2021 Second International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP) | 978-1-7281-8760-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICA-SYMP50206.2021.9358449

Drone detection using YOLOv3 with transfer learning on NVIDIA Jetson TX2

Daniel Tan Wei Xun, Yoke Lin Lim, Sutthiphong Srigrarom

***Abstract*—The rise of drones in the recent years largely due to the advancements of drone technology which provide drones the ability to perform many more complex tasks autonomously with the incorporation of technologies such as computer vision, object avoidance and artificial intelligence. However, the misuse of drones such as the Gatwick Airport drone incident resulted in major disruptions which affected approximately 140,000 passengers. To deter this from happening in the future, drone surveillance are extremely crucial. With this, it will be achieved firstly by detection and followed by tracking of drones. This paper presents and investigates the use of a deep learning object detector, YOLOv3 with pretrained weights and transfer learning to train YOLOv3 to specifically detect drones. We demonstrated that the detection results from YOLOv3 after machine learning had an average accuracy of 88.9% at input image size of 416 x 416. Finally, we integrated into NVIDIA Jetson TX2 for real-time drone detection.**

***Index Terms*—YOLOv3, Machine Learning, Deep Learning, Drone Detection, Transfer Learning, NVIDIA Jetson TX2.**

I. Introduction

In the recent years, with the increased of usage and interests with drones are due to the ability to perform increasingly complex tasks such as surveillance and offensive operations in the military, emergency response, agriculture, construction planning, security and safety of personnel to parcel deliveries. The advancements of drone technology provide drones the ability to perform many complex tasks autonomously. This is done by the incorporation of various technologies such as computer visions, object avoidance and artificial intelligence. [1] However, the misuse of drones such as the Gatwick Airport drone incident which happened between 19 to 21 December 2018 where drone sightings close to the airport were reported. This resulted in the cancellation of hundreds of flights, causing major disruptions which affected approximately 140,000 passengers. [2] These calls for the ability to detect and track drone more quickly and effectively without further disruption to the air traffic.

II. YOLOv3 Object Detector

YOLOv3 is a one-stage object detector and stands for You Only Look Once version 3 is a state-of-the-art open source real time object detector improved and developed by Joseph Redmon from YOLOv1. YOLOv3 is able to process images in real time at 30 Frames Per Second (FPS) with a mean Accurate Precision (mAP) of 57.9% on coco test-dev when using a graphic processing unit (GPU), NVIDIA Pascal Titan X. YOLOv3 predicts bounding boxes using dimension clusters as anchor box and with the use of dimension clusters along with directly predicting the bounding box centroid location will improve YOLOv3 accuracy by almost 5%. [11]

A. Network Architecture

YOLOv3 deep network architecture is Darknet-53 that consists of 106 fully convolutional layers (FCN) which can be seen in Figure 5. The deep network architecture originally has 53 layers and in order to perform detection, another 53 layers will be stacked onto it which results in a total of 106 layers. [12]

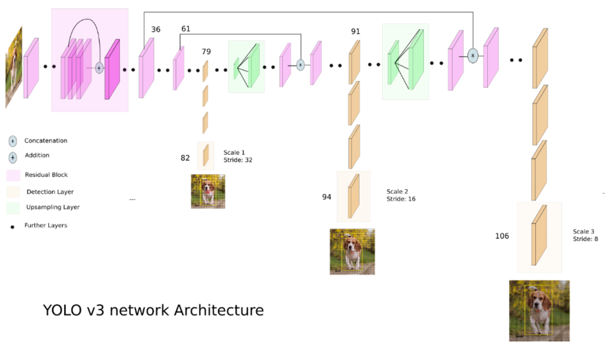


Fig. 1. Network Architecture of YOLOv3 [12]

Predictions for YOLOv3 at 3 scales and this is done by down sampling the input image by 32,16 and lastly 8. The first detection is done at the first scale in the 82nd convolutional layer, second detection is done at the second scale in the 94th convolutional layer and the third detection is done at the third scale in the 106th convolutional layer which can be seen in Fig.1 [12]. The first detection is done by the 82nd layer of the deep network architecture where the input image of size 416 x 416 is down sampled by the network with a stride of 32. This will result in a feature map of size 13 x 13 in the 82nd layer which can be seen in Fig.2. A detection is done by using a 1 x 1 detection kernel which will result in a detection feature map of 13 x 13 x 255.

