Early fusion of Dense Optical Flow with Image for Semantic Segmentation in Autonomous Driving

Prashanth Viswanath, Ganesh Sistu, Mihai Ilie, Senthil Yogamani, and Jonathan Horgan

Valeo Vision Systems, Ireland
prashanth.viswanath@valeo.com

Abstract. Precise understanding of the scene around the car is of utmost importance to achieve autonomous driving. Convolutional Neural Networks (CNNs) have been widely used for road scene understanding in the last few years with great success. However, most of these networks have a complex architecture which needs a complex system to be deployed in the car. Typical systems today take the input from cameras placed around the car and the CNNs process them to provide the understanding of the environment. Various hardware manufacturers today are including hardware accelerators in their System on Chips (SoCs) for certain computer vision tasks such as Optical Flow (OF), Stereo Vision (SV) which can achieve good accuracy and fast runtime. If these accelerators can be used in tandem with the CNN to enhance the accuracy of perception, then it is hugely beneficial. In this paper, we explore the possibility of using the Dense Optical Flow output from the hardware accelerator as input along with the image for CNNs to be able to perceive the scene better and faster. We show that by fusion of optical flow and image, mean Intersection over Union (IoU) of segmentation improves by over 1% and accuracy of major classes such as road, person, rider, motorcycle and bicycle improves by 2%, 1%, 5%, 7% and 11% respectively.

Keywords: Convolutional Neural Networks (CNN) · Dense Optical Flow (DOF) · Stereo Vision (SV) · Computer Vision · Autonomous Driving · System on Chip (SoC).

1 Introduction

Object detection and localization around the ego vehicle is of great importance for driver assistance systems and autonomous driving systems. The current trend is to use convolutional neural networks (CNNs) for the scene perception task and provide the locations of various objects around the ego vehicle. CNNs are used for providing semantic information [4] [24], object detection information [19] [8], scene 3D reconstruction [30] and object motion information [27]. Various sensory inputs like camera [4], lidar [32] and radar [9] have been used by CNNs to perceive the environment. Despite these efforts, the accurate delineation of object boundaries remain a challenge.

Most state of the art CNNs assume very high compute and often cannot be deployed in small systems that are present in the cars. There are various restrictions on systems that can be deployed in the car: thermal footprint, memory footprint, placement of the system which impacts how the sensors are connected etc, all of which have a direct impact on the cost these systems. In order to meet the thermal and memory bandwidth constraints, many hardware manufacturers are providing accelerators or fixed processing engines for CNNs, dense optical flow (DOF) and stereo vision (SV) in their Sytem on Chip (SoC) [1] [3] [2]. The typical compute supported by these accelerators are between 1 - 4 Tera-Operations per Second (TOPS) within a power budget of 5W. Given the limited compute available on the SoC, it is critical to have an optimized CNN for the perception task and obtain the best performance. Since DOF and SV engines can be run in tandem on the respective accelerators, it would be very beneficial if CNN can take advantage of the motion and depth cues to improve the perception accuracy. Also, this helps to optimize the network to be smaller and meet the accuracy and run time requirements.

[27] shows that optical flow is very useful in detecting moving objects like vehicles and pedestrians. [17] show that motion boundaries improve semantic segmentation. However, the DOF output undergoes a lot of processing before it is fused with image input. In this paper, we propose to leverage the motion cues by using the DOF outputs from the accelerators with minimal preprocessing before combining it with the image as an input to the CNN, in order to have an optimal and real-time implementation on SoCs that can be deployed in the car. In order to simulate the DOF outputs from hardware accelerators, we use the Opencv Farneback [13] function. The Opencv Farneback function gives a good representation of the DOF algorithm present in the SoCs as most hardware companies benchmark their algorithm against it and generally perform better. We consider different formats of optical flow data such as magnitude only, magnitude and direction, color wheel format etc. concatenated with the RGB channels of the image as input to the CNN and analyze its performance for semantic segmentation task.

