AirCamRTM: Enhancing Vehicle Detection for Efficient Aerial Camera-based Road Traffic Monitoring

Rafael Makrigiorgis, Nicolas Hadjittoouli, Christos Kyrkou, Theocharis Theocharides
KIOS Research and Innovation Center of Excellence, Department of Electrical and Computer Engineering,
University of Cyprus,
Panepistimiou 1, Nicosia, Cyprus
{makrigiorgis.rafael,hadjittoouli.nicolas,kyrkou.christos,ttheocharides}@ucy.ac.cy

Abstract

Efficient road traffic monitoring is playing a fundamental role in successfully resolving traffic congestion in cities. Unmanned Aerial Vehicles (UAVs) or drones equipped with cameras are an attractive proposition to provide flexible and infrastructure-free traffic monitoring. However, real-time traffic monitoring from UAV imagery poses several challenges, due to the large image sizes and presence of non-relevant targets. In this paper, we propose the AirCamRTM framework that combines road segmentation and vehicle detection to focus only on relevant vehicles, which as a result, improves the monitoring performance by $\sim 2\times$ *and provides* $\sim 18\%$ *accuracy improvement. Furthermore, through a real experimental setup we qualitatively evaluate the performance of the proposed approach, and also demonstrate how it can be used for real-time traffic monitoring using UAVs.*

1. Introduction

Road traffic monitoring (RTM) is an important component of intelligent transportation system and is critical towards providing and analyzing traffic data to characterize the performance of a roadway system. Therefore, information gathered from traffic monitoring can determine areas with high traffic congestion that have to be addressed. The need to revolutionize the intelligent transportation sector has led to a number of technologies being employed from GPS [45], WiFi [49], UAVs [11], surveillance cameras [44]. Perhaps, the most promising out of these technologies are camera-equipped UAVs. The affordability of UAVs and ease of data capturing along with the advancement of computer vision and deep learning techniques provides a great opportunity to integrate these technologies together for the purposes of road traffic monitoring. Such capabilities are useful for a wide scenario of emerging traffic monitoring applications, 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) [30].

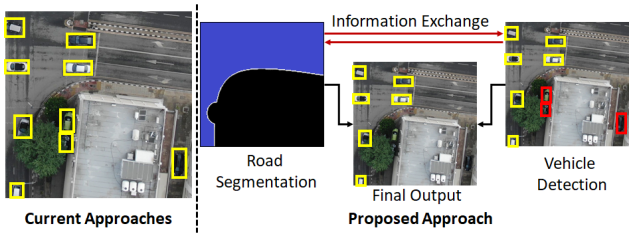

Figure 1. In contrast to traditional aerial road traffic monitoring methods, AirCamRTM enhances vehicle detection through road segmentation to improve monitoring performance and accuracy.

Through the literature most of the works focus on tackling one component of the more complicated traffic monitoring process with UAVs. Research on the topic has focused mainly on addressing the challenges of detecting vehicles in aerial and UAV imagery such as in [2], [29],[21]. Such works generally exploit generic detectors and retrain them for the task of vehicle detection and classification. Usually, they do not consider other algorithms, post-processing or pre-processing methods that could be useful for a more efficient traffic monitoring [18]. Some challenges faced by existing approaches are 1) how to pay attention only to the vehicles that matter to the traffic monitoring process, especially in scenarios of complex urban cities, and 2) how to reduce the computational complexity when monitoring hundreds of vehicles.

In order to meet the demands for more efficient monitoring with UAVs and facilitate real-time performance, we propose a composite visual processing pipeline, referred to as **AirCamRTM**. The main contribution is the integration of road segmentation with vehicle detection into a synergis-

tic framework based on deep learning that can tremendously enhance and improve the overall traffic monitoring by focusing only on the regions of interest (Fig. 1). First, we explore and evaluate different architectures for both vehicle detection and road segmentation on task-specific datasets and select the modules that provide the best accuracy-performance trade-off. Second, we integrate the aforementioned techniques into a visual traffic monitoring pipeline, and demonstrate how they can operate synergistically to improve the overall performance. Through evaluation on a composite dataset, we show accuracy improvements of up to $\sim 18\%$ for road-vehicle detection and observe a $\sim 2\times$ speedup compared to a road-agnostic approach, making it more suitable for real-time operation. Finally, through a real experimental setup, we show how the framework can be used to extract important traffic state information.

2. Related Work

2.1. UAV-based Vehicle Detection

Vehicle detection from UAVs has received a lot of attention in recent years mostly based on deep learning [8],[20]. Some common challenges confronted and tackled by the computer vision research are the computational efficiency and the detection of small size objects, since the vehicles from aerial images appear as small objects.

A hybrid approach using deep learning for feature extraction and Support Vector Machines for classification is presented in [2]. This method comes has higher computational intensity due to its brute force search method. Another hybrid method concentrates on utilising a 2-stage deep learning framework that performs tiled processing of aerial images via multi-label segmentation and then extracts the regions that correspond to vehicles to classify them into subcategories [3].

