Amphibious Self-Balancing Autonomous Surveillance UGV

Soofiyan Atar[1], Adil Shaikh[2]

[1]Department of Electronics and Telecommunication Engineering, K.J. Somaiya College of Engineering
[2]Department of Mechanical Engineering, K.J. Somaiya College of Engineering

---

Abstract - *This project aims to propose an architecture for an intelligent surveillance system. The aim is to lessen the burden on humans due to conventional surveillance systems by incorporating innovative frameworks, computer vision in co-operation with autonomous unmanned ground vehicle (UGV). We tackle the problem of robot surveillance decisions and how integrating various components in the system supports fully automated decision-making. We present a modern approach for surveillance at remote and border areas, multi-storey buildings, manufacturing facilities and natural resource extraction plants, and other such institutions that encompass a large extent. This robot is also helpful for home security where it could act as a companion of some sort to the individual. Its amphibious nature allows it to traverse on terrains such as sand, mud, snow and water with ease without hindering its functionality. It is a multifunctional robot based on IOT aspects. This UGV can assist a soldier at a border area or a guard at various facilities lessening their burden and providing tight security even at home. This robot has multiple sensors that detect the presence of the enemy, capture it in camera and give the live video and audio to the authorized person. Detect smoke, footsteps, IED's and transmit their location to the control center.*

Key Words: Path planning, chatbot, object detection, object tracking, face recognition

1. INTRODUCTION

With recent research development in computer vision, robot autonomy has the potential to revolutionize surveillance technology. Consider the workforce and concentration spent by security personnel to monitor numerous live video feeds from cameras that are presently surveilling parking lots, university campuses and shopping malls. Imagine the dexterous patrols of security guards through countless corridors. Think over the difficult strategic decisions about where and how to allocate precious human resources, corresponding to immediate security issues and anticipating the future ones to come [1]. Robot mobility has advanced to the level where robots now can navigate complex environments, patrolling as humans would do. Equipped with cameras and other sensors, they can serve as mobile surveillance nodes. For example, a robot can provide coverage to areas that may be critical due to camera failures or other such factors [3][4]. Robots also have the mobility, sensors, and actuators to react directly to a situation detected over a camera feed, thereby leveraging real-time scene analysis. To integrate these technologies effectively and make robots truly autonomous, a third key technology is an intelligent decision-making. Here robots choose their actions to accomplish a combination of objectives given constrained resources. This project resolves the problem of replacing humans with surveillance robots by finding a bridge between the two. We reduce the harm of human resources. The Robot is miniature capable enough to enter tunnels, mines and holes in the building. Also, it can survive in harsh and challenging climatic conditions for a feasible amount of time. To maintain safety and security, conventional surveillance systems rely critically on human attention, action and intelligence. However, such reliance is less productive in a society where the trend is for more cameras, embedded systems, and complex surveillance environments to fend against potential threats (from burglary to natural disasters to terrorist attacks). Its main advantage is that it will go unnoticed during the surveillance and will be giving live data feed seen by the robot to the control centre. We emphasize the need to shift onto autonomous systems to meet present-day surveillance demands and requirements. This UGV does not replace the human resource but rather aids their functioning thereby increasing the efficiency and lessening the burden put on them. This robot will integrate with already available systems present at the institution thus proving very cost effective. Mobility of the bot is feasible for a wide number of traversing regions

Object detection [5] is a vital part of any autonomous surveillance UGV. Object detection gives in-depth information about the environment, used for path planning and security purposes. In this research, we have used a state-of-the-art algorithm, MobileNet-SSDv2 [7], to detect objects and used for face recognition pipeline on an embedded device. Face recognition is an essential security feature; our face recognition technology uses an Intel real sense D435i camera and an RGB-D camera. Our face security algorithm can't be spoofed using 2D images of the person. Path planning algorithm is essential for surveillance robots because it covers the whole space perimeter and continuously avoids dynamic obstacles, parallelly running other algorithms. For mapping the environment, we have used the Robotic Operating System (ROS) [12] framework along with custom scripts. We have used Potential field planning for global planning and Dynamic Window approach (DWA) [13] for local planning, including obstacle avoidance and smooth trajectories for navigation.

This paper focuses on the complete pipeline for autonomous surveillance UGV alongside smaller size and lesser cost for the overall system. The significant contribution of the research is:

- Implementation of self-sustaining navigation pipeline along with mapping the environment.
- Implementation of object detection, object tracking, face recognition and obstacle avoidance algorithms.
- Implementation of the chatbot [26], threat detection, audio and video live streaming and instant notification algorithms.

