IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

Autonomous Quadcopter for Surveillance

Amith Roy[1], Abhishek Rasal[2], Viraj Gavhane[3], Ravi Javali[4]
Take-off Edu Group, Bangalore, India[1,2,3,4]

Abstract: *The use of drone technology is becoming increasingly popular. Finding applications, across various industries. In agriculture drones are employed for tasks such as applying pesticides and fertilizers monitoring crops and creating maps. The construction sector utilizes drones to inspect job sites and track progress. Additionally, the film industry benefits from photography using drones while search and rescue operations rely on them to reach areas. Overall drone technology has the potential to revolutionize industries by offering solutions to previously challenging problems. We propose the development of an AI based autonomous drone system that can be deployed in areas for applications including surveillance and warehouse management. This system incorporates a PI CAM V2 camera mounted on a quadcopter equipped with GPS technology. The drone operates autonomously without the need for a pilot thanks to software, like Robot Operating System (ROS) other cutting-edge technologies.*

Keywords: Multipurpose, Robot Operating System, Simulation, Surveillance, Tracking, Quadcopter, OpenCV, Machine Learning, Object Detection

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) commonly known as drones have proven to be highly versatile, over the decade. They have been utilized in applications such as monitoring, data collection and surveillance. One of their advantages is their ability to access challenging areas that may be difficult for humans to reach. As a result, UAVs are considered assets. To function autonomously and assist humans in tasks robotics in general including UAVs rely on a combination of the robot operating system, intelligence (AI) and machine learning algorithms that enable perception and interaction with the real world. This paper presents our work on designing and developing an AI ML based drone of serving purposes, like surveillance and warehouse management.

II. LITERATURE SURVEY

A. Study of Existing System

Manual drones, often known as remote-controlled drones, are drone systems that must be commanded manually by pilots to perform any function. Human operators pilot these drones by guiding them through the air with a remote control. Manually piloted drones are commonly used for hobby flying or leisure activities such as aerial photography, among others. Furthermore, they are used for business applications such as agricultural monitoring, land surveying, and infrastructure inspection. Manual drones offer the benefit of providing fine control over the drone's movement as well as the ability to function in settings where autonomous drones cannot. Nonetheless, manual drones may be less successful than autonomous drones for some jobs and require a professional operator.

B. Study of Existing System

The biggest disadvantage of typical manual control drone systems is that they require continual supervision to do any operation. Furthermore, the drone pilot must have all the necessary abilities and expertise in drone flying. However, human operated drones are more accurate in completing vital tasks such as rescue operations. However, for tasks that are repetitive in nature, such as warehouse management, where products are monitored for warehouse management, or surveillance, where a specific area is monitored repeatedly, an autonomous drone system can be implemented that can perform a repetitive task more efficiently and precisely in less time and with fewer human resources.

DOI: 10.48175/IJARSCT-15406

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

III. METHODOLGY

Today Drones may be used for a variety of purposes, including surveillance, but the fundamental disadvantage is that they require human control by a pilot or operator. Single board computers, such as the Raspberry Pi and Jetson Nano, can run robot operating systems, artificial intelligence, and machine learning, allowing drones to do tasks such as decision making, flight path identification, obstacle avoidance, image processing, and object detection. The system will include a drone, a control centre, a data collection centre, and a data logging server.

A. Objectives

- Use of technologies such as obstacle avoidance, flight path detection, intelligent mission following, machine learning, ROS (Robot Operating System) to provide drones ability to fly autonomously with less interaction of drone pilots.
- Least amount of human/pilot interaction for controlling and commanding the drone system.
- Developing multipurpose autonomous drone system that can be deployed in many areas for multiple applications.
- Making the repetitive tasks more time efficient and precise with the help of autonomous mission-oriented drones.

B. System Overview

The system is made up of a core UAV platform based on the Ardupilot and a raspberry-pi SBC that serves as a companion computer with PYMAVLINK. It also contains a drone kit API with an 8 MP camera and SPI interface for real-time object identification using a MJPG stream. To achieve autonomous capability in the drone, the system is supported by technologies such as the Robot Operating System and machine learning. A strong ground station for calculation and command, along with a high-speed WLAN network. The following are some of the tools and technologies utilized to create the proposed system.

