IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 7, 2020, accepted July 13, 2020, date of publication July 20, 2020, date of current version July 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3010307

Simple Baseline for Vehicle Pose Estimation: Experimental Validation

HÉCTOR CORRALES SÁNCHEZ, (Graduate Student Member, IEEE),
ANTONIO HERNÁNDEZ MARTÍNEZ,
RUBÉN IZQUIERDO GONZALO, (Graduate Student Member, IEEE),
NOELIA HERNÁNDEZ PARRA, IGNACIO PARRA ALONSO,
AND DAVID FERNÁNDEZ-LLORCA, (Senior Member, IEEE)
Computer Engineering Department, University of Alcalá, 28801 Alcalá de Henares, Spain

Corresponding author: Héctor Corrales Sánchez (hector.corrales@uah.es)

This work was supported in part by the Community Region of Madrid under Grant S2018/EMT-4362 SEGVAUTO 4.0-CM, in part by the Spanish Ministry of Science and Innovation under Grant DPI2017-90035-R, in part by H2020 through the BRAVE Project under Contract 723021, in part by the Acción Complementaria Internacional, MINECO, under Grant PCIN-2017-086, and in part by the Electronic Component Systems for European Leadership Joint Undertaking under Grant 737469 (AutoDrive Project).

ABSTRACT Significant progress on human and vehicle pose estimation has been achieved in recent years. The performance of these methods has evolved from poor to remarkable in just a couple of years. This improvement has been obtained from increasingly complex architectures. In this paper, we explore the applicability of simple baseline methods by adding a few deconvolutional layers on a backbone network to estimate heat maps that correspond to the vehicle keypoints. This approach has been proven to be very effective for human pose estimation. The results are analyzed on the PASCAL3D+ dataset, achieving state-of-the-art results. In addition, a set of experiments has been conducted to study current shortcomings in vehicle keypoints labelling, which adversely affect performance. A new strategy for defining vehicle keypoints is presented and validated with our customized dataset with extended keypoints.

INDEX TERMS Vehicle pose estimation, vehicle keypoints detection, CNNs, heat maps, human pose estimation, experimental validation.

I. INTRODUCTION

Deep learning and the huge evolution that Convolutional Neural Networks (CNNs) have undergone in recent years have completely changed computer vision. Many tasks, such as image classification, segmentation, or object detection, have been solved or greatly advanced. Among them, pose estimation is a hot topic with increased attention in recent years, being human pose estimation the one that has monopolised the vast majority of efforts and advances. Pose estimation is a complex and challenging task, especially human pose, due to the different possible poses and body's flexibility. But in spite of this, as Xiao *et al.* presented in [1], the existing human pose benchmarks have become saturated (MPII [2]), or huge advances have been made (COCO [3]). Along with the improvement in the accuracy of the models, their complexity has also increased. The comparison of models obtaining similar results but with very different approaches is virtually impossible. Because of this, Xiao *et al.* [1] raised the question of how good a simple method could be and presented a baseline model that achieved the state-of-the-art at COCO.

Under the shadow of human pose estimation, vehicle pose estimation has also drawn some attention. First of all, we want to clarify that, traditionally, when talking about human pose estimation, the "pose" is characterized by the 2D keypoints detected on the image plane. The reason for this is that the human body has a flexible and variable structure and can adopt different "poses". However, when talking about vehicles, there is no consensus, and it can relate to both the 2D or 3D pose as vehicles are rigid structures. We want to clarify that in this article, when we use vehicle pose, we refer to the 2D pose, the same as in humans.

Vehicle pose is an important task, with a huge variety of applications in multiple domains like surveillance or autonomous vehicles. Multiple works address the problem of

The associate editor coordinating the review of this manuscript and approving it for publication was Dalin Zhang.

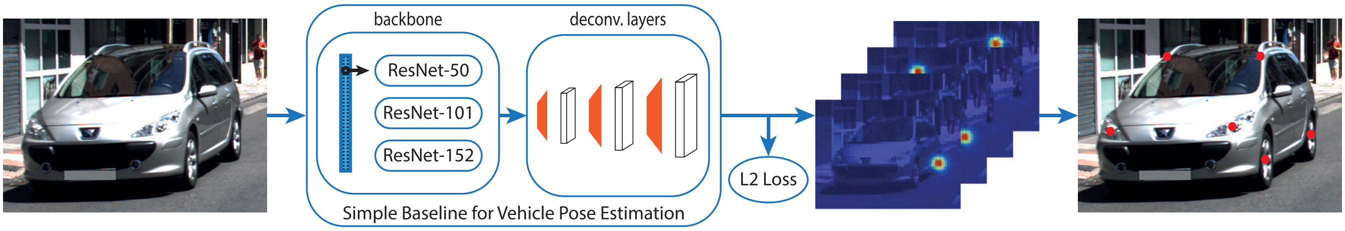

FIGURE 1. Illustration of the simple baseline network architecture proposed for vehicle keypoint estimation.

vehicle pose estimation. Among these works, we can distinguish two groups. On the one hand, those who make use of keypoints [4]–[7], either because they estimate the 2D pose or because they use them as an intermediate step to obtain the 3D pose. On the other hand, those who do not rely on keypoints. Of the latter, we can highlight works such as that of Zia *et al.* [8], in which they proposed a viewpoint-invariant method for 3D reconstruction using shape and occlusion modelling and a common scene geometry (ground plane) to fit these shapes. In [9]. Juránek *et al.* presented an object detector coupled with pose estimation using shared image features. Following this approach, Wang *et al.* [10] modified Faster R-CNN [11] to regress 3D pose parameters using only the 2D appearance. Going back to the first group, the one that uses keypoints, at first glance, vehicle keypoint estimation should be easier when compared to human as the rigid structure of vehicles limits the possible poses, and occlusion and overlapping on vehicles is less complex than on humans. However, there are other specific difficulties. For example, the impact of the camera perspective is stronger on cars than on people, and the intra-class variability is much higher due to the large number of different car makes, models, sizes, and types. Thus, the relative 3D position between keypoints, and their corresponding 2D positions projected on the image plane, can considerably vary depending on the vehicle. Another important difficulty is the definition of the most representative keypoints. Whereas the human's body shape, almost directly, suggests what the location of the keypoints should be, in the case of vehicles, the best location of the keypoints remains unsolved. As an example, we can see the current lack of consensus when labelling vehicle keypoints in the available datasets [4], [12]–[15]. This makes it very difficult to compare methods that use different datasets and, thus, different keypoints. Additionally, as we just said, unlike with humans, the best location for the keypoints is not obvious, and, to the best of our knowledge, there are no studies analysing the consistency of the labelling process and the suitability of each keypoint.

The high intra-class variability of vehicles can be alleviated by means of fine-grained vehicle classification, that is classifying car make [16], model, and year [17]. This is a considerably complex task, as the system has to learn to distinguish the subtle differences between the different but still very similar car models. As suggested in [18], in a kind of virtuous circle, the use of vehicle keypoints to obtain enriched information as pose, type of vehicle or the location of the most relevant parts, could help in this complex and challenging task, and, at the same time, the availability of car type, make, model and year, could improve the performance of the vehicle pose estimation task.

The use of keypoints has been widely explored and accepted for both human and vehicle pose estimation. In the human case, applications beyond pose estimation like gender classification [19], violence recognition [20] or more specific "sub-pose" like hand [21] or face [22] have been explored. In the vehicle case, the importance of obtaining the keypoints to represent its pose has gained attention due to the increasing number of potential applications. Besides the aforementioned structural support to improve fine-grained classification, they can be used to enhance instance segmentation like in [23]. Vehicle keypoints can also be used as anchors to retrieve 3D keypoints and structure through the use of CAD models and camera parameters as in [4]–[7]. They can help to improve traffic surveillance applications [24], including vehicle re-identification [25], when a vehicle has to be identified on multiple images from different cameras, and license plate recognition is not possible (usually because the camera resolution is very poor or the license plate is not visible). For example, Wang *et al.* [25] proposed the use of keypoints to localize relevant parts of the vehicles and use them as an attention mechanism, obtaining state-of-the-art results. Another interesting application that could benefit from the use of keypoints is vision-based accurate vehicle speed detection [26]. These systems are usually based on detecting and tracking the license plates, of which the actual size is known. Vehicle keypoints estimation methods can be easily adapted to accurately detect the four corners of the license plate [27].