Other research addressed the critical problem of small object detection in aerial images. An approach in the literature was to implement a two-stage Faster R-CNN framework with Inception V2 as a backbone. However, this approach showed a decline in the trade-off between accuracy and computational intensity and so it cannot be utilised in video processing applications [31]. To reduce the processing time different approaches proposed searching for sub regions in the image [19],[24] [37]. More recent works have utilized single-shot convolutional neural networks for faster inference. For example, the work in [29] used the YOLOv3 network [41] for top-view vehicle detection. Beyond repurposing existing networks there has also been some work on realizing smaller networks targeting lightweight embedded processing platforms [21],[48], however, not all are developed for aerial imaging. Overall, Faster R-CNN and YOLO family of algorithms are amongst the most used for the purpose of vehicle detection [20]. In the on-demand traffic monitoring case we are interested in getting results as close to real-time as possible. Hence, since region-based methods incur higher computational complexity we focus herein on the YOLO family of networks [39], [41], [6] for developing a vehicle detector.

2.2. Road Extraction and Segmentation

Segmentation of aerial imagery and road extraction in particular, has been an active research area [32]. The majority of these techniques consider off-line scenarios and also utilize GIS data which necessitates an already mapped area. On the other hand, processing of aerial imagery for extraction of road segments in real-time applications is more challenging. Road extraction can be viewed as a binary labelling problem and has been tackled with different approaches over the years.

Some of the traditional approaches are using methods such as Gaussian Mixture Models on the color distribution [51], exploration of various feature spaces [14] [22] and more recently with the utilization of GPS [45] and WiFi data [49]. The aforementioned techniques lead to results that are applied to limited case scenarios, while dynamic road extraction is more complex due to either illumination, data limitation or complex road network[32]. Modern deep learning based approaches utilize the U-Net architecture [42] as basic structure with residual learning [12] [50]. However, they target satellite images which have a considerable different viewpoint from road traffic captured from UAV.

2.3. UAV Road Traffic Monitoring

Another line of works considers how the UAVs themselves are used in the traffic monitoring process. In [1] a study is presented on the different aspects of the use of UAVs for traffic monitoring from the flight planning, to the data acquisition component. In [11] the authors study the impact of mobile UAV trajectories on the event detection rate and the number of controlled vehicles. In [22] an optimization framework is proposed to find the best position and altitude for a UAV to have the best view of the road network segment given the heights of nearby buildings.

More inline with our work are those that use the outputs of perception modules to extract higher level information such as velocities and trajectories [36]. For example, the work in [17] develops an analysis framework by implementing an ensemble a classifier of Haar cascade and CNN to detect vehicles along with a KTL-based motion estimation to compute the motion, speed and density of the traffic. [4] proposes a semi-automatic way of extracting trajectories from aerial data. However, the majority of these works focus on simpler settings such as highways [18].

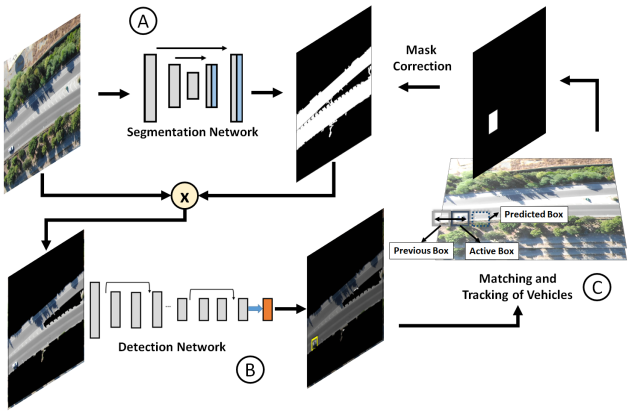

Figure 2. Proposed Synergistic AirCamRTM Framework: A) The road segmentation map is predicted. B) The vehicles in the segmented region are detected. C) The mask is corrected using the detected bounding boxes, while the algorithm is tracking and monitoring of the vehicles.

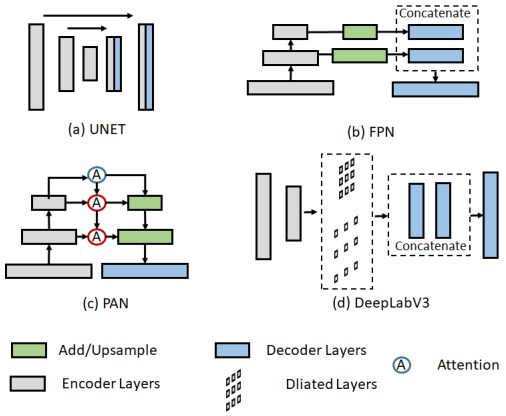

Figure 3. Semantic segmentation models meta architectures

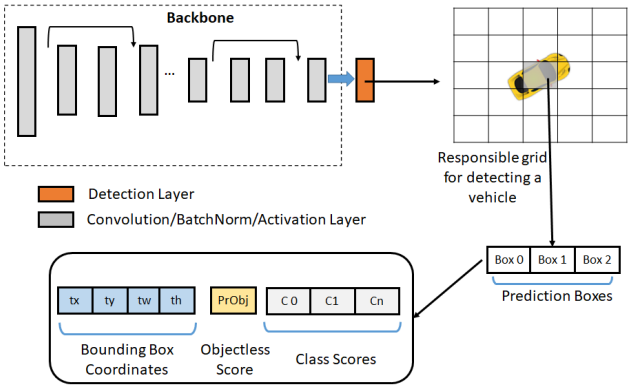

Figure 4. Single-Shot Vehicle Detection Overview

3. Proposed AirCamRTM Framework

To overcome limitations of standard methods we designed a pipeline of three distinct stages, that incorporates both a single shot detection as well as a road extraction network to isolate only relevant vehicles and estimate their traffic parameters. The overall computer vision pipeline as well as the interactions between stages are shown in Fig. 2.

