An experimental study of vehicle detection on aerial imagery using deep learning-based detection approaches

Xuanzhi Liao12*, Shahnorbanun Sahran1 and Syaimak Abdul Shukor1

1Center for Artificial Intelligence Technology, Faculty of information science & technology, Universiti Kebangsaan Malaysia, UKM Bangi, 43600, Malaysia
2Computer Teaching and Research Office, Youjiang Medical University For Nationalities, Baise, Guangxi, 533000, China
*Corresponding author’s e-mail: p98708@siswa.ukm.edu.my

Abstract. Deep convolutional neural network technology is widely used to deal with general object detection in computer vision, and it achieved remarkable progress. Unmanned aerial vehicles provide large numbers of aerial imagery that significantly facilitate several applications including traffic monitoring, surveillance, tracking, rescue, and safe military tasks. This paper presents an experimental study to evaluate the performances of several state-of-the-art deep learning-based detection approaches on vehicle detection from aerial imagery. The pre-trained models, including Faster R-CNN, R-FCN, and SSD, are adopted from the TensorFlow model zoo, and the VEDAI dataset is used as the benchmark. The results show that Faster R-CNN combined with Resnet101 backbone achieved the highest mAP, which is 39.73% on the COCO metric. This experimental study expects to be a guideline to choose suitable approaches for particular applications.

1. Introduction
Recently, unmanned aerial vehicles are widely adopted in transportation operation, safety engineering, and planning because of its price drop and surveillance video technologies[1]. Unmanned aerial vehicles provide large numbers of aerial imagery that significantly facilitate several applications, including traffic monitoring, surveillance, tracking, rescue, and safe military tasks. It is crucial to select a suitable deep learning model carefully because a lousy performance detector affects the understanding of image contents[2]. However, vehicle detection on aerial imagery holds promising interests and values in the transportation research field, and it is also a challenge because of tiny size objects.

In the past studies of vehicle detection, the traditional approaches used as 1) Gaussian Mixture Model for detecting moving objects[3]; 2) Histogram of Oriented Gradient (HOG) for extracting significant target features[4]; 3) Scale Invariant Feature Transform (SIFT) for matching and extracting features[5]; 4) Support Vector Machine (SVM) or AdaBoost for classifying the extracted features[4]. In general, the traditional vehicle detection applied machine learning methods to extract the features, and then fed them to a classifier. Although these methods have great achievement in object detection, some troubles are still suffered due to structured data-limited and weak generalization ability of feature extraction[6].

Deep learning is a subset of machine learning where it relies on numerous layers of algorithms, which provide different expression to the data it feeds on. Deep learning algorithms are widely used in
the computer vision area for object detection purposes, and they have promising achievements[7]. Current deep learning-based detection approaches can be categorized into two types, which are the one-stage approach and the two-stage approach. The one-stage approach directly maps bounding box coordinates based on classification and regression methods, such as SSD[8], Retinanet[9], and YOLO[10]. The two-stage approach is a region proposal based approach that adopts two-stage classification, regression and feature alignment, such as, R-CNN[11], Fast R-CNN[12], Faster R-CNN[13] and R-FCN[14].

Tensorflow[15] is a well-known deep learning API developed by Google. In object detection research area, Tensorflow detection model zoo[16] provides a collection of state-of-art detection models, which are discussed and analyzed on the COCO dataset[17], the Kitti dataset[18], the Open Images dataset[19] and the AVA v2.1 dataset[20]. In this paper, we carried out experiments on aerial imagery vehicle detection domain using pre-trained models provided by the Tensorflow Model zoo[16]. According to Top-1 Accuracy, we choose the backbone networks from ResNet-50, ResNet-101, Inception v2, and MobileNet v2. Therefore, the following deep learning-based detection models are adopted, which are Inception v2 and MobileNet v2 for SSD[8], ResNet101 for R-FCN[14], ResNet50, ResNet101 and Inception v2 for Faster R-CNN[13]. The results expect to guide the tips of selecting a suitable object detection model for a particular case.