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

- We study the applicability of a simple baseline approach to deal with vehicle pose estimation, and evaluate its performance compare with the state-of-the-art, following on the ideas proposed by [1] for human pose (see Fig. 1).
- Extensive experimental validation is carried out using one of the most advanced datasets so far, PASCAL3D+, including data augmentation techniques, and improving state-of-the-art results.
- The most important shortcomings of the current datasets related to the definition of the vehicle keypoints are

discussed and experimentally verified and, a new customized dataset with corrected keypoints is proposed and evaluated.

The remainder of the paper is organized as follows. Section II briefly summarizes the state-of-the-art. The description of the metrics used for evaluation, the datasets, the data augmentation techniques, and the proposed architecture are presented in Section III. The extensive experimental evaluation is provided in Section IV. Conclusions and future works are finally discussed in Section V.

II. RELATED WORK

Keypoint prediction methods can be divided in two different approaches:

- **Top-down methods**. Top-down methods [1], [13], [28]–[32] first detect all instances in the given image with an external object detector like Faster R-CNN [11], Feature Pyramid Networks (FPNs) [33] or YOLO [34] and then predict keypoints for each one of them. This approach benefits from the advances in instance detectors.
- **Bottom-up methods**. Bottom-up methods [35]–[39] detect all the present keypoints in a given image and then reconstruct each instance associating the different keypoints. This approach has been mainly studied for human pose obtaining very good results.

These two approaches have their pros and cons. Top-down methods are easier to train and more reliable as each instance in the image is processed individually but, while the throughput of bottom-up methods is more or less invariable with the number of instances in the image, top-down methods suffer with crowded scenes in terms of speed and struggle when the bounding boxes overlap. Bottom-up methods have proven that their performance is far better than the one achieved with top-down methods in highly crowded scenarios, with a high amount of occluded or overlapped instances.

Even though most of the methods have been developed for human pose estimation, generic object pose estimation and vehicle pose estimation have also received some attention. One of the first approaches to use convolutional layers is the one from Long *et al.* [40]. In it, they used five convolutional layers to extract features and feed them to a linear Support Vector Machine (SVM) for each keypoint. After them, the first approach to use a full CNN is the one from Tulsiani and Malik [28]. They calculated a likelihood map for each keypoint by combining response maps from two different scales and a viewpoint prior. In [5], Murthy *et al.* proposed a fully convolutional CNN regressor to predict the keypoints and, after that, refine them with a set of finetuning networks. Another interesting approach is the one from Li *et al.* in [12], [13]. They used intermediate shape concepts like viewpoint, keypoint visibility, and keypoints to supervise the training process.

Of particular interest are the stacked hourglass networks [29]. These networks are made up of residual convolutional modules packed in blocks with symmetric bottom-up/top-down capacity (from high to low resolutions and from low to high resolutions again) that seek to capture information at every scale. To do so, they use a single pipeline with skip layers that connect each branching with their symmetrical at the other end of the module. These modules are then stacked, and an intermediate supervision loss is applied at the end of each one. Multiple authors have used these networks, mainly for human pose estimation. In [41], Wang *et al.* proposed the use of a densely connected convolutional module instead of the residual one obtaining comparable performance on MPII while reducing the number of parameters and complexity of the network. In [42], Radwan *et al.* proposed the use of a Generative Adversarial Network (GAN) [43] scheme along with a keypoint hierarchy. Both generator and discriminator are hourglass based, and their results suggest comparable performance with other state-of-the-art methods. In [44], Wang *et al.* also used a GAN scheme. They used hourglass networks as backbone in both generator and discriminator and self-attention mechanism outperforming state-of-the-art methods on MPII.

Although the stacked hourglass was initially proposed for human pose estimation, multiple works like [6], [45], [46] and [30] have employed them for vehicle pose estimation. One of the first authors, if not the first, to consider adapting stacked hourglass to other problems was Pavlakos *et al.* [45], using a two hourglasses network with intermediate supervision for keypoint localisation. In [46], Murthy *et al.* proposed a Conditional Random Field (CRF)-Style loss function at the end of each hourglass unit to not only precisely localise each keypoint, but also enforce inter-keypoint distances constraints. In [6], Ding *et al.* built a four-layer modified hourglass network and also used intermediate supervision. In [30], Reddy *et al.* used a stacked hourglass network as initial visible keypoint detector. After this, they used an encoder-decoder scheme to predict the occluded keypoints exploiting multiple views of the object.

In the same way as with stacked hourglass networks, other human pose estimation methods have been used to localise vehicle keypoints. In [7] Song *et al.* used Convolutional Pose Machines (CPMs) [47] as its vehicle keypoint detector. Another interesting approach is the one from Nibali *et al.* in [48]. They proposed the use of a Differentiable Spatial to Numeric Transform (DSNT) along with dilated convolutions [49] to adapt fully convolutional networks to coordinate regression obtaining promising results on MPII. Later, in [27], Llorca *et al.* used this approach to detect license plate corners in an accurate and efficient way.

III. EXPERIMENTAL SETUP

As previously said, keypoint prediction is a widely explored task. The most common approach is to first detect each object in a given image and then get their keypoints individually. Nonetheless, various authors evaluated a more global approach that detects all the keypoints in a given image and then groups them with great results [35]–[39]. In our case, we are going to use the top-down method approach proposed by Xiao *et al.* [1] as its easier to train, simpler, and has proven to have state-of-the-art performance.

A. METRICS

In order to evaluate the models the metrics proposed by [50] have been used:

- **Percentage of Correct Keypoints (PCK)**. PCK measures the number of labelled keypoints that are correctly predicted. A predicted keypoint is correct if its distance to the given ground-truth keypoint is equal to or less than $\alpha * L$, with $L = max(h, w)$ (L is the bigger side of the object bounding box) and $0 < \alpha < 1$. We use $\alpha = 0.1$.
- **Average Precision of Keypoints (APK)**. As in PCK, a predicted keypoint is correct if its distance to the given ground-truth keypoint is equal to or less than $\alpha * L$. Each predicted keypoint has associated confidence and a threshold is used to calculate the area under the precision-recall curve. This evaluation penalises missed-detections and false positives. The way to compute APK can be seen in equation 1, being P_n and R_n the precision and recall at the n^{th} confidence threshold.

$$APK = \sum_n (R_n - R_{n-1}) * P_n \tag{1}$$

B. DATASETS

In the case of human pose estimation, there is a wide variety of datasets to choose from, being MPII and COCO, the most popular in recent years. Once again, in the case of vehicles, it is more complicated to find datasets suitable for the keypoint prediction task.

One of the most famous is the PASCAL3D+ dataset. Created by Xiang *et al.* [4], it has the 12 rigid categories of PASCAL VOC 2012 [51]. Focusing on cars, there are a total of 6,704 images (1,229 from PASCAL and 5,475 from ImageNet) and 7,791 instances (2,161 from PASCAL and 5,630 from ImageNet) with CAD models and a set of 12 keypoints: *the four wheels, windshield's and rear window's upper corners, headlights, and left/right side of the trunk*. The train/val split is approximately 50% with 621/608 images (1091/1070 instances) for the PASCAL subset and 2763/2712 images (2850/2780 instances) for the ImageNet subset.

