Drone Detection in Long-range Surveillance Videos

Mrunalini Nalamati
University of Technology Sydney
Broadway, Ultimo, NSW 2007, Australia
nalamati@student.uts.edu.au

Ankit Kapoor
University of Technology Sydney
Broadway, Ultimo, NSW 2007, Australia
ankit.kapoor@uts.edu.au

Muhammed Saqib
University of Technology Sydney
Broadway, Ultimo, NSW 2007, Australia
muhammed.saqib@uts.edu.au

Nabin Sharma
University of Technology Sydney
Broadway, Ultimo, NSW 2007, Australia
nabin.sharma@uts.edu.au

Michael Blumenstein
University of Technology Sydney
Broadway, Ultimo, NSW 2007, Australia
michael.blumenstein@uts.edu.au

Abstract

The usage of small drones/UAVs has significantly increased recently. Consequently, there is a rising potential of small drones being misused for illegal activities such as terrorism, smuggling of drugs, etc. posing high-security risks. Hence, tracking and surveillance of drones are essential to prevent security breaches. The similarity in the appearance of small drone and birds in complex background makes it challenging to detect drones in surveillance videos. This paper addresses the challenge of detecting small drones in surveillance videos using popular and advanced deep learning-based object detection methods. Different CNN-based architectures such as ResNet-101 and Inception with Faster-RCNN, as well as Single Shot Detector (SSD) model was used for experiments. Due to sparse data available for experiments, pre-trained models were used while training the CNNs using transfer learning. Best results were obtained from experiments using Faster-RCNN with the base architecture of ResNet-101. Experimental analysis on different CNN architectures is presented in the paper, along with the visual analysis of the test dataset.

Keywords— Drone detection, Deep learning, Faster R-CNN

1. Introduction

Automatic visual detection of small objects from amongst similar-looking objects is challenging. That is more so when the detection needs to be done from videos recorded at very long-range with low-contrast and with the small objects practically blending into the background. In the training dataset used, sometimes the small drones were hard to spot with naked eyes. Detection of small objects, such as small drones, is crucial though, due to it's rising use for illegal activities.

Advances in object detection techniques are leading to faster, more accurate results. The most recent techniques using Convolutional Neural Networks (CNN) typically involve multiple steps. Firstly, finding the regions of interest in the image, then, running these through the CNN for feature extraction and after this, classifying them using supervised classification algorithms. Finally, combining the results across the regions to correctly mark the bounding box of the object.

In this study, we have carried out experiments for object detection using the latest deep Convolutional Neural Network algorithms in an attempt to find the combination most suitable for small object detection. The challenge goal is to detect a drone appearing at some time in a short video sequence where birds are also present. The dataset is challenging due to it's long-range, blending background, varying illumination, and particularly the small size of the drones. The dataset becomes more complex at places when it has more than one drone and birds in the same frame. In some frames, the moon also appears in the night sky, posing another ambiguous object for the algorithm.

In this paper, we present our experiments with various deep learning-based object detection techniques. We have prepared data by first extracting each frame from the videos and then running various deep learning models on the training dataset. These models are frozen after a certain number of iterations. Bounding boxes are then predicted by using these frozen models on the validation dataset. IoU and mAP are measured between these predicted bounding boxes and the ground-truth. Images such as shown in Figure1 proved to be especially difficult due to the presence of

Figure 1: Sample Images from training dataset

birds as well as other drones in the same frame. We have focussed on getting the best results, especially for this kind of images.

This paper is organized as follows: section 2, *Literature Review* presents a survey of the latest deep learning based object detection techniques. Section 3, *Proposed Methodology* presents the details of the proposed model. Section 4, *Experimental Results* presents the characteristics and preparation of the dataset, details the performance on training and testing datasets and finally visualizes the results. Finally, Section 5 covers our findings and suggests future work.

2. Literature Review

Modern object detection techniques based on Deep Convolution Neural Network are described as two-stage and one-stage detectors. Two-stage detectors are Region-based CNNs, and One-stage detectors are Single Shot Detectors (SSD). There are other hybrid models as well, that address the shortcomings of these techniques. Sections below provide a brief overview of these techniques.

