IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 10, Issue 4, Aug-September 2022
ISSN: 2320 – 8791 (Impact Factor: 2.317)
www.ijreat.org

Development of Embedded System for Collision Detection using Image Processing for an UAV applications on ARM Cortex-A53 Processor

Chaitra R[1], Dr. Siva Subha Rao Patange[2] and Siddesha K[3]

[1] M. Tech student, Department of ECE, Dr. Ambedkar Institute of technology, Bengaluru, India
[2] Sr. Principal Scientist, STTD, CSIR-NAL, Bengaluru, India
[3] Assistant Professor, Department of ECE, Dr. Ambedkar Institute of technology, Bengaluru, India

Abstract

Collision detecting and recognizing in the UAV is mandatory to save the trip to the required destination however autonomous UAVs should find collision in the surrounding area. Various systems were introduced to find collision using a number of types of sensors like lidar and radar. The aim is to use the yolov5 (you only look once) raspberry pi to find the collisions for UAVs. It has a few advantages like several obstacle detection algorithms. Here by looking at the image just in one forward pass and by using a CNN it gives the bounding box and a probabilities class of an object and detects the collision faster as compared to others algorithms in an image, videos and in real-time.

Keywords: **Convolution Neural Networks (CNNs), Graphics Processing Units (GPUs), Unmanned Aerial Vehicles (UAVs), You Only Look Once (YOLO).**

1. Introduction

The collision finding in videos and pictures known as recognizing and detecting. The computer vision makes UAVs even more fragile to find things in real time. Collision detection is very important while reaching the desired destination for the UAVs. As humans detects collision more easily and faster than UAVs, thus we will be choosing a right system which will find the collision in real surrounding more faster. Numerous algorithms are present to find collision recognizing, here yolov5 is used to find the collision for UAVs in real-time. The purpose of the designed work for training real-time model, such as the first step in doing the new ideas related to interaction of machines. This system includes UAVs for private driving on a scale a place where it is less expensive and easier to innovate and later use the same in the real world. The purpose of this paper are as follows:

- Identify Yolov5 that is most efficient and effective in real period classifying, labeling and recognizing.
- Testing the results obtained before and after collision found for selected Yolo(v5) simulations for pictures, videos and real-time.

UAV systems are usually divided into three sub- systems detection, planning and control. The visual system is responsible for translating raw sensor data into the environment model, the planning system makes purposeful decisions based on the natural model, and the control system performs the planned actions.

The purpose of this thesis is to study the visual problem in the context of the discovery of real-time UAV objects. Although we stand on the brink of a new era of UAVs, this technology is far from mature. Self-propelled UAVs have to share and operate in an unpredictable environment with many characters such as pedestrians, other vehicles, and birds. Therefore, there is a need to provide autonomous vehicles with reliable sensing systems to facilitate and improve the vehicle's ability to understand the environment. In the past, the UAV relied heavily on technologies such as Radar, LiDAR, GPS and other sensors when mapping the environment. However, compared to these sensors the cameras are less expensive so it would be, from a financial point of view alone it would be more advantageous to integrate them into the UAV. In addition, cameras have a higher resolution than any other sensor used, and there are photo-based data sets available. Over the past few years, some of the video feed equipment detection systems have been developed that can be used as part of an independent vehicle recognition system. With the growth of in-depth learning, data has proven to be a limiting and driving factor when training DNNs, especially within a CV. Creating a variety of reallife data sets for driving situations is time- consuming and expensive. Instead, visual data sets can be automatically created and labeled with the highest speed and accuracy, and thus can help solve data shortages with UAV systems labels. Therefore, if real-time real-time image acquisition receivers are shown to have the same functionality as real-time real-time data acquisition data, or if transaction data can be used as consistent data during training, it can accelerate self-improvement**.**

