VehiPose: Multi-Scale Framework for Vehicle Pose Estimation

Divyansh Gupta[a], Bruno Artacho[a], and Andreas Savakis[a]

[a]Rochester Institute of Technology, USA

ABSTRACT

Vehicle pose estimation is useful for applications such as self-driving cars, traffic monitoring, and scene analysis. Recent developments in computer vision and deep learning have achieved significant progress in human pose estimation, but little of this work has been applied to vehicle pose. We propose VehiPose, an efficient architecture for vehicle pose estimation, based on a multi-scale deep learning approach that achieves high accuracy vehicle pose estimation while maintaining manageable network complexity and modularity. The VehiPose architecture combines an encoder-decoder architecture with a waterfall atrous convolution module for multi-scale feature representation. Our approach aims to reduce the loss due to successive pooling layers and preserve the multi-scale contextual and spatial information in the encoder feature representations. The waterfall module generates multiscale features, as it leverages the efficiency of progressive filtering while maintaining wider fields-of-view through the concatenation of multiple features. This multi-scale approach results in a robust vehicle pose estimation architecture that incorporates contextual information across scales and performs the localization of vehicle keypoints in an end-to-end trainable network.

Keywords: Vehicle pose estimation, human pose estimation, CNNs, atrous convolutions

1. INTRODUCTION

Vehicle pose estimation is important in multiple applications but has not been explored much compared to human pose estimation. With the recent advancements in technology for the automotive industry, the demand for accurate vehicle pose estimation[1] has gained popularity due to its applications in autonomous driving,[2] traffic monitoring[3] and scene analysis. Vehicle pose estimation involves locating specific keypoints of a particular vehicle under consideration. This is a challenging task, as there are several types of vehicles with different color, shape, and size.

Convolutional Neural Networks (CNNs) have revolutionized the field of deep learning and have been used to dramatically improve the performance of human pose estimation methods.[4] However, very little of these methods has been utilized in vehicle pose estimation. Human pose estimation is challenging due to high degree of freedom in body joints and high occlusion of those joints, whereas vehicle pose deals with a more rigid structure and has different types of occlusions. The growth of the automobile industry has resulted in high variability within each vehicle class, causing challenges for developing a reliable method for different types of vehicles. Camera viewpoint has more variations in elevation for vehicles. So far, vehicle datasets[5] are annotated for other tasks and there are no defined conventions for pose, making it difficult to find representative keypoints for training and testing deep learning models.

To deal with above challenges and improve on the generalization power of the network, our framework utilizes an encoder-decoder architecture that leverages multi-level features from the backbone (ResNet-101) and processes them with a waterfall module[6] for multi-scale representations. A related version of this configuration, without multi-level features, was beneficial for the tasks of semantic segmentation[6] and human pose estimation7. In this paper, we incorporate multi-level features in the waterfall module and demonstrate the usefulness of our framework for vehicle pose estimation.

Further author information: (Send correspondence to A.S)
A.S.: E-mail: andreas.savakis@rit.edu

Our architecture combines an encoder-decoder network along with larger field of view generated by the waterfall of atrous convolutions. Aiming to achieve better spatial and contextual representations, our multi-scale approach is designed to improve the predicted keypoint accuracy by combining atrous convolutions and low-level feature maps from the encoder network, and integrating them with the decoder module. This approach generates richer image features by concatenating them and avoiding loss of spatial information at different scales. The multi-scale approach, along with successively increasing the Field-of-View (FOV) in a waterfall architecture, helps in predicting the location of keypoints by preserving the contextual and spatial information. Our approach more efficiently incorporates the contextual information across scales and performs keypoint localization in a single stage, end-to-end trainable network. Our results demonstrate that VehiPose is a robust and efficient architecture for vehicle pose estimation. The main contributions of this paper can be summarized as the following :

- We propose the VehiPose framework, a multi-scale, end-to-end trainable, single-stage approach that produces state-of-the-art results for vehicle pose estimation.
- The waterfall framework generates multi-scale feature representations by combining the contextual and spatial information, resulting in larger FOV features for vehicle pose estimation.

