Improving Car Model Classification Through Vehicle Keypoint Localization

Alessandro Simoni[2] [a], Andrea D'Eusanio[1] [b], Stefano Pini[2] [c],
Guido Borghi[3] [d] and Roberto Vezzani[1,2] [e]

[1]*Artificial Intelligence Research and Innovation Center, University of Modena and Reggio Emilia, 41125 Modena, Italy*

[2]*Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, 41125 Modena, Italy*

[3]*Department of Computer Science and Engineering, University of Bologna, 40126 Bologna, Italy*

{*alessandro.simoni, andrea.deusanio, s.pini, roberto.vezzani*}*@unimore.it, guido.borghi@unibo.it*

Keywords: car model classification, vehicle keypoint localization, multi-task learning

Abstract: In this paper, we present a novel multi-task framework which aims to improve the performance of car model classification leveraging visual features and pose information extracted from single RGB images. In particular, we merge the visual features obtained through an image classification network and the features computed by a model able to predict the pose in terms of 2D car keypoints. We show how this approach considerably improves the performance on the model classification task testing our framework on a subset of the Pascal3D+ dataset containing the car classes. Finally, we conduct an ablation study to demonstrate the performance improvement obtained with respect to a single visual classifier network.

1 INTRODUCTION

The classification of vehicles, and specifically car models, is a crucial task in many real world applications and especially in the automotive scenario, where controlling and managing the traffic can be quite complex. Moreover, since visual traffic surveillance has an important role in computer vision, car classification can be an enabling feature for other tasks like vehicle re-identification or 3D vehicle reconstruction. Despite these considerations, a little effort has been done in the computer vision community to improve the accuracy of the existing systems and to propose specialized architectures based on the recent deep learning paradigm. Indeed, from a general point of view, car model classification is a challenging task in the computer vision field, due to the large quantity of different models produced by many car companies and the large differences in the appearance with unconstrained poses (Palazzi et al., 2017). Therefore, viewpoint-aware analyses and robust classification algorithms are strongly demanded.

[a] https://orcid.org/0000-0003-3095-3294
[b] https://orcid.org/0000-0001-8908-6485
[c] https://orcid.org/0000-0002-9821-2014
[d] https://orcid.org/0000-0003-2441-7524
[e] https://orcid.org/0000-0002-1046-6870

Only recently, some works in the literature have faced the classification problem trying to distinguish between vehicle macro-classes, such as aeroplane, car and bicycle. For instance, in (Afifi et al., 2018) a multi-task CNN architecture that performs vehicle classification and viewpoint estimation simultaneously has been proposed. In (Mottaghi et al., 2015) a coarse-to-fine hierarchical representation has been presented in order to perform object detection, to estimate the 3D pose and to predict the sub-category vehicle class. However, we note that learning to discriminate between macro-classes is less challenging than categorizing different specific car models.
In (Grabner et al., 2018) the task is addressed through the use of depth images computed from the 3D models. The proposed method not only estimates the vehicle pose, but also perform a 3D model retrieval task. Other works (Xiao et al., 2019; Kortylewski et al., 2020) are focused on the vehicle and object classification task under partial occlusions.
The work most closely related to our system has been proposed by Simoni et al. in (Simoni et al., 2020), where a framework to predict the visual future appearance of an urban scene is described. In this framework, a specific module is committed to classify the car model from RGB images, in order to select a similar 3D model to be placed into the final generated