2. Related Works
There were abundant papers discussed vehicle detection by different methods in the past decade. In this section, we review some works related to vehicle detection from aerial imagery using a deep learning approach.

Chen et al. [21] proposed a hybrid deep convolutional neural network (HDNN) to detect the large-scale variance of vehicles from satellite images. However, the proposed algorithm is time-consuming for image computation. It costs an average of 8 seconds per image, even using the GPU.

Yongzheng Xu et al. [1] adopt the Faster R-CNN method to detect the car from low-altitude UAV imagery. The Faster R-CNN demonstrates the prominent performance. Faster R-CNN outperforms viola-jones, HOG +SVG, and Vibe, with the highest completeness (96.40%) and correctness (98.43%). However, the experimental results indicate that deep learning algorithms perform significantly compared to traditional machine learning.

Jorge E. Espinosa et al. [22] compared the performance of faster R-CNN model and AlexNet+GMM combination model on, experimental results show that Faster R-CNN outperforms AlexNet+GMM, it also indicated that Faster R-CNN still has room for improvement if other high-performance feature extractors are adopted.

Jinze Li et al. [23] modified the YOLOv3[24] algorithm by replacing the residual connections to a dense connection and extend more of convolutional layers in darknet53. The modified algorithm achieved 42.03% mAP on the UAVDT benchmark.

Seongkyun Han et al. [25] proposed a lightweight one stage method called DRFBNet300, which applied MobileNet v1 as backbone SSD300 with RFB basic module[26] that used to detect real-time vehicle activities from Unmanned-Aerial-Vehicle Imagery. The proposed algorithm achieves 21 mAP on the COCO metric, which is the highest score compared to SSD [8] and RFBNet[26].

Deep learning-based detection approaches have great achievements in vehicle detection. Therefore, this paper constructs an experimental study of several state-of-art models on vehicle detection from aerial imagery, and the results could guide to select a proper algorithm for a particular case.

3. Briefing of deep learning algorithms
This section presents a short definition of deep learning algorithms adopted in our experiments, which are Faster R-CNN[13], R-FCN[14], and SSD[8].
3.1. Faster R-CNN

R-CNN[11], SPP-net[27], and Fast RCNN[13] are early object detection models depended on region proposal method, but these models spend a lot of time on the computation of region proposals which affects the overall performance. To solve this time-consuming problem, Ren et al. proposed Faster R-CNN[13], where they have used region proposal network (RPN) instead of mentioned above region proposal method. RPN is a fully convolutional network[28] (FCN) that allows an arbitrary size of the image to be input and outputs sets of rectangular candidate object proposals which are concerned to scores to detect whether the objects are contained or not.

Similar to Fast R-CNN[12], Faster R-CNN[13] uses the sets of entire images are given as inputs to the convolutional layers of Faster R-CNN[13] to generate the feature map, and then region proposal is used to identify the region proposals. In RPN, the region proposals detected per sliding window locations are called anchors. A relevant anchor box is selected by applying a threshold value over the “objectness” score. Selected anchor boxes and the feature maps computed by the initial CNN model together are fed to the RoI pooling layer for reshaping and the output of RoI pooling layer fed into FC layers for final classification and bounding box regression[29]. The following Figure 1 demonstrates the architecture of Faster R-CNN[13].

![Figure 1. The architecture of Faster R-CNN (Source from [29])](image1)

3.2. R-FCN

R-CNN series models such as R-CNN[11], Fast R-CNN[12], and Faster R-CNN[13] generate the region proposals first, and then fully connected (FC) layers are stacked after ROI pooling has been done. The process of FC layers is time-consuming and complicated because the ROI is not shared, and the number of connections is increased. Figure 2 shows the architecture of the R-CNN series. Unlike the R-CNN series, R-FCN[14] removed all FC layers after ROI pooling. All significant complexity is moved before RoI pooling to generate score maps. The same set of score maps will be used in region proposals and after RoI pooling processes in order to perform the average voting, which is a simple calculation. By removing the FC layers, R-FCN[14] is even faster than Faster RCNN[13] in a compatible mAP because of nearly cost-free. The architecture of R-FCN[14] is shown in Figure 3.

