MEMORIAL UNIVERSITY OF NEWFOUNDLAND
DEPARTMENT OF ENGINEERING AND APPLIED SCIENCE

Moving Object Detection Using Camera

ENGI 9805

Winter 2021

Design Project

By:

Jagbir Singh Aulakh

(201990225)

Vraj SanjayKumar Patel

(201990796)

Table of Content

List of Figure

Abstract

"Motion Detector Technology Using Web Camera" is the title of the project. The aim of this project, as stated in the title, is to use a webcam to decide the function of motion detection. A web camera is connected to the device in the programme. These webcams can be used as a motion detector. If an object crosses in front of the webcam, the webcam senses the movement and stores the picture in a particular position. So, this report will describe about the motion detection of the object that is coming in between the camera and it detects with respect to the frames like grayscale frame, delta, and threshold frames.

Keywords: Movement object, threshold frame, delta frame, Gray-scale and Gaussian Blur filter, Bokeh graph.

I. Introduction:

A common subset of computer vision is video analytics. One of the most critical facets of video analytics for our machine is target recognition. Via the camera, it must track objects and moving objects. The same principle can be applied to detecting motion of any form. When a camera or a webcam is looking at the feed, any moving target, whether it is an animal or a person, must be detected by the detector. A definition like this can be useful in several contexts. Protection is a perfect example of this. Over time, research has flowed into developing new ideas and refining or expanding existing research for improving object and motion detection efficiency. This project detects moving objects and motion in front of a computer webcam, as described.

So, the project would focus on object detection, as well as any moving object detection, such as an animal, in which we would use detailed techniques and methodologies to detect objects and develop a module for a method that will detect motion of that object in front of the webcam.

II. Objectives:

For the norm, a static picture frame would be used at first. The camera will begin monitoring the predetermined location where the video will be captured. In that area frame, all sorts of motions will be registered. The camera will simply capture the film, but filtering will be performed in the background to determine if there is motion in the frame.

The entire video would be analysed in RGB, and then a filtered bokeh effect graph would be created to display the motion observed in the recorded video's timeline. Considering these considerations, Python emerges as a superior forum for algorithm design and execution. It is a high-level, interpreted programming language that supports a variety of programming paradigms.

III. Design of the System:

In this project, we will develop an application that will detect moving objects using our computer's webcam and store the details about how long the moving object was visible in front of the camera on a datasheet that will generate a graph. So, that this section discusses the general approach to our proposed object detection and moving object detection algorithms, as well as how to analyse data using a time frame. Below is the flow chart that is basically discussed about the whole working of the software by block and block wise.

