Visvesvaraya Technological University

Belagavi

A Mini Project Report On

Real Time Motion Detection using MATLAB

Submitted by

Jayanth S	**1NH19EC045**
Karthik S	**1NH19EC052**
Koduru Nani	**1NH19EC056**
Hussain Peera	**1NH19EC043**

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING
IN
ELECTRONICS & COMMUNICATION

2021-22

NEW HORIZON
COLLEGE OF ENGINEERING
New Horizon Knowledge Park, Ring Road, Marathalli
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

NEW HORIZON
COLLEGE OF ENGINEERING
New Horizon Knowledge Park, Ring Road, Marathalli
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

BENGALURU-560103

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

Certified that the Mini project entitled "**Real Time Motion Detection using MATLAB**" is carried out by Mr. Jayanth S bearing USN: 1NH19EC045, Mr. Karthik S bearing USN: 1NH19EC052, Mr. Koduru Nani bearing USN: 1NH19EC056, and Mr. Hussain Peera bearing USN: 1NH19EC043, bonafide students of NHCE, Bengaluru in partial fulfillment for the award of Bachelor of Engineering in Electronics and Communication of the Visvesvaraya Technological University, Belagavi during the year 2021-22. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The mini project report has been approved as it satisfies the academic requirements in respect of the mini project work prescribed for the said degree.

Signature of the Guide	Signature of the HoD
Ms. Bhawna Khokher	Dr. Aravinda K
Assistant Professor Department of ECE NHCE, Bengaluru	Professor & HoD Department of ECE NHCE, Bengaluru

External Viva

Name of the Examiners	Signature with Date
1. _____	1. _____
2. _____	2. _____

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement helped us succeed.

We thank **Dr. Mohan Manghnani**, Chairman of **New Horizon Educational Institution**, for providing the necessary infrastructure and creating a good environment.

We also record here the constant encouragement and facilities extended to us by **Dr.Manjunatha**, Principal, NHCE, and **Dr. Aravinda K**, Head of the Department of Electronics and Communication Engineering. We extend sincere gratitude to them.

We sincerely acknowledge the encouragement, timely help, and guidance to us by our beloved guide **Ms. Bhawna Khokher** to complete the project within the stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of the electronics and communication department for their co-operation extended to us, who helped us directly or indirectly in this successful completion of the mini-project.

Jayanth S **1NH19EC045**
Karthik S **1NH19EC052**
Koduru Nani **1NH19EC056**
Hussain Peera **1NH19EC043**

ABSTRACT

The number of crimes is increasing day by day, the security concerns are at a surge. Be it in the physical world or the cyber world, a person feels threatened due to the presence of unusual activities and hackers. To reduce this stress, if not completely remove it from the picture, we have come up with a project that detects motion in the physical world in real time using video feed from cameras without utilization of any sensers, which then can be used for versatile applications depending upon the necessity, as an exemplar application we have integrated a mechanism which will save only the video feed in which motion is detected and discard the remaining, thus saving memory space and easiness the analysis of the video feed when required. The project is achieved by a Motion detection algorithm. The codes are written down in the MATLAB environment. With the help of this project, we can achieve

- The detection of any sort of movement in a frame of observation in real time.
- Record only the moment of observation, to reduce memory usage.
- Store the moment of observed movement in a video file.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The security concerns in recent times have grown up immensely. It is important for us to safeguard our property from threats such as thefts, destruction of property, and many more. To match the rising levels of intimidators, the surveillance techniques also need to be improved. Thus, came up the methodology of motion detection.

The process of detecting a change in the position of an object with respect to its surroundings is called motion detection. When done by natural organisms, it's known as motion perception. There are quite a lot of methods to achieve motion detection which can be categorized into three categories: electronic (using infrared rays, lasers, etc), mechanical (using switch or triggers), devices (PIR sensor, cameras). Major applications of motion detection are to detect unauthorized entry, turning lights on/off when someone enters/leaves a space, etc. These applications can be achieved by applying simple algorithms along with hardware components in various environments such as PyCharm using python, MATLAB, and quite a few more. Currently, there is Motion detection software available in the market which requires dedicated cameras or even as simple as Webcams on our laptops for its use. Some of these are cloud-based and hence are developed based on algorithms that are updated regularly for better results. Lately advances In ML and AI have enabled Motion Detection to learn and develop its own algorithms which update itself based on training datasets.

