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
†Deep Learning Laboratory, National Center of Artificial Intelligence (NCAI), Islamabad, Pakistan
muhammad.shehzad@seecs.edu.pk

***Abstract*—Detection and identification of military vehicles from aerial images is of great practical interest particularly for defense sector as it aids in predicting enemys move and hence, build early precautionary measures. Although due to advancement in the domain of self-driving cars, a vast literature of published algorithms exists that use the terrestrial data to solve the problem of vehicle detection in natural scenes. Directly translating these algorithms towards detection of both military and non-military vehicles in aerial images is not straight forward owing to high variability in scale, illumination and orientation together with articulations both in shape and structure. Moreover, unlike availability of terrestrial benchmark datasets such as Baidu Research Open-Access Dataset etc., there does not exist well-annotated datasets encompassing both military and non-military vehicles in aerial images which as a consequence limit the applicability of the state-of-the-art deep learning based object detection algorithms that have shown great success in the recent years. To this end, we have prepared a dataset of low-altitude aerial images that comprises of both real data (taken from military shows videos) and toy data (downloaded from YouTube videos). The dataset has been categorized into three main types, i.e., military vehicle, non-military vehicle and other non-vehicular objects. In total, there are 15,086 (11,733 toy and 3,353 real) vehicle images exhibiting a variety of different shapes, scales and orientations. To analyze the adequacy of the prepared dataset, we employed the state-of-the-art object detection algorithms to distinguish military and non-military vehicles. 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.**

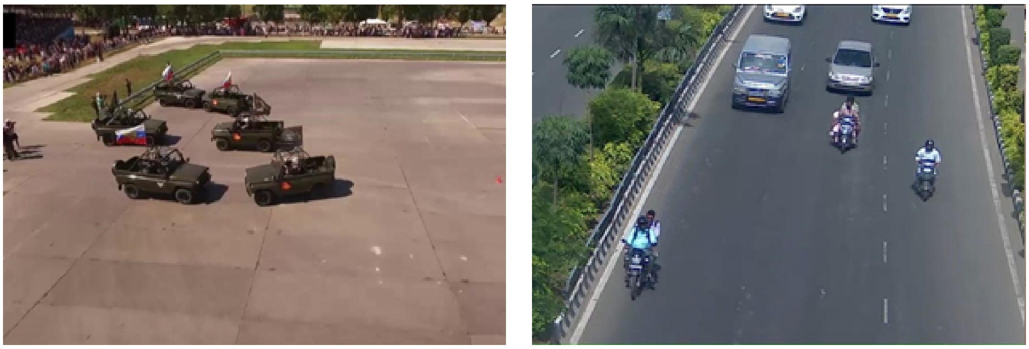
***Index Terms*—vehicle detection, vehicle classification, surveillance, military vehicle, security, military vehicle detection**

I. Introduction

Detection of military vehicles is vital for law enforcement and defense applications including surveillance, reconnaissance, security, tracking etc. These applications typically require accurate detection and distinction between military and non-military vehicles in an image. Developing an automatic algorithm that can potentially distinguish military and non-military vehicles can not only significantly reduce the work load of security personnel, but also paves the way for further high-level cognition to intelligently formalize the future course of action. Conventionally, the developed approaches aiming to solve the vehicle detection problem in aerial images focus on non-military vehicle types [3], [4], [7–9]. They typically rely on using a sliding window approach composed of hand-crafted feature extraction followed by a classifier or a cascade of classifiers. For instance, Liu and Mattyus [3] detected vehicles with two attributes (orientation and type) on aerial images using such a cascaded classifier. To localize the vehicles, it employs a fast binary detector in a soft-cascade structure whose output is fed as an input to a multiclass classifier for estimation of orientation and type of the vehicle. Tuermer et al. [13] employed a series of processing steps to extract potential vehicular regions that are later classified using a histogram of oriented features. Cheng et al. [10] performed pixel-wise dynamic Bayesian network based classification to detect vehicles for an aerial surveillance application using color and edge features. Shao et al. [11] utilized an interactive bootstrapping approach with multiple image descriptors such as histogram of gradients, local binary pattern and opponent histogram to train an intersection kernel support vector machine. Non-maximum suppression is later used to eliminate false detected vehicles.