Section 3 is the implementation part, where the algorithms and hardware integration have been explained in detail. Section 3.1 focuses on Hardware design and integration for the UGV, and section 3.2 describes the physical design of the UGV and explains different properties. From section 3.3, software pipelines are explained, such as section 3.3 is object detection following that section 3.4 is object tracking. Section 3.5 describes the mapping and path planning techniques of the UGV. Then section 3.6 concentrates on how to control the dynamics of the UGV. Section 3.7 explains all the localization techniques. Section 3.8 is all about threat assessment, with subsection 3.8.1 focuses on face recognition algorithm.

2. RELATED WORK

Systems have already been developed to analyze video feed in transportation networks autonomously and public spaces, identify actors and characterize behaviors—for example, IBM's Smart Surveillance System project [2]. There are also approaches for activity interpretation, while other works are focused on low-bandwidth requirements by locally processing surveillance images—a four-wheeled robot for Surveillance in Military Applications. Mobile Robotic System for Surveillance of Indoor Environments robots can be used to interact with the environment, with humans or with other robots for more complex cooperative actions have also been developed.

There are many algorithms for object detection with high precision also with high computation. YOLO, the state-of-the-art algorithm for object detection, also comes with high computation needs, which is not feasible for a mobile robot with low-end embedded hardware. Instead of object detection, instance segmentation gives segmented output with greater detail, but this is too computation greedy; thus, object detection is better for surveillance robots. For face recognition, there are many models. Still, as we opt for higher accuracy, we get higher computation needs as well, thus using haar cascade [16], we can obtain accurate results along with low computation. Chatbot [26] implementation on embedded devices is widespread in many surveillance robots, but our robot also uses multiple language recognition with the same accuracy for voice recognition. For localization, many UGV uses GPS and fusion of IMU, odometer and 2D lidar scans. This method gives high variance in GPS denied areas; thus, our paper focuses on landmark detection for localization and correcting our robot pose. Localization techniques using High Fidelity Maps (HFM) [15] are a very high computational algorithm that is not applicable for low-end embedded devices. HFM in many delivery robots is used with GPS using fusion algorithms, thus increasing accuracy. Still, for semi-indoor environments using landmarks recognition, we achieved almost the same accuracy. We are using the ROS framework for mapping in which the slam toolbox package is used, which gives better results for long are maps. But for navigation, we use the A* algorithm, which computes the global path under 100 ms and every 5 seconds, the global course is updated considering new obstacles observed in the environment. For local path planning, we have used the DWA [13] algorithm. The DWA algorithm uses less computation than (Time elastic band) TEB [14] planner, giving similar results as DWA but TEB is slightly more computational heavy.

The research gap of the project compared to the previous versions:

- Cost reduction in cost by making the design with reference to spherical robots and integrating them with the conventional two-wheeled self-balancing robot model [24].

- It builds on top of a partially existing idea and brings together the best of things to be used in a sector that is different from the conventional one.
- The mechanical design is brought down to a level where it can be reproduced easily with the help of additive manufacturing and is given a robust structure that can absorb and adapt to situations.
- A two-wheeled unconventional robot of original make by taking inspiration from the already existing wheeled robots and try to accommodate as many features as possible if not more as provided by the other robots.

The robot can perform up to its maximum capability despite the changes in environment (such as sand, staircase, mud etc). We try to make the robot suitable for home security where it might act as companion of sort. Many surveillance robots [3][4] have different robots for different purposes; for example, a particular robot is used for patrolling a house, and for patrolling with thermal systems, a separate system is developed. Still, our robot is used in every possible situation with some constraints, but with lower cost and higher efficiency.

3. IMPLEMENTATION

Estimating pose using IMU, 2D lidar and odometer, fusing this pose with the pose obtained from the 3D camera, object detection, object tracking, face recognition, path planning, chatbot, dynamic obstacle avoidance, live video and audio streaming and control algorithm are all part of the implementation.