The second detection is done by the 94th layer of the deep network architecture with an input image size of 416 x 416 that is down sampled by the network with a stride of 16. This will result in a feature map of size 26 x 26 in

Daniel Tan Wei Xun was at Singapore Institute of Technology. E-mail: poweruserd@live.com

Yoke Lin Lim was at Singapore Institute of Technology. E-mail: yokelin_space@hotmail.com

Sutthiphong Srigrarom is at National University of Singapore. E-mail: spot.srigrarom@nus.edu.sg

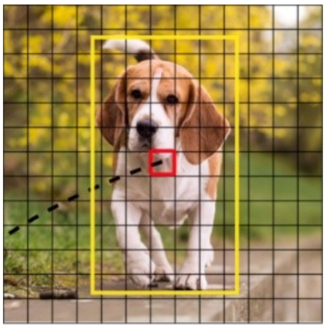


Fig. 2. Feature Map Size of 13 x 13 in the 82nd Layer [12]

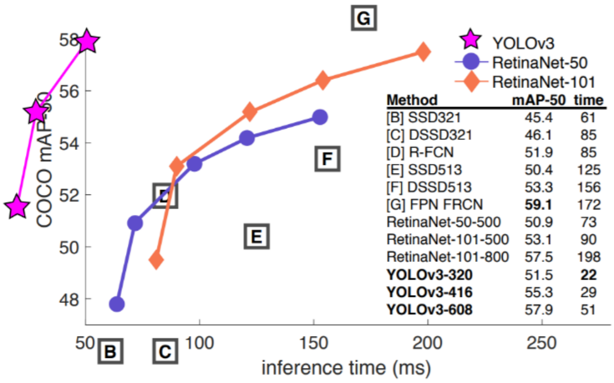


Fig. 3. Comparison of YOLOv3 with various object detectors [13]

the 94th layer. This will result in a detection feature map of 26 x 26 x 255 when using a 1 x 1 detection kernel.

The third detection is done by the 106th layer of the deep network architecture with an input image size of 416 x 416 that will be down sampled by the network with a stride of 8. By doing so, this will result in a feature map of size 52 x 52 in the 106th layer. The detection feature map will then have a size on 52 x 52 x 255 with a 1 x 1 detection kernel.

YOLOv3 object detector will be used as it is able to detect small objects fast, accurate and most importantly in real time. This is due to the detection at 3 different scales where at scale 1 the 13 x 13 detection feature map as shown in Fig.2 will be used to detect large size objects, the 26 x 26 detection feature map at scale 2 will be used to detect medium size objects and lastly the 52 x 52 detection feature map will be used to detect small objects. At the 52 x 52 detection feature map, there will be a total of 2704 cells in an image where each cell is able to generate 5 bounding boxes. This means that at the 106th convolutional layer that is used for detection of small objects, 13520 bounding boxes can be generated in a single image for predictions and is able to detect small object which is a necessity for detection of small drones or for drones which are a distance away.

The selection of YOLOv3 is also due to its fast inference time that does not compromise its accuracy when compared with various deep learning object detectors such as SSD, DSSD, R-FCN, FPN and Retinanet which can be seen in Fig.3. [13] When comparing YOLOv3 with R-CNN and Faster R-CNN, YOLOv3 performance is 1000 times faster than R-CNN and 100 times faster than Faster R-CNN. [13] This shows that YOLOv3 has the quickest inference time of 50 milliseconds with a mAP of 58, therefore YOLOv3 is the object detector with the best performance when comparing with the various detectors as shown in Fig.3.

