

Development of a Vision-Based Autonomous Drone for Intelligent Object Detection and Tracking

Dr. B. Amarendra Reddy¹, Ch. Sruthi Varma², M. Abhishikth³, D. Hima Vamsi⁴

¹Associate Professor, Department of Electrical Engineering, Andhra University, Andhra Pradesh, India

²Research Scholar, Department of Electrical Engineering, Andhra University, Andhra Pradesh, India

^{3,4}MTech Student, Department of Electrical Engineering, Andhra University, Andhra Pradesh India

Abstract- With the increased adoption of UAVs in intelligent applications, onboard systems that can perform real-time decision-making are in great demand. This paper details the development of an intelligent autonomous drone based on a dual-controller architecture consisting of the PIXHAWK flight controller and the NVIDIA Jetson Nano companion computer. The drone is designed to detect and continuously track the designated object without the aid of a manual control method, such as that using a receiver. The operations of the camera, which broadcasts images to the Jetson Nano, are based on the deep learning and computer vision (with the use of YOLO and Open CV technology) algorithms responsible for processing each image with the aim of determining the location of the object. After the object is identified, the position is estimated, the flight path is calculated, and the instructions for the flight controller are sent by the Jetson Nano, using the MAV Link protocol and UART connection. With the help of this arrangement, the Jetson Nano is involved only in complicated processes of navigation and identification.

KEYWORDS -Autonomous UAV, object detection, YOLO, object tracking, Jetson Nano, PIXHAWK, MAV Link, edge computing.

1. INTRODUCTION

Drones have advanced significantly from simple remote piloted vehicles to machines capable of sensing their surroundings and taking automated actions. A critical element of this evolution is the inclusion of onboard visual perception; with the help of a camera and modern deep learning, a drone is now capable of identifying and pursuing things of interest on its own. However, the embedded flight controllers that stabilize a UAV are optimized for low-latency control rather than for the heavy computation that convolutional neural networks require, so running a detector such as YOLO directly on a flight controller is impractical. A widely adopted solution is the companion-computer paradigm, in which a small but capable processor is mounted alongside the flight controller and dedicated to perception and high-level decision-making, while the flight controller retains responsibility for attitude stabilization and motor control [1], [2].

This paper presents the design of an AI-enabled autonomous drone built on this paradigm. The proposed platform pairs

an NVIDIA Jetson Nano companion computer with a PIXHAWK flight controller. A camera streams live video to the Jetson Nano, where a YOLO-based detector and a visual tracker identify and continuously follow a selected target. The Jetson Nano estimates the target's position within the image, derives the corresponding flight corrections, and issues them to the PIXHAWK using the MAV Link protocol over a UART serial link. Since the target has been pursued on its own by the camera's capabilities, the system doesn't rely on a person giving commands via a remote control. Recent surveys verify that this type of vision-dependent autonomy is considered the favored line of research work regarding UAVs, with YOLO detectors being named as the most frequently used sensor type in the literature. Since the target has been pursued on its own by the camera's capabilities, the system doesn't rely on a person giving commands via a remote control. Recent surveys verify that this type of vision-dependent autonomy is considered the favored line of research work regarding UAVs, with YOLO detectors being named as the most frequently used sensor type in the literature [3].

This work makes three major contributions. First, it represents a dual-controller hardware-software architecture that cleanly separates real-time flight control from computationally intensive perception. Second, an end-to-end perception-to-control pipeline is presented that integrates object detection, visual tracking, and picture-based flight guidance. Third, an evaluation methodology is defined consisting of detection, tracking, and closed-loop metrics along with a benchmarking protocol based on previously reported performance of the embedded hardware, so that the design can be experimentally tested. Applications of the design can be found in areas of surveillance, inspection of infrastructure, search and rescue operations, and environmental monitoring. This work makes three major contributions. First, it represents a dual-controller hardware-software architecture that cleanly separates real-time flight control from computationally intensive perception. Second, an end-to-end perception-to-control pipeline is presented that integrates object detection, visual tracking, and picture-based flight guidance. Third, an evaluation methodology is defined consisting of detection, tracking, and closed-loop metrics along with a benchmarking protocol based on previously reported performance of the embedded hardware, so that the design can be experimentally tested. Applications of the design can be

found in areas of surveillance, inspection of infrastructure, search and rescue operations, and environmental monitoring.