3.1 Hardware design

Fig -1 shows the hardware design housing all the required sensors. The central processor is the NVIDIA Jetson Nano which has all the peripherals connected to it. The microphone is used for a live audio stream; a smoke sensor is added to detect fire bursts, not in the visual field. GPS module has 1-3 m accuracy; thus, it is used for a global coordinate localizer. 2D lidar sensor is used for mapping, localization and obstacle avoidance. IMU is used to obtain the orientation of the robot. Encoder gives odometer readings. We have also connected a 4G modem for connectivity for live streams and sending alerts. A gyro stabilizer is also added for mechanical stability. For faster locomotion, we are using BLDC motors. The extra camera at the backside of the bot is used for security purposes. Intel Real sense D435i provides depth information which is further passed to facial recognition [10][11], object tracking [18] and obstacle avoidance modules.

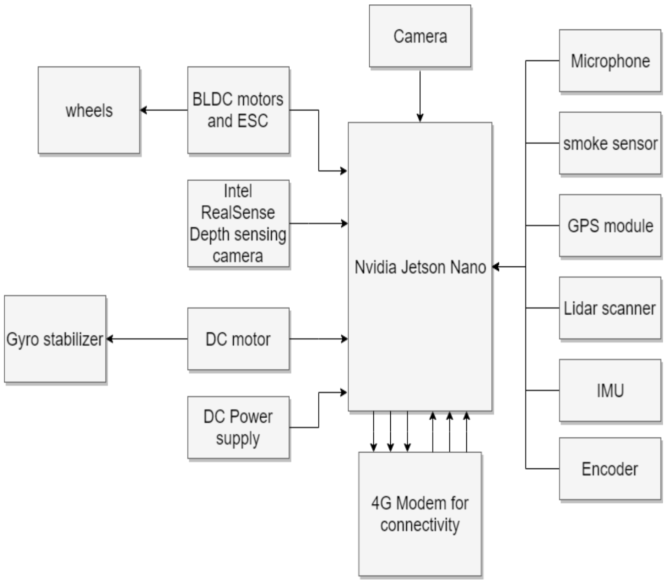

Fig -1: Hardware flow of the UGV

3.2. Design of the UGV

As shown in Fig -2 and Fig -3, the robot consists of a two-wheel self-balancing robot. Due to the unique structure and material of the wheels, the robot can accommodate itself on rough terrain and climb stairs by balancing its centre of gravity. The wheels also absorb any impact due to the robot falling from a height, making it a robust system. The wheel design is such that the robot can traverse on surfaces such as sand, snow and mud. It consists of four BLDC motors that form a combination to give a planetary gearhead reduction.

Fig -2: CAD design of the UGV

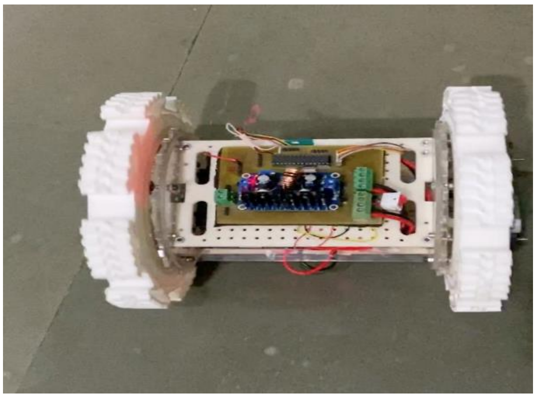

Fig -3: Hardware implementation of UGV

The distinguishing feature is the comparatively small make of the robot that helps it interact with the environment better. Fig -3 shows the naked view of the robot with hardware installed on the UGV. This design is also very cheap to build as compared to other surveillance UGV.

3.3. Object detection

For object detection, we have used the MobileNet-SSDv2 algorithm, which was scripted using python language. As shown in Table -1, MobileNet-SSDv2 [7] gave higher FPS than YOLO algorithms, and there is a very slight decrease in accuracy compared to YOLOv5s [8]. YOLOv3 [9] provides much higher accuracy, but it is challenging to process other algorithms dependent on object detection output with only two fps. The mean average precision (mAP) of MobileNet-SSDv2 with 33.7 was competitive with mAP of YOLOv5s.

Table -1: Comparison of different object detection models

Object detection models	mAP (%)	FPS (Jetson Nano)
YOLOv3	**54.3**	2
YOLOv5s	37.6	6
MobileNet-SSDv2	33.7	**15**

In total, we trained our custom dataset, which consisted of objects like humans, bikes, cars, bicycles, and other familiar things. We used the same model for threat detection, which includes detecting threat objects such as knives, scissors, guns, and many more.