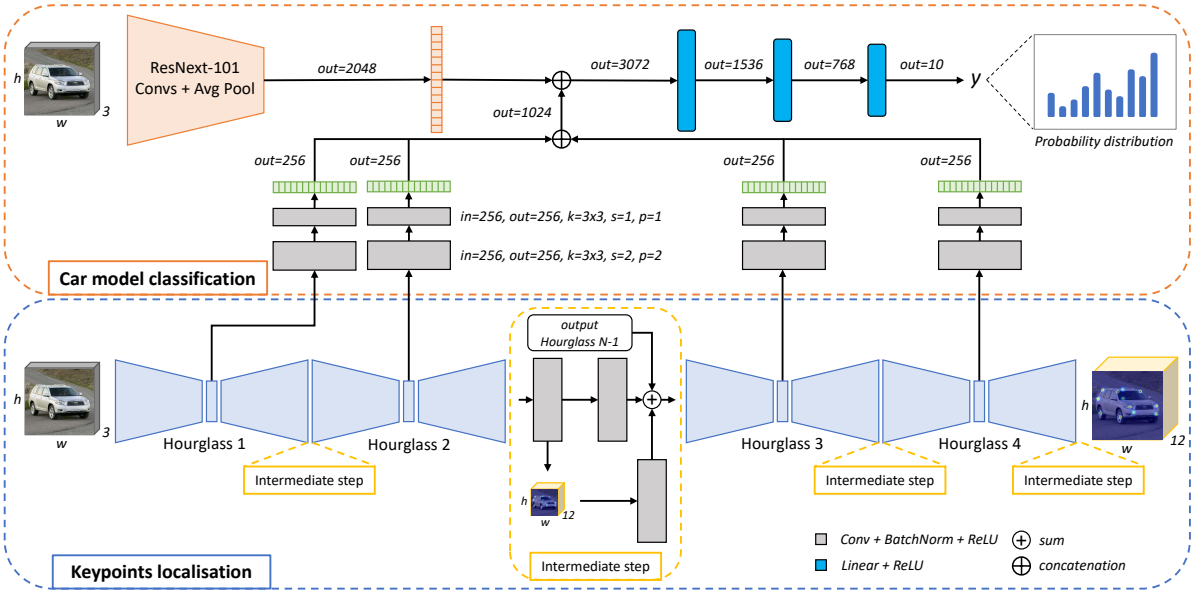

Figure 1: Overview of the proposed framework. At the top, the car model classifier is reported, where the visual features – extracted from ResNeXt-101 – are combined with the intermediate feature maps computed by the keypoint localization network – Stacked-Hourglass – shown in the bottom. The input is a single RGB image for both modules, while the output is the keypoint heatmaps and the classified car model.

images.

In this paper, we address the specific task of car model classification, in terms of vehicle typology (e.g., pick-up, sedan, race car and so on). Our starting intuition is that the localization of 2D keypoints on the RGB images can be efficiently exploited to improve the car model classification task. As training and testing dataset, we exploit the Pascal3D+ (Xiang et al., 2014), one of the few datasets containing a great amount of data annotated with 3D vehicle models, 2D keypoints and pose in terms of 6DoF. In the evaluation procedure, we investigate how the architectures currently available in the literature are able to deal with the car model classification and the keypoint detection task. Specifically, we conduct an investigation of the performance of these models applied to the specific tasks. Then, we present how to merge visual information and the 2D skeleton data, encoded from an RGB image of the car, proposing a new multi-task framework. We show that exploiting both information through a multi-task system leads to an improvement of the classification task, without degrading the accuracy of the pose detector.

The rest of the paper is organized as follows. In Section 2 the proposed method is detailed, analyzing firstly the car model classification and the 2D keypoint localization modules and then our combined approach. Section 3 contains the experimental evaluation of our proposed method and an ablation study on several tested baselines for both the tasks presented in the previous section. Finally, a performance analysis is conducted and conclusions are drawn.

2 PROPOSED METHOD

In this section, we describe our method that improves the accuracy of the car model classification by leveraging on the side task of 2D keypoint localization. The architecture is composed of two subnetworks, each tackling a different task as detailed in the following.

2.1 Car model classification

The car model classification task aims to extract visual information from RGB images of vehicles and to classify them in one of the possible classes, each corresponding to a specific 3D vehicle model. Among several classifiers, we choose the ResNeXt-101 network from (Xie et al., 2017), which is a slightly modified version of the ResNet architecture (He et al., 2016). The network takes as input an RGB vehicle image of dimension 256×256 and outputs a probability distribution over n possible car model classes. The distinctive aspect of this architecture is the introduction of an aggregated transformation technique that replaces the classical residual blocks with C parallel embedding transformations, where the parameter