2. Related works

A quick and easy way to finding real-surrounding photos are presented. The you look only once model aimed for getting pictures precisely and quickly for real-surrounding pictures. In evaluation for detecting objects before YOLO, like R-CNN, YOLO presented a only unified design for picture to a bounding-boxes & class to every box. The results obtained are much faster and more accuracy in finding the objects [1]. The standard detecting system was developed using complex degradation processes in data training sets such as noise, blurring, rotation and cropping. The system skilled with reduced training sets which resulted better standard performance and advanced durability. Tests have shown that a model trained with standard sets does not have a good ability to perform normally in damaged pictures with poor durability. The system was then skilled with degraded pictures for improving average accuracy. It was shown that average accuracy for degraded pictures was better than normal model [2]. YOLO system design is developed with new YOLO-R system design was suggested for increase system capacity for obtaining data on shallow featuring for pedestrian along with transit layers to the first YOLO system. YOLO(v2) and YOLO(R) system models are tested for INRIA dataset. Test outcomes explains that YOLO(R) system is advanced comparing actual YOLO(v2) system model. The numbers of acceptance frames were extended 25frames/s, basically meeting real-time requirement [3]. An obstacle finding method in the internal surroundings suggested by combining the YOLO obstacles finding way and the light fields cameras, classifies them to classes and label them in the picture. The pictures of the joint obstacles were labeled for training YOLO. The obstacle filter was used for taking out the indifference obstacles. The various kinds of scenes with pedestrian, chairs, books etc. are verified to give the efficiency for finding obstacles [4]. To identify solder joints for car door panels based on the YOLO method that provides the type and setting of solder joints in real-time. For more precise joints, this paper favors a YOLO by predicting of multi levels, predicts mapping of various sizes and combines predictive results to get the final result. The YOLO suggested identifies position of the solder joints accurately in real phase. This is helpful to rise productivity for making line and it has a more importance to flexibility and real phase of the welding of automobile door panels [5].

3. Methods

The Yolo(v5) high-end detector is aimed to attain great accuracy and real-period results. YOLOv5 is more progressive than ever before types. It makes use of a single neural network, predicting the shape of collision and class results with a one repetition. It is also reached by considering the recognizing of a collision problem as a retrieve tasks and it converts input images into their individual Class chances are related with points. YOLO(v5) produces numerous (S x S) grids from the picture the inputs and the B border boxes are expected, containing the length, width, center x and y. For each of these boxes has its particular P (collision) value and expects the number of classes in it as C and it has a restricted group chances are that P class in (S x S) has roughly in it. Whole prediction for the system says [S x S x (B x 5 + C) and the 5 digit signifies separately joining box as 4 and 1 as probability of the collision. In the test, the system finds the number of classes existing. P minimum is defined at the start of the test and the system finds items in (P class> P minimum). In the later processing stage, identical detection of the similar collision is stopped by means of non-plural compression.

3.1 Creating A Custom Dataset

Makesense.ai is an opensource and free to use under GPLv3 license. We do not need to install any dependency libraries as we can just use it through the browser. It will support multiple label and also support output file formats like YOLO, VOCXML, VGG JSON, CSV. It uses AI to make our work more productive.

STEP1: Collecting the several images to be identified in a folder for training and validation of the images.

STEP2: Load the pictures to the tool and create a labels to find the objects.

STEP3: Prediction box is drawn for all pictures and then labels are allocated.

STEP4: Select the **Actions → Export Annotations → A zip package containing in YOLO format → Export** and the zip file of trained-labels will be downloaded.

STEP5: Last step will be repeated to export the validation-labels.

STEP6: These folder path will be allocated in the data.yaml file.

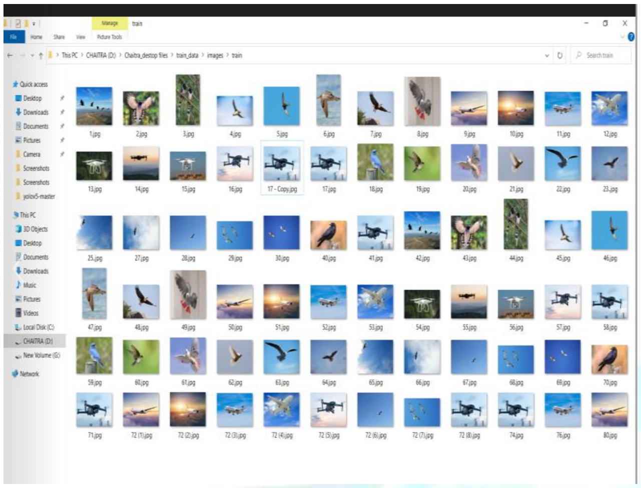
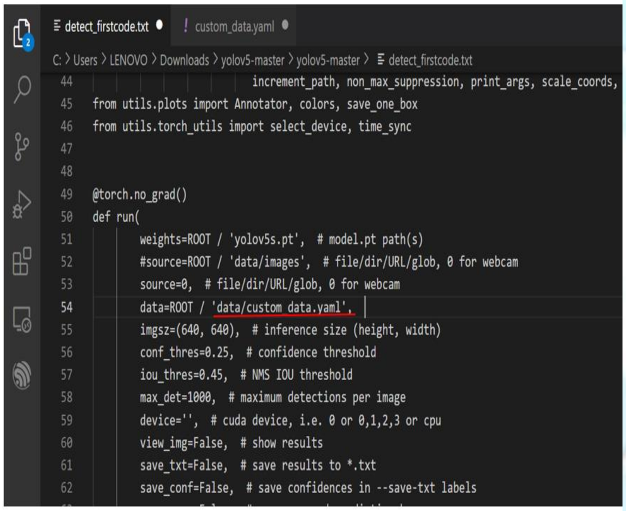


Fig. 1 Images collected for the train and validate.