3.4. Object tracking

Object tracking was used to track a suspicious person without knowing them. Our object tracking model followed only a single human/object, and future states were predicted using the Kalman filter. The robot was well equipped with fast motion so that it could cope up with human speed. For object trackers, three well-known trackers are Kernelized Correlation Filters (KCF) [22] tracker, Channel and Spatial Reliability Tracking (CSRT) Tracker [21] and Minimum Output Sum of Squared Error (MOSSE) tracker [23]. We equipped the CSRT tracker because of its speed and low frame skips.

Table -2: Comparison of different object tracking models

Object tracker	Frame skips	FPS (Jetson Nano)
CSRT	**26**	11
KCF	39	12
MOSSE	48	**25**

As shown in Table -2, CSRT has the least skipped frames and almost the same FPS as the KCF tracker [22]. MOSSE tracker [23] is high-speed, but in this tracker, frames are overlooked quite often. Frame skips obtained here are from a one-minute video with 15 fps. Apart from these, there are Intersection of Union (IOU) trackers, but they are only accurate if the motion is confined in a particular activity; it does not track irregular movement. All these trackers also handle occlusion very well with very little difference.

3.5 Mapping and Path planning

For mapping the environment, we used the ROS framework, which gave us a .pgm file as output format and using this, we implemented our path planning and motion planning algorithm using A* and DWA [13]. For A*, we used 1 Hz frequency to update the global path, and DWA was running at 10 Hz to generate smooth trajectories and avoid dynamic obstacles. The global navigation pipeline used Intel real sense camera and merged the 2D point cloud with the 2D lidar scan, thus getting a 2.5D map for the 10m range. Then using the A* algorithm, we obtain the global path; following this, we also have a few conditions as shown in Algorithm -1. Then this path is sent to another node of the local navigation pipeline by publishing this path as a topic in the ROS [12] framework. The alert sent through the robot is through SMS. The path planning algorithm is running parallel with the local navigation pipeline.

Algorithm -1: Global navigation pipeline

1. Mapping the environment using slam toolbox package
2. Get pgm file and initial location of the robot
3. **while** *True* **do**
    a. Convert 3d point cloud from Intel real sense D435i to 2D lidar scan
    b. Merge converted scan with that of 2D lidar scans
    c. Plot the merged scan on the map of the 10m range
    d. **if** *(current time is greater than one sec compared to last time)* **then**
        i. Compute A* algorithm for computing global path
        ii. **If** *(path not obtained)* **then**
            1. Stop for 5 seconds
        iii. **else**
            1. Send the path to local planner
        end
    end
    e. **if** *(robot hit obstacle)* **then**
        i. Stop and send alert
    end
end

Algorithm -2: Local navigation pipeline

1. Get global path from global navigation pipeline
2. **while** *True* **do**
    a. Convert 3d point cloud from Intel Real sense D435i to 2D lidar scan
    b. Merge converted scan with that of 2D lidar scans
    c. Plot the merged scan on the map of the 4m range
    d. **if** *(current time is greater than ten milliseconds compared to last time)* **then**
        i. Compute DWA algorithm for computing local path
        ii. **If** *(path not obtained)* **then**
            1. Stop for 5 seconds
        iii. **else**
            1. Send the path to motion pipeline
        end
    end
    e. **if** *(robot hit obstacle)* **then**
        i. Stop and send alert
    end
end

Algorithm -2 demonstrates motion planning algorithm using DWA planner. The flow of both algorithms is the same; the difference is that the local planner operates within the 4x4 m range, whereas global navigation runs on the full range of 2D lidar and Intel real sense, which is 10m. The output of the local planner is sent to the motors for getting smooth trajectories. Dynamic obstacles are also avoided in the DWA planner because it evaluates all possible paths for 10Hz frequency. DWA planner is an intense computation algorithm and running it at 10HZ becomes less severe because we are operating at a lower frequency. A* algorithm is running at 1HZ because using it at 10Hz will be a waste of resources as the environment around the robot won't change in milliseconds.