Other interesting datasets are the ones created by Li *et al.* [12], [13]. Firstly, we have the Rendered Images (Car) dataset, a vast synthetic dataset of 600K car images. They picked a subset of car CAD models from ShapeNet [14] and annotated 36 3D keypoints. After this, they rendered each CAD model using random parameters for camera viewpoint, light source, and surface reflection. These rendered images are then overlayed over real backgrounds to prevent overfitting. Secondly, we can find the KITTI-3D dataset. This dataset consists of 2,040 images from KITTI [15], labelled with 2D keypoints that they use to test the models trained with the synthetic dataset.

Unfortunately, we have not been able to obtain access to the synthetic dataset, so our experiments take place only with PASCAL3D+. A sample of the images from PASCAL3D+ can be seen in Fig. 2, being the top row images from PASCAL and the bottom row from ImageNet.

C. DATA AUGMENTATION

An important part of any CNN training is data augmentation. Three different data augmentation approaches will be tested:

- No data augmentation at all.
- Mild data augmentation: 50% chance of horizontal flip, random rotation of up to ±30°/40°/50°, and random scaling of up to ±30%.
- Hard data augmentation: mild data augmentation and a randomly selected operation between salt-and-pepper noise, poisson noise, speckle noise, blurring, colour casting, and colour jittering.

D. ARCHITECTURE

As previously said, the architecture that we are going to use is the one proposed by Xiao *et al.* [1]. This architecture consists of an ImageNet pre-trained ResNet [52] backbone with three deconvolutional layers added to the end and a 1 × 1 convolutional layer to generate the output heatmaps. Each deconvolutional layer has 256 filters with a 4 × 4 kernel. The ground truth vehicle bounding boxes have a fixed aspect ratio of *width:height = 4:3* obtained by extending the original box. This is then cropped and resized. Two input sizes (256 × 192 and 384 × 288) and all three main ResNets (50, 101, and 152) are going to be tested. Adam is used as optimiser, MSE (Mean Squared Error) as the loss between predicted and target heatmaps, trained for 140 epochs, 0.9 momentum, initial learning rate of 0.001, and a reduction of 0.1 at epochs 90 and 120. The network structure is illustrated in Fig. 1. As the model proposed by [1] is called *Simple Baselines for Human Pose Estimation* we will refer to our adaptation as *Simple Baseline for Vehicle Pose Estimation* (SBVPE).

IV. RESULTS

A. DATA AUGMENTATION

As previously said, we have tried three different data augmentation approaches. In order to perform these experiments, we used the ResNet50 backbone with 256 × 192 input size and the 1229 images from PASCAL. The different results obtained can be seen in Table 1.

As expected, the use of data augmentation is a considerable improvement, practically getting the same PCK and APK results with the mild approach (±40°/50°) and the hard approach. Seeing that mild and hard approaches get the same results, we can conclude that only the operations that change the geometry of the images appear to have a positive effect on the training, making the extra operations of the hard approach ineffective. At this point, we discarded the use of the hard approach, as it has a higher computational cost, but it does

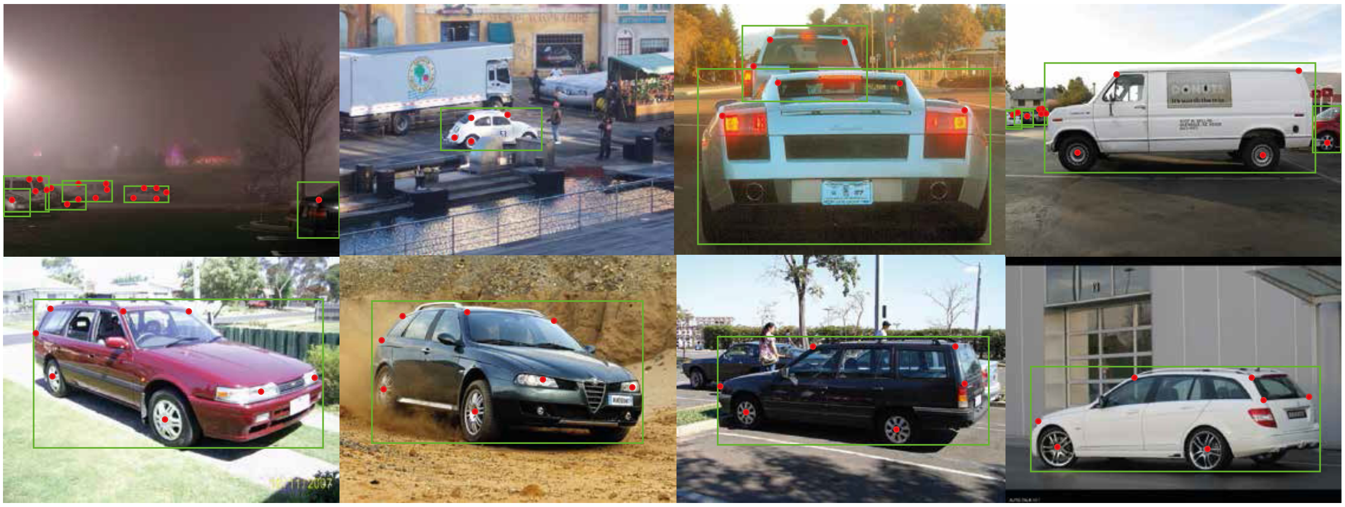

FIGURE 2. Examples of images from PASCAL3D+. In green the bounding boxes and in red the keypoints. The top row are images from PASCAL and the bottom row from ImageNet.

TABLE 1. PCK and APK with $\alpha = 0.1$ for different data augmentation strategies using the PASCAL images of PASCAL3D+.

Data Augment Strategy	PCK (%) $\alpha = 0.1$	APK (%) $\alpha = 0.1$
No data augmentation	64.09	26.77
Mild: Rot ±30°	70.42	33.66
Mild: Rot ±40°	**75.52**	**40.53**
Mild: Rot ±50°	75.5	40.43
Hard: Rot ±40°	75.41	40.2

TABLE 2. PCK and APK with $\alpha = 0.1$ for different backbones and Input sizes using the PASCAL images of PASCAL3D+.

Backbone	Input Size	PCK (%)	APK (%)	Train Time
ResNet-50	256x192	75.52	40.53	35m
ResNet-50	384x288	82.18	48.09	1h 3m
ResNet-101	256x192	75.76	41.76	46m
ResNet-101	384x288	82.75	49.8	1h 29m
ResNet-152	256x192	76.54	45.38	1h
ResNet-152	**384x288**	**83.17**	**50.69**	**1h 56m**

TABLE 3. PCK and APK with $\alpha = 0.1$ for the different subsets of PASCAL3D+. All runs with ResNet-152 backbone and input size of 384 × 288.

Train Data	Val Data	PCK (%)	APK (%)	Train Time
PASCAL (P)	PASCAL	83.17	50.69	1h 56m
ImageNet (IN)	ImageNet	98.98	74.51	4h 44m
P+IN	P+IN	97.12	72.38	6h 27m
ImageNet	PASCAL	76.26	45.28	4h 44m
P+IN	**PASCAL**	**88.49**	**55.09**	**6h 27m**

not bring any improvement. From now on, all runs have been made with the ±40° mild approach.

B. BACKBONE AND INPUT SIZE

Continuing with the experiments, we wanted to check the impact of using deeper ResNet backbones and increasing the input resolution from 256 × 198 to 384 × 288. The different configurations and their results can be seen in Table 2.

As expected, the use of a deeper backbone model has a positive impact on performance. A small increase in performance can be observed when switching from ResNet-50 to 101 and 152, passing from a PCK of 75.52% to 75.76% and 76.54%, respectively. The input image size has shown to have a critical impact, making it possible for the ResNet-50 backbone with the increased input size to outperform the ResNet-152 with standard input size with similar computational costs. Again, the best performance is for the deeper model, going from a PCK of 82.18% for the ResNet-50 model to 82.75% and 83.17% for the ResNet-101 and ResNet-152 models.

