Lightweight and Effective Convolutional Neural Networks for Vehicle Viewpoint Estimation From Monocular Images

Lightweight and Effective Convolutional Neural Networks for Vehicle Viewpoint Estimation from Monocular Images

Simone Magistri[‡*] [1], Marco Boschi[§*] [2], Francesco Sambo[†], Douglas Coimbra de Andrade[†], Matteo Simoncini[†‡], Luca Kubin[¶], Leonardo Taccari[†], Luca De Luigi[§], Samuele Salti[§]

***Abstract*—Vehicle viewpoint estimation from monocular images is a crucial component for autonomous driving vehicles and for fleet management applications. In this paper, we make several contributions to advance the state-of-the-art on this problem.**

We show the effectiveness of applying a smoothing filter to the output neurons of a Convolutional Neural Network (CNN) when estimating vehicle viewpoint. We point out the overlooked fact that, under the same viewpoint, the appearance of a vehicle is strongly influenced by its position in the image plane, which renders viewpoint estimation from appearance an ill-posed problem. We show how, by inserting in the model a CoordConv layer to provide the coordinates of the vehicle, we are able to solve such ambiguity and greatly increase performance. Finally, we introduce a new data augmentation technique that improves viewpoint estimation on vehicles that are closer to the camera or partially occluded.

All these improvements let a lightweight CNN reach optimal results while keeping inference time low. An extensive evaluation on a viewpoint estimation benchmark (Pascal3D+) and on actual vehicle camera data (nuScenes) shows that our method significantly outperforms the state-of-the-art in vehicle viewpoint estimation, both in terms of accuracy and memory footprint.

***Index Terms*—Azimuth, Convolutional Neural Networks, Machine Learning, Monocular Images, Vehicles, Yaw.**

I. Introduction

VEHICLE viewpoint estimation [1] consists in estimating the vehicle azimuth or yaw angle, i.e., the rotation of the vehicle around the axis perpendicular to the road plane with respect to a reference point of view, like a camera (see Fig. 1). Vehicle viewpoints enrich the semantic information about the road scene provided by traditional perception tasks, such as object detection, semantic segmentation and depth estimation, and lead to a better understanding of the road scene [2], [3]. Furthermore, viewpoint estimation enables accurate prediction of future vehicle motion [4], thus it is of considerable interest for applications in which prediction or classification of road maneuvers are key, like driver warning systems and risk estimation in the context of fleet management [5].

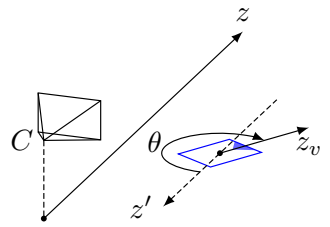


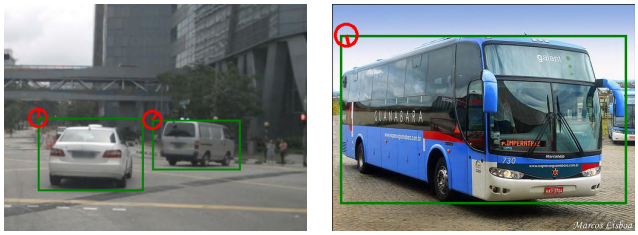
Fig. 1. Definition of azimuth θ for the blue vehicle, whose front is denoted by the solid triangle. C is the camera and z is the direction of the optical axis on the road plane. z' is the global z translated so to pass through the blue vehicle center flipped to face towards the camera and z_v is pointing in the direction of the vehicle front from the vehicle center: θ is the angle from z' to z_v measured clockwise.

For this work, we consider the scenario of a single monocular camera mounted inside the windshield of a moving vehicle, suitable for the context of after-market fleet-management applications.

Viewpoint estimation from a single monocular image has inherent difficulties such as radial distortion, motion blur, occlusion and lack of depth information. Furthermore, in a realistic scenario the camera is not mounted exactly in the same spot for every vehicle, leading to non-fixed points of view and lack of some extrinsic camera parameters, namely camera height, roll and pitch. In light of this, we aim at defining a solution that generalizes to multiple vehicle makes and models and that is independent of the camera mounting position.

Current approaches in the literature tackle the vehicle viewpoint estimation problem from a single image within the context of 3D monocular object detection [6], [7]. The main problems concerning these methods are the need to estimate depth or 3D information and the requirement of accurate camera parameters for correct localization. Furthermore, they may require higher computational resources if compared to 2D object detectors because of extra computation for 3D position estimation [8]. Thus, we explore different solutions for the viewpoint estimation problem from monocular images without trying to estimate the vehicle 3D position and we present a lightweight real-time deep learning model which uses data from a single on-board monocular camera.

*These authors contributed equally.
†Verizon Connect Research, Florence, Italy
‡University of Florence, Italy
§University of Bologna, Italy
¶ Work done while in Verizon Connect Research, Florence, Italy
Email: [1] simone.magistri@unifi.it
Email: [2] marco.boschi@verizonconnect.com
Digital Object Identifier 10.1109/TITS.2022.3216359



(a) nuScenes. (b) Pascal3D+.

Fig. 2. Samples from nuScenes (2a) and Pascal3D+ (2b) datasets. The azimuth label of each vehicle is identified by the red circle placed in the top-left corner of each bounding box.

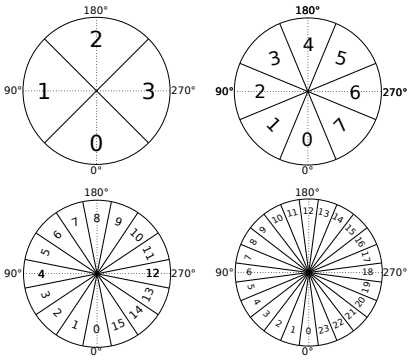


Fig. 3. Azimuth bins we adopted to evaluate our models performance. The frame of reference assumes slice 0 facing towards the camera.

Since estimating vehicle viewpoint requires first to identify the position of the vehicle within the image, we decouple the viewpoint estimation problem from the detection one, assuming to have the vehicle bounding boxes in input. This choice is justified by the fact that training a detector jointly with a viewpoint estimator results in a model that is both less versatile and harder to maintain or adapt [9]. Our models are tested on the nuScenes [10] dataset, a large-scale autonomous driving dataset, and on the Pascal3D+ [11] dataset, a popular benchmark for object viewpoint estimation (see Fig. 2).

The main contributions of this work can be summarized as follows:

- We develop a novel lightweight and accurate vehicle viewpoint model, referred to as *fine-grained* model, which improves upon the *coarse-grained* model originally proposed in [9], where it was referred to as *multi-task* model. The proposed model is independent from any 2D object detector and does not need any extra camera information, such as extrinsic and intrinsic camera parameters.
- We show that vehicle viewpoint estimation from a monocular image is strongly dependent on the position of the target vehicles in the image plane and we show that, by providing this information to the model, performance can be improved.
- We show that preserving the original aspect ratio of the source image considerably improves the viewpoint estimation.
- We successfully adapt the Siamese approach, developed by [12], in the context of vehicle viewpoint estimation.
- Since traditional approaches on viewpoint estimation methods are very sensitive to truncated objects [13], [14], we propose a task-specific data augmentation technique, which we dubbed *viewpoint crop*, to improve viewpoint estimation of truncated vehicles.