2.1. Region-Based CNNs

Faster R-CNN [12] is the fastest Region Proposal based framework. It acts on the whole image as such and is faster than it's predecessors, the R-CNN [4] and Fast R-CNN [3]. The two stages in the pipeline are region proposals and classification. Here the entire image is passed through a convolutional neural network generating a feature map. Region Proposal Network, which is another CNN, is then applied to this feature map to generate object proposals and objectness score. The proposals and score are generated by applying scale varying anchor boxes centered at each pixel location of the feature map. Region-of-Interest pooling is then applied to these to bring object proposals in the same region to the same size. Finally, these object proposals are passed through the FC layer, having softmax and linear regression to classify the objects and generate bounding boxes.

This technique has shown reliable, accurate results [12]. Figure 2 illustrates the pipeline.

2.2. Single Shot Detectors

The region-based proposal models involve sequential processing. Hence, the overall performance of the system is impacted by the lowest-performing layer [12]. Hence, a different type of fast object detection models, basically SSDs are becoming popular.

Single Shot Detectors tend to act on the entire image at once and detect objects in one pass. Models such as Single Shot Multibox Detectors, RetinaNet, and YOLO family fall under this type.

SSD at it's core uses the CNN's pyramidal feature hierarchy. VGG-16 [14] model is used for feature extraction. However, the FC layers in this network are discarded and instead use auxiliary convolution layers. A pre-trained model on ImageNet [6] is used as a start. Anchor boxes are predicted at each layer, which is responsible for objects at that scale only. Sum of localization loss and classification loss is used as the loss function. Softmax is used as the classifier. In SSD, object localization and classification are done in a single forward pass of the network. SSD is good at detecting objects of different sizes in the image. [9]

However, the performance of SSD does not match up to that of Faster R-CNN. In [7], the authors have suggested that this is because of extreme foreground-background class imbalance found during the training of dense detectors. The Focal Loss method, proposed in [7], addresses this issue by training on a set of hard examples. More weight is assigned to images where objects can be easily wrongly classified, and less weight is assigned to more obvious easy objects. This is called RetinaNet. Like in SSD, featurized image pyramid provides the backbone network. It is constructed on top of the ResNet architecture. It is seen that the RetinaNet is able to achieve the speed of SSD, and also maintain the accuracy of Region-based detectors.

In YOLO the image is divided into SxS grids. For each grid, B bounding boxes are predicted, with confidence for the boxes and C class probabilities. A single convolutional network is applied to the entire image, only once, to predict multiple objects and locations, in terms of the bounding boxes. This approach is different from the R-CNN based models, which require thousands of network evaluations of the same image. This is what makes YOLO faster than R-CNN based models. This model processes images at 45 frames per second [10]

The basic YOLO model performance is improved further in YOLOv3. Three different scales are used for feature extraction. Darknet-53 is the underlying architecture. The multi-layer scaling allows predictions at three different scales. The last layer predicts the bounding box with the objectness score and class. It uses logistic regression for objectness score. Independent logistic classifiers for each class are used instead of one softmax. Overall YOLOv3 performs better than SSD and similar to ResNet-152 [5], but is twice as fast. [11]

SSD models are faster than R-CNN. Hence, this makes them more suitable for real-time applications. Region-based, two-stage models, on the other hand, are much more accurate but require more memory and power to run. Hence, they could find better applications for offline object detections, where reliability is more important than speed. However, given the speed of present-day GPUs, the factor of memory and speed could likely be overcome to some extent.

