Automated Military Vehicle Detection From Low-Altitude Aerial Images

Farrukh Kamran*, Muhammad Shahzad† and Faisal Shafait‡
School of Electrical Engineering and Computer Science (SEECS)
National University of Sciences and Technology (NUST), Islamabad, Pakistan
Email: *fkamran.mscs16seecs, †muhammad.shehzad, ‡faisal.shafait@seecs.edu.pk

***Abstract*—Detecting military vehicles and distinguishing them out from non-military vehicles is a significant challenge in the defence sector. Detection of military vehicle could help to identify enemy's move and hence, build early precautionary measures. Recently, many deep learning based techniques have been proposed for vehicle detection purpose. However, they are developed using datasets that are not useful if military specific vehicle training and detection is required. Hyper-parameters in those techniques are not tuned to entertain low-altitude aerial imagery. We aim to develop state-of-the-art deep learning framework to detect particularly military vehicle along with other standard non-military vehicles. The major bottleneck in the application of deep learning frameworks to detect military vehicles is the lack of available datasets. In this context, we prepared a dataset of low-altitude aerial images that comprises of real data (taken from military shows videos) and toy data (taken from YouTube videos). Our dataset is categorized into three main types i.e. military vehicle, non-military vehicle and non-vehicle. We employed state-of-the-art object detection algorithms to distinguish military and non-military vehicles. Specifically, the three deep architectures used for this purpose include faster region-based convolutional neural networks (Faster RCNN), recurrent fully convolutional neural networks (R-FCN), and single shot multibox detector (SSD).We observed the impact on results by increasing training data using SSD architecture. We also did comparative analysis of three state-of-the-art architectures by increasing training data and observing it's impact on results. The experimental results show that the training of deep architectures using the customized/prepared dataset allows to recognize seven types of military and four types of non-military vehicles. It can handle complex scenarios by differentiating vehicle from it's surroundings objects. We report the mean average precision (MAP) and weighted average precision (WAP) obtained using the three adopted architectures with Faster R-CNN giving the highest WAT of around 62.79 % for military vehicle category with 380,530 iterations (3 epochs).**

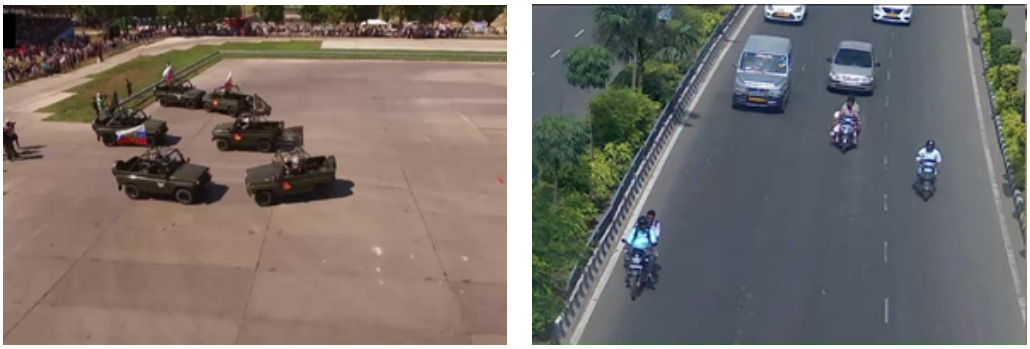
***Index Terms*—vehicle detection, vehicle classification, surveillance, military vehicle, security, military vehicle detection**

I. Introduction

Detection and localization of Military vehicles is vital for applications like surveillance, security, tracking tasks etc. These applications require accurate identification and tracking of vehicles so that military vehicles can be easily distinguished from non-military vehicles in an image. To reduce the work load of security and tracking experts, an automatic military vehicle detection system needs to be constructed. Previous techniques specifically focus on general vehicle detection in aerial images. Vehicle detection in VHR remote sensing images is vital for both in civilian and military surveillance purposes. Vehicle detection from aerial images has got attention worldwide [6, 7, 14-16]. It is challenging due to small size and variable orientation of vehicles. Aerial images with complicated backgrounds (Figure 1) increase difficulties in detection and classification. Earlier, vehicle detection was performed by applying techniques composed of hand-crafted features and a classifier or a cascade of classifiers within a sliding window approach [2, 6, 7]. Previous sliding window search approach and shallow-learning-based features [1, 17-21] based methods were mostly used for vehicle detection. The approach presented in [6] detects vehicles with two attributes (orientation and type) on aerial images. To detect location of vehicles, it adopts AdaBoost classifier in a soft-cascade structure and fast binary detector using ICFs. Later for classification of orientations and type of vehicle, HOG features were used. This lead to effective detection performance.