- Open CV: - A well-known open-source computer vision package called Open CV offers a variety of processing options for images and videos, including object detection, tracking, and identification. Robotics, autonomous systems, and machine learning are just a few of the fields where Open CV is frequently utilized. Open CV can be utilized to give autonomous drone systems vision-based perception capabilities.
- Robot Operating System: - An open-source framework called Robot Operating System (ROS) is used to create robot software. It offers a set of protocols, libraries, and tools to assist programmers in building powerful, sophisticated robot applications. Autonomous vehicles, drones, industrial automation, and healthcare are just a few of the robotics industries where ROS is extensively employed.
- Gazebo Simulator: - One of the most popular simulation environments in the robotics world is Gazebo, a physics-based robot simulator. It offers a platform for testing and evaluating robotics algorithms in a virtual environment before implementing them on actual robots. It is integrated with the Robot Operating System (ROS). Compared to physically building and testing robots, simulation in Gazebo can be a major financial savings. In Gazebo tests are conducted in a simulated setting are safer since there is little chance that real robots or the environment would be harmed
- MAV Link Protocol: - For unmanned aerial vehicles (UAVs) and other robotic systems, MAV Link (Micro Air Vehicle Communication Protocol) is a simple, open- source communication protocol. A common method of communication between various drone system parts, such as the flight controller, ground station, and other onboard electronics, is provided by MAV Link.

MAV Link is intended to work with a variety of hardware and software and is platform-independent. Many well-known open-source drone software platforms, including ArduPilot and PX4, support MAV Link, which is widely used in drone systems.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

C. System Building Process

1. Drone Building Process: - Installed 2600KV motors with 920KV 2212 ready to sky motors and Mamba 405 with Pixhawk 2.4.8. Configured it according to the needs with PID tunings. Installed 1000mAh 2S 20C Orange Li- PO with 3000mAh 3s 30C orange Li-PO and Jetson Nano with Raspberry-pi and cooling-fan. Installed Blue- tooth with ESP8266. Added an extra 18650 battery for Raspberry-PI. Installed OLED-Screen. Done PID tunings and other optimizations for stable flight.
   Result: High flight time, automated flight capabilities, very high flight stability, LOITER mode support.
2. Flight Controller and Telemetry: - Installed Mamba F405 MK2 mini flight controller with Pixhawk 2.4.8 flight controller.
   Result: Open source Ardupilot flight controller with lot more feature than F405 ex. Open-source and fully configurable, MAV Link support and Telemetry 2 support. Installed and configured Node MCU (ESP8266).
   Result: Very High-Speed telemetry and drone Wi-Fi.
3. Video Streaming Process: - To increase the data transfer rates and reduce the latency switched from 2.4 GHz to 5 GHz Wi-Fi. Created a python code for video streaming server used default IP address of the raspberry-pi as video streaming. To optimize the stream used Open CV API to change the resolution, framerate, bitrate, Auto- exposure, orientation, Buffer-size, Autofocus, zoom, camera roll and other required features of the raspberry- Pi camera.
   Result: very low latency 720p 60FPS well optimized video streaming server without lag and stuttering.
4. Object Detection Process: - As the Raspberry-Pi has very low computational power decided to use laptop as a processing station for object detection. Installed second windows-10 OS as a final deployment OS. Installed all the dependencies like CMake, latest Nvidia Drivers, etc. Installed pip3, NumPy, putty, Open CV with Open CV- contribute CUDA CUDNN world, CMake, Anaconda, Visual Studio, Visual Studio Code, Latest NVidia drivers and other required dependencies. Optimized OpenCV python code for the use of CUDA and DNN drivers.
   Result: 60 FPS object detection with CUDA acceleration with latest Nvidia Drivers.