B. YOLOv3 Detection of Drones

The pre-trained weights in YOLOv3 of multiple different classes did not include a class to detect drones. Therefore the machine learning is needed to train this deep learning object detector to specifically detect drone and to share the results which is the objective of this research paper.

C. Dataset Collection

Images of drones, hexacopters, quadcopters and unmanned aerial vehicles (UAV) have been collected. To further enhance the dataset, several experiments were done to collect drone images using camera and 360 camera at during different timings in the day to simulate sunny and cloudy. Drones images were also captured at different altitudes from 10m to 50m with an increment of 10m as we require to capture small drones. A total of 1500 images drone images were manually sorted to remove irrelevant images and 1435 images were prepared. This is to ensure the precision of dataset that was prepared.

D. Image Annotation and Image Augmentation

Image annotation aims to assign a single class label to the object that is contained in the image. [14] As the objective is to detect drones, only 1 class will be set. Image annotations were done for every image in the drone datasets which consists of 1435 images by using image labelling tools. The choice of bounding boxes will be rectangle boxes as YOLOv3 prediction bounding boxes are rectangle.

Data augmentation is a technique that is widely used in machine learning tasks such as classification of images. Data Augmentation was used to further enhance the size of the training dataset and avoids overfitting by creating new samples of the original dataset. [15]

Data augmentation was done to the original image by vertically flip the image, horizontally flip the image, adding noise to the image and rotating the image which can be seen in. The original dataset of 1435 images have been enhanced to 7175 images after data augmentation.

III. Training and Validation of Dataset

The selection of training and validation parameters of a model is an essential step in machine learning to ensure optimal performance without overfitting. This is done by dividing the dataset into 2 sets: training and validation. [16]

The selection of training and validation parameters were done and the probability is set to 0.8. A probability of 0.8 means that 80% of the images will be tagged for training and the remaining 20% of the images will be tagged for validation. The reason for choosing a probability of 0.8 is due to Pareto principal which is known as the 80/20 rule which states that roughly 80% of the effects will come from 20% of the causes. [17]

The dataset of 7175 images were divided into training and validation where 19.5% of the dataset is tagged as validation and 80.5% of the remaining is tagged as training.

IV. Training of Custom Dataset with Neural Network YOLOv3

The training of neural network YOLOv3 with custom dataset to specifically detect drones will be done by deployment on Amazon EC2. The following hyperparameters such as learning rate, epoch, batch size, input size and the type of weights initiation that will used in to configuration to train the deep learning neural network YOLOv3 will be discussed below. Deep learning neural network are trained using stochastic gradient descent algorithm and stochastic gradient descent is an algorithm that performs backpropagation. [18] Learning rate is a configurable hyperparameter that is used in the training of deep learning neural network. The learning rate controls the speed or the rate where the model learns, and it has a small positive value that ranges from 0 to 1.0. A large learning rate allows model to learn faster which decreases the training time, but this will result in an increase in average loss. A small learning rate will result in the model learning slowly which increases the training time and it may result in the training to be permanently stuck with a high training error. [19] Therefore, an optimal learning rate should be used where the learning rate is not too large or too small. The learning rate that will be used for training of custom dataset with neural network YOLOv3 will be 0.0001. Epoch is a configurable hyperparameter where 1 epoch means an entire dataset passing forward and backward through the neural network once. Passing an entire dataset forward and backward through the neural network once is insufficient as it would lead to overfitting and if the number of epochs is too large, this will result in underfitting. An optimal epoch has to be used to prevent overfitting and underfitting. The epoch that will be used for training of custom dataset with neural network YOLOv3 will be 50 where 1 epoch would mean that 7175 images in the dataset will be passed forward and backward through the neural network. It is impossible to feed a dataset of images to a neural network as it is too huge. Batch size are then introduced which divides the dataset equally into many batches. This will allow the feeding of smaller datasets in batches to the neural network. The batch size that is used for training will depend on the GPU memory, a GPU with a higher memory will allow a larger batch size to be used for training. The batch size that will be used for training of custom dataset with neural network YOLOv3 will be 12. With a dataset of 7175 images, this means that it would take 598 iterations to complete one epoch and 29900 iterations for 50 epochs. The types of weights initiation that will be used for training will be transfer learning from YOLOv3 COCO model. Transfer learning is the improvement of learning in a new task by transferring the existing knowledge that had been already learned. [20] This is done by training the last few layers of the convolutional neural network in YOLOv3 to specifically detect drones.

