

EAGLE: Large-scale Vehicle Detection Dataset in Real-World Scenarios using Aerial Imagery

Seyed Majid Azimi^{*†}, Reza Bahmanyar^{*}, Corentin Henry^{*}, and Franz Kurz^{*}

^{*}Remote Sensing Technology Institute, German Aerospace Center (DLR), Wessling, Germany

[†]Department of Aerospace, Aeronautics and Geodesy, Technical University of Munich, Munich, Germany

Corresponding author: {seyedmajid.azimi}@dlr.de



Sample aerial image from the EAGLE dataset with partial snow cover and its overlaid annotation taken in the early morning.

Abstract—Multi-class vehicle detection from airborne imagery with orientation estimation is an important task in the near and remote vision domains with applications in traffic monitoring and disaster management. In the last decade, we have witnessed significant progress in object detection in ground imagery, but it is still in its infancy in airborne imagery, mostly due to the scarcity of diverse and large-scale datasets. Despite being a useful tool for different applications, current airborne datasets only partially reflect the challenges of real-world scenarios. To address this issue, we introduce EAGLE (oriEnted vehicle detection using Aerial imaGery in real-worLd scEnarios), a large-scale dataset for multi-class vehicle detection with object orientation information in aerial imagery. It features high-resolution aerial images composed of different real-world situations with a wide variety of camera sensor, resolution, flight altitude, weather, illumination, haze, shadow, time, city, country, occlusion, and camera angle. The annotation was done by airborne imagery experts with small- and large-vehicle classes. EAGLE contains 215,986 instances annotated with oriented bounding boxes defined by four points and orientation, making it by far the largest dataset to date in this task. It also supports researches on the haze and shadow removal as well as super-resolution and in-painting applications. We define three tasks: detection by (1) horizontal bounding boxes, (2) rotated bounding boxes, and (3) oriented bounding boxes. We carried out several experiments to evaluate several state-of-the-art methods in object detection on our dataset to form a baseline. Experiments show that the EAGLE dataset accurately reflects real-world situations and correspondingly challenging applications.

I. INTRODUCTION

Automatic vehicle detection based on aerial imagery is crucial for a variety of applications such as large-scale traffic monitoring, parking lot utilization, urban planning, disaster management, as well as search and rescue missions. Aerial images, with their wide field of view, provide valuable information over large open areas in a short time [1]. Due to the steep rise in the number of vehicles, traffic monitoring and management has become tremendously more complex,

especially in urban areas. The major socio-economic impacts of the traffic-related problems such as air pollution, time loss in traffic jams, and health issues have increased the demand for developing novel automatic algorithms and adequate traffic data [2]. It has been shown that vehicle detection algorithms based on aerial imagery can provide frequent and cost-efficient information about the location, number, and the types of vehicles in different traffic scenarios such as congestion caused by infrastructure bottleneck, accidents, or even lack of parking spaces [1]. Due to the dynamic nature of traffic, the availability of large-scale information through aerial images can make traffic management more adaptive to the changing traffic conditions and help predicting infrastructure bottlenecks [3]. In disaster management, vehicle detection based on aerial imagery allows rapid localization of traffic congestion and abandoned vehicles to determine routes for effective search and rescue activities. Furthermore, in the case of natural disasters such as floods and earthquakes, aerial imagery is the most efficient means for detecting the affected vehicles [4]. Recently, a large number of studies have focused on object detection (including vehicles) in aerial imagery [5], [6], [7], [8], [9], [10]; however, despite the pronounced differences between ground and aerial images, most of the proposed methods are based on transferring object detection algorithms developed for natural-scene images to the aerial ones due to the scarcity of the large-scale aerial image datasets. For instance, to apply deep learning detection algorithms to aerial images, previous works usually relied on fine-tuning networks pre-trained on large-scale natural-scene datasets (*e.g.*, ImageNet [11], MSCOCO [12], PASCAL VOC [13]). As it can be seen in Figure 1, the scale of the objects varies widely in aerial images due to not only the differences in spatial resolution, but also in the size of objects from the same category. In addition, aerial images usually contain a large number of small



Fig. 1: Sample annotations in EAGLE: (a-b) car and trucks in purple and yellow respectively, (c-d) sunny and cloudy illumination, (e-f) cars partly occluded by buildings, (g-h) cars partly occluded by vegetation, (i-j) cars in shadowed areas, (k-l) hard to identify cars orientations, (m) difficult car example, (n) car with weak orientation, (o-p) trucks with weak orientations.

objects distributed and oriented differently over the scene (*e.g.*, from sparse density of moving vehicles in highways to tightly packed ones in parking lots). In addition, the number of the object instances in aerial images is unbalanced, from a few to thousands of objects per image.

