Burglar Detection System

Submitted in partial fulfillment of the requirements of the degree of

BACHELOR OF COMPUTER ENGINEERING

by

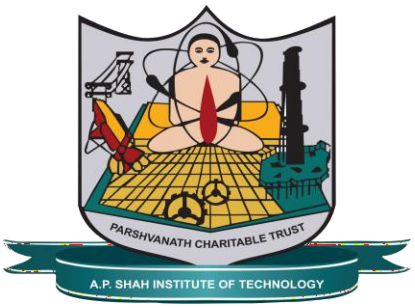
Tejas Sheth(19102026)

Poojan Shah (20202001)

Rakshit Shah (19102008)

Guide

Prof. Merlin Priya



Department of Computer Engineering

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

(2022-2023)

Abstract

The need to have good security, either in the streets, at home or at workplaces, cannot be overemphasized. Due to its significance, security experts continue to improve the mechanism and the tools used to manage security issues. The computing and information technology revolution has significantly affected how people deal with security. Machine vision application has been developed for security purposes, especially, by improving surveillance systems. The machine vision system can detect the on-screen movement efficiently and alert the user promptly. This task can be used to improve security surveillance.

As a result, a low-cost surveillance system can be made to detect any malicious activities happening in the vicinity of the camera. This report seeks to analyze how Machine Vision systems can be of help in surveillance processes including tracking burglars, strange behaviour, and crime in general.

CONTENTS

1. Introduction

The term "security" refers to people's perceptions of environmental protection and means "without fear of harm". As the global crime rate increases, so does the need for security services. The demand for investigation and security services is projected to increase from $288 billion in 2020 to $417 billion by 2025.

While conventional security services such as manual surveillance are effective, they are limited to the capabilities of the person performing them. For instance, security personnel can only watch surveillance footage for a limited time and with a limited level of accuracy.

Computer Vision is a field of Artificial Intelligence (A.I.) and can help the security sector overcome these challenges by providing higher accuracy and efficiency. The major advantages of using a surveillance system can be detecting thieves and robbers, recording evidence, monitoring suspicious activities, keeping a tab on activities, Maintain adequate records. A surveillance system should be able to detect a motion and alert the surroundings by playing an alert sound.

OpenCV plays a major role to perform and implement different modules and activities with the help of a camera device and an audio input file that acts as an alert system. This research compiles technological breakthroughs in surveillance systems, applications, and key components into a single document, providing a literature assessment on the issue of security.

2. Proposed System

We propose to help out general people and to safeguard their houses when they are not around. The plan is to build a system that is capable of detecting motion in and around the vicinity of the camera's view range, and trigger an alarm to alert the property owners, and/or the neighbours.

The plan is to achieve it using Python programming language and one of python's inbuilt libraries - OpenCV. OpenCV plays a major role in performing and implementing different modules and activities with the help of a camera device and an audio input file that acts as an alert system. We plan to utilize the predefined functions of VideoCapture, absdiff, cvtColor, GaussianBlur, threshold, dilate, findContours, boundingRect, and rectangle to achieve our objective.

A Surveillance system should be able to detect a motion and alert the surroundings by playing an alert sound. Future Scope includes implementing a machine learning model that learns to distinguish between a random walker, a property owner and someone who tries to break in or exhibits unusually strange behaviour patterns. This would make the system intelligent enough to overtime replace the need for any security guards.

3. Experimental Setup

3.1. Hardware Requirements

CPU: 2.8 GHz or faster 64-bit processor; Quad-core or better recommended.
RAM: Minimum of 4GB of ram.
Storage: 4GB of free hard disk space.
Camera: Either an in-built camera or a webcam is necessary.
Speaker: Audio output device necessary.

3.2. Software Requirements

Python: General Programming Language used to code the machine vision program, includes several libraries.
Python libraries: Python CV2, winsound
VS Code: IDE to execute the machine vision program. Plenty of extensions, open-source, and offers cross-platform support.
Git & GitHub: Version Control System used for collaboration.
Google Workspace: Project Management Tool to improve team coordination and file sharing.