2 PIXHAWK FLIGHT CONTROLLER

The PIXHAWK flight management unit is a part of modern open-source UAV research. It uses a fast 32-bit ARM Cortex chip that can process data in time. Inside the PIXHAWK flight management unit are industrial-grade IMUs, magnetometers and barometers that work together to give very accurate state information that is essential for complex control systems. The PIXHAWK flight management unit does more than just keep the aircraft stable at level; its design also allows higher-level functions to grow.



Fig 1 PIXHAWK flight controller

By using the MAV Link protocol, the PIXHAWK flight management unit talks easily to companion computers splitting heavy calculations from the main job. This design lets the PIXHAWK flight management unit keep the motors ESC signals and telemetry with ground stations such, as Mission Planner running all the time while the external computers do work like AI surveillance, real-time vision and autonomous tracking all without needing a pilot to hand-control.

3 NVIDIA JETSON NANO

The NVIDIA Jetson Nano works as an edge-computing accelerator for devices that have limited resources. It contains a 128-core Maxwell graphics processing unit, an ARM Cortex-A57 processor and 4 GB of LPDDR4 memory delivering about 472 GFLOPS of compute power. This parallel design is made for deep learning inference letting the Jetson Nano run networks object detection and path-planning algorithms in real time right on the device. For aerial vehicles the Jetson Nano stays in a very efficient 5-to-10-watt power range so it does not drain the

battery or add too much weight. When used as a companion computer the Jetson Nano takes machine-learning work off the main flight controller.

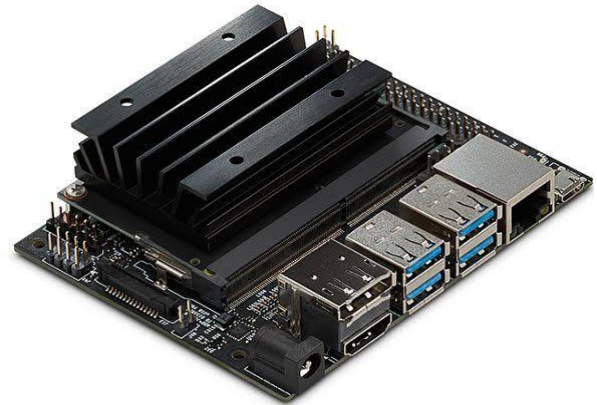


FIG. 2 NVIDIA JETSON NANO

By using libraries such as Tensor RT, with Py Torch or Tensor Flow the Jetson Nano creates a low-latency pipeline that turns live camera data into instant tracking commands.

4 SONY IMX 219 SENSOR

For the vision-based data acquisition module I chose to use the Sony IMX219, which's an 8.08-megapixel back-illuminated CMOS sensor. The Sony IMX219 uses a Type 1/4.0-inch format and features a 1.12-micron square pixel matrix, as well as Ex or R technology that modifies the shape of the photodiode in order to enable it to collect more light. This Sony IMX219 design cuts current and visual blur while keeping high light sensitivity in changing light which is essential for a dynamic outdoor UAV.



FIG. 3 Sony IMX 219 Camera

The Sony IMX219 sends telemetry straight to the companion computer through a fast MIPI CSI-2 interface

avoiding the slow USB route by using up to four lanes of serial data. With this Sony IMX219 link It can capture full-resolution rolling-shutter frames of 3280×2464 pixels at 30 frames per second. It can also get lower-delay 1080p frames at 60 frames, per second. The Sony IMX219 runs on three power supplies analog 2.8 V, digital 1.2 V and interface 1.8 V so it stays power while still delivering a clear optical data stream. This Sony IMX219 lets the system detect, lock onto and follow a target that it picks all without using a transmitter.

5 PIXHAWK AND NVIDIA JETSON NANO CONNECTIONS

To create a two-way connection between the digital world and the physical world the PIXHAWK 2.4.8 flight controller is connected to the NVIDIA Jetson Nano companion computer using the special TELEM2 telemetry port. This connection works through a type of communication called serial (UART) and uses the MAV Link protocol. The speed of this connection is set to a rate like 921,600 or 57,600 band. The wires from the PIXHAWK's TELEM2 output, input and ground are connected directly to the Jetson Nano's J41 expansion header.