3.6. Control system of the robot

For controlling the dynamics of the robot, many algorithms are available such as Proportional–Integral–Derivative (PID) controller, Linear Quadratic Regulator (LQR) [24] controller, Model Predictive Control (MPC) [25] controller and many more. But each controller have its pros and cons, PID is the most accessible algorithm to work on because of the Ziegler Nichols method, but manual tuning is a time-consuming procedure. MPC controller is computed for the receding time window, whereas LQR computes for the entire time window. We have selected the LQR controller for controlling the dynamics of UGV.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & a_{43} & 0 \end{bmatrix} \tag{1}$$

$$B = \begin{bmatrix} 0 & 0 \\ b_1 & b_2 \\ 0 & 0 \\ b_1 & b_2 \end{bmatrix} \tag{2}$$

$$a_{43} = \frac{m_b g l}{m_b l^2 + I_b} \tag{3}$$

$$b_1 = \frac{-1}{r(m_b + 2m_w + 2\frac{I_w}{r^2})} \tag{4}$$

$$b_2 = \frac{-1}{m_b l^2 + I_b} \tag{5}$$

Equation 1 and equation 2 are the A and B matrices of a closed-loop system. Equations 3, 4 and 5 are the value of the above two equations. m_b defines the mass of the robot, g is the gravity l is the height of the robot i.e. from the centre of the mass till the wheels axis. I_b is the inertia of the robot. r is the wheel's radius, m_w is the mass of the wheels and I_w is the wheel's inertia. These are the equation for the LQR controller for balancing the UGV.

3.7. Localization techniques

We have used multiple sensors for localization such as 2D lidar, odometer, IMU, Intel real sense D435i depth camera, and landmark detection. Using 2D lidar, we have implemented an Iterative Closest Point (ICP) [17] algorithm where the algorithm tries to match the previous scan using an iterative method. As shown in equation 6, T' is the final estimate transform, t_i and s_i are the target points and source points, respectively.

$$T' = \mathrm{argmin}_T \frac{1}{N} \Sigma_i^N ||t_i - T_{s_i}|| \qquad (6)$$

Using this transformation obtained through ICP, we have applied Kalman filter to fuse IMU, odometer and ICP transform to get precise localization. Using Intel real sense D435i, we can obtain landmarks rich in feature and thus recognize each place by minimizing the error between its initial location and its current location. For feature and key point extraction, we have used Binary Robust Independent Elementary Features (BRIEF) descriptor and Features from Accelerated Segment Test (FAST) key points. This combination operates with lower computation requirements. The combination of FAST and BRIEF is called as Oriented FAST and Rotated BRIEF.

Table -3: Comparison of different image matcher models

Image matcher	Time (sec)	Matches	Match rate (%)
SIFT	1.6	**166**	**65.4**
SURF	0.6	110	50.8
ORB	**0.23**	158	46.2

As shown in Table -3, ORB is way ahead in time but not that behind in match rate as compared to SIFT [19] and SURF [20]. ORB [6] completes one cycle of image matching within 230 milliseconds, and when run in parallel with other algorithms, this time increases around 500 milliseconds; thus, ORB was selected in this pipeline.

3.8. Real-Time Threat assessment

The pipeline has integrated threat assessment using different algorithms mentioned in the above sections. Using object detection, we have included threat-related objects detection and then sent instant SMS or notification using various providers (fast2sms) by just sending the request to their servers. Also, we send the photo through one drive for an unknown face, and then the control centre can scan and verify. It also has buzzers inbuilt for an emergency condition. For fire detection, we have visual feedback as well as a smoke detector sensor for unaided situations. It can also send real-time audio and video streams to check surroundings with manual control. Also, footsteps detection is implemented to look for humans before their visual appearance.

3.8.1. Face recognition

Before recognition and detection, we are first storing all relevant faces in a library and then using this library, we will try to match the look. For face detection, we are first detecting humans using an object detection pipeline, and then we can extract the face from the bounding box and then match the face using haar cascade [16]. Haar cascade works in real-time by the edge and line detection method. It uses a sliding feature algorithm where it tries to match features using sliders. In our testing, we had used 30 faces with 200+ images in total, which gave us an accuracy (match rate) of 82%. We also added 3D point feature extraction over haar cascade for extra security. The feature was extracted for the face recognized and saved using an ORB image matcher.

3. CONCLUSION

We have introduced a integration of all valuable features for surveillance UGV in a compact robust with autonomous locomotion. Its main advantage is that it will go unnoticed during the surveillance and will be giving live data feed seen by the robot to the control centre. This device does not replace the human resource but rather aids their functioning thereby increasing the efficiency and lessening the burden put on them. This robot will integrate with already available systems present at the institution thus proving very cost effective. Mobility of the bot is feasible for a vast number of traversing regions, obstacle avoidance - automatically detect and avoid obstacles, real-time threat assessment, real-time location tracking, real-time video/audio surveillance, smoke /fire detection, movement/noise detection, 4G modem for connectivity, robust structure that can absorb impact when falling from a height.