STEP7: The new customized dataset will be formed and this generated file will be uploaded in the main python script.



detect_firstcode.txt
custom_data.yaml
C: > Users > LENOVO > Downloads > yolov5-master > yolov5-master > detect_firstcode.txt

```
                                increment_path, non_max_suppression, print_args, scale_coords,
from utils.plots import Annotator, colors, save_one_box
from utils.torch_utils import select_device, time_sync


@torch.no_grad()
def run(
        weights=ROOT / 'yolov5s.pt',  # model.pt path(s)
        #source=ROOT / 'data/images',  # file/dir/URL/glob, 0 for webcam
        source=0,  # file/dir/URL/glob, 0 for webcam
        data=ROOT / 'data/custom_data.yaml',
        imgsz=(640, 640),  # inference size (height, width)
        conf_thres=0.25,  # confidence threshold
        iou_thres=0.45,  # NMS IOU threshold
        max_det=1000,  # maximum detections per image
        device='',  # cuda device, i.e. 0 or 0,1,2,3 or cpu
        view_img=False,  # show results
        save_txt=False,  # save results to *.txt
        save_conf=False,  # save confidences in --save-txt labels
```

Fig. 2 Snapshot of python scrip file

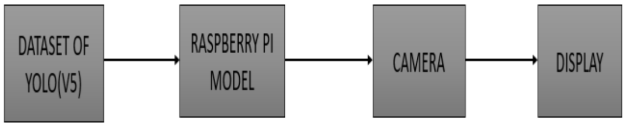


Fig. 3 General overview of the Detection using YOLOv5.

At YOLO(v5), bounding boxes has high grade of assurance are obtainable for particular as individually grid cell expects several binding boxes. Consequently, YOLO produces a tensor as the output of a value similar to [S x S x (B ∗ 5 + C)]. Then, YOLO(v5) predicts results by confidence points by obtaining a binding boxes when input is given to the algorithm in the form of an image or video.

4. Hardware and Software Components

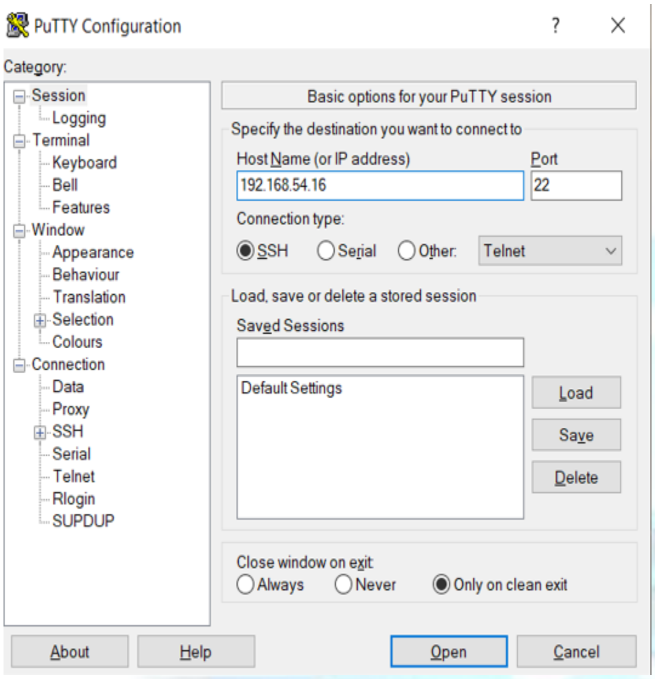
4.1 Raspberry Pi

This working includes machine-learning with opensource firmware and Raspbian OS is installed to raspberry pi model. Open-CV is the open-source firmware used here that has a huge source helps to solve actual time device difficulties and image-processing tasks with a python programming language.

Fig. 4 Raspberry Pi 3 Model B Board.