Taking into account the training times (all models trained on a 1080Ti NVIDIA GPU), and therefore, the computational cost, we can see that, as expected, the use of deeper models comes with a higher computational cost. However, it is the increased input size that has the greatest impact. While going from ResNet-50 to ResNet-101 costs 30% more, and another 30% more to go to ResNet-152, the change in input size has a cost of 80% more for ResNet-50 and 93% more for ResNet-101 and ResNet-152. From now on, all experiments will use the ResNet-152 backbone with a 384 × 288 input size.

C. PASCAL3D+: PASCAL AND ImageNet IMAGES

So far, the experiments have only been trained with the 1,229 PASCAL images from PASCAL3D+. Here we compare the impact of using the 5,475 ImageNet images and both subsets at the same time. The results can be seen in Table 3.

A considerable increase in performance, both in terms of PCK and APK, can be observed when using ImageNet and

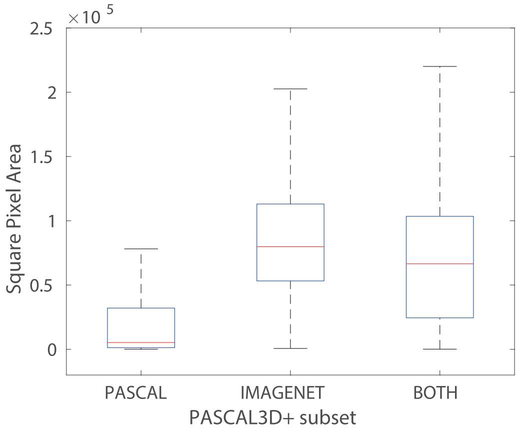

FIGURE 3. Instances area distribution for each subset of PASCAL3D+ without the outliers.

PASCAL+ImageNet subsets. This huge leap in performance (from 83.17% to 98.98% PCK) makes us wonder why. If we take a look at the subsets, we can see that while in PASCAL, there are 2,161 instances in 1,229 images, in ImageNet, there are 5,630 instances for 5,475 images. This makes us suspect that in PASCAL, the occlusions and overlaps are much more common than in ImageNet, making the PASCAL subset more complex.

Furthermore, if we take a look at the size of the objects, it can be seen how in ImageNet they have a considerably larger area than in PASCAL, accentuating, even more, the difference in complexity between these two subsets. A comparison of the objects area between subsets can be seen in Fig. 3.

These results made us strongly consider the possibility of overlearning. Therefore, we decided to test the PASCAL validation set with the models trained with ImageNet and PASCAL+ImageNet subsets. In the case of training with ImageNet, we can see, as expected, a drop in performance, which is almost certainly due to the difference in complexity between the two sets. On the other hand, in the case of training with the full PASCAL3D+ dataset, we can see a considerable improvement over training with PASCAL alone, with PCK rising from 83.17% to 88.49% and APK from 50.69% to 55.09%. This indicates that the use of ImageNet, together with PASCAL, has a positive effect and has improved the model's generalization capabilities improving the results.

In Table 4 we compare SBVPE results with previous approaches like [13], [28], [40], [46]. In these publications, it is not clear exactly which dataset is used. We believe that only the PASCAL subset is used. Nevertheless, we report results using the different subsets of PASCAL3D+.

If we compare SBVPE with previous approaches, we can see that we outperform them regardless of the subset used excepting the method introduced by Murphy *et al.* [46], which is only outperformed when using the full PASCAL3D+ dataset. In any case, assuming that they have only used the PASCAL subset, SBVPE reports consistent results with a PCK that falls near the best case and an exceptional APK. This makes us think that SBVPE is robust enough and also, being trained with larger input size, better prepared to work with high-resolution images.

TABLE 4. PCK and APK with $\alpha = 0.1$ of different methods. SBVPE trained and validated with PASCAL, PASCAL3D+/PASCAL and PASCAL3D+ respectively. All 3 methods are the ResNet-152 backbone.

Method	Input Size	PCK (%)	APK (%)
Long et al. [40]	227x227	45.7	NA
Tulsiani et al. [28]	384x384 + 192x192	81.3	40.7
Li et al. [13]	64x64	81.8	45.4
Murthy et al. [46]	64x64	93.4	NA
SBVPE-PASCAL	384x288	83.17	50.69
SBVPE-ExtPASCAL	**384x288**	**88.49**	**55.09**
SBVPE-PASCAL3D+	384x288	97.12	72.38

TABLE 5. Median and mean instance area in square pixel for PASCAL3D+ dataset and its subsets.

Subset	#Instances	#Outliers	Median Area	Mean Area
PASCAL	2,161	288	5,265	26,695
ImageNet	5,630	348	79,810	107,630
BOTH	7,791	283	66,515	85,181

D. INSTANCES SIZE AND ITS IMPACT

As we previously said, PASCAL and ImageNet subsets from PASCAL3D+ have considerable differences in complexity. While the PASCAL subset has 1.76 instances per image, the ImageNet subset only has 1.03. Focusing on instance size, the differences are even greater. By taking a look at Fig. 3 and Table 5 the huge difference in area between PASCAL and ImageNet subsets can be appreciated, with a median and mean areas more than 15 and 4 times greater respectively for ImageNet subset.

Because of these variations in size, we wanted to analyse the impact of instances sizes on the model. To do so, we divided the validation data following the setup described in [28]. A comparison of different methods can be seen in Table 6, with 'full' being the complete validation set, 'occluded' the objects marked as truncated or occluded (739 instances in PASCAL, in the case of ImageNet we do not have this information), 'big' the bigger third of instances (357 instances in PASCAL and 932 in Imagenet), and 'small' the smaller third (357 instances in PASCAL and 932 in ImageNet). We can observe that SBVPE outperforms the previous approaches in all categories.

Focusing on occlusion, our approach is a major leap with an increase of more than 21%, reaching 84.07% PCK. This indicates that SBVPE is robust to occlusions/truncations having even better performance than with low-resolution images.

On low-resolution objects, we obtain an improvement over previous models of 9.55%, achieving 83.85% PCK. If we think of real scenarios in which distant objects are small, and occlusions happen constantly, the ability of a model to perform well with this type of objects is critical.

TABLE 6. PCK with $\alpha = 0.1$ of different methods. SBVPE trained and validated with PASCAL3D+/PASCAL and PASCAL3D+ respectively. Both methods are the ResNet-152 backbone.

PCK (%)	Full	Occluded	Small	Big
Tulsiani et al. [28]	81.3	62.8	67.4	90.0
Li et al. [13]	81.8	59.0	74.3	87.7
SBVPE-ExtPASCAL	**88.49**	**84.07**	**83.85**	**92.4**
SBVPE-PASCAL3D+	97.12	84.07	92.31	98.6

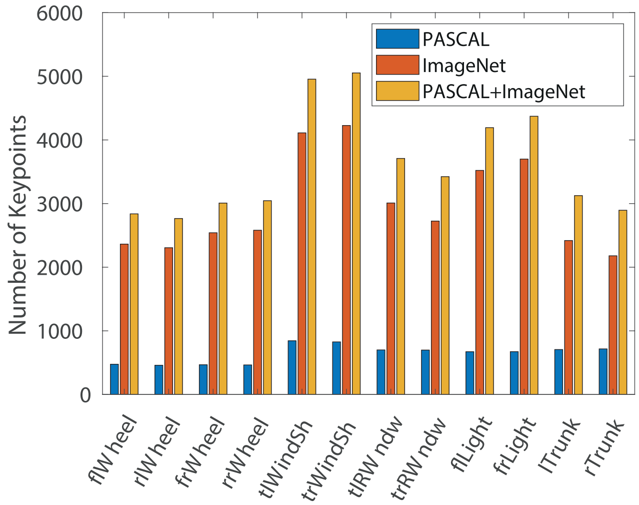

FIGURE 4. **Number of each keypoint on PASCAL3D+ dataset and its subsets.**

If we take a look at the results obtained by SBVPE-PASCAL3D+, in which we used the full PASCAL3D+, we can see a better performance than with SBVPE-ExtPASCAL, in which we used only PASCAL data for the validation. It is important to keep in mind that there are far more images from ImageNet in PASCAL3D+ than from PASCAL, and that the higher resolution of ImageNet images contributes to dilute the metrics, especially the small one.

