Informed Region Selection for Efficient UAV-based Object Detectors: Altitude-aware Vehicle Detection with CyCAR Dataset

Alexandros Kouris[1], Christos Kyrkou[2] and Christos-Savvas Bouganis[1]

***Abstract*— Deep Learning-based object detectors enhance the capabilities of remote sensing platforms, such as Unmanned Aerial Vehicles (UAVs), in a wide spectrum of machine vision applications. However, the integration of deep learning introduces heavy computational requirements, preventing the deployment of such algorithms in scenarios that impose low-latency constraints during inference, in order to make mission-critical decisions in real-time. In this paper, we address the challenge of efficient deployment of region-based object detectors in aerial imagery, by introducing an informed methodology for extracting candidate detection regions (proposals). Our approach considers information from the UAV on-board sensors, such as flying altitude and light-weight computer vision filters, along with prior domain knowledge to intelligently decrease the number of region proposals by eliminating false-positives at an early stage of the computation, reducing significantly the computational workload while sustaining the detection accuracy. We apply and evaluate the proposed approach on the task of vehicle detection. Our experiments demonstrate that state-of-the-art detection models can achieve up to 2.6x faster inference by employing our altitude-aware data-driven methodology. Alongside, we introduce and provide to the community a novel vehicle-annotated and altitude-stamped dataset of real UAV imagery, captured at numerous flying heights under a wide span of traffic scenarios.**

I. Introduction

Deep Learning has become a prominent technology in many machine vision tasks, with Convolutional Neural Networks achieving state-of-the-art accuracy in image classification, object detection and semantic segmentation. This advancement acts as an enabler for a wide range of emerging applications related to autonomous systems, such as self-driving cars [1] and Unmanned Aerial Vehicles (UAVs) [2].

In particular, Micro Aerial Vehicles (MAVs) are gaining increasing attention, as a result of the extensive availability and ease-of-use of commercially available low-cost quadcopters. Being usually equipped with both a forward- and a downward-looking camera, drones can act as a mobile remote sensing platform for real-world applications including infrastructure inspection, emergency response, search and rescue operations, surveillance and traffic monitoring, by exploiting computer vision algorithms to analyse the incoming visual data captured from their on-board visual sensors.

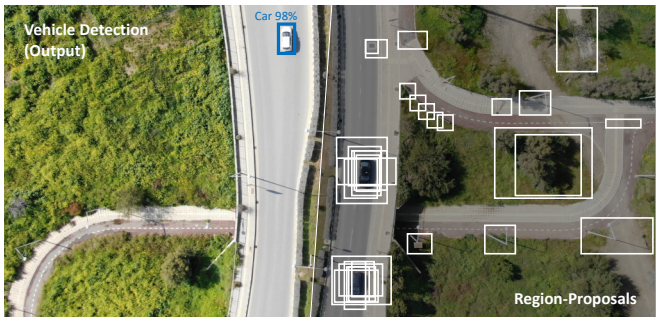

Fig. 1. (left): Output of a remote-sensing (UAV) based vehicle detector. (right): Class-agnostic region proposals extracted by region-based detectors. **Video & Dataset:** www.imperial.ac.uk/intelligent-digital-systems/cycar/

Focusing on the case of traffic monitoring, UAVs contribute towards "smart cities" by facilitating infrastructure-free situational awareness, providing real-time traffic estimation through vision-based vehicle detection. This can be used for traffic regulation purposes or, for example, to aid the route planning of emergency response vehicles etc. Moreover, the wide field-of-view along with the enhanced mobility of UAVs allow their rapid and facile deployment for remote sensing in areas-of-interest (Fig. 1), eliminating the need for expensive infrastructure in urban or rural environments, such as fixed surveillance cameras or embedded car sensors [3].

CNN-based object detectors have experienced tremendous progress through the last years, demonstrating impressive accuracy. This advancement however, came at the cost of increased computational workload, challenging their real-world deployment. In some perception tasks, where one or more UAVs are regularly deployed to collect data from an area-of-interest, a cloud-based setup for performing the analysis of data on remote base servers is applicable [4].

At the other end of the spectrum, UAV-based deep learning models can be in-the-loop of mission critical decisions including navigation and collision avoidance [5] [6]. In such cases, low-latency requirements are imposed for inference, making the cost of a wireless link between the drone and the base station prohibitive. Moreover, when deployed in remote areas such communication may not be possible to be established [7]. Besides, communication poses an added security risk for intercepting and exposing sensitive data. Near-sensor (edge) processing provides an alternative approach for such scenarios. However, when on-board processing is

The support of the EPSRC Centre for Doctoral Training in High Performance Embedded and Distributed Systems (HiPEDS, Grant Reference EP/L016796/1) is gratefully acknowledged. This work is supported by the European Union Civil Protection under grant agreement No 783299 (SWIFTERS) and by the European Union's Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE) and from the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development. The authors would like to acknowledge Petros Petrides for his contribution on the dataset collection process. Christos Kyrkou would like to acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

[1]Alexandros Kouris and Christos-Savvas Bouganis are with the Dept. of Electrical and Electronic Engineering, Imperial College London, UK
Email: {a.kouris16,christos-savvas.bouganis}@imperial.ac.uk
[2]Christos Kyrkou is with the KIOS Research and Innovation Center of Excellence, University of Cyprus, Nicosia, Cyprus
Email: kyrkou.christos@ucy.ac.cy

required, the deployment challenges are aggravated by the limited computational resources and low-power restraints necessitated by the low-payload capabilities of UAVs.

Returning to the task of vehicle detection, both cloud-centric and near-sensor processing models are considered applicable [8]. Although mission-critical decisions for the UAV itself are usually not based on the traffic information obtained by analysing the captured video feed, in many applications this information contributes to the decision-making process of traffic regulation or emergency response. Hence, softer near-real-time (NRT) requirements are frequently imposed.