Computing convolutional features separately for every candidate window is expensive [4]. Afterwards, convolutional neural networks (CNNs) were employed to classify candidate regions [1, 5]. Recently, R-CNN based detection methods have performed well in nature scene images [22]. A fully convolutional RPN is employed to generate object-like regions in Faster R-CNN, and candidate regions are inferred by a classifier after RPN. Performance of Faster R-CNN is superior to that of traditional methods (sliding windows based) because of fast speed and feature representation. Computational cost for training and testing was significantly reduced by Fast R-CNN [4] and Faster R-CNN [9]. They achieved good results on common detection benchmark datasets. In these techniques, only one convolutional feature map is shared for entire image rather than computing convolutional features separately. But performance of these techniques depend on object proposal methods. These detectors and their respective object proposals methods were developed for datasets that were different from aerial images.

Due to less training data, over-fitting problem can occur in region CNN-based methods for vehicle detection in aerial imagery. Faster R-CNN's poor performance is due to two reasons. Firstly due to coarse feature maps, RPN in Faster R-CNN is not suitable to detect small vehicles. Secondly due to less hard negative examples , classifier after RPN is un-able to differentiate vehicles and clutter backgrounds properly. After wards, SSD [32] was introduced. Its meta-architecture solves object recognition problem. A feed-forward convolution net-



(a) Military Vehicles (b) Non-Military Vehicles

Fig. 1. Low-altitude aerial images of real vehicles taken from RPTLY youtube videos [42] and our collected dataset.

work is employed to generate collection of fixed-size bounding boxes. Object class presence in each is scored. Predictions from multiple feature maps having different resolutions, is combined by this network. In this way it is able to handle objects of varying dimensions. SSD avoids proposal generation and saves computational time by encapsulating process into a single network.

In this paper, we propose a military vehicle dataset. It is composed of 13 classes which are sub-divided in 2 categories (Vehicle and Non-Vehicle). Vehicle Category is further sub-divided into military and non-military. We also investigate the applicability of SSD for detecting small objects in aerial images. Contribution of our work includes dataset which is composed of military and non-military vehicles.

Fig. 2. Vehicles from PASCAL VOC 2012 dataset [3].

This paper is organized as follows: Section II discusses available datasets. Preparation of dataset is explained in Section III. Section IV and V focus on experimental setup and results. Analysis in Section VI. Finally, Section VII concludes the paper.

Fig. 3. Two images from OIRDS [40].

II. Available Dataset

This section provides an overview of available dataset.

A. Datasets for vehicle detection

Modern approaches in deep learning need annotated training data. In addition, comparison is required to establish the most suitable approach. Table I represents summary of datasets.

Our focus is on vehicle detection and more specifically on military vehicle detection. Dataset from Pascal VOC challenge [3], contains everyday life objects. The Pascal VOC have dataset of 20 classes split into train, validation and test sets. Some vehicles from PASCAL VOC are shown figure 2. ImageNet dataset [34] have more than 14 million images and it is generally used for object detection purposes. But it is not designed to accomodate aerial images required for surveillance and security purposes.

As in [35, 36], the databases presented are different from the task we performed. Available vehicle databases mostly contain vehicles with ground view (e.g. INRIA Car dataset [37]). Work

TABLE I
Summary of existing databases for object detection

Database	Classes	# Instances	Folds	# Images
PASCAL [3]	20		train / val / test	>10,000
ImageNet [34]	21841		train / val / test	>14,000,000
OIRDS [40]	4		No cut	900
VEDAI [8]	9	2950	train / test	1268
3K Vehicle Detection [6]	2	14,235	No cut	20
Our Proposed Dataset	**13**	23,097	**train / val**	**15086**

(a) Input Image : Containing vehicles

(b) Output Image : Recognition of vehicles

Fig. 4. System performs localization and detection of real vehicles in low-altitude aerial images taken from our compiled dataset.

on target detection done in ([38, 39]) using aerial imagery but dataset is not publicly available.

Publically available dataset OIRDS (Overhead Imagery Research DataSet) [40] contains 180 vehicles in 900 annotated images. Few images are shown figure 3. It contains five classes of vehicles ('truck', 'pick up', 'car', 'van' and 'unknown'). However, no evaluation protocol is defined for this database and images are not having aerial view of vehicles that is required for our dataset.

III. Dataset Preparation

A. Challenges

Images in datasets, e.g. Pascal VOC2007 [3], in general are composed of only one or few objects that occupy a high portion of the image as compared to aerial images. Aerial Images may contain multiple objects with varying sizes and pixel-wise area in image. Currently the publically available datasets like DLR 3K Munich Vehicle Aerial Image Dataset [6] and the Vehicle Detection in Aerial Imagery (VEDAI) dataset [8], doesn't fulfill our requirement. In Figure 5, we show a representation of vehicles under different conditions. We propose a dataset that is composed of low-altitude aerial images containing military vehicles and non-military vehicles with varying backgrounds. All experiment are performed on our proposed dataset.