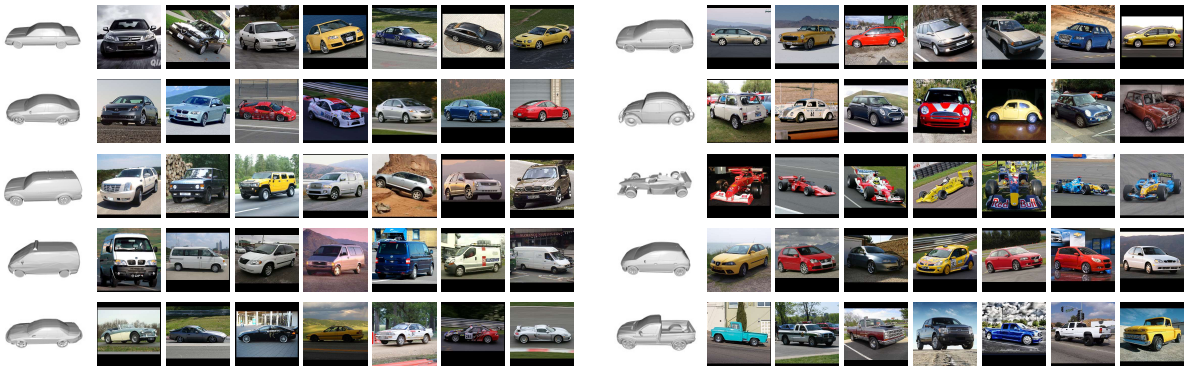

Figure 2: Image samples from Pascal3D+ dataset for each car model class.

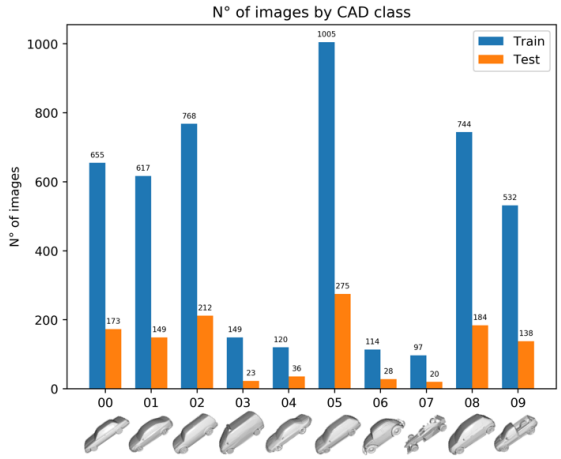

Figure 3: Images distribution through train and test set for each vehicle sub-category in Pascal3D+ dataset.

C is also called *cardinality*. The resulting embeddings can be aggregated with three equivalent operations: i) sum, ii) concatenation or iii) grouped convolutions. This data transformation has proved to obtain higher level features than the ones obtained from the residual module of ResNet. This statement is also confirmed by better performance on our task, as shown later in Section 3. We refer to this section also for a comparison between different visual classifiers.

2.2 2D keypoints localization

The second task in hand is the localization of semantic keypoints representative of the vehicle skeleton. Finding 2D object landmarks and having its corresponding 3D model can be useful to estimate and reproduce the object pose in the 3D world using well-known correspondence methods and resolving a *perspective-n-point* problem. Similarly to the classification task, there are many CNNs that can solve the 2D keypoint localization task. We choose the architecture presented by (Newell et al., 2016) between several alternatives, whose comparison is reported in Section 3. This network is called *Stacked-Hourglass* since it is composed of a encoder-decoder structure, called *hourglass*, which is repeated N times composing a stacked architecture. The network takes as input the RGB vehicle image of dimension 256×256 and every hourglass block outputs an intermediate result, which is composed by k Gaussian heatmaps where the maximum value identifies the keypoint location. The presence of multiple outputs through the architecture allows to finely supervise the training by applying the loss to every intermediate output. We tested the network with $N = [2, 4, 8]$ and chose to employ an architecture with $N = 4$ hourglass blocks which obtains the best trade-off between score and performance.

2.3 Combined approach

Testing the sole ResNeXt model as a visual classifier proved that the car classification is actually a non trivial task. Our proposal is to improve the car model prediction leveraging a multi-task technique that embraces the keypoint localization task too. As depicted in Figure 1, we combine pose features extracted by Stacked-Hourglass and visual features extracted by ResNeXt to obtain a more reliable car model classification.