Single-shot detectors (SSD) are typically meeting this requirements being positioned at the high-performance end of the performance-accuracy pareto frontier, in contrast to the computationally expensive region-proposal-based detectors lying at the other end [9]. SSD-based approaches, however, suffer from significant accuracy degradation in objects with small spatial resolution, due to the small input receptive field that acts as a prerequisite for maintaining low-latency [10]. In UAV-based visual sensing, where high flying altitudes[1] are preferred as they provide larger ground coverage on aerial imagery, this issue becomes extremely crucial since combined with the wide filed-of-view of UAV cameras it results to objects with very limited spatial resolution.

In this paper we adopt a region-proposal-based approach and introduce a data-driven region selection methodology, that incorporates application-specific domain knowledge and considers additional information provided by the UAV's on-board sensors (altitude, GPS, video etc.), to optimise its performance by discarding a large portion of unnecessary computation at runtime. The proposed approach is agnostic to the computation platform and enables the efficient deployment of state-of-the-art region-based detectors on real-time UAV applications under either a cloud-based or an embedded processing scenario. The main contributions of this work can be summarised by the following:

- We introduce a novel region selection methodology for UAV-based visual detection that evaluates candidate detection regions at runtime, leveraging additional application-specific information to provide significant latency speed-up by eliminating false positives and their corresponding computational workload at an early stage of the computation, preserving the detection accuracy.
- We created and released "CyCAR", a dataset of high-res altitude-stamped UAV images annotated for vehicle detection in urban environments, capturing a diverse set of traffic scenarios from multiple flying heights.

II. Background and Related Work

A. Convolutional Object Detectors

CNN-based object detectors, aiming to localise instances from a pre-specified set of classes in an image, have recently demonstrated significant advancement achieving state-of-the-art accuracy. We will focus on two main categories of convolutional object detectors:

[1] In the context of this work, the terms flying height, altitude and level-above-ground are used interchangeably, referring to the height above ground level (AGL).

Region-based Object Detectors Region-based detectors consist of two separate stages. During the first stage, a large number of class-agnostic candidate regions are extracted and forwarded to the second stage that evaluates each region proposal independently to predict class-specific detection probabilities along with refined bounding boxes (Fig. 1). Faster R-CNN [11] constitutes a representative example of region-based object detectors, achieving remarkable accuracy in various well-established datasets [12] [13]. In Faster R-CNN, the input image is initially pushed through the Feature Extractor part (FE) of a CNN, consisting of Convolutional and Pooling layers. Subsequently, a Region-Proposal Network (RPN) acts on the output feature maps of a selected intermediate layer of the FE to identify a set of category-independent candidate detection boxes (typically 300). These are selected based on their predicted "objectness" probability, out of a large set of refined box priors (called anchors) spread across a wide range of spatial locations, scales and aspect ratios, organised in a regular grid. In the succeeding second stage, each of the selected region proposals is independently pushed through a box predictor consecutively, after being cropped from the activations of the same intermediate layer.

Single-Shot Object Detectors consist of a feed-forward CNN that directly predicts the class probabilities and locations of multiple objects though a single pass of the input image. Hence, their architecture can be considered similar to the first stage of a region-based detector. However, in contrast to the latter that requires consecutive passes of numerous regions-of-interest through a second-stage classifier, in SSD the output of the FE is directly fed to a series of Fully-Connected layers. These are branched deeper in the network to predict class probabilities and box coordinates for different regions of the input image, all with a single pass, leading to low computational complexity. YOLO [14] and Ssd [15] form representative examples of this detector class.

B. Efficient Learning-based Detectors on UAVs

Recent literature has explored the performance-accuracy trade-off on single-shot detectors for efficient real-time vehicle detection on embedded platforms. In [16] the number and size of filters and the input image resolution are exposed to a neural architecture search methodology, whereas [17] employs a novel automated filter-pruning approach along with careful design choices, to find efficient network designs.

Application-specific customisation has also been proposed for single-shot detectors, utilising prior domain knowledge to adapt the model in a controlled manner that improves performance, with minimum effect on accuracy. For example, in [18] a single-shot detector is employed for gate detection in drone racing. Considering the simple geometry of square gates used in drone racing competitions, plenty of the unnecessary high-level feature layers are removed. Along the same lines, in [19] a Markov Decision Process is used to adjust all the tunable parameters of an object detection algorithm in order to exploit the accuracy-execution time trade-off for sign recognition on resource-constraint mobile robot platforms.

Although SSD models have almost met the accuracy of

region-based approaches for large objects, while providing lower runtimes per image, they demonstrate significantly poorer detection accuracy on small objects [9]. SSD models can only target a fixed-resolution input image, conversely to region-based, while their fast inference speed is conditioned on handling low-resolution inputs [10] (typically between 200x200 and 600x600). Hence, high-res input images have to be down-scaled to match the network's receptive field suffering, as a side-effect, a reduction in detection accuracy. Recent research has proposed pushing cropped overlapping tiles of the input image through the CNN independently, without degrading the initial resolution [20] [21] [22]. However, this approach introduces notable computational overhead that can only be partly alleviated by the use of attention and memory mechanisms for selective tile processing [23].

This shortcoming of single-shot detectors on small-scaled objects is particularly relevant in aerial imagery where different instances of similar objects appear in a vast variety of different scales, accounting to the fact that objects appear in a spatial resolution that is inversely proportional to the flying altitude of the UAV. To address relevant scale ambiguities, in the work of [24] two built-in sub-networks are incorporated to account for the variability of scales in car detection, acting as ensemble of models trained on disjoint scale ranges, leading however to considerable computation overhead.