B. Data Collection

Our generated dataset is composed of images with several types of vehicles. We have 13 classes in total. 11 classes fall in Vehicle category while 2 classes fall in non-vehicle category. Vehicle category further splits in Military and non-military vehicle category Images are collected from YouTube videos. Our dataset is composed of Real Vehicle and Toy Vehicle. Real vehicle dataset was generated through RPTLY YouTube videos [42] and through publically available EPFL Dataset [43]. While Toy vehicle dataset was generated from videos by RC Military toy YouTube channel [44] and other channels [45-46]. Our dataset is composed of Images with various resolutions having top-view angle (low-altitude) of vehicles. Images have few categories of military vehicles too in clutter environment. These conditions help to more accurately identify type and category of vehicle.

Table II shares some details w.r.t our proposed dataset (composed of low-altitude aerial images).

TABLE II
VARYING DIMENSIONS DETAILS AND IMAGES COUNT IN OUR PROPOSED DATASET.

Details of our proposed dataset	
# Images	15086 (11733 Toy images + 3353 Real vehicle images)
Dimensions	1280*720, 1280*692, 450*300, (EPFL Data set) 360*288
Shape Resizer	1024 * 600

C. Data Annotation

Starting from videos, containing military vehicle, we used VOTT tool [47] to annotate it frame by frame in-order to generate our proposed dataset. It generated annotations in PASCAL VOC format for 11 types of vehicles. Out of total 15086 Images in our dataset, 11733 are extracted from Toy video and 3353 are from Real videos. The collected data set is annotated for two categories of vehicles (Military and non-military). Images in our dataset contains multiple objects belonging to multiple classes. Number of each type of vehicles in our training dataset are shown in Table III. We have 13 classes which are split in 2 main categories. Vehicle and non-Vehicle Category. There are total 15086 images that were manually labeled all images with bounding box and type.

IV. Experimental Setup

In this section, we first focus on architectural configurations. Afterwards we briefly introduced details of our dataset. Finally we discuss implementation details of our experiments.

A. Architectural configuration

1) Feature extractors: For our experiments, we considered three feature extractors. Resnet-101 [48], which won competitions of ILSVRC and COCO 2015 (classification, detection

(a) (b)

Fig. 5. Aerial images were taken from toy video of RC youtube channel [44-45].

and segmentation). We also used Inception v2 [49], which set the state of the art in the ILSVRC 2014 classification and detection challenges. Its network employs 'Inception units' to increase the depth and width of a network without increasing computational cost. Recently, Inception Resnet (v2) in [50], combines the optimization benefits by residual connections with the computation efficiency of Inception units.

2) Number of proposals: Number of region proposals to be sent to the box classifier can be choosed at test time in Faster R-CNN and R-FCN. In experiments, we used 300 number of proposals.

3) Location loss: For all of our experiments, following [4, 9, 32], we used the Smooth L1 [51] loss function.

4) Training and hyper parameter tuning: For Faster RCNN we used batch size of 2 (because models were trained using images with different dimensions). For SSD and R-FCN, we used batch size of 4 (we had to reduce the batch size for memory reasons). Learning rate for Faster RCNN and RFCN was 0.0003 while that in SSD was 0.004 For Faster RCNN and RFCN, in configuration settings for Images resizing, min_dimension and max_dimension was set to 600 and 1024 respectively. While for SSD, fixed_shape_resizer section was added and it's parameters height and width was set to 600 and 1024 respectively.

B. Low-Altitude Aerial Imagery Datasets

Experiments are performed on our proposed dataset that is shown in figure 6. Images in our dataset contain different sizes of objects with varying backgrounds. Main characteristics of our dataset is summarized in Table II. It comprises of real data videos and toy data videos. Real data was acquired from military shows video by RPTLY Channel and toy data videos used for training were from RC channel. Images in the dataset are of different resolutions. We manually annotated collected images in PASCAL VOC format. Experiments are performed on our low-altitude aerial images dataset that includes 2 annotated main categories i.e. vehicle (11 classes) and non-vehicle (2 classes). Vehicle category is further divided into two categories i.e. Military vehicle and Non-military vehicle. Our dataset contains 15086 images collected from Youtube videos. For performing experiments, we divided dataset into 83% training set, 17% validation set. Performance of few classes is not good due to their limited number of annotations and images. Detail of categories used can be seen in Table III.