3.1. Road Segmentation

The road extraction methodology involves using deep semantic segmentation networks to predict a road mask that would enhance the traffic monitoring task. In contrast to other works that use a single network architecture, our work first investigates a combination of encoders and decoders to identify the most suitable architecture for this task.

Primarily, the encoders developed for embedded devices were investigated that have the potential for an UAV onboard deployment. *EfficientNet* encapsulates a family of models, that provide the user a way to choose the desired trade-off between efficiency and accuracy [46]. *MobileNetV2* has gained large popularity in mobile applications as it leverages separable convolutions and inverted residuals [43]. Finally, *ResNet18*, is a small network that utilizes residual connections [15]. The above encoders vary in terms of depth, width and operations.

Moreover, we explore different decoder architectures that focus on the more widely used approaches that have demonstrated promising results for a variety of applications. Therefore, the architectures of UNet [42], FPN [27], PAN [25] and DeepLabV3 [7] are investigated. The networks above differ in terms of how they decode high-order information. Thus, the impact of skip connections [42], pyramidal structure [27], attention mechanisms [25] and atrous convolutions [7] is observed.

3.2. Vehicle Detection and Classification

In this paper, we specifically investigate the models based on the YOLO family which are compared in terms of accuracy and performance, instead of the region proposal method used in two-stage detectors. the YOLO family of networks starting with YOLOv2 [39] detects objects by dividing an image into grid units. Therefore, the detection speed is much faster than that of conventional methods. The feature map of the YOLO output layer is designed to output bounding box coordinates, the objectness score, and the class scores as shown in Fig. 4. Thus YOLO enables the detection of multiple objects with a single inference. The detection accuracy is low for small or dense objects when utilizing YOLOv2 [39]. Therefore, this paper investigates YOLOv3 [41] and YOLOv4 [6] capabilities to observe their potential in vehicle detection from an aerial point of view.

YOLOv3 consists of convolutional layers, as shown in Fig. 4, and applies a residual skip connection to solve the vanishing gradient problem of deep networks and uses an up-sampling and concatenation method that preserves fine-grained features for small object detection. The most promi-

nent feature is the detection at three different scales. This allows YOLOv3 to detect objects with various sizes. The predicted results of the three detection layers are combined and processed using non-maximum suppression. Recently, YOLOv4 [6] introduced a number of improvements both in the architecture of the convolutional network and its training methodology. Therefore, in terms of the trade-off between accuracy and speed, the YOLO family of methods is considered suitable for the task of real-time vehicle detection in aerial images. We compared both networks as well as their tiny versions to train top-view vehicle detectors and evaluate their performance and trade-offs.

3.3. Vehicle Detections Post Processing

The result of the vehicle detection process after being filtered with the segmentation masks, are a set of bounding boxes enclosing the detected vehicles. These bounding boxes are independent between frames, without any association, meaning that vehicles detected in one frame are not matched with vehicles detected in another frame. To estimate trajectories and velocities, however, it is necessary to track the same vehicles over time. In order to address this, first a matching approach is used to associate bounding boxes of the same vehicle, along with a tracking algorithm to maintain an estimate of the vehicle position for whenever a bounding box is missing either by occlusion or is not detected by the model.

3.3.1 Vehicles Temporal Association and Consistency

The algorithms that were adopted for vehicles' tracking are Hungarian algorithm [16] and Kalman Filter [47]. This tracking algorithm utilizes the Intersection over Union (IoU) score as a similarity metric for the bounding boxes and the Hungarian algorithm to associate these boxes over time. The IoU for achieving similarity between the boxes A, B is given in Eq. 1 and each is defined by $[t_x, t_y, t_h, t_w]$. A bounding box in the current frame is associated to the box in the previous frame that has the highest IoU score.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{1}$$

The tracking approach operates on a sequence of frames for which a set of bounding boxes is extracted in every frame. These bounding boxes of the previous frame are associated over with the ones in the current frame to form a tracking trajectory for each vehicle. Previous frame bounding boxes are either in an active or an inactive state depending on whether a current detection is associated with it. Each bounding box acquired in the first frame is given a unique ID and it is set as active and the Kalman filter is initialized and associated with it. For each succeeding frame the previous active boxes are being compared with the current boxes for matching. For inactive boxes we instead use the predicted location from the respective Kalman filters. Once a match has been found the corresponding Kalman filter is updated with that information.

The nearest detections are found by implementing K-Dimensional Tree (KDtree) [5], which is a space partitioning tree data structure for organizing points in the k-dimensional space. The tree also supports all-neighbors queries so it has the ability to search using the x, y coordinates. Once all new detections are put inside a KDtree, finding the nearest detections of each active vehicle is done by calling a query to the tree. Hence, the final tracking algorithm compares the IoU of each one of the nearest detections with the Kalman filter coordinates of the active vehicle being checked. After computing the IoU scores, if one or more of the compared detections scores have a higher value than a given threshold, then the detection with the highest IoU score gets associated with the vehicle that is being checked. Eventually, after all boxes in the current frame are checked for any match with the previous frame, those that haven't been matched might represent newly entered vehicles in the image. Furthermore, there might be some false detection cases which can be partly addressed by focusing only in the main road region. Hence, to further handle these cases, some consistency rules were introduced.