2. RELATED WORK

Vehicle pose estimation is a relatively new topic with multiple applications, such as traffic surveillance and autonomous driving. However, there are very few methods for estimating the vehicle pose. There are essentially two main approaches to pose estimation: the top-down approach as shown in Ref. 8, 9 and the bottom-up approach as shown in Ref. 10, 11, and 12. The top-down[13] approaches begin by detecting and localizing objects independently, using a bounding box object detector, such as YOLO[14] or Faster R-CNN.[15] After identifying the total number of instances present in the image, the locations of the keypoints are estimated for every instance. These top-down methods for pose estimation are dependent on precise object detection and suffer if the object detector fails. In a bottom-up approach,[1] all the keypoints in the image are detected first, followed by clustering those keypoints belonging to distinct instances. The bottom-up approaches offer robustness and have the potential to decouple runtime complexity from the total number of instances present in the image.

Stacked hourglass networks[16] were proposed for human pose estimation and have been utilized for vehicle pose estimation in Ref. 17, 18, and 19. These networks consist of multiple stages that are made up of residual convolutional blocks with skip connections in a symmetric design capturing information at every block. The challenge of using an encoder feature generation module is the loss of resolution due to successive pooling layers. To tackle this problem, Fully Convolutional Networks (FCN) network[20] applied upsampling techniques to upsample the image to its input dimensions. Corrales *et al.*[1] explored estimating the 2D vehicle pose in a manner similar to human pose, by proposing a simple baseline method. A ResNet[21] backbone network was utilized along with few deconvolution layers to generate heatmaps corresponding to vehicle keypoints. This approached obtained good results but was limited by the loss of spatial and contextual information of the input image during progressive convolutional layers in the network. Wang *et al.* estimated the vehicle keypoints for the task of vehicle re-identification, improving the performance of their model in distinguishing between similar vehicles.

2.1 Feature Representations with Atrous Convolution, ASPP and Res2Net

Atrous or dilated convolutions are used to increase the size of the receptive field, while maintaining the input size, and avoid the loss of resolution due to downsampling. Yu *et al.* systematically usds dilated convolutions for preserving the contextual information of the input image by proposing a multi-scale context aggregation module.[22]

Further improving the FOV while maintaining the same resolution by using atrous convolutions at larger dilation rates in parallel branches, Deeplab[23] proposed the Atrous Spatial Pyramid Pooling (ASPP) module to increase the receptive field of the network at the same resolution. DeepLab combined four branches with increasing dilation rates for larger FOV to deal with loss of resolution in the encoder module. The main disadvantage of this network was the increased computational cost and memory consumption. Res2Net[24] used

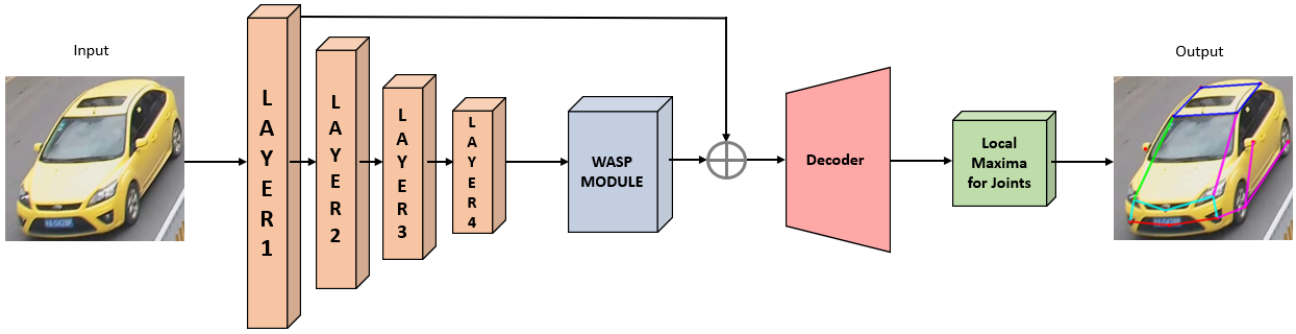

Figure 1. The proposed VehiPose architecture for 2D vehicle pose estimation. The input color image is fed into the ResNet backbone and the last layer features are processed by the WASP module to obtain 304 feature maps after the concatenation of WASP and low level features at $\oplus$. The decoder module generates K heatmaps, one per joint, and the exact location of each keypoint is extracted by applying a local maximum operation.

a multi-scale approach for extracting features by introducing hierarchical connections in a single residual block of the CNN model. The proposed Res2Net block can be plugged into many CNN based models for multi-scale feature extraction.

