International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-07, Issue-02, MAY 2021

Security Patrolling Robot

[1]Rutuja Mandale, [2]Kiran Mahamunkar, [3]Dr. Shikha Nema

[1,2]Student, [3]HOD of Electronics and Communication Department, [1,2,3]Usha Mittal Institute of Technology- SNDT, Mumbai, India, [1]mandale.rutuja12@gmail.com, [2]kiranm1999.km@gmail.com, [3]shikha.Nema@umit.sndt.ac.in

Abstract: The pandemic brought an exponential surge in the number of thefts and robberies. Absence of security guards due to this unforeseen situation was a prime factor behind this surge. Moreover, each year, companies and homeowners set aside a large budget only for security measures. To address these issues, we propose a security patrolling robot, eliminating or minimizing the need for a human security guard. The line following robot moves along a particular path and is equipped with camera and sensors. The faces detected by this robot are sent to a server. This server runs the face recognition algorithm on this image. We have used OpenCV, Python and deep learning for face detection and recognition. Deep learning approach is more efficient as compared to non - deep learning approaches since it requires a smaller dataset. If the face detected doesn't match with the faces in the dataset then the buzzer on the robot will switch on. We also vary the tolerance factor of our face recognition algorithm and put forward the factor which gives the most accurate result.

Keywords* — *deep learning, face detection, face recognition, line follower, OpenCV, RaspberryPi, security

I. Introduction

The pandemic brought an exponential surge in the number of thefts and robberies. Absence of security guards due to this unforeseen situation was a prime factor behind this surge. FBI data for the first six months of 2020 shows murder and non-negligent homicide as up nearly 15 percent compared to the same time period last year. According to a report by Research Now, 70 percent thefts in India are home thefts. Moreover, each year, companies and homeowners set aside a large budget only for security measures. On an average a security guard's annual salary is around 25,000 USD. Its expense increases with the number of alarm systems, surveillance cameras and hired security guards. It is not feasible to have multiple guards for patrolling of the entire area simultaneously. Though manual registration is made compulsory where people have to provide personal details like name, mobile number, reason of visiting, etc.; but these details can be misleading and fraudulent. Also life of security guard is at risk under extreme circumstances, where the intruder can attack the guard which may lead to physical damage or fatal.

In the existing system, some human intervention is needed to monitor the activity recorded by the cameras. So here we propose an autonomous security patrolling robot eliminating or minimizing the need for a human security guard. The system is automatically alarmed when the recognized face does not belong to the dataset of authorized faces.

II. Literature Survey

A night vision patrolling robot is proposed in [1]. In this proposed system, the robot is mounted with night vision camera. It starts capturing videos and images upon sound detection and sends it to the Blynk server. In this system a person has to remotely monitor if there are any intrusions happening in the footages transmitted by the robot. This system also uses Node MCU for connecting to Wi-Fi.

In the patrolling robot proposed [2] in addition to the previous system, this system uses Local Binary Pattern Histogram for face detection and recognition. In our proposed system we use convolutional neural networking algorithm which is more efficient than LBPH.

This system uses Eigen face approach for face detection [3]. In our proposed system we use convolutional neural networking algorithm which is more efficient than Eigen face approach.

The Raspberry Pi is a crucial embedded structure and a single board PC used to reduce the multifaceted idea of systems constantly applications for image processing. This [4] uses viola-jones algorithm which uses cascaded xml file for the face detection. For having optimum speed for face detection on raspberry pi, the reduction in the camera resolution will increase the frame per second.

The surveillance systems are building up with multiple cameras which are placed in different angles of view to track human objects this increases the number of cameras

 DOI : 10.35291/2454-9150.2021.0228

used in the system [5]. An intelligent surveillance system with multiple cameras is complicated and costly. In this system a single camera is installed on the robot and it can move in all directions to take photos at different angles. Two gear motors are sufficient to produce the movement of spy robot and the motor driver module is used to supply enough current to drive two gear motors which protects the Raspberry-pi module from the damage. The major advantage of using the minimum number of gear motor is minimizing the power consumption. The entire system is based on a Raspbian operating system like Linux platform.

The setup of the home security system involves the use of a Raspberry Pi 3 model B to which a Passive Infrared Sensor (PIR sensor) and a webcam are connected [6]. The Raspberry Pi 3 model B uses a 1.2 GHz quad-core ARM Cortex-A53 for processing, while the PIR sensor is used to detect motion and the webcam is used to perform face detection and image capturing processes using OpenCV image processing library and Python programming language. During implementation of the system the PIR sensor is setup to detect motion in the secured environment, if motion is detected then the Raspberry Pi 3 model B makes use of the webcam to initiate the face detection process and on finding a face captures an image of the secured environment. The captured image is then sent to the E-Mail account of the owner using the concept of IoT, SMTP and MIME.