- Kalman prediction isn't performed in case the bounding box is detected less than k times, where k represents a threshold of frames below the framerate f. This prevents false positive detections to be missmatched with other active vehicles due to Kalman predictions.
- If a vehicle isn't detected for 3 consecutive seconds, depending on the framerate of the video stream, it will be set to inactive state and won't be checked again for matching with the future detections.
- When a bounding box prediction from the Kalman filter exceeds the image boundaries, it is set to inactive state, since it has left the camera's field of view.

3.3.2 Velocity and Direction Calculation

Following the association of bounding box over time for each vehicle, it is then possible to estimate their direction and velocity in order to provide traffic analytics. Direction estimation is done by simply subtracting the current x,y coordinates with the previous detection. We assign the vehicle to one of 8 possible directions top, bottom, left, right and their 4 intermediates.

To calculate the velocity of each vehicle, we estimate the distance travelled over time, by first converting pixels to actual distance. This can be computed using the ground

sample distance [23]. The ground sample distance algorithm uses the camera parameters, camera sensor dimensions ($C_h \times C_w$), focal length f and the image dimensions in pixels ($W \times H$) as well as the UAV altitude h, in order to find the representation of pixel to meters of the frame. The total GSD is calculated as shown in Eq. 2 and is measured in $km/pixels$. Then, the Euclidean distance covered d is converted from pixels in meters. Hence, calculating a more robust result using the average of the last M distances covered by a vehicle and multiplying it with the duration difference between frames t_f results to the current average velocity V_i of each vehicle i which is measured in Km/h.

$$GSD = max(\frac{h \times C_w}{f \times W}, \frac{h \times C_h}{f \times H})/1000 \tag{2}$$

$$V_i = \frac{\frac{1000}{M} \times \sum_{j=1}^{M} d_j}{t_f * 3600} \tag{3}$$

3.4. Mask Correction Using Detections

To improve the robustness of the segmentation output, detection information is utilized to update and fill missing regions that weren't predicted accurately by the segmentation model. As depicted in Fig. 2 a rectangle is created around each vehicle, having $1.5\times$ scale of its bounding box, and as vehicles move and reach missing segmentation regions, the segmentation output is expanded by adding the non-overlapping rectangular region. The updated mask is then used in subsequent frames to filter vehicle detections while it continues to be updated based on the updated detected vehicles. It is important to note that the update is only based on active vehicle detections which are more reliable and does not include transient detections.

3.5. Traffic Analytics

All of the data calculated during the aforementioned tracking processes such as the velocity, direction, trajectories and bounding boxes for each vehicle in each frame can be further processed to analyze statistics about each road segment. Such statistics are traffic density, fundamental diagram of traffic flow, spotting when and from where traffic is causing issues, average velocities, distances between vehicles and much more [30]. These statistics can then assist stakeholders and operators to apply the right traffic light policies and traffic managements schemes.

4. Datasets and Training Process

4.1. Datasets

Different datasets (Table 1) are used for training and evaluating both the vehicle detection and road segmentation models to find the most suitable architecture. In addition, a joint dataset was used for compositely training and testing of both models and evaluate that only relevant vehicles on the road are detected.

Dataset	Segmentation		Detection		Traffic Monitoring
	Train	Test	Train	Test	
OSM [34] (Images)	✓	✓			
UAVDT [10] (Images)			✓	✓	
Aerial KITTI [33] (Images)	✓	✓	✓	✓	
Custom UAV Videos					✓

Table 1. Datasets utilized to train & evaluation.

Detection: For the vehicle detection process, around $11k$ images were used in total, $9k$ for training and $2k$ for validation. From these images, $5k$ of them are taken from the UAVDT Dataset [10]. The UAVDT images were re-annotated to 4 classes ('Cars', 'Busses', 'Trucks' and 'Motorbikes'), instead of one that it was initially annotated. The rest of the images are taken from video footage collected through real UAV flights in varying altitudes, The footage captured took part in hourly sessions, on multiple busy locations in Cyprus, during morning hours where traffic is at its peak. Furthermore, since weather conditions should be optimal for a drone flight, the plethora of the data was taken during sunny days or in the worst-case scenario with light clouds. The onboard camera's pose was vertical to the ground, and the data were annotated from scratch to the same 4 classes as aforementioned. In total, $335,637$ vehicles were annotated in which includes $318,860$ light vehicles (e.g., cars), $165,41$ heavy vehicles (e.g., busses, trucks). As a note, the resolution of multiple images in the dataset differs as well as the altitude of the drone that took the images. The altitude varies between 150 to 250 meters high and the resolution from 720p to 4k

Segmentation: The open-source dataset of OpenStreetMap (OSM) [34] with the satellite images from various urban cities was utilized to train the road segmentation models. Because of the various image sizes in the dataset, a pre-processing stage was to resize images to 1024×1024 and randomly crop size of 512×512 for the training and the model was test on the center crops. A total number of $5,013$ training and $1,671$ testing labelled images were prepared.