4.2 Putty

The PuTTY is a firmware application so we can run from our desktop/laptop to connect the raspberry-pi commanding-line interface. SSH (secure shell) is used for terminal window opening on our desktop/laptop, where the commands will be sent to the Raspberry-Pi and receive data. Several programs can be uses to connect the terminal of the Raspberry-Pi, however PuTTY is by far the most popular. Here putty-app is used for accessing the raspberry-pi to our PC. Then IP address of the model will be entered. In this case the IP was 192.168.54.16.



Specify the destination you want to connect to

Host Name (or IP address): 192.168.54.16 Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions

Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

Fig. 5 PuTTY Configurator

4.3 OpenCV and Python

Python was chosen as the language as python an advanced planning language, easy to read and code, to do is a programming language that is widely used to train YOLOv5 model. CUDA are involved, as they allow for algorithm training on the GPU, which makes it faster and more efficient than training in the CPU.

Open-cv can be termed as an excellent basis of source information machine-learning and picture processing and shows a main part in actual time results which is considered as a important in today's structures with this data can processes pictures videos to find obstacles in UAV. Open-cv code is done using Python language. There are links to python java and MATLAB the API for these workspaces can be found in online publications wrap words in a few editing languages have been developed to encourage more audience acceptance in version the python script and python binding of the selected sub-set of OpenCV functions is released as open-cv for use on web sites cuda-based GPU interface is still in progress open-cv works for following desktop programs windows.

5. Working of Yolo(v5)

First, a picture is taken and the YOLO algorithm is used. In our example, the picture is splited to 3x3 grids matrix. We can split an picture to any number grids, depending on the complexity of the image.

Once a picture divided, each grid goes through stages again localization of the object. Self-confidence points for each grid are available.

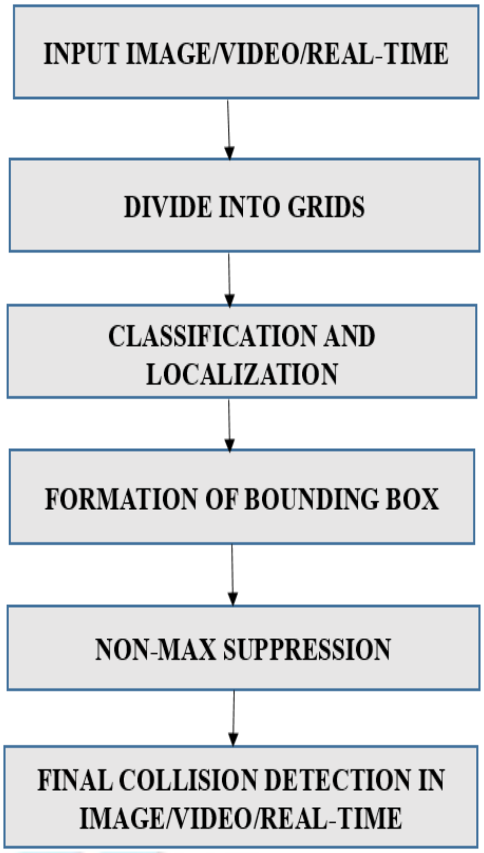

INPUT IMAGE/VIDEO/REAL-TIME
DIVIDE INTO GRIDS
CLASSIFICATION AND LOCALIZATION
FORMATION OF BOUNDING BOX
NON-MAX SUPPRESSION
FINAL COLLISION DETECTION IN IMAGE/VIDEO/REAL-TIME

Fig. 6 Flowchart of working of YOLOv5

If nothing else was found on the grid, and then the value and number of the binding box the grid will be zero or if something is found in the grid then the resistance will be 1 plus the box binding value it will be its corresponding acquisition rates thing. To increase the accuracy of the discovery of the collision the anchor boxes are used.

5.1 Predictions of Bounding box:

Yolov5 is used to find the exact boxes attached to the image video later image separates the s x s grids by providing connecting boxes with the class then we apply image classification with local techniques and the label is given to grid then check each grid and mark the conflict label with the box if obstacles are found then the value of 1 is given and value of 0 is assigned if no obstacles are found in the grid.

Consider the fig5, the input picture is given and later it is divided to 3 x 3 matrixes. The label is given to all the grids and it go through both picture classification and obstacles localization. The vector label is named as Y and it gives 7 values for each grid.