Object detection in ground imagery owes its significant promotion to the large datasets such as MSCOCO, ImageNet, and PASCAL VOC. However, for aerial imagery, similar datasets in terms of image number and annotation details are scarce, which has highly limited the progress in developing methods for aerial images. The current available aerial image datasets *e.g.*, [14], [15], [16], [17], [18] suffer from either low number of images and annotated instances or low-quality annotations. The largest currently available aerial image dataset for object detection is DOTA [18] which comprises 2,800 images with fifteen categories and about 188,000 bounding box annotations using already processed Google Earth and satellite images; however, it contains only 43,462 vehicles. Other datasets such as TAS [14], VEDAI [15], COWO [19], DLR-3K-Munich-Vehicle [16], and UCAS-AOD [17] which mainly focus on vehicle detection also contain very limited number of annotated vehicles: TAS (1,319), VEDAI (3,270), COWO (32,716), DLR-3K-Munich-Vehicle (14,235), and UCAS-AOD (2,819). In addition to the number of instances, the inadequate diversity and complexity of the images used (*e.g.*, clear background and limited object distribution heterogeneity) in these datasets prevents them from representing real-world situations. Table I shows detailed statistics from the current major aerial image datasets for object detection. To promote research on vehicle detection including vehicle detection, counting, and tracking, we propose a new and yet largest aerial image dataset for vehicle detection in real-world aerial imagery scenarios, called *oriEnted object detection using Aerial imaGery in real-worLd scEnarios (EAGLE)*.

Altogether, the main contributions of this paper are:

- EAGLE, which is to the best of our knowledge the largest aerial image dataset for vehicle detection and the first dataset of its kind addressing real-world scenarios.
- Its high-quality annotations can contribute to the development and evaluation of practical airborne vehicle detection systems as well as haze, shadow, in-painting and super-resolution applications.
- Benchmarks of state-of-the-art object detection algorithms as baseline for future works by defining benchmarks for all three possible detection possibilities and two dataset split approaches.

II. EAGLE DATASET

The EAGLE dataset consists of 8,820 aerial images with size of 936×936 px, acquired during several flight campaigns carried out between 2006 and 2019 in various time of day and year with different weather and illumination conditions. The images were taken under different traffic conditions and situations involving vehicles such as motorways, urban/rural areas, industrial districts, floods, wildfires, earthquakes, as well as search and rescue missions over multiple locations in five countries (see Figure 2). The images contain a large diversity of vehicle orientation angle and number of objects per image as shown in Figure 3 with a higher number of vehicle instances compared to previous datasets (see Figure 4). Figure 5 showcases some example image patches from the dataset. We acquired the images using a camera system comprised of three standard DSLR cameras (Canon EOS cameras) mounted on an airborne platform with different looking angles, a nadir-looking (top-down vertical) and two side-looking cameras. According to the conditions of the flight campaigns, the camera setups such as aperture size, image size, and ISO were adjusted differently. The platform was installed either

TABLE I: Comparison between EAGLE and datasets for object detection in aerial images. BB is short for bounding box. One-dot refers to annotations with only the center coordinates of an instance provided. Fine-grained categories are not taken into account. For example, EAGLE features 2 different categories with additional difficulty flags with respect to the class and orientation.

Datasets	# Vehicle Instances	# Vehicle Categories	# All Categories	# Images	# All Instances	Image Width (px)	Annotation Approach	Year
TAS [14]	1,319	1	1	30	1,310	792	HBB	2008
NWPU-VHR10 [20]	232	1	10	800	3,775	1000	HBB	2014
VEDAI [15]	3,270	6	9	1,210	3,640	1024	OBB	2015
UCASAOD [17]	2,819	1	2	910	6,029	1280	HBB	2015
DLR-3K-Vehicle [16]	14,232	2	2	20	14,235	5616	OBB	2015
COWC [19]	32,716	1	1	53	32,716	2000-19,000	One-Dot	2016
HRSC2016 [21]	0	0	1	1,070	2,976	1000	OBB	2016
RSOD [22]	0	0	4	976	6,950	1000	HBB	2017
DOTA [18]	43,462	2	15	2,806	188,282	300-4000	RBB	2017
EAGLE (ours)	215,986	2	2	8,280	215,986	936	OBB	2020

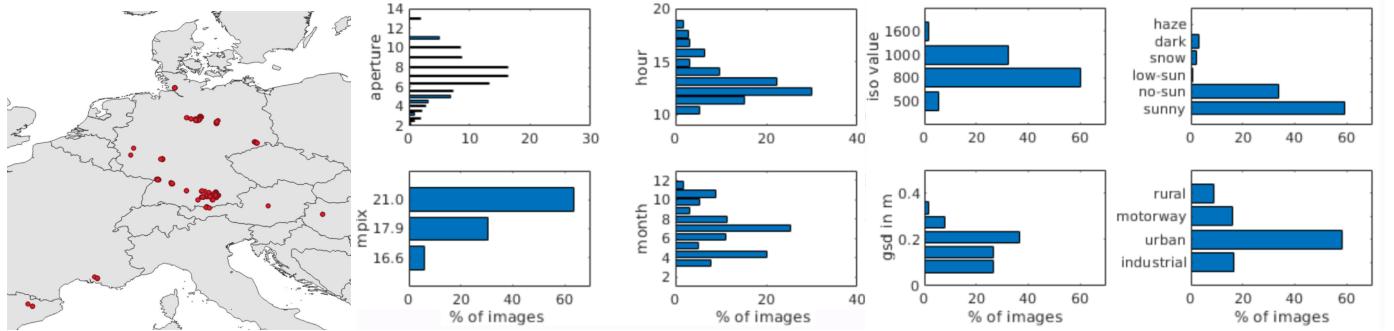


Fig. 2: Distribution of image acquisition locations over central Europe, as well as the statistics on camera parameters, image, and scenery properties.