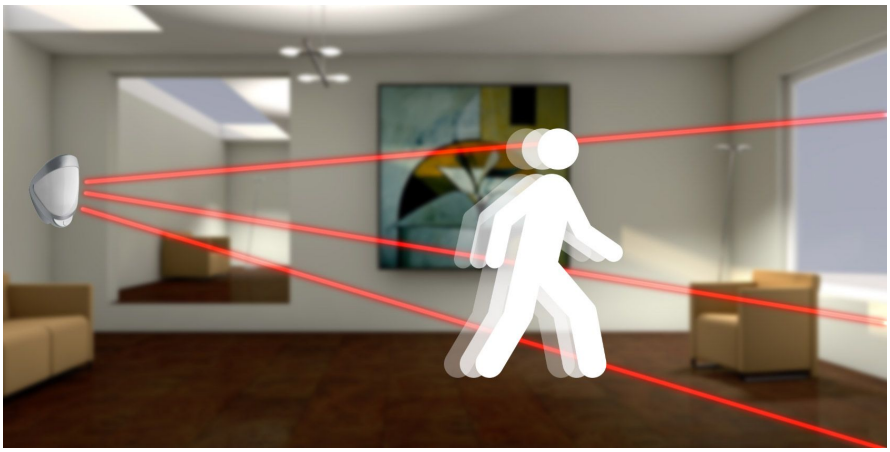

Fig 1.1 Cam Base Motion Detection Depiction

CHAPTER 2

LITERATURE SURVEY

Table No: 2 Literature Survey

No.	Article Title	Findings	Outcome
1	https://in.mathworks.com/help/vision/ref/vision.foregrounddetector-system-object.html	Working of Foreground analysis algorithm.	Key words to access foreground analysis algorithm and its understanding
2	https://in.mathworks.com/help/vision/ref/blobanalysis.html	Working of blob analysis.	Keywords to utilize blob analysis and depict the output of foreground analysis
3	https://in.mathworks.com/videos/video-processing-in-matlab-68745.html	Methods to control and access video and webcam from MATLAB	Commands to access webcam or any other cam-based peripherals using MATLAB
4	Detection and tracking system of moving objects based on MATLAB, Habib Mohammed Hussien	Basic working of motion detection	Video analysis for motion detection

Chapter 3

Existing system

Motion detection: Currently motion detection finds its uses in various fields, especially in IoT based projects such as home automation where motion sensors and few other sensors are used to detect motion in a room. If there is motion detected, the lights in that room are turned on. Once there is no motion detected at the same time, no human presence detected in that room, the lights are automatically turned off. Another application of motion detection in the IoT field is security.

Motion sensors are installed in buildings to detect unauthorized movement in restricted areas, particularly after hours. This can be connected to intruder alarms which can inform authorities about the movement.

Under motion detection, there are many implementations available where the motion is detected in real time but also the object that's under movement is analysed and recognised, if it is an animal, human, vehicle, etc.

Fig 3.1 Existing system depiction

Chapter 4

4.1 Proposed system

We intend to build a Realtime motion detection system. Motion detection is achieved using features of MATLAB such as foreground detection and background subtraction and blob analysis which combined is a part of computer vision. The motion which is detected is stored in a video file for the user.

Block Diagram of Proposed System:

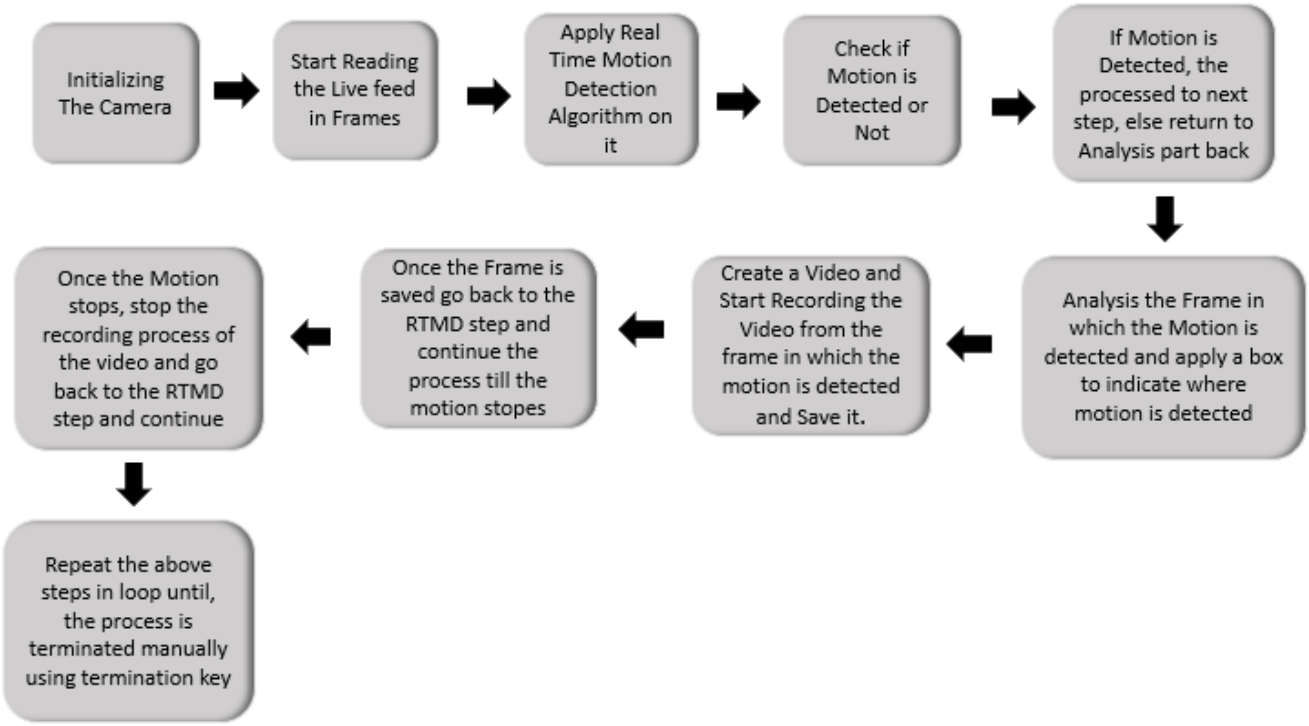

Fig 4.1 Block Diagram of Proposed System

In the MATLAB environment, motion detection can be achieved using a couple of features which have been mentioned above and are present in the MATLAB development kit as toolboxes. The features are:

4.1.1 Computer vision

Computer vision is a feature present in the MATLAB environment which uses images and videos to detect, classify and track objects or events in order to understand a real-world scene. In general, it is a field of Artificial Intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos, and visual inputs and take actions or come to conclusions based on the information gathered. In order for the computer to understand and recognize the distinct types of data being fed to it, it has to be calibrated and taught with vast quantities of data. The whole process can be linked to the real-world system of parents and teachers helping out kids to get to know the world they're living in by enlightening them with facts and information about the world. Computer vision now plays a huge role in various fields, from convenience stores to testing driverless cars, the healthcare sector, agriculture, banking, industrial, and many more. Top companies such as Amazon, Google, Microsoft, and Facebook have invested millions of dollars in the research and development of computer vision.

4.1.2 Blob Analysis

The method of analysing an image that has undergone binarization process i.e., setting a threshold value as a standard and converting a grey image into values of 0 and 1 to focus only on the object of inspection is called blob analysis.
It is the most basic method of for analysing the shape features of an object such as the presence, number, area, position, length, and direction of the blobs (lumps). The process of blob analysis consists of three steps – Extraction, refinement, and analysis.
In the first step of extraction, one of the image thresholding techniques is applied to obtain a region corresponding to the objects being inspected. The extracted region is often flawed by the noise of various kinds thus, in the next step refinement, the extracted region is enhanced using region transformation techniques.
In the final step analysis, the refined region is subjected to measurements and the final results are computed. If the region represents multiple objects, it's split into individual blobs and each of them is inspected separately. Blob analysis plays an important role in object detection and related projects.

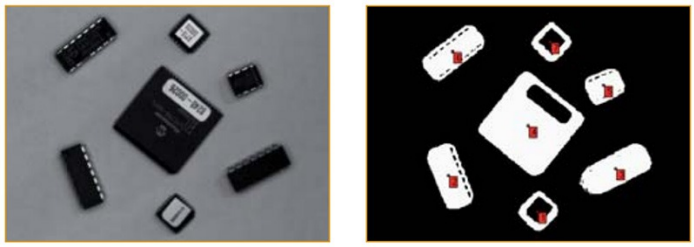

Fig 4.2 Blob Analysis Images

4.1.3 Foreground detection and Background subtraction

It plays a major role in the field of image processing. Here, the moving objects are referred to as foreground, and the parts that don't change are referred to as the background.