V. Training and Detection Results

A. Training Results

Training results after machine learning for YOLOv3 to specifically detect drones must be analysed. This can be seen in Figure 8 which shows the epoch vs loss from 1 to 50 epochs during training.

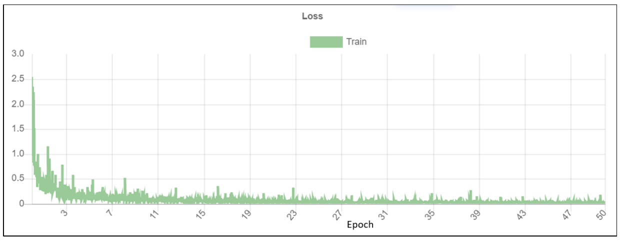


Fig. 4. Overall Training Results for 0-50 epochs

The objective will be to export YOLOv3 custom trained weights from the epoch with the least amount of losses to ensure the accuracy of the detection results. An epoch is considered as a checkpoint and the checkpoint that was exported as YOLOv3 custom trained weights will be checkpoint 33 that can be seen in Fig.5 which shows the least amount of loss of 0.009536. Checkpoint 33 has the least amount of losses after analysing and comparing with all 50 different checkpoints.

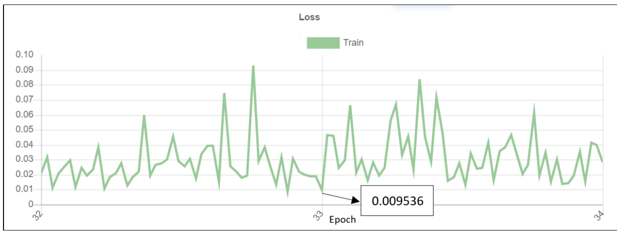


Fig. 5. Checkpoint 33 which has the least loss of 0.009536

B. YOLOv3 Custom Drone Detection Results

YOLOv3 custom trained weights to specifically detect drones after machine learning were sucessfully deployed in YOLOv3 to perform detections on an image and a video showing drones which can be seen in Fig.6 and Fig.7

The custom trained weights were also successfully deployed in real time with a webcam and in real time with an external spherical camera (Ricoh Theta S) which can be seen in Fig.8 and Fig.9 .

Fig. 6. Detection of drone in an image

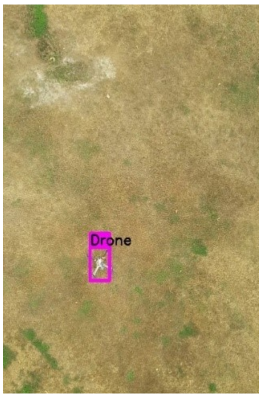


Fig. 7. Detection of drone in a video

Fig. 8. Detection of drone with machine webcam in real time

Fig. 9. Detection of drone with an external spherical camera Ricoh Theta S in real time

1) Results of YOLOv3 Drone Detection on Cloudy Environment and various Altitudes: YOLOv3 with custom trained weights was successful in detecting drones on the images that was taken during the simulation of different altitude and cloudy conditions which can be seen in Figs. 10,11,12,13. The accuracy results and the coordinates of the detected drones can be seen in Figs.14,15,16,17 respectively.