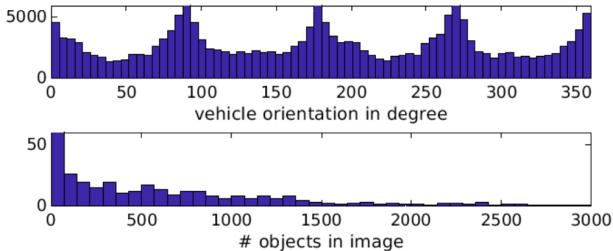


Fig. 3: Statistics of annotated vehicles with respect to vehicle orientation (top) and instances per image (bottom).

on an airplane or on a helicopter flying at altitudes between 300 m and 3000 m, resulting in a range of *ground sampling distances (GSDs)*, or spatial resolution, from 5 cm to 45 cm per pixel. The images were taken from early in the morning until the evening in various weather conditions (*e.g.*, sunny, snowy, rainy, and foggy) with different illumination levels. Altogether, the variability in image parameters and scenes allows our dataset to cover a wide range of real-world situations involving vehicles. Figure 2 represents further statistics on the EAGLE dataset.

A. Image annotation

Taking into account the relevance of the vehicle categories for the real-world applications of aerial imagery according to experts in the domain, we decided on two main categories for

our dataset, namely small vehicles (cars, vans, transporters, SUVs, ambulances, police cars) and large vehicles (trucks, large-trucks, minibuses, buses, firefighter trucks, construction vehicles, trailers). The annotation contains the coordinates of all four vehicles corners having right angle between sides as well as orientation degree between 0° to 360° indicating the angle of vehicle head with respect to the trigonometric circle. Table I shows a comparison between EAGLE and other existing aerial imagery datasets for vehicle detection. The EAGLE contains 215,986 annotated vehicles, ranging from 1 to 3,567 annotations per image in all possible orientations (see Table II), making it the largest aerial image dataset for vehicle detection by a large margin (5× more vehicle instances than in the current largest dataset). Furthermore, for each instance, the visibility condition (totally/partly/hardly visible) and orientation clarity (clear/unclear) of the vehicle were provided. Stitched images with original sizes are 345 ones of 5616 × 3744 px size. As visible in Table II, the EAGLE dataset contains 208,963 small and 7,023 large vehicles. A category-wise comparison is provided in Figure 4.

B. Annotation method

We have addressed various challenges during the annotation of the vehicles in our aerial images. Due to the diversity of the scene locations, the acquisition time, as well as the weather and illumination conditions, precise annotation of the vehicles could be a very challenging task. For example, in an

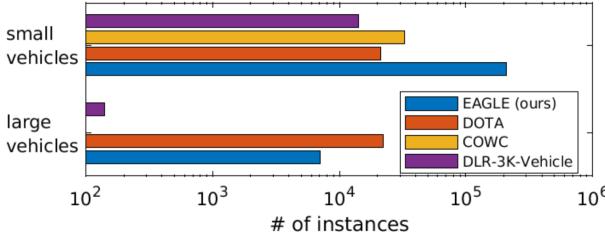


Fig. 4: Comparison between the number of annotated small and large vehicles in the EAGLE dataset and the vehicle sets of other aerial image datasets.

	Small vehicles	Large vehicles
# Annotations	208,963	7,023
# Weak orientation	311	10
# Partly visible	18,188	184
min/max/avg objects per image	1/3,567/630	0/140/16

TABLE II: Category-wise statistics in EAGLE.