The rest of the paper is organized as follows: Section 2 provides information on the related work. Section 3 details the proposed method for incorporating optical flow input in segmentation task. Section 4 shows the experimental results and discussions. Finally, section 5 provides concluding remarks.

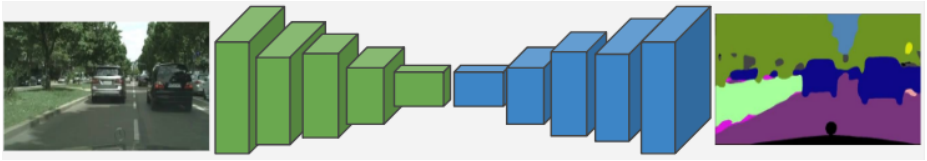


Fig. 1: Typical encoder-decoder architecture of CNN based semantic segmentation network.

2 Related Work

Semantic Segmentation: [20] were the first to propose an end-to-end CNN for semantic segmentation. They modified the last layers of the CNN, thus producing fully convolutional neural network (FCN). Due to the large receptive fields of FCNs, the localization of object boundaries is insufficiently precise. In order to overcome this, many solutions were proposed such as applying a fully connected conditional random fields (CRFs) to the output of the CNN [10] or introducing global energy model along with boundary cues [6]. These post processing steps require additional parameter tuning and compute time. [4] proposed an encoder-decoder based architecture which requires one fourth of memory usage and about half the inference time compared to FCNs, making it an ideal architecture for efficient segmentation. Figure 1 shows the encoder-decoder type architecture for semantic segmentation. The encoder extracts features from the image which is then decoded to produce the semantic segmentation output. ImageNet [12] pre-trained networks such as VGG16 [28], Resnet [16] are typically used as encoder. In early architectures [4] [26], decoder was a mirror image of encoder and had the same complexity. Newer architectures use a relatively smaller decoder. There can also be additional connections from encoder to decoder. For example, Segnet [4] passes max-pooling indices and U-Net [26] passes intermediate feature maps to decoder as well.

Motion Estimation: Optical flow is an important step in deriving motion boundaries. [17] use motion boundaries along with images to improve semantic segmentation. The motion boundaries are computed based on a learning based prediction proposed in [31]. This post processing of optical flow to obtain motion boundaries involves additional computation and memory usage, unlikely to be available on SoCs that are deployed in the car. [27] consider two stream approach where they have separate encoders to extract features from the image channels and the DOF channels and concatenate these features. This results in duplicating the encoder network which hugely impact the size and run-time of the network. Also, the optical flow input is obtained from Flownet [14] type CNN which outputs color wheel representation of the optical flow, which requires additional processing to generate it. [23] derived motion boundaries from the gradient of optical flow computed by traditional computer vision approach and concentrated on motion of only single object in the scene, which is typically not the case in an autonomous driving system. [6] also show that motion boundaries can be leveraged to improve semantic segmentation. However, the motion boundaries are used as additional modality in a late fusion post processing step, which increases the computation and complexity of the system, similar to [27]. [27] and [17] use KITTI [15] and CamVid [7] dataset respectively, which have very few images with segmentation annotations. [27] uses a total of 1950 frames from KITTI raw dataset [15] and [17] uses only 367 images for training and 233 images for testing from the CamVid [7] dataset.