In practice, we leverage the features coming from each hourglass block and analyze them with two convolutional layers with 256 kernels of size 3×3, shared weights and ReLU activation function. While the first layer has stride and padding equal to 2, the second one has stride and padding 1. Since the Stacked-Hourglass architecture has $N = 4$ hourglass blocks, 4 pose features are obtained. We thus combine them with an aggregation function and concatenate them to the visual features extracted by ResNeXt-101. The fused features are passed through 2 fully connected

layers, with 1536 and 768 hidden units and ReLU activation functions. Finally, a linear classifier with n units followed by a softmax layer provides the probability distribution over the 3D car models.

Two different approaches are taken into account for the aggregation of the features obtained by Stacked-Hourglass. One approach consists in summing the 1D tensors of every encoder and concatenating the summed tensors to the 1D tensor containing the visual features of ResNeXt. Another approach corresponds to first concatenate the 1D tensors of every encoder and then the one extracted by ResNeXt. In both cases, the resulting features are passed through the 3 fully connected layers that perform the classification task.

To further explain our method, we describe the mathematical formulation of the performed operations. Our multi-task car classification method can be defined as a function

$$\Phi : \mathbb{R}^{w \times h \times c} \rightarrow \mathbb{R}^{n} \tag{1}$$

that maps an RGB image I to a probability distribution of n possible car model classes. This function is composed of the two subnetworks presented above and defined as following.

The Stacked-Hourglass architecture is a function

$$\mathrm{H} : \mathbb{R}^{w \times h \times c} \rightarrow \mathbb{R}^{w \times h \times k} \tag{2}$$

that maps the RGB image I to k heatmaps representing the probability distribution of each keypoint. The keypoint location is retrieved computing the maximum of each heatmap.
The function H is composed of N encoder-decoder blocks called hourglass. Each encoder E_i of the i-th hourglass outputs a set of features $\mathbf{v}^i_{\text{enc}}$ containing $m = 256$ channels. A series of two convolutional layers are further applied to the output of the encoder block:

$$\Psi : \mathbb{R}^{(w/64) \times (h/64) \times m} \rightarrow \mathbb{R}^{m} \tag{3a}$$

$$\mathbf{u}^i_{\text{pose}} = \Psi(\mathbf{v}^i_{\text{enc}}) \tag{3b}$$

where the resulting features have lost their spatial resolution.

Similarly, the parallel ResNeXt architecture, used as visual feature extractor, can be represented as a function

$$\mathrm{G} : \mathbb{R}^{w \times h \times c} \rightarrow \mathbb{R}^{l} \tag{4a}$$

$$\mathbf{u}_{\text{vis}} = \mathrm{G}(I) \tag{4b}$$

that extracts $l = 2048$ visual features from the RGB image I.

The pose features extracted from the hourglass architecture can be aggregated in two ways, as defined previously. Following the *sum* approach, the operation is defined as

$$\dot{\mathbf{u}}_{\text{pose}} = \mathbf{u}^1_{\text{pose}} + \ldots + \mathbf{u}^N_{\text{pose}} \tag{5}$$

Alternatively, the *concatenation* approach is defined as

$$\dot{\mathbf{u}}_{\text{pose}} = \mathbf{u}^1_{\text{pose}} \oplus \ldots \oplus \mathbf{u}^N_{\text{pose}} \tag{6}$$

In both cases, $N = 4$ is the number of hourglass blocks and $\oplus$ represents the concatenation operation.

Then, the pose and visual features are combined and given as input to a series of two fully connected layers followed by a linear classifier

$$\mathbf{y} = Y(\dot{\mathbf{u}}_{\text{pose}} \oplus \mathbf{u}_{\text{vis}}) \tag{7}$$

obtaining a probability distribution over the n classes of 3D car models.

3 EXPERIMENTAL EVALUATION

In this paragraph we report details about the dataset, the training procedure and the results in terms of several metrics, execution time and memory consumption.

3.1 Pascal3D+ Dataset

The Pascal3D+ dataset (Xiang et al., 2014) was presented for the 3D object detection and pose estimation tasks. However, to the best of our knowledge, it is still one of the few datasets that contains RGB images annotated with both 3D car models and 2D keypoints. The dataset is split in 12 main categories from which we select the *car* category. This category is further split in 10 car models (*e.g.* sedan, hatchback, pickup, suv) and contains 12 keypoints, listed in Table 4. As can be seen in Figure 2 and 3, every image is classified into one of ten 3D models sub-categories and both 3D and 2D keypoints are included. Filtering the images of the *car* class, we obtain a total of 4081 training and 1024 testing images. We process these images in order to guarantee that each vehicle, with its keypoints, is completely visible, *i.e.* contained into the image.
All the images are center cropped and resized to a dimension of 256×256 pixels. Following the dataset structure, we set the number of predicted classes $n = 10$ and the number of predicted heatmaps $k = 12$.