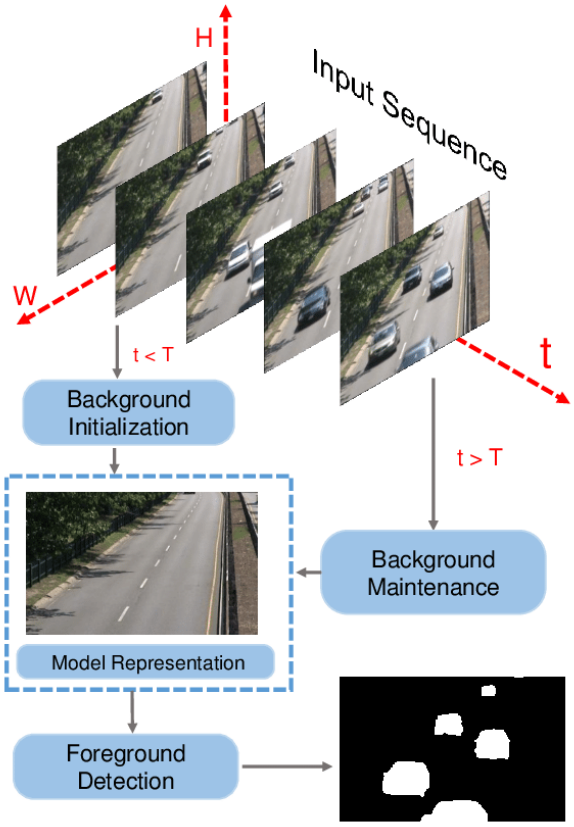

Fig 4.3 Foreground Analysis Depiction

The process starts with considering a series of images that are to be analysed. Then, the foreground detection compares these images to find out the change in positions of objects present in fame. We then get these objects removed to obtain the parts that stay still in all these compared images by applying the background subtraction process on these images. In simple terms, the foreground is extracted from the considered images to obtain the background. This analysis finds its applications in various projects like video surveillance, optical motion capture, human-computer interaction, traffic monitoring, real-time motion gesture recognition, and many

4.2 Proposed Methodology

The main aim of this project is to identify the Realtime motion in our surrounding. This is achieved by using video feed from cameras which record and saves it in an .avi video file format, only when the motion takes place.

We have used two main algorithms i.e. Foreground analysis and Blob analysis. These help to differentiate background and foreground, thus creating a picture to indicate the motion respectively. Previously mentioned algorithms have been used to implement Realtime Motion detection.

As soon as Realtime Motion detection algorithm is initialized, the MotionDetection function is called, which consists of 4 main functions to achieve Realtime Motion detection.

1. SetUpObjects is the first function to be called, it is used to initialize all the libraries and algorithms for object, so that they can be used to analyse the motion. The objects created in this function are:
   - webcam object to access the camera in order to collect the video in terms of frames which will be used for Realtime Motion detection.
   - ForegroundDetector object to differentiate between foreground and background, it is programmed to take initial 10 frames as background and build on top of it for analysis.
   - BlobAnalysis object to create an image indicating where motion is taking place, this object creates an image in such a way that the region where the motion is taking place is taken in as white and static background as black
   - VideoWrite object is created to save the video in which motion is detected.

2. Initialize is the second function which is called, this function is used to initialize all the different variable, which maintain the record of various parameters such as if motion is taking place or not, if video under record state etc.

3. StartMotionDetection function is the third and most important function to be called, this analyses the video frames which is being collected by the camera. As soon as this function is called motion detection process is initiated through the following steps

   - First using the cam object, a frame (photo) is taken using the camera
   - The frame is then analysed for foreground and background analysis and return the details where the motion is taking place
   - These details can be feed to blob analysis object to create a black and white image to indicate where the motion is taking place to the user.
   - A box is inserted around the region where the motion is taking place
   - If motion is detected then the video write object is called and video recording is initialized as long as motion is taking place and terminated once it stops.

- All the above-mentioned steps are enclosed in an infinite while loop which can only be terminated using special key termination method (in our case it is “q”).

4. The final CloseAll function is called out when the loop is terminated using termination key, this function is used to release all the objects that are created for freeing up of memory.