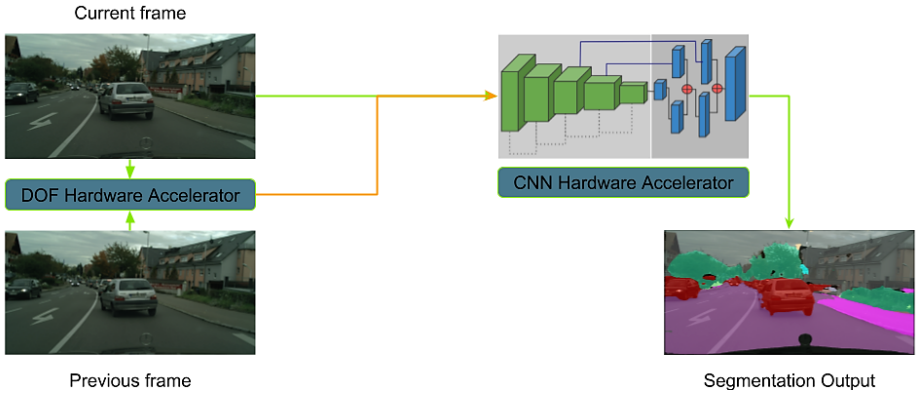
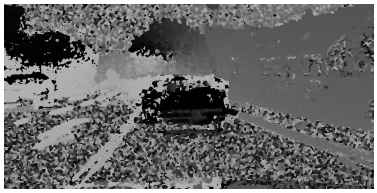


Fig. 2: Overview of the pipeline used in our approach.

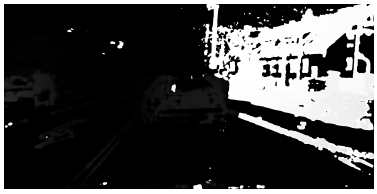
(a) Sample image of normalized magnitude of flow vectors.



(b) Sample image of angle of flow vectors computed.

(c) Sample image of color-wheel representation of flow vectors.

(d) Sample image of scaled magnitude representation of flow vectors.



(e) Sample image of flow vector in direction x cast into 8-bit.

(f) Sample image of flow vector in direction y cast into 8-bit.

Fig. 3: Different formats of DOF inputs considered.

3 Proposed Method

In this section, details of our approach are provided. Figure 2 shows the block diagram explaining the pipeline of our approach. The DOF output is obtained using the current and previous frame. The DOF output is then concatenated with the current frame as additional channels before they are input to the CNN. There are multiple methods of representing the DOF data as discussed in Section 3.1. The most important aspect of concatenating optical flow with image is the normalization of the optical flow data such that the value of the flow vectors are in the same range as that of image pixels. The most effective representation which provides optimal run-time and improved segmentation performance is determined by various experiments as discussed in Section 4. We propose a method of scaling the flow vectors by a fixed constant in order to reduce the amount of additional processing requirements and still improve semantic segmentation performance.

3.1 Dense Optical Flow data

The DOF data is computed using the Opencv Farneback DOF algorithm [13]. The default settings are considered to generate the flow output. The Farneback algorithm outputs 32-bit floating point flow vectors in x and y direction. From this, different formats of DOF inputs for CNN were computed which are as follows:

- Normalized magnitude: Magnitude is computed from the dx,dy flow vectors and normalized in the range 0-255 8-bit unsigned integer format to be in the same range as image channel input as shown in Figure 3a.
- Angle: Angle of direction is computed from dx,dy flow vectors and represented in degrees in range 0-180 8-bit unsigned integer format as shown in Figure 3b.
- Color wheel format: The flow vectors are represented in the color wheel format similar to Middlebury dataset [5] where the color represents the direction of the flow and intensity of color represents the magnitude of the flow as shown in Figure 3c.
- dx, dy: The flow vectors in each direction dx, dy typecast to 8-bit unsigned integer format as shown in Figure 3e and Figure 3f.
- Scaled magnitude: Magnitude is computed from the dx,dy flow vectors and scaled by a fixed number (255) uniformly as shown in Figure 3d. This is done in order to simplify the preprocessing step.

All the above formats of DOF output were considered for fusion with image channels to evaluate the performance of semantic segmentation, which are discussed in Section 4.