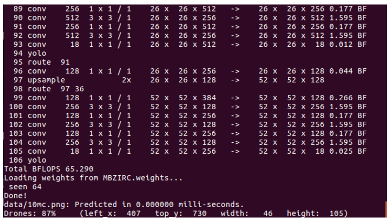
Fig. 10. Detection of drone at 10m height in a cloudy environment

Fig. 11. Detection of drone at 20m height in a cloudy environment

Fig. 12. Detection of drone at 30m height in a cloudy environment

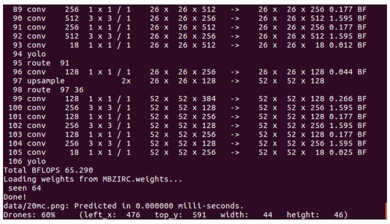
2) Results of YOLOv3 Drone Detection on Sunny Environment and various Altitudes: YOLOv3 with custom trained weights was successful in detecting drones on the

Fig. 13. Detection of drone at 40m height in a cloudy environment



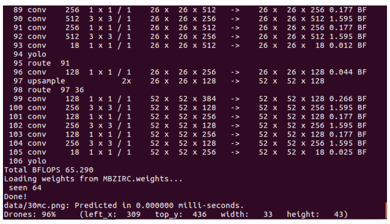
```
 89 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
 90 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
 91 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
 92 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
 93 conv     18  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x  18 0.012 BF
 94 yolo
 95 route  91
 96 conv    128  1 x 1 / 1    26 x  26 x 256   ->    26 x  26 x 128 0.044 BF
 97 upsample            2x    26 x  26 x 128   ->    52 x  52 x 128
 98 route  97 36
 99 conv    128  1 x 1 / 1    52 x  52 x 384   ->    52 x  52 x 128 0.266 BF
100 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
101 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
102 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
103 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
104 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
105 conv     18  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x  18 0.025 BF
106 yolo
Total BFLOPS 65.290
Loading weights from MBZIRC.weights...
 seen 64
Done!
data/10mc.png: Predicted in 0.000000 milli-seconds.
Drones: 87%     (left_x:  407   top_y:  730   width:   46   height:  105)
```

Fig. 14. Accuracy and coordinates of detected drone at 10m height in a cloudy environment



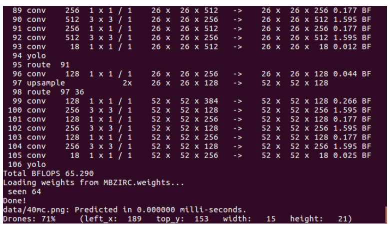
```
 89 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
 90 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
 91 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
 92 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
 93 conv     18  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x  18 0.012 BF
 94 yolo
 95 route  91
 96 conv    128  1 x 1 / 1    26 x  26 x 256   ->    26 x  26 x 128 0.044 BF
 97 upsample            2x    26 x  26 x 128   ->    52 x  52 x 128
 98 route  97 36
 99 conv    128  1 x 1 / 1    52 x  52 x 384   ->    52 x  52 x 128 0.266 BF
100 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
101 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
102 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
103 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
104 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
105 conv     18  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x  18 0.025 BF
106 yolo
Total BFLOPS 65.290
Loading weights from MBZIRC.weights...
 seen 64
Done!
data/20mc.png: Predicted in 0.000000 milli-seconds.
Drones: 60%     (left_x:  476   top_y:  591   width:   44   height:   46)
```

Fig. 15. Accuracy and coordinates of detected drone at 20m height in a cloudy environment



```
 89 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
 90 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
 91 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
 92 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
 93 conv     18  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x  18 0.012 BF
 94 yolo
 95 route  91
 96 conv    128  1 x 1 / 1    26 x  26 x 256   ->    26 x  26 x 128 0.044 BF
 97 upsample            2x    26 x  26 x 128   ->    52 x  52 x 128
 98 route  97 36
 99 conv    128  1 x 1 / 1    52 x  52 x 384   ->    52 x  52 x 128 0.266 BF
100 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
101 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
102 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
103 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
104 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
105 conv     18  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x  18 0.025 BF
106 yolo
Total BFLOPS 65.290
Loading weights from MBZIRC.weights...
 seen 64
Done!
data/30mc.png: Predicted in 0.000000 milli-seconds.
Drones: 96%     (left_x:  309   top_y:  436   width:   33   height:   43)
```