All these contributions but the Siamese approach are improvements to the preliminary version of this work published in [9]. The paper is organized as follows: in Section II we provide the current state of the art approaches for vehicle viewpoint estimation, in Section III we describe the *fine-grained* and *coarse-grained* architectures and present our contributions; in Section IV, we introduce the datasets we employ in our benchmarks and we provide details on the pre-processing we used; in Section V we compare our proposed models with viewpoint estimation state-of-the-art methods and provide an ablation study on the effectiveness of our contributions; finally, in Section VI we summarize our results and we list some possible future developments.

II. Related Works

Most of the methods in the literature address the viewpoint estimation problem with a classification approach, which was shown by Massa et al. in [15], [16] to outperform the regression one. Focusing on the classification approach, Ghodrati et al. [17] leverage image features extracted by a Convolutional Neural Network (CNN) to estimate a discretized object viewpoint. Tulsiani et al. [14] directly estimated the viewpoint from the object images, using a CNN based on VGG16 architecture [18], taking into account the object class. Zhou et al. [19] reduce the prediction error of [14], changing the convolutional backbone to the more recent residual architecture ResNet18 [20]. Su et al. [21] introduced a discretization into 360 bins for viewpoint prediction and they proposed a geometric structure aware loss function. Divon et al. [12] proposed a CNN-based architecture that jointly solved detection, classification and viewpoint estimation, introducing a loss function based on the idea of the Siamese Networks [22]. More recently, Xiao et al. [23] used a class-agnostic CNN to estimate object poses and introduced a contrastive learning method based on a contrastive loss and pose-aware data-augmentations to improve the performance.

III. Proposed Models

The objective of this research is to develop a framework based on CNNs to estimate vehicle viewpoints given a single road view image and 2D vehicle detection information. Given the road view image, we crop the vehicles according to their bounding box information and we feed the cropped images, together with the bounding box coordinates (see Section III-C), into our proposed models. In the original work [9] we investigated whether vehicle classes can be useful to improve the network predictions. Since we found that vehicle class provides marginal improvements, in this extended work we ignore the class information, focusing on a model using only visual and positional information.

We address azimuth estimation as a classification problem, thus we discretize the ground truth azimuth labels into 360

sectors and we design our models output layer to have 360 units, i.e. the *logits*. This 360 dimensional vector is fed to a softmax classifier [24] to predict the azimuth angle at different levels of quantization. In particular, we evaluate the performance of our models by mapping their predictions into discrete uniform bins, equally spanned in the unit circle (Fig. 3), with four scale levels ranging from coarser to finer, i.e., the standard evaluation protocol of Pascal3D+.

The developed models share a common structure: a convolutional backbone, followed by a Global Average Pooling Layer and by a linear output layer with 360 units. The proposed models differ for the training loss and for the specific logits smoothing technique.

In the remainder of this section, we provide a detailed description of the explored architectures, the *fine-grained* (Fig. 4b) and the *coarse-grained* model (Fig. 4c), and we discuss the need to take into account the coordinates of the vehicles to properly estimate their viewpoint, the adaptation of the Siamese loss to our setup, and the proposed data augmentation, which improves robustness in the presence of truncated vehicles.

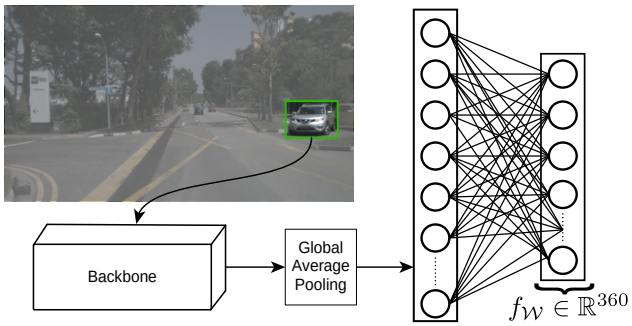
A. Fine-grained Model

The *fine-grained* model is an evolution of the single-task model proposed in our original work [9], i.e., a single output head predicting a single level of discretization. We increase the granularity of the prediction with more bins of a smaller size, resulting in a more *fine-grained* model.

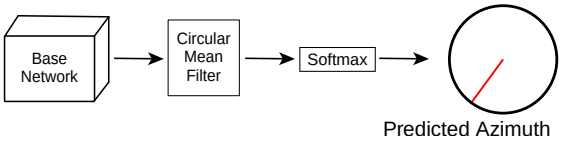
In this model we apply a circular moving average filter (see Fig. 4b) with stride 1 and window size 15, preceding the softmax activation function. We dub it *circular* because of the padding strategy used, with values wrapping around at the edges: given the network output $(l_1, l_2, \ldots, l_N)$ we add a 7 element wide padding as $(l_{N-6}, \ldots, l_{N-1}, l_N, l_1, l_2, \ldots, l_N, l_1, l_2, \ldots, l_7)$, resulting in a vector of $N + 14$ elements. This, after application of the filter, is reduced to a cardinality of $N + 14 - (15 - 1) = N$, the same as the network output.

The intuition behind the usage of a circular moving average filter comes from the error analysis of viewpoint estimation methods by Redondo-Cabrera et al. [13]. They show that a consistent portion of the errors made by the viewpoint models concentrates on nearby and opposite viewpoint predictions with respect to the ground truth viewpoints, i.e., the prediction error e usually satisfies either $15° \leq e \leq 30°$ or $e > 160°$. In order to correct for these errors, we would like the viewpoint probabilities distribution of the network to be unimodal [25], i.e. the probability mass should gradually decrease on both sides of the viewpoint that has most of the mass and should be centered on the correct viewpoint. A circular mean filter has the objective of smoothing the 360 network logits in order to filter isolated peaks and to gradually decrease the probability mass function away from the overall highest logit.

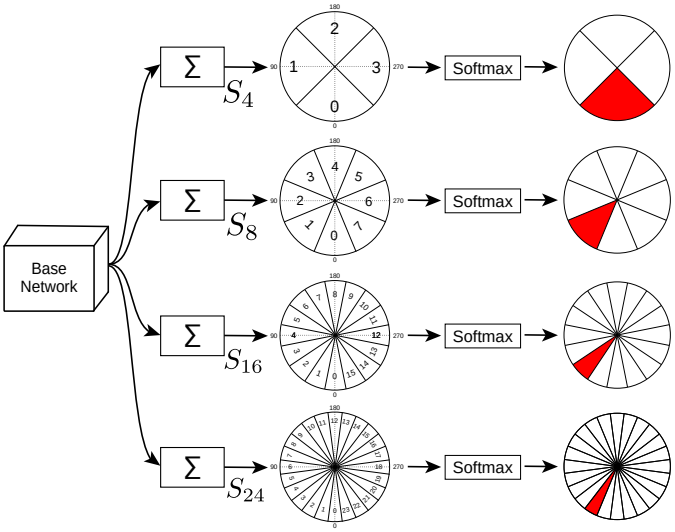
The loss function used to train this model is the categorical cross entropy loss between the smoothing filter output and the ground truth azimuth discretized into 360 sectors.



(a) Base network shared by the two models.



(b) Fine-grained model. The model has a single output head which results in the azimuth prediction as an angle with 1° of precision. Such precision is achieved by smoothing the logits $f_{\mathcal{W}}$ with a circular mean filter and allows the predicted azimuth to be used as-is or discretized with large bins.