3.2 Segmentation

An encoder-decoder type architecture similar to MultiNet [29] is used for the segmentation task. The encoder is Resnet10 [16] architecture and the decoder is

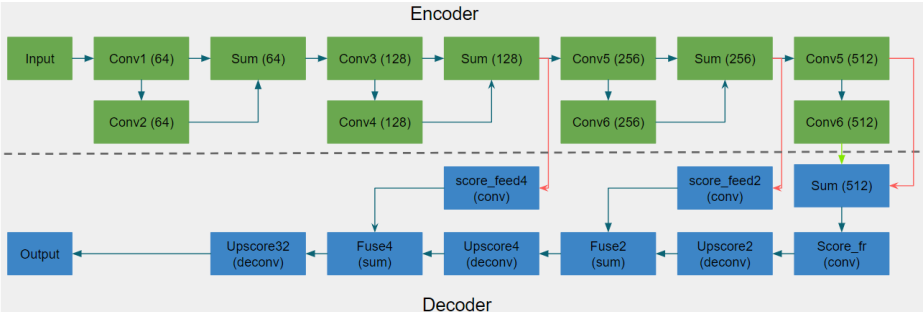


Fig. 4: Network architecture diagram.

a cut down version of FCN8 [20] architecture, with only three upsample layers similar to the MultiNet [29] architecture. The encoder and decoder is combined similar to the MultiNet architecture, where the intermediate layers from the encoder are connected to the decoder using skip connections. The network architecture is as shown in Figure 4. Different inputs for the motion stream along with image are considered with the same network architecture. Pixel-wise cross entropy loss is used for the network. In order to compensate for the low representation of certain classes, other loss functions such as median frequency based weighted cross entropy loss function [4] and alpha focal loss [18] were also tried.

4 Experiments and Results

In this section, the experimental setup and the results of various experiments are detailed.

4.1 Dataset

The proposed framework is trained and tested on the challenging Cityscapes dataset [11]. Although there exists other motion segmentation datasets such as [15] [7] [25] [21] [22], they are either synthetic [21], relatively small [15] [7] [22] or has limited camera motion [25] unlike what is present in autonomous driving scenes. The Cityscapes dataset [11] provides 5000 images with fine pixel-wise annotations, along with the sequence of images which can be used to compute DOF data. Out of 5000 images, 2975 images are used for training and 500 images are used for evaluation. The results presented by the various experiments are based on the evaluation set. For computing DOF, only two frames (current and previous frames) were considered in order to mimic the actual hardware accelerator setup.

4.2 Experimental Setup

For all the experiments, the network architecture is kept same as shown in Figure 4. A baseline with the network configuration using image only input is obtained

Table 1: Semantic Segmentation Results on Cityscapes. global avg accuracy, precision, recall, F1-score, mean IoU, and per-class accuracy is shown. Only 7 classes of the 20 are shown due to limited space.

	Avg Accuracy	Precision	Recall	F1-score	Mean IoU	Per-Class Accuracy						
						Road	Sidewalk	Person	Rider	Car	Motorcycle	Bicycle
Image (baseline)	84.91	87.16	84.91	85.24	39.59	95.01	72.16	65.60	6.65	92.05	8.50	46.25
NormMag+Image	85.10	**87.72**	85.11	85.63	39.65	96.12	67.60	62.40	11.32	**92.10**	14.49	58.20
NormMag+Ang+Image	84.83	87.02	84.83	85.08	39.74	94.87	73.23	63.57	4.89	90.40	9.51	52.84
Colorflow+Image	84.67	86.31	84.67	84.75	39.56	94.98	72.23	63.53	3.19	90.57	9.51	56.41
dx+dy+Image	85.03	87.40	85.03	85.42	39.18	95.65	69.75	58.60	2.82	89.87	5.10	48.93
ScaledMag+Image	**85.33**	87.48	**85.33**	**85.68**	**40.79**	**97.05**	67.32	66.75	11.59	91.32	14.19	57.34
Image(WCEL)	73.78	70.91	73.78	69.57	32.83	85.95	68.36	62.68	**54.54**	79.98	**47.92**	63.34
NormMag+Image(WCEL)	73.95	72.00	73.95	69.89	33.63	82.30	**75.21**	**71.98**	40.78	81.31	41.67	**71.47**

first. Adam optimizer is used with a learning rate of $5e^{-5}$. No decay of learning rate is used during the training and L2 regularization is used while training. The network is trained for a maximum of 30 epochs with early stopping based on validation loss with a patience of 5 enabled.The encoder is initialized with the Resnet pretrained weights on Imagenet [12] and the transposed convolution layers of the decoder are initialized to bilinear upsampling, while training the network with image only input. For the network structure with additional channel input from optical flow data, the pretrained weights from the network trained using image only input, are used. The image resolution is 1024x512.

