No number

Semantic Segmentation with High Inference Speed in Off-Road Environments

Bradley Selee, Max Faykus, Melissa Smith
Clemson University, South Carolina

Copyright © 2022 Society of Automotive Engineers, Inc.

1 ABSTRACT

This work focuses on evaluating the inference speed of semantic segmentation on off-road environments using a state-of-the-art deep learning model, SwiftNet. Semantic segmentation is an integral component in many autonomous vehicle systems used for tasks like path identification and scene understanding. Autonomous vehicles must make decisions quickly enough so they can react to their surroundings, therefore, they must be able to segment the environment at high speeds. There has been a fair amount of research on semantic segmentation, but most of this research focuses on achieving higher accuracy, using the mean intersection over union (mIoU) metric rather than higher inference speed. More so, most of these semantic segmentation models are trained and evaluated on urban areas instead of off-road environments. Because of this there is a lack of knowledge in semantic segmentation models for use in off-road unmanned ground vehicles. In this research, SwiftNet was implemented, a semantic segmentation deep learning model designed for high inference speed and accuracy on images with large dimensions. SwiftNet was pre-trained on the ImageNet dataset, then trained on 70% of the labeled images from the Rellis-3D dataset. Rellis-3D is an extensive off-road dataset designed for semantic segmentation, containing 6234 labeled 1920x1200 images. SwiftNet was evaluated using the remaining 30% of images from the Rellis-3D dataset and achieved an average inference speed of 24 frames per second (FPS) and an mIoU score 73.8% on a Titan RTX GPU.

2 INTRODUCTION

Ever since the rise of artificial intelligence (AI), state-of-the-art computer vision algorithms have greatly favored deep learning (DL) techniques [1]. These recent improvements in DL, such as object detection [2] and image segmentation [3], have enabled the advancements made in autonomous vehicles. Current advanced driver assistance systems (ADAS) support low-level autonomy like lane assists and blind spot alerting [4]. These ADAS operate at what is considered level 1 and level 2 autonomy because drivers are still required to perform some actions. However, research is now focused on advancing higher levels of autonomy: level 3 (conditional automation), level 4 (high automation), and, ultimately, level 5 (full automation) [5]. Perception algorithms play a vital role in these autonomous systems because they must quickly understand the vehicles surroundings in order to make accurate decisions. Two major types of image classification exist in autonomous vehicle systems, object detection and semantic image segmentation. Object detection can be defined as classifying objects of interest and identifying their locations in an image using bounding boxes [6], while semantic segmentation classifies every pixel in an image into a defined set of class labels [7, 8]. Compared to object detection, semantic segmentation provides a better understanding of the environment but at a significant computational overhead cost [9, 10]. Most modern semantic segmentation algorithms utilize deep neural networks (DNN) and many DNNs adopt the encoder-decoder structure [11, 3]. The encoder consists of a feature extractor, like ResNet [10], which uses convolutional layers to downsample the original image into a feature map. After extracting meaningful features, the decoder upsamples the feature map back to its original image size, creating a semantic map of class probabilities [8].

Typically, more accurate semantic segmentation models require longer inference time. Likewise, models with faster inference speed suffer in accuracy loss [12]. This makes perception a challenge in autonomous vehicles. A vision

system must have acceptable accuracy in order to make correct decisions but at the same time needs to make these decisions in real-time [4]. In recent years, research has made great progress developing DNN architectures that speed up semantic segmentation inference time while maintaining an acceptable level of accuracy [12, 13, 14]. Many of these fast inference models were designed and trained for urban environments, most notably for use in self-driving cars [15]. Additionally, the primary benchmark dataset used for training and evaluation was the Cityscapes dataset [16], which contains thousands of semantically labeled images (2048x1024) from urban environments.

Unfortunately, there is a lack of research evaluating semantic segmentation in the off-road setting; this can be seen from the absence of research investigating this domain and the scarcity of labeled off-road datasets [17, 18]. Autonomous off-road driving covers a wide range of applications like exploration, rescue, and military use. Just like in self-driving cars, perception systems are critical for autonomy in unmanned ground vehicles (UGVs) and other off-road vehicles [19]. The evaluation of semantic segmentation DNNs in off-road environments is a necessity to integrate higher levels of autonomy in UGVs. Furthermore, there are two key features which can make off-road semantic segmentation more challenging than urban settings [20]. Firstly, the environments are unstructured by nature, which makes identifying traversable areas more difficult. Next, the environments are very noisy due to the type of objects. This can lead to poor generalization of a deep learning model. On the other hand, due to the application of most UGVs, it may be acceptable for semantic segmentation to be less accurate in off-road environments. Because UGVs are typically very large and durable, they can drive over smaller objects. Therefore, detecting obstacles like logs and bushes can be less important, while focusing on faster inference speed could be considered more important. In this research, SwiftNet [13] was implemented, a semantic segmentation deep learning model designed for high inference speed and accuracy on images with large dimensions. The model was trained and evaluated on the Rellis-3D dataset [18, 21]. Rellis-3D is an extensive off-road dataset designed for semantic segmentation, containing 6234 labeled 1920x1200 images.