C. Benchmarking procedure

Training was performed using Intel Core i7-7700K processor with 2 NVIDIA Titan-X- GPU's having 12 GB memory each. The operating system was Linux Ubuntu 16.04. For performance comparison of three architectures, initial fine-tuning on their respective pre-trained models was performed till 380K steps using our latest dataset with 15K Images. We also demonstrated SSD performance using our proposed military vehicle dataset. Initially pre-trained model was fined-tuned on collected data (8476 Images) till 200K steps (1 epoch). Then it was further fine-tuned on same data till 500K steps (2 epochs). Afterwards we fine-tined ckpt-500K on new data (15086 Images) till 800K steps (3 epochs).

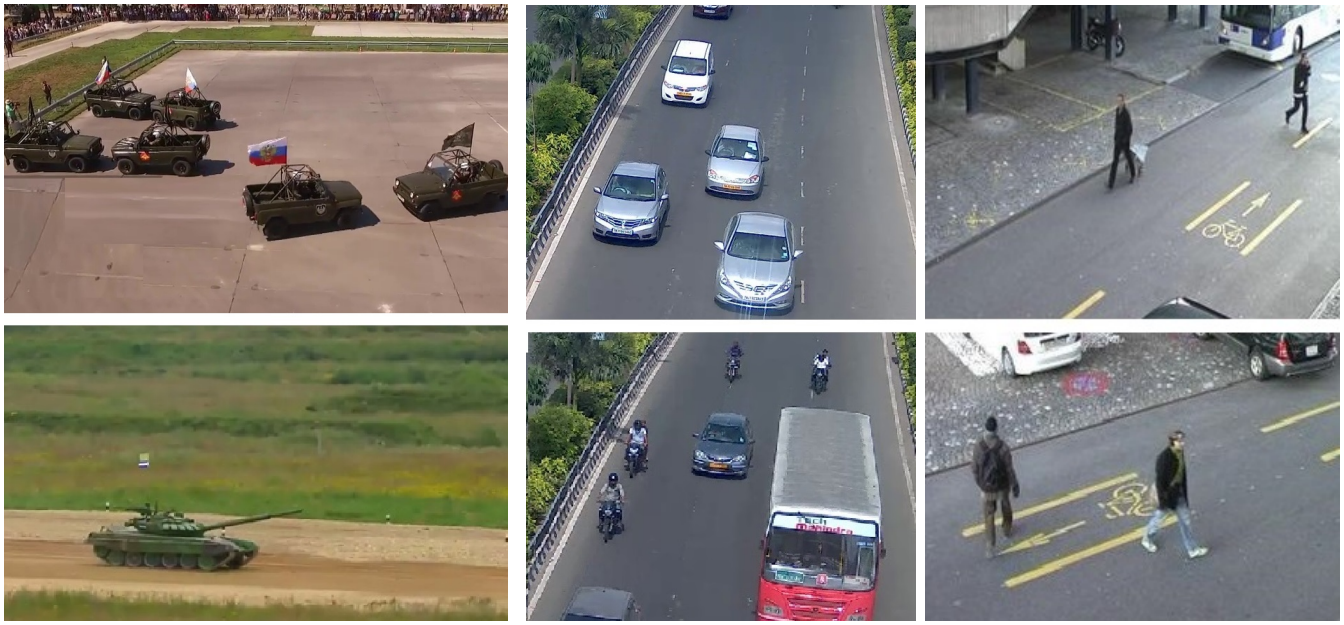
D. Implementation Details

As the training set was of limited size, we used a pre-trained model that was trained on COCO (Common Object in Context) dataset. Experiments were performed using three state of the art meta-architectures. While TF Object Detection API [31]

TABLE III

NUMBER OF INSTANCES OF CLASSES BELONGING TO MILITARY, NON-MILITARY AND NON-VEHICLE CATEGORY. IT INCLUDES MILITARY ARMOURED (M_ARMOURED), HEAVY EXPANDED MOBILITY TACTICAL TRUCK (HEMTT), MILITARY TRUCK (M_TRUCK), HIGH MOBILITY MULTI-PURPOSE WHEELED VEHICLE (HMMWV), MILITARY CAR (M_CAR), MILITARY AMBULANCE (M_MEDICS)

Category	**Military vehicle**							**Non-military vehicle**				**Non-vehicle**	
Class	Tank	M_Armoured	HMMWV	HEMTT	M_Truck	M_Car	M_Medics	Vehicle	Car	Truck	Bus	Non-vehicle	Person
# of instances	5472	1724	651	1039	1796	231	14	4284	458	1541	8	4754	1125



(a) Military vehicle category (b) Non-military vehicle category (c) Non-vehicle category

Fig. 6. Our collected dataset comprises of 3 categories (Taken from RPTLY youtube videos [42] and our collected dataset).