3.2 Training

The training of our model can be defined as a *two-step* procedure. Therefore, in order to extract meaningful pose features for vehicle keypoints, we first train

Method	Fusion	Accuracy
(Simoni et al., 2020)	-	65.91%
ResNeXt-101	-	66.96%
Stacked-HG-4 + (Simoni et al., 2020)	*sum*	67.61%
Stacked-HG-4 + (Simoni et al., 2020)	*concat*	69.07%
Ours	*sum*	68.26%
Ours	***concat***	**70.54%**

Table 1: Average accuracy results on features fusion classification method.

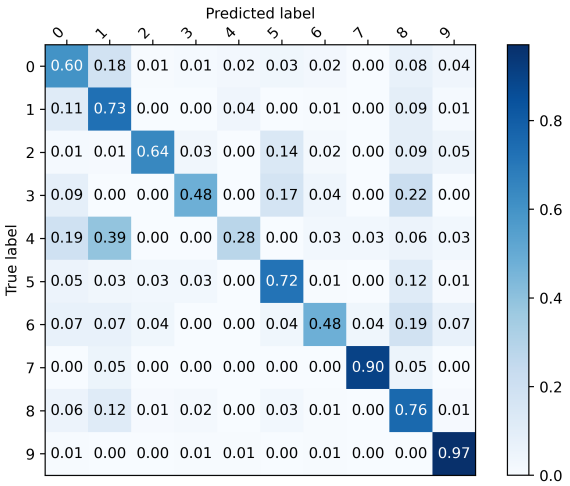

Figure 4: Normalized confusion matrix for features concatenation classification method.

the Stacked-Hourglass model on Pascal3D+ for 100 epochs, using an initial learning rate of $1e^{-3}$ and decreasing it by a factor of 10 every 40 epochs. The network is trained with a Mean Squared Error loss computed between the predicted and the ground truth keypoints heatmaps.

The second step starts by freezing both a ResNeXt model, pre-trained on ImageNet (Deng et al., 2009), and the Stacked-Hourglass model, trained on Pascal3D+; the aim is to train the convolutional layers, that modify the hourglass embedding dimensions, and the final fully connected layers, that take as input the concatenated features, on the classification task. This training lasts for 100 epochs using a fixed learning rate of $1e^{-4}$. In this case, we employ the standard Categorical Cross Entropy loss.

Code has been developed using the PyTorch (Paszke et al., 2017) framework and for each step we used Adam (Kingma and Ba, 2014) as optimizer.

3.3 Results

Here, we report the results obtained by our multi-task technique and compare them with a baseline, *i.e.* the plain ResNeXt-101 finetuned on the car model

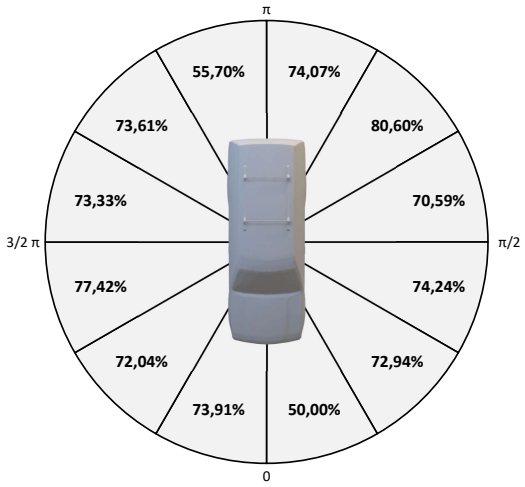

Figure 5: Average accuracy results with regard to vehicle viewpoint orientation.

classes, and the literature.