(c) Coarse-grained model. The output of the model consists of 4 heads, which sum in a circular fashion the network logits f_W. Each head obtains a prediction at a different discretization level. S_α represents the summation performed for α bins with $\alpha \in \{4, 8, 16, 24\}$.

Fig. 4. Proposed models architecture.

B. Coarse-grained Model

The *coarse-grained* model output layer [9] (see Fig. 4c) consists of four sums of the network logits, resulting in multiple output heads. The summations are designed to map the network logits to the four discretization levels defined in Fig. 3. The summation for α bins, where $\alpha \in \{4, 8, 16, 24\}$, can be thought of as a mono dimensional circular convolution with a unitary filter with window and stride fixed to $\lceil \frac{360}{\alpha} \rceil$. Similarly to the *fine-grained* model, applying these filters results in assigning less importance to isolated peaks in the predicted probability mass function.

The loss function adopted to train this model is made up of the sum of four categorical cross entropy losses between the ground truth azimuth discretized into 4, 8, 16 and 24 bins

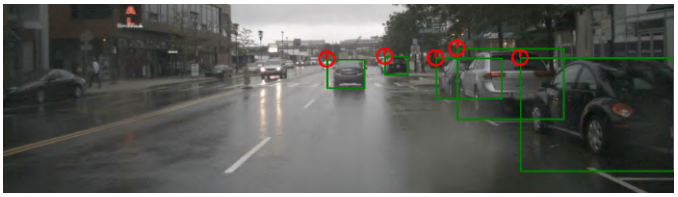


Fig. 5. Example from the nuScenes dataset of the intrinsic ambiguity of viewpoint and appearance.

and the corresponding logits summation. This kind of loss is designed to take into account errors at different granularities, to smooth isolated peaks in the probability mass of the network probability distribution and to train a single model able to predict viewpoint at different quantizations.

C. Vehicle Coordinates

Estimating viewpoint reasoning only on the appearance of the vehicle within the bounding box suffers of an intrinsic ambiguity. Indeed, a vehicle with the same azimuth angle with respect to the camera exhibit different appearance onto the image plane of the camera according to its horizontal distance with respect to the camera optical axis. For instance, a vehicle whose azimuth is $180°$, i.e., moving in the same direction of the camera, will have only its rear visible in the image if it is in front of the camera, but both the rear and the left side if it is in the adjacent right lane [26] (see Fig. 5).

To make the network able to estimate the same azimuth angle in the presence of different appearances, we propose to provide also pixel coordinates in the original image of the cropped bounding box with a CoordConv layer [27]. Such layer, when applied to a color image represented as a 3D tensor with channels, width and height as dimensions, will construct two additional channels: one for the X coordinates along the horizontal axis of the image and one for Y coordinates along the vertical axis of the image. Each channel is constructed using as values the coordinates of the elements themselves for that dimension, i.e., for the X channel the first column will be filled with 0, the second with 1 and so on, while for the Y channel the first row is all 0, the second all 1, and so on; both coordinates channels are then normalized to $[-1, 1]$. The result is an image with five channels, three carrying color information and two for the coordinates. After creating the additional channels for the input, the CoordConv layer behaves like a normal convolutional layer.

Since we are interested in the coordinates relative to the full image and our models accepts only the crop corresponding to the 2D bounding box of the target vehicle, we first apply the coordinate channel generation, then crop the bounding box and finally feed the image to the network.

D. Siamese Network

A Siamese network [22] consists of two or more copies of the same convolutional neural networks that share learnable weights. In the context of object viewpoint estimation, each sub-network takes in input a different image and improves the standard neural network training by adding geometrical constraints into the training loss.

In our work, we adapt to our task the Siamese loss, originally proposed by Divon et al. [12] to jointly train a 2D object detector and a viewpoint estimator. We propose a Siamese approach for a model trained independently from a 2D object detector and we modify the Siamese network input such that it accepts both the bounding box vehicle coordinates and the input image, as discussed above.

Formally, let us consider the triplet (X, c, θ) where X is the cropped vehicle image, $\theta \in [0, 360)$ is the corresponding azimuth label and $c = (x_0, y_0, x_1, y_1)$ represents the bounding box coordinates of the vehicle in the source image adopting the standard top-left bottom-right coordinate system.

The Siamese network is fed with two images: X and its horizontally flipped version X_{flp}. Flipping the image X implies that both the bounding box coordinates in the original image and the ground truth azimuth label associated to X_{flp} are horizontally flipped. Let θ_{flp} be the mirrored version of θ on the Y axis and let $c_{flp} = (w - x_0, y_0, w - x_1, y_1)$ be the flipped vehicle coordinates in a image with width w. We implement our methodology by defining a training loss consisting of three terms: $\mathcal{L}$, $\mathcal{L}_{flp}$ and D. $\mathcal{L}$ is the loss on the image X with ground truth θ, $\mathcal{L}_{flp}$ is the loss on the image X_{flp} with ground truth θ_{flp} and $D : \mathbb{R}^{360} \times \mathbb{R}^{360} \mapsto \mathbb{R}$ is a distance function between the network output $f_{\mathcal{W}}$ on the image and on its flipped version $f_{\mathcal{W}}^{flp}$. The final Siamese training loss can be summarized as follows:

$$\mathcal{L}_s = \mathcal{L} + \mathcal{L}_{flp} + \lambda D(f_{\mathcal{W}}, flip(f_{\mathcal{W}}^{flp})) \tag{1}$$

where $f_{\mathcal{W}}, f_{\mathcal{W}}^{flp} \in \mathcal{R}^{360}$, both parametrized by shared weights $\mathcal{W}$, are the network outputs for the pairs (X, c) and (X_{flp}, c_{flp}) respectively, $\lambda \in \mathbb{R}$ is a hyperparameter, and *flip* : $\mathbb{R}^n \mapsto \mathbb{R}^n$ is an operator mapping $y = (y_1, y_2, \ldots, y_n)$ to $flip(y) = (y_1, y_n, y_{n-1}, \ldots, y_2)$, i.e. it swaps logits between an angle α and its correspondent $-\alpha$ under the horizontal flip while leaving the first logit, corresponding to $0°$, fixed.

We expect that the distance between the output of the network on the image X and the *flip* of the network output on X_{flp} should be zero. We used as D function either the squared ℓ_2 distance, as proposed in [12]

$$D(x, y) = \|x - y\|_2^2 , \tag{2}$$

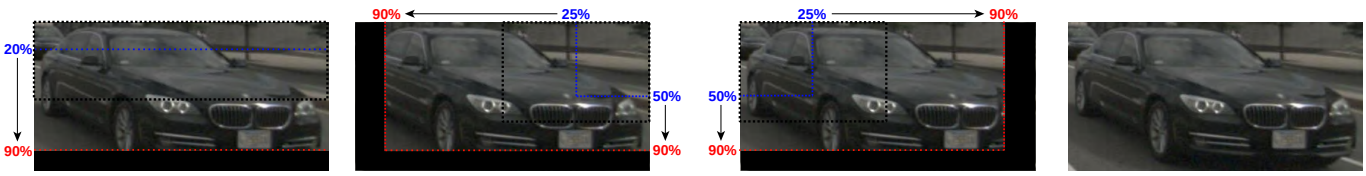
or the angular distance

$$D(x, y) = \frac{1}{\pi} \arccos \frac{x \cdot y}{\|x\| \, \|y\|}, \tag{3}$$

where $x, y \in \mathbb{R}^n$.

