EXPERIMENT 3: MATHEMATICA: WEBCAM CONTROL AND APPLICATION

ABSTRACT:

In this experiment, we use the Mathematica software to develop a webcam based monitoring system and plot a parameter dynamically as a function of time as part of the monitoring.

MATERIALS REQUIRED:

A latest version of the Wolfram Mathematica software available

ABOUT MATHEMATICA SOFTWARE:

Wolfram Mathematica is a software system with built-in libraries for several areas of technical computing that allow machine learning, statistics, symbolic computation, manipulating matrices, plotting functions and various types of data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other programming languages. It was conceived by Stephen Wolfram, and is developed by Wolfram Research of Champaign, Illinois. The Wolfram Language is the programming language used in Mathematica.

THEORY:

Mathematica provides an interface for performing complex applications. This software provides us with a huge library of functions. For this experiment, we use different types of functions that help us access a webcam and monitor a region continuously, plot a parameter dynamically as a function of time and many more. In this experiment, as part of the monitoring, an alarm should be set-off when there is a change in the picture background of the webcam. We use the following Mathematica functions listed below for achieving our aim:

```
IMAQ`StartCamera[]; (* Starts the camera*)
IMAQ`StopCamera[] (*Stops the camera*)
```

```
CurrentImage[]; (*Takes a single snap*)
CurrentImage[n]; (* Takes n successive images*)
```

```
Dynamic[CurrentImage[]]; (* Note the use of the function, Dynamic.
It takes continuously images*)
```

```
Dynamic[ImageDifference @@ CurrentImage[2]]; (*Continuously takes the difference between
two images*)
```

```
ColorConvert[myImage, "Grayscale"](*Converts the image to greyscale*)
Flatten@ImageData[grayImage]; (*Converts the to 1 D data array*)

First@FindDevices[]; (* Lists out camera devices *)

DeviceOpen["Camera"];

DeviceConfigure[dev, "FrameRate" -> 1/6]; (* Configure the device  parametrs *)
Pause[2];

DeviceRead[dev]; (* Takes a snap *)
```

```
(*Some functions related to audio*)
```

```
Beep[]
```

```
Play[Sin[1500×2 Pi t], {t, 0, 1}]
```

```
Sound[SoundNote[]]
```

```
(* Note the audio interface *)
Audio[File["/Users/kasi/Documents/2021/Test21/Bella-Mira.mp3"]]
```

```
EmitSound[Sound[SoundNote[]]] (* Note the use of the function EmitSound *)
```

PROCEDURE:

1. Access the camera in Mathematica.
2. Capture the images.
3. Convert the images to Grayscale.
4. Find the difference between successive images using ImageDifference function.
5. Extract the Array corresponding to the ImageDifference function, flattening and adding all the values to get the intensity.
6. Plotting the Intensity as a function of time and trigger an alarm when the value of the intensity is greater than the threshold intensity.

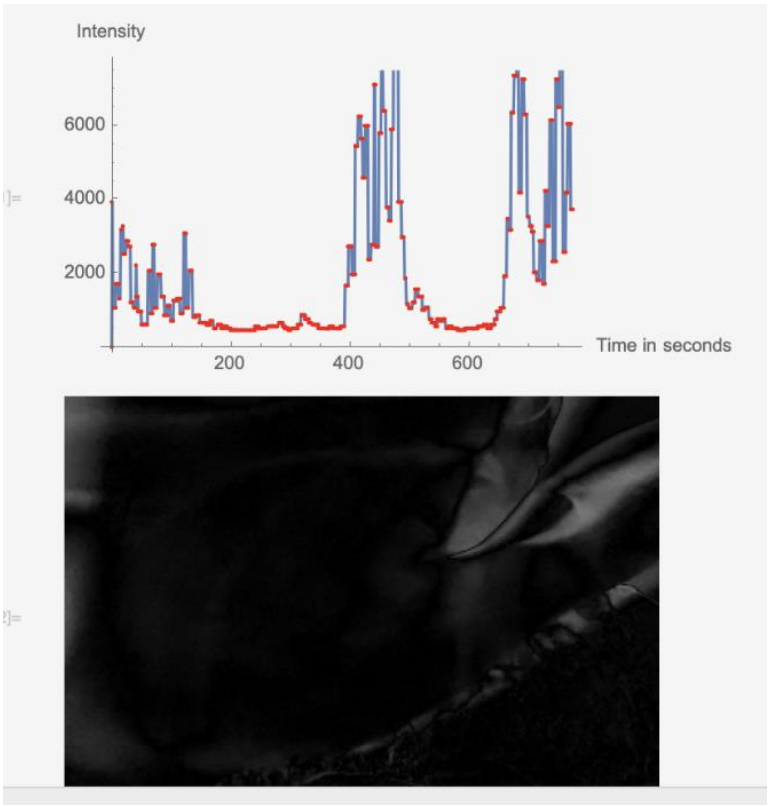
OBSERVATION AND ANALYSIS:

Based on the requirement there are two approaches in which we can program the intrusion software.