The evaluation metrics used in the segmentation are global average accuracy, precision, recall, F1-score and mean intersection over union (IoU). The individual class accuracies are also evaluated based on the confusion matrix results.

Table 2: Semantic Segmentation Results on Cityscapes. Per-Class IoU for the 7 classes.

	Per-Class IoU						
	Road	Sidewalk	Person	Rider	Car	Motorcycle	Bicycle
Image (baseline)	88.04	54.99	42.46	5.96	75.30	7.56	39.65
NormMag+Image	88.09	**55.01**	42.44	9.32	75.98	**10.74**	43.03
NormMag+Ang+Image	87.73	54.16	41.39	4.53	76.47	8.11	40.77
Colorflow+Image	87.90	54.84	40.93	3.03	76.42	7.95	41.84
dx+dy+Image	**88.23**	54.70	41.29	2.70	76.77	4.76	38.52
ScaledMag+Image	87.96	53.96	**44.18**	9.65	**76.96**	6.44	**43.56**
Image(WCEL)	80.97	42.90	36.37	13.74	65.84	6.13	34.50
NormMag+Image(WCEL)	78.85	40.48	35.25	**14.91**	67.86	7.14	32.55

4.3 Results and Analysis

Table 1 and Table 2 shows the evaluation results of various experiments performed on the Cityscapes dataset. It clearly shows that using optical flow along with image channels improves the average accuracy and mean Intersection over

(a) Result with image only as input.

(b) Result with image and norm magnitude as input.

(c) Result with image and color format flow as input.

(d) Result with image and norm magnitude and angle flow as input.

(e) Result with image and dx,dy flow as input.

(f) Result with image and fixed scaled magnitude as input.

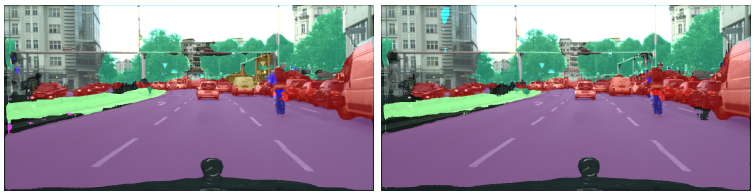
(g) Result with image only as input using weighted Cross Entropy loss.

(h) Result with image and norm magnitude as input using weighted Cross Entropy loss.

(i) Ground truth result of segmentation overlay on the image.

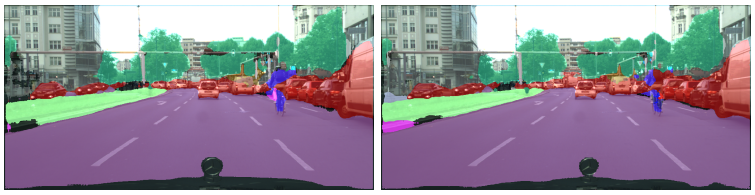
(j) Ground truth result of segmentation of the image.

Fig. 5: Sample results from various experiments on Lindau sequence of Cityscapes dataset .



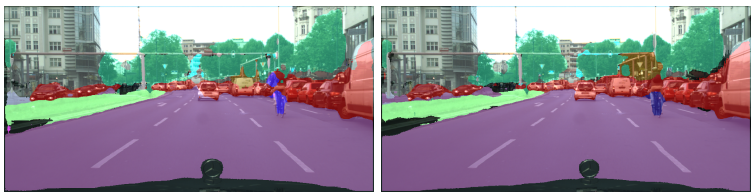
(a) Result with image only as input.