Improving upon DeepLab and Res2Net, Artacho introduced the waterfall architecture of the WASP module[6] which incorporates multi-scale features of the Res2Net block and the cascade of atrous convolutions from the DeepLab model but without immediately parallelizing the input stream. The WASP module resembles a waterfall flow by progressively extracting the larger FOV from a series of atrous convolutions at different dilation rates and parallelizing the branches of the atrous convolutions. The waterfall architecture was found to be more computationally efficient and produced better results for semantic segmentation[6] and human pose estimation.[7]

3. VEHIPOSE ARCHITECTURE

We propose the Vehipose framework, a unified multi-scale framework which produces state-of-the-art results for vehicle pose estimation without any intermediate supervision or postprocessing. The proposed architecture is shown in Figure 1. The input image is fed in the ResNet backbone, generating 2048 feature maps at the second last layer of the network which are fed into the WASP module. The waterfall of atrous convolutions in the WASP module helps in preserving the spatial and contextual information due to the larger Field-of-View (FOV) and multi-scale feature representation. The WASP module outputs 256 feature maps which are concatenated with 48 low-level feature maps, generated from the first block of the ResNet backbone after applying 1×1 convolution and max-pooling operation to match the dimensions. After concatenation, the 304 feature maps become the input for our decoder module, which converts the feature maps into heatmaps corresponding to the total number of keypoints.

3.1 WASP module

The success of atrous convolutions in the tasks of semantic segmentation[6] and human pose estimation[7] inspired us to include the waterfall of atrous convolutions in our architecture for the task of vehicle pose estimation. The proposed waterfall architecture, along with the decoder module, is shown in Fig. 2. The four branches in WASP have different FOV and are arranged in a waterfall-like fashion. The WASP module goes beyond the cascade approach by combining the streams from all its branches and average pooling of the original input to achieve a multi-scale representation. WASP is designed with the goal of reducing the number of parameters in order to deal with memory constraints and overcome the computational limitation of atrous convolutions.

3.2 Decoder module

Our decoder module converts the 304 feature maps to heatmaps, each corresponding to a joint or keypoint. The output consists of K heatmaps that are used for keypoint localization after performing a local maximum operation in each heatmap.

