Winter Semester 2022-23

WOMEN SAFETY NIGHT PATROLLING CAR REPORT

Submitted by

Jesmine Akhter, 20BCE1945

Bacham Sai Venkat Teja, 20BCE1551

Tathagata Biswas, 20BCE1844

CSE 3009, Internet of Things

Project Guide

Dr. Manjula V

Associate Professor

School of Information Technology and Engineering

B.Tech in Computer Science and Technology

IN

SCOPE

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Winter Semester 2022-23

Review III

Report Submission

Report submission Date : 12.04.2023

ABSTRACT

There are certain places where it is difficult to maintain surveillance at all times or at all, and also prompt the necessary action at once, if required; which is not possible by just using CCTV cameras that record the occurrences that occur within the designated area of concern. In order to address this issue, it is necessary to have a robot that can detect and deter potential intruders in certain areas as well as identify suspicious activity in other areas where human surveillance is not possible at all times.The proposed robot will use an IR-based method and sound sensing in accordance with the framework to monitor designated zones. In order to prevent a collision, the motion of the robot is also consequently regulated by deterrent recognising sensors. It uses a camera installed on the top of the robot to take photographs, record videos, and communicate the images to the customer as it scans each area for any interruptions. Additionally, it can give the client access to watch the live video feed.Thus, by combining all the above stated functionalities, it makes a useful entity for proper surveillance.

TABLE OF CONTENTS

Chapter Number	Topic	Page Number
1	Introduction	5
1.1	Problem Statement	5
1.2	Objectives	5
2	Literature Review	6
3	Proposed Methodology	12
4	Implementation	13
5	Testing and Performance Evaluation	15
5.1	Testing Object Avoidance function	15
5.2	Surveillance function	16
5.3	Image Detection Model	17
6	Conclusion and Future Enhancements	20
7	Appendix	21
7.1	Code for ultrasonic sensor and motor for aiding in object avoidance	21
7.2	Code for ESP32 camera to render live video feed to the user	23
7.3	Machine learning model for person detection	26
8	References	30

TABLE OF FIGURES

SL.NO	Description	Page Number
1	Proposed Methodology	12
2	Surveillance Car Top view	14
3	Surveillance Car Front view	14
4	Object Avoidance algorithm output on Serial monitor	15
5	Video Live Stream on server side which is being captured by esp32 cam	17
6	Image Detection Model recognising people	18
7	Image Detection Model printing the count of humans in the picture taken by esp32 cam	18
8	Model architecture	19
9	Model output	20

1. INTRODUCTION

1.1 PROBLEM STATEMENT

Violence against women – such as intimate partner violence, rape and emotional abuse – poses a significant public health problem in every country. It seems as if, with the passage of time- while the advancement in technology is reaching new heights, so are the ways in which women are being harassed and assaulted. In a world where one in the three women now experience physical, emotional or sexual violence at least once in her lifetime, raising awareness about violence against women on a global scale and finding effective measures to prevent the same has become very important and is now one of the primary concerns the government of every country. The technological advancements are now being put to use to formulate effective surveillance and alert systems -CCTV cameras and women's safety watches being prime examples- to protect women and help them feel safe. As an aid to the same cause, we are proposing a smart night patrolling robot- with the intent of enabling surveillance of dedicated areas which will be capable of sending real-time alerts when it detects any unusual activity in its surroundings.

1.2. OBJECTIVES

● Automation of surveillance, where employing manpower may not be possible

There are a few spots where people cannot always be engaged with looking at the camera or where sometimes keeping a keen watch may not be possible due to various reasons. In order to deal with such circumstances, designing a robot which both identifies trespassers in the territory like workplaces, home, building and so forth and report any suspicious activity can be very effective. Existing systems use CCTV cameras to record incidents, but this is not a preventative strategy for women's protection. Our suggested robot will patrol in its assigned area and check for anonymous behaviors and alarm when it detects the same. As a result, it reduces the probability of violent crime by recognizing, defending and looking up resources to help people, especially women, in hazardous situations.

● No predefined path- to aid in better surveillance

Our night patrolling robot vehicle moves in a random path while watching. The proposed system utilizes the IR based way following the framework for watching allocated zones. It uses sound sensors to detect the presence of probable sound sources which, in ideal circumstances, will not be present in an otherwise quiet environment, and the robot moves in the direction of the sound upon perceiving the same. The motion of a robot is additionally controlled consequently through deterrent recognizing sensors to stay away from the crash. It screens every zone to recognize any interruption or unusual change in frame utilizing a camera which is mounted on the top of the robot to catch the pictures, record and send them to the client. It can likewise impart the ongoing video signs to the client. The principal goal of this undertaking is to recognize the dubious exercises in the regions where human presence can't be seen.

2. LITERATURE REVIEW

[1]

Iot Based Night Patrolling Robot for Women Safety

Volume 10 Issue 2, 2021

Algorithm: LPC2148, GPS and GSM - based strategy.Remote Health Monitoring System.

Objectives: The project uses Arduino and a GSM module with GPS to send an emergency message with location and generate an alarm in order to mitigate and ensure the safety of women. The device can send real-time messages with precise position information, including longitude and latitude, using GSM and GPS. These values are used by robots and attempt to reach the area it got with an alarm while reaching the area.

Limitations: There are distinctive sorts of security partner gadgets, which are costly and not at all highlights are achievable. Those gadgets are too complicated to use. Therefore, the oddity of this work is attempting to make a cheap and client inviting security right hand and badgering avoidance gadget for everybody.

