Autonomous Surveillance Drone

Gaurav Shetty[1], Vraj Shah[2], Parva Juthani[3], Unik Lokhande[4]

[1]Fr. Conceicao Rodrigues College of Engineering
gauravshetty8703[at]gmail.com

[2]Fr. Conceicao Rodrigues College of Engineering
vrajshah05[at]hotmail.com

[3]Fr. Conceicao Rodrigues College of Engineering
juthani.parva00[at]gmail.com

[4]Professor, Fr. Conceicao Rodrigues, College of Engineering
unik.lokhande[at]fragnel.edu.in

Abstract: ***Drones are aircraft without pilot, crew or passengers on board. They are formally known as unmanned aerial vehicles which can be controlled remotely by humans using a controller or even be completely autonomous. They can be used for various purposes like aerial photography, delivering items, reconnaissance and other civil and military applications. Our project aims to develop a drone which is completely autonomous and used for surveillance purposes. The user just needs to input the GPS coordinates of the path where it wants the drone to go, and then the drone automatically generates the path and starts surveillance. We have also incorporated computer vision to detect objects, human and other aircraft using the state of the art YOLOV4 algorithm. Successfully tested all systems with positive results.***

Keywords: Optical flow, YOLOV4, INS, IRS, Computer Vision, PID

1. Introduction

This project is primarily built for border surveillance. Instead of sending army personnel for patrolling near the border, drones can be sent which is more efficient in detecting threats and safer. The use of the system can be further extended for patrolling along the perimeter of other important areas where it is dangerous for humans to go. We have achieved autonomous flight incorporating Inertial Navigation System(INS) and Proportional Integral Derivative(PID) algorithm[2]. Using Inertial Reference System we track the live location of the drone since we do not have a GPS sensor integrated with the drone. For obstacle avoidance we have used an optical flow algorithm for detecting any object approaching the drone. The drone then performs certain maneuvers to doge the obstacle and return back to the original path. Our next task is to identify objects such as weapons, aircrafts, and humans which we accomplish using Computer Vision. Dataset is taken from Open Image Dataset V6.

2. Literature Survey Methodology

After a thorough survey of previous papers related to the topic we decided to use the YOLOv4 algorithm for object detection as it gives results much faster than other algorithms and the frame rate was also higher. It also uses genetic algorithm for finding optimal hyperparameters along with new normalizing techniques and self adversarial training for data augmentation. Figure 1 and 2 shows the fps rate various object detection algorithms give and latency when used in v100 gpu . From the figures it's clear that YOLOV4 gives the best fps rate as compared to others with good speed, that is 3.7x times faster than others.

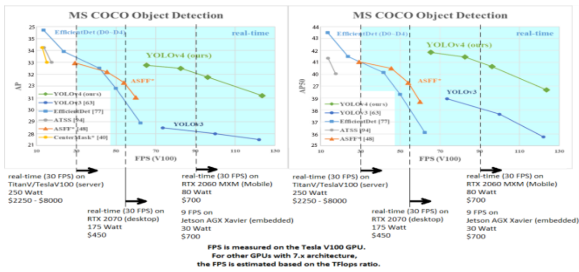

Figure 1: YOLOv4 training result

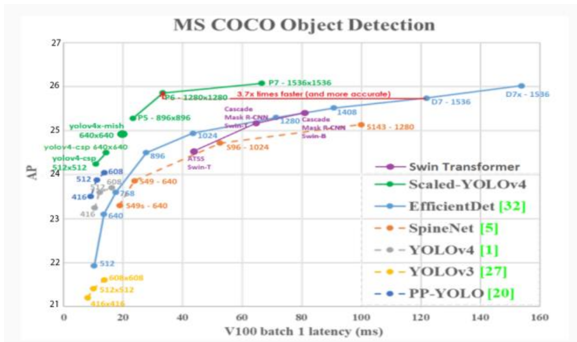

Figure 2: YOLOv4 test speed

For the obstacle avoidance system, previous research shows that we can train a model for a particular object and then create an algorithm that gives the necessary actuator commands to the drone to doge the object. But the drawback of the system was that it can not identify other objects and hence collide with it and training the model for so many objects would be unfeasible. Some research shows application of sensors like LiDar to detect objects in front of it[5]. However our drone didn't have this sensor in it. Hence we went with an optical flow algorithm which shows positive results on identifying objects based on the movement. The Inertial Navigation System was the only choice for navigating the drone as there weren't any better methods for this task as we don't have a GPS sensor in the drone.