E. Viewpoint Crop

Traditional approaches on viewpoint estimation are highly sensitive to truncated vehicles [13], [14]. In the context of vehicle viewpoint estimation from a single on-board monocular camera, vehicles can be heavily truncated, since they can be really close to it or placed on the margins of the camera field of view. For this reason we propose a task-specific data augmentation for vehicle viewpoint estimation, that we



(a) Option 1: close front vehicle.

(b) Option 2: vehicle is close to the camera and partially beyond the left margin of the camera field of view.

(c) Option 3: vehicle is close to the camera and partially beyond the right margin of the camera field of view.

(d) Option 4: fully visible vehicle.

Fig. 6. The *viewpoint crop* data augmentation. The blue and the red dashed lines represent respectively the minimum and the maximum dimensions of the crop for each option, while the black dashed line represents a possible crop applied during training.

name *viewpoint crop*, with the aim of improving performance on truncated vehicles. The proposed data augmentation is performed online during training and chooses with probability $25\,\%$ one of four cropping options, each designed to simulate different truncations of vehicles captured by a moving camera:

1) The vehicle is directly in front of the camera and very close to it (Fig. 6a), simulated by cropping only the image bottom side;
2) The camera is very close to a vehicle placed on its left side, i.e. part of the vehicle is just beyond the left edge of the field of view of the camera (Fig. 6b), simulated by cropping the left and bottom image side.
3) The mirrored version of the previous case, simulating vehicles on the right side of the camera (Fig. 6c).
4) The vehicle is fully visible (Fig. 6d).

The crop target sizes are extracted as $w \sim \mathcal{U}(w_{min}, w_{max})$ and $h \sim \mathcal{U}(h_{min}, h_{max})$, where $\mathcal{U}$ stands for the uniform probability distribution and w_{min}, h_{min}, w_{max}, h_{max} represent respectively the lower and the upper bounds of the percentage of the vehicle original width and height. In Figure 6, we provide the bounds of each cropping option.

IV. Dataset

Pascal3D+ [11] is an object viewpoint estimation dataset. It contains 12 object classes and samples from Pascal VOC 2012 [28] and a subset of Imagenet [29], enriched with 3D annotations (azimuth, elevation and tilt).

nuScenes [10] is a large-scale autonomous driving dataset containing a full suite of vehicular sensor data and 3D bounding box and viewpoint annotations.

All images in nuScenes are acquired by 6 cameras mounted on the front, front-right, front-left, back-right, back-left and back of a moving vehicle. Thus, the elevation and tilt of the objects are almost the same for each image. Pascal3D+ images, on the other hand, are captured by a single fixed camera, located at different positions. As a result, elevation and tilt of the objects may significantly vary across different images.

Data Pre-Processing In the case of nuScenes, we collect the images from the 6 cameras provided, projecting the 3D bounding boxes to each image plane, following the procedure provided in the development kit of [10]. We extract crops around the vehicles from the projected 3D boxes and we retain the azimuth. For training and testing we retain only those bounding boxes with enough data to correctly infer the orientation according to the same criteria defined in [9].

TABLE I
PASCAL3D+ AND NUSCENES SAMPLES DISTRIBUTION.

		%					
Dataset	**Split**	**Bike**	**Bus**	**Car**	**Motorbike**	**Truck**	**Total**
nuScenes	Train	1.7	3.0	74.9	1.7	18.6	$\sim$ 400k
	Valid	2.4	3.4	74.8	2.3	17.1	$\sim$ 35k
	Test	2.1	3.5	72.8	2.2	19.5	$\sim$ 90k
Pascal3D+	Train	14.9	11.9	52.5	13.9	6.8	$\sim$ 10k
	Valid	19.8	14.1	46.1	17.5	2.5	$\sim$ 1k
	Test	18.6	14.7	44.4	17.8	4.5	$\sim$ 2k

In the case of Pascal3D+, we extract tight crops around the vehicles using the provided 2D bounding boxes and we retrieve the provided azimuth angle. Furthermore we manually identify and label the trucks contained in Pascal3D+ to preserve consistency with nuScenes vehicle classes.

Finally, we define one training validation split for nuScenes and one for Pascal3D+: for the latter we split 50/50 its official training set stratifying by vehicle type, while for nuScenes the split is obtained choosing respectively 650 and 50 scenes for training and validation set.

As for the test set, for Pascal3D+ we use the Pascal VOC 2012 validation set as test set, as specified in [11], including the vehicles annotated as *difficult*, *occluded* and *truncated*, while for nuScenes, we use the official validation set. In Table I the percentage of vehicle types in each split and the effective size of datasets are shown.

V. Experimental Results

Our experimental results are reported on the nuScenes and Pascal3D+ dataset. For each model, we measure the accuracy per class, the average accuracy across vehicle classes and the overall accuracy for each discretization level (i.e. 4, 8, 16, 24 bins) on the test set. Using these metrics, we compare the *fine-grained* model and the *coarse-grained* model performance with state-of-the-art approaches of viewpoint estimation [19], [21], [14], [23] and the preliminary version of this work [9].

In the ablation study, we show how much the convolutional backbone and our contributions affect the performance of the *coarse-grained* and the *fine-grained* models. For this analysis, we use the nuScenes dataset, since it is more representative of the task we want solve, i.e., vehicle viewpoint estimation from road view monocular images. Finally, we provide the inference time and the memory footprint of the developed models.

TABLE II
STATE OF THE ART COMPARISON. "RESNET18 CS" REFERS TO "RESNET18 CLASS SPECIFIC" [19], [14], AND "MBIKE" TO THE "MOTORBIKE" CLASS.

Bins	Architecture	% Bike	Bus	Car	Mbike	Truck	Avg	Total
4	Render for CNN [21]	57.35	79.59	89.92	66.65	79.49	74.60	86.33
	ResNet18 CS [19], [14]	59.95	82.80	90.95	73.57	81.23	77.70	87.74
	PoseContrast [23]	64.36	85.19	91.80	71.16	83.79	79.26	88.98
	Multi-Task [9]	65.66	**86.20**	92.26	73.36	83.65	80.23	89.40
	Coarse-grained Model	65.56	84.93	**93.83**	**74.23**	84.66	80.64	90.71
	Fine-grained Model	**66.23**	85.47	93.77	73.67	**85.02**	**80.83**	**90.75**
8	Render for CNN [21]	40.88	73.65	81.95	54.35	68.72	63.91	77.61
	ResNet18 CS [19], [14]	49.87	76.86	83.11	60.40	71.05	68.26	79.35
	PoseContrast [23]	48.36	81.21	85.35	60.81	75.10	70.17	81.89
	Multi-Task [9]	54.75	80.61	85.46	62.40	75.24	71.69	82.15
	Coarse-grained Model	**55.74**	**82.36**	89.38	**66.03**	79.58	**74.62**	86.01
	Fine-grained Model	**55.74**	82.04	**89.81**	63.78	**79.77**	74.23	**86.29**
16	Render for CNN [21]	29.87	62.62	71.27	36.58	57.86	51.64	66.73
	ResNet18 CS [19], [14]	29.71	66.88	73.31	41.75	60.85	54.50	69.05
	PoseContrast [23]	37.82	70.15	74.88	45.39	64.04	58.46	71.18
	Multi-Task [9]	39.95	71.23	76.17	44.52	65.76	59.53	72.52
	Coarse-grained Model	**44.47**	74.86	82.60	**54.15**	70.64	65.34	78.58
	Fine-grained Model	44.16	**75.68**	**83.83**	51.33	**71.73**	**65.35**	**79.64**
24	Render for CNN [21]	22.81	56.07	64.96	27.36	50.52	44.34	60.13
	Resnet18 CS [19], [14]	21.82	59.98	67.38	31.97	53.57	46.94	62.70
	PoseContrast [23]	26.08	65.03	70.73	37.19	59.10	51.63	66.59
	Multi-Task [9]	29.19	63.57	70.11	35.45	58.09	51.28	65.92
	Coarse-grained Model	34.81	72.57	79.03	43.55	67.07	59.41	74.77
	Fine-grained Model	**36.99**	**73.33**	**80.42**	**44.77**	**67.92**	**60.70**	**76.05**

