC)

  - Core of the processor is in integer form
  - Floating Point Operations are done in the coprocessor.

- Vector Interrupt Control (VIC) Interface

  - To handles all the interrupts.
  - Deals mainly with external systems and their interrupts.
  - Faster interrupt can be allowed through Request signal.

Pi Camera

The Camera Module is a very effective accessory for the Raspberry Pi. It allows users to take still photos and record videos in High Definition (HD).

In order to integrate the Pi camera to Raspberry Pi: -

While the Pi is switched off, connect the Camera Module to Raspberry Pi's camera port, then power up the Pi and make sure the software is enabled.

1. Locate the camera port on the Pi and attach the camera (figure 9 shows the camera port):

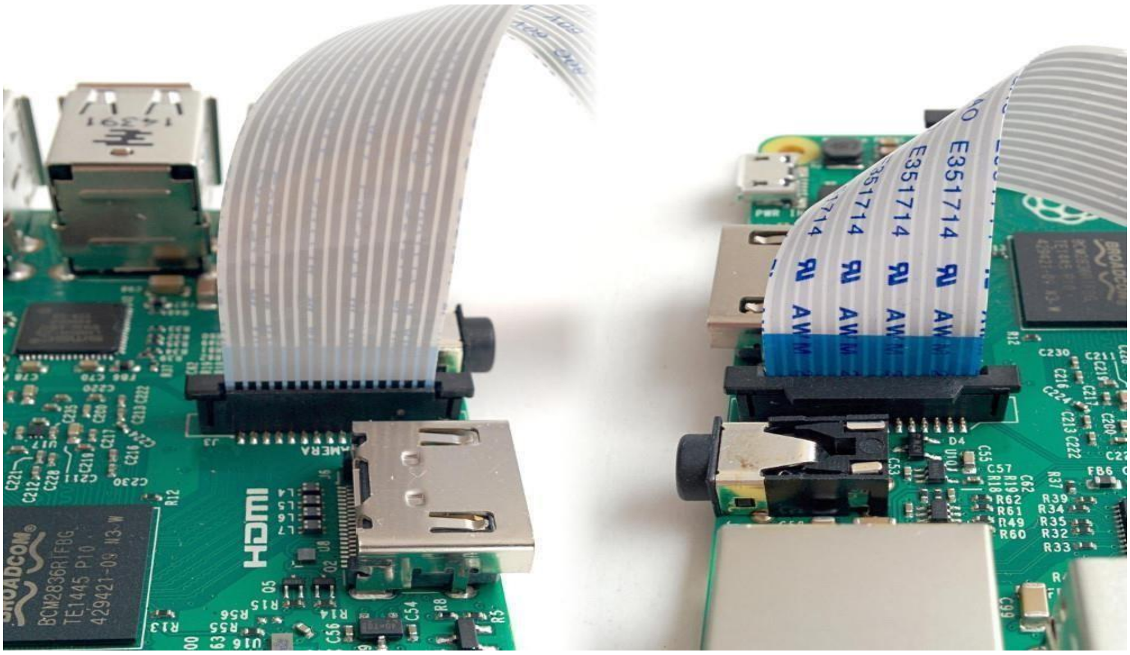


Figure 9: Connecting Camera

2. Start up the Pi.

3. From the main menu, open the Raspberry Pi Configuration Tool.

4. Ensure that the camera software is enabled (figure 10):

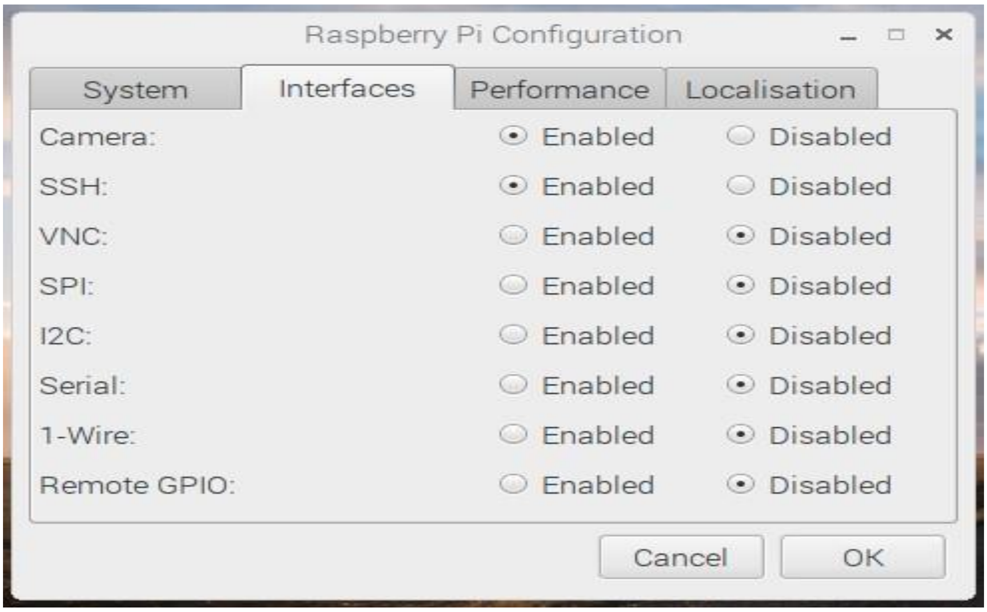


Figure 10: Enabling Raspberry Pi Camera

5. If it is not enabled, enable it. Then reboot your Pi to begin.

<u>Applications</u>

1. It can convert your Old TV in to a smart TV. (It can enable you to play Videos, 3D Games, Music, Browse Internet and much more)
2. Raspberry Pi can play the role of Full HD 1080p Media Player.
3. Its a Mini Computer which costs as low as Rs. 2,350/-
4. You can use it as a normal computer by connecting a Monitor, Keyboard and Mouse.
5. Its Graphics Capabilities are even better than Apple Products.

