Intelligent Surveillance Robotics:

Integrating Wireless Communication and Computer Vision for Enhanced Human Activity Recognition

Sunil Tayade
E&TC, VIT
Bibewadi, Pune 411037
sunil.tayade@vit.edu

Rushikesh Gholap
E&TC, VIT
Bibewadi, Pune 411037
rushikesh.gholap21@vit.edu

Arjun Lande
E&TC, VIT
Bibewadi, Pune 411037
arjun.lande21@vit.edu

***Abstract*-**
The proposed paper introduces a cutting-edge solution for enhancing surveillance in restricted public areas using a modern robotic approach. This robot is equipped with a wireless camera that can capture real-time footage during both day and night. The robot's design allows it to be easily controlled through a mobile app, leveraging an ESP Wifi module for seamless communication with an Android device. To enhance its capabilities, the robotic vehicle is integrated with a Machine Learning model through a Raspberry Pi for software processing. The Blynk App enables users to manually control the robot's movements based on commands received from the Android device, significantly reducing the need for human presence in hazardous environments that require continuous supervision and security. This system aims to autonomously identify various human activities through live video streaming, thanks to the integration of a machine learning model. The Android application not only facilitates remote control but also enables users to navigate the robot from a substantial distance using WIFI communication. Looking forward, the project holds promise for future advancements that could extend its applications to defense and mining areas. The robot is designed to distinguish between different types of human activities, monitoring live streaming information and transferring it to a connected Android device.

Index Terms- *Surveillance, robot, spy, supervision.*

I. INTRODUCTION

The fusion of wireless communication technology with camera systems has brought about remarkable advancements with far-reaching implications in the realms of robotics and surveillance. This study endeavors to craft an advanced surveillance robotic vehicle by leveraging key components such as the ESP8266 module, a high-resolution integrated camera, a user-friendly Android app for remote control and real-time streaming, and a machine learning model capable of discerning various human activities like falling, chit-chatting, and walking. The overarching goal is to facilitate understanding of human presence and ongoing activities in locations where direct human interaction is challenging. The integration of the ESP8266 module, known for its robust Wi-Fi connectivity, is a pivotal aspect of this project, enhancing the robotic vehicle's ability to establish a stable wireless communication channel.

The suggested methodology uses CV and ML algorithms to identify three human activities—walking, talking, and fallen—in order to get beyond these restrictions and limitations. Human activity can be detected even in without the presence of gear, sensors, or attachments. The work that has been put into reality aims to create a model that can precisely identify human behavior. The paper is divided into many sections in order to accomplish this goal. An essential feature of the system is its capacity to capture live streaming video through a meticulously incorporated high-definition camera. The recorded footage is then transmitted to a designated Android application via the ESP8266 module. The app's simple UI enables users to operate the car from a distance, keep an eye on the environment in real time, and make quick judgements using visual clues. The development of an Android application, serving as a user interface for remote control and live streaming reception, adds another layer of sophistication to this comprehensive approach. Moreover, the real-time video input from the camera serves a dual purpose: aiding in human activity recognition and providing valuable data for preprocessing. The extracted features are then classified according to predicted activity classes. The study also encompasses the development of an Android application, serving as a user interface for remote control and live streaming reception. This comprehensive approach underscores the potential of the surveillance robotic vehicle in addressing diverse surveillance challenges.

II. LITERATURE REVIEW

The study by T. Akilan presents an IoT-driven spying robot that is managed by Arduino UNO and controlled by a PC and smartphone. The robot streams wireless cameras in real-time day and night while recording hazardous surroundings, eliminating the need for humans to participate. PIR, ultrasonic, and gas sensors connected to an Arduino improve monitoring. This innovation indicates uses for increased security in mining, border surveillance, and defense [1].