image taken over a flooded area when haze is present with low illumination or resolution, the visibility of the vehicles gets limited considerably. In addition, the occlusion due to other objects or strong shadow could cause difficulties in finding the vehicles. Furthermore, spotting vehicles in large aerial images of remote places (*e.g.*, mountains) is not trivial. Moreover, categorizing the vehicles into either small or large vehicles could be sometimes tricky due to the uncertainty about the category of some borderline cases such as large transporters or buses. To ease the latter situation, we assumed the one-cabin vehicles with a width or a height smaller than a specific threshold (specified by an expert) as small vehicles and otherwise as large vehicles. We also assigned a difficulty flag for the occluded vehicles which can help to better train algorithms to overcome occlusion. Detecting the occluded vehicles is very important in real-world scenarios such as in disasters like flood when the vehicles are trapped or partially under water. In the ground imagery , objects are usually annotated by *horizontal bounding box (HBB)*, where an HBB can be defined by its top-left (TL) and bottom-right (BR) vertices, $(x_{TL}, y_{TL}, x_{BR}, y_{BR})$; or by its center point (x_c, y_c) together with the width w and height h , (x_c, y_c, w, h) . HBB is an efficient object annotation approach; however, it does not consider the objects' orientation, which can lead to imprecise outlines of arbitrary oriented objects. Moreover, HBBs considerably overlap when objects are tightly packed, which can confuse even state-of-the-art algorithms trying to distinguish them. An approach toward alleviating the limitations of HBB is using arbitrary quadrilateral bounding boxes, the so-called *rotated bounding box (RBB)* [18], which can be described by $\{(x_i, y_i), i = 1, 2, 3, 4\}$, where (x_i, y_i) are the vertex coordinates which can be with a clockwise order [18]. A specific case is a rotated rectangle when the sides make right angle with each other. Inspired by [23], [18] and the annotations in the common object detection benchmarks such as MSCOCO and PASCAL VOC, we propose a right-angle constrained *oriented bounding box (OBB)* which can be described as $\{(x_i, y_i), i = 1, 2, 3, 4; \theta\}$, where (x_i, y_i)

are the vertex coordinates and θ indicates the bounding box orientation. OBB can be also represented as (x_c, y_c, w, h, θ) , where the bounding box edges are oriented according to θ . This approach ensures the precision of the object outlines.

C. Dataset splits

We split the dataset into training, validation, and test sets based on two approaches. In the first approach, we randomly assign 1/2, 1/6, and 1/3 of the images respectively. In this case, images from similar flight campaigns can be present in both train and test sets, which makes the detection task easier and similar to DOTA. Thus, in the second approach, we split the dataset so that the images from the same flight campaigns are either in the training or test set. This approach is similar to the real-world scenarios in which there is no prior knowledge about future flight missions and their locations, weather or illumination conditions.

D. Contributions over the existing datasets

The existing datasets containing vehicle instances (*e.g.* DOTA) suffer from inconsistent or inaccurate annotations, low degree of diversity and a small number of vehicle instances, limiting their practical applications. Therefore, vehicle detection datasets such as EAGLE with thorough annotations even for tiny yet visible vehicles (see Figure 6) are lacking in the community. Moreover, EAGLE enables researchers to do research on haze and shadow removal as well as super-resolution, in-painting and instance segmentation. Our dataset is featuring major differences compared to the DOTA dataset:

- EAGLE focuses on vehicle detection in real-world and practical scenarios with images of diverse location, time, resolution, weather and illumination conditions while DOTA is a multi-class general-purpose detection and classification dataset.
- DOTA suffers from incomplete and noisy annotations (see Figure 6) especially for small vehicles [24], whereas EAGLE provides precise and comprehensive annotations (even for partially visible vehicles).
- Due to overlaps between the training and test sets in DOTA, the task is less challenging than EAGLE in which two training/test splits are proposed: (1) a random patch-based split, and (2) a more realistic and challenging campaign-based split, where the test set contains locations and adverse conditions unseen during training.

III. EVALUATION

We assess the performance of state-of-the-art object detection methods on EAGLE. For HBB object detection, we choose Cascade (Mask-)RCNN [25], Mask-RCNN [26]¹, FPN [9], Faster RCNN [27], FCOS [28]², TridentNet [29], SNIPER [30]³, R-FCN [31]⁴, YOLOv3 [32],

¹<https://github.com/facebookresearch/Detectron>

²<https://github.com/tianzhi0549/FCOS>

³<https://github.com/MahyarNajibi/SNIPER>

⁴<https://github.com/msracver/Deformable-ConvNets>



Fig. 5: Examples of annotated images (left to right, top to bottom): low sun, rural scene, industrial scene, parking space, mixed illumination, snow, mega city, mixed parking, flood scene, oblique view, highway scene, service area, suburban area, festival scene, haze, motorway. Magenta: small vehicles. Yellow: large vehicles. Cyan triangle: driving direction.



Fig. 6: High-quality EAGLE labels (left column) and incomplete DOTA labels (right column).

RefineDet [33], and SSD [34]⁵ having ResNet101 [35], ResNext101 [36], Triple-ResNeXt152, InceptionV2 [37] or VGG16 [38] backbone-networks as our baseline benchmark algorithms on the test set for their excellent performance in object detection on ground images by HBBs. Furthermore, we modify the original Cascade Mask-RCNN to detect objects with RBBs described by $\{(x_i, y_i), i = 1, 2, 3, 4\}$. We further adapt the algorithm to able to detect objects with OBBs denoted as (x_c, y_c, w, h, θ) , as θ means the vehicle head angle.

In order to evaluate the benchmark algorithms on EAGLE, we propose three different tasks including detection by HBB, RBB, and OBB. As the evaluation metric, we employ *mean average precision (mAP)* similar to PASCAL VOC. The image patches are stitched to form the original image before the evaluation step. In order to remove the redundant detected boxes in the overlapping regions as well as the patches themselves, we apply *non-maximum suppression (NMS)* with a threshold of 0.3 for HBB and 0.1 for both RBB and OBB.