After an extensive literature survey we found that autonomous navigation without GPS is less accurate as it relies on a good Inertial Measurement Unit(IMU) sensor. For obstacle avoidance traditional methods couldn't be used because it's not practical. Hence had to use Optical flow algorithm.

3. Methodology

The project can be divided into four subsystems. First being the computer vision part used to detect anomalies. Second the inertial reference system and inertial navigation system used for the autonomous navigation of the drone. Third the integration of obstacle avoidance in the autonomous navigation system and lastly an interactive graphical user interface to interact with the drone and also get live video feed from it for ground personnel monitoring.

1) Computer Vision

The YOLOV4 architecture can be divided into many subparts, which are - The input which comes first and it is basically what we've as our set of training images which will be fed to the network - they are processed in batches in parallel by the GPU. Next are the Backbone and the Neck which do the feature extraction and aggregation. The Detection Neck and Detection Head together can be called the Object Detector. And finally, the head does the detection/prediction. Mainly, the Head is responsible for the detection (both localization and classification). Figure 3 shows the one stage and two stage detector of the YOLOV4 network.

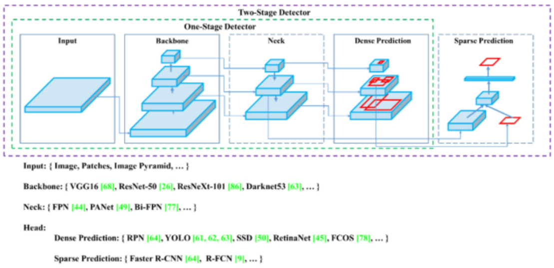

Figure 3: YOLOv4 architecture

After a lot of testing and experimental results they chose CSPDarknet53. CSPDarkNet53 is based on the DenseNet design. An additional block called SPP (Spatial Pyramid Pooling) is added in between the CSPDarkNet53 backbone and the feature aggregator network (PANet), this is done to increase the receptive field and separates out the most significant context features and has almost no effect on network operation speed. It is connected to the final layers of the densely connected convolutional layers of CSPDarkNet[2] . Figure 4 shows the detailed architecture of the YOLOv4 network.

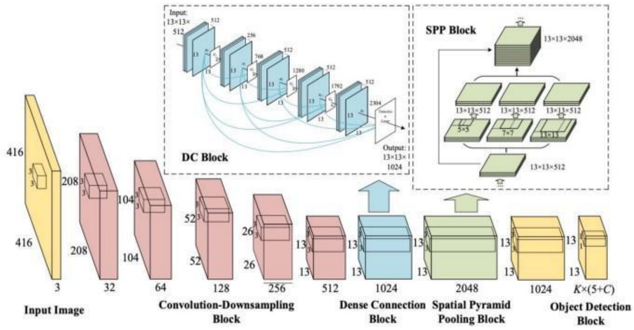

Figure 4: YOLOv4 neural network layer

Here is an overview of the operation performed. [6]

Table 1: Operational performance

Layers	Parameters: Filters	Parameters: Size / Stride	Output	Layers	Parameters: Filters	Parameters: Size / Stride	Output
Conv 1	32	3×3/1	416×416×32	DC Block	1024	3×3 / 1 ×4	13×13×2304
Maxpool 1		2×2 / 2	208×208×32	Conv 14-21	256 or 512	1×1 / 1	
Conv 2	64	3×3 / 1	208×208×64	Conv 22	1024	3×3 / 1	13×13×1024
Maxpool 2		2×2 / 2	104×104×64	Conv 23	512	1×1 / 1	13×13×512
Conv 3	128	3×3 / 1	104×104×128	SPP Block Maxpool 6-8		5×5 / 1, 7×7 / 1, 13×13 / 1 Concat	13×13×2048
Conv 4	64	1×1 / 1	104×104×64				
Conv 5	128	3×3 / 1	104×104×128				
Maxpool 3		2×2 / 2	52×52×128	Conv 26	512	1×1 / 1	13×13×512
Conv 6	256	3×3 / 1	52×52×256	Conv 27	1024	3×3 / 1	13×13×1024
Conv 7	128	1×1 / 1	52×52×128	Reorg Conv13		/ 2	13×13×256
Conv 8	256	3×3 / 1	52×52×256	Concat -1, -2			13×13×1280
Maxpool 4		2×2 / 2	26×26×256	Conv 30	1024	3×3 / 1	13×13×1024
Conv 9-12	512	3×3 / 1, 1×1 / 1 ×2	Conv 31	Conv31	K*5+C	1×1 / 1	13×13×(K*5+C)
Conv 13	512	3×3 / 1	26×26×512	Detection			
Maxpool 5		2×2 / 2	13×13×512				

The main function here is locating bounding boxes and performing classification.