<u>Advantages of Raspberry Pi are:</u>

1. It is low in cost and power usage.

2. It is very simple to use
3. It has widespread use as an educational device for youths and hobbyists to learn about programming.
4. Versatile and can be used in different kinds of projects and applications
5. Compact in Size
6. It contains immense amount of resources for different projects

<u>Some of the Disadvantages are:</u>

1. It does not have a Hard Disk integrated with it. Hence if we require permanent storage of files, we need to connect a hard disk externally or use an SD card for the purpose.

2. It is not removable or swappable as the RAM is a (package-on-package) POP package on top of the (system-on-chip) SoC.

3. There is no Real time clock associated with the board. Adding a Real Time Clock is expensive.

To get started with Raspberry Pi, you need an operating system. In order to install an Operating System, NOOBS (New Out Of Box software) comes of use:

3.2 NOOBS Setup

NOOBS (New Out Of Box Software) is an simple operating system installation manager for Raspberry Pi.

Some SD cards with NOOBS preinstalled are available from many distributors and freelance retailers, such as “Pimoroni”, “Adafruit” and “The Pi Hut”. For older models of Raspberry Pi, a full-size SD card is needed; for the Pi Zero, A+, B+, Pi 2 and Pi 3 a micro SD card is required.

DOWNLOAD

Using an SD card with a minimum capacity of 8GB is recommended.

1. Visit the Downloads page using a computer with an SD card reader.

2. Click the “Download ZIP” button under ‘NOOBS (offline and network install)’, and select a folder to which you want to save it.

3. Extract files from the zip file.

FORMAT YOUR SD CARD

Before copying the NOOBS files onto it, format the SD card. This is how formatting can be done:

1. Visit the “ridgecrop” website and download their SD formatter for either Windows or Mac.

2. Install the software.

3. Insert SD card using the computer or laptop’s SD card reader to usb port of pc and make a note of the drive allocated to it, e.g. G:/

4. Open the software, select the drive folder for your SD card and format it.

DRAG AND DROP NOOBS FILES

1. Drag the files in the extracted NOOBS(New Out Of Box Software) folder and drop them onto the SD card drive.

2. All the necessary files will be transferred to your SD card.

3. Safely remove the SD card and insert it into your Raspberry Pi when this process has finished.

FIRST BOOT

1. Keyboard, mouse, and monitor cables are to be plugged in.

2. Now plug the USB power cable into your Pi.

3. Raspberry Pi will boot, and a window will appear with a list of different operating systems that you can install. It is recommended to use Raspbian – tick the box next to Raspbian and click on Install.

4. Raspbian will then run through its installation process.

5. When the installation process is complete, the Raspberry Pi configuration menu (raspi-config) will load. Here we are able to set the time and date for our region, enable a Raspberry Pi camera board, or even create users. You can exit this menu by using Tab on your keyboard to move to Finish.

LOGGING IN AND ACCESSING THE GRAPHICAL USER INTERFACE

The default login for Raspbian is username pi with the password raspberry. Note that you will not see any writing appear when you type the password. This is a security feature in Linux.

3.3 Installing OpenCV 3 on Raspbian Jessie

Installing OpenCV 3 is a multi-step process. It is a time-taking but crucial process. The proper process to install OpenCV on Raspbian Jessie is as follows:-

Step 1: Installation of dependencies

Update and upgrade any existing packages, followed by updating the Raspberry Pi firmware.

1. $ sudo apt-get update

2. $ sudo apt-get upgrade

3. $ sudo rpi-update

Timing: 3m 30s

Reboot your Raspberry Pi after the firmware update:

1 .$ sudo reboot

Installation of a few developer tools comes next:

1. $ sudo apt-get install build-essential git cmake pkg-config

Timing: 50s

Now, install image I/O packages which allows us to load image file formats such as JPEG, PNG, TIFF, etc.:

1. $ sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev

    libpng12-dev

Timing: 45s

Just like we need image I/O packages, we also need video I/O packages. These packages allow us to load various video file formats as well as work with video streams:

1. $ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev

2. $ sudo apt-get install libxvidcore-dev libx264-dev

Timing: 60s

Install the GTK development library so we can compile the “highgui” sub-module of OpenCV, which enables the display of images to our screen and builds simple GUI interfaces:

1. $ sudo apt-get install libgtk2.0-dev

Timing: 2m 50s

Various operations inside of OpenCV (such as matrix operations) can be optimized using added dependencies:

1. $ sudo apt-get install libatlas-base-dev gfortran

Timing: 50s

Lastly, we need to install the Python 2.7 and Python 3 header files so we can compile our OpenCV + Python bindings:

1. $ sudo apt-get install python2.7-dev python3-dev

<u>Step 2: Grab the OpenCV source code</u>

At this point we have all of our prerequisites installed, so let’s grab the 3.0.0 version of OpenCV from the OpenCV repository.