(b) Result with image and norm magnitude as input.



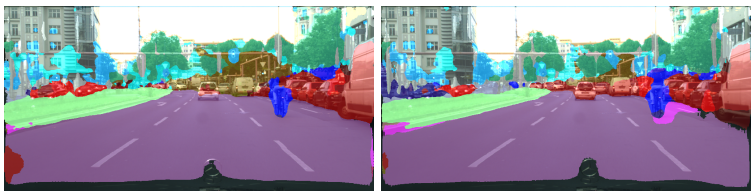
(c) Result with image and color format flow as input.

(d) Result with image and norm magnitude and angle flow as input.



(e) Result with image and dx,dy flow as input.

(f) Result with image and fixed scaled magnitude as input.



(g) Sample result with image only as input using weighted Cross Entropy loss.

(h) Sample result with image and norm magnitude as input using weighted Cross Entropy loss.

Fig. 6: Sample results from various experiments on Berlin sequence of Cityscapes dataset.

Union (mean IoU). A closer look at other metrics shows that using the normalized magnitude as shown in Figure 3a provides the best precision, with significant increase in class accuracies for road, rider, car, motorcylcle and bicyle, but decreasing class accuracies for persons and sidewalk. The flow vectors for person and sidewalk is very less and hence when normalized, it is close to zero. The mean IoU is less compared to the state of the art. This is due to two factors:

- The small size of the network used to obtain real-time performance
- The under represented classes such as pole, wall, fence, truck, bus, train, motorcycle, rider, traffic sign and traffic light classes of the Cityscapes dataset

The per-class IoU improves significantly for all moving objects such as persons, riders, cars and bicycle as shown in Table 2. Computing the normalized magnitude involves significant amount of preprocessing. First, the distribution of the flow in an image has to be computed and then remapped to 0-255 range by multiplying each flow with a different scaling factor. In order to reduce the amount of preprocessing, a simple fixed scaling of magnitude was implemented where the magnitude of each flow vector was multiplied by 255 which is as shown in Figure 3d and the results are as shown in row 6 of Table 1. As it can be seen, the overall metrics are improved further. The accuracy for person is also improved due to the scaling, as compared to the normalized magnitude approach. The proposed scaling approach scales any flow vector greater than 0 to 255, thereby removing the importance of flow vectors for objects that are moving faster, essentially converting it into a binary image. The scaling factor can be adjusted to maintain the importance of fast moving objects and can even be a learned parameter. Figure 5 and Figure 6 shows sample results of segmentation considering various formats of input to the network. One interesting observation from the results is the improvement in accuracy in the segmentation of road class, which is counter intuitive. This is because the optical flow is inaccurate on the road surface and hence typically made invalid or void for those regions, thus helping the CNN to classify the road class better.

Experiments with different loss functions such as weighted cross entropy and alpha focal loss were tried in order to improve the segmentation of classes such as rider, motorcycle and bicycle which are under represented. Row 7 and 8 of Table 1 shows the results of the network trained with image and image + normalized magnitude of optical flow input respectively, using weighted cross entropy loss. Median frequency [4] of classes were used to weight the loss function accordingly. It clearly shows that the under represented classes such as rider, motorcycle and bicycle hugely improve. However, the other classes such as road, pedestrian and car which have good representation in the dataset suffer as they are weighted less. Along the same lines, alpha focal loss [18] was also tried, but no improvement was observed.

5 Conclusion

In this paper, we explored combining dense optical flow data in various formats along with image to improve semantic segmentation. We have shown that by

combining normalized magnitude of optical flow with image, the accuracy for segmenting moving objects and road improves a lot. We also present a simpler method to scale magnitude of optical flow and combining it with image, thereby reducing the amount of preprocessing needed and still improve the segmentation results. Furthermore, we can deduce the scaling parameter by a learning approach. DOF and CNN accelerators are present in several SoCs and hence we have provided analysis on how best to utilize them in the SoC.