4. System Implementation

To initiate video capture:

We used the CV2 method `VideoCapture` to start the system camera or any video recording device. It requires an argument to select one of the webcam if multiple cameras are connected.

```
camera = cv2.VideoCapture(0)
```

To detect motion using the difference of 2 consecutive frames:

Read the camera feed frame by frame, here we utilize 2 frames to evaluate the difference and thus detect movement if it happens. The `read()` method returns 2 values, retrieve and frame of the camera. We make use of `absdiff()` method of `the CV2` package, which takes 2 frames as parameters.

```
ret, frame1 = camera.read()
ret, frame2 = camera.read(
frameDifference = cv2.absdiff(frame1,
frame2)
```

Gray Scaling:

The difference is colourful and bright, and such images are not recommended as they produce minor errors. Thus we convert it to grayscale using `cvtColor()` method, which takes the image

difference and `cv2.COLOR_BGR2GRAY` as parameters and return a grayscale image.

```
grayImg=cv2.cvtColor(frameDifference,
cv2.COLOR_BGR2GRAY)
```

Blurring:

We use `GaussianBlur() method` to blur the grey image, passing the grayscale image, a kernel size and sigmaX.

```
blurImg = cv2.GaussianBlur(grayImg, (5,5),
0)
```

Thresholding:

To eliminate the unwanted noise, we use the `threshold()` method, which takes the blurred image, threshold values, maximum threshold value, and threshold type `THRESH_BINARY.` It returns 2 values, but we use only the threshold image.

```
_,thresh
=cv2.threshold(blurImg,20,255,cv2.THRESH_B
INARY)
```

Dilation:

It is the opposite of thresholding. It is used to enhance the actual image after thresholding. We use `dilate()` method, passing threshold image, some kernel or None, and the number of times to dilate. It returns the dilated image.

```
dilated =
cv2.dilate(thresh,None,iterations=3)
```

Contour Detection:

We need contours to distinguish between objects in motion and those that are static. For this purpose, we use `findContours()` method passing the dilated image, mode and method as parameters.

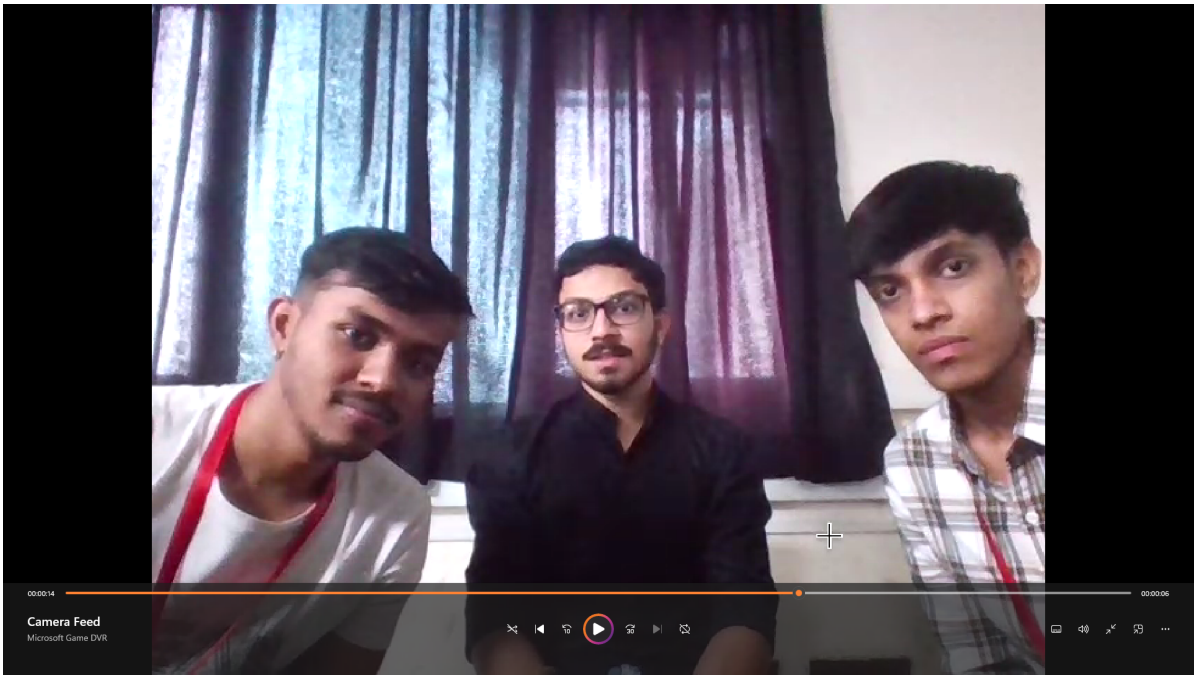
```
contours,_ =
cv2.findContours(dilated,cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
```