E. KEYPOINT DISTRIBUTION STUDY

Once the model's general performance has been analysed, it is interesting to carry out a more detailed analysis at a keypoint level. The distribution of keypoints in each subset (PASCAL and ImageNet) and in the full dataset can be seen in Fig. 4.

Taking a look at the dataset, as previously stated, we have 12 keypoints. These keypoints are the following: *left front wheel, left rear wheel, right front wheel, right rear wheel, upper left windshield, upper right windshield, upper left rear window, upper right rear window, left front light, right front light, left rear trunk, and right rear trunk*. In order to perform this analysis, it is essential to know the keypoint distribution.

By taking a look at the keypoint distribution, we can obtain relevant information. We can see that the amount of keypoints is quite homogeneous regardless of the subset, having practically the same amount of images for both sides of the vehicles (same amount of wheel keypoints) and a certain predominance of the front view of the vehicles (windshield and front lights compared with rear window and trunk).

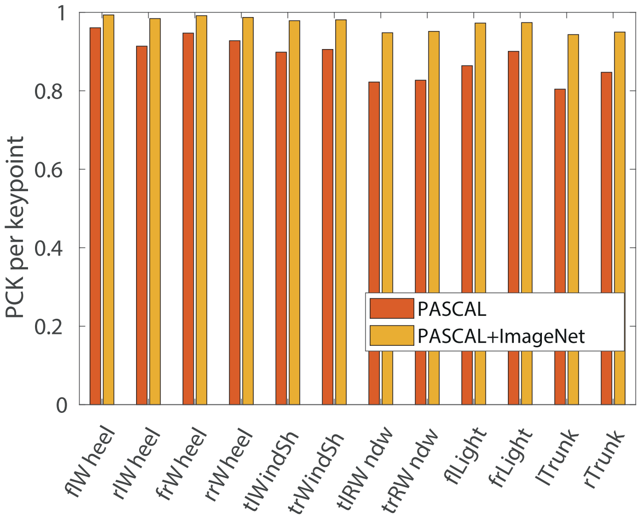

FIGURE 5. **Per keypoint PCK with $\alpha = 0.1$ for PASCAL and PASCAL3D+.**

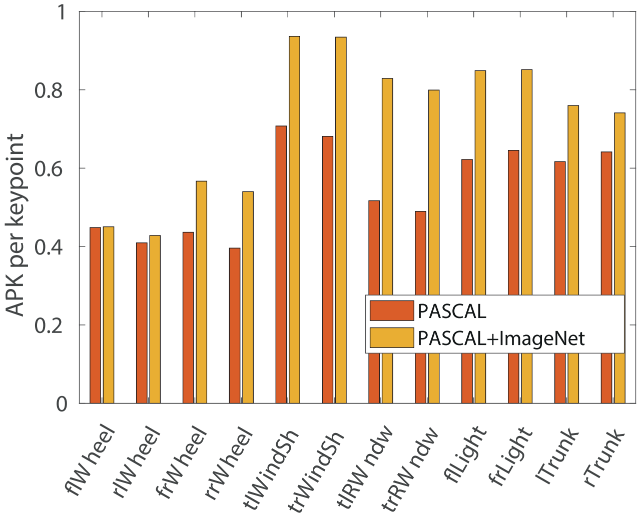

FIGURE 6. **Per keypoint APK with $\alpha = 0.1$ for PASCAL and PASCAL3D+.**

After analysing the keypoints distribution, we have to evaluate the system performance for each one of them. In Fig. 5 and Fig. 6 we have the per keypoint PCK and APK with $\alpha = 0.1$ for PASCAL and PASCAL3D+. Focusing on PCK, we can extract some interesting information. In both cases, the wheels have the best performance even though they are the least represented keypoints, while for the remaining keypoints, the higher the number of samples, the higher the performance.

Only when we take a look at the APK, we notice a curious phenomenon. The wheels, which were the keypoints with the best results in PCK, get the worst results in APK. Before thinking about what this means, it is interesting to meditate on the difference between these two metrics. While PCK is telling us that the model accurately finds the wheels, APK

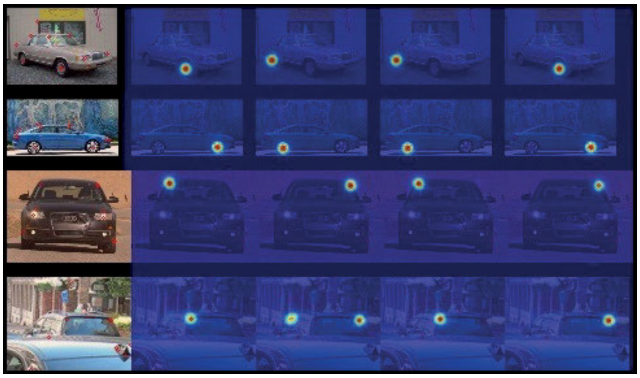

FIGURE 7. Output heatmaps for SBVPE trained with PASCAL3D+ and validated on PASCAL. Examples of the keypoint confusion phenomenon for the wheels (first and second row) and the windshield/rearwindow (third and fourth rows).

is telling us that the model is finding wheels where there are none, or rather, confusing one wheel with another. So, the only feasible explanation is that the model is perfectly capable of finding a wheel, but it has problems differentiating if it is a front or rear wheel or its side. In order to verify this theory, we performed some tests and empirically observed that, indeed, the model is not only having trouble with the side but also with the front/rear position, which explains the APK results. Taking a look at the rest of keypoints, we can appreciate a correspondence between APK and PCK/amount of each keypoint. In any case, we have empirically observed that the phenomenon that occurs with the wheels also occurs with the corners of the windshield and the rear window, although with less impact. Some examples of this phenomenon can be seen in Fig. 7.

F. CUSTOM KEYPOINTS

As we have seen while analysing the per keypoint performance, the system has a good PCK, but APK revealed some problems in the keypoints. This has led us to ask ourselves why these 12 keypoints? Also, analysing the PASCAL3D+ labels, we have noticed some flaws in it, especially in the front lights and trunk keypoints. In the case of the lights, in really old models with a single headlight, it is easy to label the centre of the headlight but, in more recent models with multiple and complex systems, where should the keypoint be placed? On the other hand, in the case of the trunk, we have detected inconsistencies in the labelling, which is not very homogeneous and has high variability. Additionally, if we think about the wheels, we will never be able to see all four at the same time, so we could only use two keypoints, one for the front wheel and another for the rear wheel, obtaining the lateral information by context from the rest of the keypoints.

For all of the above, we decided to re-label part of the images from PASCAL3D+ focusing on the images from ImageNet, as they are of higher quality facilitating the labelling process and build our own custom dataset with these ImageNet images along with images from CompCars [53] and images extracted from our recordings in real driving scenarios for the PREVENTION Dataset [54].