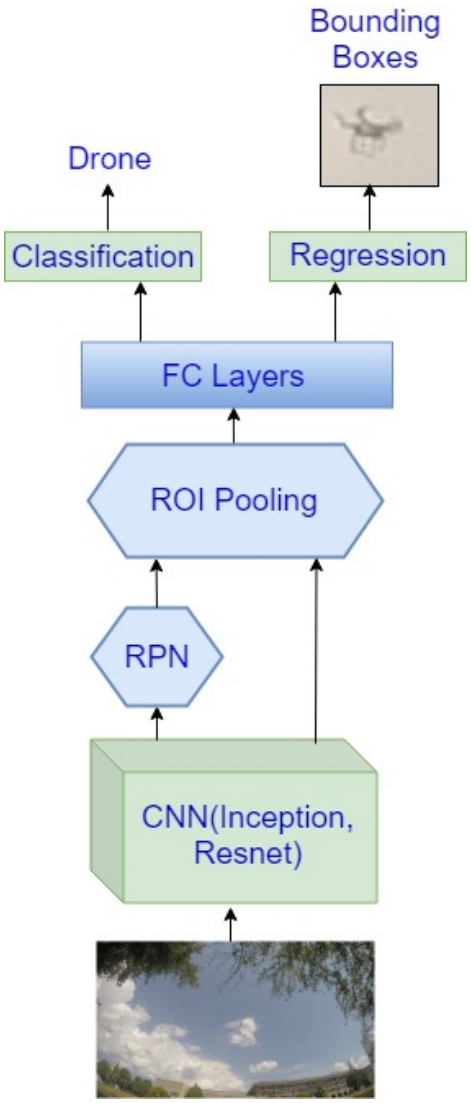


Figure 2: Faster R-CNN pipeline

3. Proposed Methodology

This study was done with the region-based detectors, Faster R-CNN[12] with base architectures of Inception v2[15] and Resnet-101[5] and SSD[9] with Inception v2. To speed up convergence, we started training with transfer learning from publicly available pre-trained COCO[8] models. We have experimented with different types of network architectures such as Inception v2[15], ResNet-101[5]. We used TensorFlow libraries for these networks. In [13], the authors have observed that VGG16 performed best on a similar dataset. We have chosen ResNet-101 since the dataset has the drone objects at very different scales. These drones appear miniature when the drone is far away from the camera and significantly big when it is closer to the camera. Figure 3 shows the proposed network architecture.

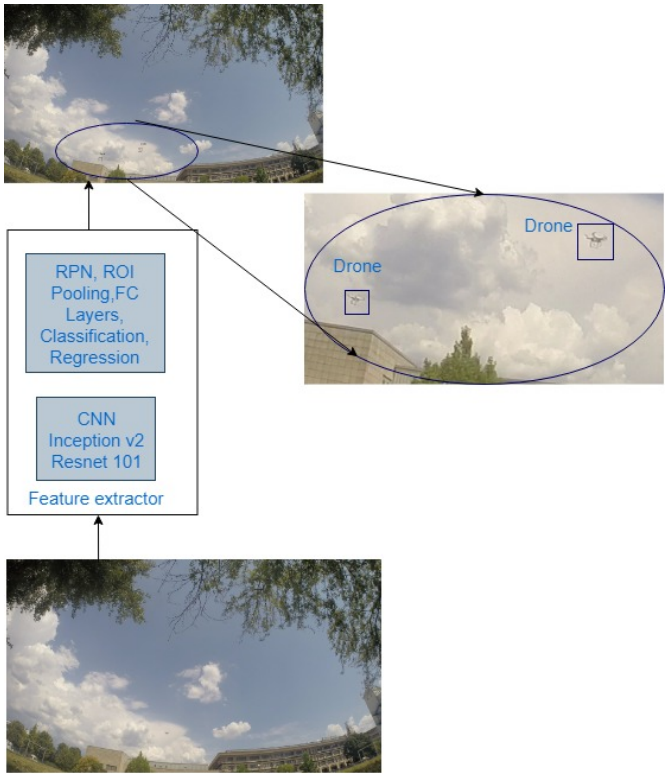


Figure 3: Proposed pipeline

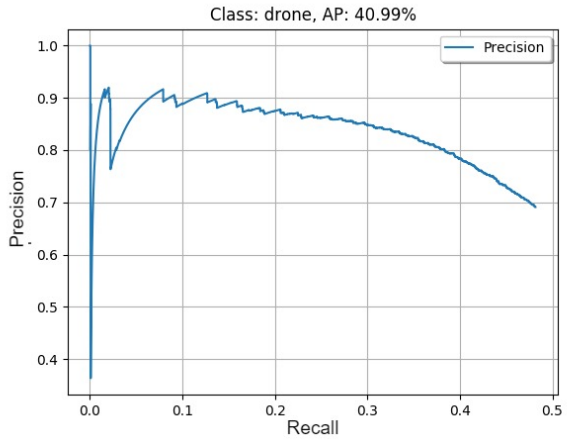
4. Experimental Results

In the following sections, we present the details about the data used, preparation of the data, object detection models used and the results we observed.