1. $ cd ~
2. $ wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.0.0.zip
3. $ unzip opencv.zip

Timing: 2m 30s

For the full install of OpenCV 3 (which includes features such as SIFT and SURF), be sure to grab the “opencv_contrib” repo as well.

1. $ wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.0.0.zip
2. $ unzip opencv_contrib.zip

Timing: 1m 55s

Step #3: Setup Python

The first step in setting up Python for our OpenCV compilation is to install “pip” , a Python package manager:

1. $ wget https://bootstrap.pypa.io/get-pip.py

2. $ sudo python get-pip.py

Timing: 25s

Isolated Python environments can be created using “virtualenv” and “virtualenvwrapper”,which allows you to create in an environment separate from your system install of Python. This makes it capable of running multiple versions of Python, with different versions of packages installed into each virtual environment — this solves the *“Project A depends on version 1.x, but Project B needs* 4.x” problem that often arises in software engineering.

Again, in Python community, using virtual environments is a standard practice.

1 $ sudo pip install virtualenv virtualenvwrapper

2 $ sudo rm -rf ~/.cache/pip

Timing: 18s

After “virtualenv” and “virtualenvwrapper” have been installed, we need to update our “~/.profile” file and insert the following lines at the bottom of the file:

1. # virtualenv and virtualenvwrapper

2. export WORKON_HOME=$HOME/.virtualenvs

3. source /usr/local/bin/virtualenvwrapper.sh

Any editor may be used to edit this file, such as ‘vim’ , ‘emacs’ , ‘nano’ , or any other graphical editor available from the Raspbian Jessie distribution. Again, all tht is needed to be done is open the file located at “/home/pi/.profile” and insert the lines above at the bottom of the file. Now that “~/.profile has” been updated, reload it so the changes may take effect. To force a reload of the “~/.profile” file you can

(1) logout and log back in,

(2) close your terminal and open up a new one, or

(3) just use the source command:

$ source ~/.profile

Note: You are generally required to run the source “~/.profile” command *each* time a new terminal is opened up to ensure your environment has been setup correctly.

The next step is to create Python virtual environment where computer vision work is to be done:

1. $ mkvirtualenv cv

Using *Python 2.7,* the above command creates a virtual environment named cv.

It’s important to note that the cv Python environment is entirely independent from the default version of Python included in the download of Raspbian Jessie.

In case you ever reboot your system, logout and log back in, or open up a new terminal, you’ll need to use the workon command to re-access the cv virtual environment, otherwise you’ll be using the system version of Python instead:

1. $ source ~/.profile

2. $ workon cv

You can ensure you are in the cv virtual environment by examining your command line. If you see the text *“(cv)”* preceding your prompt, *then you* are *in the cv virtual environment*:

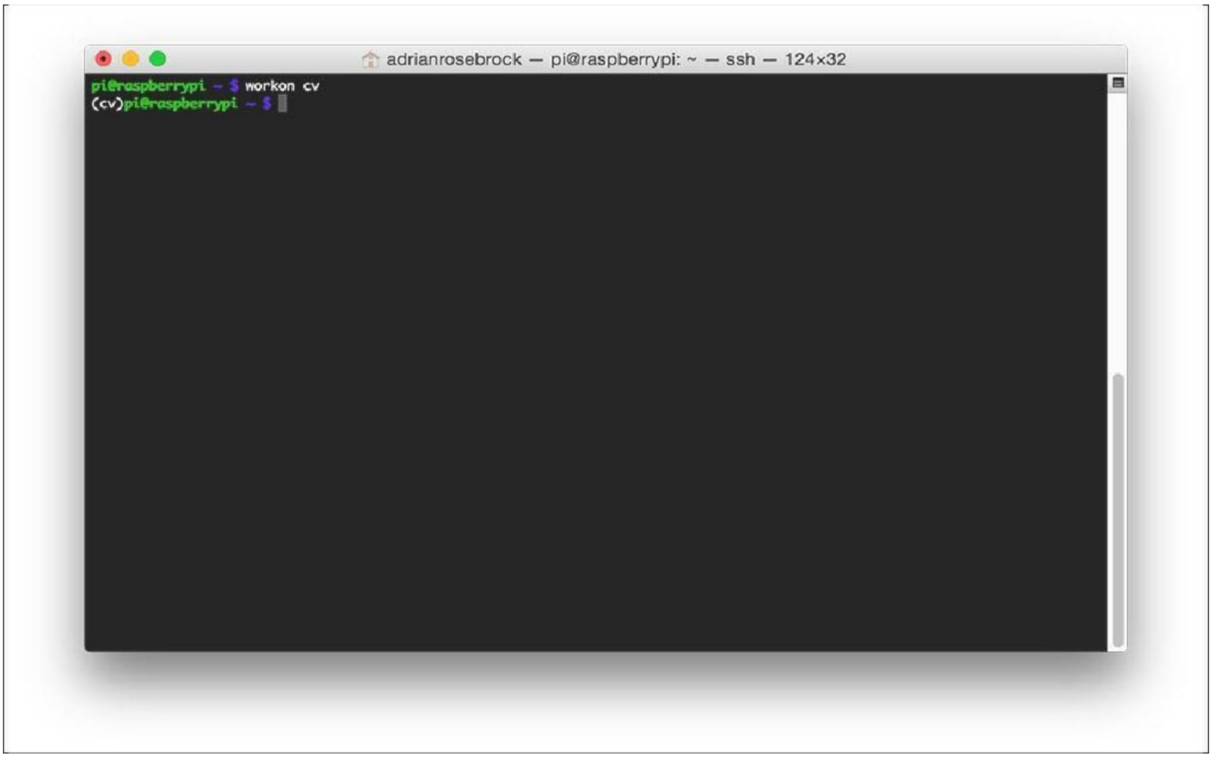


Figure 11: Make sure you see the “(cv)” text on your prompting, indicating that you are in the cv virtual environment.