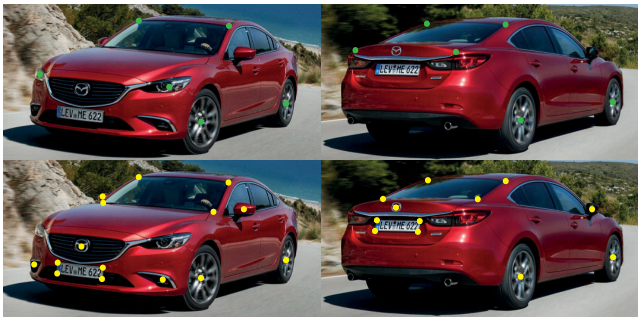

FIGURE 8. Visual comparison of PASCAL3D+ keypoints and ours. Top row/green PASCAL3D+, bottom row/yellow ours.

TABLE 7. PCK and APK with $\alpha = 0.1$. Comparison of performance between PASCAL3D+ keypoints and our keypoints. All 3 methods are the ResNet-152 backbone.

Method	Train/Val	PCK (%)	APK (%)
SBVPE	PASCAL3D+/PASCAL	88.49	55.09
SBVPE	PASCAL3D+	97.12	72.38
SBVPE	**Custom**	**98.83**	**81.92**

We decided to label 19 keypoints, being these the following: *front and rear wheels, the four corners of the windshield, the four corners of the rear window, left and right foglight, left and right rear mirror, the four corners of the license plate, and the logo*. These keypoints were chosen because we believe that they provide a greater amount of structural information about the vehicles and are less susceptible to crossed confusion. As we have said, we have chosen to eliminate the side of the wheels, keeping only two wheels (front/rear). The four corners of the windshield and the rear window serve to characterise the upper structure of the vehicle. Instead of using the front lights, which, as we have said, are complex and of varied shapes, we have chosen the fog lights, as they are much more homogeneous and are usually placed in the same area. We also considered it appropriate to add the rear-view mirrors, as they are easily distinguishable elements and provide information about the limits of the vehicle. And finally, the four corners of the license plate and logo (as with the wheels, regardless of their location at the front or rear of the vehicle), as we believe that having these elements located can be very helpful for other applications such as maker and model recognition or surveillance. A visual comparison of PASCAL3D+ keypoints and our keypoints can be seen in Fig. 8.

The total amount of labelled images is 4,042 with 4,080 instances (2,801 from the PASCAL3D+ ImageNet subset, 898 from CompCars and 381 from the PREVENTION Dataset) and we opted for a train/val split of 70/30, making a total of 2,857 instances for training and 1,223 for validation (equally distributed among the three groups).

In Table 7, we can see a comparison of performance between using the 12 PASCAL3D+ keypoints and our 19 custom keypoints.

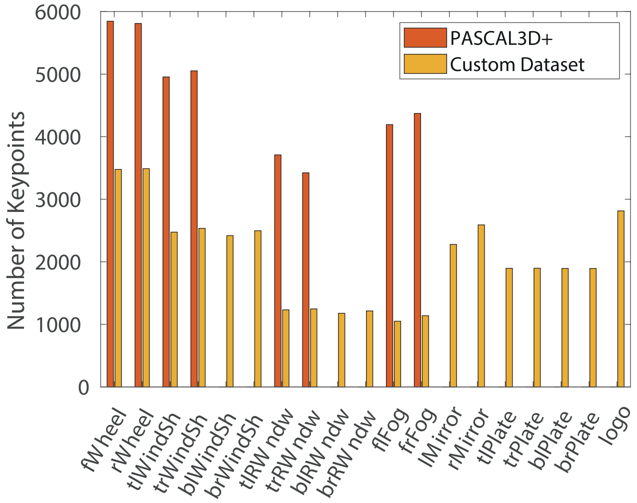

FIGURE 9. Number of each keypoint on PASCAL3D+ dataset and our custom dataset.

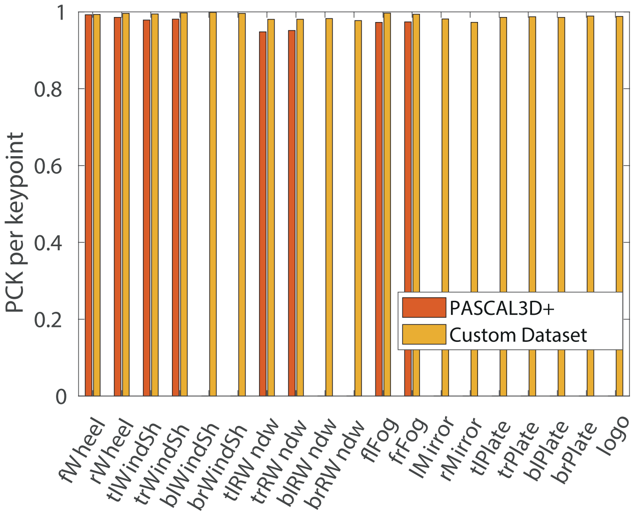

FIGURE 10. Per keypoint PCK with $\alpha = 0.1$ for PASCAL3D+ and our custom dataset.

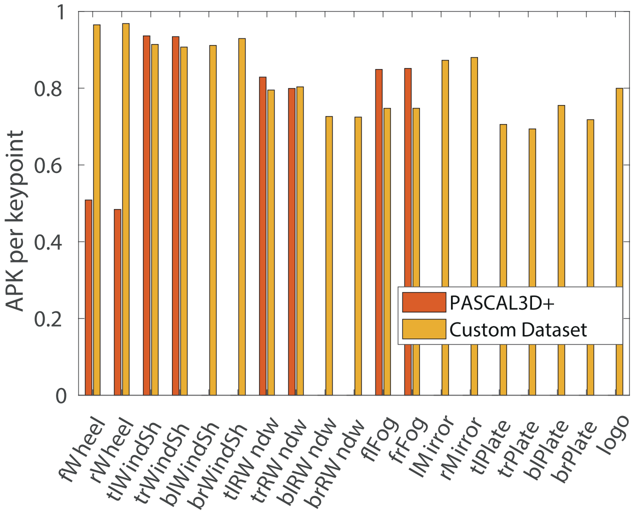

FIGURE 11. Per keypoint APK with $\alpha = 0.1$ for PASCAL3D+ and our custom dataset.

On PCK, we obtain a 98.83%, an improvement of 10.34% and 1.71% with respect to validation with PASCAL and PASCAL3D+. Focusing on APK, we obtain a huge improvement from 55.09% and 72.38% for PASCAL and PASCAL3D+ to 81.92% when using our set of custom keypoints.

These results support our keypoint proposal, and although the scenarios are not directly comparable and the PASCAL subset is still more complex, our proposal has a similar complexity and resolution to the ImageNet subset and serves to contrast its suitability, which, in the absence of the keypoint analysis, seems to be an improvement over the 12 keypoints used in PASCAL3D+.

As with PASCAL3D+, it is interesting to know the keypoint distribution. In Fig. 9 we can see the amount of each keypoint in our custom dataset compared with the corresponding keypoints in PASCAL3D+.

The matching keypoints are the wheels (we grouped the four keypoints from PASCAL3D+ for the comparison), the top corners of the windshield and rear window, and the lights (we compared the headlights from PASCAL3D+ with our foglights).

As expected, the amount of keypoints in PASCAL3D+ is higher (more instances), and the keypoint distribution is proportional.

Focusing again on the per keypoint performance, we have Fig. 10 and 11 in which a comparison of PCK and APK between PASCAL3D+ and our custom dataset can be seen. As expected, we have a consistent, almost perfect PCK, but the interesting info is in the APK. As with PASCAL3D+, we can see a correlation between the number of keypoints and APK. If we focus on the problems detected in the PASCAL3D+ keypoints that we intended to solve, we have, in the first place, the wheels. We proposed to move to a two keypoint approach in which only front and rear wheels are labelled ignoring the side. As we can see, our proposal is solid, with an APK more than 40% greater. Taking a look at the windshield and rear window corners, we see a consistent performance, with practically the same performance as with PASCAL3D+ for the top corners and equivalent for the bottom ones. It is interesting to point out that the rear window points have lower APK, which we believe is mostly caused by the lower amount of keypoints in the dataset.