In the study, a cost-effective surveillance robot is introduced that makes use of an Arduino microcontroller and easily available Android smartphones, greatly lowering the system's traditional high costs. For remote control and real-time visual feedback, the smartphone's built-in features—such as the camera, GPS, and communication systems—are deployed. The robot has additional features including rescue operations in a variety of locations [2].

In order to solve security problems in limited and dangerous zones, the publication proposes an IOT-based surveillance robot.

The robot, which can be controlled manually using Wi-Fi and the Blynk app, has ultrasonic and PIR sensors for obstacle detection.

The system's goal is to increase monitoring in dangerous regions while lowering human risk and giving administrators access to real-time data. The robot's useful performance in risky environments is illustrated by the potential for autonomy and alarm systems for danger alerts [3]. In this study, an innovative approach for monitoring outdoor security is presented. This robot can use an RFID tag to determine if a person is authorized or not, and a metal detector sensor can find metal explosives. A stepper motor is utilized to rotate the wireless camera in a 360-degree direction while it is placed on the robot to give us with continuous streaming of the designated outside region. The person and metal detection message are sent using a GSM module [4].

In the study, a flexible cognitive robot for human rescue missions in hazardous environments is proposed. The robot is controlled manually via wireless human input as well as autonomously by sensors. It uses mounted cameras for feedback together with IR and PIR sensors for navigation. The system's capacity to operate in manual, automated, and stepping modes allows for flexibility in a variety of situations. The robot uses a two-level, cost-effective human detecting technology that includes PIR, IR, and an IP camera to reduce human risk during rescue operations. The design's performance is confirmed by actual experiments and simulations, giving it a dependable option for human detection and rescue in challenging situations [5]. In order to prevent enemy penetration and protect military life, the research paper introduces a robot made for border and battle zones monitoring. The robot has a wireless camera with night vision that transmits real-time recordings to lower dangers and illegal actions. The study focuses on creating the Android software that is used to operate the robot model, which was created using the MIT software Inventor platform. Future improvements are suggested, including the use of gas sensors and a bomb defusal kit [6].

In the study, a robot that can be operated by a smartphone app and connects Arduino UNO and Android devices through Bluetooth is introduced. It highlights the robot's flexibility, quality, and consistency, which enable it to be used in a variety of applications. The study highlights the robot's capacity for spying, which is made possible by a wireless camera that records photos, videos, and audio on a laptop. Its value is increased by the use of sensors, which make it possible to identify obstacles and move on its own. With the aim of making additional improvements, it is stated that GSM integration may extend the communication range [7].

The study describes a surveillance robot created to increase battlefield security by advance gathering enemy information for planning. The robot, which uses IOT to be controlled, has a laser gun for quick action and allows remote control using an Android phone. A laser guns, wireless camera streaming, detecting sensors, GSM technology, human and automated control, and other stages are all included in the project. The robot's flexible uses include bomb detection and disaster relief, all with the goal of reducing deaths and enhancing safety [8].