Recently deep neural network based architectures, e.g., convolution neural networks (CNN), have shown great success in object recognition and detection tasks. They have a powerful, discriminative feature extraction and representation ability which allows to replace hand-crafted features with deep features specific for the intended task. Although literature on military vehicle detection from optical aerial imagery is quite sparse, several researchers have proposed deep architectures to detect non-military vehicles in aerial imagery. R-CNN based detection methods have performed well in nature scene images [12]. Computational cost for training and testing was significantly reduced by Fast R-CNN [2] and Faster R-CNN [6]. They achieved good results on common detection benchmark datasets. In these techniques, only one convolutional feature map is shared for the entire image rather than computing convolutional features separately. Single Shot Detector (SSD) [15] solves the object detection problem by avoiding proposal generation and saves computational time by encapsulating the process into a single network.

While deep neural network based architectures perform very well in terms of accuracy but generally require a huge amount of training data which is often not available. Specifically in the case of military vehicles, no such dataset exists up to the best of our knowledge. To this end, in this paper, we have presented a military vehicle dataset. It is composed of 13 classes which are sub-divided in two categories (vehicle and



(a) Military Vehicles

(b) Non-Military Vehicles

Fig. 1. Low-altitude aerial images of real vehicles taken from RPTLY YouTube videos [24] and our collected dataset.

non-vehicle). The vehicle category is further subdivided into military and non-military vehicle types. We evaluated the state-of-the-art object detection algorithms including Faster Region based CNN (Faster-RCNN), Recurrent Fully CNNs (R-FCN) and Single Shot MultiBox Detector (SSD) on the prepared dataset due to their remarkable performance on generic object detection in natural scenes.

Fig. 2. Vehicles from PASCAL VOC 2012 dataset [1].

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

II. Existing Datasets for Vehicle Detection

Modern approaches in deep learning need annotated training data. In addition, comparison is required on common benchmark datasets to establish the most suitable approach for solving a particular problem. Table I represents summary of datasets.

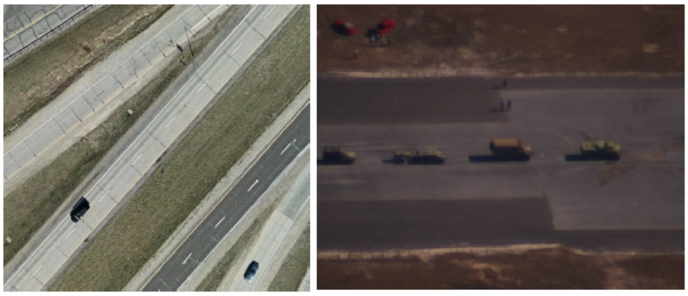


Fig. 3. Two sample images from OIRDS Dataset [22].

Our focus in this work is on vehicle detection and more specifically on military vehicle detection. Dataset from PASCAL VOC challenge [1], contains everyday life objects. PASCAL VOC dataset consists of 20 classes split into train, validation and test sets. Among other classes, one of the target classes in the PASCAL VOC challenge is Vehicle. Some vehicles from PASCAL VOC dataset are shown in Figure 2. ImageNet dataset [16] have more than 14 million images and it is generally used for object detection purposes. However, it is not designed to accommodate aerial images required for surveillance and security purposes.