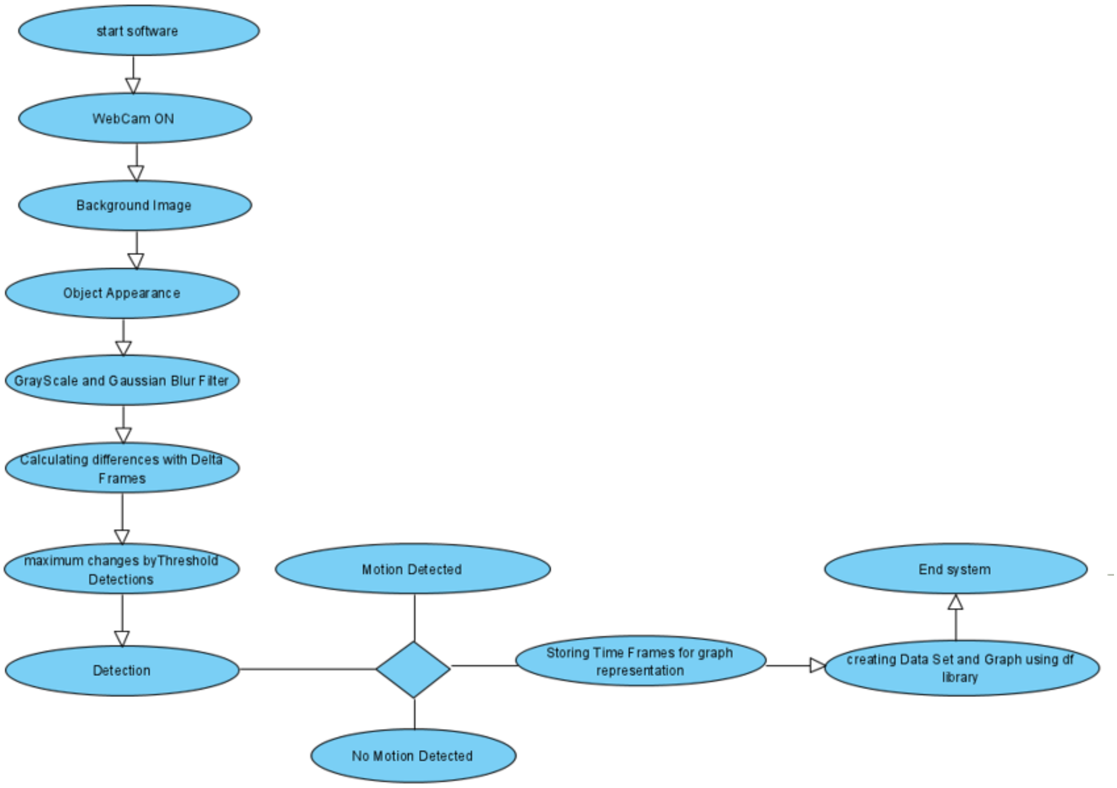

start software
WebCam ON
Background Image
Object Appearance
GrayScale and Gaussian Blur Filter
Calculating differences with Delta Frames
maximum changes byThreshold Detections
Detection
Motion Detected
No Motion Detected
Storing Time Frames for graph representation
creating Data Set and Graph using df library
End system

Figure 1. Flow Diagram of Motion Detection of Object with Camera

In the above diagram, if software is start, the web camera is on then it starts taking the visual of the scene of the view and determine the background images and object appearances. After that the system will apply the two mathematical concepts. First apply the grayscale to make the image as Gray with pixel and after that apply the gaussian filter to reduce or mitigate the unwanted noise from the image.

After that we compare first grayscale frame and current grayscale and makes a new frame called delta frame and from the delta frame with THRESH_BINARY function to find the total change in the movement of the object. If the motion is detected then store the frames for graph representation and create the data set and graph using the df library with bokeh graph and at last, the system is terminated. Moreover, the below figure is determining the clear representation of the internal technical system of the systems with respect to the frames:

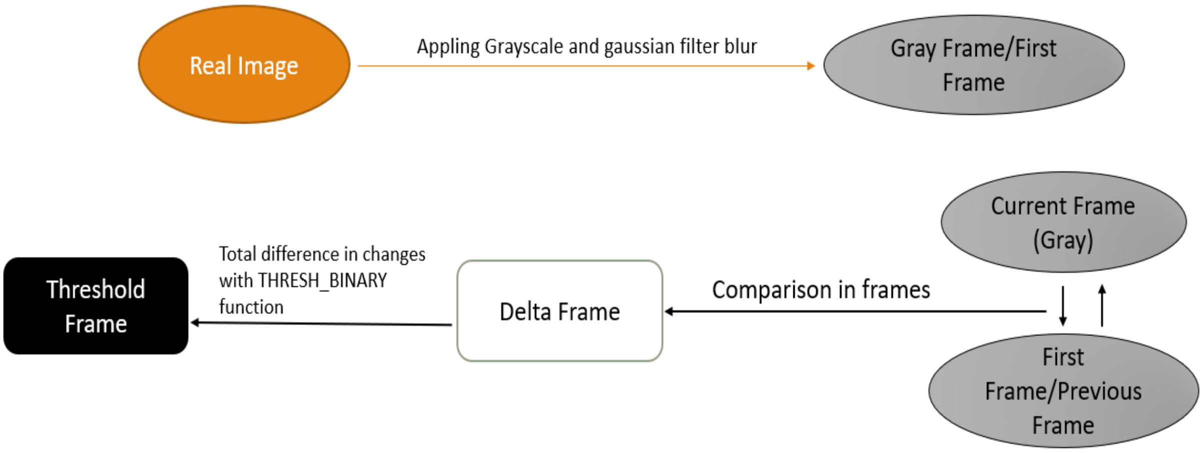

Real Image
Appling Grayscale and gaussian filter blur
Gray Frame/First Frame
Current Frame (Gray)
Comparison in frames
First Frame/Previous Frame
Delta Frame
Total difference in changes with THRESH_BINARY function
Threshold Frame