Nevertheless, in vehicle detection from aerial imagery, high accuracy detectors supporting small spatial resolution of objects are essential to permit high-altitude flights, resulting to larger ground-area coverage [8]. Recent research has shown that region-based detectors demonstrate remarkable accuracy under such challenging conditions [25], at the expense of increased computational payload. In the work of [9], the speed-accuracy trade-off in convolutional detectors is broadly discussed. Specifically for Faster R-CNN based detectors, it is shown that the number of RPN's candidate detections pushed through the second-stage detector is affecting the overall latency remarkably. Thus, reducing the number of region proposals results to a noteworthy computation saving, to the proportional detriment of detection accuracy.

In this work, we exploit UAV sensor measurements to perform an informed application-specific online pruning of RPN's region proposals, reducing effectively the workload of the second-stage box predictor through early identification of false positives, while avoiding a compromise in accuracy. This data-driven approach pushes the optimality frontier in the accuracy-latency trade-off, by combining prior domain knowledge with sensor information injected to the system at runtime. Specific to the vehicle detection task of this work, the utilised information comprises of: (i) *the flying altitude* used to reason about the expected spatial *size*, *shape* and *density* of vehicles on the input image, given the camera configuration of the UAV and (ii) *computer-vision filters* applied on the input frames to segment regions-of-interest providing priors for the expected spatial location of vehicles. Both sources contribute on identifying a variable-sized subset of "meaningful" region proposals forwarded to the classifier.

III. The CyCAR Dataset

To facilitate our experiments we have created "CyCAR", a dataset of *altitude-stamped* high-resolution UAV imagery targeted on the task of ground vehicle detection.

Context: The data collection process involved real UAV flights above the city of Nicosia in Cyprus, in varying altitudes, with the on-board camera's pose being vertical to the ground. The dataset is *altitude-stamped* based on the (discretised) flying height estimated by the UAV's air-pressure sensor readings. A subset of the collected frames is annotated by human experts with tight bounding boxes enclosing all captured cars in the camera's Field-of-View (FoV). The annotated frames span from heavily congested to clear traffic situations. CyCAR comprises a wide variety of scenarios including: (i) *a static UAV above an area-of-interest*, (ii) *a moving UAV on a pre-specified path* and (iii) *a moving UAV following a target vehicle on the ground*. These contexts facilitate a comprehensive set of traffic monitoring scenarios, such as persistent monitoring of an area for traffic regulation purposes, periodical data collection for extraction of traffic statistics, and live traffic density estimation in the surrounding area of a moving target (e.g. for assisting emergency vehicle navigation), respectively.

UAV Altitude: A key feature of the introduced dataset is the broad range of UAV flight altitudes that have been poured into the data collection process. With a minimum level of 20m above ground, highly-detailed close-distance images of vehicles and their surrounding environment are captured; whereas the maximum flight-level of 500m above ground provides enormous and challenging coverage of ground area embracing a large number of vehicles and cluttered background in a single frame. A large portion of the captured frames is focused on the range of [70m,130m] above ground, that forms typical flying levels of commercially available UAVs in urban environments [26].

Annotation: Although we are actively extending our dataset, at the moment of writing CyCAR already comprises more than 27 minutes of high-res (4K, 2.7K and FullHD) real-flight video (translating to approximately 48K frames), with a subset of 450 images containing more than 5.000 vehicle instances being annotated with tight Horizontal Bounding Boxes (HBB), following the annotation format of the Pascal VOC Dataset [12]. Altitude stamps indicating the flying height of the UAV with respect to its take-off point at the moment each frame was captured (discretised into 10m segments) are also included in the annotation.

Existing datasets specific to top-view vehicle detection mostly comprise of satellite [27] [28], or static surveillance camera [29] imagery. UAV-based datasets are usually restricted to specific scenes such as carparks [30] or incorporate a limited range of flying altitudes, e.g. [31] features 5-25m flights. UAVDT [32] and VisDrone2018 [33] comprise the most diverse large-scale UAV benchmarks for detection and tracking to date. Based on our literature review, UAVDT is the only dataset labelling its frames according to the UAV height, using however only 3 coarse classes (*low*, *mid*, *high*).

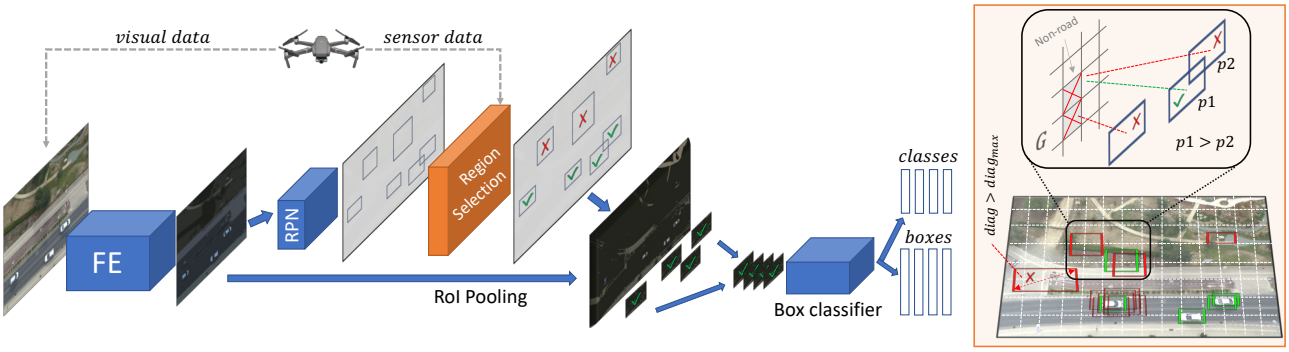

Fig. 2. (left): The introduced informed region selection methodology, within the Faster R-CNN pipeline. Region proposals are filtered based on sensor data and prior knowledge at runtime, resulting to early elimination of false-positive candidate detections and corresponding computation savings. (right): Altitude-aware pruning of candidate detection regions based on their spatial scale, density and location based on the proposed altitude-specific density grid.