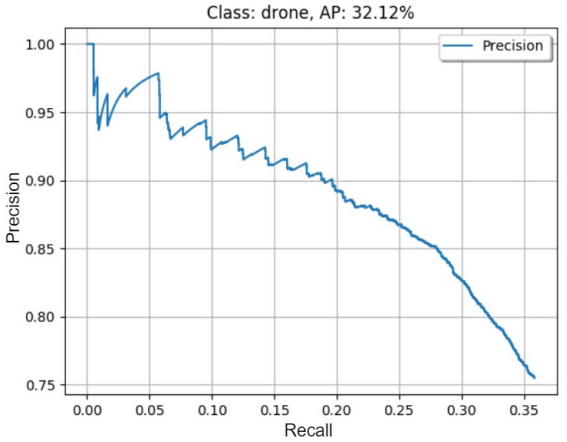
4.1. Data Preparation

The dataset for this competition is provided by the Bird-Vs-Drone project[1]. It is a collection of *11 MPEG4-coded* videos. Annotations are provided in one XML file per video. The bounding boxes for drones are provided as (framespan,y,x,width,height), framespan indicating the frame number in which the drone is present, y,x is the top left corner location of the bounding box with the given width and height. The drone can be seen in almost every frame of the video. Some videos are taken at night, with the moon in sight. Some are on very cloudy days, with bright white clouds in the background. All the videos are long-range, and the drone or birds appear very small. In many instances, the drone is difficult to spot even with the naked eye.

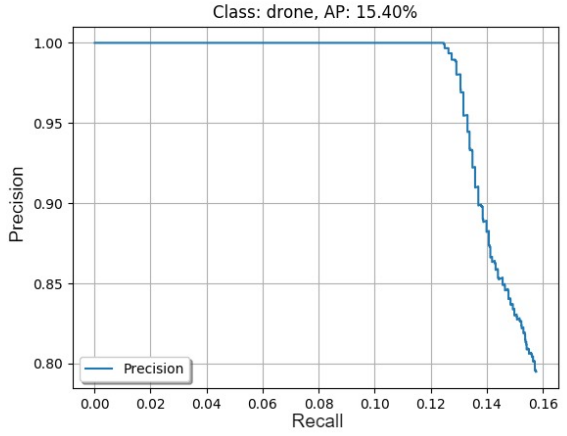
From these videos, a total of *8771* frames were extracted in JPG format. The ground-truth annotations were separated from one XML file per video to one per frame and converted to (topx topy width height) format for each bounding-box of the drone. At times there are more than one drone objects in each frame, and thus in the XML file. A sample of the frames from training dataset is shown in Figure 1. The converted XML files are in PASCAL VOC [2] format. These JPG and XML files were then converted



(a) Faster R-CNN with ResNet-101



(b) Faster R-CNN with InceptionV2



(c) SSD

Figure 4: Precision-Recall Graphs

to TF record format for the TensorFlow object detection library. The training was done against 70% of the dataset, 20% was used as a validation set, and testing was done initially against the remaining 10%. TF record files for each of training, validation, and testing datasets were created separately.

4.2. Training Phase

The training was done on 2 X Nvidia Quadro P6000 GPUs with a learning rate of *0.0001*. For Region-based proposal networks, a batch size of 32 is used. Performance at several points in the training iterations is observed for the different network architectures. The training models are frozen at iteration 90,000. Intersection over Union (IoU) is calculated for the detected bounding boxes against the ground-truth bounding boxes. The ones with IoU greater than 0.4 are considered as True Positives(TP). The ones with IoU below *0.4*, are False Positives(FP), which are incorrectly detected objects not actually present. The objects present, which were not detected are False Negatives(FN), and the ones which are correctly not detected are the True Negatives(TN). The Precision and Recall are calculated as per equations 1 and 2 at multiple points during training.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Precision-Recall curve is plotted, as seen in Figure 4. Average Precision(AP) is used as a popular metric to determine the accuracy of detection of each class of objects, e.g. Birds, Drones. mean Average Precision (mAP) is the mean of the AP across the classes. In our case, since we are detecting only one class, the mAP would be the same as AP.

4.3. Results Discussion

The plotted graphs in Figure 4 show the positive predictive value against the true positive rate. In a perfect graph, the Precision-Recall(PR) curve would be closer to the upper right-hand corner and slowly bow towards (1.0, 1.0). We observe that the PR curve for Faster R-CNN with ResNet-101 gives us the best result. The Average Precision (AP) is 40.99%, and the curve stays towards the high end of precision until 0.5 recall. Whereas, the PR curve for Faster R-CNN with Inceptionv2 starts falling from the very start and ultimately goes to 0 precision at recall close to 0.35. The worst performance is seen for SSD. The PR curve starts well but suddenly falls to 0 precision at a very early recall value close to 0.15. Though the SSD algorithm was the fastest to execute amongst the three, we see that it performs very poorly. The comparison of results across these three models can be seen in Table 1. Figure 5 shows the Classification and Localization losses for Faster R-CNN with Resnet-101. It shows that the loss is fairly controlled and mostly below 0.2. The least loss is found in the range of 30k to 70k iterations. Hence we have used these in our models.