Figure 2. Representation of the system with respect to the frames

IV. Explanation of the solution:

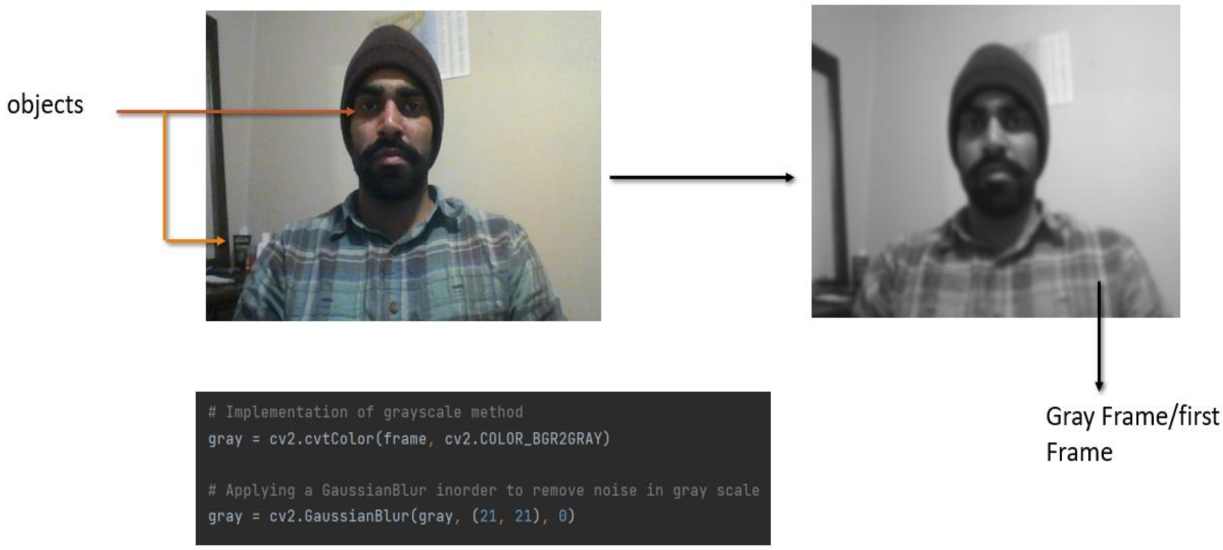
This section outlines the project's procedure and approach in depth, as well as a general block diagram of our proposed object detection and moving object detection, as well as a time frame for analysing the data. To start, we turn on our computer's webcam. So, followings are some methods that are used to implements the object movement detection with camera.

A. Gray-Scale and Gaussian Filter Implementation:

When an animal or a person enters the picture, the image is greyed out by the machine. The Gaussian blur was used to eliminate noise from the grayscale picture. A grayscale or greyscale image is one in which the value of each pixel is a single sample reflecting just a sum of light, that is, it carries only intensity information in digital imaging, computer-generated imagery, and colorimetry.

The strength of light at each pixel can be used to create grayscale images. As a result, the first frame of video capture will be saved in a vector, and the frame can be converted to a grayscale image later. The while loop will then loop through the current frames, and the process will repeat for the previous frames.

- Gray-Scale Implementation: - Coverts the Image into full Gray Scale colour
- Gaussian Filter Implementation: - Mitigate the unwanted noise from the camera.



```
# Implementation of grayscale method
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

# Applying a GaussianBlur inorder to remove noise in gray scale
gray = cv2.GaussianBlur(gray, (21, 21), 0)
```