Flowchart of Program:

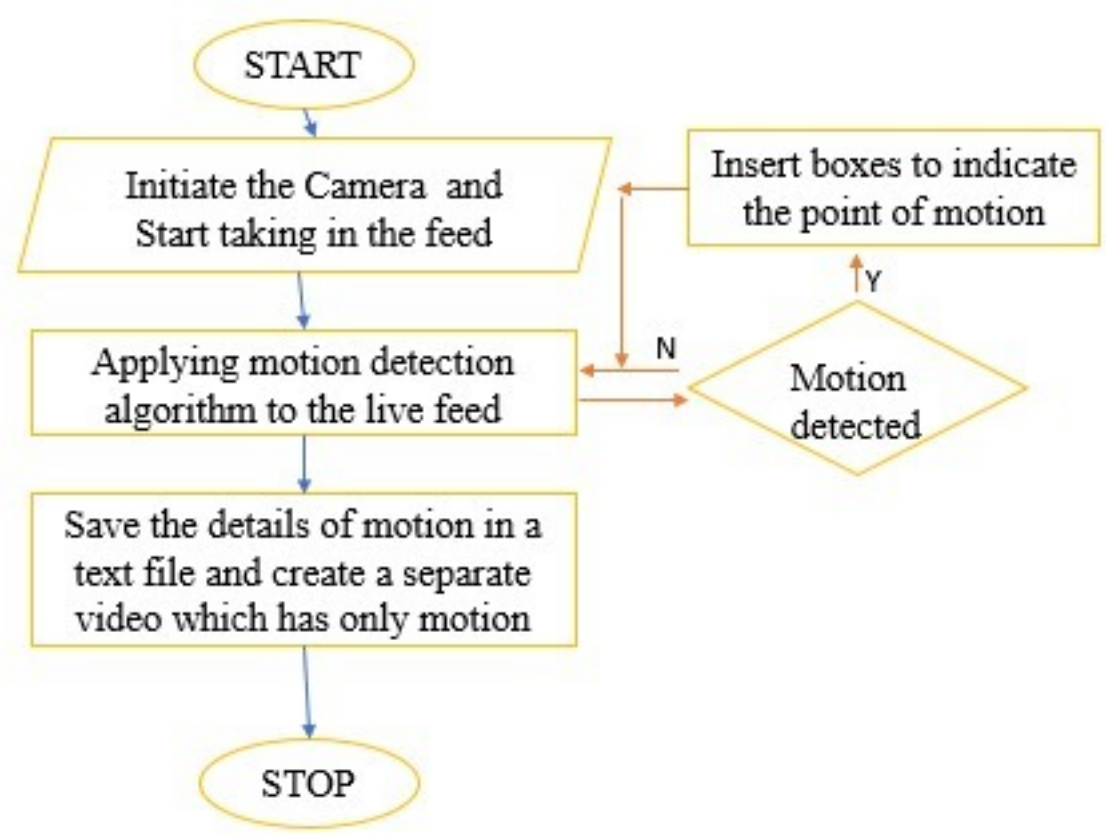

Fig 4.4 Flowchart of Program

Chapter 5

PROJECT DESCRIPTION

5.1 Software Description

The software used to achieve this project is MATLAB. MATLAB, an abbreviation for Matrix Laboratory is a programming and numeric computing platform that is used by millions of engineers and scientists to analyse data, develop algorithms and create models. It finds its use in design, test, and implementation of control systems, deep learning, image processing and computer vision, machine learning, robotics, signal processing, wireless communication among its complete range of usability. Though primarily intended for numeric computing, an optional toolbox allows access to symbolic computing abilities. In addition to this, another package called Simulink can be used to add graphical multi-domain simulation and model-based design for dynamic and embedded systems.

The MATLAB system consists of five main parts:

MATLAB language:

It is a high-level matrix/array language with functions, control flow statements, input/output statements, data structures, and object-oriented programming features. It allows both "programming in the small" to create quick and throw-away programs and "programming in the large" to create large and complex application programs.

MATLAB working environment:

It is the set to tools and facilities that the user works with. It includes facilities for managing the variables in the workspace and importing and exporting data. It includes tools for managing, developing, debugging, and profiling M-files.

Handle graphics:

It is the MATLAB graphics system. It includes high-level commands for 2D and 3D data visualization, animation, image processing, and presentation graphics. It includes low-level commands too that allow the user to fully customize the appearance of the graphics and to build a complete Graphical User Interface (GUI).