Assuming that you are in the cv virtual environment, we can install NumPy, an important dependency when compiling the Python bindings for OpenCV.

1. $ pip install numpy

Timing: 16m 10s

Step #4: Compile and install OpenCV

At this point, we are ready to compile OpenCV.

First, make sure you are in the cv virtual environment:

1. $ workon cv

Followed by setting up the build:

1. $ cd ~/opencv-3.0.0/
2. $ mkdir build
3. $ cd build
4. $ cmake -D CMAKE_BUILD_TYPE=RELEASE \
5. -D CMAKE_INSTALL_PREFIX=/usr/local \
6. -D INSTALL_C_EXAMPLES=ON \
7. -D INSTALL_PYTHON_EXAMPLES=ON \
8. DOPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-
3.0.0/modules \
9. -D BUILD_EXAMPLES=ON ..

Ensuring that Python 2.7 will be used for the compile. Notice how both the Interpreter and numpy variables point to the cv virtual environment.

Now that our build is all setup, we can compile OpenCV:

1. $ make -j4

Timing: 1h 35m

The -j4 switch stands for the number of cores to use when compiling OpenCV. Since we are using a Raspberry Pi 3, we'll leverage all four cores of the processor for a faster compilation. However, if your make command errors out, I would suggest starting the compilation over again and only using one core:

1. $ make clean
2. $ make

Using only one core will take much longer to compile, but can help reduce any type of strange race dependency condition errors when compiling.

Assuming OpenCV compiled without error, all we need to do is install it on our system:

1. $ sudo make install
2. $ sudo ldconfig

Step #5: Finishing the install

For Python 2.7:

Provided you finished Step #4 without error, OpenCV should now be installed in/usr/local/lib/python2.7/site-packages :

1. $ ls -l /usr/local/lib/python2.7/site-packages/

2. total 1636

3. -rw-r--r-- 1 root staff 1675144 Oct 17 15:25 cv2.so

Note: In some instances OpenCV can be installed in /usr/local/lib/python2.7/dist-packages (note the distpackages rather than site-packages). If you *do not* find the cv2.so bindings in site-packages , be sure to check distpackages as well.

The last step here is to sym-link the OpenCV bindings into the cv virtual environment:

Step #6: Verifying your OpenCV 3 install

At this point, OpenCV 3 should be installed on your Raspberry Pi running Raspbian Jessie!

Now, let's verify that the OpenCV installation is working by accessing the cv virtual environment and importing cv2, the OpenCV + Python bindings:

1. $ workon cv

2. $ python

3. >>> import cv2

4. >>> cv2._version___

5. '3.0.0'

3.4 Installation of Dropbox

Dropbox is a software that keeps your files and your team in synchronisation. Some of the major advantages of Dropbox over other drives is that anything you save in Dropbox can be easily accessed from all of your devices. Sending files is comparatively easy. Files can be sent to people who don’t even have a Dropbox account. And with features like the “doc scanner”, “shared folders”, “offline access”, and more, collaborating with others is simple.

<u>Features of Dropbox:</u>

- Shared folders allow you to work with others
- Use the document scanner to turn receipts, whiteboards, and notes into PDFs
- Share feedback with your team by commenting on the shared files.
- Word, Excel, and PowerPoint files can be synchronised, shared and edited.