(a) nuScenes.

Bins	Architecture	% Bike	Bus	Car	Mbike	Truck	Avg	Total
4	Render for CNN [21]	68.5	81.5	68.4	73.2	N/A	72.9	74.3
	ResNet18 CS [19], [14]	73.5	82.9	71.3	76.1	67.4	74.3	74.1
	PoseContrast [23]	72.7	80.4	73.6	75.5	68.6	74.2	74.5
	Multi-Task [9]	71.6	**84.0**	**80.0**	74.9	**76.7**	77.5	78.0
	Coarse-grained Model	74.9	**84.0**	76.1	77.0	**76.7**	77.8	77.2
	Fine-grained Model	**76.3**	83.6	77.3	**80.8**	**76.7**	**79.0**	**78.7**
8	Render for CNN [21]	58.6	70.5	58.3	62.5	N/A	62.5	63.2
	ResNet18 CS [19], [14]	62.3	73.3	61.0	63.7	61.6	64.8	63.5
	PoseContrast [23]	60.6	**74.0**	66.0	65.5	54.7	64.1	65.6
	Multi-Task [9]	60.3	71.2	**70.2**	66.4	**68.6**	67.3	67.8
	Coarse-grained Model	64.8	72.6	67.6	65.8	60.5	66.3	67.2
	Fine-grained Model	**65.1**	71.5	68.7	**70.8**	65.1	**68.2**	**68.6**
16	Render for CNN [21]	39.4	63.3	46.5	45.1	N/A	48.6	49.7
	ResNet18 CS [19], [14]	45.4	65.8	48.8	44.8	38.4	48.6	49.5
	PoseContrast [23]	45.6	66.9	55.6	**50.2**	43.0	52.3	53.9
	Multi-Task [9]	42.5	68.0	57.9	42.8	**50.0**	52.2	53.5
	Coarse-grained Model	47.9	66.9	**58.5**	46.0	43.0	52.5	54.9
	Fine-grained Model	**48.7**	**68.7**	58.0	44.8	45.4	**53.1**	**55.0**
24	Render for CNN [21]	34.9	55.9	40.8	36.0	N/A	41.9	42.6
	ResNet18 CS [19], [14]	33.2	**60.9**	45.9	36.6	32.6	41.8	43.5
	PoseContrast [23]	37.8	60.1	50.0	37.2	37.2	44.5	46.4
	Multi-Task [9]	32.7	56.2	49.3	36.3	**41.9**	43.3	44.6
	Coarse-grained Model	38.3	59.4	**50.7**	33.0	40.7	44.4	46.1
	Fine-grained Model	**40.6**	60.5	49.2	**41.6**	40.7	**46.5**	**47.5**

(b) Pascal3D+.

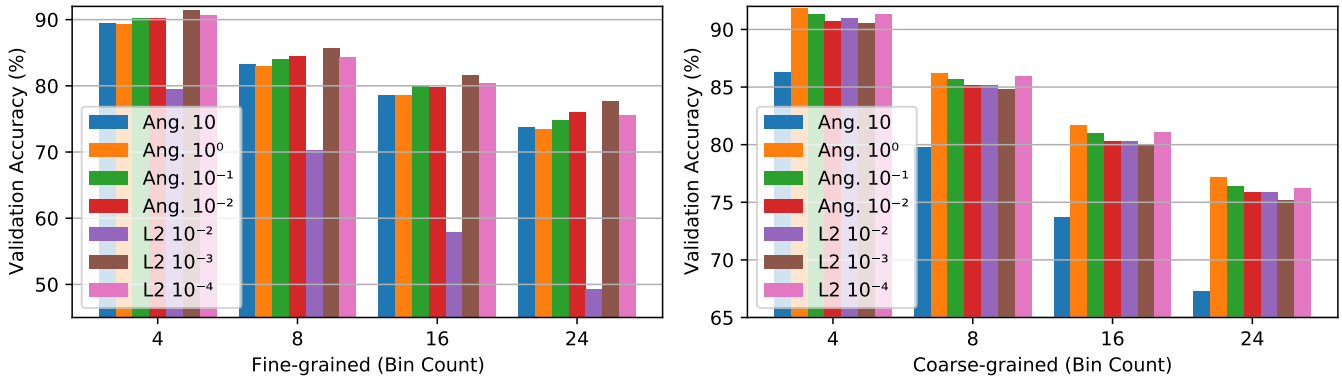


Fig. 7. Performance on the validation dataset varying the distance and λ value used for the Siamese training loss.

A. Implementation Details

Training settings We use convolutional backbones, with different size and structure, to evaluate the performance of our proposed models: MobileNetV2 [30], ResNet50 [20] and DenseNet [31]. We fine-tune all the layers for each backbone, starting from pre-trained Imagenet [29] weights. The models are trained using Adam [32] as optimizer with learning rate 10^{-3} and weight decay set to 10^{-4}. During training, the learning rate is reduced by a factor 0.1 when no improvement on the overall accuracy on the validation set for each discretization level is detected for three epochs. Each training is performed setting a maximum number of epochs to 100 with early stopping when the learning rate reaches 10^{-5} and the validation set accuracy stops to improve.

Image Pre-Processing The crops, extracted from the full images, were resized to 224×224 pixels and were normalized by subtracting the mean and dividing by the standard deviation of the Imagenet dataset, we call this approach *SquareResize*. We note that resizing all images to a square can distort the content because the aspect ratio is not preserved, hence we propose a different approach that we call *KeepRatio*: the images are resized so that only the largest side is 224 pixels, then zero-padding is applied to center the image in the target 224×224 square, finally the images are normalized as for *SquareResize*. In the ablation study (see Section V-C), we will provide a detailed comparison between the two different data pre-processing. As an additional data augmentation step, we apply random horizontal flipping with probability $50\,\%$ to all the training images and their respective coordinates, except when the Siamese approach (see Section III-D) is employed.