MATLAB mathematical function library:

It is a vast collection of computational algorithms that ranges from basic functions such as sum, sine, cosine, and complex arithmetical commands to more sophisticated functions such as matrix inverse, eigen values of a matrix, Bessel functions, and fast Fourier.

MATLAB application program interface (API):

It is a library that allows the user to write codes in C and Fortran that will interact with MATLAB. It includes facilities for calling routines from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MATLAB files.
Uses of MATLAB include:

- Math and computation.
- Modelling, simulation, and prototyping.
- Algorithm development.
- Scientific and engineering graphics.
- Data analysis, exploration, and visualization.
- Application development including Graphical User Interface (GUI) building.

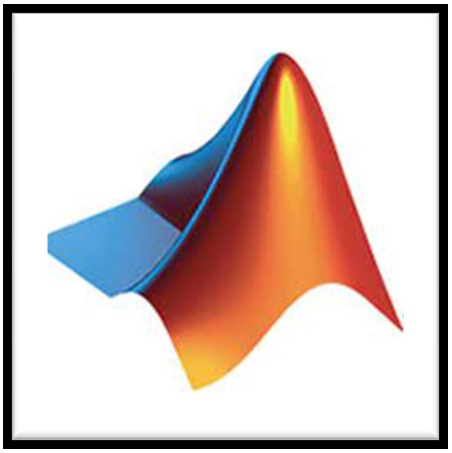

Fig 5.1 MATLAB Icon

Chapter 6

Advantages and Application

Currently the project is king of skeleton, which can be built upon for various application such as wild life protect, home security, video feed analysis, automated attendance and monitoring system, authenticated access only systems, smart stores (Amazon go) etc, through appropriate changes and add on features.

Advantage of using such a system, reduction in E-Waste, saving of space, versatile application, cost friendly, simple maintenance, faster deployment, Reduced instrumental requirement, etc.

Few applications of the project are:

- Used by forest rangers to detect the presence of poachers and other unusual activities.
- Used by normal civilians outside their homes and official spaces to detect any unusual activities taking place outside.
- Can be used in military fields where one army can detect the motion taking place in the enemy territory and the details of this can be conveyed to the other groups situated at various other locations at the camp without arousing any suspicions.
- Used in the field of cyber security for video analysis. Can be used to convey information about counter-attacks

Chapter 7

Results and Discussion

Results

The objective of the project is successfully achieved, that is a connection to a camera was established and live video is being taken in as feed which is being utilized for motion detection processing in real time using motion detection algorithms, motion is been detected successfully and video is recorded only if any motion is detected.