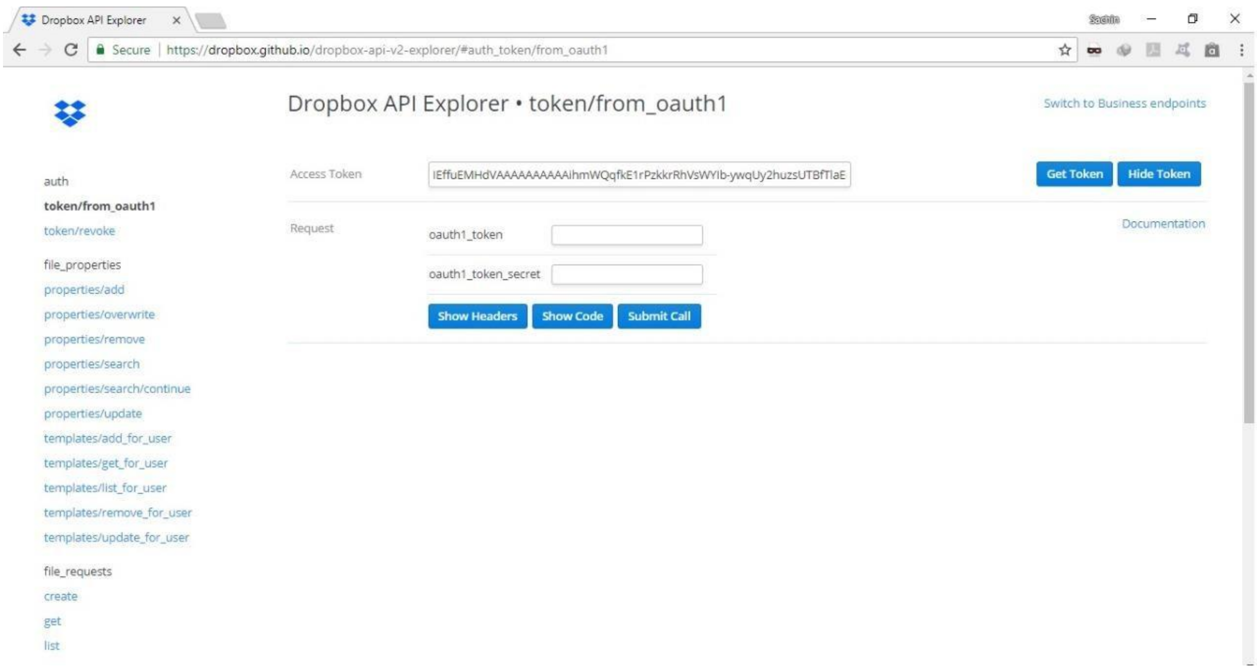


Figure 12: Getting Authorization Token for Dropbox App access

The above screenshot (figure 12) shows Dropbox App. We need to get access to Dropbox through authorization. Once that is done, the files of any form can be accessed from anywhere, shared and discussed. We have used Dropbox in order to access the photos from the installed surveillance cameras anywhere and anytime.

4. RESULTS AND ANALYSIS

The algorithms used by the proposed system along with their flowchart and the respective results have been discussed in the following section.

4.1 Basic CCTV System (Layer 1)

A basic surveillance system or first Layer is the most basic type of the analysed Surveillance systems. It captures frames every second. These frames are uploaded to Dropbox and can be viewed at any time and place using Dropbox. It uses up more memory than the much advanced Memory Efficient Surveillance System (Layer 2) and Further Memory efficient Surveillance System(Layer 3). Even though it uses up more memory and is not memory efficient it is widely used and provides the user a sense of security as it captures pictures every second and the user can view the most recent pictures at the touch of a button. Figure 13 shows a captured frame from layer 1 surveillance system.

Result

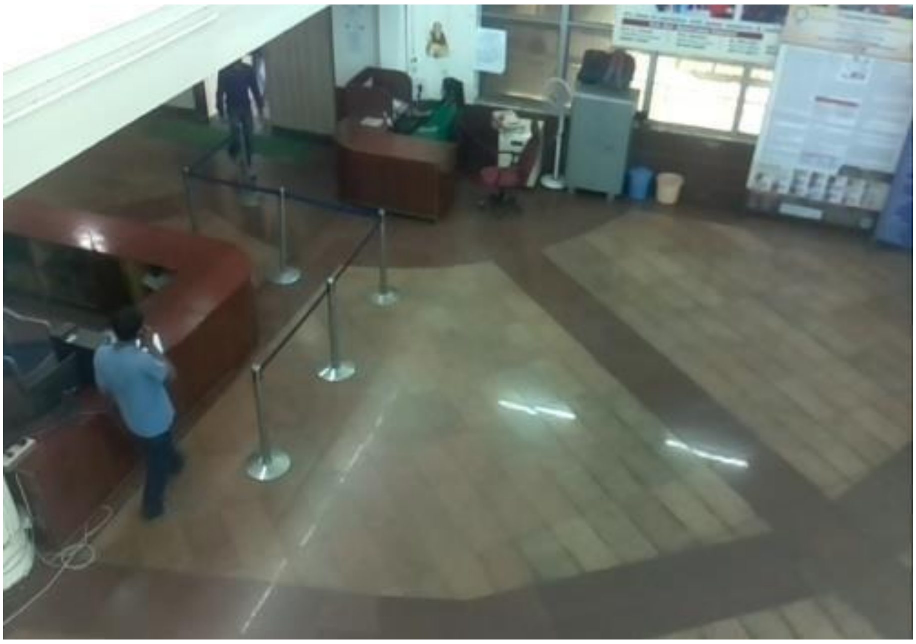
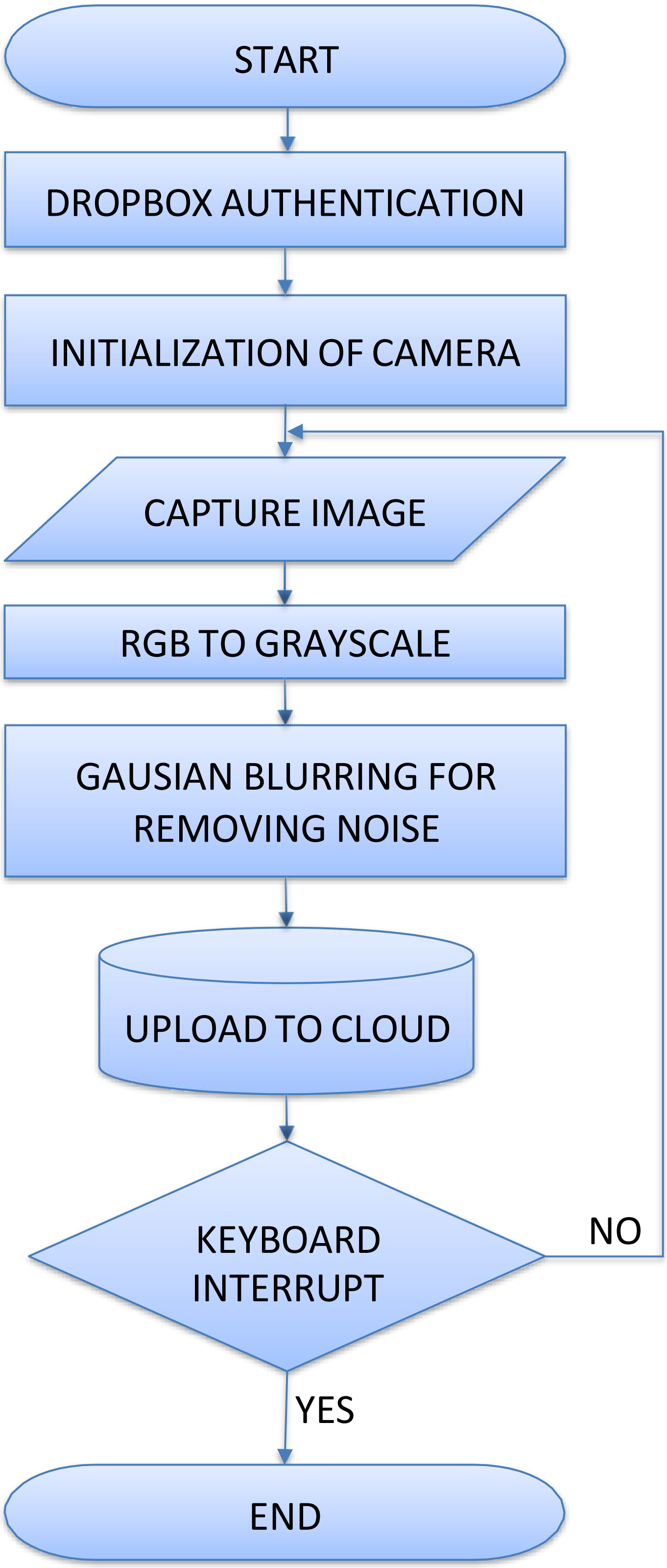