Composite: For a more qualitative and joint evaluation, the Aerial KITTI dataset was introduced for training and testing the model [13] [33]. A smaller crop size of 512×512 was chosen for the Aerial KITTI images because of its limited amount of data. A total number of $1,272$ training and 318 testing labelled images were prepared. Note that, only road annotations were used throughout this paper and the cropped images without road annotations were excluded in both of the datasets. Moreover, we manually annotated the vehicles in the Aerial KITTI dataset for the object detection training and evaluation.

Custom UAV Videos: To evaluate the traffic monitoring capabilities, custom UAV videos of various traffic condi-

tions and locations were used for the estimation of the traffic monitoring parameters such as density and speed.

4.2. Training details

The neural networks for vehicle detection and road segmentation were all trained on an NVidia RTX 2080Ti and V100 GPUs, accordingly. For vehicle detection the DarkNet framework is used [38], while for road segmentation we use PyTorch library [35].

For the training of all the segmentation models, the same parameters of resolution, epochs, batch size, augmentation, learning rate and loss function were used. The models were trained with a batch size of 32 of with 512×512 size, for 100 epochs. The models were initialised with the ImageNet weights [9]. Furthermore, targeted augmentations were applied with a 0.3 probability. The learning rate was kept constant at 0.0001. Lastly, a combination of focal loss [28] and soft-jaccard loss was implemented for updating the weights during back-propagation.

All detection networks have been trained using their default configurations with some minor changes. Input resize of images of all configurations were set to 608×608. All chosen networks are using 9 anchors except tiny YOLOv3 which has 6 and were calculated using k-means clustering method [26]. For the tiny versions we modify the original tiny YOLOv4 [40] with 2 detection layers to have 3 instead since this improved the detection results with significant processing increase. This also resulted in a more stable detection. For augmentation, random resizing is used which resizes the network input size every 10 batches (iterations) from $\times 0.7$ to $\times 1.4$ while keeping initial aspect ratio of network size. In addition, for YOLOv4 models mosaic data augmentation is used, which combines 4 training images into one in certain ratios. Every network has been trained using a batch size of 64 for 200 epochs.

For the composite dataset (Aerial KITTI), the FPN decoder with ResNet18 as an encoder backbone and the Tiny YOLOv4@3layers were selected for the reasons that are further discussed in Section 5. The configuration was exactly the same as mentioned above except for the input size, for the detector model, which was set to 416×416 due to the size of the prepared KITTI images. The same train and test images were used to compositely evaluate both models.

5. Experimental Evaluation and Results

To verify the proposed framework, use both static image data as well as video frames collected from a real UAV flight. 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 as shown in Fig. 7. The data captured by the aerial camera is streamed to a laptop equipped with an Nvidia RTX 2060 GPU, where it is processed in real-time through the pipeline outlined in Section 3. First,

Model	mAP@0.5	IoU@0.5	FPS-GPU	FPS-CPU
Tiny YOLOV3	0.69	0.64	**33**	**11**
YOLOv3	0.79	0.74	11	0.9
Tiny YOLOv4@3layers	0.82	0.76	25	5
YOLOv4	**0.85**	**0.77**	10	1

Table 2. Performance of the different vehicle detection models on the laptop ground station.

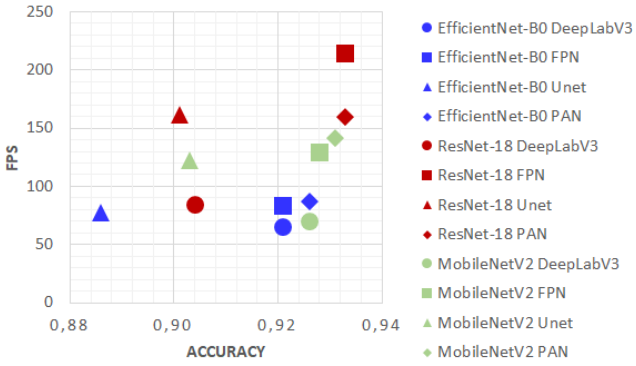

Figure 5. Performance of semantic segmentation encoder-decoder model combinations.

we investigate the sub modules for vehicle detection and road segmentation and then evaluate the complete framework by extracting traffic information.

5.1. Vehicle Detection Results

To compare with the different models (YOLOv3/4 as well as Tiny YOLOv3/4), we test their accuracy and performance on the detection datasets with the UAVDT and custom UAV traffic dataset and the comparison results as displayed in Table 2. A first observation is that the tiny YOLO networks can provide competitive processing speed even on CPU platforms. However, larger YOLO networks do require a GPU platform to provide competitive results. In terms of accuracy the V4 models perform slightly better than V3. Due to having really close accuracy to YOLOV4, which has the highest mAP and IoU, and being second best in terms of FPS the tiny-YOLOv4 is selected to be used in the combined experiments. The resulting performance of 25 FPS on the laptop ground station is enough for real-time applications while maintaining competitive accuracy.