TABLE IV
DETAIL OF COMMON CONFIGURATION PARAMETERS FOR TRAININGS TILL 250 K STEPS WITH ALL 3 ARCHITECTURES.

Parameters	**Values**
Initial Learning Rate	0.003 (BUT 0.004 for SSD)
num_epochs	1
Batch Size	4 (BUT 2 For Faster-RCNN)
num_hard_examples	4000
shuffle	True
num_steps	0K—150K—250K—380K

TABLE V
COMMON CONFIGURATION PARAMETERS FOR TRAININGS (200K, 500K AND 800K STEPS) WITH SSD ONLY.

Parameters	**Values**
Initial Learning Rate	0.004
num_epochs	1
Batch Size	16
num_hard_examples	3000 / 3000 / 3500 / 4000
shuffle	True
num_steps	0K—200K—500K—800K

was used for training and evaluation. In order to achieve better results, we configured values for hyper parameters as shown in Table V. We evaluated MAP on different trainings steps. As shown in table VI, the results on training ckpt-500K are better than those achieved by using ckpt-200K. If IOU ratio is bigger than 0.5 w.r.t ground truth box, candidate region is selected as a positive sample. Apart from this, we also analyzed performance of three state of the architectures using our dataset.

Table IV shows settings for training using three architectures for comparison and table V shows specific configuration settings for SSD specific analysis.

V. RESULTS AND EVALUATION

We evaluate the state of the art object detection methods on our proposed dataset. We select Faster R-CNN [9], R-FCN [41] and SSD [32] as our benchmark testing algorithms for their good performance on general object detection.

Backbone networks are Inception Resnet (v2) [50] for Faster R-CNN, ResNet-101 [48] for R-FCN and Inception V2 [49] for SSD .

A. Quantitative Results

System implements combination of 3 state of the art architectures and feature extractors. System performance is evaluated on the basis of IoU, and the average precision (AP), introduced in the Pascal VOC Challenge [3].

$$\mathbf{IoU}(\mathbf{A},\mathbf{B}) = \left|\frac{A \cap B}{A \cup B}\right| \tag{1}$$

In Equation 1, $\mathbf{A}$ shows the ground-truth box collected in the annotation while predicted result is represented by $\mathbf{B}$. If estimated IoU is greater than threshold value then predicted result is a true positive else it is a false positive. Number of false positives determines the accuracy of the network. IoU method is used for evaluation of accuracy of an object detector.

Initially, we trained a model with small amount of data and kept on increasing training data. Experiment is performed using the same meta-architecture SSD with feature extractor inception v2. Initial Pre-trained Model (trained on coco dataset) was fined-tuned on our collected data (8476 Images) till 200K iterations. Afterwards it is further fine-tuned till 500K (300K times more) iterations on same data. Finally, we

increased the amount of data and further fine-tuned it on New Data (15086 Images which included previous data too) till 800K iterations. Table VI shows that by increasing training data and fine-tuning a pre-trained model, system is able to improve the weighted average precision for each class. When model was further fine-tuned from 500 K iteration till 800 K iterations, the weighted average precision slightly decreased due to diverse dataset.

TABLE VI
RESULTS OF TRAINING DONE TILL 800K ITERATIONS USING SSD ARCHITECTURE.

Total Iterations → Class ↓	**Average precision**		
	200K	***500K***	***800K***
Tank	91.21%	9358%	94.07%
HEMTT	83.31%	9033%	94.36%
Vehicle	81.54%	86.18%	86.00%
M_Armoured	81.54%	90.53%	89.40%
MAP @ 0.5 IOU	69.63%	79.14%	77.67%

Total Iterations → Category ↓	**Weighted average precision**		
	200K	***500K***	***800K***
Military vehicle	83.64%	90.27%	89.67%
Non-military vehicle	76.08%	86.89%	85.56%
Non_Vehicle	55.95%	79.84%	81.16%

TABLE VII
RESULTS OF TRAINING DONE TILL 380K ITERATIONS USING THREE STATE OF THE ART ARCHITECTURES.

	Average precision		
	Faster R-CNN	***R-FCN***	***SSD***
Feature Extractors → Class ↓	Inception resnet v2	ResNet-101	Inception v2
Tank	70.55%	68.59%	81.31%
HEMTT	88.46%	79.07%	13.72%
Vehicle	71.53%	05.94%	26.00%
M_Armoured	61.13%	51.73%	57.89%
MAP @ 0.5 IOU	50.69%	35.10%	32.36%

Architecture → Category ↓	**Weighted average precision**		
	Faster R-CNN	***R-FCN***	***SSD***
Military vehicle	62.79%	57.65%	61.56%
Non-military vehicle	58.88%	09.26%	31.35%
Non_Vehicle	60.72%	37.43%	30.51%

We also performed experiment to compare results of three architectures using our latest dataset having 15086 Images. Detection Results achieved after performing training till 380K iterations show that Faster R-CNN performed better as compared to RFCN and SSD. Results are given in Table VII.