The bounding box coordinates (x, y, height and width) as well as scores are detected. Here the x & y coordinates are the center of the b-box expressed relative to the boundary of the grid cell. Width & Height are predicted relative to the whole image.

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_h = p_h e^{th}$$
$$b_w = p_w e^{tw}$$

2) Inertial Navigation System and Inertial Reference system

Since we do not have a GPS sensor in our drone we use an IMU sensor for detecting the acceleration of the drone in x,y and z direction and its roll, pitch, yaw rate. Then we double integrate the acceleration to get its velocity and distance traveled. We also integrate roll, pitch and yaw rate to get the euler angles. Once we have all this value we can calculate the exact position of the drone by calculating the distance traveled in x,y and z direction with respect to time.

For its autonomous navigation we first input the GPS coordinate from the user using the GUI. Then we calculate the shortest path for completing surveillance at all the points by using an algorithm like Dijkstra's algorithm and append it in a list. We then find the distance between each successive point in the list and create a new list with this data. Then we start the drone maneuvering by finding the angle between the heading vector and targeted vector and also the distance from the list and give necessary actuator commands for it to perform the surveillance.

There is also a safety mechanism included, which makes the drone automatically come back to its base that is the starting point if the battery is very low orelse if the ground personnel presses return to base command. By this we can assure that we don't lose the drone.

3) Obstacle Avoidance System

Implementation of optical flow algorithm for detection of an object and also its direction of motion[8]. Our Drone detects any moving object approaching it and doges it by moving left or right. As you can see in the images, it uses the algorithm to identify moving objects and gives appropriate commands to the drone actuators for performing the doging motion. In figure 5 we can see that the drone detects the object in front of it along with the direction the object is moving.

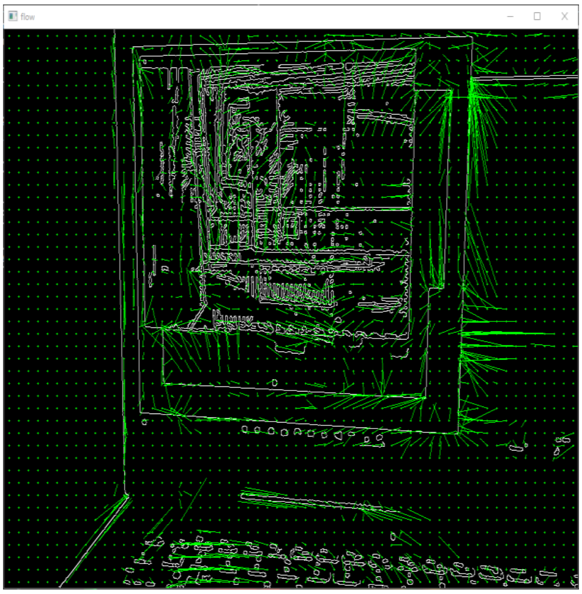

Figure 5: Optical flow

This system is integrated with the autonomous navigation system for safe movement of the drone in the desired path.

4) Graphical User Interface

We have developed an in-house GUI shown in figure 8 for interacting with the drone. It shows live video feed of the drone along with an alarm system to go off when an anomaly is detected. It also has an interactive map made using leaflet so that the user can select GPS coordinates for the path formation. It even has features to create, select and load previous paths. It also displays the orientation of the drone, that is the roll, pitch, yaw as well as its vertical speed and drone velocity.

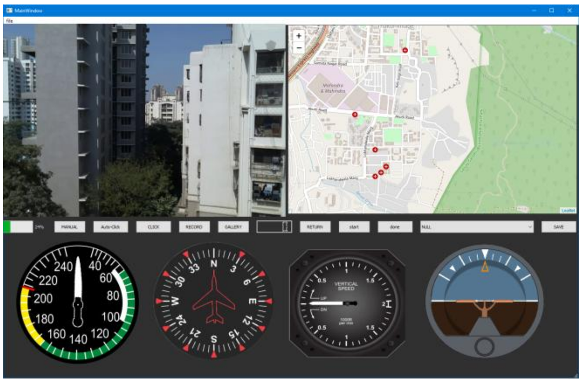

Figure 6: Graphical User Interface (GUI)

5) Execution flowchart

In figure you can see the execution of the program. First the device is switched on and all systems are initialized. Then the user can enter the GPS coordinates of the once he enters that the system creates a flight path using Dijkstra's algorithm. Then the drone is powered on and the autonomous navigation algorithm starts executing. It sees if the drone is in the right path throughout and corrects the path of the drone if it deviates simultaneously the drone transmits the video feed to the ground station where it is analyzed using the computer vision algorithms to detect anomalies. If found the image is captured for later inspection and an alarm is raised.