A. Image splitting

In the training phase, due to the large size of the images (5616×3744 px) in the EAGLE dataset which cannot be fitted into the object detectors for the training process, we crop them into 1024×1024 px patches with a 50% overlap in a sliding window fashion, resulting in 70 patches per image leading to 12075, 4025, and 8050 patches of training, validation and test respectively. The overlaps of the patches allows keeping all the objects, even if partially clipped at image boundaries. Patches thus ending up partially outside the image are shifted back into the image window. Patch-wise predictions are stitched into full images and overlaps were merged using NMS. This process could cut some vehicles into two parts. In this case, we compute the ratio between the area covered by each part (A_i , $i = 1, 2$) and that of the complete vehicle (A_O) as

⁵https://github.com/tensorflow/models/tree/master/research/object_detection

$U_i = A_i/A_O$ similar to [18], but with the difference that we adapt the parts' ground truths to the image boundaries to have the highest intersection with the original object. After that, for $U_i \geq 0.7$, the attribute of the part remains unchanged, for $0.1 \leq U_i < 0.7$, the attribute of the part is changed to "difficult", and for $U_i < 0.1$, the part is ignored. Moreover, the part which does not include the front part of the vehicle (depicting the orientation) is assigned a "difficult" flag to its orientation attribute. For the testing step, we crop the images, but with a stride of 912 px (10% overlap to ensure the coverage of the vehicles in their full appearance as well.

B. Horizontal Bounding Boxes (HBB) baselines

We generate the ground truth for HBB by calculating the center coordinates of the minimum and maximum in x and y coordinates in the original rotated bounding box ground truth. We train the baseline algorithms with their default settings and hyper-parameters for a fair comparison. Table III shows the HBB detection results which indicates how challenging this dataset is for the-state-of the-art methods, with Cascade Mask-RCNN achieving the best performance of 39.29% mAP. SSD and Yolov3 have very low performance compared to the others. This could be due to the random crops during data augmentation suggested by [18]. Furthermore, the results depict a considerable difference between the ground-level and aerial objects concerning their size, scale and appearance.

C. Rotated Bounding Boxes (RBB) baselines

Since most of the state-of-the-art algorithms are designed for non-oriented objects, direct application of the algorithms for detecting the oriented-objects is not efficient which makes the benchmark of the existing algorithms for RBB challenging. We select and modify the Cascade Mask-RCNN [25] algorithm for predicting rotated bounding boxes, due to its accuracy on the HBB task of the EAGLE dataset. For the rest of algorithms, we train the algorithms on the HBB annotations of our dataset and test them on the RBB annotations. Cascade Mask-RCNN is composed of one *region proposal network (RPN)* and three detection and segmentation heads with thresholds $U = \{0.5, 0.6, 0.7\}$. While RBB ground truth is defined by $\{(v_{xi}, v_{yi}), i = 1, 2, 3, 4\}$ vertices, RPN generates horizontal rectangles denoted by their top-left (TL) and bottom-right (BR) vertices $RoI = (x_{TL}, y_{TL}, x_{BR}, y_{BR})$. Therefore, we adapt the ground truth to rectangles by $x_{TL} = v_{x1} = v_{x4}$, $x_{BR} = v_{x2} = v_{x3}$, $y_{TL} = v_{y1} = v_{y4}$, and $y_{BR} = v_{y2} = v_{y3}$, similar to [18]. An alternative would be using rotated RPN as mentioned in [24]. However, we try to preserve the structure of the algorithm as much as possible. In the detection heads, the output target $T = \{(t_{xi}, t_{yi}), i = 1, 2, 3, 4\}$ for each RoI and its ground truth $G = \{(g_{xi}, g_{yi}), i = 1, 2, 3, 4\}$ are defined as:

$$t_{xi} = (g_{xi} - v_{xi})/w, \quad t_{yi} = (g_{yi} - v_{yi})/h \quad (1)$$

where $w = x_{BR} - x_{TL}$ and $h = y_{BR} - y_{TL}$, similar to [39]. We consider the coordinates of each ground truth G as the object mask to prepare the mask for the segmentation