Figure 13: Result of Basic Surveillance System showing even a background frame may be captured

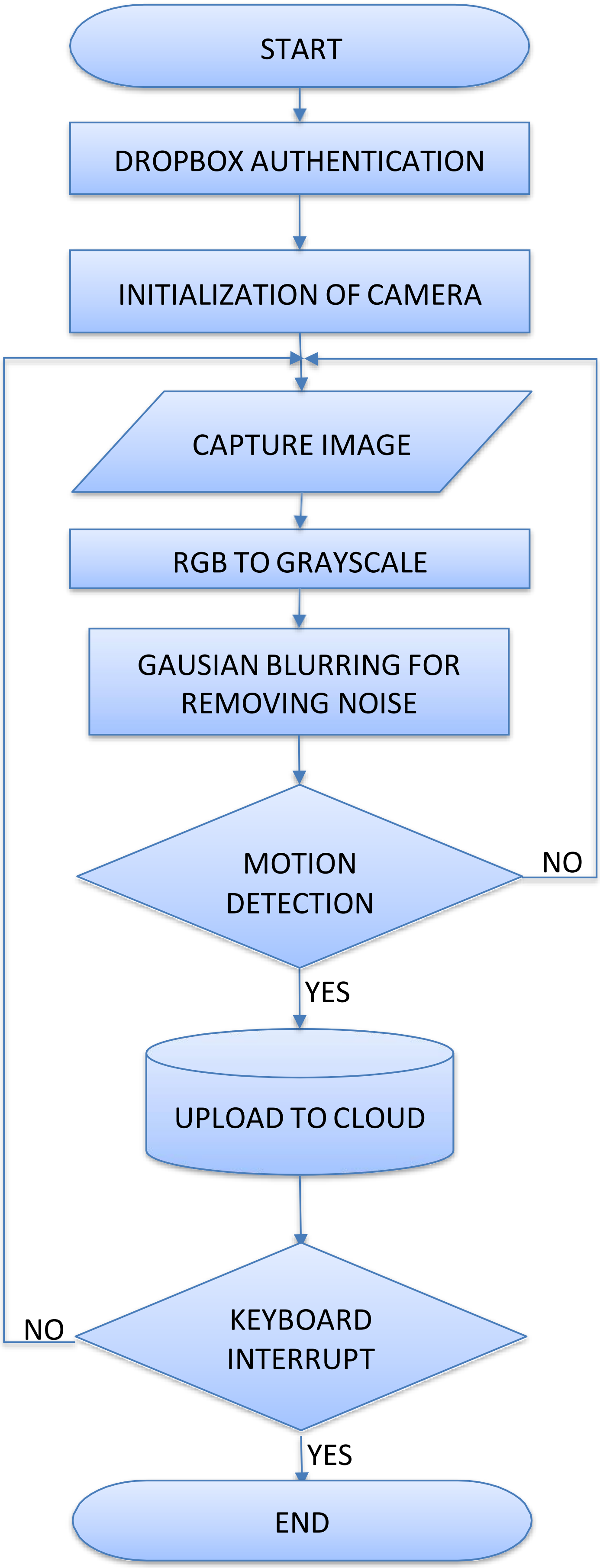

START
DROPBOX AUTHENTICATION
INITIALIZATION OF CAMERA
CAPTURE IMAGE
RGB TO GRAYSCALE
GAUSIAN BLURRING FOR REMOVING NOISE
UPLOAD TO CLOUD
KEYBOARD INTERRUPT
NO
YES
END