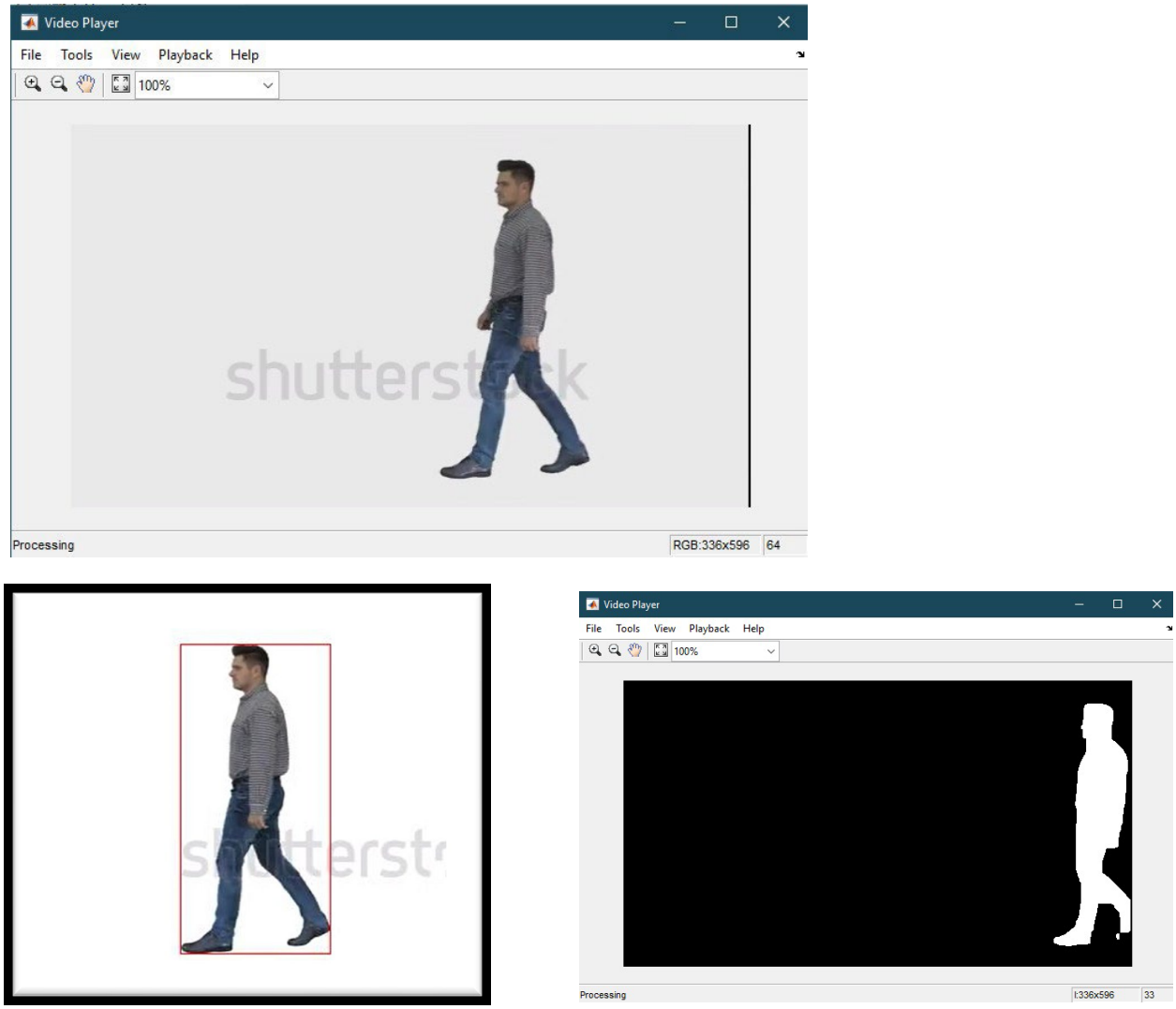

Fig 7.1 Images of Output results of Blob Analysis and Motion Detection

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

The project is developed as a core code which can be applied in various fields with few modifications made according to the needs. The main aim of this project was to design a system that can monitor movements that the naked human eyes can't detect in real time. Therefore, the project acts as a solution to physical threats and can be developed further to work in a much more efficient manner and also with upgrades that will increase the boundaries the project can be used in.

REFERENCE

- https://cimss.ssec.wisc.edu/wxwise/class/aos340/spr00/whatismatlab.htm
- https://www.edureka.co/blog/steganography-tutorial#SteganographicTechniques
- https://en.wikipedia.org/wiki/Foreground_detection
- https://en.wikipedia.org/wiki/Computer_vision
- http://what-when-how.com/introduction-to-video-and-image-processing/blob-analysis-introduction-to-video-and-image-processing-part-1/