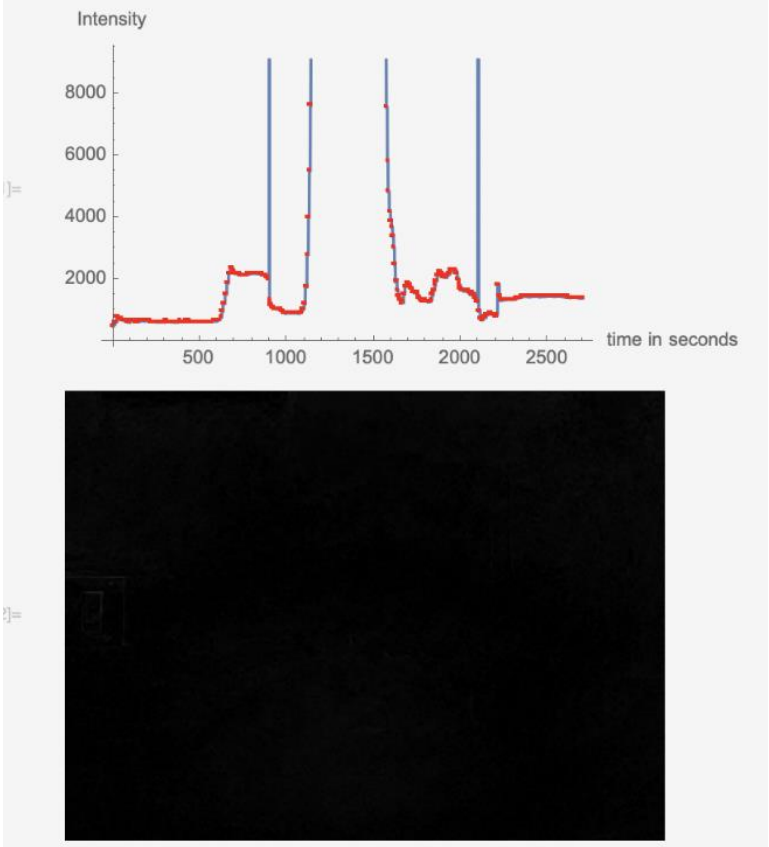
1. When there is a change in the background compared to the previous image, an alarm should be triggered.
2. When there is a change in the initial background, an alarm should be triggered.

Type 1:

- In this method, the Alarm is triggered only when there is a change in the background with respect to the previous image.
- This can be used in monitoring places like offices and workplaces with constant crowd and movement at most times. When there is any change in the frame it alerts us for a second and then it shuts down.
- In this method, the difference between the current image and the previous image is compared to monitor the place and trigger the alarm if there is any significant change identified between them.



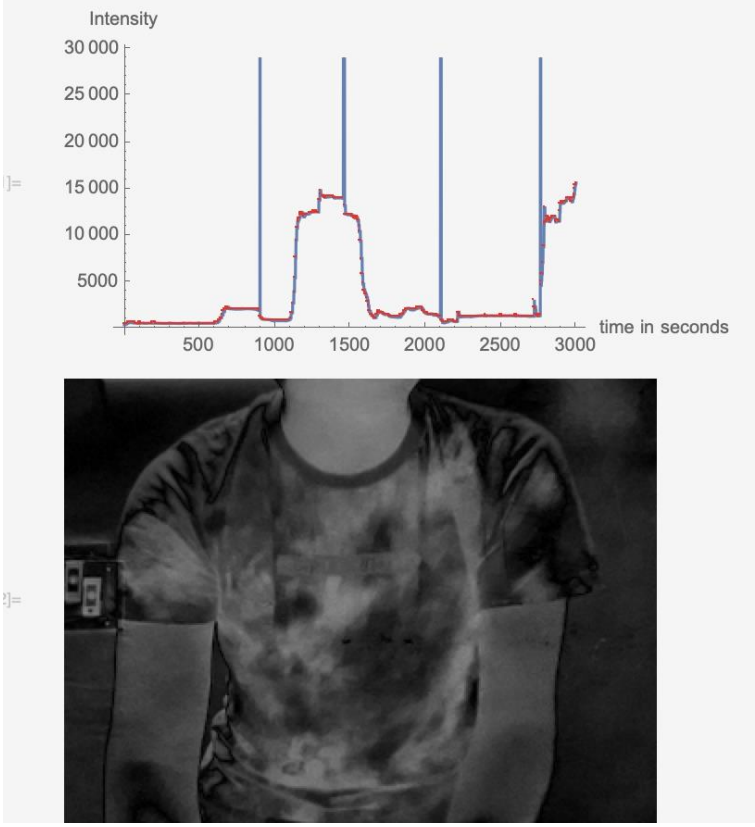
The above image shows a high intensity variation when we move our hand in front of the webcam (Type 1)



The above image shows a very low intensity variation when there is no intrusion detected (Type 1)

Type 2:

- This method is used to continuously monitor places and the objective is to alert us by triggering the alarm when there is a change in the background.
- In this method, the Alarm is triggered continuously when there is a change in the background with respect to the initial background.
- In this method, the difference between the current image and the first image is compared to monitor the place and trigger the alarm if necessary.



The above image shows that the intensity variation is high when there is intrusion in the background with respect to the first image (Type 2)

The Mathematica codes for the above two cases has been submitted in a separate Mathematica file attached with this report.

CONCLUSION:

Therefore we were able to achieve our objective of a web-based monitoring system using Mathematica. We see that Mathematica is a very useful program with a library of functions that help us build software for real world complex applications in an easier interface. We were also able to plot intensity as a dynamic function w.r.t to time for the intrusion software that helped us trigger an alarm when the intensity breached the threshold.