Hence, to the best of our knowledge, CyCAR is the first UAV-based dataset for vehicle detection featuring fine-grained flying altitude annotation (10m). This, in our opinion, appoints CyCAR to a valuable complementary data source for developing and benchmarking data-driven methodologies (e.g. for traffic monitoring applications), such as the altitude-aware vehicle detector introduced in Sec. IV.

IV. Informed UAV-based Region Proposals

Region-based object detectors select a fixed-cardinality set of region proposals out of a much larger group of class-agnostic candidate regions, based on their RPN-predicted probability of *objectness*. Each selected region is then processed independently by a box classifier, that predicts class probabilities and a refined bounding box for each proposal.

To eliminate duplicate detections of the same object that would hurt the overall detection accuracy of the model, greedy Non-Maximum Suppression (NMS) is applied to the final output of the detector, in a per-class manner. This way, all detected regions that overlap considerably with a higher-scoring detection window of the same class are rejected, based on a tunable Intersection-Over-Union (IOU) threshold.

In recent literature, this post-processing step has been enhanced to utilise additional application-specific information, in order to eliminate false-positive detections that downgrade the model's accuracy, such as bounding boxes with unexpected size and/or aspect ratio [25] [16]. However, since these methods are only applied on the final output of the predictor, they provide no improvement on its execution time.

Inspired by these works, the key idea of this paper is to inject application-specific information in much earlier stages of the detection pipeline, pushing the rejection of false positives to a point that it results to significant computation savings and thus improve the detection speed. We introduce a data-driven region selection methodology for UAV-based object detectors, that filters the original RPN's region proposals in an informed way, utilising additional data from the UAV's sensors along with prior domain knowledge. Only candidate regions that meet the pre-specified criteria are propagated to the computationally-expensive second-stage predictor, improving the efficiency of the overall model and achieving higher performance, without mitigating accuracy.

We choose to embody the proposed methodology in the Faster R-CNN pipeline (Fig. 2), due to its state-of-the-art detection accuracy. However, our approach is applicable on any region-based detector with variable gains, depending on its underlying workload breakdown between the two detection stages. Furthermore, for the rest of this paper, the discussion will be focused on the task of UAV-based vehicle detection [34]. Again, it is noteworthy that with appropriate adaptation of the region selection criteria, the proposed approach can be applied on a wide range of UAV-based detection tasks. Specific to vehicle detection, three main region selection criteria are examined:

A. Candidate Region Scale (Size and Shape)

Since UAVs are equipped with a camera that has fixed parameters (maximum resolution res=$\{res_x, res_y\}$ and Field-of-View fov) and can fly on an outspread of altitudes, cars on the ground appear on a wide spectrum of different scales. Utilising the vehicle annotations from the altitude-stamped CyCAR dataset $\mathbb{D}$:$(\mathbf{X}^{(k)}, h^{(k)}, \mathbb{V}^{(k)})$, where $\mathbf{X}^{(k)}$∈$[0,255]^{res_x \times res_y \times 3}$ denotes the k-th image, while its associated altitude-stamp is denoted by $h^{(k)}$∈$\{20, 30, ..., 500\}m$ and its set of vehicle annotations by $\mathbb{V}^{(k)}$:$(x^{k_{(i)}}_{min}, y^{k_{(i)}}_{min}, x^{k_{(i)}}_{max}, y^{k_{(i)}}_{max})$ with i∈$\{1, 2, ..., N^{(k)}\}$ where $N^{(k)}$ denotes the number of vehicles in image k, a polynomial model $f_{\mathbb{D}}$ is built in order to extract and abridge the encapsulated domain knowledge, by fitting the anticipated range of scales (size and shape) of vehicle bounding boxes as a function of the UAV flying altitude h, given a set of camera characteristics, in the form of:

$$[scale^{(h)}_{min}, scale^{(h)}_{max}] = f_{\mathbb{D}}(h, res, fov) \quad (1)$$

The bounding box *area* in pixels^2, *diagonal*, *height* and *width* or *smallSide* (=$min(height, width)$) and *largeSide* in pixels, the *aspect-ratio* or any combination of the above can be effectively employed as metrics of $scale$, depending on the application-specific prior knowledge that is incorporated.

By injecting that application-specific knowledge along with the UAV's flying altitude (estimated by its on-board sensor readings) to the RPN's post-processing, candidate regions with scale outside of the expected range are rejected at runtime, reducing the number of boxes that are pushed through the second-stage classifier only to those with "meaningful" size/shape. This altitude-aware optimisation results to

Fig. 3. (a,b): Region proposals of the original method, for different IOU threshold values. (c): Output of the proposed informed region selection methodology.

a significant reduction of the evaluated box proposals, as the attention of the original RPN is considerably attracted by feature-rich areas on the input image (Fig. 3a,b), independently of their scale and existence of vehicles.

Horizontal Bounding Boxes (HBB) of vehicles whose orientation is not well-aligned with one of the image axes, dissipate more area and demonstrate reduced aspect-ratio compared to axis-aligned boxes captured from the same height (Fig. 4). This results to extended variance in the discussed metrics of vehicle scale, reducing the effectiveness of the proposed region selection approach. Dealing with this ambiguity is important, due to the exclusive support of HBB by most deep learning frameworks. For this purpose, the typical range of axis-aligned vehicles' aspect-ratio is extracted from a subset of $\mathbb{D}$ using the model of Eq.1. Subsequently, an exponential model is developed to estimate the "essential" scale (Fig. 4) of each of the RPN's region proposals. This model is predominantly affecting region proposals corresponding to axis-unaligned vehicles, identified by their deviation from the expected aspect-ratio, expressed as:

$$\widetilde{sc}^{(i)} = sc^{(i)} - (\hat{r} - r)^a \cdot b \cdot sc^{(i)} \ , \quad \forall i \in [1, N] \qquad (2)$$

where $\widetilde{sc}^{(i)}$ denotes the inferred essential *scale* for the i^{th} (out of N) RPN's region proposal with initial scale $sc^{(i)} \in \{area, diag, ...\}$ and aspect ratio r, $\hat{r}$ represents the mean aspect ratio of axis-aligned vehicle detections in the dataset, and a, b correspond to tunable parameters of the exponential model. $\widetilde{sc}^{(i)}$ can replace $sc^{(i)}$ in the region proposal pruning, as it provides a *scale*-estimate of the equivalent axis-aligned bounding box to the original HBB.