[2]

Night Vision Security Patrolling Robot Using Raspberry Pi

Volume-3, Issue-8, Aug-2020

Algorithm: AdaBoost based algorithm, Background Subtraction algorithm, ARM and Quick Motion Stimulus algorithm.

Objectives: Raspberry pi is installed with the night vision camera that helps the system find the intruders and go for the automation. The robot covers a specific area and checks for any intruders when an intruder is detected, the owner is alerted by the buzzer sound.

Limitations: Installing a wired system can take a considerable amount of time and money. The establishments are a permanent part of that. When the owner moves the machine must stay in place.

[3]

Women Safety Night Patrolling Robot

Volume 10 Issue VI June 2022

Algorithm: AdaBoost based algorithm.

Objectives: Sound alert system. The system uses an Infrared (IR) based path following system for patrolling assigned areas providing vigilance with two HD cameras.

Limitations: Night Vision Mode is not enabled, motion sensors would be tough, and image detection would be tough if captured blurry. A significant communication delay would exist.

[4]

IOT and AI Based Women's Safety Night Patrolling Robot

13 July, 2022

Algorithm: GPS and GSM - based strategy, Node MCU.

Objectives: Ultrasonic sensors, Arduino, PIR sensors and a night vision camera which is mounted on the robotic vehicle.

Limitations: Numerous sensors can be used like gas sensors, temperature sensors, pressure sensors, proximity sensors and many more for extra functions. Metal detectors can be used to detect the bombs. So, it can be used as a bomb defuser and bomb disposal team. It can be used to detect intruders at borders, hence can be used in border security. We can also add audio communication features so that we can listen to the strategy of the intruders.

[5]

Night Vision Patrolling Robot for Security Patrolling Using Raspberry Pi

Vol. 11, Issue 5, (Series-VI) May 2021

Algorithm: Adaboost, Bagging and Enhanced Reweight algorithms, outlier identification or (k-NN), Genetic Algorithm.

Objectives: Any type of sound resulting in Sending notification to the user through the Blynk. Robot is consisting of a night vision camera from which the live video can be seen through the smart phone where it will capture the image and directly send it to the user.

Limitations: None mentioned in the paper.

[6]

Decentralised control of a heterogeneous human-robot team for exploration and patrolling

August 2021

Algorithm: Connectivity - maintenance algorithm, extends to integrate exploration tasks.

Objectives: Confusion matrix to visualize the two feedback modes - audio and haptic.

Limitations: The temporary loss of team connectivity is not considered by the algorithm.

[7]

A Survey of Multi-robot Regular and Adversarial Patrolling

July 2019

Algorithm: Regular patrolling and Adversarial Patrolling.

Objectives: Normalized value to determine the individual's model's contribution to the system (MRRP) Detection Probability of detecting an intruder in the environment.

Limitations: A significant communication delay exists, which may not only deteriorate the performance of a multi-robot system, but also impact its stability when the delay is large.

[8]

Intelligent Surrounding Recognition for Robot Direction Control

November 2020

Algorithm: Local Planning using LSR, double scanning and image auxiliary method to improve accuracy.

Objectives: Best Passing Way (BPW) width is used for the traveling of the robot.

Limitations: Stable connection of ZigBee is needed to calculate the location of the robot.

[9]

To Detect Abnormal Event ATM System using Image Processing and IOT

September 2022

Algorithm: Convolutional Neural Networks (CNN) - regular and Bayesian.

Objectives: Model learning curve- depicting validation loss and accuracy along with training loss and accuracy.

Limitations: Overfitting is observed.

[10]

Intrusion Detection through Image Processing and getting Notified Via SMS and Live Streaming

December 2018

Algorithm: KL Transform Algorithm. Pre-Processing:Median Filter.

Objectives: KL Transform Algorithm is used for compression of the image produced from the infrared camera. Pre-Processing:Median Filter will be applied to remove noise from the image camera.
Limitations: Proper detection cannot be done if images captured are blurry. Continuous energy supply will be required. Bandwidth consumed will be large, not feasible for IOT systems.

[11]
Night Patrol Robot for Detecting and Tracking of Human Motions using Proximity Sensor
November 2022
Algorithm: Image motion analysis, Object tracking, robot vision.
Objectives: Proximity Sensor measures the number of sensations that it has recorded. Accordingly human interactions will be estimated.
Limitations: High Standard Maintenance Required.

[12]
Web-Based Wireless Controlled Robot for Night Vision Surveillance Using Shell Script with Raspberry Pi
June 2018
Algorithm: GINI record, Boosting method.
Objectives: Procedures which are utilized to improve the arrangement execution. It is a strategy which can be utilized to improve the exhibition of any feeble classifier. The broadly utilized boosting calculation is AdaBoost, which iteratively assembles a gathering of models.
Limitations: Wireless connectivity beyond range becomes problematic. And once the robot crosses the field of influence, it will become difficult to track it.

[13]
A Robot - Centric Group Detection and Tracking System
September 2022
Algorithm: REGROUP (Robot-Centric Group Detection and Tracking), Cluster point classification.

Objectives: A new method that enables robots to detect and track groups of people from an ego-centric perspective using a crowd-aware, tracking-by-detection approach.
Limitations: When people are accumulated in large numbers, because of the high number of cluster points.

[14]
Surveillance Robot Using Arduino Microcontroller, Android APIs and the Internet
May 2021
Algorithm: API end requests, Python flask frameworks.
Objectives: Sound alert system. The system uses an Infrared (IR) based path following system for patrolling assigned areas providing vigilance with two HD cameras.
Limitations: This can have some trouble when some other picture comes into the image. There is a planar reflection model that is useful to get the force circulation of various pixels with an infrared camera. It just assumes and decides so that the further cycle gets started.