![Figure 2. The architecture of R-CNN series (Source from[12])](image2)
3.3. Single Shot Multi-Box Detector (SSD)
Liu et al. [8] proposed a one-stage approach for object detection called Single Shot Multi-Box Detector (SSD). SSD was inspired by the anchors adopted in Multi-Box, RPN, and multi-scale representation that aims to deal with strong spatial constraints imposed on bounding box predictions to overcome the shortcoming of detecting small objects in groups. SSD extracts feature maps from different scales where they could use to recognize small and large objects well. Also, the SSD applies anchor boxes with varying ratios of aspects to discretize the space of bounding boxes. The SSD framework is demonstrated in Figure 4. SSD uses VGG16 net as the backbone, and extra feature layers are added to the end of networks that are responsibly predicting boxes with different scales, aspect ratios, and confidences.

4. Experimental Evaluation
This section presents a comparative evaluation of deep learning detection algorithms, Faster-RCNN[13], R-FCN[14], SSD[8] are selected in our experiments. Firstly, we introduce a computer that supports our experiments with a specific configuration. Secondly, we make a briefing of the VEDAI dataset, and experimental results will be presented in the end. For all experiments, the performances of algorithms will be evaluated by Mean Average Precision (mAP) on the COCO metric[17].

4.1. Computer configuration
During the implementation, GPU-enabled TensorFlow 1.14.0 is adopted, which uses CUDA 10.0 and cuDNN v7.6.1 GPU library to speed up the training progress. The configuration of the computer used in our experiments is shown in Table 1.
Table 1. Computer configuration

|                |          |
|----------------|----------|
| GPU            | Nvidia Geforce GTX 1660 (6GB) |
| CPU            | Intel Core i3-8100 |
| RAM            | DDR4 16GB |
| O/S            | Windows 10 |
| GPU Library    | CUDA10.0 with cuDNN v7.6.1 |
| Toolkit        | Tensorflow-GPU 1.14.0 |

4.2. Dataset Specification

The Vehicle Detection in Aerial Imagery (VEDAI) [30] benchmark is used in this paper, which could be downloaded from https://downloads.greyc.fr/vedai/. VEDAI dataset contains two sizes of aerial images: VEDAI 1024(1024 *1024 pixels) and VEDAI 512(512 * 512 pixels). VEDAI benchmark comprises nine classes of objects, the detailed data distribution and the image examples of the VEDAI dataset are demonstrated in follow Table 2 and Figure 6. However, our study emphasis on vehicle detection, some categories like planes, boats, and others contribute numbers scarcely. Therefore, these three categories are discarded, and the remaining six categories (cars, pick-ups, camping cars, trucks, tractors, and vans) are merged into one class named vehicle.

Table 2. Data distribution of the VEDAI dataset

| Type           | Number | Type   | Number | Type   | Number |
|----------------|--------|--------|--------|--------|--------|
| Car            | 1408   | Truck  | 307    | Plane  | 48     |
| Pick-up        | 955    | Tractor| 190    | Boat   | 171    |
| Camping-car    | 397    | Van    | 101    | others | 211    |

Figure 5. Examples of images from dataset VEDAI

Our experiments use VEDAI 1024 as training and testing purposes. VEDAI 1024 consists of 1250 instances in total, and 1164 images are remaining after we removed the scarce categories mentioned above. One thousand fifty images are selected randomly as the training set, and the rest of the 114 images are testing set. All selected deep learning algorithms are trained in 250000 iterations; batch size is one, and momentum is 0.9, with an equivalent learning rate of 0.0003. The rest of the parameters are initialized.