4.2 Memory Efficient CCTV System (Layer 2)

The algorithm that has been used by us for Memory Efficient Surveillance System is shown below. It captured images only when motion is detected in the frame, as opposed to Basic Surveillance System that captured frame at every second irrespective of the movement in the frames. Since the images are capture only when there is motion in the frames, there is massive utilisation of memory as the irrelevant images are not captured. It is an advanced form of surveillance systems that are becoming increasingly popular. If motion is detected in the frames then the image is instantly captured and stored on the Dropbox drive where it can be viewed at any time. It also saves the time of the user as the redundant frames are not captured and the viewer can easily sort through the lesser number of relevant frames. It has proved to be of great use even in burglar alarm systems and theft detection.

Result

Figure 14: Background Subtraction leads to storing frames which are significantly different than background but may lead not result in humans always

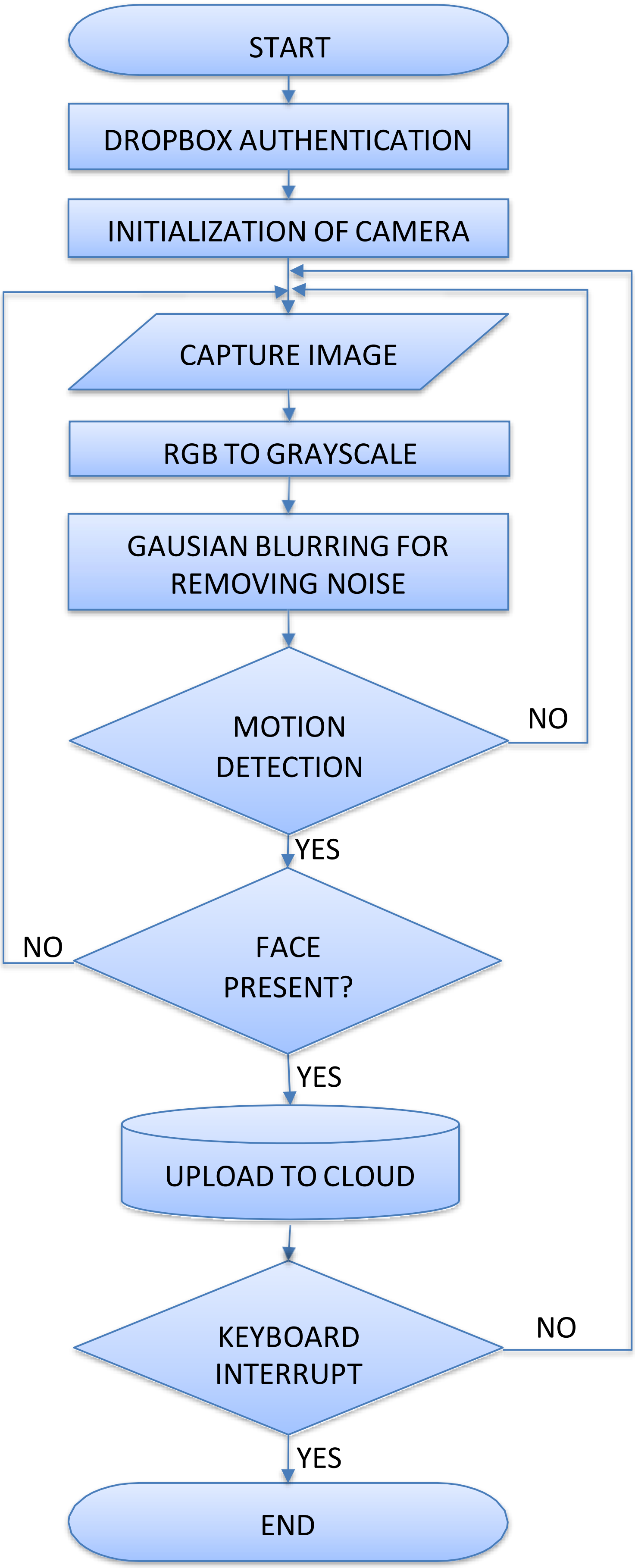

START
DROPBOX AUTHENTICATION
INITIALIZATION OF CAMERA
CAPTURE IMAGE
RGB TO GRAYSCALE
GAUSIAN BLURRING FOR REMOVING NOISE
MOTION DETECTION
NO
YES
UPLOAD TO CLOUD
KEYBOARD INTERRUPT
NO
YES
END

4.3 Further Memory Efficient CCTV System (FMESS)

Further Memory Efficient Surveillance System is the most advanced form of surveillance systems in terms of memory utilisation. The image is captured only when face is detected in the frame. Thus, the system is in constant surveillance, but an image will be captured as soon as face or faces are detected. The captured image is then uploaded to the Dropbox. From the Dropbox, the user can access all the uploaded images and see all who have been captured in the frame of surveillance.