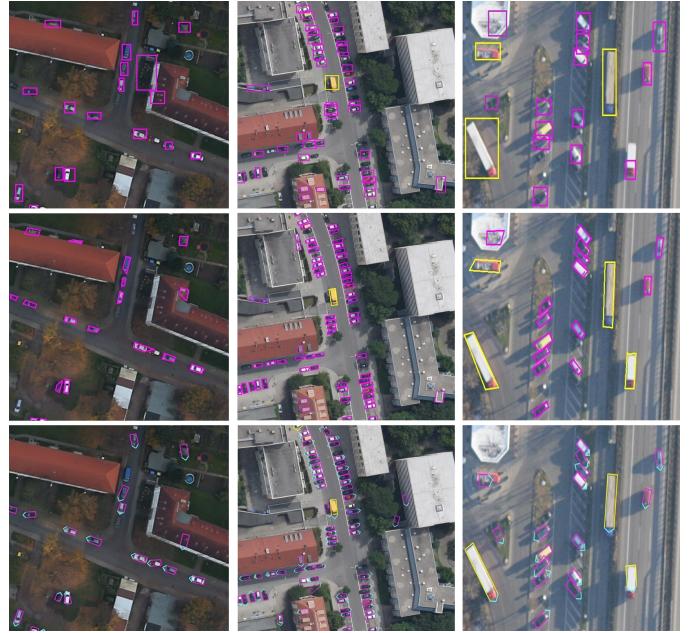


Fig. 7: Test prediction samples of Cascade Mask-RCNN trained on the EAGLE dataset. The first row is the result of horizontal bounding box (HBB), the middle row for rotated quadrilateral bounding boxes (RBB), and the bottom row is the result of oriented bounding boxes (OBB). Magenta is for small-vehicle and yellow for large-vehicle. The orientation is depicted in cyan.

head. Table III shows the results of the modified Cascade Mask-RCNN trained and tested on RBB compared with other baselines trained on HBB and tested based on RBB ground truth. We denote the modified method as *Cascade Mask-RCNN-Rotated*. The results show that by adapting the algorithm to rotated bounding box detection, we can achieve an improvement of about 7% mAP points. It also indicates that RBB task is a more difficult task than general HBB.

D. Oriented Bounding Boxes (OBB) baselines

For the benchmark based on OBB, we modify the detection heads of Cascade Mask-RCNN to predict the bounding box angles, and denote it as *Cascade Mask-RCNN-Oriented*. To this end, We regress over $T = (x_c, y_c, w, h, \theta)$ instead of $(x_{TL}, y_{TL}, x_{BR}, y_{BR})$. Other possibilities are regression over $T = \{(x_i, y_i), i = 1, 2, 3, 4; \theta\}$, or considering the clockwise order of bounding box vertices. The angle regression is defined as:

$$t_\theta = \tan(g_\theta - v_\theta), \quad (2)$$

where *tangent* function is used to ensure the periodicity of the angle regression, but other regression approaches can be considered. Similar to the Fast-RCNN [41] algorithm, we use the smooth L_1 loss for bounding box regression and Cross-entropy loss for classification. We evaluate the performance of the algorithm on the OBB task by comparing the center coordinates, angle, width and height of predicted oriented

TABLE III: Benchmark of the state of the art on the horizontal bounding box (HBB) and the rotated bounding box (RBB) detection task; mAP means mean Average Precision, higher is better. Mask-RCNN-H means trained on horizontal bounding box. Mask-RCNN-R means trained on rotated bounding box.

Method	Backbone	AP [%] (HBB)			AP [%] (RBB)		
		Mean	SV	LV	Mean	SV	LV
Yolov3 [32]	Darknet-53	20.29	30.45	10.13	13.28	21.34	5.23
SSD [34]	InceptionV2	12.06	20.67	3.45	7.31	12.34	2.28
RefineDet [33]	VGG16	22.23	32.25	12.21	14.78	22.67	6.89
R-FCN [31]	ResNet101	30.61	46.85	14.37	21.06	35.56	6.56
Faster-RCNN [27]	ResNet101	31.84	48.34	15.34	23.15	39.29	7.02
Mask-RCNN [26]	ResNet101	30.81	46.51	15.11	22.54	36.65	8.43
Cascade-RCNN [40]	ResNet101	33.49	49.65	17.34	23.58	38.97	8.19
SNIPER [30]	ResNet101	30.74	48.34	13.14	21.97	38.23	5.72
FPN [9]	ResNet101	37.10	50.76	23.45	27.11	39.78	14.45
TridenNet [29]	ResNet101	30.53	47.16	13.91	22.53	37.16	7.91
FCOS [28]	ResNeXt101	38.80	52.94	24.67	27.67	41.24	14.10
Cascade Mask-RCNN-H [25]	Triple-ResNeXt152	39.29	53.45	25.14	30.22	43.84	16.60
Cascade Mask-RCNN-R [Ours]	Triple-ResNeXt152	-	-	-	37.23	51.27	23.19

bounding box. For orientation estimation, we divide the angles in the range of $(-180, 180)$ into 16 output bins and we consider an angle prediction to be correct if it falls into the same bin as the ground truth. Cascade Mask-RCNN-Oriented achieves 43.87% mAP which is 59.45% *average precision (AP)* and 28.29% AP for small and large vehicle and with the angle accuracy of 67.34%.