[15]
Moving object tracking based on mobile robot vision
August 2019
Algorithm: The paper describes a robotic application that tracks a moving object by utilizing a mobile robot with the capacity of avoiding obstacles. Real-time tracking algorithm is based on mobile robot vision using adaptive color matching and Kalman filter.
Objectives: Adaptive color matching is used to obtain motion vectors of moving objects in the robot coordinate system. It can adjust color matching threshold adaptively to reduce the influence of lighting variations in the scene.
Limitations: Tracking algorithms using vision should be able to adapt to the complex variations.

3. PROPOSED METHODOLOGY

We propose an automated patrolling security to secure any premises. The car is equipped with a camera which supports night vision and ultrasonic sensors for object avoidance - which allows it to move in a free path. This will help in enhancing the environment sensing capacity of the robot. It monitors its environment using ESP32 camera module to detect any intrusion. Here, we are making use of Local Area Network (LAN) to connect the camera with the user's device so that the user can get live video of the area, via a web interface which even allows the user to change the resolution of the incoming feed, and change the orientation of the video. Further, we are putting forth an image detection model which can detect the number of persons currently in front of the camera,so that the user can get an exact idea about the number of intruders if any. So we're putting forward a fully autonomous security car that patrols large areas alone to secure the facility.

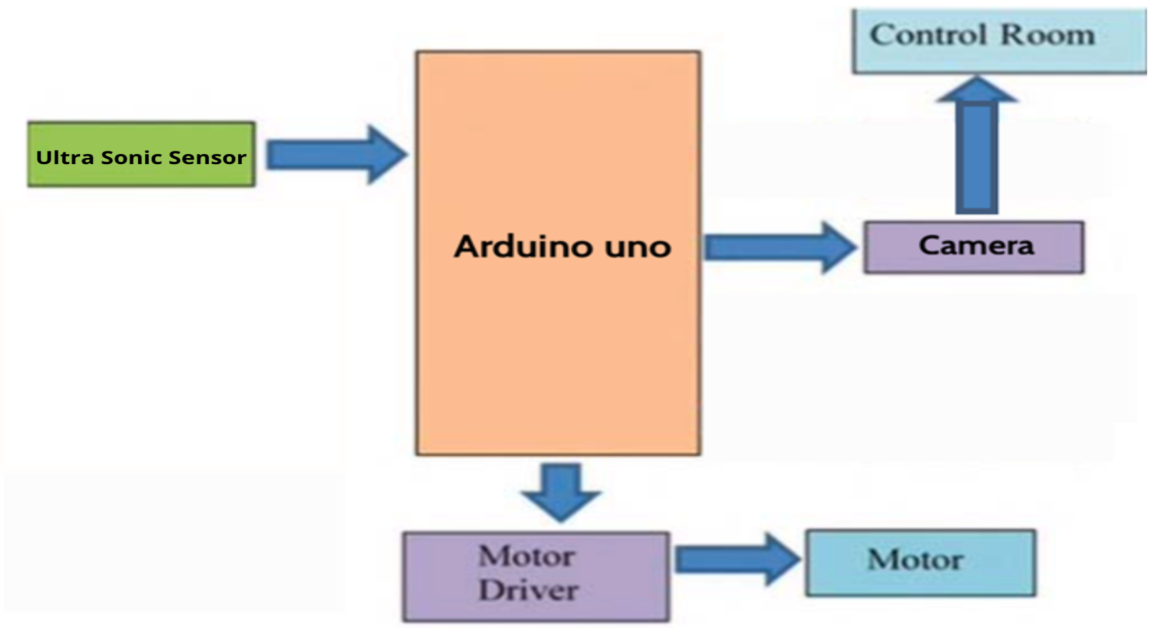

Fig. 1: Proposed Methodology

4. IMPLEMENTATION

We are making use of Arduino UNO as our main system controller. All the components that we have used in our model are automated by Arduino, which instructs them what to do based on the input received. The Arduino is powered by an external power source supplying 5V and 3A as the input. We use a L293d motor driver to allow the robot (in our case, a 4 wheeler chassis car) to move in the necessary direction depending on the input. The motor driver also acts as a bridge with the battery, allowing it to be damaged rather than damaging the motor itself. The Arduino is powered by the motor driver as well, which in turn powers the camera as well.

We have made use of an ultrasonic sensor HC SR04 to enable the robot to avoid obstacles as it detects any potential entities in its range. This is done by the transmitter mounted on the ultrasonic sensor which sends out ultrasonic waves and measuring the time which the waves take to reach back the sensor after reflection. For our model, we have set a detection range of 40cm, so whenever the sensor detects any object in the range of 40cm in front of it, it deters from that direction. In order to allow for constant object avoidance, the ultrasonic sensor checks for the presence of objects in its vicinity every 1s. Upon detection of an object, the ultrasonic sensor sends the information in the form of a digital input signal to the Arduino, and the Arduino upon receiving this signal, instructs the motor pins to turn accordingly - for example, if there is an object in front of the car, then the Arduino instructs the right motors to move, which causes a change in direction. If there is no object, then the car keeps on moving in a straight line.

The ESP32 camera which is connected to the arduino serves for providing live video feed to the user. It renders the output to a web interface which can be accessed by the user via the IP address of the Arduino. One important thing that needs to be noted is that the Arduino and the device that will be used by the user to access the live feed should be connected to the same Local Area Network (LAN).