Fig. 16. Accuracy and coordinates of detected drone at 30m height in a cloudy environment



```
 89 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
 90 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
 91 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
 92 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
 93 conv     18  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x  18 0.012 BF
 94 yolo
 95 route  91
 96 conv    128  1 x 1 / 1    26 x  26 x 256   ->    26 x  26 x 128 0.044 BF
 97 upsample            2x    26 x  26 x 128   ->    52 x  52 x 128
 98 route  97 36
 99 conv    128  1 x 1 / 1    52 x  52 x 384   ->    52 x  52 x 128 0.266 BF
100 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
101 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
102 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
103 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
104 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
105 conv     18  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x  18 0.025 BF
106 yolo
Total BFLOPS 65.290
Loading weights from MBZIRC.weights...
 seen 64
Done!
data/40mc.png: Predicted in 0.000000 milli-seconds.
Drones: 71%     (left_x:  189   top_y:  153   width:   15   height:   21)
```

Fig. 17. Accuracy and coordinates of detected drone at 40m height in a cloudy environment

Fig. 18. Detection of drone at 10m height in a sunny environment

Fig. 19. Detection of drone at 20m height in a sunny environment

images that was taken during in a sunny condition which can be seen in Figs. 18,19,20,21. The accuracy and the coordinates of the detected drones can be seen in Figs. 22,23,24,25.

YOLOv3 with trained weights to specifically detect drones were subsequently deployed on NVIDIA Jetson TX2 and proven successful. However, there was limitation due to the lower processing capability of the Jetson TX2 which resulted in a very low fps even after overclocking the Jetson TX2 module. The demonstrations can be seen in these 2 Youtube links (consecutive clips): *https://youtu.be/ylaIAil4qDw* and *https://youtu.be/vsP7NoUK61I* .

VI. Conclusion

This paper presents the implementation of an object detector to detect drones. With transfer learning to train

Fig. 20. Detection of drone at 30m height in a sunny environment

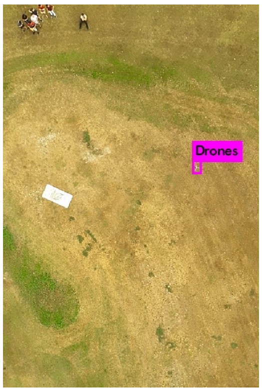
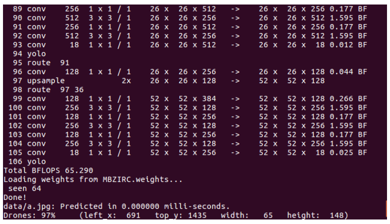


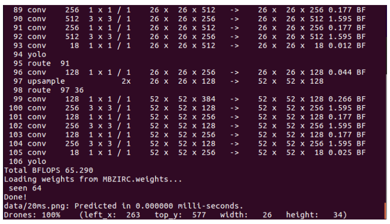
Fig. 21. Detection of drone at 40m height in a sunny environment



```
  89 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
  90 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
  91 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
  92 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
  93 conv     18  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x  18 0.012 BF
  94 yolo
  95 route  91
  96 conv    128  1 x 1 / 1    26 x  26 x 256   ->    26 x  26 x 128 0.044 BF
  97 upsample            2x    26 x  26 x 128   ->    52 x  52 x 128
  98 route  97 36
  99 conv    128  1 x 1 / 1    52 x  52 x 384   ->    52 x  52 x 128 0.266 BF
 100 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
 101 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
 102 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
 103 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
 104 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
 105 conv     18  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x  18 0.025 BF
 106 yolo
Total BFLOPS 65.290
Loading weights from MBZIRC.weights...
 seen 64
Done!
data/a.jpg: Predicted in 0.000000 milli-seconds.
Drones: 97%     (left_x:  691   top_y: 1435   width:   65   height:  148)
```