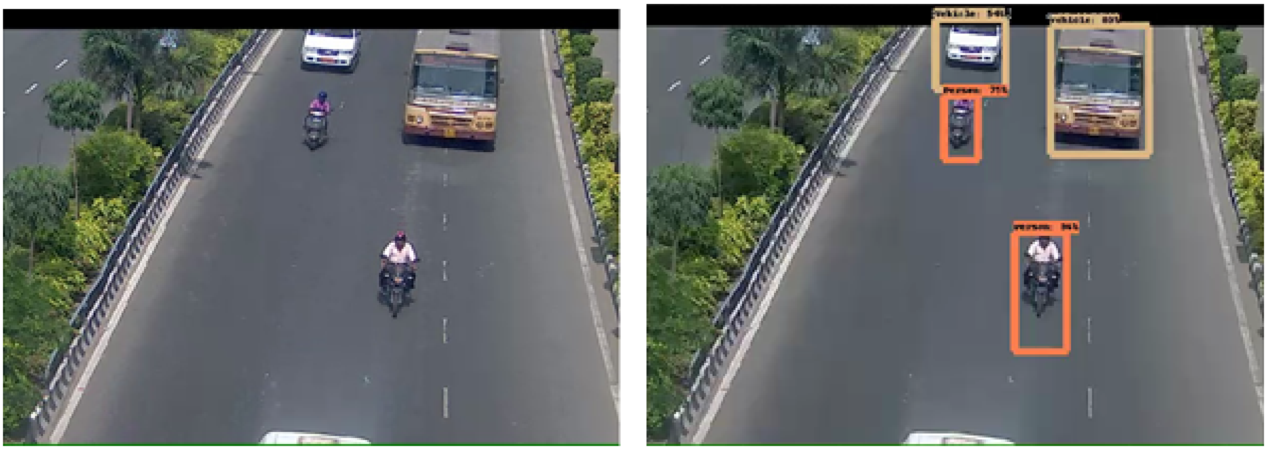
Available vehicle databases mostly contain vehicles with ground view; e.g. INRIA Car dataset [19] and the ones presented in [17], [18]. Work on target detection done in [20], [21] uses aerial imagery but unfortunately the dataset is not publicly available.

Publicly available dataset OIRDS (Overhead Imagery Research DataSet) [22] contains 180 vehicles in 900 annotated

TABLE I

Summary of existing databases for object detection

Database	Classes	# Instances	Folds	# Images
PASCAL [1]	20		train / val / test	>10,000
ImageNet [16]	21,841		train / val / test	>14,000,000
OIRDS [22]	4		No cut	900
VEDAI [5]	9	2,950	train / test	1,268
3K Vehicle Detection [3]	2	14,235	No cut	20
Our Proposed Dataset	**13**	23,097	**train / val**	**15,086**



(a) Input image containing vehicles

(b) Output image depicting recognition of the vehicles

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

images. A few images from that dataset are shown in 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 purpose.

III. Dataset Preparation

A. Challenges

Images in datasets, e.g. PASCAL VOC 2007 [1], in general are composed of only one or a few objects that occupy a high portion of the image as compared to aerial images. Aerial Images may contain multiple objects with varying sizes occupying a relatively small pixel-wise area in the image. Currently the publicly available datasets like DLR 3K Munich Vehicle Aerial Image Dataset [3] and the Vehicle Detection in Aerial Imagery (VEDAI) dataset [5], doesn’t fulfill our requirement since they do not contain any military vehicle. 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 both military vehicles and non-military vehicles with varying backgrounds. All experiments in this paper are performed on our proposed dataset.

B. Data Collection

Our collected dataset is composed of images with several types of vehicles. We have 13 classes in total. Eleven classes fall in the Vehicle category while two classes fall in the Non-Vehicle category. Vehicle category further splits into Military and Non-Military vehicle categories. The images are collected from YouTube videos consisting of real vehicles as well as toy vehicles. The real vehicles dataset was generated through RPTLY YouTube videos [24] and through publicly available EPFL Dataset [25], whereas toy vehicle images was generated from videos by RC Military toy YouTube channel [26] and other channels [27], [28]. Our dataset is composed of Images with various resolutions having a top-view angle (low-altitude) of vehicles. The collected images have a few categories of military vehicles in cluttered environment, which mimics real-world application scenarios. These conditions help the trained algorithms to more accurately identify the type and the category of the vehicle in practical settings.

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

TABLE II
Details of the variations in image dimensions and images count in our proposed dataset.