B. Candidate Region Density

Apart from the spatial resolution of objects, their maximum density in the original image is also affected by the UAV flying altitude. Experiments demonstrated that although NMS is applied on the RPN's candidate boxes, there still exists a notable number of partly overlapping region proposals with similar size and small displacement, translating to unnecessary computational workload for the second-stage predictor (Fig. 3a). Solving this problem by selecting a more greedy IOU threshold to the NMS to force the RPN to spread its region proposals, also results to increased false negative rate, especially in the case of HBB where the candidate regions of vehicles frequently overlap in congested cases.

In this work we address this issue in an informed way, considering the UAV's altitude h to estimate the maximum possible vehicle density. Thereafter, the spatial density of the RPN's proposals forwarded to the second-stage classifier

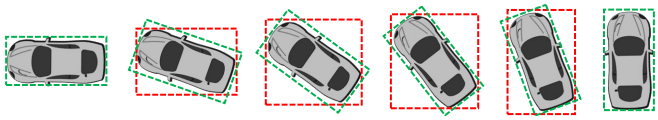

Fig. 4. Red: Horizontal Bound Box, Green: "Essential" Bounding Box

is restricted accordingly. In more detail, we firstly extract the expected length of the smaller bounding-box side for a given altitude $minL$=$smallSide^{(h)}_{min}$. A grid $G^{(k)}$ of $[minL/2, minL/2]$-sized cells across each input image is constructed and every candidate region is assigned to a cell of that grid, based on the coordinates of its centre (Fig. 2). The proposed method allows only a single proposal from each cell to be forwarded for classification, considering the RPN's predicted objectness probability for each candidate region to resolve conflicts. This is equivalent to parsing the input image with a sliding window of size $minL \times minL$ and stride $minL/2$ (Nyquist Sampling Rate), to extract a single proposal per step, significantly sparsifying the number of candidate detections, while maintaining a robust set of meaningful proposals to be fed to the second stage (Fig. 3c).

C. Candidate Region Spatial Location

In previous work [35], output bounding boxes of the detector are aligned with a visually extracted road mask, to eliminate false positives by discarding detections outside of the input image's road segments. We adopt the key idea of this methodology, pushing it however towards an earlier stage of the detection pipeline, to exploit such application-specific knowledge for alleviating further unnecessary computation.

Light-weight computer vision filters or GPS-aligned (heat-)maps can be employed to identify regions-of-interest on the input image. These regions are then projected to the image cells of the altitude-specific grid $G^{(k)}$ based on a minimum coverage threshold, adding a dimension to the region selection methodology by only considering proposals assigned on the subset of cells that have been identified as regions-of-interest. This approach discards a greater expanse of object proposals by incorporating prior knowledge regarding their spatial location's context on the input image.

In this work, a simple computer-vision method is employed to identify road segments in each input image, based on per-pixel HSV thresholding. A smoothing moving-window averaging filter is also applied on the output pixel-mask in order to eliminate false-negative road regions, usually caused by vehicles on the road especially in low-altitude flights. More advanced road-segmentation techniques from the literature can also be applied at this stage [36] instead of

the proposed HSV-based filtering. However, our experiments found this approach sufficient for this work, while being immensely computation efficient due to its simplicity.

By combining the aforementioned criteria, an enhanced region selection methodology is established, based on which a candidate region is forwarded to the box classifier if it:

- features the expected *scale*, given the flying altitude,
- is the highest-scoring region proposal on its *density* cell,
- belongs to a cell that corresponds to a *region-of-interest*.

V. Evaluation

A. Experimental Setup

In this section, the experimental evaluation of the proposed methodology is discussed. Experiments have been conducted in two different settings: *(i) Cloud-based high-end processing* in which the MAV transmits the captured video stream along with other sensor data to a remote on-the-ground server, located in a base station, for processing on a high-end GPU and *(ii) On-board embedded processing* where all the computations are performed on-board the resource-constrained UAV platform, on an embedded GPU device. The server is equipped with an Intel Xeon E5-2630 CPU, 64GB RAM and a 2560 CUDA-Core Nvidia GTX1080 GPU (Pascal Architecture). For the embedded setting we use an Nvidia Jetson TX2 board featuring a 256 CUDA-Core embedded GPU (Pascal Architecture), 8GB of RAM and an Quad ARM A57 CPU. The models have been developed and trained using the Object Detection API [9] of Tensorflow (v1.12) on the same server; the deployment on the TX2 platform resorted to Tensorflow v1.9. The UAV used in the experiments is a DJI MAVIC 2 Enterprise, equipped with a high-definition camera with Field-of-View of 82.6^o.

B. Single-Shot vs Region-Based Detectors on UAV Imagery

A comparison between single-shot (SSD [15]) and region-based (Faster R-CNN [11]) detectors on aerial imagery is initially performed, employing the concept of meta-architectures introduced in [9]. Meta-architectures provide a level of abstraction by capturing the algorithmic body of each detector family, while decoupling it from its original implementation (i.e. allowing any CNN model to be used for feature extraction and classification). A series of COCO-pretrained models that have been identified as key points on the performance-accuracy optimality frontier are selected, featuring different choices of backbone CNN architectures. We train all selected models on the DOTA multi-class dataset of aerial imagery [22] until convergence. Subsequently, fine-tuning of the models on the CyCAR Dataset is performed. Various data augmentation techniques have been used on both datasets, including: random horizontal flipping, image re-scaling and adjustment of contrast and brightness. The PascalVOC-established metric of mean-Average Precision (mAP) [12] is reported across the Validation Set of each dataset, summarising the shape of the precision-recall curve, defined as the mean precision at a set of eleven equally-spaced recall levels in the range of [0,1].