III. METHODOLOGY

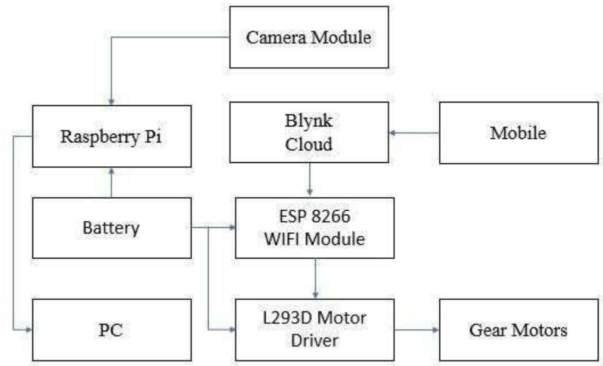


Fig. 1 System Block Diagram

Fig. 1. presents the block diagram outlining the proposed model, detailing the essential components and their connections. The central control unit guiding the system is the ESP 8266Module, responsible for managing the vehicle's motion. Acting as the control interface for the vehicle, the Bylnk App empowers users to adjust its direction and speed, serving as a mobile-based remote control. Driving the physical movement of the vehicle are the L293D Motor Driver and 360-degree DC motors, functioning analogously to a car's wheels. The motor driver enables precise control over the motors, influencing the overall movement of the vehicle. A critical sensory component is the camera module utilized by the authors. This module captures live real-time video input, transmitted to the Raspberry Pi for image processing. In this phase, image preprocessing and feature extraction occur, laying the foundation for activity recognition. The processed outcomes are then displayed on a PC, providing a visual representation of the recognized activities. Powering the Raspberry Pi and ESP 8266 Module is a 5V power supply, ensuring stable and dependable operation. Concurrently, the four DC motors receive power from a 12V battery supply, sustaining the physical movements of the vehicle. This comprehensive setup seamlessly integrates hardware and software components, showcasing a well-coordinated system for surveillance and real-time activity recognition in practical scenarios.

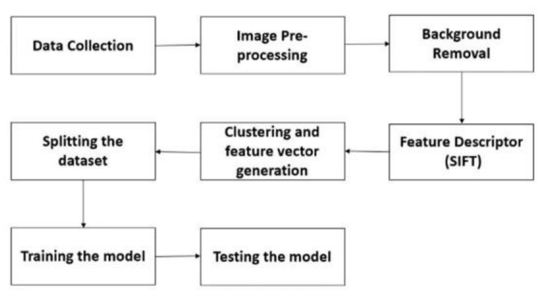


Fig. 2. Block Diagram of object detection using CV model

From Fig. 2 we performed the required operations for preprocessing as well as all the feature extraction and model testing and training. Below given are the process performed.

A) Data collection

For this model, the data was gathered independently. This model's data was gathered. The pictures come in three different file types: jpeg, jpg, and png. The photos in the data are automatically converted to JPG format when they are sent for pre-processing. For the Fall class, 390 pictures were gathered, 420 images for the walking class, and 460 images for the chit-chatting class.

TABLE I. DATASET DESCRIPTION

Class	Total Images	Format
Fall	370	.jpg
Walking	369	.jpg
Chit-Chatting	372	.jpg

TABLE I. shows the dataset that contain different images of Fall human, Walking, Chit-Chatting of two or more human.

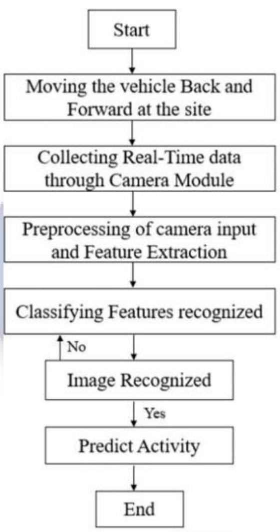


Fig.3 Flowchart of the Idea implemented

The flow of the project working is explained in the flowchart Fig.3. First the Vehicle with the camera is set in area where there is presence of human in the surrounding. Realtime images are being captured by the webcam attached externally to obtain the images of the activities of human. Preprocessing of images from camera module and the feature extraction is done in the cv model.

Depending on the feature extraction of the images, classification of feature recognition is done. Activities are being distinguished on the basis of features classification and recognition. If the images are being recognized by the model, then it is sent to further processing. If the images are not recognized it is again sent to the image recognition, and then again, the image is processed. After the proper recognition the activities are identified. Then this information is observed by the end user monitoring and controlling the vehicle from outside the site. After the activities are recognized, the necessary actions are being performed with respective of understanding the condition at the vehicle site.