# Images	15,086 (11,733 Toy images + 3,353 Real vehicle images)
Dimensions	$1280 \times 720, 1280 \times 692, 450 \times 300$, (EPFL Data set) 360×288
Shape Resizer	1024×600

C. Data Annotation

Starting from the videos containing military vehicle, we used VOTT tool [29] to annotate the videos frame by frame in order to generate our proposed dataset. It generated annotations in PASCAL VOC format for 11 types of vehicles. Out of the total 15,086 images in our dataset, 11,733 are extracted from toy videos and 3,353 are from real videos. The collected data set is annotated for two categories of vehicles (Military and Non-Military). The images in our dataset contain multiple objects belonging to multiple classes. The number of each type of vehicles in our training dataset is shown in Table III. We have 13 classes which are split into two main categories: Vehicle and Non-Vehicle category. There are a total of 15,086 images that were manually labeled with bounding box and type of each object of interest present in them.

IV. Experimental Setup

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

(a)

(b)

Fig. 5. A few samples of aerial images taken from toy videos of RC YouTube channel [26], [27].

A. Architectural configuration

1) Feature extractors: For our experiments, we considered three feature extractors. Resnet-101 [30], which won competitions of ILSVRC and COCO 2015 (classification, detection and segmentation). We also used Inception v2 [31], which set the state-of-the-art in ILSVRC 2014 classification and detection challenges. Its network employs 'Inception units' to increase the depth and the width of a network without increasing the computational cost. Recently proposed Inception-Resnet (v2) [32], combines the optimization benefits by residual connections with the computation efficiency of Inception units.

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

3) Location loss: For all of our experiments, following [2], [6], [15], we used the Smooth L1 loss function [33].

4) Training and hyper parameter tuning: For Faster RCNN we used a batch size of two (because models were trained using images with different dimensions). For SSD and R-FCN, we used a batch size of four (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 were set to 600 and 1024 respectively.

B. Low-Altitude Aerial Imagery Datasets

as shown in Figure 6. The images in our dataset contain different sizes of objects with varying backgrounds. The main characteristics of our dataset are 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 two primary annotated categories i.e. Vehicle (11 classes) and Non-Vehicle (two classes). Vehicle category is further sub-divided into two categories i.e. Military vehicle and Non-Military vehicle. Our dataset contains 15,086 images collected from YouTube videos. For performing experiments, we divided the dataset into 83% training set and 17% validation set. Performance of a few classes is not good due to their limited number of annotations and images. Detail of categories used can be seen in Table III.

