A Study on Detecting Drone Using Deep Convolutional Neural Network

Muhammad Saqib
University of Technology Sydney
Broadway, Ultimo, NSW 2007, Australia
muhammad.saqib@student.uts.edu.au

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

Sultan Daud Khan
Makkah Technology Valley
Kingdom of Saudi Arabia
sdkhan@gistic.org

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

Abstract

The object detection is a challenging problem in computer vision with various potential real-world applications. The objective of this study is to evaluate the deep learning based object detection techniques for detecting drones. In this paper, we have conducted experiments with different Convolutional Neural Network (CNN) based network architectures namely Zeiler and Fergus (ZF), Visual Geometry Group (VGG16) etc. Due to sparse data available for training, networks are trained with pre-trained models using transfer learning. The snapshot of trained models is saved at regular interval during training. The best models having high mean Average Precision (mAP) for each network architecture are used for evaluation on the test dataset. The experimental results show that VGG16 with Faster R-CNN perform better than other architectures on the training dataset. Visual analysis of the test dataset is also presented.

1. Introduction

An object can be any physical quantity with semi-rigid structure and sometimes a repeatable pattern. Object detection in natural environment is a challenging task due to high variation among the objects of the same type. Additionally, changes in appearance, illumination, and viewpoint significantly reduces the performance of an object detector. Most of the object detectors perform poorly in the case of changes to the scale and deformation. Occlusion and background clutter/noise adds more complexity to the object detector.

Traditional object detection systems are variants of the following pipeline: Firstly, find potential objects and their bounding boxes, then do feature extraction, and finally classify using a good classifier. Selective Search (SS) [19] enjoyed being the state-of-the-art for detection on PASCAL VOC [6], ILSVRC [17], MS COCO [11] etc. competitions. HOG [3] and SIFT [14] are the popular choices for feature extractions. A classifier is applied on image pyramid to overcome problems with scale and thus help in reduction of false positives. A non-maxima suppression technique is generally used to remove redundant bounding boxes.

A relatively more recent traditional object detection technique uses Deformable Part-based Models (DPM) [7]. DPM uses HOG detector as a root filter and high-resolution part-based filters for different parts. These models are based on handcrafted features which have low-representation ability for the objects and therefore does not perform well in the challenging environment.

On the contrary, current state-of-the-art object detectors such as R-CNN [9], Fast R-CNN [8], Faster R-CNN [16], YOLO [15], SSD [12] etc. are based on convolutional neural networks (CNN) and have outperformed the traditional techniques. The key to the success of CNNs is their ability to extract/learn generic features.

Furthermore, the advancement in computational resources such as high-performance GPUs and its easy availability through the use of high-performance cloud computing platforms, played an important role in the recent success of neural networks. Deep learning so far has been successfully applied to the traditional machine learning problems such as segmentation [13] and detection [16]. The features extracted by deep learning architectures are more expressive and robust than their traditional machine learning counterpart. Deep learning currently holds a state-of-the-art position in almost every task in machine learning and computer vision.

In this study, we have extensively carried out experimentation with state-of-the-art object detectors based on

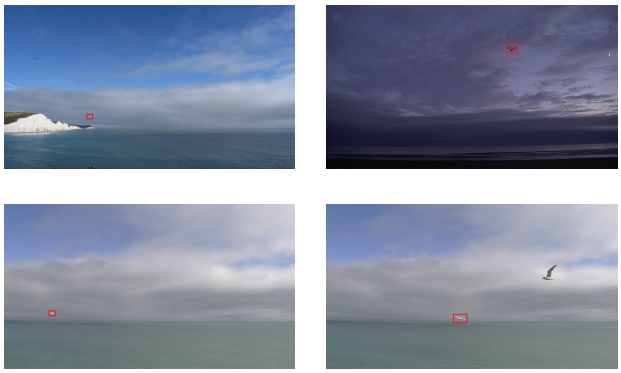

Figure 1: Sample images from competition dataset with ground-truth annotations

deep learning to detect drones in the Drone-vs-Bird detection challenge. The challenge is to detect and differentiate drones from near by flying birds as shown in Fig 1. The challenge dataset is quite complex because of varying illumination, scale change, and viewpoint variation. Misuse of small drones for illegal activities namely smuggling of drugs, terrorism activities, forms the motivation of the challenge. Hence, surveillance and tracking of drones is very essential to prevent unforeseen situations and security threats.