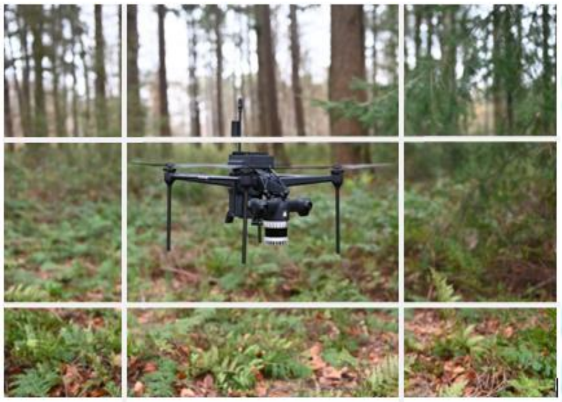


Fig. 7 3x3 grids Image.

$$Y = \begin{Bmatrix} Pc \\ Bx \\ By \\ Bh \\ Bw \\ C1 \\ C2 \end{Bmatrix}$$

Table. 1 Vector Y values.

Pc – shows any obstacle detected in the grid. If found Pc=1 else 0.
Bx and By - shows center co-ordinates.
Bh and Bw – shows bounding height & bounding box weight.
C1 and C2 –shows the obstacle class.

In the input image, no obstacle was detected in the first grid, so it is given as

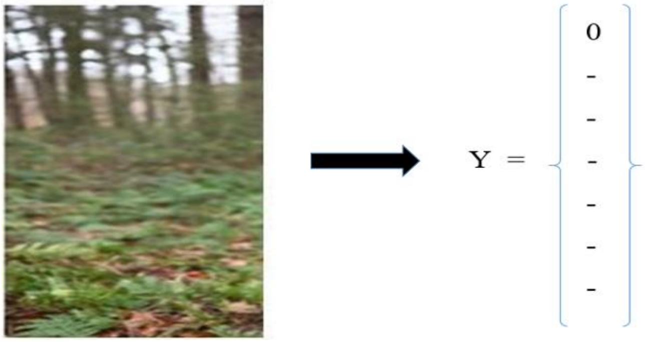


Fig. 8 Representation of Bounding box and class values for grid1.

The 5th grid contains an Obstacles. Then the grid 5 can be shown as

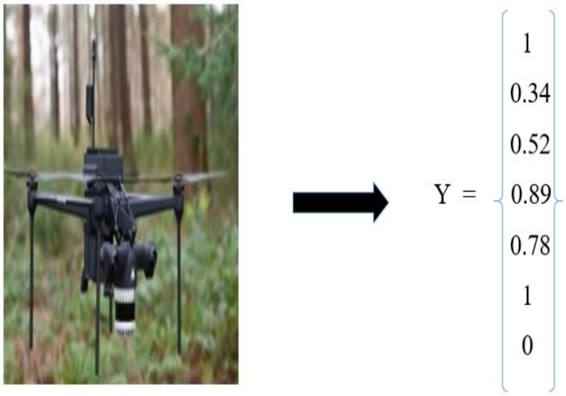


Fig. 9 Representation of Bounding box and class values for grid5.

Then the grid 5 can be shown as the two or more-grid cell has the like obstacle at the mid area of the collision is located, along a cell point is taken. With this to obtain an accurate analysis of approximately we can use the two ways. There is a Intersections over Union- and Non-max-suppression. In IoU it takes the real and analytical bond box value to determine the value IoU for two boxes using formula. If the IoU value is above, equivalent to limit value and then check for a better prediction. The maximum value is a pretend value. This also take a huge bound value for rise in accuracy nor for better expected of object. One more choice is non-max-suppression, for such instances, the utmost value for above box gets and the same boxes IoU is pushed. Frequent test is done until the cell box is designated over for a binding box for that obstacles.

IoU = Area of Intersection / Area of Union.

By using binding boxes to get a obstacles, only one obstacle can be identified by a cell. So, to find out more than one then anchoring box suits well. The picture here shown has both airplane and pilot in the midpoint arrives below the same cell. In such instance, anchor box method is used.

Any number of anchoring boxes can be used to get a more than one obstacles in a single picture. Here we have taken two anchor box. The picture shows the two-anchor box. Vertical box for a single airplane horizontal is for a pilot box. For this type of scattering detection, the Y-label contains 14values i.e., the value of both the boxes. The Pc in both anchoring box shows the presence of the obstacles and also bx-bh-bw in both cases the anchoring box assigns the values of the equivalent bounding box.