4.3. Data pre-processing

Data preprocessing is the most time cost work on object detection because objects need to be annotated from thousands of images manually. Razakarivony and Jurie[30] provide the annotation in TXT format, we proposed a data preprocessing kit on the VEDAI dataset that facilities the data preprocessing phase. Data preprocessing kit on the VEDAI dataset could be downloaded on this link https://github.com/liaoxuanzhi/VEDAI.
4.4. Experimental results

In this experimental study, SSD, Faster RCNN, and RFCN combined with various feature extractors are compared.

The following Table 3 demonstrates mean average precision (mAP) and detection results of object detection frameworks trained on the VEDAI dataset. We can see that Faster RCNN ResNet101 achieved the highest mAP score with almost 23 hours of training time. The mAP of Faster RCNN Inception v2 is 32.72%, but it is less time consuming, the training time is shortened to 8 hours 38 minutes 37 seconds. Faster RCNN ResNet50 achieved the highest mAP at 0.5 IOU level, which is 78.60%. This result indicated that the object detection approach applies different backbones might differ the training times and performances.

Although R-FCN is second place in mAP ranking with a value of 38.40%, mAP in 0.5 IOU is higher than Faster RCNN ResNet101. Besides, it takes 19 hours 56 minutes 37 seconds, which is shorter than Faster RCNN ResNet101 three hours.

SSD performances poorly compared to other models, the mAP only achieved 7.532% in SSD Inception v2 and 6.294% in SSD MobileNet v2. The experimental results showed that the SSD approach is not working well with small object detection. More surprisingly, the training time conflicts with the Tensorflow model zoo[16]. The results on the Tensorflow model zoo[16] indicated that SSD is the fastest compared to other approaches in the list, but our experiment results gave an opposite conclusion. Moreover, an error message was received when the SSD algorithm was launched. It was said, “failed to get the convolutional algorithm. This is probably because cuDNN failed to initialize, so try looking to see if a warning log message was printed above.” It is because the SSD approach requires a high-performance graphics card with excellent computing ability and sufficient video RAM. The algorithm eventually worked after GPU allow_growth is configured. Nvidia Geforce GTX 1660 (6GB) is used for our implementation, and the video RAM is only 6GB. Therefore, the performance gap gains due to small video RAM.

In general, the two-stage approach shows a better mAP than the one-stage approach impressively. Meanwhile, the one-stage approach requires an excellent GPU to initialize the convolutional algorithm. Otherwise, the speed of algorithms can’t be performed remarkably, and more training time is consumed. Figure 7 shows the detection results of trained on the VEDAI dataset.

| Framework     | Backbone    | iterations | mAP (%) | mAP0.5IOU (%) | Training time |
|---------------|-------------|------------|---------|---------------|---------------|
| Faster RCNN   | ResNet101   | 250000     | 39.73%  | 77.08%        | 22h-59m-58s   |
| Faster RCNN   | Inception v2| 250000     | 32.72%  | 76.86%        | 8h-38m-37s    |
| Faster RCNN   | ResNet50    | 250000     | 31.42%  | 78.60%        | 15h-44m-8s    |
| R-FCN         | ResNet101   | 250000     | 38.40%  | 77.11%        | 19h-56m-10s   |
| SSD           | Inception v2| 250000     | 7.532%  | 27.98%        | 24h-33m-58s   |
| SSD           | MobileNet v2| 250000     | 6.294%  | 21.98%        | 23h-7m-8s     |

5. Conclusions

In this paper, we presented an experimental study of deep learning-based object detection approaches on vehicle detection from aerial images. We adopted Faster R-CNN, R-FCN, and SSD from the TensorFlow model zoo. The experimental results could be generated and highlight as follows:

• We proposed a kit on the VEDAI dataset that facilitates the data preprocessing phase. Therefore, researchers could spend more time on algorithm improvements rather than data preprocessing.

• In our experimental study, the two-stage approach generally has better mAP than the one-stage approach. The highest mAP is achieved by Faster R-CNN, followed by R-FCN, and SSD.
• The premise of the one-stage algorithm that could be performed and computed effectively is that GPU must have enough video RAM and great computing ability. Conversely, it must be time-consuming work.