In order to allow person detection , we have made use of OpenCV and a two-dimensional convolutional neural network model. The model consists of layers, including two dropout layers to prevent model overfitting. For training purposes, we have made use of a surveillance image dataset, consisting of about 2000 images , which are taken from different surveillance cameras. The final dense layer of the model is responsible for generating the model output. The CNN model is responsible for learning the features from the images and use it for detection of persons in the images and to avoid detecting any other random objects which may be present. For implementing the same, we have made use of various libraries of python. The implementation softwares that we have used consists of Arduino IDE and Google Colab.

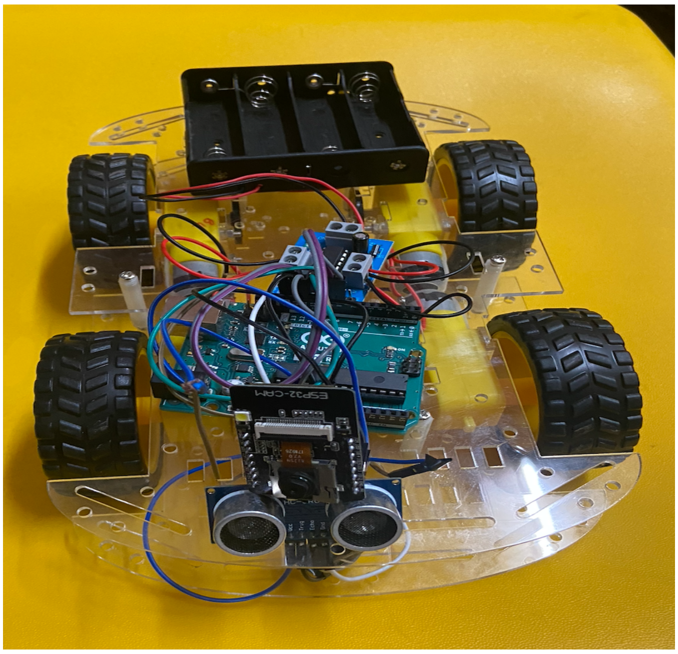

Fig. 2: Surveillance Car Top view

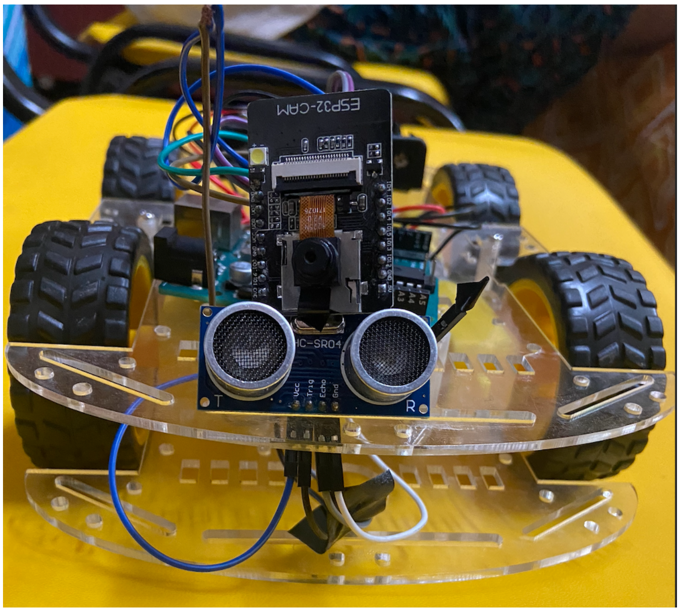

Fig. 3: Surveillance Car Front view

5. TESTING AND PERFORMANCE EVALUATION

We had divided our entire model implementation in separate modules and tested them individually. Once the modules were successfully implemented, they were integrated together to make the final model. There are mainly three modules which we have used for our system:the first one being the object avoidance function and the second one is the real time surveillance output and the third and final module is the image detection model. The modules are described below:

5.1 Testing object avoidance function

In the first stage of the experiment, we implemented the object avoidance algorithm, which helped the robot detect and avoid obstacles in its path. The main idea was to ensure that the robot does not get stuck in its path while following a potential intruder. To perform the same, we made a baseline model using ultrasonic sensor only.

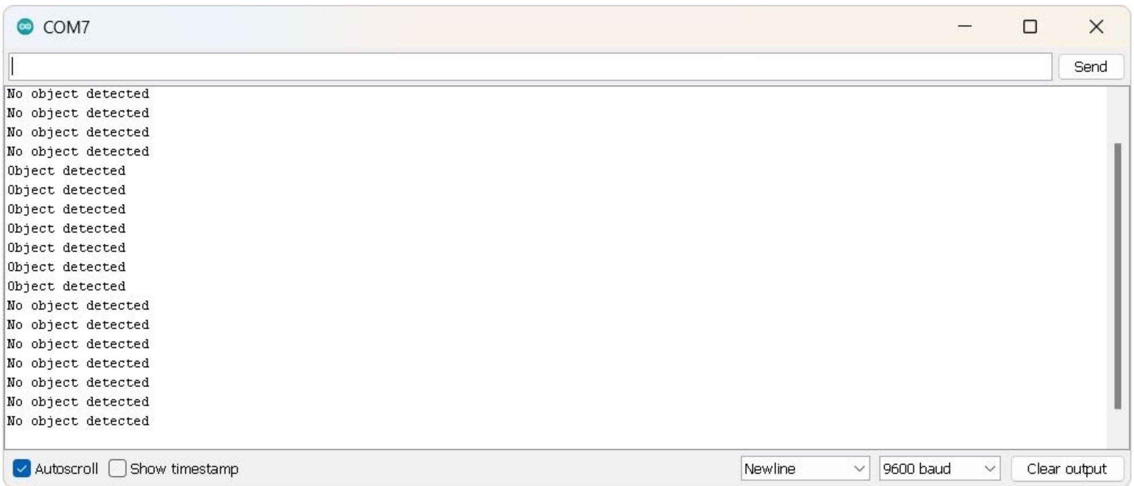

Fig. 4: Object Avoidance algorithm output on Serial monitor

Procedure:

This module is implemented by making use of the ultrasonic sensor. The sensor is attached to the head of the car, and depending on the sensing, the appropriate action is performed, and the same is coded in the Arduino. To test the model, we designed a simple obstacle course using cardboard boxes of varying sizes and shapes, placed in different positions. We then activated the robot and placed it at the starting point of the obstacle course, and monitored its movement as it navigated the course. We also placed an object in the path of the robot to see how it would react.

Observation:

The robot successfully cleared the path without colliding to the path boundaries, It also detected the obstacle placed in its path and deterred from it. Thus,we will be utilizing it in the final model since the robot performed in the intended way.

5.2 Surveillance function

The surveillance function is the next important criteria for our models' success.

Procedure:

We first connected the ESP32 camera to the Arduino and programmed it so that we can see the live video feed from any device - both mobile and website. Then, we supplied power to the Arduino and connected it to a phone, using the mobile's hotspot and the camera's IP address, to see whether it is working accordingly.

Result:

Using the esp32 camera, our system is currently able to do surveillance by live monitoring of the place to the end user through the website. Depending on the strength of internet connectivity, and considering the power supplied to it, live streams suffer from buffering.

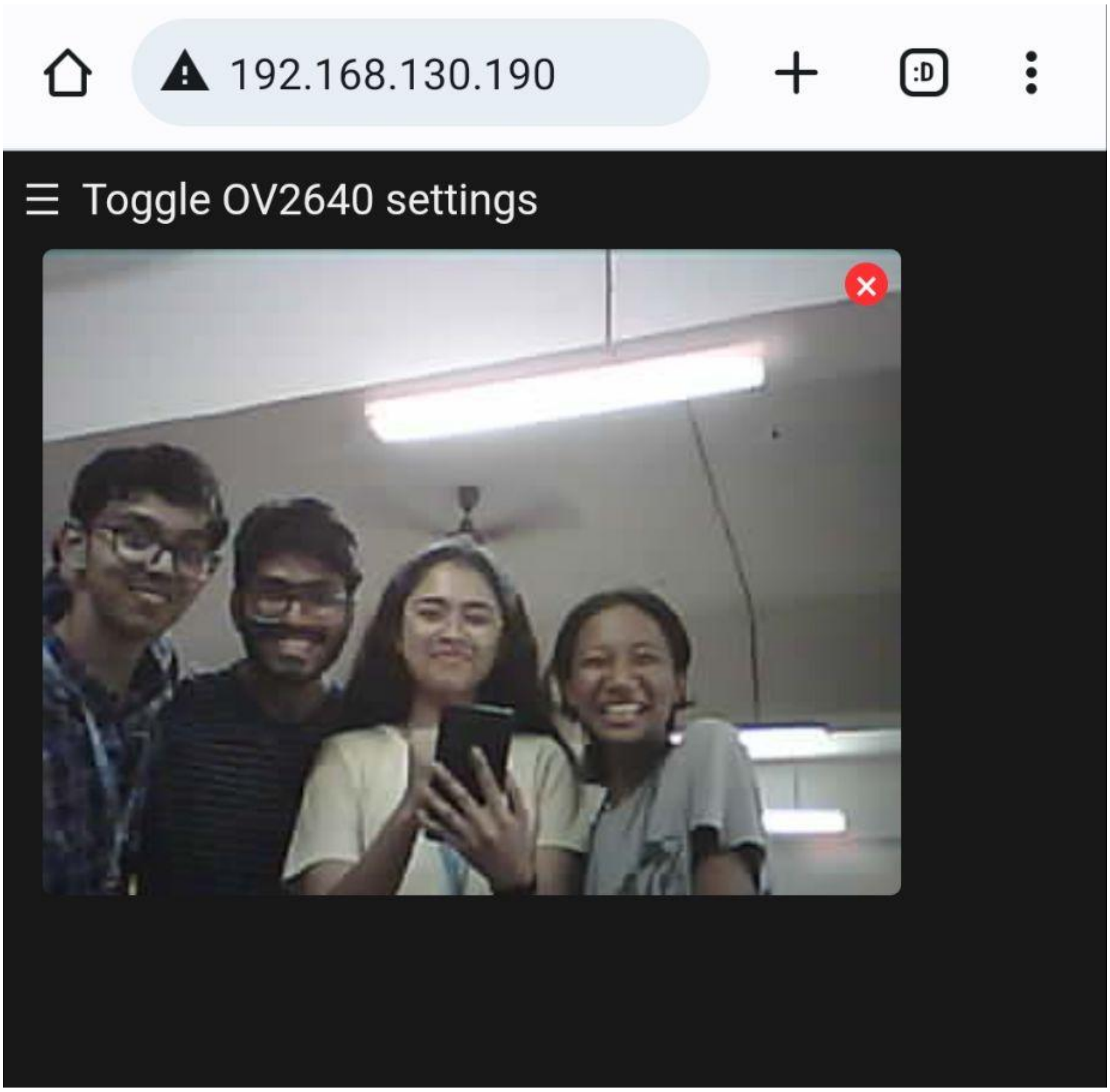

Fig. 5: Video Live Stream on server side which is being captured by esp32 cam

5.3 Image Detection Model

The image detection model that we have incorporated as a proposed work is a deep neural network, which makes use of convolutional layers to extract the features from the images and use them for identifying humans in the image, if any.