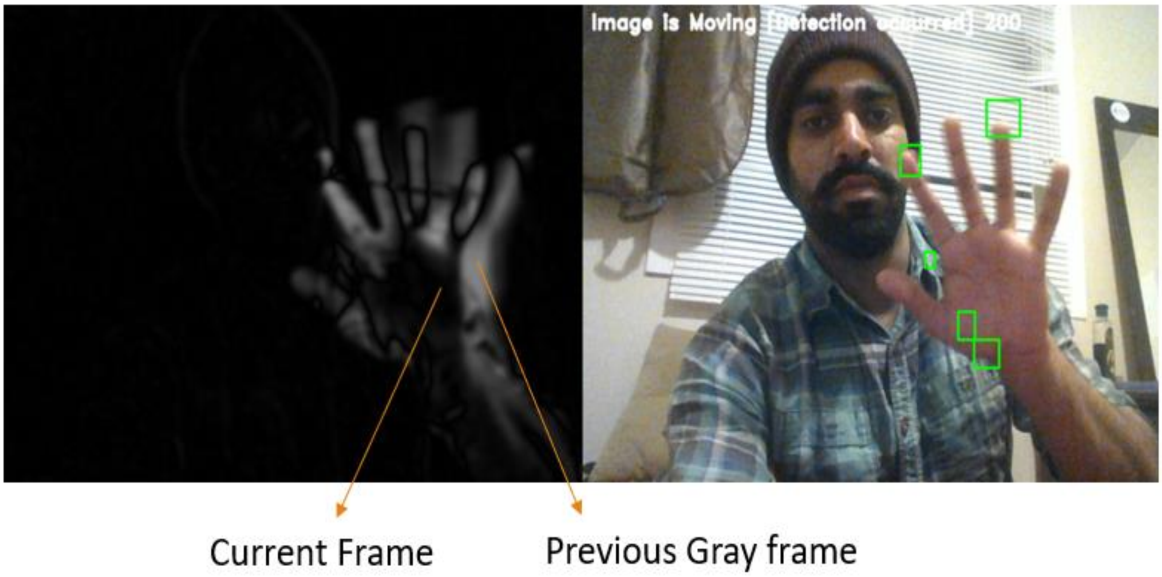
Figure 3. Gray-Scale and Gaussian filter implementation

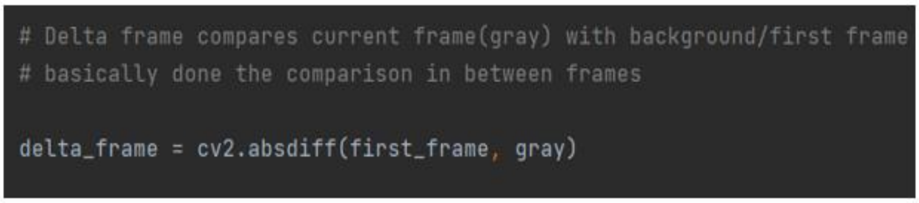
The above figure describes the first part of the solution to converts the normal real image from the camera of laptop and then apply the Gray-scale and Gaussian filter to produce the first Gray frame. As for this project, python is used and its OpenCV library function cv2.cvtColor() function is used to converts and image into Gray-scale first and then apply cv2.GaussianBlur() function is remove the unwanted noise from the system.

B. Delta frame implementations:

Since translating them to grayscale and applying the gaussian blur filter, the delta frame is applied to the disparity between those two grayscale images in the current iteration of the loop. The actual frame (grey) is compared to the background/first frame in a delta frame. It depicts the transformation from the original to the new picture and provides the following picture of delta frame.

So, basically the delta frame is basically achieved by the comparing the first two Gray frames that is implements by the Gray-Scale and Gaussian Blur Filter mathematical concepts and only change the movement or difference is captured in between the two Gray frames. Every time when any movement is capture, it is basically achieved by comparing the current Gray frame to the previous Gray Frame. To achieve the delta frame in python, we used the OpenCV library called cv2.absdiff() function to achieve the comparing between the frames.





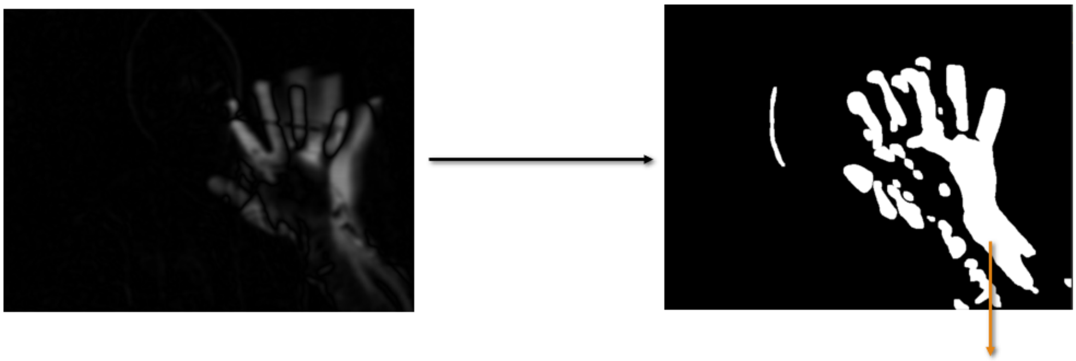
```
# Delta frame compares current frame(gray) with background/first frame
# basically done the comparison in between frames

delta_frame = cv2.absdiff(first_frame, gray)
```