<u>Result</u>

Figure 15: Our system leads to maximum efficiency as it focusses on human face detection


START
DROPBOX AUTHENTICATION
INITIALIZATION OF CAMERA
CAPTURE IMAGE
RGB TO GRAYSCALE
GAUSIAN BLURRING FOR REMOVING NOISE
MOTION DETECTION
NO
YES
FACE PRESENT?
NO
YES
UPLOAD TO CLOUD
KEYBOARD INTERRUPT
NO
YES
END

4.4 Comparative Analysis

The three real time scenes were monitored: Home, College Library and College Classroom, each for thirty minutes. The results were found for Basic Surveillance System (BSS), Memory Efficient surveillance system (MESS), and Further Memory Efficient Surveillance System (FMESS). Results are tabulated below:

Table 1: Number of Frames Saved in Memory for Dataset 1

System	No. of Frames saved	Percentage of BSS
BSS	420	100%
MESS	159	38 %
FMESS	52	12.44%

Table 2: Number of Frames Saved in Memory for Dataset 2

System	No. of Frames saved	Percentage of BSS
BSS	445	100%
MESS	187	42.09%
FMESS	80	18.05%

Table 3: Number of Frames Saved in Memory for Dataset 3

System	No. of Frames saved	Percentage of BSS
BSS	250	100%
MESS	87	34.93%
FMESS	21	8.383%

Table 4: Efficiency Comparission

SYSTEM	AVERAGE MEMORY REQUIREMENT (WITH RESPECT TO BASIC SURVEILLANCE SYSTEM)
MEMORY EFFICIENT SURVEILLANCE SYSTEM (MESS)	38.51%
FURTHER MEMORY EFFICIENT SURVEILLANCE SYSTEM (FMESS)	13.21%

On comparing MEMORY EFFICIENT SURVEILLANCE SYSTEM (MESS) with BSS we come to the observation that MESS uses only 38.51% of the memory used by BSS , therefore saving 61.49% of memory.

On comparing FURTHER MEMORY EFFICIENT SURVEILLANCE SYSTEM (FMESS) with BSS we come to the observation that MESS uses only 13.21% of the memory used by BSS , therefore saving 86.79% of memory.

Hence, our system is more memory efficient.

5. CONCLUSION AND FUTURE SCOPE

We can conclude that layer 3 system is more memory efficient than general systems such as the Basic Surveillance System that we analyzed, which did not use capabilities like motion detection and face detection. Further, our system uses Internet of Things concept and enables access to any location from any other location in the world. Security feature is enhanced by use of multiple cameras. The advantages of Memory Efficient Surveillance System is listed as follows:

- Real time access from anywhere in the world
- Memory Efficient
- Less computational complexity and simple to use
- Easy & Fast detection
- Highly Efficient

Applications and Future Scope

Our project has several important applications including:

Crime Detection and prevention - This is the most significant and the most obvious benefit of installing such smart surveillance security cameras. Whether you install the cameras in your home or at the workplace, you can prevent crime and avoid nuisance from occurring. It enables us to easily detect any criminal or unwanted people trying to enter.

Monitor Scenarios and Activities - It helps to observe the activities of people visiting your home and office as well as the incidents taking place at these places. This is an efficient way to observe suspicious people and keep tabs on their activities.

Industrial processes – They may be used for the surveillance of industrial processes including machines and manpower.

Use in schools – This project has widespread applications in schools and playschools where proper restrictions are needed and unauthorized person are not allowed to get close to children.

Home Security – Aside from general security, the system may be used to check upon people who visited our house in our absence.

However, as with every other system, there's a lot of work which can be done in future, including:

- Electrical appliances can be controlled and scheduled to automate their operation
- Live surveillance video of your home can be watched through your web-enabled device
- Integrating the alarm system with an intelligent lighting system so that
  - all the lights in your property will automatically switch on or flash incessantly when an intruder is detected
  - Setting your lights to randomly come on and off while you are on holiday to fool would-be burglars.

REFERENCES

Research Papers:

[1] Hjelmas, E.,Low,B.K.,2001 "Face detection: A survey.

Computer vision and image understanding" 83, 236-274.

[2] Comaniciu D, Ramesh V, Meer P, 2003, "Kernel-based

object tracking", IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 25, No. 5

[3] Yilmaz A, Javed O, Shah M, 2006, "Object Tracking: A

Survey, ACM Computing Surveys", Vol. 38, No. 4, Article 13

[4] P. Viola and M. Jones, "Rapid object detection using a

boosted cascade of simple features," in Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, 2001, vol. 1, pp. I-511-518

[5] Chen Q, Sun Q, Heng P.A., Xia D.S., 2010, "Two-stage

object tracking method based on kernel and active contour", IEEE Transactions On Circuits And Systems For Video Technology, Vol. 20, No. 4

Websites:

1. https://www.dropbox.com
2. https://www.opencv.org
3. https://www.pyimagesearch.com/2015/10/26/how-to-install-opencv-3-on-raspbian-jessie/
4. https://www.raspberrypi.org
5. https://www.raspberrypi.org/help/noobs-setup/2/