E. Experimental analysis

By analyzing the results shown in Table III, we observe that the HBB detection is still challenging with respect to very small size objects, densely crowded regions, and occlusions in aerial images. In Figure 7, we provide a comparison of small and large vehicle detection methods of HBB, RBB, and OBB. As shown in Figure 7, for areas in which vehicles are parked tightly , we observe that HBB is less accurate than RBB and OBB in precise localization of vehicles in which several detection results are suppressed by NMS and other post-processing steps. Furthermore, we see that some vehicles do not have right-angle detections for the RBB task leading to mistakes in the localization while OBB does not have this issue, resulting in a better performance. Therefore OBB is the more accurate way in oriented object detection in aerial images. As for false positives, some non-vehicles objects appear similar to vehicles, confusing detectors as shown in the left column of Figure 7, showing false positives over the roofs. Also in the results of RBB in the middle column, a trash bin was detected as small vehicle. The less accuracy of the detector in large-vehicle detection compared to small-vehicle is the higher number of small-vehicle instances compared to large-vehicle ones leading to an unbalanced dataset. Also, in highly dense areas, results of both RBB and OBB are not satisfying implying the high difficulty of this task.

F. Impact of data-related factors on the performance

The smaller GSD is already known to improve performance drastically [42], [24], but requires very-high resolution image

TABLE IV: Benchmark of the best method from benchmark on the second split approach by splitting based on flight campgain. Cascae Mask-RCNN-O, -H, and -R means Cascade Mask-RCNN trained on oriented, horizontal, and rotated bounding boxes respectively.

Method	Task	mAP [%]	AP [%]	
			small-vehicle	large-vehicle
Cascade Mask-RCNN-H	HBB	33.54	50.16	16.92
Cascade Mask-RCNN-R	RBB	30.18	46.82	13.54
Cascade Mask-RCNN-O	OBB	32.02	48.13	15.91

TABLE V: Comparison of results on EAGLE and DOTA using Cascade Mask-RCNN. The comparison is based on mAP. SL and LV stand for small-vehicle and large-vehicles respectively. (scores in mAP)

Training set	Test set	Avg.	SV	LV
DOTA	DOTA	59.95	61.23	58.67
DOTA	EAGLE	28.23	38.89	17.57
EAGLE	DOTA	53.25	57.34	49.16
EAGLE	EAGLE	39.29	53.45	25.14

acquisition, which may not always be possible. Smaller size and scale can also degrade the performance. The segmentation of objects down to 2px-wide at different scales was already successfully presented [43]. Experiments on EAGLE indicates other challenges such as low-illumination, haze, shadow and occlusion as critical factors preventing state-of-the-art object detectors from performing well. EAGLE will support future works aiming at solving these real-world issues.

G. Cross-dataset validation

We do a cross-dataset generalization to evaluate the generalization capability of EAGLE dataset. We select DOTA for comparison and its validation set for testing. We choose Cascade Mask-RCNN for validation experiments with HBB ground truth. Table V shows that a model trained on EAGLE generalizes well to DOTA, scoring only 6% mAP below a model trained on DOTA, indicating that EAGLE contains

features of DOTA to a large extend. Moreover, as the annotation quality in EAGLE is significantly higher than in DOTA specially with respect to very small vehicles (as mentioned in Section II-A), a portion of false positives in this comparison is due to the detection of vehicles which are generally not annotated and ignored in DOTA, due to their small size. As for DOTA, the model trained on it only achieves 28.23% mAP on EAGLE (-11% mAP of the model trained on EAGLE) reflecting that EAGLE is significantly more diverse and challenging than the current available datasets which makes it appropriate for real-world vehicle detection scenarios.

IV. CONCLUSION

We present EAGLE, a large-scale dataset for task of vehicle detection in aerial imagery, which is multiple times larger than existing datasets. Unlike common object detection datasets, we provide a high number of annotated instances with oriented bounding boxes. We build a dataset specifically focusing on real-world scenarios which includes a variety of situations in aerial photography such as time, weather, and places. The detection of vehicles in any situation regardless of their size and appearance with arbitrary orientations contains useful information for different applications, making it useful for many practical applications. Our benchmarks show EAGLE is a very challenging dataset for the current state-of-the-art object detection algorithms. We also showcase a general method on object detection which can be modified to detect oriented objects. We believe EAGLE addresses the task of vehicle detection in remote vision bringing it to the next practical level. It also introduces interesting challenges to object detection domain in computer vision.

V. ACKNOWLEDGEMENT

We thank Ternow AI GmbH for the data labeling support.