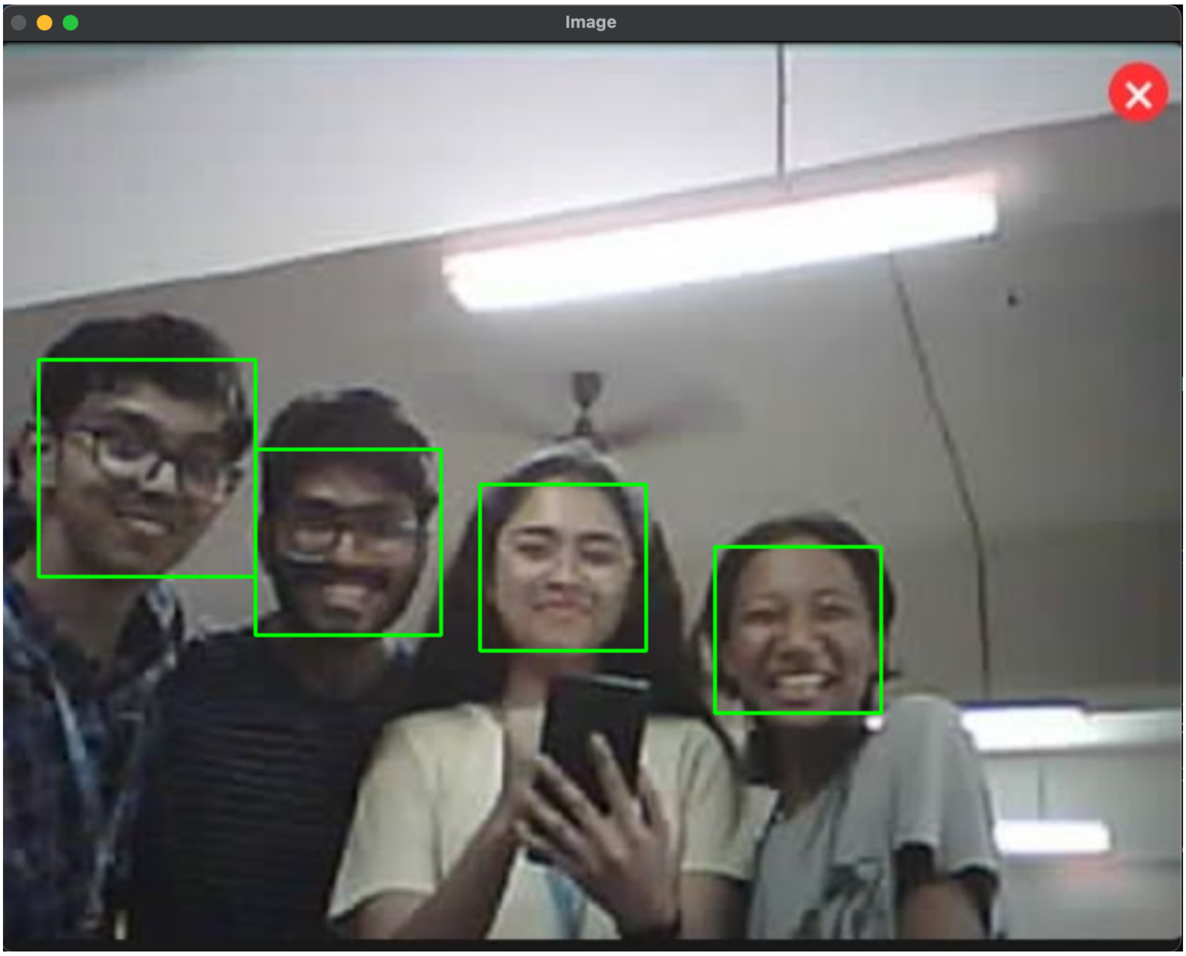

Fig. 6: Image Detection Model recognising people

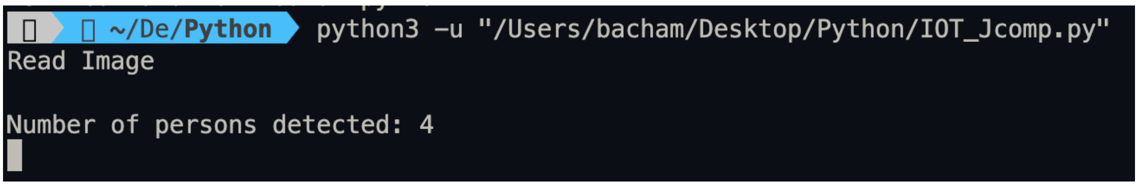

Fig. 7: Image Detection Model printing the count of humans in the picture taken by esp32 cam

Procedure:

After performing the preliminary data pre-processing and dividing the data into training and test sets, in the ratio 9:1, we proceeded with defining our deep learning model.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d (Conv2D)              (None, 478, 638, 64)      1792      
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 239, 319, 64)      0         
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 237, 317, 128)     73856     
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 118, 158, 128)     0         
_________________________________________________________________
dropout (Dropout)            (None, 118, 158, 128)     0         
_________________________________________________________________
flatten (Flatten)            (None, 2386432)           0         
_________________________________________________________________
dense (Dense)                (None, 128)               305463424 
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129       
=================================================================
Total params: 305,539,201
Trainable params: 305,539,201
Non-trainable params: 0
```

Fig. 8: Model architecture

Then, we proceeded with training the model using our training set. For training, the main performance metric that we have used is Mean Average Error or MAE. After training the model till the overall MAE was reduced, we proceeded with the model testing.

Observation:

The model is able to identify any persons present in the image given as input and also give an estimate of the number of people present in it.

Fig. 9: Model output

Finally, we integrated all the three modules which resulted in the final model that we have proposed. After the integration, all the modules are working as intended. The car constantly gives live surveillance to the end user, while deterring from its path whenever any object is detected.

6. CONCLUSION AND FUTURE ENHANCEMENTS

In this project, we have successfully designed and implemented an Arduino-based women safety night patrolling car using an ESP32 cam. The system uses ultrasonic sensors, which is used to implement object avoidance algorithms & also to detect any suspicious activity in the area and notify the authorities in real-time. The system also includes a live video streaming feature, which enables the authorities to monitor the area remotely.

The system has shown promising results in terms of its functionality and performance. However, there are still some areas of improvement that can be addressed in the future.

Future Enhancements:

Battery optimization: Currently, the system relies on a battery for power, which can be a limiting factor in terms of the system's runtime. Future enhancements can include optimizing the battery life by implementing power-saving techniques and using high-capacity batteries.

Better image processing: The system uses an ESP32 cam for image processing, which can be improved by using more advanced cameras and image processing algorithms. This can improve the accuracy and reliability of the system's surveillance capabilities.

Integration with AI: The system can be integrated with AI-based algorithms to improve the accuracy of its detection capabilities. This can enable the system to differentiate between real threats and false alarms more effectively.

Improved communication: The system's communication capabilities can be improved by using more advanced communication protocols and devices. This can enable the system to transmit data more quickly and reliably.

User interface: The system's user interface can be improved by designing a more user-friendly and intuitive interface. This can improve the ease of use and accessibility of the system for users.

7. APPENDIX

7.1 Code for ultrasonic sensor and motor for aiding in object avoidance:

```
const int trig = 13;
const int echo = 12;
const int in1 = 11;
const int in2 = 10;
const int in3 = 9;
const int in4 = 8;


void setup()
{
 pinMode(trig, OUTPUT);
 pinMode(echo, INPUT);
 pinMode (in1, OUTPUT);
 pinMode (in2, OUTPUT);
 pinMode (in3, OUTPUT);
 pinMode (in4, OUTPUT);
}
long dura, dis;


void loop()
{
 digitalWrite(trig, LOW);
 delayMicroseconds(2);
 digitalWrite(trig, HIGH);
 delayMicroseconds(10);
 digitalWrite(trig, LOW);
 dura = pulseIn(echo, HIGH);
```

```
dis = dura/58.2;
 if(dis<40)
   {
     digitalWrite(in1, LOW);
     digitalWrite(in2, HIGH);
     digitalWrite(in3, HIGH);
     digitalWrite(in4, LOW);
      delay(200);
     digitalWrite(in1, LOW);
     digitalWrite(in3, LOW);
     digitalWrite(in2, HIGH);
     digitalWrite(in4, LOW);
     delay(500);
     Serial.print("No object detected");
   }

 else
   {
     digitalWrite(in1, HIGH);
     digitalWrite(in2, LOW);
     digitalWrite(in4, HIGH);
     digitalWrite(in3, LOW);
   Serial.println("Object detected");
   }
 delay(0);
}
```

7.2 Code for ESP32 camera to render live video feed to the user:

```
#include "esp_camera.h"
#include <WiFi.h>
```

```
#define CAMERA_MODEL_AI_THINKER // Has PSRAM
#include "camera_pins.h"

const char* ssid = "AndroidAP";                //username and password of the device's
hotspot used
const char* password = "*******";

void startCameraServer();

void setup() {
  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
```

```
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;

  // if PSRAM IC present, init with UXGA resolution and higher JPEG quality
  //                      for larger pre-allocated frame buffer.
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

#if defined(CAMERA_MODEL_ESP_EYE)
  pinMode(13, INPUT_PULLUP);
  pinMode(14, INPUT_PULLUP);
#endif

  // camera init
  esp_err_t err = esp_camera_init(&config);
  if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
  }

  sensor_t * s = esp_camera_sensor_get();
  // initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
```

```
  }
  // drop down frame size for higher initial frame rate
  s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE) ||
defined(CAMERA_MODEL_M5STACK_ESP32CAM)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");

  startCameraServer();