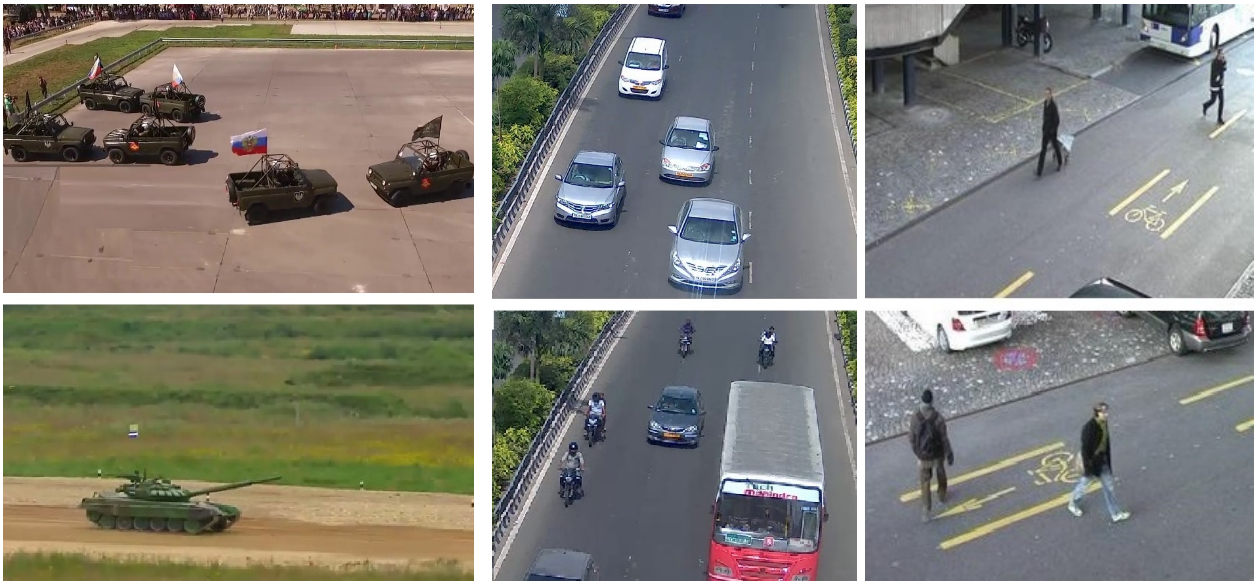
C. Benchmarking procedure

Training was performed on a machine having an Intel Core i7-7700K processor with two NVIDIA Titan-X- GPUs having 12 GB memory each. The operating system was Linux Ubuntu 16.04 LTS. For performance comparison of the 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

TABLE III

THE NUMBER OF INSTANCES OF THE 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	
# of instances	5,472	1,724	651	1,039	1,796	231	14	6,291	5,879



(a) Military vehicle category

(b) Non-military vehicle category

(c) Non-vehicle category

Fig. 6. Our collected dataset comprises of three categories (taken from RPTLY YouTube videos [24] and our collected dataset).

model was fined-tuned on collected data (8,476 Images) till 200K steps (1 epoch). Then, it was further fine-tuned on the same data till 500K steps (two epochs). Afterwards, we finetined ckpt-500K on new data (15,086 images) till 800K steps (three epochs).

TABLE IV
THE DETAILS OF COMMON CONFIGURATION PARAMETERS FOR TRAINING TILL 250K STEPS WITH ALL THREE ARCHITECTURES.

Parameters	**Values**
Initial Learning Rate	0.003 (0.004 for SSD)
num_epochs	1
Batch Size	4 (2 for Faster-RCNN)
num_hard_examples	4,000
shuffle	True
num_steps	0K—150K—250K—380K

D. Implementation Details

As the training set was of a 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 architectures. TensorFlow Object Detection API [14] was used for training and evaluation. In order to achieve better results, we fine-tuned values for hyper-parameters as shown in Table V. We evaluated Mean Average Precision (MAP) on different training steps. As shown in Table VI, the results on training ckpt-500K are better than those achieved by using ckpt-200K. If Intersection-over-Union (IOU) ratio is bigger than 0.5 w.r.t. the ground truth box, the candidate region is selected as a positive sample. Apart from this, we also analyzed the performance of three state of the architectures using our dataset.

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

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

Parameters	**Values**
Initial Learning Rate	0.004
num_epochs	1
Batch Size	16
num_hard_examples	3,000 / 3,000 / 3,500 / 4,000
shuffle	True
num_steps	0K—200K—500K—800K

V. RESULTS AND EVALUATION

We have evaluated the state-of-the-art object detection methods on our collected dataset. We have selected Faster R-CNN [6], R-FCN [23] and SSD [15] as our benchmark testing algorithms for their good performance on general object detection. The corresponding backbone networks are Inception-Resnet (v2) [32] for Faster R-CNN, ResNet-101 [30] for R-FCN and Inception V2 [31] for SSD .

A. Quantitative Results

Three state of the art architectures and feature extractors are evaluated on the basis of IoU and the average precision (AP) as introduced in the PASCAL VOC Challenge [1].