The remainder of the paper is organized as follows. In Section 2, the current state-of-the-art object detection techniques are discussed. Section 3, presents the proposed methodology and analysis of the experimental results are presented in Section 4. Finally, the paper is concluded in Section 5.

2. Literature review

In this section, the current state-of-the-art methods for object detection using Deep Convolutional Neural Networks (CNN) are discussed. In particular, a brief overview of R-CNN [9], Fast R-CNN [8], Faster R-CNN [16], and YOLO [15] is presented. In general, an object detector works in two steps: identifying objects candidates, and classifying the candidates to a specific object type/class based on a confidence score. Among the most widely used methods for finding object candidates there are Selective Search [19], CPMC [2], MCG [1], Edge boxes [21], etc. SS [19], CPMC [2], and MCG [1] operates at pixel level and merge them if they have similar low-level features. On the contrary, Edge boxes [21] are based on sliding window technique and are faster than the pixel based methods.

In CNN, regions in the input image are connected with the region in the output layer in the form of local connections. This is in contrast to the traditional feedforward neural networks where every input layer is fully connected with output layer. A filter is convolved with the input image to compute the output, and the weights of the filter are learned in the training phase for a particular task. Deep CNN is a compositional model in which features are extracted in a hierarchy of layers. The lower layers in the network represent low-level features such as edges, and the middle layers represent blob-like structures. Finally, the last layers extract high-level features such as shapes and complex structures.

Recent advances in object detection techniques presented the community with Region-Based Convolutional Neural Network (R-CNN) and its successors (Fast and Faster R-CNN). R-CNN [9] uses Selective Search (SS) to compute (2k) object proposals of different scales and positions. For each of these proposals, image regions are warped to fixed size $(227X227)$ pixels. The warped image regions are then fed to the CNN for detections. The proposed network architecture uses classification head for classifying region into one of the classes. The SS does not necessarily provide perfect proposals. Therefore, to make up for the slightly wrong object proposals, regression head uses linear regression to map predicted bounding boxes to the ground-truth bounding boxes. R-CNN is very slow at test time where every individual object proposals are passed through CNN. The feature extracted are cached to the disk. Finally, a classifier such as SVM is trained in an offline manner. Therefore, the weights of the CNN did not have the chance to update itself in response to these offline part of the network. Moreover, the training pipeline of the R-CNN is complex.

In Fast R-CNN [8] the order of the extracting region of proposals and running the CNN is exchanged. In this architecture whole image is passed once through the CNN and the regions are now extracted from convolutional feature map using ROI pooling. This change in architecture reduces the computation time by sharing the computation of convolutional feature map between region proposals. The region proposal are projected to the corresponding spatial part of convolutional feature volume. Finally, fully connected layer expect the fixed size feature vector and therefore the projected region is divided into grid and Spatial Pyramid Pooling (SPP) is performed to get fixed size vector. SPP deals with the variable window size of pooling operation and thus end-to-end training of the network is very hard. The generation of the region proposals is the bottle neck at the test time. In above mentioned approaches, CNN was used only for regression and classification. The idea was further extended to use CNN also for region proposals. The latest offspring from the R-CNN family, the Faster R-CNN [16] proposed the idea of small CNN network called Region Proposal Network (RPN), build on top of the convolutional feature map. A sliding window is placed over feature map in reference to the original image. The notion of anchor box is used to capture object at multiple scales. The center of the anchor box having different aspect ratio and size coincide with the

center of sliding window. RPN generates region proposals of different sizes and aspect ratios at various spatial locations. RPN is a two layered network which does not add to the computation of overall network. Finally, regression provides finer localization with the reference to the sliding window position.