Table I summarises the results of this comparison. As expected, the single-shot architecture based on MobileNet V2, achieves the lowest latency both on the high-end and the embedded device. This comes at the expense of a significant accuracy compromise of approximately 24 and 30 percentage points (p.p.) in the validation sets of DOTA and CyCAR dataset respectively, compared to the highly-accurate Faster R-CNN-based detector (built on Inception-ResNet-v2). The small input receptive field of the SSD detector is mainly responsible for its poor classification accuracy, especially in aerial imagery were a large portion of objects frequently appear in (very) limited spatial resolution.

TABLE I
Comparison of Pareto Optimal (Speed/Accuracy) Detectors

Detector Model		Performance (Latency)		mAP	
Meta-Arch	**Feature Extractor**	*GTX1080*	*TX2*	*DOTA*	*CyCAR*
SSD	MobileNet V2	36.31ms	150.81ms	36.23%	46.63%
Faster R-CNN	ResNet50	166.51ms	1070.43ms	50.56%	64.82%
Faster R-CNN	Inception-ResNet-v2	573.05ms	-	60.22%	76.31%

The high-accuracy region-based detector however, requires over 15x more computation time on the high-end GPU (achieving up to 1.75fps) compared to the SDD model, while the embedded platform's memory resources could not accommodate its execution. The Faster R-CNN detector employing ResNet50 as its feature extractor, provides a controlled trade-off between performance and accuracy, suffering an accuracy drop of 10.5 p.p. on average, while being 3.4x faster than the accuracy-optimised region-based detector, being able to perform detection at up to 6fps on the server and slightly under 1fps on the embedded GPU device.

In order to improve the SSD accuracy, multiple separate runs covering different windows of the input image have been proposed [21]. Indicatively, in our experiments we managed to match the accuracy of Faster R-CNN with ResNet50 using a 4x4 grid of overlapping windows, which however results to a prohibitive increase in execution time. Moreover, the accuracy of the Inception-ResNet-v2-based Faster R-CNN could not be matched by any grid configuration, conceivably due to the use of a weaker FE model on SSD.

C. Efficient Region Selection Methodology

In this section, the effect of the proposed region selection methodology in the accuracy and speed of the detection models is evaluated. We select the high-accuracy Faster R-CNN instance (with Inception-ResNet-v2) for the cloud-processing scenario, as well as the ResNet50-based detector for our on-board processing experiments.

1) Baseline: Figures 5 and 6 demonstrate the speed-accuracy trade-off that arises by imposing a fixed (off-line tuned) constraint on the number of region proposals propagating through the second-stage classifier. Although pronounced performance gains can be achieved by reducing the region proposal number, it is evident that when no additional information is used during the pruning of candidate detections a significant undesired accuracy drop is provoked.

2) Quantitative Results: Instead of employing a constant reduced number of proposals, the introduced methodology utilises UAV sensor data and prior domain knowledge to select a variable-cardinality set of proposals at runtime, tailored for each input image, after evaluating the RPN's candidate

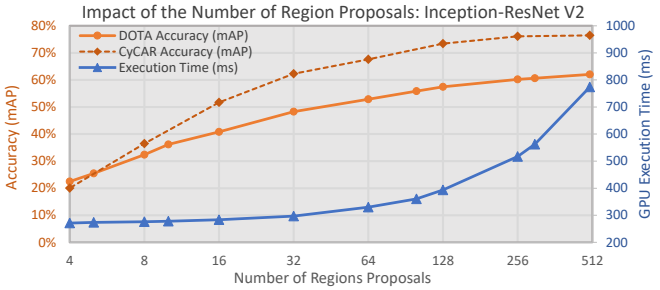

Fig. 5. Speed-accuracy trade-off in Faster R-CNN (Inception-ResNet-v2), as a function of region proposals being evaluated on the second-stage.

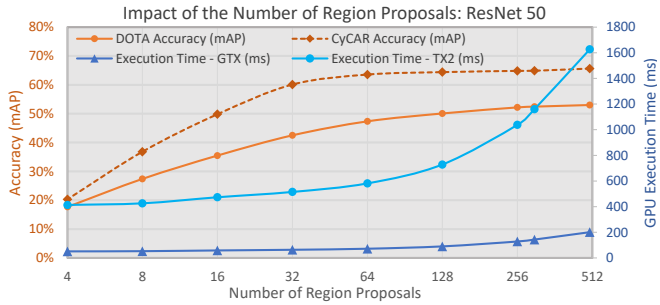

Fig. 6. Speed-accuracy trade-off in Faster R-CNN (ResNet50), as a function of region proposals being evaluated on the second-stage.

regions with respect to altitude-aware *scale*, *density* and *position* criteria (Sec. IV). This informed way to boost the performance of region-based detectors is also preserving the detection accuracy at the level of the original model.

Table II demonstrates the effectiveness of the proposed approach in the task of vehicle detection. Our method achieves a significant reduction in the average number of selected proposals, as a result of exploiting additional information to intelligently prune the region proposals of the RPN. Detection Accuracy along with its constitutive components are reported to concede a better understanding of the impact of each configuration, namely: absolute number of True-Positive (TP), False-Positive (FP) and False-Negative (FN) detections, across a previously unseen CyCAR test-set, comprising of 30 images with 207 cars in total. Our approach is compared with an original implementation of the corresponding detector, as well as two *oracle* baselines that employ a constant reduced number of proposals, hand-tuned to match the achieved performance and accuracy of the proposed methodology on the specific test-set respectively.