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

In Equation 1, **A** represents the ground-truth bounding box collected in the annotation while the predicted result is represented by **B**. IoU measure is used for evaluation of accuracy of an object detector. If the calculated IoU is greater than the threshold value then the predicted result is a true positive, else it is a false positive. The ratio of true positives to the total detections determines the accuracy of the network. Initially, we trained a model with a small amount of data and kept on increasing the training data. The experiment was performed using the same architecture (SSD with feature extractor Inception v2). The initial pre-trained model (trained on COCO dataset) was fined-tuned on our collected data (8,476 Images) till 200K iterations. Afterwards, it was further fine-tuned till 500K (300K times more) iterations on the same data. Finally, we increased the amount of training data and further fine-tuned it on the new Data (15,086 images which included the previous data as well) till 800K iterations. Table VI shows that by increasing the training data and fine-tuning a pre-trained model, the algorithms are able to improve the weighted average precision for each class. When the model was further fine-tuned from 500K iteration till 800K iterations, the weighted average precision did not significantly change (slightly decreased due to over-fitting).

TABLE VI
THE RESULTS OF THE TOP PERFORMING CLASSES WHEN THE TRAINING WAS DONE TILL 800K ITERATIONS USING SSD ARCHITECTURE.

	Average precision (%)		
Feature Extractor→	**Inception v2**		
Total Iterations → Class ↓	***200K***	***500K***	***800K***
Tank	91.21	93.58	94.07
HEMTT	83.31	90.33	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

	Weighted average precision (%)		
Total Iterations → Category ↓	***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
THE RESULTS OF THE TOP PERFORMING CLASSES WHEN THE TRAINING WAS DONE TILL 380K ITERATIONS USING THREE STATE OF THE ART ARCHITECTURES.

	Average precision (%)		
Architecture →	***Faster R-CNN***	***R-FCN***	***SSD***
Feature Extractor → 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

	Weighted average precision (%)		
Architecture → Category ↓	***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 experiments to compare the results of the three architectures using our latest dataset having 15,086 images. The detection results achieved after performing the training till 380K iterations show that Faster R-CNN performed better as compared to RFCN and SSD. The results are given in Table VII.

B. Qualitative Results

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

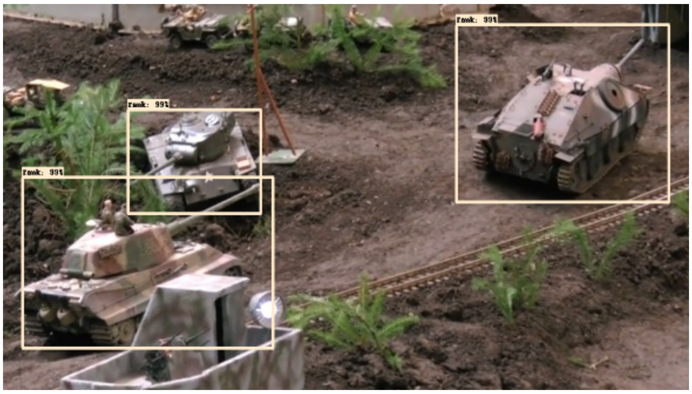


Fig. 7. The detection results on military vehicles (taken from toy videos of RC YouTube channel [26], [27]).

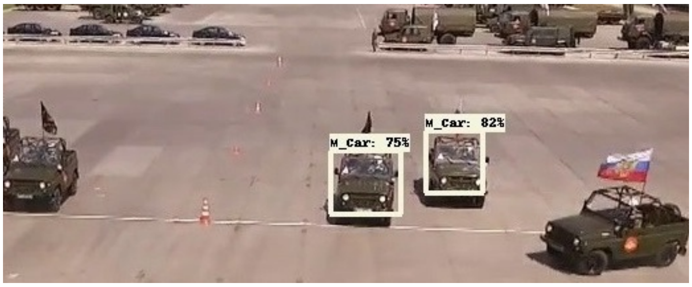


Fig. 8. An example of miss-classification in real military vehicles taken from RPTLY YouTube videos [24].

VI. Analysis of the success and failure cases

Overall, it was observed that two classes i.e. Tank and HEMTT, performed well in both of our experiments. The reason behind this is that both classes have more training data as compared to the other classes. The details of our proposed dataset are given in Table III. During analysis, we observed that the presented system had good performance on test cases and on unseen data, but there were difficulties for some classes for which the training set size was small. It performs well on tanks in unseen data as compared to other military vehicles because the ratio of tank images in the training data is more than the other classes. Figure 8 shows the results of extensive fine-tuning (detail in Table VI). The accuracy of the model drops on the data, on which it was previously performing better. This is also happening because of the fewer number of training samples. Figure 8 and Figure 9 show a few cases of miss-detections and wrong classifications.