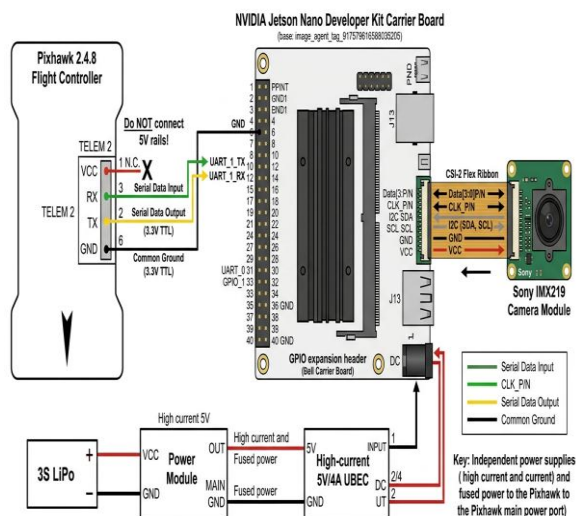


FIG. 4 connection diagram of PIXHAWK flight controller and NVIDIA JETSON NANO with camera

These wires are matched to the level of electricity, so everything works properly. At the time the camera system is set up by connecting the Sony IMX219 camera to the Jetson Nano's built-in MIPI CSI-2 port. A special flat cable is used to make sure the camera sends pictures. This system makes sure the camera and the flight controller are separate but working together. The camera sends pictures directly to the Jetson Nanos memory and the computer vision software on the Jetson Nano sends instructions back, to the PIXHAWK. This allows the drone to fly on its own without needing a control.

6 PERSON DETECTION

To make real-time awareness of where things are the self-driving system needs a strong way to find things and get details from the edge. The helper computer uses a kind of computer network like a small YOLO or Mobile Net-SSD that is made to work quickly on devices. This is done using tools that make the computer work faster. As the high-quality video comes into the computer vision part the system checks each picture to find people and makes a moving box around them. This helps find if there is a person and also gets the size and center of the box. By using a way to process pictures the system keeps tracking the person even when things are moving fast or the background is hard to see. After finding the person the system needs to change the 2D picture data into 3D space using ways to guess how far away things are. The drone does not have two cameras or a special sensor for depth so it uses the idea of how a camera works. The system uses the height of a person and how big that person looks in the picture. By adding in the cameras numbers like how far the lens is the system finds out how far the person is from the drone. This math helps find the distance without using too much power. Finally, this important information about where the person's other details needs to be sent to the base station so the person in charge can see what is happening without having to do anything. The data like distance how sure the system is and the exact place is put into parts of the communication system. The helper computer sends this data through a connection to the flying part. The flying part adds this data with important info like where the drone is and sends everything through a radio to the base. This way while the drone is tracking someone the person on the ground gets up to date information, about how the person is and how the drone is doing.

7 PERSON TRACKING

To achieve autonomy in person identification and tracking the system uses a closed-loop visual serving framework. This framework connects the computing of the NVIDIA Jetson Nano with the stable flight control of the PIXHAWK 2.4.8. The Jetson Nano receives high-definition video through the MIPI CSI-2 interface. It runs a hardware-accelerated neural network. This network is optimized using Tensor RT. Is based on the YOLO (You Only Look Once) model. It detects figures in real time. It draws two-dimensional bounding boxes around each detected person.

The vision algorithm then computes how far the center of the bounding box is from the center of the camera's view. It translates these pixel-level differences into real-world movement needs. A PID (Proportional-Integral-Derivative) controller takes these spatial errors. It calculates the needed velocity adjustments and the required yaw rotation. These commands are sent as MAV Link SET_POSITION_TARGET_LOCAL_NED messages. They are

sent quickly through the UART TELEM2 port to the flight controller.

The PIXHAWK operates in off board or guided mode. It uses the velocity commands directly. It adjusts the multirotor pitch, roll and throttle as needed. The drone follows the person smoothly. It keeps the target perfectly centered in the camera's field of view. This all happens without any help, from a ground station. The system works continuously and autonomously.

8 APPLICATIONS

Deploying edge-computed visual serving frameworks on UAVs opens large-scale abilities, especially in harsh environments where cloud connection and steady pilot communication are weak. By cutting the need for data transfer this autonomous design fits many important uses:

8.1 Search and Rescue: Quickly finding missing people in difficult or far-off areas. The edge processor alone spots shapes and sends simple telemetry just the distance seen and the coordinates to ground stations. This shortens the period of time needed for rescue teams to come to the individual.