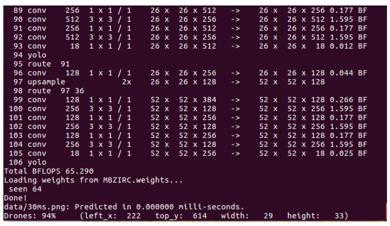
Fig. 22. Accuracy and coordinates of detected drone at 10m height in a sunny environment

the deep learning detector to specifically detect drones, the modified YOLOv3 was able to successfully detect large, medium and small drones. The results from using YOLOv3 custom detector shows that it is able to accurately detect drones at a confidence level between 60% to 100% and with an average confidence level of 88.9% with a dataset of 7175 images.



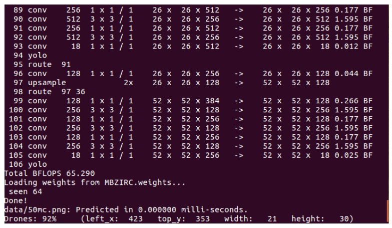
```
  89 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
  90 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
  91 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
  92 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
  93 conv     18  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x  18 0.012 BF
  94 yolo
  95 route  91
  96 conv    128  1 x 1 / 1    26 x  26 x 256   ->    26 x  26 x 128 0.044 BF
  97 upsample            2x    26 x  26 x 128   ->    52 x  52 x 128
  98 route  97 36
  99 conv    128  1 x 1 / 1    52 x  52 x 384   ->    52 x  52 x 128 0.266 BF
 100 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
 101 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
 102 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
 103 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
 104 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
 105 conv     18  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x  18 0.025 BF
 106 yolo
Total BFLOPS 65.290
Loading weights from MBZIRC.weights...
 seen 64
Done!
data/20ms.png: Predicted in 0.000000 milli-seconds.
Drones: 100%    (left_x:  263   top_y:  577   width:   26   height:   34)
```

Fig. 23. Accuracy and coordinates of detected drone at 20m height in a sunny environment



```
  89 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
  90 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
  91 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
  92 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
  93 conv     18  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x  18 0.012 BF
  94 yolo
  95 route  91
  96 conv    128  1 x 1 / 1    26 x  26 x 256   ->    26 x  26 x 128 0.044 BF
  97 upsample            2x    26 x  26 x 128   ->    52 x  52 x 128
  98 route  97 36
  99 conv    128  1 x 1 / 1    52 x  52 x 384   ->    52 x  52 x 128 0.266 BF
 100 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
 101 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
 102 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
 103 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
 104 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
 105 conv     18  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x  18 0.025 BF
 106 yolo
Total BFLOPS 65.290
Loading weights from MBZIRC.weights...
 seen 64
Done!
data/30ms.png: Predicted in 0.000000 milli-seconds.
Drones: 94%     (left_x:  222   top_y:  614   width:   29   height:   33)
```

Fig. 24. Accuracy and coordinates of detected drone at 30m height in a sunny environment



```
  89 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
  90 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
  91 conv    256  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x 256 0.177 BF
  92 conv    512  3 x 3 / 1    26 x  26 x 256   ->    26 x  26 x 512 1.595 BF
  93 conv     18  1 x 1 / 1    26 x  26 x 512   ->    26 x  26 x  18 0.012 BF
  94 yolo
  95 route  91
  96 conv    128  1 x 1 / 1    26 x  26 x 256   ->    26 x  26 x 128 0.044 BF
  97 upsample            2x    26 x  26 x 128   ->    52 x  52 x 128
  98 route  97 36
  99 conv    128  1 x 1 / 1    52 x  52 x 384   ->    52 x  52 x 128 0.266 BF
 100 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
 101 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
 102 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
 103 conv    128  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x 128 0.177 BF
 104 conv    256  3 x 3 / 1    52 x  52 x 128   ->    52 x  52 x 256 1.595 BF
 105 conv     18  1 x 1 / 1    52 x  52 x 256   ->    52 x  52 x  18 0.025 BF
 106 yolo
Total BFLOPS 65.290
Loading weights from MBZIRC.weights...
 seen 64
Done!
data/50mc.png: Predicted in 0.000000 milli-seconds.
Drones: 92%     (left_x:  423   top_y:  353   width:   21   height:   30)
```