- Algorithm for Proposed Model implementation

Algorithm 1: Algorithm of working

1. Start
2. for every image in dataset
3. apply image processing
4. Collecting the real time activity data through camera module
5. Pre-processing and feature extraction of camera input
6. Extract features
7. Activity Recognition
8. *if (Predicted class == True)*
9. Activity Recognized
10. Send the information to model
11. Show the predicted class on the Screen
12. *else*
13. Repeat step 5
14.
15. *end of* Algorithm

The Algorithm 1 initiates by processing images from a dataset, applying image processing techniques, and collecting real-time activity data through a camera module. Subsequently, the captured data undergoes pre-processing and feature extraction, where relevant patterns are identified. The extracted features are then employed for activity recognition. If the predicted class aligns with the true activity, the system acknowledges the recognition, sends information to the model, and displays the predicted class. In cases where the prediction is inaccurate, the algorithm iteratively returns to the pre-processing and feature extraction stage until an accurate recognition is achieved, marking the end of the algorithm's execution.

B) Image Pre-processing

1) Image Resizing:

The technique of altering a picture's proportions to a certain size or aspect ratio is known as image resizing. This is frequently done to increase the image's visual clarity or lower the amount of computing burden required to process it. When working with a collection of various- sized photographs, resizing may also be utilized to guarantee uniformity. As an illustration, consider the figures (a), (b), and (c) in Fig. 4. By lowering the pixel count, it reduces the processing burden.

In Figure 4, the characters (a), (b), and (c) stand for walking, chit-chatting, and fall respectively.

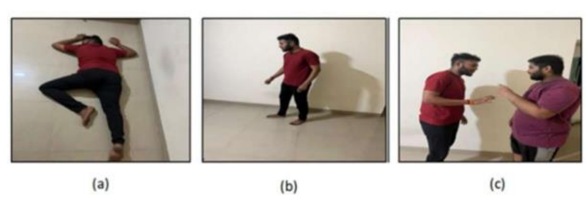


Fig.4 Resized Images

2) Apply Gray scaling to the resized images:

Visuals are used A common technique for converting a color image to grayscale is called greyscaling. To do this, the image's color information is obtained, leaving only the intensity information unchanged.

Single-channel pictures that have grayscale values between 0 and 255—0 representing black and 255 representing white—are referred to as grayscaled images.

Fig.5 Grayscaled Images

In Fig 5, (a) represents Fall, (b) represents Walking, (c) represents Chit Chatting that are grayscaled that are in intensity 255-0.

3) Image Filtering:

Thisprocess employs deep learning algorithms to separate the main subject of an image from its background, as illustrated in Figure 6. The library offers a user-friendly interface for image loading and the application of the background removal algorithm, resulting in the creation of transparent images.

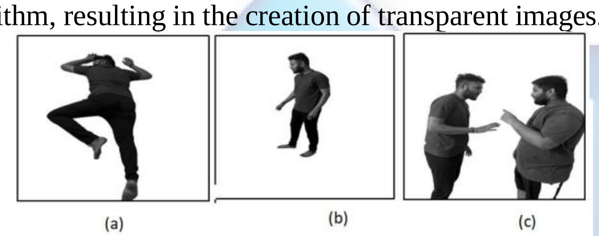


Fig.6 Images after removing background

In Fig 6, (a) represents Fall, (b) represents Walking, (c) represents Chit Chatting.

4) Canny Edge Detection

Prewitt edge detection is a technique for locating edges in a picture by using a certain filter. Gradient magnitudes are created when this filter is applied to each pixel separately to highlight color or intensity changes. As seen in Figure 7, Prewitt edge detection is very good at identifying edges that are oriented either vertically or horizontally.

In Fig 7, (a) represents Fall, (b) represents Walking, (c) represents Chit Chatting.

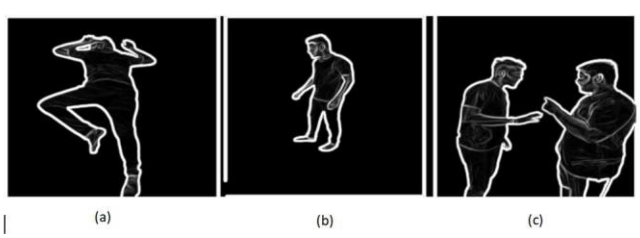


Fig. 7 Images after canny edge detection

C) Feature Description and Dimensions Reduction:

D. G. Lowe developed the Scale Invariance Feature Transform (SIFT) technique in 2004. The DoG operator, which is an approximation of the LoG, is applied by the SIFT detector to locate feature locations as it looks for local maxima at different scales of the input pictures.