Paper ID: MR22428174616 DOI: 10.21275/MR22428174616

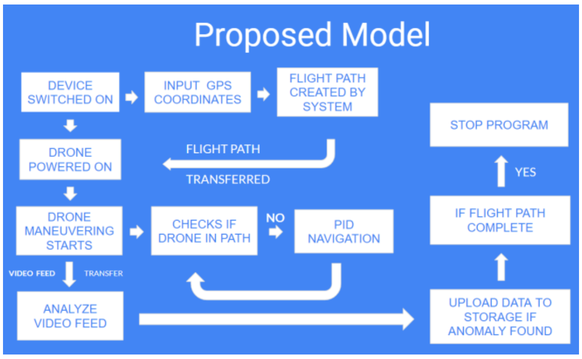

Figure 7: Execution Flowchart

4. Result

The software has been implemented on DJI Tello drone which has a 720 pixel camera with IMU sensor and infrared sensor. The graphical user interface initially ran slow but after implementing parallel computing, it was very responsive. Selecting the GPS coordinate from the map is difficult if the coordinates are at a distance less than 10 meter. The FPS rate of the video feed was around 15 and successfully triggered alarm whenever necessary and stored the data for further use. The data received from the IMU sensor was successfully displayed by the GUI in real time. Figure 8, 9 and 10 shows the result of object detection using computer vision.

Figure 8: YOLOv4 test result

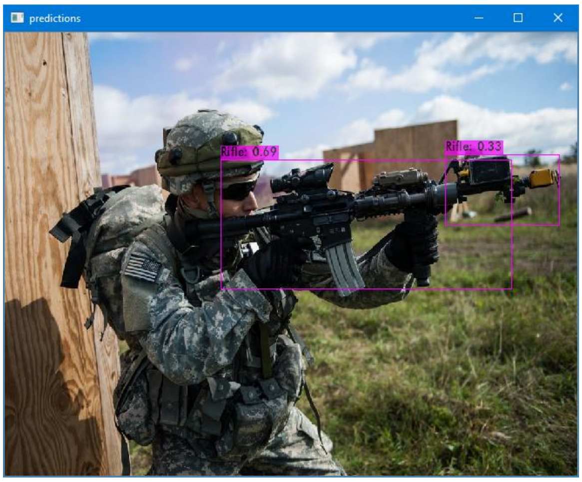

Figure 9: YOLOv4 test result

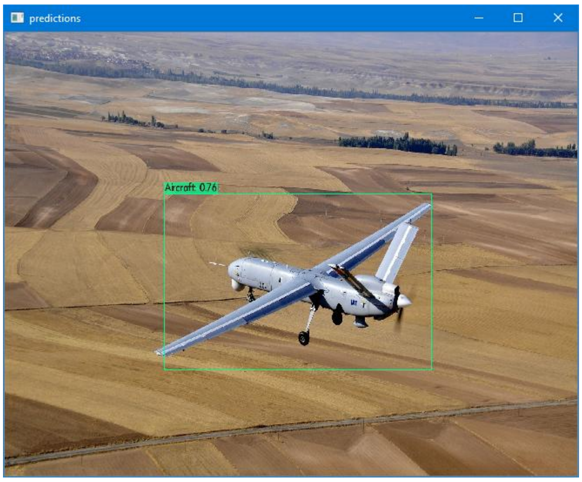

Figure 10: YOLOv4 test result

In the figure you can see that the YOLOv4 algorithm has successfully detected the objects it has been trained on with good accuracy. The model is currently trained to detect rifle, aircraft, person, handgun, shotgun, drones and other weapons. More objects can be added but simply include it in the training dataset.

5. Conclusion

The YOLOv4 proved to be very effective in terms of detecting objects and classifying them with high accuracy and even quickly. The model created can be used to recognise more objects by simply adding more training dataset of the desired objects. The autonomous navigation could have been more precise if the IMU sensors provided more accurate data of the drone movement. Adding a GPS sensor would be more reliable but may create problems in the region where the connection with satellite is not possible. The obstacle avoidance system proved to be good and reliable as it could detect each moving object that approached the drone and successfully give actuator commands to doge it.