We will optimize these algorithms and implement them on more powerful hardware to house more complex algorithms in future work. In addition, we will also implement this pipeline in a more dynamic condition to demonstrate its highly efficient and effective behaviors.

REFERENCES

[1] Nayyar, Anand, et al. "Smart surveillance robot for real-time monitoring and control system in environment and industrial applications." Information Systems Design and Intelligent Applications. Springer, Singapore, 2018. 229-243.

[2] Tian, Ying-li, et al. "IBM smart surveillance system (S3): event based video surveillance system with an open and extensible framework." Machine Vision and Applications 19.5 (2008): 315-327.

[3] Witwicki, Stefan, et al. "Autonomous surveillance robots: A decision-making framework for networked muiltiagent systems." IEEE Robotics & Automation Magazine 24.3 (2017): 52-64.

[4] Zaheer, Muhammad Zaigham, et al. "An Anomaly Detection System via Moving Surveillance Robots With Human Collaboration." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[5] Papageorgiou, Constantine P., Michael Oren, and Tomaso Poggio. "A general framework for object detection." Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271). IEEE, 1998.

[6] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: an efficient alternative to sift or surf. pages 2564–2571.

[7] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[8] Huang, Rachel, Jonathan Pedoeem, and Cuixian Chen. "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers." 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018.

[9] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).

[10] Phillips, P. Jonathon, et al. FERET (face recognition technology) recognition algorithm development and test results. Adelphi, MD: Army Research Laboratory, 1996.

[11] Naseem, Imran, Roberto Togneri, and Mohammed Bennamoun. "Linear regression for face recognition." IEEE transactions on pattern analysis and machine intelligence 32.11 (2010): 2106-2112.

[12] Marin-Plaza, Pablo, et al. "Global and local path planning study in a ROS-based research platform for autonomous vehicles." Journal of Advanced Transportation 2018 (2018).

[13] Henkel, Christian, Alexander Bubeck, and Weiliang Xu. "Energy efficient dynamic window approach for local path planning in mobile service robotics." IFAC-PapersOnLine 49.15 (2016): 32-37.

[14] Quinlan, Sean, and Oussama Khatib. "Elastic bands: Connecting path planning and control." [1993] Proceedings IEEE International Conference on Robotics and Automation. IEEE, 1993.

[15] Mustafa, Mustafa, et al. "CosmoGAN: creating high-fidelity weak lensing convergence maps using Generative Adversarial Networks." Computational Astrophysics and Cosmology 6.1 (2019): 1.

[16] Soo, Sander. "Object detection using Haar-cascade Classifier." Institute of Computer Science, University of Tartu 2.3 (2014): 1-12.

[17] Rusinkiewicz, Szymon, and Marc Levoy. "Efficient variants of the ICP algorithm." Proceedings third international conference on 3-D digital imaging and modeling. IEEE, 2001.

[18] Yilmaz, Alper, Omar Javed, and Mubarak Shah. "Object tracking: A survey." Acm computing surveys (CSUR) 38.4 (2006): 13-es.

[19] Ng, Pauline C., and Steven Henikoff. "SIFT: Predicting amino acid changes that affect protein function." Nucleic acids research 31.13 (2003): 3812-3814.

[20] Bay, Herbert, et al. "Speeded-up robust features (SURF)." Computer vision and image understanding 110.3 (2008): 346-359.

[21] Farhodov, Xurshedjon, et al. "Faster RCNN detection based OpenCV CSRT tracker using drone data." 2019 International Conference on Information Science and Communications Technologies (ICISCT). IEEE, 2019.

[22] Li, Yang, and Jianke Zhu. "A scale adaptive kernel correlation filter tracker with feature integration." European conference on computer vision. Springer, Cham, 2014.

[23] Bolme, David S., et al. "Visual object tracking using adaptive correlation filters." 2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, 2010.

[24] Sun, Liang, and Jiafei Gan. "Researching of two-wheeled self-balancing robot base on LQR combined with PID." 2010 2nd International Workshop on Intelligent Systems and Applications. IEEE, 2010.

[25] Afram, Abdul, and Farrokh Janabi-Sharifi. "Theory and applications of HVAC control systems–A review of model predictive control (MPC)." Building and Environment 72 (2014): 343-355.

[26] Ranoliya, Bhavika R., Nidhi Raghuwanshi, and Sanjay Singh. "Chatbot for university related FAQs." 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2017.

[27] Welch, Greg, and Gary Bishop. "An introduction to the Kalman filter." (1995): 127-132.