8.2 Tactical Perimeter Surveillance: self-protecting sites. When the UAV detects a human being, it automatically transfers from normal patrol mode to live tracking mode. The UAV maintains the necessary distance while transmitting real-time coordinates to the central security office keeping personnel safe and secure on the ground.

8.3 First responder coordination: facilitating acquisition of knowledge during emergencies, such as when there are fires or accidents in plants. The UAV can simply target. Keep track of rescuers without help from people, and furnish the control headquarters with always changing and reliable aerial information.

9 CHALLENGES AND LIMITATIONS

There are various limitations to having an autonomous vision tracker of this nature. The primary limitation concerns onboard computing because, from previous reports, we understand that the processing capabilities of Jetson Nano are limited, thus leading to limited detection speeds, affecting perception frame rates and response times [4], [5]. Besides this, limitations in terms of integer tensor cores further limit the benefits possible through quantization [6]. Another limitation arises from the need for inference and communication latency to work within a specific budget for closed-loop control to be stable and effective. Perception could be affected by certain parameters such as illumination, motion blur or partial occlusion that may lead to lost detections. Correlation filter trackers may be ineffective in cases of large appearance changes, deformation

or rotation [8]. There is a common challenge in finding small objects in aerial imaging due to distance. Another challenge comes from power and heat constraints that limit the computing power in the device. Lastly, safety and security should be taken into account because the basic protocol of MAV Link does not have any built-in encryption, hence, message signing or using a secure link should be kept in mind for implementations in situations where the command link can be compromised [7].

10 CONCLUSION AND FUTURE WORK

The paper discusses how a vision-enabled autonomous drone was built to identify and track objects automatically. The main focus of the project was the dual-controller approach where NVIDIA Jetson Nano was paired with a PIXHAWK flight controller and connected via the MAV Link protocol using a UART serial protocol. The Jetson Nano has the ability to detect objects using YOLO algorithm and offers tracking functionality by visual means.

Meanwhile, the PIXHAWK makes sure the drone remains stable and allows it to control the engines. An evaluation methodology and the whole benchmark process have been developed so it can be used in finding the experimental validation for the design made. The future work is going to focus on the implementation of the evaluation procedure that is going to include the physical testing of hardware and software before commencing with flying tests. Future directions will involve switching to the new edge computing devices (for example, Jetson Orin Nano) and utilizing more advanced models of object detection such as YOLOv8 to make the onboard frame rate higher.

REFERENCES

- [1] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A Micro Aerial Vehicle Design for Autonomous Flight Using Onboard Computer Vision," *Autonomous Robots*, vol. 33, no. 1–2, pp. 21–39, 2012.
- [2] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A Node-Based Multithreaded Open Source Robotics Framework for Deeply Embedded Platforms," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2015, pp. 6235–6240.
- [3] R. B. Nagananda et al., "Autonomous UAV Navigation Using Deep Learning-Based Computer Vision Frameworks: A Systematic Literature Review," *Array*, 2024.
- [4] T. Jacob, R. Galliera, M. Ali, and S. Bagui, "Marine Vessel Tracking Using a Monocular Camera," in *Proc. Int. Conf. Deep Learning Theory and Applications (DeLTA)*, 2021, pp. 17–28.

- 5) [5] J. Lee, B. Varghese, R. Woods, and H. Vandierendonck, "TOD: Transprecise Object Detection to Maximise Real-Time Accuracy on the Edge," in Proc. IEEE Int. Conf. Fog and Edge Computing (ICFEC), 2021, pp. 53–60.
- 6) [6] P. Ganesh, Y. Chen, Y. Yang, D. Chen, and M. Winslett, "YOLO-ReT: Towards High Accuracy Real-Time Object Detection on Edge GPUs," in Proc. IEEE/CVF Winter Conf. Applications of Computer Vision (WACV), 2022.
- 7) [7] A. Koubâa, A. Allouch, M. Alajlan, Y. Javed, A. Belghith, and M. Khalgui, "Micro Air Vehicle Link (MAVLink) in a Nutshell: A Survey," IEEE Access, vol. 7, pp. 87658–87680, 2019.
- 8) [8] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Real time Tracking with a Deep Association Metric," in Proc. IEEE Int. Conf. Image Processing (ICIP), 2017, pp. 3645–3649.