III. Problem Definition

The main aim of our proposed system is to make intrusion detection fully autonomous. In the system proposed in [1], there is no processing algorithm on the images captured by the robot. Hence human supervision is necessary to detect intrusion. The systems proposed in [2] and [3] use LBPH and Eigen face approach algorithms respectively. These two algorithms are non- deep learning algorithms and require larger dataset as compared to deep learning algorithms. Hence we see convolutional neural networking approach fit for our system.

IV. Block Diagram of Proposed System

Fig. 1 illustrates the block diagram of security patrolling robot's system. All the components are connected to the main controller i.e., RaspberryPi 4. The power supply which is connected to raspberry pi will switch on the circuit. The 5 MP Raspberry Pi 4 Model B camera module will capture images.

HC-SR04 Ultrasonic sensor, Infrared Sensor (IR), and Sound sensor is connected to Raspberry Pi module and are mounted on the robot chassis. L293D motor driver is connected to two motors which are connected to the wheels of the robot. Also a buzzer is connected to Raspberry pi. The images captured by the camera will be transmitted to a Django Framework server. Face recognition algorithm is programmed at the backend of this server and the system is alerted if the captured image does not match with any of the images in the dataset of the stored authorized faces.

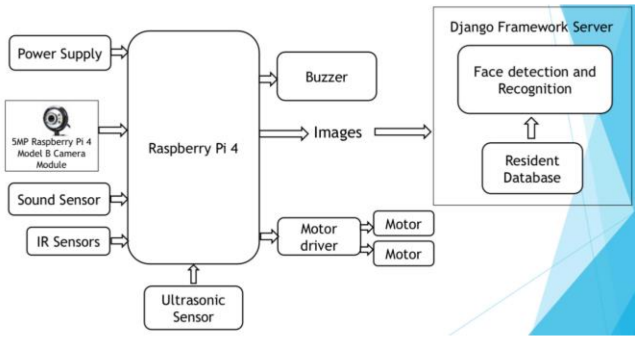

Fig. 1 : Block Diagram

V. System Flow Chart

Fig. 2 depicts the flow chart of our system. Once powered up, the robot starts following a predefined line in the direction of sound. If the robot detects any human face along its path, it stops and captures the image and starts transmitting it immediately to the Django server where the dataset of the residents or authorized people and face recognition algorithm is already stored. If the face recognized belongs to the dataset, the robot continues moving. If not, then the buzzer on the robot will switch on thereby alerting the system.

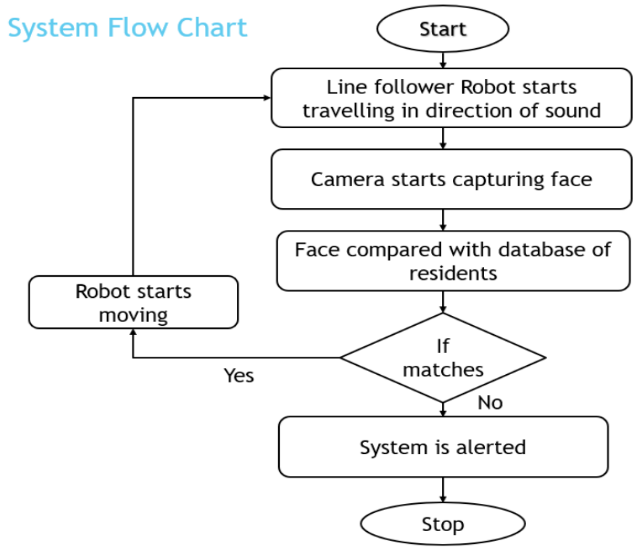

Fig. 2 : System Flow Chart

VI. Hardware: Line Follower Robot

We have designed our line follower robot using IR and Ultrasonic sensors all connected to Raspberry Pi. An optional mic is also connected for sound triggered operation. IR sensors help the robot to follow a black line path and ultrasonic sensor help the robot for obstacle detection.

A. Sensors

1) IR Sensor Module: IR sensors help the robot to follow the black line. The IR sensor module has an IR transmitter and receiver. The IR transmitter transmits

DOI : 10.35291/2454-9150.2021.0228

the light and the receiver waits for the transmitted light to return back. IR light is completely reflected by a white surface and completely absorbed by a black surface.

2) Ultrasonic sensor: The Ultrasonic Transmitter in the Sensor generates a 40 KHz Ultrasound. This signal then propagates through air. If there is any obstacle in its path, the signal hits the object and bounces back. This bounced signal is then collected by the Ultrasonic receiver. Based on the signal's time of travel, the distance of the object can be calculated as the speed of sound.