Figure 4. Delta Frame Implementation

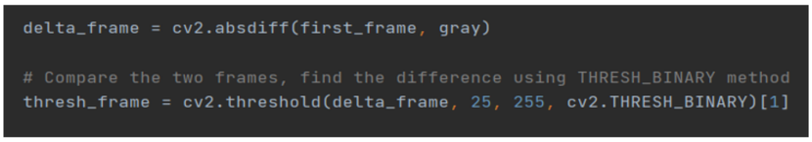
C. Threshold Frame:

The binary method employs black and white colour to determine the difference between two frames or image threshold. Thresholding is the most basic form of image segmentation. Thresholding can be used to generate binary images from a grayscale background.

The Gray-scale picture will now have a black and white frame after implementing the threshold form. Basically, if the intensity is greater than the threshold, they will be converted to black, and if the intensity is less than the threshold, they will be white, as seen in the image below:



Provide maximum change in movement by comparing the two frames.



```
delta_frame = cv2.absdiff(first_frame, gray)

# Compare the two frames, find the difference using THRESH_BINARY method
thresh_frame = cv2.threshold(delta_frame, 25, 255, cv2.THRESH_BINARY)[1]
```

Figure 5. Threshold Frame

Above figure describe the implementation of the threshold frame from the delta frame as by apply the OpenCV library function cv2.threshold() as using THRESH_BINARY function to finds the maximum change in the frames.

D. Contours and rectangular Frames:

The contours of the object would be discovered after calculating the threshold frame for the present frame. A contour is essentially the image's object's boundary. To identify or categorise objects, various representations of contours (e.g., chain code, Fourier descriptors, form context) are used. If the area of the image is smaller than 2000 pixels, it would not be called a moving object. Essentially, a contour is a curve that links all consecutive points (along a boundary) that are the same colour or strength, and it is used to establish the black and white threshold frame's boundaries.

The next step would be to draw rectangles across the contours that are wider than the minimum area after contouring. It can draw rectangles across the gestures that are large enough. This rectangle would be focused on the gestures that are large enough.

E. Saving and Storing data:

The motion of the object that entered the video frame and when it exited or stopped moving. It is going to happen. The frame captures movement as a moving object approaches the background, and it senses the time frame to store it. This data will be used to construct a graph that reflects the overall motion detection time.

V. Maximum Persistence:

The maximum persistence is basically describing as how many objects are detects at the time to capture the movement and for this project, I set the MOVEMENT_DETECTED_PERSISTANCE = 200; and below is figure that shows the maximum objects.