The SIFT descriptor generates a total of 128 bin values after extracting a.

16x16 neighborhood and dividing the region into sub-blocks around each recognized feature.

D) K-means Clustering:

K centroids are first chosen at random from the dataset. Each datapoint is then iteratively allocated to the nearest center, and the mean of the points allotted to each cluster is then calculated. Until the centroids no longer fluctuate noticeably or until the allotted number of iterations is achieved.

Furthermore, the method is sensitive to the centroids that are first chosen, meaning that various beginning locations may produce different outcomes. Lastly, it makes the assumption that the clusters are uniformly sized and spherical.

(Note: K is the number of clusters)

Using the Euclidean distance, K-means clustering locates the closest centroid for every data point and assigns it to the appropriate cluster. (Equation 4) denotes the formula for calculating Euclidean distance.

$$\text{Euclidean} = \sum_{i=1}^{n} (ki - mi)^2 \quad \ldots\ldots\ldots (2)$$

Equation 2) "n" is the number of features whereas, "k" and "m" are the 2 vectors.

TABLE. II. ACCURACY OF CLASSIFIER FOR BRISK FOR VALUE K

Classifier	K= 3	K=5	K=7
Random Forest	76.48	78.55	80.01
Decision Tree	71.85	73.93	74.98
K-Nearest Neighbor	75.23	78.01	79.59

Table II presents the accuracy offered by three distinct classifiers for varying K values, resulting from the application of K-Means Clustering on the SIFT feature descriptor.

E) Training and Testing

Three feature descriptors (SIFT) and three classes (Fall, Walking, and Chit- Chatting) are total. The essential characteristics for every class will be extracted by each feature descriptor, which will then store them in a different stack. When using SIFT as an example, the SIFT Feature Descriptor will gather features for class 0 "Fall" and store themin a different stack; similarly, it will gather features for class 1 "Walking" and store it in a different stack; and finally, it will gather features for class 2 "Chit-Chatting" and store it in a different stack. These three distinct stack files are applied to the cluster on it after being appended together. 1890 rows of features are created and saved in a stack file following clustering. Next, the last stack is split into 70-30 proportions.

Since training data helps the model find and understand important patterns, it typically gets used in larger quantities than testing data. This is required so that when fresh, unknown testing data is added to the model, it can predict results appropriately.

After training, the model accepts the patterns from the training set and applies this understanding to forecast the testing set. Three distinct classifiers are utilized to generate three different accuracies in order to measure accuracy, and the resulting results are then compared.

1) Random Forest Algorithm:

Regression and classification are two examples of predictive tasks that use RF, an efficient machine learning technique, to produce predictions based on input variables or features. With many decision trees combined, each trained on a distinct subset of the data, it serves as an ensemble learning method. Because of its flexibility and capacity to manage huge datasets with numerous attributes, this method is frequently employed in a variety of industries. Random Forests can also offer a measure of feature significance and lessen overfitting. At 80.01%, the random forest has the best accuracy.

The ROC curve, a graph that displays the classifier's performance, is displayed in Figure 8.