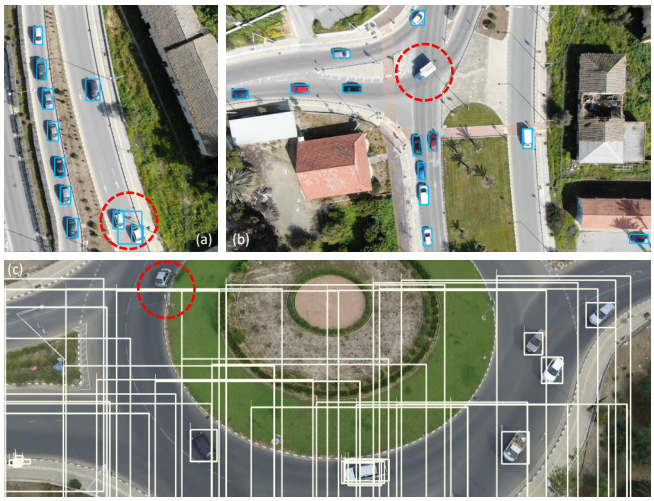

Fig. 7. (a) A false-positive detection of the original model, corrected by the proposed methodology. (b) A false-negative detection of the proposed methodology, caused by extremely strict thresholding of the expected vehicle size. (c) False-negative region proposal after baseline pruning (50 boxes).

TABLE II

COMPARISON OF THE PROPOSED APPROACH WITH BASELINES

Approach	NumProposals	Latency (GTX/TX2)	Accuracy	TP	FP	FN
Inception-ResNet-v2						
Original	300	562.65ms / -	90.05%	190	4	17
This Work	4 - 40	278.41ms / -	89.90%	187	1	20
Matching Acc.	190	462.14ms / -	89.05%	187	3	20
Matching Perf.	10	278.01ms / -	63.64%	133	2	74
ResNet50						
Original	300	143.28ms / 1160.6ms	79.72%	169	5	38
This Work	3 - 54	54.11ms / 439.7ms	78.95%	165	2	42
Matching Acc.	210	110.40ms / 904.2ms	78.20%	165	4	42
Matching Perf.	10	54.28ms / 440.0ms	69.19%	146	4	61

In the case of the Inc.-ResNet-v2-based model, the original implementation extracting 300 candidate regions, achieves high accuracy (90.05%) by detecting 190/207 cars and suffering four false-positive detections, at a consistent latency of 562.65ms per image (1.77fps). Our methodology achieves a similar accuracy of 89.90% by detecting 187/207 cars and also reducing the number of false positives, while requiring more than 2x less computation time (3.6fps) on average. Experiments also indicate that the oracle baseline employing a fixed reduced number of proposals (being hand-tuned aware of the expected output), requires 4.75x more proposals to achieve similar accuracy to our approach, leading to a 66% increase in computation time. Accordingly, a baseline model with constant proposals' number constraint in the same latency budget could only reach an accuracy of 63.64% (translating to an increase of 3.7x in missed detections).

Applying the proposed methodology to the ResNet50-based detector provides a speed-up of 2.64x on the high-end GPU achieving a frame-rate of 18.5fps compared to the rate of 7fps of the original model, while the accuracy difference is preserved within 1p.p. Near real-time processing (reaching 2.3fps) is also enabled by our approach on the embedded platform, being 2.67x faster than the original model.

3) Qualitative Discussion: Employing runtime-adjustable number of region proposals for each image, the proposed methodology pushes the speed-accuracy optimality frontier by achieving remarkable speed-up with little to no compromise in accuracy. This is achieved by the effective utilisation of the additional UAV sensor information to select the most meaningful proposals for evaluation and eliminate outlier candidate detections on an early stage of the computation.

In our experiments we noticed a consistent reduction in false-positive detections by the proposed approach. This is accounted to duplicate detections of the original model that were not captured by the NMS post-processing due to large difference in their relative spatial resolution (e.g. Fig. 7a), and were eliminated by the scale and density criteria of the proposed methodology.

At the other end, it has been noticed that aggressive thresholding on the expected scale of vehicles for a particular

flying level-above-ground may increase the number of false-negatives, when outliers are present in the input image. For example, in the case of Fig. 7b, a light truck with larger scale than the rest of the vehicles in the frame, was not detected by the proposed approach. Adopting a less tight range of expected scales in each altitude resolves this issue, with marginal effect on the computational workload.

Finally, it is noteworthy that the baseline approach employing a fixed reduced number of region proposals requires a much larger-cardinality set of candidate regions in order to match the accuracy of the proposed approach. This is accounted to two reasons: (i) Some frames, especially on high-altitude flights or above congested areas, capture a large number of vehicles easily surpassing small pre-specified threshold values in proposals' number. (ii) Even in less congested frames, the original method retains a fixed-cardinality set of proposals that achieve the highest "objectness" probability values. Frequently, when a small number of proposals is selected, multiple candidate detections of the same vehicle (in different scales and aspect ratios) may surpass the "objectness" probability of vehicles in more challenging regions of the input image (such as shadowed areas). This may result to accuracy degradation, even when a surplus is achieved on the proposal threshold compared to the actual number of vehicles in the frame (e.g. Fig. 7c). The proposed approach, effectively handles the above cases, by jointly considering the spatial resolution, density and location of candidate regions in the image, to dynamically adjust the number of proposals for every frame at runtime.

VI. Conclusion

We have presented a novel region selection methodology for region-based object detection in UAV imagery. By exploiting additional sensor information and lightweight computer vision filters along with prior application-specific knowledge, our approach can significantly reduce the inference time of state-of-the-art object detection models without mitigating accuracy; extending their applicability on near real-time applications, such as UAV-based traffic monitoring. A new dataset of real UAV imagery annotated for the task of vehicle detection and altitude-stamped to support the development of data-driven methods is also introduced, featuring a variety of flying heights and traffic scenarios.