Although Faster R-CNN and its predecessors perform well with high accuracy, they are computationally very expensive and time consuming, make them undesirable for real-time applications. Faster-RCNN works at a rate of 7 frames per second, while maintaining high accuracy.

Recent attempts in the development of object detectors with real-time applications as target, YOLO [15] and Single Shot MultiBox Detector(SSD) [12] were developed. YOLO [15] follow completely different approach from region proposals and sliding windows based approaches. It divides the image into a grid of cells. Each cell then predicts the bounding box and class for the object. The predicted bounding box with the high score of confidence shows the certainty of the object. Bounding box and class prediction together provide the final score for the object category. The SSD method is based on a feed-forward convolutional neural network which generates fixed-sized bounding boxes along with the confidence scores for each class. Non-maxima suppression is used to refine and produce final detection results.

Based on the brief investigation of the state-of-the-art, Faster R-CNN was considered in this study for experiments on drone detection. Different CNN architectures were used with Faster R-CNN for analysis.

3. Proposed methodology

It is a study which considers various state-of-the-art methods using deep CNN. We have used Caffe [10] deep learning library for our experiments. The Caffe-based pretrained models are publically available for most of the object detectors. As there are less number of images for deep learning system to learn from scratch. Therefore to take full advantage of network architectures, we have used transfer learning from ImageNet [4] to fine-tune our models. The fine-tuning process helps our system to converge faster and perform better. We have used various network architectures such as ZF [20], VGG16 [18], and VGG_M_1024 [18] to train the system and evaluate the performance on the test dataset. ZF is a 8 layered architecture containing 5 convolutional layers and 3 fully-connected layers. Similarly, VGG16 is a 16 layered architecture that has 13 convolutional layers and 3 fully connected layers.

4. Experimental results

4.1. Dataset

We have comprehensively carried out experimentation on the Bird-Vs-Drone dataset. This dataset contains 5 MPEG4-coded videos taken at different time. There are 2727 frames having a resolution of $1920X1080$. The drone appears in the scene at a different scale, viewpoint, and illumination. The annotations are only provided for the drones. The objective is to detect drones and also at the same time not to confuse with birds. The annotations provide width, height and top left (x,y) coordinate for the ground truth bounding box of the drone. For experiments, these annotations are converted to various formats compatible with different object detection methods.

4.2. Performance on training dataset

We trained our models with Nvidia Quadro P6000 GPU with a learning rate of 0.0001 and batch size of 64. The RPN batch size is kept constant at 128 for region based proposal networks. We have analyzed the performance of each network architecture at a different iteration. In training, the snapshot of trained models are saved at the interval of $10k$. Among all the iterations, best results obtained for each network architectures are reported in Table 1. Detections with overlap greater than the 50% Intersection Over Union (IOU) threshold with the corresponding ground-truth bounding box are considered as true positive and all other detections as false positive as shown in Eq. 1 [5].

$$IOU = area\left(B_{pred} \cap B_{gt}\right) / area\left(B_{pred} \cup B_{gt}\right) \quad (1)$$

where B_{pred} and B_{gt} denotes predicted bounding box and ground truth bounding box respectively. The ground truth box with no matching detection are considered false negative detection. To evaluate the detection performance, we use Average Precision calculated from the area under the Precision-Recall (PR) curve [5]. While, mAP is used for a set of detections and is the mean over classes, of the interpolated AP for each class.The reported results show the best performance of VGG16 is $0.66(mAP)$ at the $80k^{th}$iteration and ZF is $0.61(mAP)$ at the $100k^{th}$ iteration. The complete analysis is provided in the graph given in Fig. 2.

4.3. Visual analysis of test results

We evaluate the best trained model of each network architecture on test dataset. The performance can be seen on sample frames from test dataset in Fig 3. The first row shows the input frames from original test dataset. The second row shows the detection results using VGG16, and the third shows the result using ZF model. The fourth row show the result of VGG_M_1024.