B. *Principle*

The two IR sensors are connected at the front of the robot in such a way that they point on the two extreme edges of the black line. If none of the sensors detect the black line then the pi instructs the motors to move forward. If the left sensor detects the black line then the pi instructs the robot to turn left by rotating the right wheel alone. If right sensor detects the black line then the pi instructs the robot to turn right by rotating the left wheel alone. If both sensors detect the black line, the robot stops. In such a way the robot follows the black line.

Fig. 3 shows how our line follower robot looks

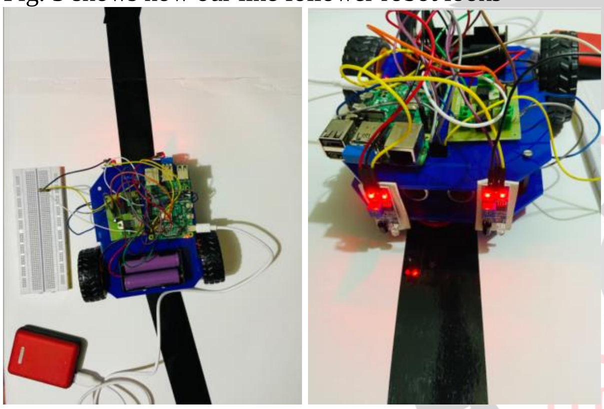

Fig. 3 : Line Follower Robot

VII. SOFTWARE

A. *Face Detection using OpenCV and Python*

Face detection using Haar cascades is a machine learning based approach where a cascade function is trained with a set of input data.

1) The detection works only on grayscale images. So it is important to convert the color image to grayscale.
2) detectMultiScale function is used to detect the faces. It takes 3 arguments the input image, scaleFactor and minNeighbours.
3) faces contains a list of coordinates for the rectangular regions where faces were found. We use these coordinates to draw the rectangles in our image.

An example of face detection is shown in Fig. 4

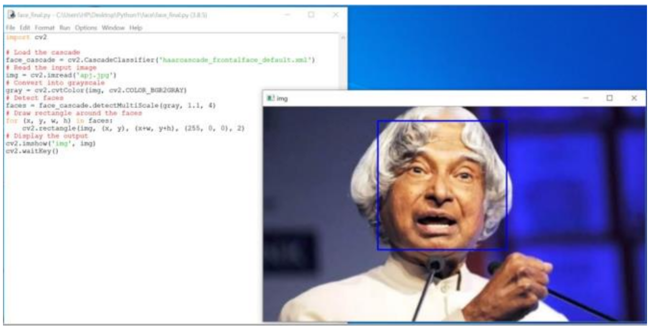

Fig. 4 : Face Detection

B. *Face Recognition*

For face recognition we have used convolutional neural networks. It involves the following steps:

1) The detection works only on grayscale images. So it is important to convert the color image to grayscale.
2) Calculate the face encodings of the detected face using dlib library
3) Compare these face encodings to the face encodings of the images stored in the dataset

C. *Django REST framework server*

We have created this server using the REST framework. The images captured by the robot are sent to this server. The face detection and recognition algorithms are programmed in this server and the results are displayed on it. This server can be accessed from anywhere on the same Wi-Fi network. Fig. 5 shows the basic look of our server.

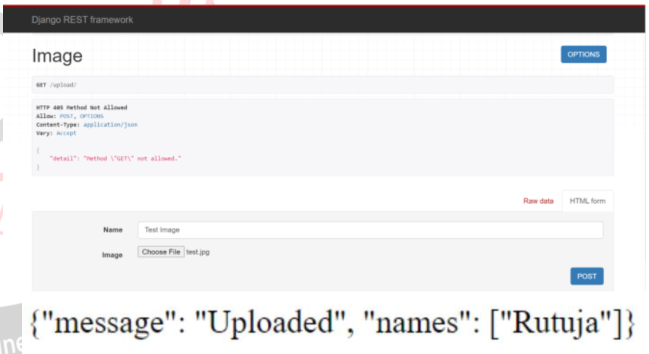

Fig. 5 : Django Framework Server

VIII. RESULTS

Table 1 depicts the results of our face recognition algorithm. Here we have varied the tolerance of the face recognition algorithm and observed the output against various test cases.

It is observed that lower the tolerance value (i.e., 0.1-0.3), stricter is the comparison. Hence to avoid false negatives, we have to choose higher values. The tolerance factor 0.4-0.5 work well for our algorithm. At tolerance factors higher than these values, false positives are observed. Hence, tolerance of 0.4-0.5 is ideal for our system.

