Real Time Object and Gesture Recognition Using Programmable Drone

Shilpa N R[1], Yashitha P N[2], Lalith M P Bharadwaj[3], Mr. Prakash B Metre[4]

[1,2,3] *Student, Department of Computer Science and Engineering, School of Engineering, Presidency University, Bangalore*

[4]*Guide, Department of Computer Science and Engineering, School of Engineering, Presidency University, Bangalore*

Abstract- **Drones entered the commercial space as exciting, recreational albeit expensive toys, slowly transforming into a multi-billion-dollar industry with myriad commercial applications ranging from asset inspections to military surveillance.**
AI can replace humans at various levels of commercial drone use- they can autonomously control the drone flight, analyze sensor data in real time or even examine the data post-flight to generate insights. At any of these levels, it is often required to identify and locate objects-of-interest around the drone through the data captured by its sensors, making Object Detection fundamentally important to impart intelligence to a drone

I. RATIONALE

Help push boundaries for real time application in emergency rescue operations at times of floods, to detect the number of people in danger. Here, the object classifier is trained to detect the human on the screen.

Asset inspections in military surveillance help detect the human or any trained object.

II. OBJECTIVES

1. Real time object recognition
- Get an object bounding box to return from camera feed.
- Create a function that maps bounding boxes to the center of the screen.
- Add functionalities to tello's controls.
- Be able to launch tello drone and have it track the trained object.

2. Control the tello using node.js
- Build node backend which will power the tello.

- Make react frontend which makes the controller.

III. REQUIREMENTS

1. Hardware Requirements:
- RyzeRobotics tello
- Transmitter
- Windows Operating System

64 bit architecture, 4 or more CPUs/cores with at least 16GB of RAM.

2. Software Requirements:
- Python numpy
- Python OpenCV
- JavaScript
- node.js

IV. METHODOLOGY / PLANNING OF WORK

1. Real time object recognition
- Get an object bounding box to return from camera feed.
  That is whenever the object is detected on the camera feed; it needs to return an object bounding box which is done using OpenCV.
- Create a function that maps bounding boxes to the center of the screen.
  Make an algorithm where the object's bounding box/meets the centre of the screen.
- Add functionalities to tello's controls.
- Be able to launch the tello drone and have it track the trained object.

After we have trained the object using the model/classifier, we then use it to track the trained object.

2. Control the tello using node.js

- Build node backend which will power the tello.
  We will build a node for the backend in the system to power the tello.
- Make react front end which makes the controller.

We use react a JavaScript frontend platform where the controls are placed.

V. EXPECTED OUTCOMES

The expected outcome is to detect and recognize the object which are trained and make it operate with the added performance such as, to follow the trained object where the object is set as the target.

For control performance we should be able to control the drone using react frontend, which is powered by backend node.js.

REFERENCES

[1] Paul Viola and Michael Jones "Rapid Object Detection using a Boosted Cascade of Simple Features", accepted conference paper on Computer Vision and Pattern Recognition 2001.

[2] Zhong-Qiu Zhao, Member, IEEE, Peng Zheng,Shou-tao Xu, and Xindong Wu, Fellow, IEEE, "Real-Time Object Detection with Yolo", 16 April 2019.

[3] N.Revathy, T.Guhan, "Face recognition system using back propagation artificial neural networks", International Journal of Advanced Engineering Technology, vol.3, no. 1, 2012.

[4] Apoorva Raghunandan, Mohana, Pakala Raghav and H. V. Ravish Aradhya, "Object Detection Algorithms for video surveillance applications" International conference on communication and signal processing (ICCSP), India, 2018, pp. 0570-0575.

[5] https://www.researchgate.net/publication/283556 334_Multiple_Object_Detection_using_OpenCV _on_an_Embedded_Platform.

[6] https://arxiv.org/pdf/1807.05511

[7] https://www.researchgate.net/publication/310238 833_Visionbased_object_detection_and_tracking _A_review