REFERENCES

- [1] A. Ajay, V. Sowmya, and K. P. Soman, “Vehicle detection in aerial imagery using eigen features,” in *ICCS*, 2017.
- [2] M. Lewandowski, B. Paczek, M. Bernas, and P. Szymaa, “Road traffic monitoring system based on mobile devices and bluetooth low energy beacons,” *Wireless Communications and Mobile Computing*, 2018.
- [3] A. Souza, C. Brennand, R. Yokoyama, E. Donato, E. Madeira, and L. Villas, “Traffic management systems: A classification, review, challenges, and future perspectives,” *International Journal of Distributed Sensor Networks*, 2017.
- [4] A. Makiuchi and H. Saji, “Vehicle detection using aerial images in disaster situations,” in *Recent Advances in Technology Research and Education*, G. Laukaitis, Ed. Springer International Publishing, 2019.
- [5] P. Pinheiro, T. Lin, R. Collobert, and P. Dollár, “Learning to refine object segments,” *ECCV*, 2016.
- [6] S. Honari, J. Yosinski, P. Vincent, and C. Pal, “Recombinator networks: Learning coarse-to-fine feature aggregation,” *CVPR*, 2016.
- [7] G. Ghiasi and C. C. Fowlkes, “Laplacian pyramid reconstruction and refinement for semantic segmentation,” *ECCV*, 2016.
- [8] A. Newell, K. Yang, and J. Deng, “Stacked hourglass networks for human pose estimation,” *ECCV*, 2016.
- [9] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” *CVPR*, 2017.
- [10] X. Yang, H. Sun, K. Fu, J. Yang, X. Sun, M. Yan, and Z. Guo, “Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks,” *Remote Sensing*, 2018.
- [11] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” *CVPR*, 2009.
- [12] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and L. Zitnick, “Microsoft COCO: Common objects in context,” *ECCV*, 2014.
- [13] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *IJCV*, 2010.
- [14] G. Heitz and D. Koller, “Learning spatial context: Using stuff to find things,” in *ECCV*, D. Forsyth, P. Torr, and A. Zisserman, Eds. Springer Berlin Heidelberg, 2008.
- [15] S. Razakarivony and F. Jurie, “Vehicle detection in aerial imagery: A small target detection benchmark,” *JVCIR*, 2016.
- [16] K. Liu and G. Mattyus, “Fast multiclass vehicle detection on aerial images,” *GRSL*, 2015.
- [17] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and X. Ji, “Orientation robust object detection in aerial images using deep convolutional neural network,” *ICIP*, 2015.
- [18] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. J. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, “DOTA: A large-scale dataset for object detection in aerial images,” *CVPR*, 2017.
- [19] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, “A large contextual dataset for classification, detection and counting of cars with deep learning,” in *ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., 2016.
- [20] G. Cheng, P. Zhou, and J. Han, “Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images,” *TGRS*, 2016.
- [21] Z. Liu, H. Wang, L. Weng, and Y. Yang, “Ship rotated bounding box space for ship extraction from high-resolution optical satellite images with complex backgrounds,” *GRSL*, 2016.
- [22] Y. Long, Y. Gong, Z. Xiao, and Q. Liu, “Accurate object localization in remote sensing images based on convolutional neural networks,” *TGRS*, 2017.
- [23] B. Shi, X. Bai, and S. Belongie, “Detecting oriented text in natural images by linking segments,” in *CVPR*, 2017.
- [24] S. M. Azimi, E. Vig, R. Bahmanyar, M. Körner, and P. Reinartz, “Towards multi-class object detection in unconstrained remote sensing imagery,” in *ACCV*, 2018.
- [25] Z. Cai and N. Vasconcelos, “Cascade R-CNN: high quality object detection and instance segmentation,” *TPAMI*, 2019.
- [26] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *ICCV*, 2017.
- [27] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *NIPS*, 2015.
- [28] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” *arXiv preprint arXiv:1904.01355*, 2019.
- [29] Y. Li, Y. Chen, N. Wang, and Z. Zhang, “Scale-aware trident networks for object detection,” *arXiv preprint arXiv:1901.01892*, 2019.
- [30] B. Singh, M. Najibi, and L.-S. Davis, “SNIPER: Efficient multi-scale training,” in *NeurIPS*, 2018.
- [31] J. Dai, Y. Li, K. He, and J. Sun, “R-FCN: Object detection via region-based fully convolutional networks,” in *NeurIPS*, 2016.
- [32] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.
- [33] S. Zhang, L. Wen, X. Bian, Z. Lei, and S.-Z. Li, “Single-shot refinement neural network for object detection,” in *CVPR*, 2018.
- [34] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg, “Ssd: Single shot multibox detector,” in *ECCV*, 2016.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CVPR*, 2016.
- [36] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *CVPR*, 2017.
- [37] B. Normalization, “Accelerating deep network training by reducing internal covariate shift,” *CoRR.–2015.–Vol. abs/1502.03167.–URL: http://arxiv.org/abs/1502.03167*, 2015.
- [38] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks For Large-Scale Image Recognition,” *ICRL*, 2015.
- [39] M. Liao, B. Shi, and X. Bai, “Textboxes++: A single-shot oriented scene text detector,” *Transactions on Image Processing*, 2018.
- [40] Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving into high quality object detection,” *CVPR*, 2018.
- [41] G. Ross, “Fast R-CNN,” *CVPR*, 2015.
- [42] J. Shermeyer and A. Van Etten, “The effects of super-resolution on object detection performance in satellite imagery,” in *CVPRW*, 2019.

- [43] S. M. Azimi, C. Henry, L. Sommer, A. Schumann, and E. Vig, “Skyscapes - fine-grained semantic understanding of aerial scenes,” in *ICCV*, 2019.