Fig. 25. Accuracy and coordinates of detected drone at 40m height in a sunny environment

REFERENCES

[1] "38 Ways Drones Will Impact Society: From Fighting War To Forecasting Weather, UAVs Change Everything," CBINSIGHTS, 24 January 2019. [Online]. Available: https://www.cbinsights.com/research/drone-impact-society-uav/. [Accessed 6 April 2020].

[2] R. Spinks, "UK police still have no proof of the drone attack that grounded 1,000 Gatwick flights," QUARTZ, 31 December 2018. [Online]. Available: https://qz.com/1511667/the-gatwick-drone-incident-is-still-a-mystery-to-british-police/.

[3] S. Agarwal, J. O. D. Terrail and F. Jurie, "Recent Advances in Object Detection in the Age of Deep Convolutional Neural Networks," hal-01869779, pp. 1-107, 2018.

[4] S. Gupta and Y. J. Singh, "Object Detection using Shape Features," ResearchGate, pp. 2-5, 2014.

[5] Y. Yoshimura, B. Cai, Z. Wang and C. Ratti, "Deep Learning Architect: Classification for Architectural Design through the Eye of Artificial Intelligence," arXiv, pp. 1-22, 2018.

[6] Z.-Q. Zhao, P. Zheng, S.-t. Xu and X. Wu, "Object Detection with Deep Learning: A Review," arXiv, pp. 1-21, 2018.

[7] P. Doshi, S. Punktambekar, N. Kini and S. S. Dhami, "Object Detection and its Application using Convolutional Neural Network," Academia, pp. 1-4.

[8] R. Girshick, J. Donahue, T. Darrell, J. Malik and U. Berkeley, "Rich feature hierarchies for accurate object detection and semantic segmentation," arXiv, pp. 1-21, 2014.

[9] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," arXiv, pp. 1-14, 2016.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," arXiv, pp. 1-17, 2016.

[11] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," CVPR , pp. 1-9, 2016.

[12] A. Kathuria, "What's new in YOLO v3?," Towards Data Science, 23 April 2018. [Online]. Available: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b. [Accessed 5 April 2019].

[13] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv, pp. 1-6, 2018.

[14] Y. Niu, Z. Lu, J.-R. Wen, T. Xiang and S.-F. Chang, "Multi-Modal Multi-Scale Deep Learning for Large-Scale Image Annotation," arViv, pp. 1-11, 2018.

[15] H. Inoue, "Data Augmentation by Pairing Samples for Images Classification," arXiv, pp. 1-8, 2018.

[16] G. Tennenholtz, T. Zahavy and S. Mannor, "TRAIN ON VALIDATION: SQUEEZING THE LEMON," arXiv, pp. 1-17, 2018.

[17] P. Gulipalli, "The Pareto Principle for Data for Data Scientists," KDnuggets, 2019. [Online]. Available: https://www.kdnuggets.com/2019/03/pareto-principle-data-scientists.html. [Accessed 8 April 2019].

[18] L. Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent," in Proceedings of COMPSTAT'2010, NEC Labs America, Princeton NJ 08542, USA, 2010.

[19] Y. Bengio, "Practical Recommendations for Gradient-Based Training of Deep Architectures," arXiv, pp. 1-33, 2012.

[20] L. Torrey and J. Shavlik, "Transfer Learning," IGI Global, pp. 1-22, 2009.