Deep learning has significantly facilitated the development of computer vision, and we have benefited a lot during this experimental study. We expect that the preliminary results above could be tips to select a suitable algorithm for a particular case.

References
[1] Xu, Y., Yu, G., Wang, Y., Wu, X., & Ma, Y. (2017). Car detection from low-altitude UAV imagery with the faster R-CNN. Journal of Advanced Transportation, 2017.
[2] Acatay, O., Sommer, L., Schumann, A., & Beyrer, J. (2018, November). Comprehensive Evaluation of Deep Learning based Detection Methods for Vehicle Detection in Aerial Imagery. In 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-6). IEEE.
[3] Stauffer, C., & Grimson, W. E. L. (1999, June). Adaptive background mixture models for real-time tracking. In Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149) (Vol. 2, pp. 246-252). IEEE.
[4] Yan, G., Yu, M., Yu, Y., & Fan, L. (2016). Real-time vehicle detection using histograms of oriented gradients and AdaBoost classification. Optik, 127(19), 7941-7951.
[5] Ma, X., & Grimson, W. E. L. (2005, October). Edge-based rich representation for vehicle classification. In Tenth IEEE International Conference on Computer Vision (ICCV’05) Volume 1 (Vol. 2, pp. 1185-1192). IEEE.
[6] Li, X., Liu, Y., Zhao, Z., Zhang, Y., & He, L. (2018). A Deep Learning Approach of Vehicle Multitarget Detection from Traffic Video. Journal of Advanced Transportation, 2018.
[7] Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems.
[8] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.
[9] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).
[10] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).
[11] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).
[12] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).
[13] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).
[14] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. In Advances in neural information processing systems (pp. 379-387).
[15] Google. (2015) TensorFlow. tensorflow.org.
[16] Wu, N., & Rathod, V. (2017). Tensorflow detection model zoo.
[17] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.
[18] Geiger, A., Lenz, P., & Urtasun, R. (2012, June). Are we ready for autonomous driving? the KITTI vision benchmark suite. In 2012 IEEE Conference on Computer Vision and Pattern Recognition (pp. 3354-3361). IEEE.

[19] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., ... & Ferrari, V. (2018). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. arXiv preprint arXiv:1811.00982.

[20] Gu, C., Sun, C., Ross, D. A., Vondrick, C., Pantofaru, C., Li, Y., ... & Schmid, C. (2018). AVA: A video dataset of spatio-temporally localized atomic visual actions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6047-6056).

[21] Chen, X., Xiang, S., Liu, C. L., & Pan, C. H. (2014). Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geoscience and Remote Sensing Letters, 11(10), 1797-1801.

[22] Espinosa, J. E., Velastin, S. A., & Branch, J. W. (2017, November). Vehicle detection using AlexNet and Faster R-CNN deep learning models: a comparative study. In International Visual Informatics Conference (pp. 3-15). Springer, Cham.

[23] Li, J., Wang, R., & Ding, J. (2019) Tiny Vehicle Detection from UAV Imagery. In: Wang Y., Huang Q., Peng, Y. (eds) Image and Graphics Technologies and Applications. IGTA 2019. Communications in Computer and Information Science, vol 1043. Springer, Singapore.

[24] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

[25] Han, S., Yoo, J., & Kwon, S. (2019). Real-Time Vehicle-Detection Method in Bird-View Unmanned-Aerial-Vehicle Imagery. Sensors, 19(18), 3958.

[26] Liu, S., & Huang, D. (2018). Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 385-400).

[27] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1904-1916.

[28] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[29] Sultana, F., Sufian, A., & Dutta, P. (2019). A Review of Object Detection Models based on Convolutional Neural Network. arXiv preprint arXiv:1905.01614.

[30] Razakarivony, S., & Jurie, F. (2016). Vehicle detection in aerial imagery: A small target detection benchmark. Journal of Visual Communication and Image Representation, 34, 187-203.