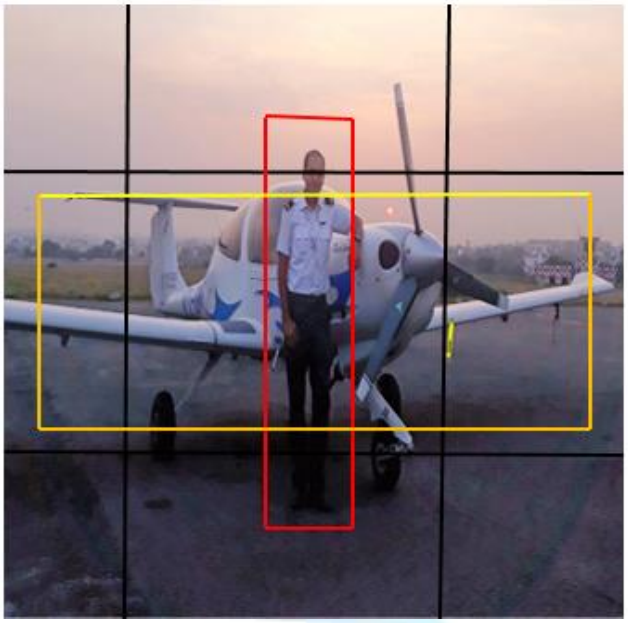


Fig. 10 Example for anchor box.

The value of class in first anchoring box is (1-0) since roughly obtained was airplane. In the circumstances of the anchoring box second, the obstacle found is a pilot so value of class is (0-1).

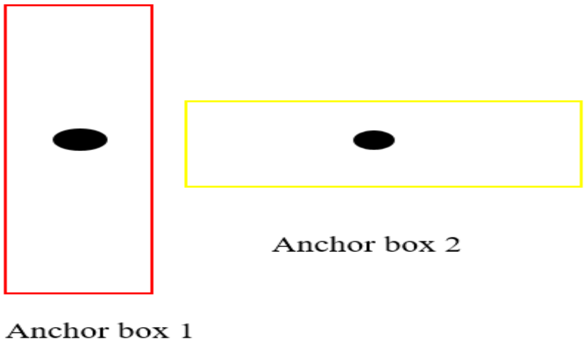


Fig. 11 Representation of Anchor box.

$$Y = \left\{\begin{matrix} Pc \\ Bx \\ By \\ Bw \\ Bh \\ C1 \\ C2 \\ Bx \\ By \\ Bw \\ Bh \\ C1 \\ C2 \end{matrix}\right\} = \left\{\begin{matrix} 1 \\ Bx \\ By \\ Bw \\ Bh \\ 1 \\ 0 \\ Bx \\ By \\ Bw \\ Bh \\ 0 \\ 1 \end{matrix}\right\} \begin{array}{l} \text{Anchor box 1} \\ \text{AIRPLANE} \\ \\ \text{Anchor box 2} \\ \text{PILOT} \end{array}$$

Table. 2 Anchor box prediction values.

In this case, the Y matrix form will be Y = 3x3x14 or Y=3x3x2x7. For the two anchoring boxes, it is a 2x7. Following the completion of the algorithm training process, in an image, in a video covering and in real environment using the raspberry pie model. The performance results obtained for the conducted for the given statistics, it is worth noting that the models are successful finding and recognizing several objects at different angles and distances, with a more level of accuracy and confidence. Tests are also repeated, providing live feeds from web camera and use real environment as test input.

In this paper by using Yolov5 & raspberry pi for finding obstacles in an image/ video covering and in real surroundings. Finding the obstacles in a real-surroundings using webcam was the main aim of this project. GPU version is really fast for the functionalities to achieve accurate using anchoring boxes. With this YOLOV5 & raspberry pie it is easy to find the obstacles with the prediction boundaries and grids. For the live detection we have used webcam to detect live obstacles.

6. Results and Discussions.

Following the completion of the algorithm training process, in an image, in a video covering and in real environment. To test the YOLOv5 method. The performance results obtained for the conducted tests illustrated as

(i) From the given statistics, it is worth noting that the obstacles at different angles with a more level of confidence.

(ii) Test are also repeated, providing live feeds from web camera.

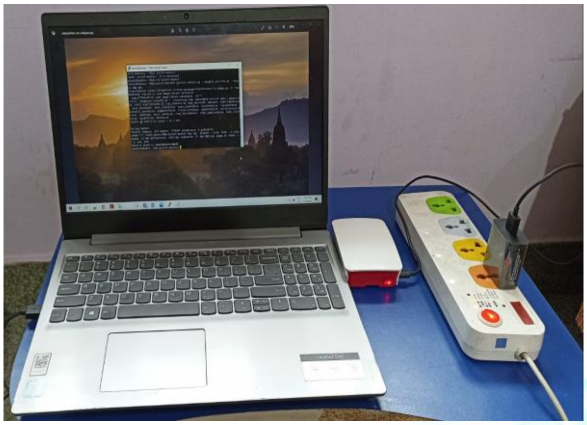


Fig. 12 Snapshot of experimental setup.