B. Qualitative Results

As shown in Figure 4 and Figure 7, system is able to classify and localize vehicles in a low-altitude aerial images. Estimated results were compared with the ground truth using an IoU >0.5. While increasing training data and evaluating performance of SSD architecture, we observed that training till 200 K iterations performed well on classes like person, vehicle, Bus, Car. While training till 500 K iteration of able to perform well on un-seen data, especially involving tanks and few military vehicles.

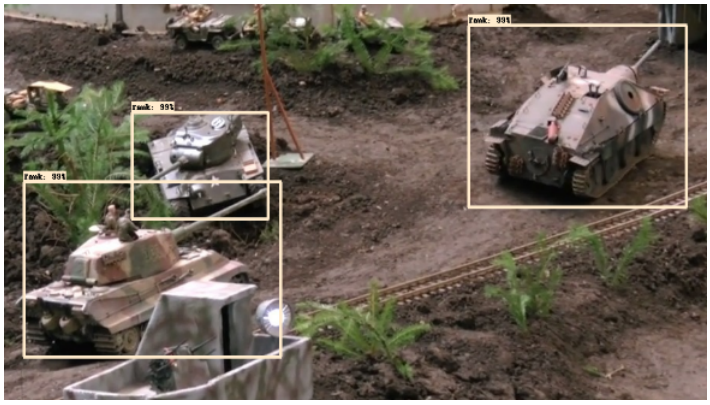


Fig. 7. Detection Results on military vehicles (taken from toy videos of RC youtube channel [44-45]).

Fig. 8. Miss-classification in real military vehicles taken from RPTLY youtube videos [42].

VI. ANALYSIS OF SUCCESS AND FAILURE CASES

Overall it was observed that two classes i.e. Tank and HEMTT, perfomed well in both of our experiments. The reason behind that is both classes were having more training data as compared to other classes. Details of our proposed dataset are given in Table III. During analysis, we observed that system had good performance on test cases and on un-seen data, but there were difficulties in some cases for which we had a small training set. It performs well on tanks in un-seen data as compared to other military vehicles because the ratio of its training data is more. Figure 8 shows the result of extensive fine-tuning (detail in Table VI). The model starts performing bad on data, on which it was previously performing better. This also because of less training data w.r.t that class. Figure 8 and figure 9 show cases of miss-detections and wrongly classified.

VII. CONCLUSION

We proposed a detector based on deep learning approach for military vehicle detection and classification from aerial images, which are taken from prepared dataset that we generated. This system detects the class and location of military and non-military vehicle in captured aerial images. The main reason

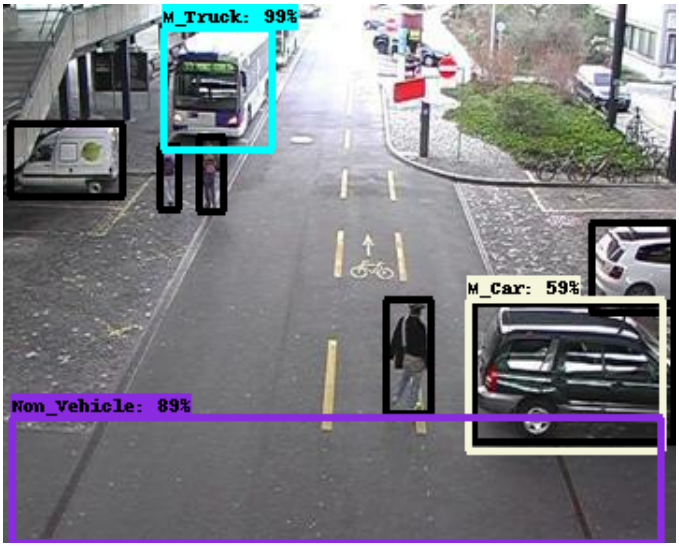


Fig. 9. Wrong classification of vehicles (taken from EPFL dataset [43]).

due to which it differs from existing methods for vehicle detection and classification is that the detector is applied on images captured from Real Vehicle Videos (Military shows) and Toy Vehicle videos (RC YouTube videos) and software system using GPUs process them. Furthermore, our collected dataset contains different scenarios, like size of vehicles, background variations etc. For selecting the best suitable architecture for this task, we performed comparative analysis between different deep-learning architectures (with feature extractors combination). Experimental results demonstrate that by applying deep-learning-based detector using our proposed dataset, it is able to detect 2 different categories of vehicles (with 11 classes). In addition, 2 more classes were added to accommodate Non-Vehicle category. We expect that our proposed dataset will make a significant contribution to the Military-Defence sector. Our target for future work is to focus on improving current detection results and extend the idea of Military-Vehicles recognition to work on other surveillance and security related projects. We demonstrated the performance of 3 state-of-the-art architecture for military-vehicle detection purposes. For SSD architecture, we systematically evaluated per category detection improvement based on change in hyper parameters and increase in images per category. We proposed a dataset and hyper parameters settings for handling small objects in aerial images for best detection results. As per our knowledge, no work is currently done on military vehicle detection from aerial images.