Fig 1. Drone Prototype

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

D. System Architecture Diagram

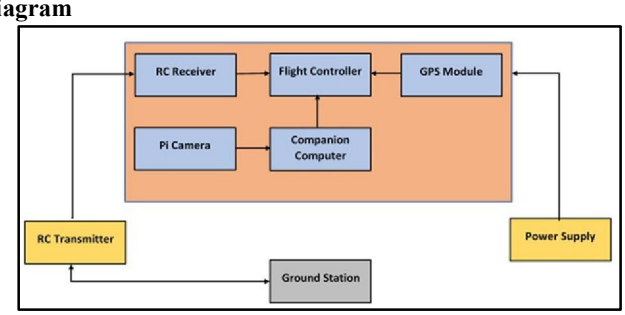

Fig 2. System Architecture Diagram

IV. IMPLEMENTATION RESULTS

A. Automated Drone Arming

A drone must be "armed" before it can take off by turning on its motors. However, it is crucial to confirm that all safety measures have been taken before arming the drone to avoid any mishaps or drone damage. The general procedures for arming a drone are as follows:

Prior to arming the drone, it is crucial to inspect the flight environment for any potential risks, such as power wires or trees, and to make sure that no people or animals are present.

Before arming the drone, conduct a pre-flight check to make sure it is in good condition, the battery is fully charged, and all sensors are operating as they should.

Turn on the remote controller: Make sure the drone is linked to the remote controller and turn it on.

Arm the drone: The technique for arming the drone can vary depending on the type of drone and flight controller. In most cases, arming the drone requires pressing a few switches or buttons on the remote controller, such as pushing both sticks to the bottom-right corner.

Check to see if the drone is armed: When the drone is armed, the motors will begin to spin. Check the LED lights or listen to the motors to make sure the drone is properly armed.

Launch: After being armed, the drone is prepared for take-off. Turn up the throttle gradually to raise the drone off the ground.

The above-mentioned arming process is automated by us with the help of python programming language. We created one python script that automates the whole arming process in very efficient and easier way.

B. Object Detection Process

For object detection, we utilized OpenCV, a machine learning-based Python package. The Caffe (Convolutional Architecture for Fast Feature Embedding) model, a deep learning framework designed specifically for image classification and segmentation, was utilized to accomplish the object detection problem, i.e. human face detection. Human face detection will be useful for jobs like monitoring, among others. This module was created in the early stage of project development to demonstrate one of the drone system's uses. The photos below demonstrate the real operation of this module.

Fig 3. Object Detection using Open CV

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

C. Simple Obstacle Avoidance with Range Finder

To safely navigate hard settings, autonomous drones must be able to avoid obstacles, making this a key ability. In this module, we employed a Lidar range finder sensor, which uses lasers to determine distances between objects and is a standard obstacle avoidance sensor. This sensor is situated on the drone's front side and detects objects at a 40-degree angle before determining the distance between them and the drone. Depending on the distance, the drone will execute more actions. Maze Following using Coordinate System in ROS.

D. Maze Following with Coordinates System in ROS

In this module drone maze navigation accomplished using the ROS coordinate system for maze following it is simulated using Gazebo. The following general procedures describe how to perform maze following for a drone in ROS using a coordinate system:

Create a map of the maze by using a mapping application, such gmapping or Cartographer. To navigate the maze, a route will be planned using this map.

Create a coordinate system: Create a system of co-ordinates that converts the drone's location in the maze into a set of (x, y, z) coordinates. The data from the drone's onboard sensors, including a GPS or a LiDAR range finder, as well as the data from the odometry and odometry sensors can be used for this.

Create a path: Using the maze's map and the specified coordinate system, create a path from the drone's current location to the destination. An algorithm for path planning, such as A* or Dijkstra's algorithm, can be used to do this.

Control the drone's movement by following the predetermined route through the maze. A control algorithm, like a state machine or a PID controller, can be used to do this.

Continually track the drone's location using odometry data and information from its onboard sensors, and up- date the drone's position in the coordinate system as it navigates the maze.

Obstacle detection: To identify obstacles in the drone's path, use sensors like a camera or a LiDAR range finder. Then, change the planned course as necessary.