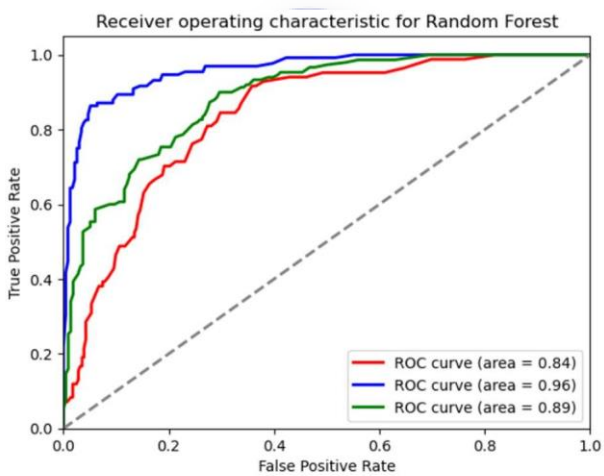


Fig.8 Receiver Operating Characteristics (ROC) graph of true Positive Rate to False Positive rate

2) Decision Tree:

DT is ML classifier and it is a representation of decision-making process and it is a model that take shape of the tree. Decision trees are a popular option in various fields, such as business and medicine, because of their ease of understanding. The Decision tree provides the highest accuracy of 73.93%.

3) K-nearest Neighbor:

KNN is a non-parametric algorithm used in machine learning for classification and regression tasks. Unlike other algorithms, KNN does not make any assumptions about the underlying distribution of the data. The method works by finding the k closest data points in the training set to the point being classified or regressed. The value of k decides the number of neighbors used for classification or regression. The random forest provides the highest accuracy of 75.45%

IV. CLASSIFIER MODEL RESULTS

Table III. is a comparison of 4 different classifier. As the observed Random Forest classifier provided the highest accuracy of 89.41%

TABLE III. Comparison of Classifier Accuracy

SIFT Feature Descriptor				
Classifier	Accuracy	Precision	Recall	F1-Score
Random Forest Classifier	78.48	78.81	78.48	78.61
K Neighbors Classifier	75.45	75.81	75.45	75.55
Decision Tree Classifier	73.93	75.73	73.93	73.78
SVM	65.15	84.87	65.15	69.90

V. LIST OF HARDWARE COMPONENTS USED

COMPONENTS	QUANTITY
ESP 8266 WIFI Module	1 no.
L298N motor driver.	1 no.
Gear Motor	4 no.
Raspberry Pi	1 no.
USB Camera Module	1 no.
12v Battery	1 no.

VI. CIRCUIT DIAGRAM

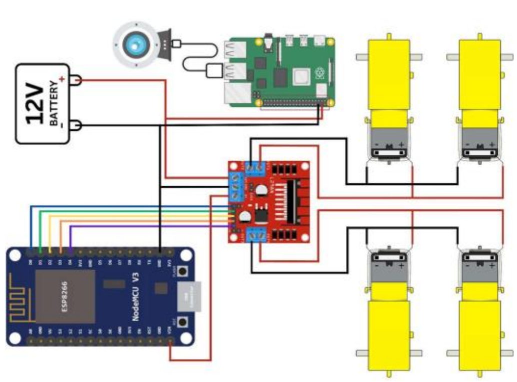


Fig. 11. Circuit Diagram of hardware connection

Actual Implementation:

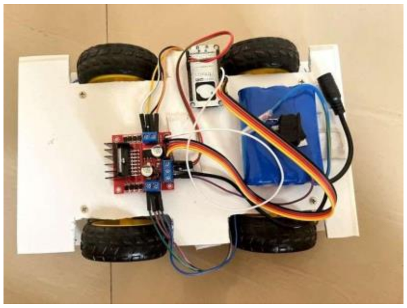
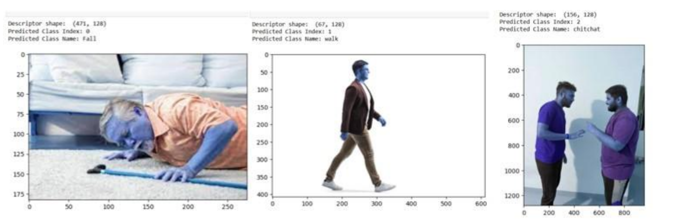