4.4. Visual Analysis of Test Results

Figures 6, 7 and 8 show the detected bounding boxes on two images from the validation set. The image on the left has the drone

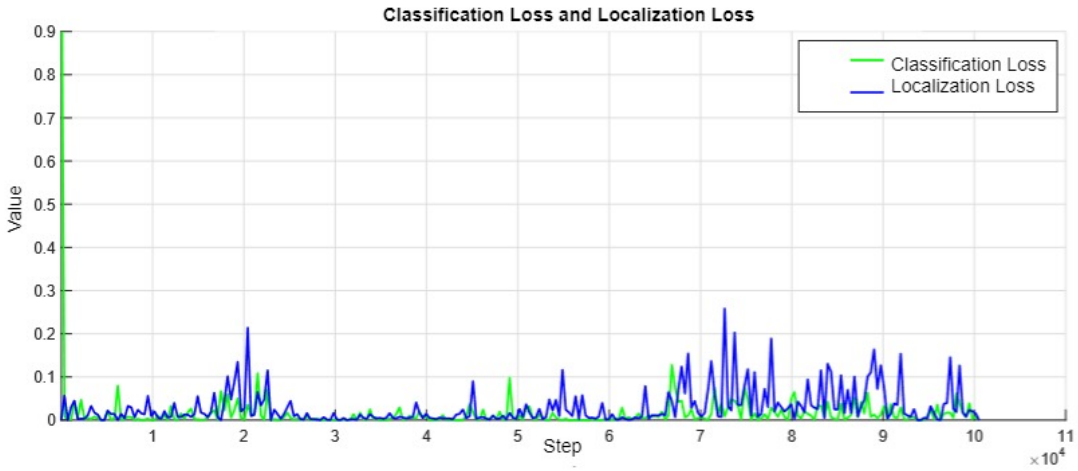


Figure 5: Classification-Localization loss

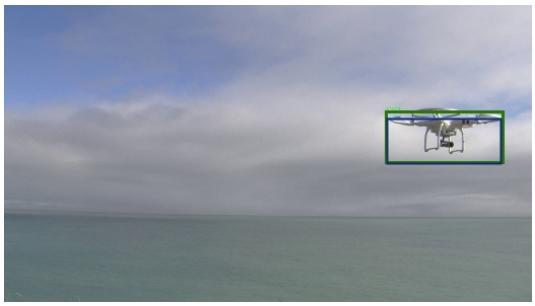
(a) (b)

Figure 6: Results with Faster R-CNN and ResNet-101

(a) (b)

Figure 7: Results with Faster R-CNN and Inceptionv2

closer to the camera and appears big. Bounding boxes detected by all three models are shown in green, the ground-truth bounding boxes are in blue. As can be seen, all models are able to detect the drone with a good IoU with ground-truth bounding box. The image on the right is a distant view of the drone. This image also has two drones. As seen, Faster R-CNN with Inceptionv2 is able to detect one drone. Here Faster R-CNN with ResNet-101 performs best, as it can detect both the drones, with reasonable accuracy. As we observe, SSD was not able to detect either of the drones, which makes it unsuitable for small object detection.



(a)

(b)

Figure 8: Results with SSD

Models	Iteration	mAP
Faster R-CNN with ResNet-101	60K	0.49
Faster R-CNN with Inceptionv2	65K	0.35
SSD	60K	0.15

Table 1: Comparison of different results.

5. Conclusion and Future Work

In this paper, we have presented the results of our application of various object detection models and fine-tuned them to get better results for small drones. From the results, it is seen that Faster R-CNN with ResNet-101 performs best on the training and testing dataset. In this work, we have not focussed on measuring the time taken for detection. Future work could focus on measuring this metric in order to assess suitability and further improvements of the model for real-time applications.

References

[1] A. Coluccia, M. Ghenescu, T. Piatrik, G. De Cubber, A. Schumann, L. Sommer, J. Klatte, T. Schuchert, J. Beyerer, M. Farhadi, et al. Drone-vs-bird detection challenge at ieee avss2017. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017. 3

[2] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 3

[3] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. 2, 3

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2

[7] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 3

[9] 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*, pages 21–37. Springer, 2016. 2, 3

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788. 2

[11] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2

[12] 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*, pages 91–99. 2, 3

[13] M. Saqib, S. D. Khan, N. Sharma, and M. Blumenstein. A study on detecting drones using deep convolutional neural networks. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–5. 3

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826. 3