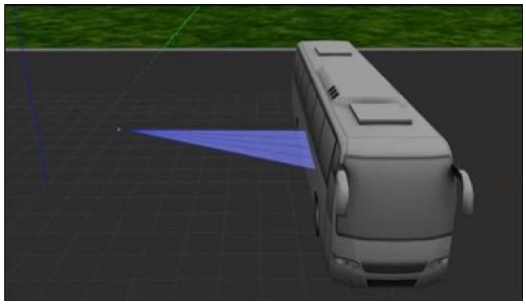

Fig 4. Simple Obstacle Avoidance with Range Finder Simulation.

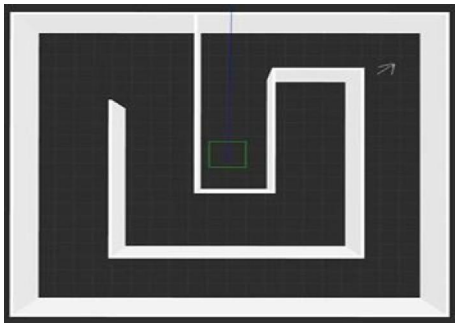

Fig 5. Maze Path Following

E. Obstacle Detection & Path Finder

An autonomous drone needs to have the ability to identify obstacles and determine the route. To achieve this a Lidar (Light Detection and Ranging) sensor can be used for both obstacle detection and path finding. By emitting laser pulses and measuring the time it takes for them to reflect the Lidar sensor creates a map of the drones' surroundings. This data is then utilized to plot a path that avoids obstacles and guides the drone to its desired destination. Unlike drone's autonomous drones equipped with a Lidar sensor possess enhanced precision and safety when navigating complex situations. For example, they can navigate through areas, with trees, buildings or electrical lines that would pose challenges or even be impossible for controlled drones.

In this module we incorporated a Lidar sensor, into a drone's setup. This specific Lidar sensor could detect objects within a 40-degree range. When faced with obstacles surrounding the drone our approach involves rotating the drone 360 degrees while simultaneously using the lidar sensor to identify obstacles. Based on this information we can determine a path in which we can safely direct the drone towards its intended direction. To validate this method, we conducted tests using Gazebo simulator as outlined below.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

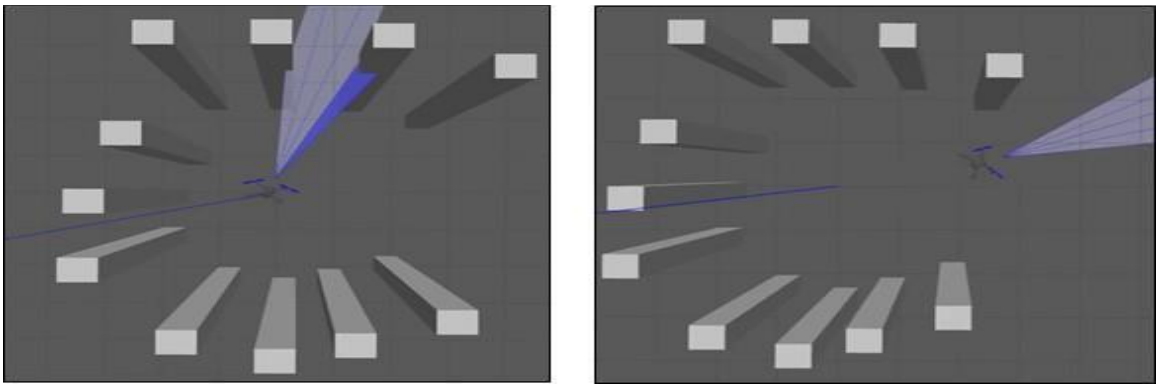

Fig 6. Obstacle Avoidance and Path Finding Simulation

F. Precision Landing with Aruco Marker & Open CV

For an autonomous drone system, it is important that drone should land precisely. In this module we have made us of Open CV and Aruco marker to land drone precisely. Here Aruco maker is like QR code but the difference between them is that Aruco marker takes less bits as compared to QR code to form Aruco marker code, because of this it has become easier to detect Aruco marker as compared to QR code. We have made use of Open CV prebuilt libraries for Aruco marker detection. Following steps need to be taken to perform precision landing using Aruco marker.