Regarding the lights, we have a slightly lower performance with our approach, but this can be caused by the vast difference in the amount of keypoints. We decided to switch to the foglights because of the considerable variability of today's headlights, and even though our choice is not backed by an increase in APK with regard PASCAL3D+, we believe that this is due to the fact that the vast majority of PASCAL3D+ vehicles are old and do not yet have this variability in headlights.

Finally, analysing the rest of the new keypoints, we can see a good APK for the rear mirrors, the license plate, and

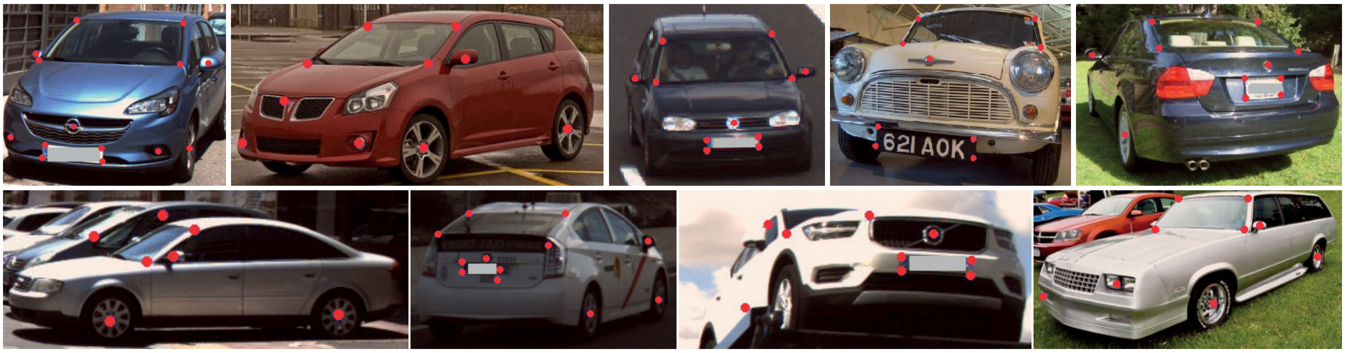

FIGURE 12. Examples of predictions from our custom dataset. Top and bottom row are good and bad predictions respectively.

the logo. We do not know for sure the reason for the "low" license plate APK, and we believe that the logo could be influenced by the fact that practically all the makers put the logos on the wheels. In any case, we place ourselves to a future further study on the viability of each keypoint.

Some prediction examples from our custom dataset can be seen in Fig. 12 being the top row correct predictions and the bottom row wrong ones.

V. CONCLUSIONS AND FUTURE WORK

This paper presents an evaluation of different keypoint prediction methods. We proposed the use of a human pose estimation state-of-the-art method (Simple Baselines for Human Pose Estimation and Tracking [1]) in order to predict vehicle keypoints efficiently. We have used PCK and APK, widely accepted metrics, to measure the performance of keypoint prediction systems. In order to train and compare our adaptation SBVPE (Simple Baseline for Vehicle Pose Estimation) with previous methods, we used the PASCAL3D+ dataset. We performed a series of experiments with which we wanted to find the best data augmentation approach and architecture, resulting in a data augmentation approach of 50% chance horizontal flip, random rotation of ±40° and scaling of ±30% and the use of ResNet-152 backbone and input size of 384 × 288.

We conducted an exhaustive analysis of PASCAL3D+ and its subsets PASCAL and ImageNet. In the first place, we used various train/val configurations in order to find the best option, finding out that the PASCAL validation set benefits from joint training with Imagenet and achieves state-of-the-art results both in PCK and APK only behind the method proposed by Murthy *et al.* [46] in PCK (APK not reported). Next, we continued analysing the impact of instances size, finding out that the PASCAL subset is by far more complex than ImageNet subset due to the higher resolution and instance size of the last one and the number of instances per image, with 1.76 in PASCAL and 1.03 in ImageNet.

After this, we analysed the keypoint distribution of PASCAL3D+. The results show a consistent PCK and an interesting APK, which, along the diverse experiments carried out, show diverse issues with the PASCAL3D+ keypoints, specifically with the wheels, the trunk, and the lights. To address these issues, we developed a custom dataset composed of images from ImageNet, CompCars, and the PREVENTION dataset with 19 keypoints. And even though the results are not directly comparable due to the differences in complexity, they show that SBVPE achieves a slightly better mean PCK and far better mean APK, with a huge increase in the performance of the wheels, comparable performance for the previous keypoints, and solid performance for the new ones.

As future work, we plan to continue expanding our dataset and studying different keypoints, its viability, and the impact they have on performance as well as explore other architectures, especially those with a bottom-up approach. Additionally, we will go one step further and make the jump from 2D to 3D pose to fully characterise the vehicle structure in a similar way to the one used by [4]–[7], [55].

ACKNOWLEDGMENT

The Joint Undertaking receives support from the European Union Horizon 2020 research and innovation programme and Germany, Austria, Spain, Italy, Latvia, Belgium, Netherlands, Sweden, Finland, Lithuania, Czech Republic, Romania, Norway. Some of the GPUs used to develop this research were donated by the NVIDIA GPU Program.

REFERENCES

[1] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 466–481.

[2] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3686–3693.

[3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[4] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond Pascal: A benchmark for 3D object detection in the wild," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2014, pp. 75–82.

[5] J. K. Murthy, G. V. S. Krishna, F. Chhaya, and K. M. Krishna, "Reconstructing vehicles from a single image: Shape priors for road scene understanding," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 724–731.

[6] W. Ding, S. Li, G. Zhang, X. Lei, and H. Qian, "Vehicle pose and shape estimation through multiple monocular vision," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2018, pp. 709–715.

[7] X. Song, P. Wang, D. Zhou, R. Zhu, C. Guan, Y. Dai, H. Su, H. Li, and R. Yang, "ApolloCar3D: A large 3D car instance understanding benchmark for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5452–5462.

[8] M. Z. Zia, M. Stark, and K. Schindler, "Towards scene understanding with detailed 3D object representations," *Int. J. Comput. Vis.*, vol. 112, no. 2, pp. 188–203, Apr. 2015.

[9] R. Juranek, A. Herout, M. Dubska, and P. Zemcik, "Real-time pose estimation piggybacked on object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2381–2389.

[10] Y. Wang, X. Tan, Y. Yang, Z. Li, X. Liu, F. Zhou, and L. S. Davis, "3D pose estimation for fine-grained object categories," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 619–632.

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[12] C. Li, M. Z. Zia, Q.-H. Tran, X. Yu, G. D. Hager, and M. Chandraker, "Deep supervision with shape concepts for occlusion-aware 3D object parsing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5465–5474.

[13] C. Li, M. Z. Zia, Q.-H. Tran, X. Yu, G. D. Hager, and M. Chandraker, "Deep supervision with intermediate concepts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1828–1843, Aug. 2019.

[14] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*. [Online]. Available: http://arxiv.org/abs/1512.03012

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[16] D. F. Llorca, R. Arroyo, and M. A. Sotelo, "Vehicle logo recognition in traffic images using HOG features and SVM," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 2229–2234.

[17] D. F. Llorca, D. Colas, I. G. Daza, I. Parra, and M. A. Sotelo, "Vehicle model recognition using geometry and appearance of car emblems from rear view images," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 3094–3099.

[18] H. Corrales, D. F. Llorca, I. Parra, S. Vigre, A. Quintanar, J. Lorenzo, and N. Hernández, "CNNs for fine-grained car model classification," in *Computer Aided Systems Theory—EUROCAST* (Lecture Notes in Computer Science), vol. 12014. Cham, Switzerland: Springer, 2020, pp. 104–112.