  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}
```

7.3 Machine learning model for person detection:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import shutil
from sklearn.model_selection import train_test_split
import tensorflow as tf
from sklearn.model_selection import train_test_split



# getting the labels corresponding to the image
label_df = pd.read_csv('/kaggle/input/crowd-counting/labels.csv')
label_df.columns = ['id' , 'people']
label_df.head()

# loading the images in vector format
img = np.load('/kaggle/input/crowd-counting/images.npy')
#img = img.reshape(img.shape[0], img.shape[1], img.shape[2], img.shape[3],1)
img.shape
labels = np.array(label_df['people'])
labels

# setting features and target value

x_train, x_test, y_train, y_test = train_test_split(img, labels, test_size=0.1)
print(x_train.shape[0])
print(x_test.shape[0])

# create model

model = tf.keras.Sequential([
```

```
    tf.keras.layers.Conv2D(64, (3,3), input_shape=(480,640,3),
activation=tf.keras.activations.relu),
    tf.keras.layers.MaxPool2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation=tf.keras.activations.relu),
    tf.keras.layers.MaxPool2D(2,2),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation=tf.keras.activations.relu),
    tf.keras.layers.Dense(1)

])

model.compile(loss=tf.keras.losses.Huber(), optimizer=tf.keras.optimizers.Adam(),
metrics=['mae'])
model.summary()

# adding a learning rate monitor to get the lr with smoothest prediction

lr_monitor = tf.keras.callbacks.LearningRateScheduler(
                lambda epochs : 1e-8 * 10 ** (epochs/20))

# train the model

history = model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=20,
batch_size=32, callbacks=[lr_monitor])

# plot mae
plt.semilogx(history.history['lr'], history.history['loss'])
plt.axis([np.min(history.history['lr']), np.max(history.history['lr']),
np.min(history.history['loss']), 15])
plt.show()

# changed the learning rate to 1e-5 and re-ran the model
```

```
model.compile(loss=tf.keras.losses.MeanSquaredError(),
optimizer=tf.keras.optimizers.Adam(lr=1e-6), metrics=['mae'])
model.summary()

# training the model

history = model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs=20,
batch_size-32)
# plot mae
plt.plot(history.history['mae'])
plt.plot(history.history['val_mae'])
plt.legend(['mae', 'val_mae'])
plt.ylim(1, 4)
plt.xlim(0, 50)

plt.xticks(np.arange(0,50, 5))

plt.xlabel('epochs')
plt.ylabel('mean absolute error')
plt.title('Mae in every epoch')
plt.show()

import matplotlib.pyplot as plt
from mpl_toolkits.axes_grid1 import ImageGrid


# set figure size

fig = plt.figure(figsize=(15,15))
grid = ImageGrid(
    fig, 111,
    nrows_ncols=(2,2),
```

```
    axes_pad=0.5
)

for x in range(0,4):

    grid[x].set_title('Number of people => ' + str(labels[x]))
    grid[x].imshow(img[x])
```

8. References

1. M. Aggravi, G. Sirignano, P. R. Giordano and C. Pacchierotti, "Decentralized Control of a Heterogeneous Human–Robot Team for Exploration and Patrolling," in IEEE Transactions on Automation Science and Engineering, vol. 19, no. 4, pp. 3109-3125, Oct. 2022, doi: 10.1109/TASE.2021.3106386.

2. L. Huang, M. Zhou, K. Hao and E. Hou, "A survey of multi-robot regular and adversarial patrolling," in IEEE/CAA Journal of Automatica Sinica, vol. 6, no. 4, pp. 894-903, July 2019, doi: 10.1109/JAS.2019.1911537.

3. S. -C. Hsia, S. -H. Wang, B. -Y. Wang and C. -Y. Chang, "Intelligent Surrounding Recognition for Robot Direction Control," in IEEE Access, vol. 8, pp. 200275-200282, 2020, doi: 10.1109/ACCESS.2020.3035617.

4. A. Raganna, Nithesh K., Neha B., Omchandra V. Shrivastav, Praveen T. Musaguppi, “Iot Based Night Patrolling Robot for Women Safety”, IJMA, vol. 10, no. 2, pp. 3886 - 3894, May 2021.

5. J. N. Amrutha and K. R. Rekha, “Night Vision Security Patrolling Robot Using Raspberry Pi", IJRESM, vol. 3, no. 8, pp. 432–436, Aug. 2020.

6. Irfan Ahmad Pindoo, et. al. International Journal of Engineering Research and Applications, ISSN: 2248-9622, Vol. 11, Issue 5, (Series-VI) May 2021, pp. 01-07

7. International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538 Volume 10 Issue

VI June 2022

8. D. C. Jullie Josephine, G. N. P, S. Frank, A. Manikandan, S. Bhuvaneskumar and T. Sakthivel, "Night Patrol Robot for Detecting and Tracking of Human Motions using Proximity Sensor," 2022 International Conference on Edge Computing and Applications (ICECAA), 2022, pp. 912-915, doi: 10.1109/ICECAA55415.2022.9936381.

9. , Hasan Salman, Samuel Acheampong, He Xu,"Web-Based Wireless Controlled Robot for Night Vision Surveillance Using Shell Script with Raspberry Pi, Complex, Intelligent, and Software Intensive Systems", 2019, Volume 772; ISBN : 978-3-319-93658-1

10. A. Taylor and L. D. Riek, "REGROUP: A Robot-Centric Group Detection and Tracking System," 2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2022, pp. 412-421, doi: 10.1109/HRI53351.2022.9889634.

11. Bhatia, Sandeep and Verma, Soniya and Singh, Nidhi and Saxena, Isha and verma, Ishika," IOT and AI Based Women's Safety Night Patrolling Robot" ,July 14, 2022.

12. Gadhawe, N., Kohade, B., Dongre, S., Urkude, R., Tayde, R., Divya, B.N., Hegde, B., Chaithanya, B.R., Raju, M.M., Shambhavi, S. and Kitchagiri, P., 2021. Night Patrolling Device Using IOT. IJSRD-International Journal for Scientific Research & Development, 9(1), pp.2321-0613.

13. Gavaskar, K., Ragupathy, U.S., Elango, S. et al. A novel design and implementation of IoT based real-time ATM surveillance and security system. Adv. in Comp. Int. 2, 1 (2022).

14. Bhardwaj, R., Bera, K., Jadhav, O., Gaikwad, P. and Gupta, T., 2018. Intrusion Detection through Image Processing and getting notified via SMS and Image.

15. Lin Rui, Du Zhijiang and Sun Lining, "Moving object tracking based on mobile robot vision," 2009 International Conference on Mechatronics and Automation, 2009, pp. 3625-3630, doi: 10.1109/ICMA.2009.5246022.