As detailed in Section 2, the proposed method can combine the pose features encoded by the Stacked-Hourglass network with two different approaches, namely *sum* and *concatenation* (concat). As a baseline, we employ the plain ResNeXt-101 architecture, finetuned with the Categorical Cross Entropy loss for 100 epochs. In order to compare with the literature, we report the results obtained by (Simoni et al., 2020), that employ a VGG-19 architecture for the task of car model classification. Moreover, we adapt our proposed architecture, which combines Stacked-Hourglass and ResNeXt-101, to integrate Stacked-Hourglass, for the keypoint localization, with the method proposed in (Simoni et al., 2020), for the model car classification.

In table 1, we show the results in terms of car model classification accuracy. As shown, the proposed method outperforms the baseline and the competitors. Moreover, the combination of the keypoint localization task and the car model classification one steadily improves the results, regardless of the employed classification architecture. Regarding the different combination approaches, the *sum* approach improves the classification score of an absolute $+1.3\%$ with respect to the baseline (ResNext-101). The *concat* approach benefit even more the classification results doubling the accuracy improvement ($+3.6\%$) with respect to the sum approach.

We report in Figure 4 the confusion matrix of the proposed method, in the concatenation setting. As it can be seen, most of the classes are recognized with high accuracy, *i.e.* 60% or higher. The sole exceptions are the classes 3, 4 and 6 that are, along with class 7, the less represented classes in both the train and the test set. In particular, even though the class 7 is one of the

Network	Layers	Accuracy
VGG16 (Simonyan and Zisserman, 2014)	*last fc*	65.18%
VGG16 (Simonyan and Zisserman, 2014)	*all fc*	65.10%
ResNet-18 (He et al., 2016)	*last fc*	59.01%
ResNet-18 (He et al., 2016)	*all*	58.20%
DenseNet-161 (Huang et al., 2017)	*last fc*	65.02%
ResNeXt-101 (Xie et al., 2017)	***last fc***	**66.96%**

Table 2: Average accuracy results over the 10 car model classes. The second columns show the trained layers while other layers are pretrained on ImageNet.

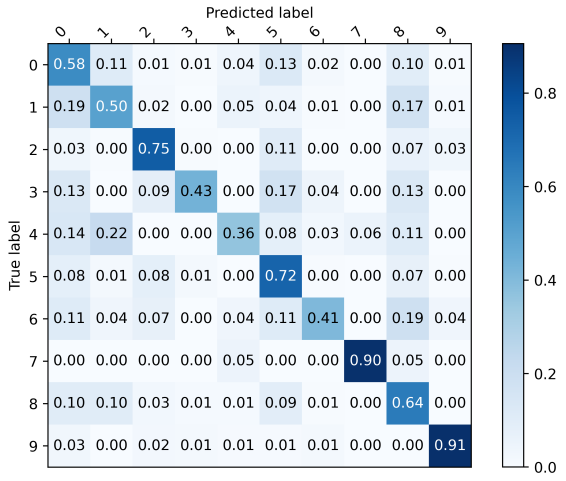

Figure 6: Normalized confusion matrix for ResNeXt-101 classification network.

less represented classes, it has an high classification score because it represents sports cars whose images features are more likely to be different from the other classes. It is worth noting that we are aware of the class imbalance problem of the dataset (as depicted in Figure 3), but, as we observed in some experiments using an inverse weighting during training (*i.e.* samples from the most common classes are weighted less than samples from the uncommon classes), the results do not have any relevant improvements.

In addition, we show the model accuracy with respect to the azimuth of the vehicle in Figure 5. Among values steadily above the 70%, there is a significant drop in accuracy for the angles ranging in $[0, \frac{\pi}{6}]$ and $[\pi, \frac{7}{6}\pi]$. This may be caused by the viewpoint, that may be less informative than the others, by a less represented azimuth range in the training set, or by a more frequent azimuth range for rare or complex cars. This behavior will be the subject for future investigation.

3.4 Ablation study

This section covers a quantitative ablation study over several classification and keypoint localization networks, showing the results for both tasks.

Model	PCKh@0.5
(Long et al., 2014)	55.7%
(Tulsiani and Malik, 2015)	81.3%
OpenPose-ResNet152 (Cao et al., 2017)	84.87%
OpenPose-DenseNet161 (Cao et al., 2017)	86.68%
(Zhou et al., 2018)	90.00%
HRNet-W32 (Wang et al., 2020)	91.63%
HRNet-W48 (Wang et al., 2020)	92.52%
(Pavlakos et al., 2017)	93.40%
Stacked-HG-2 (Newell et al., 2016)	93.41%
Stacked-HG-4 (Newell et al., 2016)	**94.20%**
Stacked-HG-8 (Newell et al., 2016)	93.92%

Table 3: Average PCK score (PCKh@0.5 with $\alpha = 0.1$) for every keypoint localization baseline (HG = Hourglass).