REFERENCES

[1] X. Chen, S. Xiang, C.-L. Liu, and C.-H. Pan. Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geoscience and remote sensing letters, 11(10):17971801, 2014.

[2] G. Cheng and J. Han. A survey on object detection in optical remote sensing images. ISPRS Journal of Photogrammetry and Remote Sensing, 117:1128, 2016.

[3] 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):303338, 2010.

[4] R. Girshick. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 14401448, 2015.

[5] G. V. Konoplich, E. O. Putin, and A. A. Filchenkov. Application of deep learning to the problem of vehicle detection in uav images. Soft Computing and Measurements (SCM), 2016 XIX IEEE International Conference on, pages 46. IEEE, 2016.

[6] K. Liu and G. Mattyus. Fast multiclass vehicle detection on aerial images. Geoscience and Remote Sensing Letters, IEEE, PP(99):15, 2015.

[7] T. Moranduzzo and F. Melgani. Detecting cars in uav images with a catalog-based approach. IEEE Transactions on Geoscience and Remote Sensing, 52(10):63566367, 2014.

[8] S. Razakarivony and F. Jurie. Vehicle detection in aerial imagery: a small target detection benchmark. Journal of Visual Communication and Image Representation, 34:187 203, 2016.

[9] 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 9199, 2015.

[10] M. Teutsch and W. Kruger. Robust and fast detection of moving vehicles in aerial videos using sliding windows. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 2634, 2015.

[11] S. Tuermer, F. Kurz, P. Reinartz, and U. Stilla. Airborne vehicle detection in dense urban areas using hog features and disparity maps. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 6(6):23272337, 2013.

[12] L. Zhang, L. Lin, X. Liang, and K. He. Is faster r-cnn doing well for pedestrian detection? In European Conference on Computer Vision, pages 443457. Springer, 2016.

[13] Lars Wilko Sommer, Toblas Schuchert, Jurgen Beyerer. Fast Deep Vehicle Detection in Aerial Images in IEEE Winter Conference on Applications of Computer Vision, 2017

[14] Leitloff, J.; Rosenbaum, D.; Kurz, F.; Meynberg, O.; Reinartz, P. An Operational System for Estimating Road Traffic Information from Aerial Images. Remote Sens. 2014, 6, 1131511341.

[15] Moranduzzo, T.; Melgani, F. Automatic Car Counting Method for Unmanned Aerial Vehicle Images. IEEE Trans. Geosci. Remote Sens. 2014, 52, 16351647.

[16] Chen, Z.; Wang, C.; Luo, H.; Wang, H. Vehicle Detection in High-Resolution Aerial Images Based on Fast Sparse Representation Classification and Multiorder Feature. IEEE Trans. Intell. Transp. Syst. 2016, 17, 22962309.

[17] Cheng, H.Y.;Weng, C.C.; Chen, Y.Y. Vehicle detection in aerial surveillance using dynamic Bayesian networks. IEEE Trans. Image Process. 2012, 21, 21522159.

[18] Shao, W.; Yang, W.; Liu, G.; Liu, J. Car detection from high-resolution aerial imagery using multiple features. IEEE Int. Geosci. Remote Sens. Symp. 2012, 53, 43794382.

[19] Kluckner, S.; Pacher, G.; Grabner, H.; Bischof, H. A 3D Teacher for Car Detection in Aerial Images. In Proceedings of the IEEE 11th International Conference on Comuter Vision, Rio de Janeiro, Brazil, 1421 October 2007; pp. 18.

[20] Kembhavi, A.; Harwood, D.; Davis, L.S. Vehicle detection using partial least squares. IEEE Trans. Pattern Anal. Mach. Intell. 2010, 33, 12501265.

[21] Chen, Z.; Wang, C.; Wen, C.; Teng, X. Vehicle Detection in High-Resolution Aerial Images via Sparse Representation and Superpixels. IEEE Trans. Geosci. Remote Sens. 2015, 54, 114.

[22] Cheng, G.; Zhou, P.; Han, J. Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images. IEEE Trans. Geosci. Remote Sens. 2016, 54, 74057415.