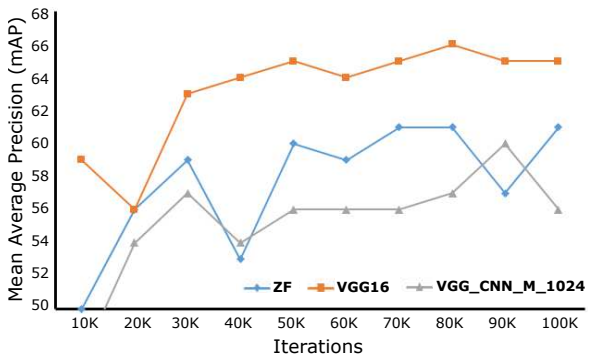

Figure 2: Performance of network architecture at each iteration

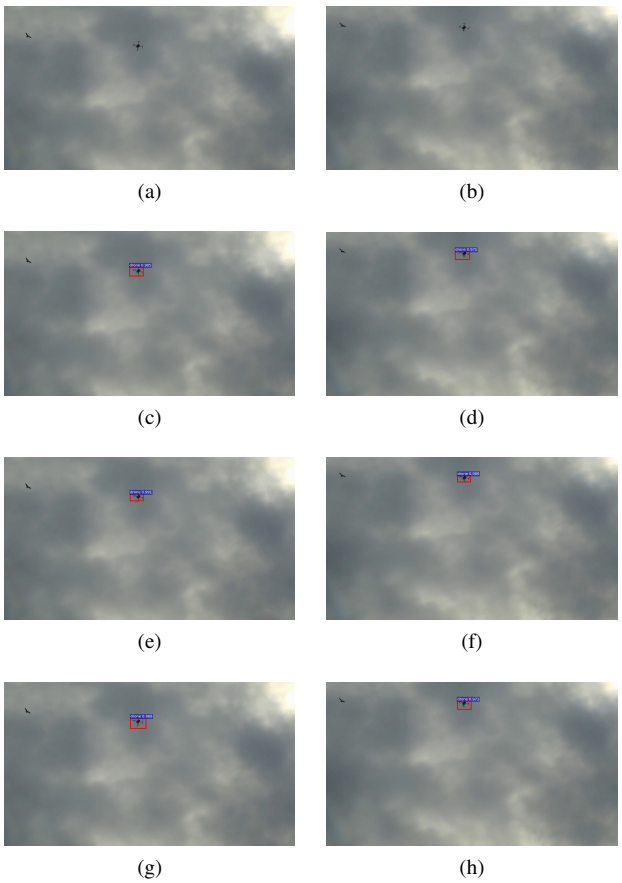

Figure 3: Results on test dataset

5. Conclusions

In this paper, we have evaluated different object detector for detection of drones. It is demonstrated through experimentation that the $VGG16$ perform better on training dataset.The results can be improved if the birds are also annotated. Considering bird as a separate class will reduce false positives and the trained model will be able to clearly differentiate between birds and drones.

Models	Iteration	mAP
ZF [20]	$100k$	0.61
VGG16 [18]	$80k$	0.66
VGG_CNN_M_1024 [18]	$90k$	0.60

Table 1: Performance of various network architectures on training dataset.

References

[1] P. Arbelez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 328–335, June 2014. 2

[2] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1312–1328, July 2012. 2

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 1

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 3

[5] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. 3

[6] M. Everingham, A. Zisserman, C. K. Williams, L. Van Gool, M. Allan, C. M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorkó, et al. The pascal visual object classes challenge 2007 (voc2007) results. 2007. 1

[7] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 1

[8] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 1, 2

[9] 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. 1, 2

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678. ACM, 2014. 3

[11] 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. 1

[12] 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. 1, 3

[13] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1

[14] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999. 1

[15] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016. 1, 2, 3

[16] 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, 2015. 1, 2

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3, 4

[19] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 1, 2

[20] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 3, 4

[21] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014. 2