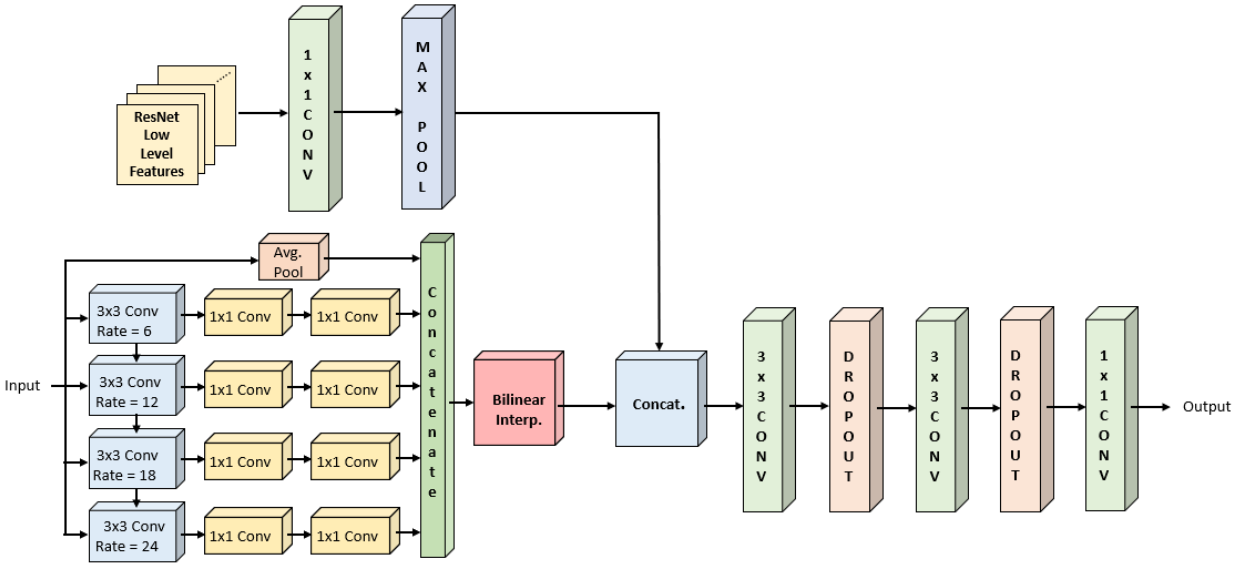

Figure 2. Waterfall module architecture along with the decoder module used in the VehiPose pipeline. The inputs to the decoder are 304 feature maps by concatenating 48 channels of ResNet low-level features and 256 channels of the WASP feature maps. The decoder outputs K heatmaps corresponding to K joints, where K is the total number of keypoints.

Table 1. VeRi-776 dataset keypoint positions.

Index	Location	Index	Location
1	left-front wheel	11	left rear-view mirror
2	left-back wheel	12	right rear-view mirror
3	right-front wheel	13	right-front corner of vehicle top
4	right-back wheel	14	left-front corner of vehicle top
5	right fog lamp	15	left-back corner of vehicle top
6	left fog lamp	16	right-back corner of vehicle top
7	right headlight	17	left rear lamp
8	left headlight	18	right rear lamp
9	front auto logo	19	rear auto logo
10	front license plate	20	rear license plate

4. EXPERIMENTS

4.1 Datasets

We performed experiments on the VeRi-776 dataset[25] composed of single vehicle images. VeRi-776 dataset consists of more that 50,000 images. Each image contains 20 labelled keypoints annotations for a single vehicle. Table 1 presents the details of the keypoint locations. Vehicles are mostly centrally located in images, allowing a good assessment of the network performance for the task of single vehicle pose estimation.

4.2 Evaluation Metric

For the evaluation of VehiPose, we used Percentage of Correct Keypoints (PCK) as the evaluation metric. It considers the prediction to be correct when the keypoint lies within a certain threshold σ from the ground truth location. For Example, $PCK(@0.2) = P(\sigma)/K$, means for a threshold σ of 0.02 and input image of size w $\times$ w,

Figure 3. Vehicle pose estimation examples from the VeRi-776 dataset.

PCK is defined as the number of predicted keypoints (P) that are within the threshold range of $\sigma \times 0.2$ of the ground truth keypoints location divided by the total number of keypoints (k).

4.3 Implementation Details

We considered different rates of dilation on the WASP module and larger rates resulted in better prediction. A set of dilation rates of r = 6, 12, 18, 24 was selected for the WASP module. Training was performed for 100 epochs with a batch size of 16 images. The learning rate was set initially at 10^{-4} and then reduced progressively for best results.

5. RESULTS

We tested VehiPose on VeRi-776 dataset and obtained the results shown in Table 2. We performed a series of experiments to compare the performance of ASPP and WASP modules with our decoder module. We also reported the computational cost and number of parameters of each network to show the computational complexity and memory requirements. The WASP module performs better than ASPP, improving PCK@0.2 results by 1.75% for vehicle pose estimation. In addition, it is computationally more efficient and requires fewer parameters. Examples of VehiPose detections for the VeRi-776 dataset are shown in Fig. 3. These examples illustrate that VehiPose deals effectively with occlusion and vehicles with different color, size, and shape.

Table 2. Results on VeRi-776 dataset using various configurations of the VehiPose framework with a ResNet backbone.

ASPP	WASP	Decoder	PCK@0.2	Params (M)	GFLOPs
-	-	✓	53.15	47.8	35.5
✓	-	✓	54.37	59.3	34.9
-	✓	✓	56.12	47.5	29.2

6. CONCLUSION

We presented the VehiPose architectures for 2D vehicle pose estimation. VehiPose is a single-stage, end-to-end trainable framework that leverages the waterfall multi-scale approach to accurately predict the vehicle keypoints. Our framework shows promise for further use in a broader range of applications, including 3D vehicle pose estimation.

ACKNOWLEDGEMENTS

This research was supported in part by the National Science Foundation grant ♯1749376 and the Air Force Office of Scientific Research (AFOSR) grant FA9550-18-1-0121.