5.2. Road Segmentation Evaluation

The segmentation-related experiments involved evaluating various encoder-decoder combinations in an attempt to evaluate their accuracy and performance. To evaluate this, the segmentation models were trained on an open-source dataset from OpenStreetMap (OSM) [34], for road detection as a binary class problem. The models where evaluated on the prepared crops of size 512×512, extracted from the full resolution images from the dataset. The results of the different models are shown in Fig. 5. Distinctively, the

Results	Without Segmentation	Segmentation
Accuracy	0.41	**0.59**
F1-score	0.58	**0.74**
mAP@0.5	0.44	**0.45**
mAP@0.3	0.50	**0.56**

Table 3. Composite evaluation results for the vehicle detection of Aerial KITTI test images. We consider as ground truth detections only the vehicles on the road, parked vehicles were excluded for fair comparison.

ResNet-18 with FPN network provides both the best frame-rate as well as accuracy. Hence, we utilize this model as the basis of the integrated framework. Beyond that, we observe that models with *ResNet* and *MobileNetV2* encoders and *PAN* and *FPN* meta-architectures provide the best results overall. *UNet*-based models seem to suffer more as the incorporation of low-level encoder features does not seem to work that well for the task of road segmentation.

5.3. Composite Evaluation

Furthermore, the composite dataset was utilized in order to evaluate the full potential of the combined framework. The amended Aerial KITTI dataset is used for this purpose. First, the ground truth labels were modified in order to contain only the on-road vehicles (1650 in total). Using this method, we can calculate the correct accuracy of the combined method of using the detector and the segmentation of the road in order to identify only the needed vehicles for traffic monitoring, which are the ones on the road. The segmentation network was further trained on the Aerial KITTI dataset for up to 200 epochs using early stopping strategy. The segmentation architecture achieved 77.7% mIOU and 96.3% mean accuracy.

Following this, two separate evaluations were conducted utilizing the segmentation and detection networks. For the first one, the input to the detector were the full images of the Aerial KITTI test dataset and for the second one, the segmentation mask was first applied to the image before passing it through the detector. Table 3 depicts the results of this evaluation on the vehicle detection task. The accuracy, F1 and Mean Average Precision (MAP) scores are higher with the mask which indicates that the mask is indeed working properly and the detector is able to identify the road vehicles and reduce the detection of unwanted vehicles. In addition, the number of total detections when using the full images is significantly larger (2× increase on average) which can negatively impact performance.

5.4. Traffic Monitoring Framework Evaluation

Finally, the integrated framework is evaluated on the video frame sequences from the experimental UAV setup capturing traffic in urban areas. First, we observe how the

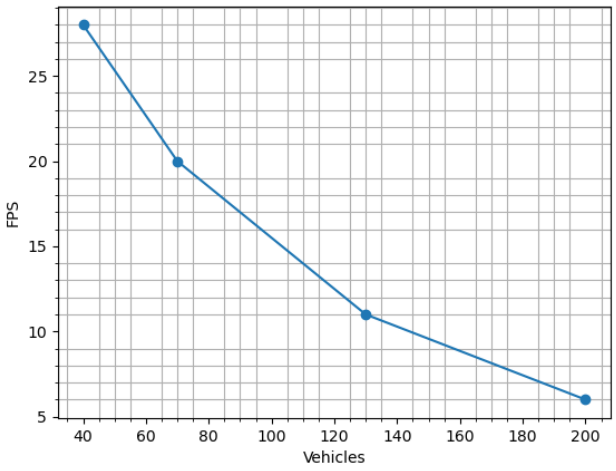

Figure 6. Impact of number of vehicles on performance of the whole framework. As the number grows tracking becomes the bottleneck hence, the need to only look at relevant vehicles.

Figure 7. 1) DJI Mavic is used to capture traffic video data. 2) The video is streamed through a local network. 3) A NGIX service is running on laptop to receive video and process it in real time

amount of detected vehicles can affect the performance of the whole pipeline as described in Section 3.3. As shown in Fig. 6, the number of vehicles drastically influences the time required to perform the associations and Kalman filter updates, and thus reduces the overall frame-rate. This exemplifies the need for efficient processing methodologies that only focus on relevant main road vehicles. By incorporating the road mask as shown in Fig. 8 the resulting number of detected vehicles can be dramatically reduced. On average we observe that the application of the road mask can reduce the number of vehicles from 155 to 70 on the test video frames in an urban environment, due to having numerous parked vehicles that aren't required for traffic monitoring, which can then improve the tracking performance dramatically. By utilizing the road segmentation mask the average FPS was around 20 and without the mask drops to approximately 8 FPS. In addition, we also test the proposed approach on a highway sequence where vehicles are only present on the road. This further examines if the segmentation mask hides some important vehicles of the main road, and hence negatively affects the performance of the detector. For both cases the accuracy is at 98% which means

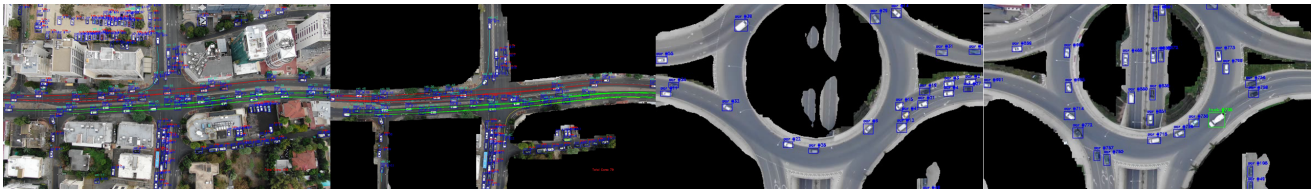

Figure 8. (From left to right) 1) Vehicle Detection Only. 2) Road Extraction and Detection. 3) Initial road segmentation. 4) Corrected segmentation using detection output.