References

1. Movidius myraid x vpu
2. Product specifications of the r-car v3h
3. S32v234: 64-bit multi-core a53 processor for vision and adas applications
4. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)
5. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision **92**(1), 1–31 (Mar 2011). https://doi.org/10.1007/s11263-010-0390-2, https://doi.org/10.1007/s11263-010-0390-2
6. Bertasius, G., Shi, J., Torresani, L.: Semantic segmentation with boundary neural fields. CoRR **abs/1511.02674** (2015), http://arxiv.org/abs/1511.02674
7. Brostow, G.J., Fauqueur, J., Cipolla, R.: Semantic object classes in video: A high-definition ground truth database. Pattern Recogn. Lett. **30**(2), 88–97 (Jan 2009). https://doi.org/10.1016/j.patrec.2008.04.005, http://dx.doi.org/10.1016/j.patrec.2008.04.005
8. Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: European Conference on Computer Vision. pp. 354–370. Springer (2016)
9. Capobianco, S., Facheris, L., Cuccoli, F., Marinai, S.: Vehicle classification based on convolutional networks applied to fm-cw radar signals. arXiv preprint arXiv:1710.05718v3 (2017)
10. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:1606.00915 (2016)
11. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
12. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
13. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Bigun, J., Gustavsson, T. (eds.) Image Analysis. pp. 363–370. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
14. Fischer, P., Dosovitskiy, A., Ilg, E., Häusser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. CoRR **abs/1504.06852** (2015), http://arxiv.org/abs/1504.06852
15. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR) (2013)

16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
17. Huang, Y.H., Oramas, J., Tuytelaars, T., Gool, L.V., Leuven, K.D.V.K.U.: Do motion boundaries improve semantic segmentation ? (2016)
18. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. CoRR **abs/1708.02002** (2017), http://arxiv.org/abs/1708.02002
19. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
20. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440 (2015)
21. Mayer, N., Ilg, E., Häusser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. CoRR **abs/1512.02134** (2015), http://arxiv.org/abs/1512.02134
22. Ochs, P., Malik, J., Brox, T.: Segmentation of moving objects by long term video analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(6), 1187 – 1200 (Jun 2014), http://lmb.informatik.uni-freiburg.de/Publications/2014/OB14b, preprint
23. Papazoglou, A., Ferrari, V.: Fast object segmentation in unconstrained video. In: Proceedings of the 2013 IEEE International Conference on Computer Vision. pp. 1777–1784. ICCV '13, IEEE Computer Society, Washington, DC, USA (2013). https://doi.org/10.1109/ICCV.2013.223, http://dx.doi.org/10.1109/ICCV.2013.223
24. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147 (2016)
25. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: Computer Vision and Pattern Recognition (2016)
26. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. CoRR **abs/1505.04597** (2015), http://arxiv.org/abs/1505.04597
27. Siam, M., Mahgoub, H., Zahran, M., Yogamani, S., Jagersand, M.: Modnet: Moving object detection network with motion and appearance for autonomous driving. arXiv preprint arXiv:1709.04821v2 (2017)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014), http://arxiv.org/abs/1409.1556
29. Teichmann, M., Weber, M., Zöllner, J.M., Cipolla, R., Urtasun, R.: Multinet: Real-time joint semantic reasoning for autonomous driving. CoRR **abs/1612.07695** (2016), http://arxiv.org/abs/1612.07695
30. Usenko, V., Engel, J., Stuckler, J., Cremers, D.: Reconstructing street-scenes in real-time from a driving car. In: International Conference on 3D Vision. IEEE (2015)
31. Weinzaepfel, P., Revaud, J., Harchaoui, Z., Schmid, C.: Learning to detect motion boundaries. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2578–2586 (2015)
32. Zelener, A., Stamos, I.: Cnn-based object segmentation in urban lidar with missing points. In: Fourth International Conference on 3D Vision. IEEE (2016)