In this paper by using Yolov5 &raspberry pi for finding obstacles in an image/ video covering and in real surroundings. Finding the obstacles in a real-surroundings using webcam was the main aim of this project. GPU version is really fast for the functionalities to achieve accurate using anchoring boxes.

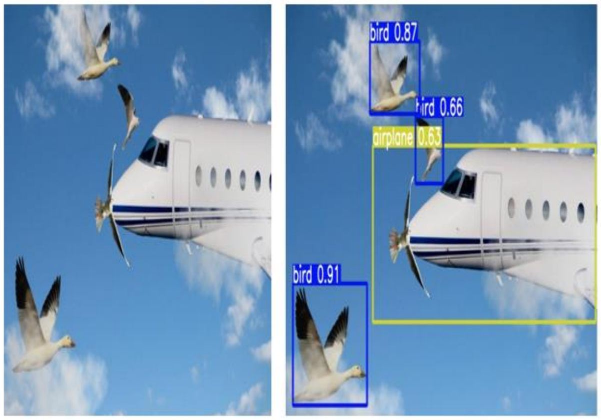


Fig. 13 Before and after obstacle detected.

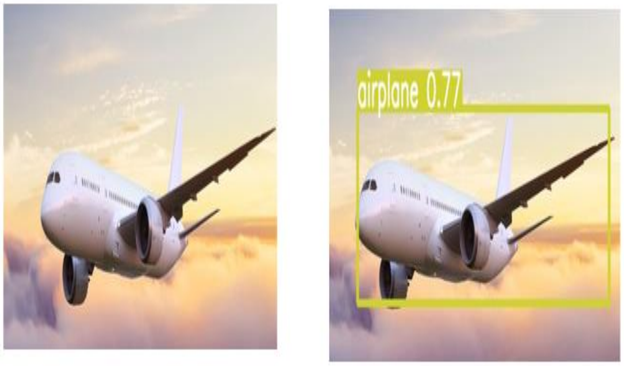


Fig. 14 Before and after obstacle detected.

With this YOLOV5 & raspberry pie it is easy to find the obstacles with the prediction boundaries and grids. For the live detection we have used webcam to detect live obstacles. The appropriate sum for predictions completed for the system above context as predictions. For better measure, specially for directing the periods of flexibility to be alike in data. The illustration is shown as

Accuracy = Correct predictions/Total no of predictions.

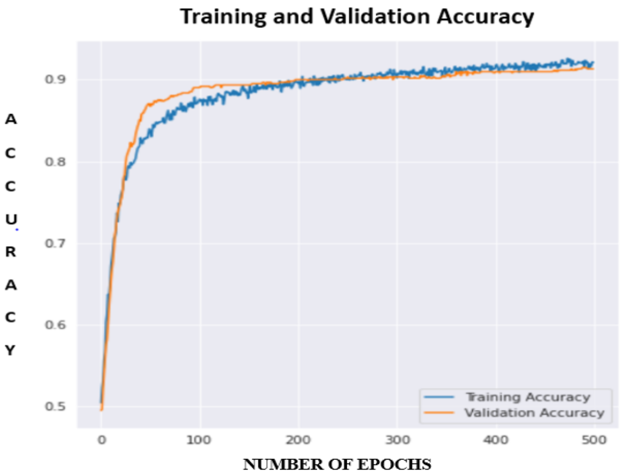


Fig. 15 Training and validation accuracy.

Therefore, accurate positive is well-defined as the exact acquisition of a trained obstacle category. Accurate Negative is well- defined as misinterpretation, meaning no obstacles found. The False Good can be described like discovery of negative, meaning that a discovery there is something to view. False Negative termed as the simple truth is no more found means that the system futile to detect vital obstacle to be viewed.

7. Conclusions and Future work.

In this project, we have worked on the YOLOv5 algorithm for the purpose of objects detection in UAV. This algorithm is standard, very efficient different and general techniques from natural image son different domains. The Yolov5 method is easy to form and train directly on the complete picture. Regular plans limit the extractor somewhere. YOLOv5reaches out to the whole picture/real-surroundings in predicted box boundaries. Also, to predicts a limited incorrect points in the related location and successfully able to detect collision in real-surroundings on embedded devices. The future work is to integrate this YOLO(v5) technique to detecting collision approaching the sideways and maneuver the UAV's accordingly.