Fig. 10. Actual implementation of hardware connection

VII. MODEL RESULTS

Output Result of the CV Model:



a)Class predicted as Fall b)Class predicted as Walk c)Class predicted as ChitChat

Fig. 11. Class predicted as a) Fallen, b) Walking, c) Chitchatting

From Fig.11, we get the predicted class as Fallen where class code was 0, along with it we got the descriptor class shape along with the Histogram Values. Same is for Fig.12 we get Class predicted as Walking and From Fig.13 we get class predicted is Chitchatting. This shows us that we obtained the assumed output perfectly. From the Classification algorithm used, the highest accuracy we obtained is by using Random Forest Classifier that is 80.01% which gave authors correct prediction.

VIII. RESULT AND OUTPUT

The final results of the project show a significant level of achievements in reaching the initial objectives. The effective use of the computer vision algorithms and machine learning model that have been deployed can be seen by the system's ability to precisely recognize human actions. The accuracy with which traits are recognized—for example, differentiating between falls, small talk, and walking—demonstrates the strength of the technology in use. This feature recognition accuracy adds to the system's overall ability to properly discriminate between different activities. Furthermore, an integrated and functioning system is shown by the effective integration of hardware elements, such as the ESP8266 module, high-definition camera, and motors. The project's outcomes demonstrate a remarkable level of achievement in several important areas. At first glance, the ability of the algorithm to recognize and differentiate between human activities like walking, talking, and falling is proof of the effectiveness of the machine learning model that was put into reality. The system's practical use in surveillance depends on this activity recognition precision, which also increases the system's relevance and dependability.

IX. FUTURE SCOPE

This study's future goals include developing advanced machine learning models for human activity recognition, integrating cutting-edge sensors for improved perception, creating autonomous navigation capabilities, utilizing edge computing for immediate decision-making, expanding for multi-robot coordination, highlighting sustainability and energy efficiency, creating customizable user interfaces, deploying in specific environments, working with emergency services, and utilizing global connectivity for data sharing. These approaches seek to increase the project's influence while guaranteeing its efficacy and flexibility in a variety of surveillance-related applications.

X. References

1. Akilan, T., Satyam Chaudhary, Princi Kumari, and Utkarsh Pandey. "Surveillance robot in hazardous place using IoT technology." In *2020 2nd International conference on advances in computing, communication control and networking (ICACCCN)*, pp. 775-780. IEEE, 2020.
2. Shah, Mohammad Shoeb, and P. B. Borole. "Surveillance and rescue robot using Android smartphone and the Internet." In *2016 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1526-1530. IEEE, 2016.
3. Sirasanagandla, Sushma, Mounisha Pachipulusu, and Ramesh Jayaraman. "Development of surveillance robot to monitor the work performance in hazardous area." In *2020 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1559-1562. IEEE, 2020.
4. Meghana, S., Teja V. Nikhil, Raghuveer Murali, S. Sanjana, R. Vidhya, and Khurram J. Mohammed. "Design and implementation of surveillance robot for outdoor security." In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 1679-1682. IEEE, 2017.
5. Maurya, Amit, Mayuri Sonkusare, Akshata Raut, Dhanashree Tamhane, and Deepali Palase. "Surveillance robot with human detection." In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 107-110. IEEE, 2018.
6. Patoliya, Jignesh, Haard Mehta, and Hitesh Patel. "Arduino controlled war field spy robot using night vision wireless camera and Android application." In *2015 5th Nirma University International Conference on Engineering (NUiCONE)*, pp. 1-5. IEEE, 2015.
7. Singh, Akash, Tanisha Gupta, and Manish Korde. "Bluetooth controlled spy robot." In *2017 International Conference on Information, Communication, Instrumentation and Control (ICICIC)*, pp. 1-4. IEEE, 2017.
8. Telkar, Aishwarya K., and Baswaraj Gadgay. "Iot based smart multi application surveillance robot." In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 931-935. IEEE, 2020