Siamese Hyperparameters We evaluated different λ values for the Siamese training loss (Equation 1). Since the distance metrics we evaluate have different scales, we perform a grid-search over two different search ranges: $\lambda \in \{10^k\}$, $k \in \{-2, -1, 0, 1\}$ for the angular distance and $\lambda \in \{10^k\}$, $k \in \{-4, -3, -2\}$ for the ℓ_2 distance. In our tests (Fig. 7) we verified that both our proposed models (i.e., *coarse-grained*

and *fine-grained*) are pretty robust with respect to these two parameters, except for the ℓ_2 distance with $\lambda = 10^{-2}$ that involves a significant performance drop when used to train the *fine-grained* model. From this analysis we fix the distance type and λ to the values gaining the best performance on the validation set, i.e., the angular distance with $\lambda = 10^{0}$ for the *coarse-grained* model and the ℓ_2 distance with $\lambda = 10^{-3}$ for the *fine-grained* one, which gained the best performance on the validation set. The best hyperparameters are slightly different from the ones reported in our original work [9], where the grid search is performed fixing the data pre-processing to *SquareResize*. Here we fix the data pre-processing to *KeepRatio* since it provides better performance.

Viewpoint Crop Settings Our proposed data augmentation (Section III-E) is intended to improve the viewpoint estimation performance on the closer vehicle, thus we apply it to vehicles with a reasonable distance from the on-board camera, namely not too near, since probably the vehicle is already truncated, and not too far. For Pascal3D+ the *viewpoint crop* is applied to all the available images, since the majority of vehicle are centered and occupy the entire foreground of the image. Instead, for nuScenes we use as proxy for the distance from the camera the bounding box area. For each vehicle class, we sort by increasing area the bounding boxes and we empirically verify that those between the 40$^{\text{th}}$ and the 80$^{\text{th}}$ percentile are at the right distance from the camera. We use the corresponding areas as thresholds for each vehicle class to select bounding boxes suitable for *viewpoint crop* augmentation.

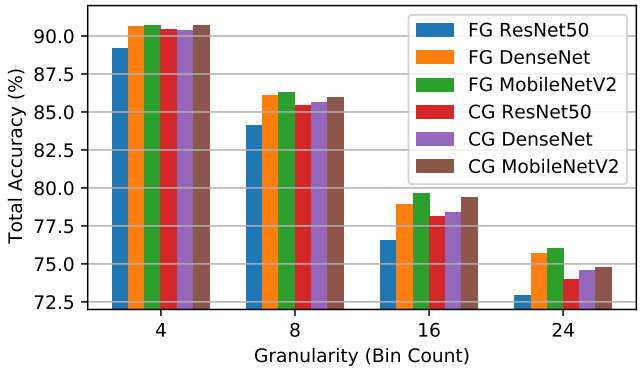


Fig. 8. Performance of different backbones for the fine-grained (FG) and the coarse-grained (CG) model.

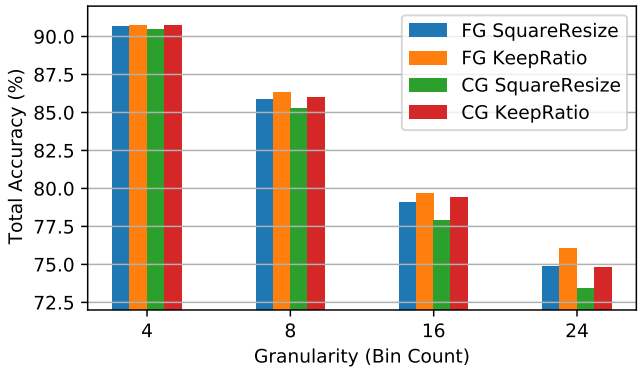


Fig. 9. Performance of the *SquareResize* and *KeepRatio* pre-processing methods for the fine-grained (FG) and the coarse-grained (CG) model.

B. Comparison with the State-of-the-Art

We compare our models performance on the Pascal3D+ and the nuScenes datasets with the architecture defined by Su et al. [21], by Tulsiani et al. [14] and by Xiao et al. [23]. In order to compare our model with state-of-the-art approaches, we use the provided weights when available (e.g. [21] pre-trained on Pascal3D+[1]), otherwise we re-train the model in order to predict only the azimuth angle keeping the original implementation training setting and using our performance evaluation. For the model defined by Tulsiani et al. [14], we change the backbone from VGG16 [18] to ResNet18, since it has been proved more effective by Zhou et al. [19].

For both the *coarse-grained* and the *fine-grained* model we use MobileNetV2 as a backbone, since it provides the best results when compared to ResNet or DenseNet models (see Section V-C) and, due to its reduced number of parameters and careful design, is one of the fastest CNNs.

While training on the nuScenes dataset, we augment the training set with Pascal3D+ images since in our tests we verified that it helps to better generalize, while when training is performed on Pascal3D+ we use only images from the original challenge to perform a fair comparison with the state-of-the art approaches. All our models (i.e., the *coarse-grained* and the *fine-grained* models) are trained using the Siamese approach, *viewpoint crop* data augmentation and the CoordConv layer, to add the information of the original position of the vehicle in the source road view image.

In Table II, we report the results on both datasets. The *fine-grained* model is the top performer among state-of-the-art approaches and further improves the results obtained by the previously proposed *multi-task* model [9], where the viewpoint crop, the *KeepRatio* pre-processing and the CoordConv layer were not employed. Furthermore, our models show a large increase in performance with respect to the state-of-the-art on nuScenes. It should be noted that the CoordConv layer impact is more prominent the finer the quantization.

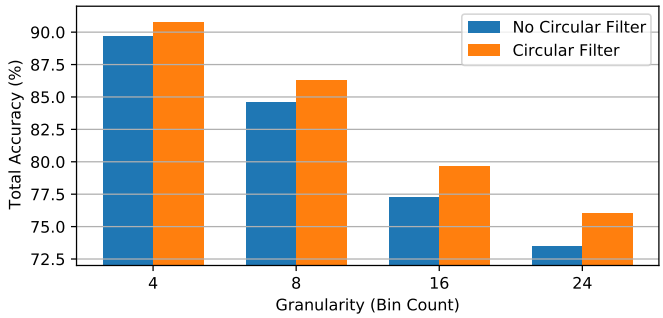
It is worth noting that different types of vehicles pose different difficulty levels in correctly estimating the viewpoint as can be seen in Table II and it is a known limitation of viewpoint estimation [12]. This is especially true for bicycles and motorbikes, for which is challenging to recognize the front from the rear, especially if the handlebar is occluded.

C. Ablation Study

In this section we discuss the impact on the performance of our models of backbone size, data pre-processing, and the proposed design decisions presented in Section III.

Backbone Size In Fig. 8 we show the overall total accuracy of the *coarse-grained* model and the *fine-grained* one, varying

[1]This model was trained without the *truck* class as it is not part of the official Pascal3D+ dataset.



(a) Performance impact in terms of accuracy.

(b) Distribution of errors granularity at 24 bins, lower is better.

Fig. 10. Performance impact of the mean circular filter, with window size 15 and stride 1 for the fine-grained model.