In conclusion we can say that the drone performance was really good given its low cost of fabrication, only the GUI can be made faster in terms of processing the data by using more sophisticated parallel computing techniques.

References

[1] Hu, J.; Lanzon, A. (2018). "An innovative tri-rotor drone and associated distributed aerial drone swarm control". *Robotics and Autonomous Systems*. **103**: 162–174. doi:10.1016/j.robot.2018.02.019.
[2] Kiam Heong Ang, G. Chong and Yun Li, "PID control system analysis, design, and technology," in IEEE Transactions on Control Systems Technology, vol. 13, no. 4, pp. 559-576, July 2005, doi: 10.1109/TCST.2005.847331.
[3] arXiv:2004.10934v1
[4] L. Guangling and P. Yonghui, "System Design and Obstacle Avoidance Algorithm Research of Vacuum Cleaning Robot," 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), 2015, pp. 171-175, doi: 10.1109/DCABES.2015.50.

[5] D. Ghorpade, A. D. Thakare and S. Doiphode, "Obstacle Detection and Avoidance Algorithm for Autonomous Mobile Robot using 2D LiDAR," 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA), 2017, pp. 1-6, doi: 10.1109/ICCUBEA.2017.8463846.
[6] "YOLOv4 model architecture",Accessed: Sept. 19, 2021. [Online]. Available:https://iq.opengenus.org/yolov4-model-architecture/
[7] H. Chao, Y. Gu and M. Napolitano, "A survey of optical flow techniques for UAV navigation applications," 2013 International Conference on Unmanned Aircraft Systems (ICUAS), 2013, pp. 710-716, doi: 10.1109/ICUAS.2013.6564752.
[8] E. Dur, "Optical Flow-based obstacle detection and avoidance behaviors for mobile robots used in unmaned planetary exploration," 2009 4th International Conference on Recent Advances in Space Technologies, 2009, pp. 638-647, doi: 10.1109/RAST.2009.5158270.
[9] O. C. Ann and L. B. Theng, "Human activity recognition: A review," 2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014), 2014, pp. 389-393, doi: 10.1109/ICCSCE.2014.7072750.
[10] krishna, M & Neelima, M & Mane, Harshali & Matcha, Venu. (2018). Image classification using Deep learning. International Journal of Engineering & Technology. 7. 614. 10.14419/ijet.v7i2.7.10892.
[11] Development of an Autonomous Drone for Surveillance Application .International Research Journal of Engineering and Technology (IRJET)
[12] Dinesh M.A. 1, Santhosh Kumar S. 2, Sanath J. Shetty3, Akarsh K.N. 4, Manoj Gowda K.M.
[13] Salih, Atheer & Moghavvemi, Mahmoud & Haf, Mohamed & Gaeid, Khalaf. (2010). Flight PID Controller Design for a UAV Quadrotor. Scientific research and essays. 5. 3660-3667.
[14] Arjun M. Nair, Manoj A. Talreja, Prithvi Bhaskar, Kiran Kumar T, 2014, Design Solution For Flight Control of A Quadcopter, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 03, Issue 05 (May 2014),
[15] Yannick Benezeth, Pierre-Marc Jodoin, Venkatesh Saligrama, Christophe Rosenberger. Abnormal events detection based on spatio-temporal co-occurrences. Conference on Computer Vision and Pattern Recognition, Jun 2009, Miami, United States. ff10.1109/CVPRW.2009.5206686ff. Finra-00545513ff.
[16] Sharma, Anshika. (2016). A survey on object recognition and segmentation techniques.
[17] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma. Pcanet: A simple deep learning baseline for image classification? IEEE Transactions on Image Processing, 2015
[18] K. Goya, X. Zhang, K. Kitayama, and I. Nagayama. A method for automatic detection of crimes for public security by using motion analysis. In International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009.
[19] Arjun M. Nair, Manoj A. Talreja, Prithvi Bhaskar, Kiran Kumar T, 2014, Design Solution For Flight Control of A Quadcopter, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 03, Issue 05 (May 2014),
[20] Xianyu Chang Chaoqun Yang Junfeng Wu Xiufang Shi and Zhiguo Shi Anti Unmanned-vehicle Group, State Key Lab. of Industrial Control Technology, Zhejiang University, China College of Information Science and Electronic Engineering, Zhejiang University,
[21] N. Gageik, P. Benz and S. Montenegro, "Obstacle Detection and Collision Avoidance for a UAV With Complementary Low-Cost Sensors," in IEEE Access, vol. 3, pp. 599-609, 2015, doi: 10.1109/ACCESS.2015.2432455.