3.4.1 Classification

As shown in Table 2, we tested several baselines as visual classifiers. We trained each network for 150 epochs with a fixed learning rate of $1e^{-4}$ and the Adam optimizer. The objective is the Categorical Cross Entropy loss.

It's worth noting that the best results are obtained by ResNeXt-101 with an average accuracy of 66.96%, in spite of the fact all networks except ResNet-18 are quite close to each other. The results also reveal that networks with a good amount of parameters (see Tab. 5) tend to perform better on the Pascal3D+ dataset than smaller networks like ResNet-18.

Moreover, the good performance of ResNeXt-101 can be clearly observed in Figure 6, where the accuracy score is defined for each class. We noted, with respect to the other networks, that ResNeXt-101 generates a less sparse confusion matrix, *i.e.* the classifier tends to swap fewer classes one another.

3.4.2 Keypoints localization

Similarly to the classification task, we tested three architectures, named respectively OpenPose (Cao et al., 2017), HRNet (Wang et al., 2020) and Stacked-Hourglass (Newell et al., 2016), to address the keypoints localization. These architectures are studied as human pose estimation architectures, but we adapt them to our vehicle keypoint estimation task. Each network is trained for 100 epochs using a starting learning rate of $1e^{-3}$ decreased every 40 epochs by a factor of 10 and the Adam optimizer. To evaluate each network we use the PCK metric presented in (Andriluka et al., 2014). In details, we adopt the PCKh@0.5 with $\alpha = 0.1$, which represents the per-

Keypoint $(*)$	**HG-2**	**HG-4**	**HG-8**	**OP-ResNet**	**OP-DenseNet**	**HRNet-W32**	**HRNet-W48**
lb trunk	93.27	**94.69**	94.18	83.94	86.86	91.72	94.45
lb wheel	92.27	**94.17**	93.09	81.58	84.85	90.26	91.78
lf light	92.85	**93.27**	93.22	86.29	86.34	90.87	91.27
lf wheel	94.41	**95.49**	94.27	86.10	87.70	91.48	89.17
rb trunk	92.59	**92.97**	92.72	83.19	87.00	91.94	92.25
rb wheel	91.50	**91.87**	93.33	79.67	84.35	92.00	91.61
rf light	93.01	**93.79**	93.28	86.47	84.81	89.59	91.54
rf wheel	91.73	**92.71**	92.54	81.52	82.00	89.12	91.16
ul rearwindow	94.67	**95.82**	95.18	86.34	88.06	91.08	93.63
ul windshield	96.00	**96.51**	96.10	89.29	91.37	94.47	95.62
ur rearwindow	93.27	**93.52**	93.91	85.21	87.45	92.39	92.82
ur windshield	95.47	**95.58**	95.17	88.80	89.32	94.59	94.91

Table 4: PCK scores (%) for each vehicle keypoint (HG = Hourglass, OP = OpenPose).
$(*)$ lb = left back, lf = left front, rb = right back, rf = right front, ul = upper left, ur = upper right

Model	**Parameters** (M)	**Inference** (ms)	**VRAM** (GB)
VGG19	139.6	6.843	1.239
ResNet-18	11.2	3.947	0.669
DenseNet-161	26.5	36.382	0.995
ResNeXt-101	86.8	33.924	1.223
Stacked-HG-4	13.0	41.323	0.941
OpenPose	29.0	19.909	0.771
HRNet	63.6	60.893	1.103
Ours	106.8	68.555	1.389

Table 5: Performance analysis of the proposed method. We report the number of parameters, the inference time and the amount of video RAM (VRAM) needed to reproduce experimental results. We used a NVidia 1080Ti graphic card.

centage of keypoints whose predicted location is not further than a threshold from the ground truth. The value 0.5 is a threshold applied on the confidence score of each keypoint heatmap while α is the tunable parameter that controls the area surrounding the correct location where a keypoints should lie to be considered correctly localized.