References

[1] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-End Learning of Driving Models from Large-Scale Video Datasets," in *CVPR*, 2017.

[2] A. Loquercio, A. I. Maqueda, C. R. del-Blanco, and D. Scaramuzza, "DroNet: Learning to Fly by Driving," *RA-L*, vol. 3, no. 2, 2018.

[3] C. Kyrkou, S. Timotheou, P. Kolios, T. Theocharides, and C. Panayiotou, "Drones: Augmenting Our Quality of Life," *IEEE Potentials*, vol. 38, no. 1, 2019.

[4] J. Lee, J. Wang, D. Crandall, S. abanovi, and G. Fox, "Real-Time, Cloud-Based Object Detection for Unmanned Aerial Vehicles," in *IEEE Int. Conf. on Robotic Computing (IRC)*, 2017.

[5] N. Smolyanskiy, A. Kamenev, J. Smith, and S. Birchfield, "Toward Low-Flying Autonomous MAV Trail Navigation Using Deep Neural Networks for Environmental Awareness," in *IROS*, 2017.

[6] A. Kouris and C. Bouganis, "Learning to Fly by MySelf: A Self-Supervised CNN-Based Approach for Autonomous Navigation," in *IROS*, 2018.

[7] A. Montanari *et al.*, "Surveying Areas in Developing Regions Through Context Aware Drone Mobility," in *ACM Workshop on Micro Aerial Vehicle Networks, Systems and Applications*, 2018.

[8] K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis, "Survey of Unmanned Aerial Vehicles (UAVs) for Traffic Monitoring," *Springer Handbook of Unmanned Aerial Vehicles, Ch. 110*, 2015.

[9] J. Huang *et al.*, "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors," in *CVPR*, 2017.

[10] A. Suleiman *et al.*, "Towards Closing the Energy Gap Between HOG and CNN Features for Embedded Vision," in *ISCAS*, 2017.

[11] S. Ren *et al.*, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *NIPS*, 2015.

[12] M. Everingham *et al.*, "The Pascal Visual Object Classes (VOC) Challenge," *Int. Journal of Computer Vision*, vol. 88, no. 2, 2010.

[13] T.-Y. Lin *et al.*, "Microsoft COCO: Common Objects in Context," in *ECCV*, 2014.

[14] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *CVPR*, 2017.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single Shot Multibox Detector," in *ECCV*, 2016.

[16] C. Kyrkou, G. Plastiras, T. Theocharides, S. I. Venieris, and C. Bouganis, "DroNet: Efficient Convolutional Neural Network Detector for Real-time UAV Applications," in *DATE*, 2018.

[17] T. Ringwald *et al.*, "UAV-Net: A Fast Aerial Vehicle Detector for Mobile Platforms," in *CVPRW*, 2019.

[18] S. Jung, S. Hwang, H. Shin, and D. H. Shim, "Perception, Guidance, and Navigation for Indoor Autonomous Drone Racing Using Deep Learning," *RAL*, vol. 3, no. 3, 2018.

[19] P. Pandey, Q. He, D. Pompili, and R. Tron, "Light-Weight Object Detection and Decision Making via Approximate Computing in Resource-Constrained Mobile Robots," in *IROS*, 2018.

[20] F. Ozge Unel, B. O. Ozkalayci, and C. Cigla, "The Power of Tiling for Small Object Detection," in *CVPRW*, 2019.

[21] N. Audebert, B. Le Saux, and S. Lefèvre, "Segment-Before-Detect: Vehicle Detection and Classification Through Semantic Segmentation of Aerial Images," *Remote Sensing*, vol. 9, no. 4, 2017.

[22] G. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, "DOTA: A Large-Scale Dataset for Object Detection in Aerial Images," in *CVPR*, 2018.

[23] G. Plastiras, C. Kyrkou, and T. Theocharides, "Efficient ConvNet-based Object Detection for Unmanned Aerial Vehicles by Selective Tile Processing," in *Int. Conf. on Distributed Smart Cameras*, 2018.

[24] L. Ding, Y. Wang, R. Laganière, X. Luo, and S. Fu, "Scale-Aware RPN for Vehicle Detection," in *Advances in Visual Computing*, 2018.

[25] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, "Car Detection from Low-Altitude UAV Imagery with the Faster R-CNN," *Hindawi Journal of Advanced Transportation*, 2017.

[26] P. Petrides, C. Kyrkou, *et al.*, "Towards a Holistic Performance Evaluation Framework for Drone-based Object Detection," in *Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2017.

[27] 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*, 2016.

[28] S. Razakarivony and F. Jurie, "Vehicle Detection in Aerial Imagery: A Small Target Detection Benchmark," *Journal of Visual Communication and Image Representation*, vol. 34, 2016.

[29] L. Wen *et al.*, "UA-DETRAC: A New Benchmark and Protocol for Multi-object Detection and Tracking," *arXiv preprint arXiv:1511.04136*, 2015.

[30] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based Object Counting by Spatially Regularized Regional Proposal Network," in *ICCV*, 2017.

[31] M. Mueller, N. Smith, and B. Ghanem, "A Benchmark and Simulator for UAV Tracking," in *ECCV*, 2016.

[32] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian, "The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking," in *ECCV*, 2018.

[33] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, "Vision Meets Drones: A challenge," *arXiv preprint arXiv:1804.07437*, 2018.

[34] J. Gleason, A. V. Nefian, X. Bouyssounousse, T. Fong, and G. Bebis, "Vehicle Detection from Aerial Imagery," in *ICRA*, 2011.

[35] C. Kyrkou, S. Timotheou, P. Kolios, T. Theocharides, and C. Panayiotou, "Optimized Vision-Directed Deployment of UAVs for Rapid Traffic Monitoring," in *Conf. on Consumer Electronics*, 2018.

[36] Y. Lin and S. Saripalli, "Road Detection from Aerial Imagery," in *ICRA*, 2012.