TABLE III
ABLATION STUDY ON VIEWPOINT CROP, COORDCONV, AND SIAMESE LOSS. CLASS AVERAGE AND OVERALL ACCURACY ARE REPORTED

	Viewpoint Crop	CoordConv	Siamese	Avg	Total
Coarse-grained (24 bin)	✓	✓	✓	**59.41**	**74.77**
	✗	✓	✓	59.30	74.09
	✓	✗	✓	51.18	65.69
	✓	✓	✗	55.88	73.08
Fine-grained (24 bin)	✓	✓	✓	**60.70**	**76.05**
	✗	✓	✓	**60.70**	75.82
	✓	✗	✓	51.26	66.49
	✓	✓	✗	55.18	72.59

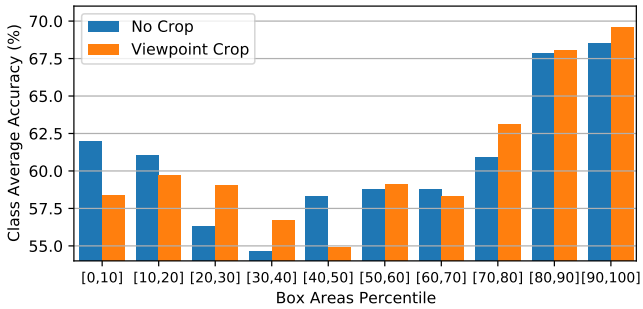


Fig. 11. Viewpoint crop effect on the nuScenes test set. A bar pair in the plot represents the 24 bins average accuracy on the classes considering only the box areas included in the percentile range of the bin, percentile values are computed for each vehicle class. This setup ensures that each bin contains one tenth of the whole dataset while preserving the class ratios (Table I).

the convolutional backbone. Our results show that, for each discretization level the performance drops when more complex models are used, which seems counter intuitive. We conjecture that, for this task, the reduced number of parameters helps to prevent overfitting, an hypothesis supported by the increasing accuracy when the size of the backbone decreases from the larger ResNet50 (24.2 M parameters) to DenseNet(7.3 M) and finally to the chosen MobileNetV2 (2.7 M).

Data Pre-Processing In Fig. 9 we compare the impact of the data pre-processing on our models. The results shows that preserving the original aspect ratio using *KeepRatio* provides a small but consistent improvement in accuracy compared to *SquareResize*, especially on finer discretization levels.

Smoothing Circular Filter for Fine-grained Model In Fig. 10a we compare the model accuracy of the *fine-grained* model with and without the circular mean filter before the softmax activation function. Our results show that our intuition about smoothing the logits and the analysis provided in [13] is correct, providing a significant boost in accuracy over the model trained without the smoothing by reducing the errors in the nearby bins and in those corresponding to a viewpoint flip, i.e., an error close to $180°$ (see Fig. 10b).

Viewpoint Crop Data Augmentation In Table III, we evaluate the effect of the proposed cropping strategy for data augmentation over our models. The results show that using of *viewpoint crop* provides a noticeable, although small, improvement in the overall models accuracy. Since we empirically found that the larger the area of a box the more likely it is to contain a cropped vehicle, we also evaluate the accuracy for different box areas to gain more insights on the benefits brought in by *viewpoint crop*. In particular, we compute the average accuracy per vehicle classes on the nuScenes test set when we consider only a specific range of box areas. Results of this study for the *fine-grained* model are reported in Fig. 11; similar trends can be seen also for the *coarse-grained* model.

It clearly appears that viewpoint crop provides significantly more accurate predictions on larger boxes, likely to correspond to nearby and possible occluded vehicles. Improving model performance on nearby vehicles is in practice very relevant for a broad range of applications, e.g. safety critical applications like car crashes identification, which usually require the highest performance on the closest vehicles.

Vehicle Coordinates In Table III we show the effect of the use of vehicle position in the original image as an additional input. Our experiments designate this as the most significant factor in improving the accuracy of the network, providing a remarkable increase of $10\,\%$ in terms of total accuracy.

We note that the use of a CoordConv layer is also a limitation since the network will learn to expect vehicles with specific coordinates to have a certain aspect, meaning that a significant change in camera positioning between training and test images will result in a significant performance drop. When testing our models trained with nuScenes on Pascal3D+ we obtained a total accuracy of $35.71\,\%$ for the *fine-grained* model at 24 bins, a drop of about $10\,\%$ when compared to

TABLE IV
MEMORY REQUIREMENTS AND TIMING.

	CPU Time (ms)	**GPU Time** (ms)	**# Parameters**
Render for CNN [21]	74.51 ± 7.71	5.43 ± 0.03	64.2M
ResNet18 CS [19], [14]	192.31 ± 14.61	10.82 ± 0.03	11.2M
PoseContrast [23]	662.67 ± 28.85	33.71 ± 0.05	25.8M
Coarse-grained Model Fine-grained Model	277.10 ± 10.91	12.05 ± 0.02	2.7M

the models trained only on Pascal3D+; moreover by training without the CoordConv layer on nuScenes the performance on Pascal3D+ do not drop as much, achieving $44.68\,\%$, thus highlighting the limitation of using vehicle coordinates.

Siamese Network In Table III we show the effects of training using a Siamese approach. Both models benefits from this training approach and exhibit a significant increase in average accuracy, with the *fine-grained* one considerably increasing also the total accuracy.

D. Memory Requirements & Timing

In Table IV we provide the inference time and the number of trainable parameters of our models, which we note are the same given the shared architecture. Since in nuScenes [10] there are on average 20 vehicles per image, the timing are measured fixing the batch size to 20 vehicle crop images and averaging the time over 300 executions running on an AMD Ryzen 7 3700X 8-Core CPU and a GeForce RTX 2070 GPU. Our models outperform in terms of memory requirements other state-of-the-art approaches and, thanks to the small memory footprint, are suitable to be deployed on the edge.

Of note is that a smaller number of parameters does not lead to faster inference times, which is also influenced by how the parameters are used, e.g. in sequential operations generally slower instead of faster parallel processes or convolutions. This is the case for Render for CNN [21] and ResNet18 [19], [14] when compared to our models; this behavior has also been reported in benchmarks for different models [33].

VI. CONCLUSION

In this paper we presented two lightweight deep learning models able to predict the viewpoint of vehicles from monocular images, taken, e.g., from a camera mounted on a vehicle driving on the road. We have shown that the *fine-grained* model improves the *coarse-grained* one, developed in our original work [9], and outperforms state-of-the-art results. We achieved this result by means of several contributions.

First, we show that applying smoothing techniques to the network output can noticeably improve vehicle viewpoint estimation performance. Specifically, applying a circular mean filter before the network output, as done for the *fine-grained*, provides better results than summing the network logits, i.e. the approach of the *coarse-grained* model. Additionally, we show that adding geometrical constraints to the training loss by means of a Siamese network further improves the results.

We also introduce the task-specific data augmentation technique *viewpoint crop* with the aim of improving its performance on truncated vehicles at test time, which is an important trait in practical deployments of a viewpoint estimation algorithm, e.g. to estimate azimuth of involved vehicles in the case of car crashes.

Finally, we point out an ambiguity of vehicle viewpoint prediction from monocular images and propose an effective solution to it in the form of CoordConv layers [27]. Both the *fine-grained* and *coarse-grained* model considerably improved their performance on the nuScenes dataset.

Experimental results on the nuScenes dataset show that a small convolutional backbone, like MobileNetV2 [30], is able to obtain results which are comparable to or even better than the ones obtained by much more complex backbones. Thanks to the speed and the size of the backbone, our model is suitable to be deployed on edge devices. We obtain state-of-the-art results also on Pascal3D+, although the absolute performance is lower than on nuScenes, likely because the fixed point of view of the camera with respect to the road eases vehicle viewpoint estimation.