Although recent architectures like OpenPose and HRNet demonstrate impressive results on human joint prediction, the older Stacked-Hourglass overcomes these competitors in the estimation of the 12 semantic keypoints of the Pascal3D+ vehicles, as shown in Table 3 and Table 4. It is worth noting that its precision is not only superior on the overall PCK score averaged on all keypoints listed in Table 3, but also on the single PCK score for each localized keypoint, as show in Table 4.

3.5 Performance analysis

We also assess the performance of the tested and the proposed methods in terms of number of parameters, inference time on a single GPU and VRAM occupancy on the graphic card. In particular, we compare our approach to all the baselines that performs separately each task. We test them on a workstation with an *Intel Core* i7-7700K and a *Nvidia GeForce GTX* 1080Ti.

As illustrated in Table 5, our approach has a large number of parameters, but it can perform both keypoints localization and car model classification at once. Taking into account the inference time and the memory consumption, our architecture works largely in real time speed with low memory requirements while performing two tasks in an end-to-end fashion obtaining better results than using the single networks.

4 CONCLUSIONS

In this paper, we show how visual and pose features can be merged in the same framework in order to improve the car model classification task. Specifically, we leverage on the ResNext-101 architecture, for the visual part, and on Stacked-Hourglass, for the car keypoint localization, to design a combined architecture. Experimental results confirm the accuracy and the feasibility of the presented method for real world applications. Moreover, the performance analysis confirm the limited inference time and the low amount of video memory required to run the system. Nonetheless, our method can improve in particular for misclassified classes which are less represented in the dataset. We leave further analysis and experiments on this issue for future work.

ACKNOWLEDGEMENTS

This research was supported by MIUR PRIN project "PREVUE: PRediction of activities and Events by Vision in an Urban Environment", grant ID E94I19000650001.

REFERENCES

Afifi, A. J., Hellwich, O., and Soomro, T. A. (2018). Simultaneous object classification and viewpoint estimation using deep multi-task convolutional neural network. In *VISIGRAPP (5: VISAPP)*, pages 177–184. 1

Andriluka, M., Pishchulin, L., Gehler, P., and Schiele, B. (2014). 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pages 3686–3693. 6

Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7291–7299. 6

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee. 5

Grabner, A., Roth, P. M., and Lepetit, V. (2018). 3d pose estimation and 3d model retrieval for objects in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3022–3031. 1

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. 2, 6

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708. 6

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 5

Kortylewski, A., He, J., Liu, Q., and Yuille, A. L. (2020). Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8940–8949. 1

Long, J. L., Zhang, N., and Darrell, T. (2014). Do convnets learn correspondence? In *Advances in neural information processing systems*, pages 1601–1609. 6

Mottaghi, R., Xiang, Y., and Savarese, S. (2015). A coarse-to-fine model for 3d pose estimation and sub-category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 418–426. 1

Newell, A., Yang, K., and Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer. 3, 6

Palazzi, A., Borghi, G., Abati, D., Calderara, S., and Cucchiara, R. (2017). Learning to map vehicles into bird's eye view. In *International Conference on Image Analysis and Processing*, pages 233–243. Springer. 1

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch. 5

Pavlakos, G., Zhou, X., Chan, A., Derpanis, K. G., and Daniilidis, K. (2017). 6-dof object pose from semantic keypoints. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2011–2018. IEEE. 6

Simoni, A., Bergamini, L., Palazzi, A., Calderara, S., and Cucchiara, R. (2020). Future urban scenes generation through vehicles synthesis. In *International Conference on Pattern Recognition (ICPR)*. 1, 5

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. 6

Tulsiani, S. and Malik, J. (2015). Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519. 6

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al. (2020). Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*. 6

Xiang, Y., Mottaghi, R., and Savarese, S. (2014). Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE winter conference on applications of computer vision*, pages 75–82. IEEE. 2, 4

Xiao, M., Kortylewski, A., Wu, R., Qiao, S., Shen, W., and Yuille, A. (2019). Tdapnet: Prototype network with recurrent top-down attention for robust object classification under partial occlusion. *arXiv preprint arXiv:1909.03879*. 1

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500. 2, 6

Zhou, X., Karpur, A., Luo, L., and Huang, Q. (2018). Starmap for category-agnostic keypoint and viewpoint estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–334. 6