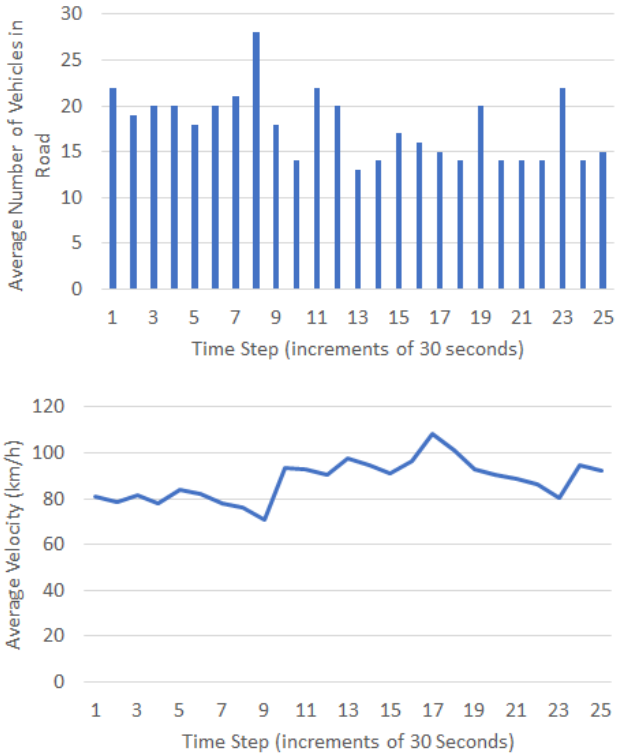

Figure 9. (Top) Average number of vehicles in road. (Bottom) Average vehicle speed.

that the segmentation mask does not hinder the vehicle detection. Additionally, when the mask is applied either by averaging it after a few frames or by applying it only in the first frame, it does not significantly affect the performance.

Finally, we experimentally verify the proposed framework through a real system shown in Figure 7. The UAV streams the video feed to a laptop that is connected on the same local network, which is able to process it in real time. We can gather the results and extract useful analytics such as velocity and density as shown in Figure 9. It is worth noting that extracting such data using only the detection part would significantly skew the results. This is because cars not affecting the road network would be considered. This is especially challenging for urban environments such as the one shown in Fig. 8, where a detector-only approach would also detect cars in non-road areas.

Following this, the limitations observed from the experimentation is that the road prediction is directly affected by the altitude and position of the drone along with the shadows and the complexity of the road network. This can be mitigated by the iterative feedback of the correction mask that operated surprisingly well and managed to shape the main road network entirely (Fig. 8). One potential drawback is that this method completely relies on the detector. Therefore, the assumption of having vehicles in the captured image is essential for the pipeline to function properly. Secondly, the detector functionality is sometimes hindered by long-period occlusions caused by bridges and tall buildings, which can be overcome by bringing in more consistency rules in the algorithm and adjusting the positioning of the drone as in [22].

6. Conclusion

In this work, AirCamRTM is introduced as a deep learning pipeline for road traffic monitoring from aerial video. The framework demonstrates how the simultaneous extraction of roads and vehicle detection can be integrated to gather traffic information efficiently. Through real experimental evaluation, it has been demonstrated that performance can be improved by $2\times$, while the road-segmentation-aware enhancements do not negatively impact the vehicle detection performance. We intent to further integrate the detection and segmentation networks to avoid redundant processing and improve the generalization capabilities. Currently, we are also working on expanding the dataset with data from different times and conditions, for both segmentation and detection.

Acknowledgements

The project is co-financed by the European Regional Development Fund and the Republic of Cyprus through the Cyprus Research Innovation Foundation ('RESTART 2016-2020' Program) (Grant No. INTEGRATED/0918/0056) (RONDA). This work was also supported by the European Unions Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE) and from the Government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

References

[1] Uav-based traffic analysis: A universal guiding framework based on literature survey. *Transportation Research Procedia*, 22:541 – 550, 2017.

[2] Nassim Ammour, Haikel Alhichri, Yakoub Bazi, Bilel Benjdira, Naif Alajlan, and Mansour Zuair. Deep learning approach for car detection in uav imagery. *Remote Sensing*, 9(4):312, Mar 2017.

[3] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre. Segment-before-detect: Vehicle detection and classification through semantic segmentation of aerial images. *Remote Sensing*, 9(4), 2017.

[4] Emmanouil N. Barmpounakis, E. Vlahogianni, and John C. Golias. Extracting kinematic characteristics from unmanned aerial vehicles. 2016.

[5] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[6] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[8] X. Chen, S. Xiang, C. L. Liu, and C. H. Pan. Vehicle detection in satellite images by parallel deep convolutional neural networks. In *2013 2nd IAPR Asian Conference on Pattern Recognition*, pages 181–185, Nov 2013.

[9] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[10] Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 370–386, 2018.

[11] M. Elloumi, R. Dhaou, B. Escrig, H. Idoudi, and L. A. Saidane. Monitoring road traffic with a uav-based system. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2018.

[12] Lin Gao, Weidong Song, Jiguang Dai, and Yang Chen. Road extraction from high-resolution remote sensing imagery using refined deep residual convolutional neural network. *Remote Sensing*, 11(5):552, Mar 2019.