[23] Van de Sande, K.E.A.; Uijlings, J.R.R.; Gevers, T.; Smeulders, A.W.M. Segmentation as selective search for object recognition. In Proceedings of the International Conference on Computer Vision, Barcelona, Spain, 613 November 2011; pp. 18791886.

[24] Alexe, B.; Deselaers, T.; Ferrari, V. What is an object? In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 1318 June 2010; pp. 7380.

[25] Uijlings, J.R.R.; Sande, K.E.A.V.D.; Gevers, T.; Smeulders, A.W.M. Selective Search for Object Recognition. Int. J. Comput. Vis. 2013, 104, 154171.

[26] Kuo, W.; Hariharan, B.; Malik, J. DeepBox: Learning Objectness with Convolutional Networks. In Proceedings of the IEEE International Con-

ference on Computer Vision, Los Alamitos, CA, USA, 713 December 2015; pp. 24792487.
[27] Tuermer, S.; Kurz, F.; Reinartz, P.; Stilla, U. Airborne Vehicle Detection in Dense Urban Areas Using HoG Features and Disparity Maps. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 2013, 6, 23272337.
[28] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 2015, 38, 1.
[29] Mining Tianyu Tang, Shilin Zhou *, Zhipeng Deng, Huanxin Zou and Lin Lei. Vehicle Detection in Aerial Images Based on Region Convolutional Neural Networks and Hard Negative Example, 2016
[30] A Robust Deep-Learning Based Detector for RealTime Tomato Plant Diseases and Pests Recognition Sensors, 2017
[31] Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K. Speed/accuracy trade-offs for modern convolutional object detectors, 2017
[32] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the European Conference on Computer VisionECCV, Amsterdam, The Netherlands, 816 October 2016; pp. 2137.
[33] Everingham, M.; Van Gool, L.; Williams, C.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. Int. Comput. Vis. 2010, 88, 303338.
[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255.
[35] A. Torralba, A. A. Efros, Unbiased look at dataset bias, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 1521-1528.
[36] J. Ponce, T. L. Berg, M. Everingham, D. A. Forsyth, M. Hebert, S. Lazebnik, M. Marszalek, C. Schmid, B. C. Russell, A. Torralba, et al., Dataset issues in object recognition, in: Toward category-level object recognition, Springer, 2006, pp. 29-48.
[37] P. Carbonetto, G. Dorko, C. Schmid, H. Kuck, N. De Freitas, Learning to recognize objects with little supervision, International Journal of Computer Vision 77 (2008) 219-237.
[38] J. Gleason, A. V. Nefian, X. Bouyssounousse, T. Fong, G. Bebis, Vehicle detection from aerial imagery, in: IEEE International Conference on Robotics and Automation, 2011, pp. 2065-2070.
[39] U. Stilla, E. Michaelsen, U. Soergel, S. Hinz, H. Ender, Airborne monitoring of vehicle activity in urban areas, International Archives of Photogrammetry and Remote Sensing 35 (2004) 973-979.
[40] F. Tanner, B. Colder, C. Pullen, D. Heagy, M. Eppolito, V. Carlan, C. Oertel, P. Sallee, Overhead imagery research data set: an annotated data library and tools to aid in the development of computer vision algorithms, in: Proceedings of IEEE Applied Imagery Pattern Recognition Workshop, 2009, pp. 1-8.
[41] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. page 379-387. NIPS, 2016.
[42] RussiaToday. RT. YouTube, YouTube, www.youtube.com/channel/UCpwvZwUam- URkxB7g4USKpg.
[43] Multi-View Multi-Class Detection Dataset — CVLAB, 4 Mar. 2013, cvlab.epfl.ch/data/multiclass.
[44] RC RC RC! YouTube, YouTube, www.youtube.com/channel/UCOM2W7YxiXPtKobhrYasZDg.
[45] modellkran1 RC LIVE ACTION YouTube, www.youtube.com/channel/UCT4l7A9S4ziruX6Y8cVQRMw.
[46] bunterfisch.YouTube, YouTube, www.youtube.com/channel/UCH6AYUbtonG7OTskda1_slQ
[47] Microsoft. Microsoft/VoTT GitHub, github.com/Microsoft/VoTT.
[48] Zhang, et al. Deep Residual Learning for Image Recognition [1402.1128] Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition, 10 Dec. 2015, arxiv.org/abs/1512.03385.
[49] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.
[50] Sergey, et al. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [1402.1128] Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition, 2 Mar. 2015, arxiv.org/abs/1502.03167.
[51] Huber, Peter J. Robust Estimation of a Location Parameter Communications in Mathematical Physics, Springer-Verlag, projecteuclid.org/euclid.aoms/1177703732.