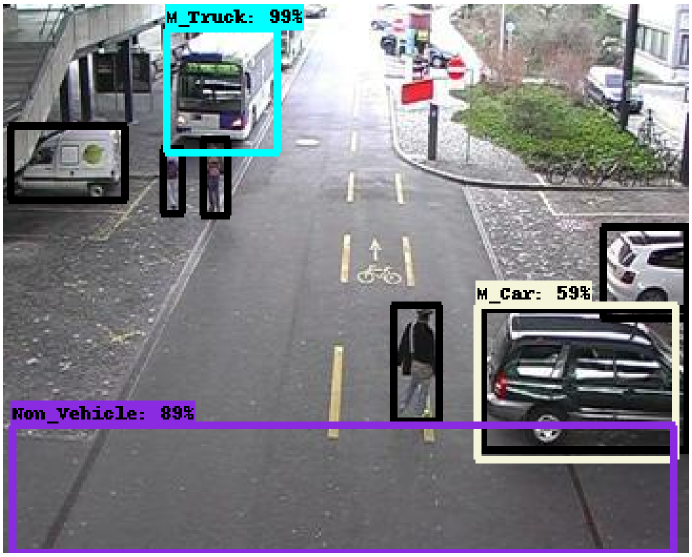


Fig. 9. A sample taken from EPFL dataset [25] on which the classification results were incorrect.

VII. Conclusion

We proposed a framework based on deep learning for military vehicle detection and classification from aerial images. This system detects the class and location of military and non-military vehicles in the captured aerial images. A new dataset of military vehicles has been prepared by collecting images captured from real vehicle videos (Military shows) as well as toy vehicle videos (RC YouTube videos). 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 a comparative analysis between different deep-learning architectures (with feature extractors combination). Experimental results demonstrate that by applying deep learning based detector on our proposed dataset, it is able to reliably detect two different categories of vehicles (with 11 classes). In addition, two 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 the current detection results and extend the idea of Military-Vehicles recognition to work on other surveillance and security related objects. We demonstrated the performance of three state-of-the-art architectures for military-vehicle detection purposes. For SSD architecture, we systematically evaluated per category detection improvement by tuning its hyper-parameters and increasing the number of 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, it is the first work focusing on military vehicle detection from aerial images.

VIII. Acknowledgement

We are grateful to the newly established National Center for Artificial Intelligence, Islamabad, Pakistan for supporting this work.

References

[1] 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.

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

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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

[15] 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.

[16] 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.

[17] A. Torralba, A. A. Efros, Unbiased look at dataset bias, in: IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 1521-1528.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object Detection via Region-based Fully Convolutional Networks. page 379-387. NIPS, 2016.

[24] RussiaToday. RT. YouTube, YouTube, www.youtube.com/channel/UCpwvZwUam- URkxB7g4USKpg.

[25] Multi-View Multi-Class Detection Dataset — CVLAB, 4 Mar. 2013, cvlab.epfl.ch/data/multiclass.

[26] RC RC RC! YouTube, YouTube, www.youtube.com/channel/UCOM2W7YxiXPtKobhrYasZDg.

[27] modellkran1 RC LIVE ACTION YouTube, www.youtube.com/channel/UCT4l7A9S4ziruX6Y8cVQRMw.

[28] bunterfisch.YouTube, YouTube, www.youtube.com/channel/UCH6AYUbtonG7OTskda1_slQ

[29] Microsoft. Microsoft/VoTT GitHub, github.com/Microsoft/VoTT.

[30] 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.

[31] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[32] 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.

[33] Huber, Peter J. Robust Estimation of a Location Parameter Communications in Mathematical Physics, Springer-Verlag, projecteuclid.org/euclid.aoms/1177703732.