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[14] A. Grote, M. Butenuth, and C. Heipke. Road extraction in suburban areas based on normalized cuts. In *International Archives of Photogrammetry and Remote Sensing*, pages 51–56.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[16] Roy Jonker and Ton Volgenant. Improving the hungarian assignment algorithm. *Operations Research Letters*, 5(4):171–175, 1986.

[17] R. Ke, Z. Li, J. Tang, Z. Pan, and Y. Wang. Real-time traffic flow parameter estimation from uav video based on ensemble classifier and optical flow. *IEEE Transactions on Intelligent Transportation Systems*, 20(1):54–64, 2019.

[18] Eui-Jin Kim, Ho-Chul Park, Seung-Woo Ham, Seung-Young Kho, and Dong-Kyu Kim. Extracting vehicle trajectories using unmanned aerial vehicles in congested traffic conditions. *Journal of Advanced Transportation*, 2019:1–16, 04 2019.

[19] Alexandros Kouris, Christos Kyrkou, and Christos-Savvas Bouganis. Informed region selection for efficient uav-based object detectors: Altitude-aware vehicle detection with cycar dataset. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 51–58, 2019.

[20] Michael Krump, Martin Ruß, and Peter Stütz. Deep learning algorithms for vehicle detection on uav platforms: First investigations on the effects of synthetic training. In Jan Mazal, Adriano Fagiolini, and Petr Vasik, editors, *Modelling and Simulation for Autonomous Systems*, pages 50–70, Cham, 2020. Springer International Publishing.

[21] C. Kyrkou, G. Plastiras, T. Theocharides, S. I. Venieris, and C. Bouganis. Dronet: Efficient convolutional neural network detector for real-time uav applications. In *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, pages 967–972, 2018.

[22] C. Kyrkou, S. Timotheou, P. Kolios, T. Theocharides, and C. G. Panayiotou. Optimized vision-directed deployment of uavs for rapid traffic monitoring. In *2018 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–6, 2018.

[23] Jon Leachtenauer and Ronals Driggers. *Surveillance and Reconnaissance Imaging Systems-Modeling and Performance Prediction*. 01 2001.

[24] Changlin Li, Taojiannan Yang, Sijie Zhu, Chen Chen, and Shanyue Guan. Density map guided object detection in aerial images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[25] H. Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid attention network for semantic segmentation. In *BMVC*, 2018.

[26] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

[27] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017.

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[29] Xin Luo, Xiaoyue Tian, Huijie Zhang, Weimin Hou, Geng Leng, Wenbo Xu, Haitao Jia, Xixu He, Meng Wang, and Jian Zhang. Fast automatic vehicle detection in uav images using convolutional neural networks. *Remote Sensing*, 12(12):1994, Jun 2020.

[30] Rafael Makrigiorgis, Panayiotis Kolios, Stelios Timotheou, Theocharis Theocharides, and Christos G Panayiotou. Extracting the fundamental diagram from aerial footage. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pages 1–5. IEEE, 2020.

[31] A. Mansour, W. M. Hussein, and E. Said. Small objects detection in satellite images using deep learning. In *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 86–91, 2019.

[32] H. Mayer, Stefan Hinz, U. Bacher, and E. Baltsavias. A test of automatic road extraction approaches. 2006.

[33] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun. Enhancing road maps by parsing aerial images around the world. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1689–1697, 2015.

[34] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org`, 2017.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[36] H. Q. Pham, M. Camey, K. D. Pham, K. V. Pham, and L. R. Rilett. Review of unmanned aerial vehicles (uavs) operation and data collection for driving behavior analysis. In Cuong Ha-Minh, Dong Van Dao, Farid Benboudjema, Sybil Derrible, Dat Vu Khoa Huynh, and Anh Minh Tang, editors, *CIGOS 2019, Innovation for Sustainable Infrastructure*, pages 1111–1116, Singapore, 2020. Springer Singapore.

[37] G. Plastiras, S. Siddiqui, C. Kyrkou, and T. Theocharides. Efficient embedded deep neural-network-based object detection via joint quantization and tiling. In *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 6–10, 2020.

[38] Joseph Redmon. Darknet: Open source neural networks in c, 2013.

[39] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.

[40] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[41] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[43] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[44] S. SASI PRIYA, S. Rajarajeshwari, K. Sowmiya, and P. Vinesha. Road traffic condition monitoring using deep learning. In *2020 International Conference on Inventive Computation Technologies (ICICT)*, pages 330–335, 2020.

[45] Tao Sun, Zonglin Di, Pengyu Che, Chun Liu, and Yin Wang. Leveraging crowdsourced GPS data for road extraction from aerial imagery. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 7509–7518. Computer Vision Foundation / IEEE, 2019.

[46] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019.

[47] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[48] A. Womg, M. J. Shafiee, F. Li, and B. Chwyl. Tiny ssd: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 95–101, 2018.

[49] M. Won, S. Sahu, and K. Park. Deepwitraffic: Low cost wifi-based traffic monitoring system using deep learning. In *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 476–484, 2019.

[50] Z. Zhang, Q. Liu, and Y. Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018.

[51] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi. Efficient road detection and tracking for unmanned aerial vehicle. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):297–309, 2015.