Possible directions for future research involve the use of videos to train and test our proposed models. We believe that temporal information from videos can improve the quality of vehicle viewpoint estimation by imposing constraints on the rigid object motion through time. Another path is to further evaluate the limitation to a specific camera and positioning imposed by the use of a CoordConv layer and how to compensate the differences between the train and test setup.

ACKNOWLEDGMENT

The authors would like to thank professor Fabio Schoen for the useful suggestions and the support to carry out this work.

REFERENCES

[1] M. Ozuysal, V. Lepetit, and P. Fua, "Pose estimation for category specific multiview object localization," in *CVPR*, 2009, pp. 778–785.
[2] Q. Li, H. Lu, X. Liu, X. Huang, C. Song, S. Huang, and J. Huang, "Optimized 3d street scene reconstruction from driving recorder images," *Remote Sensing*, vol. 7, pp. 9091–9121, 07 2015.
[3] N. Cornelis, B. Leibe, K. Cornelis, and L. V. Gool, "3D Urban Scene Modeling Integrating Recognition and Reconstruction," *International Journal of Computer Vision*, vol. 78, pp. 121–141, 2007.
[4] H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, J. Schneider, D. Bradley, and N. Djuric, "Deep kinematic models for kinematically feasible vehicle trajectory predictions," in *ICRA*, 2020, pp. 10 563–10 569.
[5] M. Simoncini, D. C. de Andrade, S. Salti, L. Taccari, F. Schoen, and F. Sambo, "Two-stream neural architecture for unsafe maneuvers classification from dashcam videos and gps/imu sensors," in *ITSC*, 2020, pp. 1–6.
[6] Z. Liu, Z. Wu, and R. Toth, "Smoke: Single-stage monocular 3d object detection via keypoint estimation," in *CVPRW*, 2020, pp. 4289–4298.
[7] A. Simonelli, S. Rota Bulo, L. Porzi, M. Lopez Antequera, and P. Kontschieder, "Disentangling monocular 3d object detection: From single to multi-class recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.
[8] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 2019.
[9] S. Magistri, F. Sambo, F. Schoen, D. C. d. Andrade, M. Simoncini, S. Caprasecca, L. Kubin, L. Bravi, and L. Taccari, "A lightweight deep learning model for vehicle viewpoint estimation from dashcam images," in *ITSC*, 2020, pp. 1–6.
[10] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.

[11] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond PASCAL: A benchmark for 3D object detection in the wild," in *WACV*, vol. 00, March 2014, pp. 75–82.

[12] G. Divon and A. Tal, "Viewpoint estimation—insights and model," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 265–281.

[13] C. Redondo-Cabrera, R. López-Sastre, Y. Xiang, T. Tuytelaars, and S. Savarese, "Pose estimation errors, the ultimate diagnosis," in *ECCV*, 2016.

[14] S. Tulsiani and J. Malik, "Viewpoints and keypoints," in *CVPR*, 2015, pp. 1510–1519.

[15] F. Massa, M. Aubry, and R. Marlet, "Convolutional neural networks for joint object detection and pose estimation: A comparative study," *CoRR*, vol. abs/1412.7190, 2014. [Online]. Available: http://arxiv.org/abs/1412.7190

[16] R. M. Francisco Massa and M. Aubry, "Crafting a multi-task cnn for viewpoint estimation," in *BMVC*, E. R. H. Richard C. Wilson and W. A. P. Smith, Eds. BMVA Press, September 2016, pp. 91.1–91.12.

[17] A. Ghodrati, M. Pedersoli, and T. Tuytelaars, "Is 2d information enough for viewpoint estimation?" in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2015.

[19] X. Zhou, A. Karpur, L. Luo, and Q. Huang, "Starmap for category-agnostic keypoint and viewpoint estimation," in *ECCV*, September 2018.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, June 2016, pp. 770–778.

[21] H. Su, C. R. Qi, Y. Li, and L. J. Guibas, "Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views," in *ICCV*, December 2015.

[22] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *NIPS*, 1993, p. 737–744.

[23] Y. Xiao, Y. Du, and R. Marlet, "Posecontrast: Class-agnostic object viewpoint estimation in the wild with pose-aware contrastive learning," in *3DV*, 2021.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[25] C. Beckham and C. Pal, "Unimodal probability distributions for deep ordinal classification," in *ICML*, D. Precup and Y. W. Teh, Eds., vol. 70, Aug 2017, pp. 411–419.

[26] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká, "3d bounding box estimation using deep learning and geometry," in *CVPR*, 2017, pp. 5632–5640.

[27] R. Liu, J. Lehman, P. Molino, F. Petroski Such, E. Frank, A. Sergeev, and J. Yosinski, "An intriguing failing of convolutional neural networks and the coordconv solution," in *NIPS*, vol. 31, 2018.

[28] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR*, 2009.

[30] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," *CVPR*, pp. 4510–4520, 2018.

[31] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2015.

[33] S. Bianco, R. Cadène, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018.

Simone Magistri gained his master's degree in Computer Engineering at the University of Florence, Italy, in 2020. He is now PhD student of Smart Computing at Florence University, department of Information Engineering. His research interests include machine learning, computer vision, deep learning and optimization techniques.

Marco Boschi received the MSc degree in computer engineering from the University of Bologna, Italy, in 2020. He is currently working at Verizon Connect, Florence, Italy as Data Science Engineer. His research interests include scene understanding, machine learning, computer vision and their applications.

Francesco Sambo holds a PhD in bioinformatics and artificial intelligence from the university of Padova. He is currently Chief Data Scientist at Verizon Connect. His research interests include road scene understanding and predictive maintenance. He is author or coauthor of more than 50 scientific papers and 10 patents.

Douglas Coimbra de Andrade graduated mechanical aeronautical engineer in 2005 and received his D. Sc. in the field of Aerospace Systems and Mechatronics in 2017, both from Instituto Tecnologico de Aeronautica, Brazil. His research interests include AI applied to computer vision, video analytics and HPC.

Matteo Simoncini received the MSc degree in computer engineering from the University of Florence, Italy, in 2016. He is currently working toward an industrial Ph.D. degree at Verizon Connect Research, Florence, Italy and the University of Florence, Italy. His research interests include intelligent transportation systems, machine learning, computer vision and their applications.

Luca Kubin received the MSc degree in communication engineering from the University of Parma in 2015. His research interests include deep learning, computer vision, and digital signal processing.

Leonardo Taccari received the Ph.D. in Operations Research from Politecnico di Milano in 2015. He is currently lead scientist at Verizon Connect. He is author or coauthor of more than 20 scientific articles and 10 patents. His research interests include machine learning and mathematical optimization.

Luca De Luigi received the MSc degree in computer engineering in 2015 from the University of Bologna, where is currently PhD student at the Computer Vision Laboratory (CVLab). His research focuses on deep learning for computer vision problems, especially dealing with 3D geometry and implicit neural representations.

Samuele Salti is currently associate professor at the Department of Computer Science and Engineering (DISI) of the University of Bologna, Italy. His main research interests are computer vision and machine learning. Dr. Salti has co-authored 51 publications and 8 patents. He received two best paper awards runner-up (3DIMPVT 2011, 3DV 2021). In 2020, he co-founded the start-up eyecan.ai.