APPENDIX

Code Review:

```
% Author : Jayanth.S, Hussain Peera, Karthik.S, and Koduru Nani
% Date : 28/05/2022
% Title : Real Time Motion Detion Using Matlab

MotionDetection()


%% Initialization
function Ini = Initializ()
   Ini.NumOfFramesObjDet = 0;
   Ini.NumOfFramesObjNotDet = 0;
   Ini.MotionState = 0; %0 means motion not taking place, 1 means motion taking place
   Ini.motionDet = false; %false means motion has stoped, thus stop writing video
            %True means motion is taking place, thus keep writing video
   Ini.StopWriting = false; %false stop writing video %True keep writing video
end
%% Functions
```

```
function Obj = SetUpObjects()
   % Setup of cam
   Obj.mycam = webcam;
   Obj.mycam.Brightness = -40;
   % creating foreground anlaysis object through which foreground analysis will be done
   Obj.detector = vision.ForegroundDetector("NumTrainingFrames", 10,...
                     "InitialVariance", 60*60,...
                     "MinimumBackgroundRatio", 0.7,...
                     "NumGaussians", 3);
   %creating Blob analysis object to do blob analysis through this object
   Obj.blobAnalyser = vision.BlobAnalysis('MinimumBlobArea', 500);
   %creating an object to terminat the infinite loop
   Obj.stop=figure('position',[0 0 eps eps],'menubar','none');
   % video writing
   Obj.mywriter = VideoWriter("mymovie.avi");
   open(Obj.mywriter);
   % Set up video player
   Obj.player = vision.VideoPlayer("Position", [180, 100, 700, 400]);
   Obj.player2 = vision.VideoPlayer("Position", [300, 200, 700, 400]);
end

%%% Creating Foreground mask
function mask = foregroundMask(img,Obj)
      mask.maskImg = Obj.detector.step(img);
      mask.maskImg = imopen(mask.maskImg, strel("rectangle", [3,3]));
      mask.maskImg = imclose(mask.maskImg, strel("rectangle", [15, 15]));
      mask.maskImg = imfill(mask.maskImg, "holes");
end

%%% video player
function PlayVideo(img,mask,Obj)
      Obj.player.step(img);
      Obj.player2.step(mask);
end

%%% Starting the Motion Detion in the frame
function StartMotionDetection(Obj,Ini)
   % Creating Infinite Loop to analysis
   while 1

      % Termination of Infinite Loop
      if strcmp(get(Obj.stop,'currentcharacter'),'q')
        close(Obj.stop)
        break
      end
      % force the event queue to flush
```

```
        figure(Obj.stop)
        drawnow

        img = snapshot(Obj.mycam); % Extraction Images from cam in frames
        mask = foregroundMask(img,Obj); % Foreground
        [~,~,bbox] = Obj.blobAnalyser.step(mask.maskImg); % Find bounding box
        ObjDet = size(bbox,1); % To indicate if object is detected or not

        %checking if there is any motion
        if(ObjDet>1)
          Ini.NumOfFramesObjDet = Ini.NumOfFramesObjDet+1; %used to count the no
of frames the motion is detected to consider it as moving object
          if(Ini.NumOfFramesObjDet >= 5) %True of object is detected in more than 5
frames
            if((ObjDet>1)&&(Ini.MotionState==0))
              fprintf("Motion Detected\n");
              Ini.StopWriting = false;
              Ini.motionDet = true; %used to write video when it is true
              Ini.NumOfFramesObjDet = 0; %setting no of frames object detected to 0
              Ini.MotionState=1; %to indicate that motion has begin (if 0 motion was not
there till then/ if 1 motion was there till then)
            end
          end
        end
        %checking if there is no motion
        if(ObjDet==0 && ~Ini.StopWriting)
          %used to count the no of frames the motion is not detected to consider that there
is no motion
          Ini.NumOfFramesObjNotDet = Ini.NumOfFramesObjNotDet+1;
          if(Ini.NumOfFramesObjNotDet >= 15)
            if((ObjDet == 0)&&(Ini.MotionState==1))
              fprintf("Motion Stopped\n");
              %Ini.StopWriting = false;
              Ini.motionDet = false; %reads the ending time of motion
              Ini.NumOfFramesObjNotDet = 0;
              Ini.MotionState=0; %to indicate that motion has begin (if 0 motion was not
there till then/ if 1 motion was there till then)
             end
          end
        end
        ObjFrame = insertShape(img,"rectangle",bbox,"color","r");
        PlayVideo(ObjFrame,mask.maskImg,Obj); % Play Video
        WriteVideo(Obj,Ini,img,bbox);
    end
end
```

```
%% Writing Video
function WriteVideo(Obj,Ini,img,bbox)
   %writing the video where the motion is detected
   if(Ini.motionDet)
      %insert bounding box in frame
      ObjFrame = insertShape(img,"rectangle",bbox,"color","r");
      %writing video
      writeVideo(Obj.mywriter,im2double(ObjFrame));
   end
   %inserting empty frames to indecate the ending of the video
   if(~Ini.motionDet && Ini.StopWriting)
      for i=1:50
         writeVideo(Obj.mywriter, zeros(480,640));
         Ini.StopWriting = true;
      end
   end
end

%% Clean up
function CloseAll(Obj)
   delete(Obj.stop);
   delete(Obj.mycam);
   close(Obj.mywriter);
   release(Obj.detector);
   release(Obj.blobAnalyser);
   release(Obj.player);
   release(Obj.player2)
end

%% Real Time Motion Detion Using Matlab
function MotionDetection()
   Obj = SetUpObjects();
   Ini = Initializ();
   StartMotionDetection(Obj,Ini);
   CloseAll(Obj);
end
```