Attach an Aruco marker to the landing pad: The landing pad's centre should be marked with an Aruco marker, a sort of fiducial marker used in computer vision applications. The marking needs to be big enough to be seen from afar.

Identify the marker: To identify the Aruco marker in the visual stream, use Open CV and the drone's onboard camera. The Aruco module of Open CV can be used for this because it has capabilities for both detecting and decoding Aruco markers.

After the marker has been located, use the Aruco module to determine the marker's pose, or its position, distance, and orientation in relation to the camera. The position of the drone with respect to the landing pad can be calculated using this information.

Implement safety precautions: Take precautions to avoid collisions or other mishaps while landing. To detect obstructions in the environment, additional sensors, such as a LiDAR sensor or an ultrasonic sensor, may be used, such as creating a safety perimeter around the landing pad.

In this way precision landing can be done using Lidar sensor. Following are some snapshots of Precision Landing with Aruco Marker Open CV performed and tested using ROS based Gazebo Simulator:

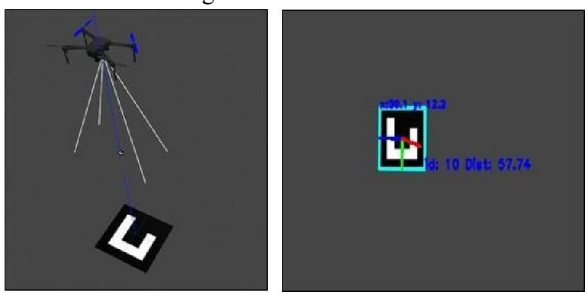

Fig 7. Detecting Aruco Marker and Distance Between Marker and Drone

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.53
Volume 4, Issue 2, February 2024

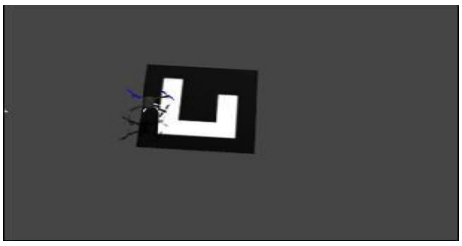

Fig 8. Precision Landing with Aruco Marker and Open CV Simulation

V. CONCLUSION

In this study article, we conclude that this is a highly robust, dependable, and diverse autonomous drone system that can be implemented in various regions for a variety of purposes, reducing human labor and increasing work productivity in an efficient manner.

ACKNOWLEDGMENT

We are grateful for our supervisor's constant advise and support during this project. Their constructive criticism and insightful suggestions assisted us in refining our research and achieving our objectives. We also appreciate the ongoing help and direction we received from everyone at the Take-off Project and the team members throughout our study. We would also want to convey our heartfelt appreciation to all of the scientists and researchers who provided us with access to their essential resources and allowed us to finish our assignment. Their contributions had a significant impact on our study, allowing us to explore new avenues in this field.

REFERENCES

[1].OpenCV: Cascade Classifier.
[2].Chandan G, Ayush Jain, Harsh Jain, Mohana "Real Time Object Detection and Tracking Using Deep Learning and OpenCV" Proceedings of the International Conference on Inventive Research in Computing Applications (ICIRCA 2018).
[3].Aakash Sehrawat, Tanupriya Choudhury, Gaurav Raj "Surveillance Drone for Disaster Management and Military Security" International Conference on Computing, Communication and Automation (IC- CCA2017).
[4].Reem Alshanbari, Sherjeel Khan, Nazek El-Atab "AI Powered Un- manned Aerial Vehicle for Payload Transport Application".
[5].Leo Pauly, Deepa Sankar, "A Novel Online Product Recommendation System Based on Face Recognition and Emotion Detection", (ICCICCT, 2015).