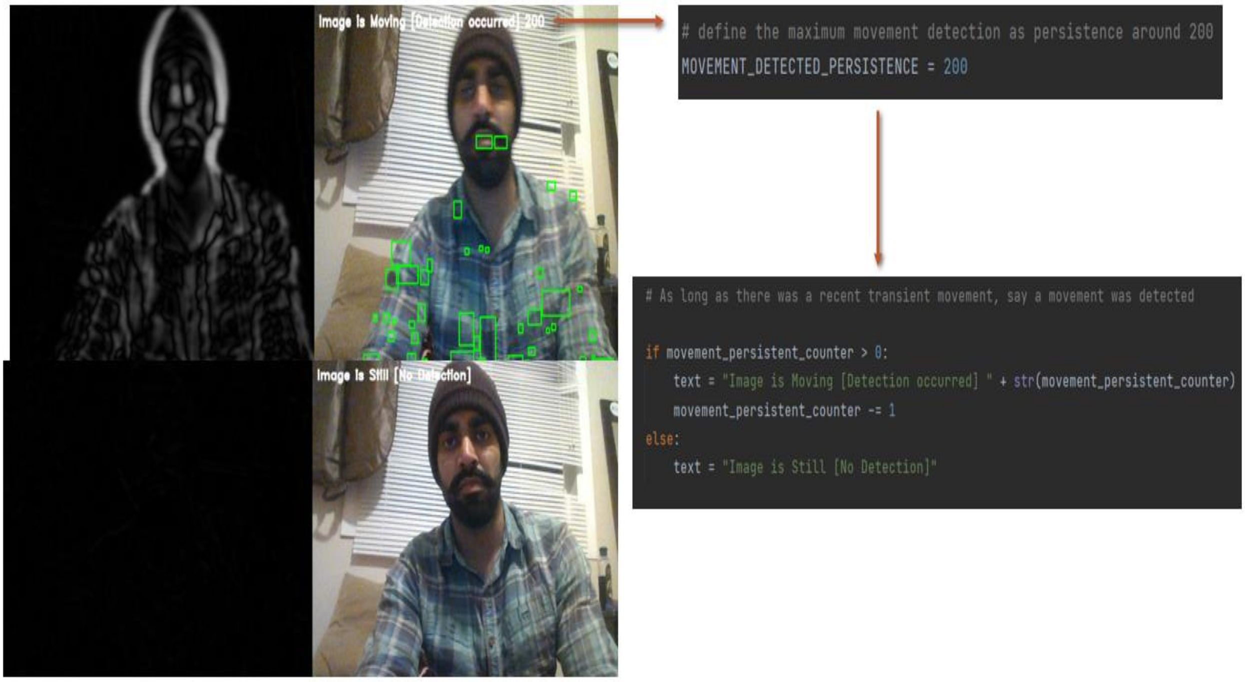


Image is Moving [Detection occurred] 200
Image is Still [No Detection]

```
# define the maximum movement detection as persistence around 200
MOVEMENT_DETECTED_PERSISTENCE = 200
```

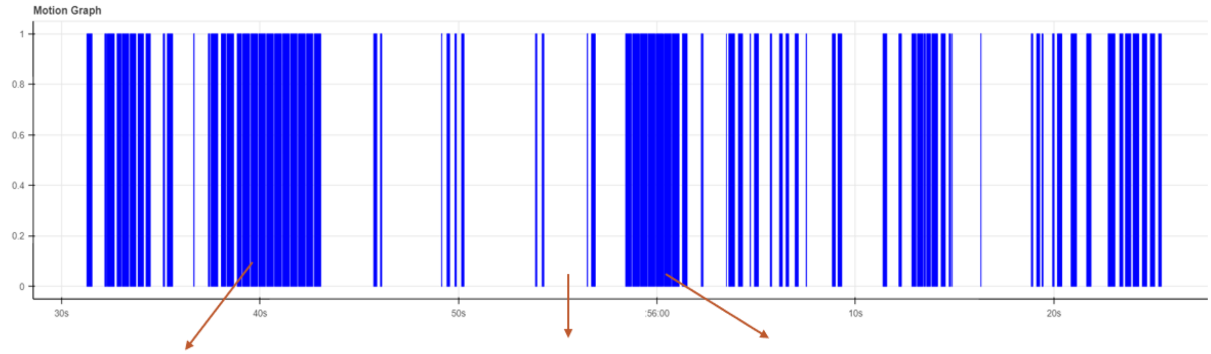
```
# As long as there was a recent transient movement, say a movement was detected

if movement_persistent_counter > 0:
    text = "Image is Moving [Detection occurred] " + str(movement_persistent_counter)
    movement_persistent_counter -= 1
else:
    text = "Image is Still [No Detection]"
```

Figure 6. Maximum detection persistence

The above figure is described the maximum object detection persistence that is shown in green symbols and if there is movement that it detected maximum of 200 objects and it shown on the top right corner of the image and if there is no detection means image is still then there is no movement detection.

VI. Data Sheet and Bokeh Graph:

The timing details of motion detection using the webcam is contained in this project, which results in a dataset. This data collection includes total motion detection of any entity, whether it is an animal or a person. A dataset would be created during the implementation to store the details about how long the moving object was in the picture. We got the following graph based on our dataset, which shows a moving object.



Motion Detected in blue

No Motion Detected

Motion Detection with maximum persistence

Figure 7. Data Set and Graph Results

The above figure is described the graph in terms of the movement is detected by the software. Whereas, in the dense blue lines, where maximum persistence is reached at 200 and where is no blue lines which means that image is still and no movement detection.

VII. Conclusion:

Today, one of the major problems in our society is social welfare and keeping a human eye on duty 24 hours a day, seven days a week is simply impractical. This project may be helpful in the defence sector as an eye for a human being. Our project is only one of the tools that will assist us in accomplishing this task. While this project is small, it does have some limitations, such as concentrating on human detection and overlooking some motion detection, such as wind impact on objects. It will, however, be enhanced and made more useful.