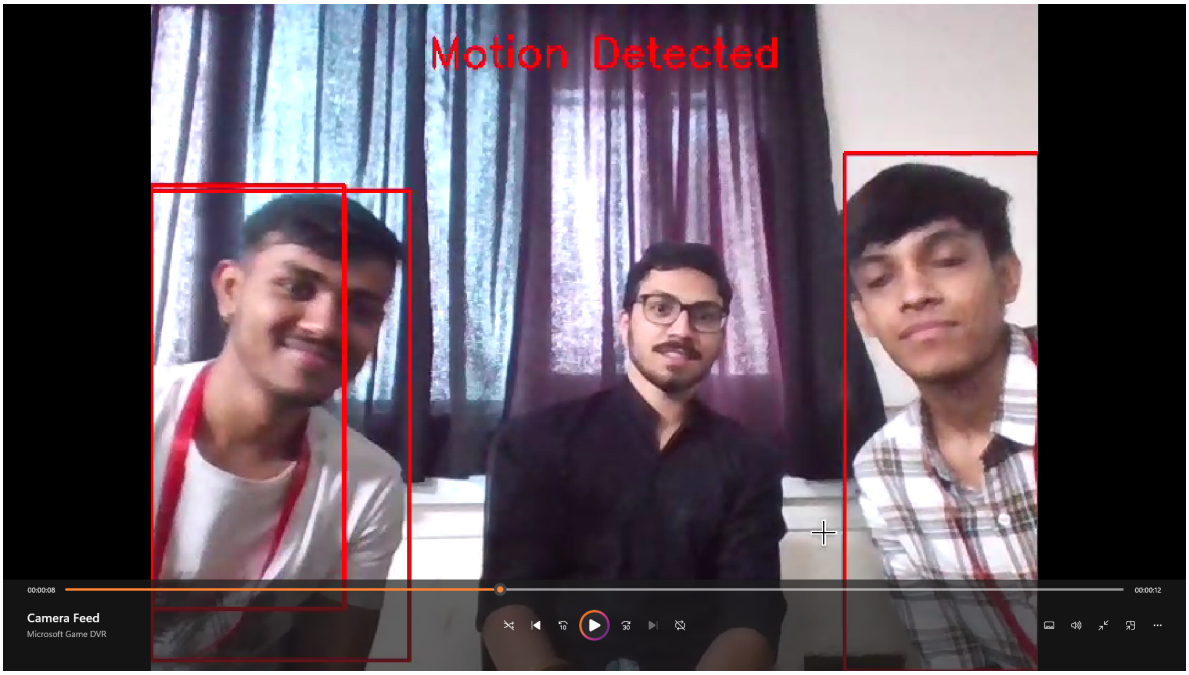
Bounding Rectangle:

Eradicating the smaller contours using `contourArea()`, we use the method to obtain the X and Y coordinate and the height and width of the rectangle to the plot, using the `rectangle()` method.

```
xAxis,yAxis,width,height =
cv2.boundingRect(contourSize)
cv2.rectangle(frame1,(xAxis,yAxis),(xAxis+
width,yAxis+height),(0,0,255),2)
```

5. Results


00:00:14
00:00:06
Camera Feed
Microsoft Game DVR


Motion Detected
00:00:08
00:00:12
Camera Feed
Microsoft Game DVR

Acknowledgement

We have great pleasure in presenting the mini project report on **“Burglar Detection System”.** We take this opportunity to express our sincere thanks towards our guide **Prof. Merlin Priya,** Department of Computer Engineering, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of the project.

We thank **Prof. Sachin Malave, Head of Department**, Computer Engineering, APSIT for his encouragement during the progress meeting and providing guidelines to write this report.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Student Name1: Tejas Sheth
Student ID1: 19102026

Student Name2: Poojan Shah
Student ID2: 20202001

Student Name3: Rakshit Shah
Student ID3: 19102008