Table 1: Results

Tolerance	Actual	Output	Accuracy
0.3	Person 1	Intruder	0
	Person 2	Intruder	
	Person 3	Intruder	
	Intruder	Intruder	

 DOI : 10.35291/2454-9150.2021.0228

0.4	Person 1	Person 1	100
	Person 2	Person 2	
	Person 3	Person 3	
	Intruder	Intruder	
0.5	Person 1	Person 1	100
	Person 2	Person 2	
	Person 3	Person 3	
	Intruder	Intruder	
0.6	Person 1	Person 1	25
	Person 2	Person 2,3	
	Person 3	Person 1,2,3	
	Intruder	Person 1,2	
0.9	Person 1	Person 1,2,3	0
	Person 2	Person 1,2,3	
	Person 3	Person 1,2,3	
	Intruder	Person 1,2,3	

IX. CONCLUSION

The deep learning approach to face recognition presented in this paper does not require a large dataset, i.e., only one or two images per person are sufficient for the neural network to correctly identify the person. Other non-deep learning approaches like LBP and Eigen face approach require a large dataset and hence are less efficient as compared to convolutional neural networks as far as security systems are concerned. Because we have used deep learning, new additions of authorized people can be easily done just by feeding one or two images of the person to the algorithm. From our findings we conclude that face recognition using convolutional neural networks with tolerance factor 0.4 - 0.5 gives accurate results and is ideal for security systems.

Further, we can extend our system by creating a mobile application where the alerts pop up on the user's mobile, and the user can know the exact time and location of the intruder. The scope of this system is not only related to indoor security systems but can also be extended to secure international borders and other areas which require high security.

ACKNOWLEDGMENT

Foremost, we would like to express our sincere gratitude to our guide Professor Dr. Nema Shikha, Head of Department (Electronics and Communication) for her valuable guidance and continuous support during the whole process of this project; her patience, motivation, enthusiasm, and immense knowledge. Her direction and mentoring helped us to work successfully on the project.

Our sincere gratitude to Dr. Sanjay Pawar, Principal (Usha Mittal Institute of Technology) for his valuable encouragement and insightful comments.

We would also like to thank to all the teaching and non-teaching staff for their valuable support.

Last but not the least we would like to thank to our parents and friends.

REFERENCES

[1] P. Manasa, "NIGHT VISION PATROLLING ROBOT", Journal of Xi'an University of Architecture & Technology, Vol.12, no.5, Bangalore, India-2020.

[2] J. Amrutha and K. Rekha, "Night Vision Security Patrolling Robot Using Raspberry Pi", *International Journal of Research in Engineering*, Science and Management, vol.3, no.8, Banglore, India, 2020.

[3] I. Gupta, V. Patil, C. Kadam and S. Dumbre, "Face Detection and Recognition using Raspberry Pi", in *IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, Pune, India, 2016, pp.83-86.

[4] V. Bhanse and M. Jaybhaye, "Face detection and tracking using image processing on raspberry pi", in *International Conference on Inventive Research in Computing Applications*, Pune, India, 2018, pp.1099-1103.

[5] G. O. E. Abdalla and T. Veeramanikandasamy, "Implementation of spy robot for a surveillance system using Internet protocol of Raspberry Pi," 2017 *2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore,

[6] D. Abhilash, C. Chandrashekar and S. Shalini, "Economical, energy efficient and portable home security system based on Raspberry Pi 3 using the concepts of OpenCV and MIME," *2017 International Conference on Circuits, Controls, and Communications (CCUBE)*, Bangalore, 2017, pp. 60-64, doi: 10.1109/CCUBE.2017.8394155.

[7] K. Goyal, K. Agarwal and R. Kumar, "Face detection and tracking: Using OpenCV," *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, 2017, pp. 474-478, doi: 10.1109/ICECA.2017.8203730.

[8] V. K. Sharma, "Designing of Face Recognition System," *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India, 2019, pp. 459-461, doi: 10.1109/ICCS45141.2019.9065373.

[9] Ning Jiang, Yijun Lu, Shaopeng Tang and S. Goto, "Rapid face detection using a multi-mode cascade and Separate Haar Feature," *2010 International Symposium on Intelligent Signal Processing and Communication Systems*, Chengdu, 2010, pp. 1-4, doi: 10.1109/ISPACS.2010.5704623.

[10] H. A. Patel and R. D. Rajput, "Smart Surveillance System Using Histogram of Oriented Gradients (HOG) Algorithm and Haar Cascade Algorithm," *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, 2018, pp. 1-4, doi: 10.1109/ICCUBEA.2018.8697464.

 DOI : 10.35291/2454-9150.2021.0228