[19] P. Barra, C. Bisogni, M. Nappi, D. Freire-Obregon, and M. Castrillon-Santana, "Gender classification on 2D human skeleton," in *Proc. 3rd Int. Conf. Bio-Eng. Smart Technol. (BioSMART)*, Apr. 2019, pp. 1–4.

[20] M. M. Soliman, M. H. Kamal, M. A. E.-M. Nashed, Y. M. Mostafa, B. S. Chawky, and D. Khattab, "Violence recognition from videos using deep learning techniques," in *Proc. 9th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Dec. 2019, pp. 80–85.

[21] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4645–4653.

[22] P. Barra, S. Barra, C. Bisogni, M. De Marsico, and M. Nappi, "Web-shaped model for head pose estimation: An approach for best exemplar selection," *IEEE Trans. Image Process.*, vol. 29, pp. 5457–5468, 2020.

[23] D. Zhou and Q. He, "PoSeg: Pose-aware refinement network for human instance segmentation," *IEEE Access*, vol. 8, pp. 15007–15016, 2020.

[24] S. Zhang, C. Wang, Z. He, Q. Li, X. Lin, X. Li, J. Zhang, C. Yang, and J. Li, "Vehicle global 6-DoF pose estimation under traffic surveillance camera," *ISPRS J. Photogramm. Remote Sens.*, vol. 159, pp. 114–128, Jan. 2020.

[25] Z. Wang, L. Tang, X. Liu, Z. Yao, S. Yi, J. Shao, J. Yan, S. Wang, H. Li, and X. Wang, "Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 379–387.

[26] D. F. Llorca, C. Salinas, M. Jimenez, I. Parra, A. G. Morcillo, R. Izquierdo, J. Lorenzo, and M. A. Sotelo, "Two-camera based accurate vehicle speed measurement using average speed at a fixed point," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 2533–2538.

[27] D. F. Llorca, H. Corrales, I. Parra, M. Rentero, R. Izquierdo, A. Hernández-Saz, and I. García-Daza, "License plate corners localization using CNN-based regression," in *Computer Aided Systems Theory—EUROCAST* (Lecture Notes in Computer Science), vol. 12014. Cham, Switzerland: Springer, 2020, pp. 113–120.

[28] S. Tulsiani and J. Malik, "Viewpoints and keypoints," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1510–1519.

[29] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 483–499.

[30] N. D. Reddy, M. Vo, and S. G. Narasimhan, "Occlusion-Net: 2D/3D occluded keypoint localization using graph networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7326–7335.

[31] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded pyramid network for multi-person pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7103–7112.

[32] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[33] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[34] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[35] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7291–7299.

[36] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2277–2287.

[37] L. Pishchulin, E. Insafutdinov, S. Tang, B. Andres, M. Andriluka, P. Gehler, and B. Schiele, "DeepCut: Joint subset partition and labeling for multi person pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4929–4937.

[38] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele, "DeeperCut: A deeper, stronger, and faster multi-person pose estimation model," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 34–50.

[39] S. Kreiss, L. Bertoni, and A. Alahi, "PifPaf: Composite fields for human pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11977–11986.

[40] J. L. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1601–1609.

[41] Z. Wang, G. Liu, and G. Tian, "A parameter efficient human pose estimation method based on densely connected convolutional module," *IEEE Access*, vol. 6, pp. 58056–58063, 2018.

[42] I. Radwan, N. Moustafa, B. Keating, K.-K.-R. Choo, and R. Goecke, "Hierarchical adversarial network for human pose estimation," *IEEE Access*, vol. 7, pp. 103619–103628, 2019.

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[44] X. Wang, Z. Cao, R. Wang, Z. Liu, and X. Zhu, "Improving human pose estimation with self-attention generative adversarial networks," *IEEE Access*, vol. 7, pp. 119668–119680, 2019.

[45] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DoF object pose from semantic keypoints," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 2011–2018.

[46] J. K. Murthy, S. Sharma, and K. M. Krishna, "Shape priors for real-time monocular object localization in dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1768–1774.

[47] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4724–4732.

[48] A. Nibali, Z. He, S. Morgan, and L. Prendergast, "Numerical coordinate regression with convolutional neural networks," 2018, *arXiv:1801.07372*. [Online]. Available: http://arxiv.org/abs/1801.07372

[49] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. 4th Int. Conf. Learn. Represent. ICLR*, Y. Bengio and Y. LeCun, Eds. San Juan, Puerto Rico, May 2016, pp. 2–14. [Online]. Available: http://arxiv.org/abs/1511.07122

[50] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2878–2890, Dec. 2013.

[51] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. Accessed: Apr. 14, 2020. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[53] L. Yang, P. Luo, C. C. Loy, and X. Tang, "A large-scale car dataset for fine-grained categorization and verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3973–3981.

[54] R. Izquierdo, A. Quintanar, I. Parra, D. Fernandez-Llorca, and M. A. Sotelo, "The PREVENTION dataset: A novel benchmark for PREdiction of VEhicles iNTentIONs," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3114–3121.

[55] L. Bertoni, S. Kreiss, and A. Alahi, "MonoLoco: Monocular 3D pedestrian localization and uncertainty estimation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6861–6871.

HÉCTOR CORRALES SÁNCHEZ (Graduate Student Member, IEEE) received the B.S. degree in telecommunications engineering (telematics specialty) and the M.S. degree in telecommunications engineering (intelligent transportation systems specialty) from the University of Alcalá (UAH), Alcalá de Henares, Spain, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree in information and communications technologies. His current research interests include machine learning, computer vision, deep learning, and autonomous driving.

ANTONIO HERNÁNDEZ MARTÍNEZ received the B.S. degree in industrial engineering (electronic and automatic specialty) and the M.S. degree in industrial engineering (robotics and perception specialty) from the University of Alcalá (UAH), in 2017 and 2019, respectively, where he is currently pursuing the Ph.D. degree in information and communications technologies. His current research interests include autonomous driving, deep learning, and computer vision focused on intelligent transportation systems security.

RUBÉN IZQUIERDO GONZALO (Graduate Student Member, IEEE) was born in Madrid, Spain, in 1990. He received the M.Eng. degree in industrial engineering from the University of Alcalá (UAH), Alcalá de Henares, Spain, where he is currently pursuing the Ph.D. degree with the Computer Engineering Department, with a focus on the improvement of trajectory planning for autonomous vehicles based on the intention prediction of human-driven vehicles.

NOELIA HERNÁNDEZ PARRA received the M.Sc. and Ph.D. degrees in advanced electronics systems (intelligent systems) from the University of Alcalá (UAH), in 2009 and 2014, respectively. Her thesis presented a new approach for estimating the global position of a mobile device in indoor environments by using Wi-Fi devices, which received the Best Ph.D. Award by UAH, in 2014. She is currently an Associate Professor with the Computer Engineering Department, UAH. Her research interests include indoor and outdoor localization, artificial intelligence, and intelligent transportation systems.

IGNACIO PARRA ALONSO received the M.S. and Ph.D. degrees in telecommunications engineering from the University of Alcalá (UAH), in 2005 and 2010, respectively. He is currently an Associate Professor with the Computer Engineering Department, UAH. His research interests include intelligent transportation systems and computer vision. He received the Master Thesis Award in eSafety from the ADA Lectureship at the Technical University of Madrid, Spain, in 2006.

DAVID FERNÁNDEZ-LLORCA (Senior Member, IEEE) received the M.S. and Ph.D. degrees in telecommunications engineering from the University of Alcalá (UAH), Madrid, Spain, in 2003 and 2008, respectively. He is currently a Full Professor with UAH. He has authored more than 120 refereed publications in international journals, book chapters, and conference proceedings. His research interests include intelligent vehicles and traffic technologies. He received the Best Young Researcher Award from the IEEE ITS Society, in 2018. He is currently the Editor-in-Chief of the *IET Intelligent Transport Systems Journal*.