References

[1] Joseph Redmon,Santosh Divvala,Ross Girshick,Ali Farhadi. ”You Only Look Once: Unified, Real-Time Object Detection” [J].2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2016:779788.

[2] Chengji Liu ,Yufan Tao ,Jiawei Liang ,Kai Li1 ,Yihang Chen, ”Object Detection Based on YOLO Network” 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC 2018).

[3] Wenbo Lan, Jianwu Dang, Yangping Wang and Song Wang, ”Pedestrian Detection Based on YOLO Network Model” 978-1-5386-60751/18/$31.00 ©2018 IEEE.

[4] Rumin Zhang, Yifeng Yang, ”An Algorithm for Obstacle Detection based on YOLO and Light Filed Camera”, 2018 Twelfth International Conference on Sensing Technology(ICST).

[5] Zhimin Mo1, Liding Chen1, Wenjing You ”Identification and Detection of Automotive Door Panel Solder Joints based on YOLO” 978-1-72810106-4/19$31.00 ©2019 IEEE.

[6] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99.

[7] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park,: “Pvanet: Deep but lightweight neural networks for real-time object detection,” arXiv preprint arXiv:1608.08021, 2016.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.- Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision. Springer, 2016, pp. 21–37.

[9] J. Dai, Y. Li, K. He, and J. Sun, “R-fcn: Object detection via region-based fully convolutional networks,” in Advances in neural information processing systems, 2016, pp. 379–387.

[10] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767,2018.

[11] S. Agarwal, J.O.D. Terrail, F. Jurie, Recent Advances in Object Detection in the Age of Deep Convolutional Neural Networks. arXiv:1807.04606(2018).

[12] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, S.W. Baik, Convolutional neural networks based fire detection in surveillance videos. IEEE Access 6, 18174–18183(2018).

[13]S. Ioffe, C. Szegedy, in International Conference on Machine Learning. Batch normalization: accelerating deep network training by reducing internal covariate shift (2015).

[14] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA. Rethinking the inception architecture for computer vision (2016).

[15] Rajan P Thomas, Jithin K K, Hareesh K S, Habeeburahman C A, Jithin Abraham,“ Range Detectionbased on Ultrasonic Principle”, February 2014.

[16] Shridevi A Mali1, G Sravanthi2*, Siva Subba Rao P3, Raja S3,Sushma S J1, A R Reddy2, Rohith P Maben3, “AnAlgorithm For Obstacle Avoidance Controller Using Ultrasonic Sensor For Mini Aircraft Applications”International Journal of Engineering Technology,Management and Applied Sciences, , Volume 5,Issue 3,March 2017.

[17] G Sravanthi1, Shridevi A Mali2, Siva Subba Rao P3, Intelligent Landing Technique.

[18] Ultrasonic Sensor for Mav Applications, International Journal Of Pure And Applied Mathematics, Volume 114, No. 12 2017.

[19] Jeffery B. Saunders, Brandon Call, Andrew Curtis, Randal W. Beard, “Static And Dynamic Obstacle Avoidance In Miniature Air Vehicles”, American Institute Of Aeronautics And Astronautics, September 2005.

[20] K.R.Prashanth, Preetham Shankpal, B.Nagaraja, Govind R.Kadambi, S.R.Shankapal, “Real Time Obstacle Avoidance And Navigation Of A Quad-Rotor Mav Using Optical Flow Algorithms”, Sastech Journal, Volume 12, Issue 1, April 2013.

[21] Marilyn Prascilla S, K.Sridhar, Siva Subbarao P, Obstacle Recognition and Collision Avoidance using HC-SR04 Sensor for Quadrotor Micro Air Vehicle Application, IJIRSET, Vol. 8, Issue 11, November 2019.

[22] Sharmila A, Ramesh P, Siva Subba Rao Patange ,Dr.S.Raja, Dr.Aby. K.Thomas,Obstacle Detection using Ultrasonic sensor in MAV (Micro Air Vehicle), (IRJET) Volume: 05 Issue: 04 | Apr-2018.

[23]Siddesha K and G V Jayaramaiah, “A Review on Techniques for Power Management in Embedded Systems”, Journal of Emerging Technologies and Innovative Research, ISSN:2349-5162, Vol.6, Issue 5, page no.514-519. 2019.