REFERENCES

[1] Sánchez, H. C., Martínez, A. H., Gonzalo, R. I., Parra, N. H., Alonso, I. P., and Fernández-Llorca, D., "Simple baseline for vehicle pose estimation: Experimental validation," *IEEE Access* **8**, 132539–132550 (2020).

[2] Chen, X., Ma, H., Wan, J., Li, B., and Xia, T., "Multi-view 3D object detection network for autonomous driving," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], (2017).

[3] Zhang, S., Wang, C., He, Z., Li, Q., Lin, X., Li, X., Zhang, J., Yang, C., and Li, J., "Vehicle global 6-DoF pose estimation under traffic surveillance camera," *ISPRS Journal of Photogrammetry and Remote Sensing* **159**, 114–128 (2020).

[4] Cao, Z., Simon, T., Wei, S.-E., and Sheikh, Y., "Realtime multi-person 2D pose estimation using part affinity fields," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 1302–1310 (2017).

[5] Geiger, A., Lenz, P., and Urtasun, R., "Are we ready for autonomous driving? the KITTI vision benchmark suite," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 3354–3361 (2012).

[6] Artacho, B. and Savakis, A., "Waterfall atrous spatial pooling architecture for efficient semantic segmentation," *Sensors* **19**, 5361 (Dec. 2019).

[7] Artacho, B. and Savakis, A., "Unipose: Unified human pose estimation in single images and videos," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], (June 2020).

[8] Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., and Sun, J., "Cascaded pyramid network for multi-person pose estimation," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], (June 2018).

[9] Simon, T., Joo, H., Matthews, I., and Sheikh, Y., "Hand keypoint detection in single images using multiview bootstrapping," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], (July 2017).

[10] Xiao, B., Wu, H., and Wei, Y., "Simple baselines for human pose estimation and tracking," *arxiv:1804.06208* (2018).

[11] Kreiss, S., Bertoni, L., and Alahi, A., "Pifpaf: Composite fields for human pose estimation," *arxiv:1903.06593* (2019).

[12] Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T. S., and Zhang, L., "Higherhrnet: Scale-aware representation learning for bottom-up human pose estimation," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], (2020).

[13] Juránek, R., Herout, A., Dubská, M., and Zemcík, P., "Real-time pose estimation piggybacked on object detection," in [*IEEE International Conference on Computer Vision (ICCV)*], 2381–2389 (2015).

[14] Redmon, J. and Farhadi, A., "YOLOv3: an incremental improvement," *arxiv:1804.02767* (2018).

[15] Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards real-time object detection with region proposal networks," *arxiv:1506.01497* (2016).

[16] Newell, A., Yang, K., and Deng, J., "Stacked hourglass networks for human pose estimation," *arxiv:1603.06937* (2016).

[17] Ding, W., Li, S., Zhang, G., Lei, X., and Qian, H., "Vehicle pose and shape estimation through multiple monocular vision," *arxiv:1802.03515* (2018).

[18] Reddy, N. D., Vo, M., and Narasimhan, S. G., "Occlusion-net: 2D/3D occluded keypoint localization using graph networks," in [*IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 7326–7335 (2019).

[19] Pavlakos, G., Zhou, X., Chan, A., Derpanis, K. G., and Daniilidis, K., "6-DoF object pose from semantic keypoints," in [*IEEE International Conference on Robotics and Automation (ICRA)*], 2011–2018 (2017).

[20] Long, J., Shelhamer, E., and Darrell, T., "Fully convolutional networks for semantic segmentation," *arxiv:1411.4038* (2015).

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

[22] Yu, F. and Koltun, V., "Multi-scale context aggregation by dilated convolutions," *arxiv:1511.07122* (2016).

[23] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L., "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**(4), 834–848 (2018).

[24] Gao, S.-H., Cheng, M.-M., Zhao, K., Zhang, X.-Y., Yang, M.-H., and Torr, P., "Res2net: A new multi-scale backbone architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **43**, 652–662 (Feb 2021).

[25] Wang, Z., Tang, L., Liu, X., Yao, Z., Yi, S., Shao, J., Yan, J., Wang, S., Li, H., and Wang, X., "Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification," in [*2017 IEEE International Conference on Computer Vision (ICCV)*], 379–387 (2017).