3 RELATED WORK

The trade off between accuracy and inference speed in semantic segmentation has been realized for quite a while now [10, 4, 21]. Several efforts have been made to speed up semantic segmentation inference while reducing the accuracy as little as possible. Although, once again, not much research evaluates inference speed and accuracy in an off-road setting.

High inference speed semantic segmentation DNNs: One of the first successful attempts at high inference speed and maintaining an acceptable level of accuracy was ICNet, accomplished by creating a multi-branch architecture [12]. First, the original image is downsampled by a factor of 2 and 4, then each image size is passed to a branch of the network with the smallest image going through the most amount of convolutional layers, and the larger images going through the least amount of convolutional layers. Finally, each branch is fused together by a cascade module. BiSeNet [14] is another multi-branch attempt to utilize larger and smaller image sizes on different branches. BiSeNet has two paths. The first path uses a lightweight model and is designed to grab context information and the second path has only three convolutional layers which extract spatial information. A feature fusion module is used to merge the extracted features from both paths. Several other attempts exist which experiment with similar, but different, architectures to retain representative image features with less expensive convolutional operations [11]. While many of these models are effective, they are only evaluated on the primary benchmark dataset Cityscapes, and secondary datasets like, CamVid (another city-like environment), and COCO-stuff (common objects in context) [22, 23]. Most models were not evaluated in an unstructured, off-road environment.

Off-road environment semantic segmentation: Even though many semantic segmentation models do not evaluate performance on off-road datasets, there still exists research which investigates semantic segmentation in the off-road setting with datasets like RUGD and Rellis-3D [17, 18]. Even with the existence of off-road datasets, there is still the issue of class imbalance and irregular objects. To overcome these problems, OFFSEG [19] strives to create a two stage framework for rich semantic segmentation results. The first stage condenses the ontology into four classes (sky, obstacles, traversable regions, and non-traversable regions), then trains a BiSeNet model on this reduced ontology. In the second stage, the regions of interest are extracted from the segmented image, and then further sub-classes (grass, mud, puddle, etc.) are created using k-means clustering [24] from the regions of interest. Finally, the segmented sub-classes are appended to the first segmented image. This results in a detailed semantic image of class labels. Another approach explores transfer learning to alleviate the lack of off-road datasets [25, 26]. In this approach, a lightweight model based on VGG-16 [27] is trained using synthetic data as the intermediate domain. Then, the synthetically pre-trained modeled is trained with a real off-road dataset called the Freiburg Forest dataset [28]. Unfortunately, the accuracy of the model pre-trained with synthetic data achieved about the same accuracy as the model without pre-trained synthetic data.

This research evaluates a state-of-the-art semantic segmentation DNN, SwiftNet, on a large off-road dataset, Rellis-3D. The primary goal of this study is to investigate the inference speed of SwiftNet on off-road images while maintaining an acceptable level of accuracy. Because this perception system is for autonomy in large UGVs,

(a) (b) (c)

Figure 1: (a) Raw image from the Rellis-3D dataset (b) Ground truth label image with the color map applied (c) Ground truth label image of class IDs

accuracy may not need to be as high as in perception systems for self-driving cars. This manuscript first covers the methodology such as software and hardware used, SwiftNet implementation and replication, then training and evaluating protocols. Next, the accuracy and inference speed of SwiftNet are analyzed and interpreted, future work is also briefly explored. Lastly, the overall goal, outcome, and limitations on the experiment are concluded.

4 METHODOLOGY

This section covers the methods used to design, implement, run, and evaluate the experiment. An explanation of the SwiftNet architecture and theory is included.

4.1 SWIFTNET ARCHITECTURE SwiftNet follows two common methodologies in architecture design. The first is pre-training the image encoder on the ImageNet dataset [29] in order to benefit from the knowledge learned. The second is the encoder-decoder structure. The encoder is built from lightweight feature extractors, ResNet-18 and Mobile Net V2 [30]. The decoder, which upsamples the rich feature map to the original input size, consists of a ladder-style structure with two inputs: the low-resolution feature maps from the preceding upsampling module, and the high-resolution feature maps from the corresponding encoder block. Finally, the upsampled input features and adjacent encoder features are combined with elementwise summation and fused with a 3x3 convolution. The SwiftNet implementation has two architectures, a single-scale architecture and a multi-scale architecture with pyramidal fusion [13]. The multi-scale architecture extracts features at three different image sizes using three different branches. Each of the four encoder blocks share the same parameters between branches. SwiftNet also uses a unique loss function called the boundary-aware loss [31]. This emphasises the importance of pixels near semantic boundaries in hopes of avoiding overfitting in small objects [32]. In this work, the multi-scale architecture based on the ResNet-18 backbone is opted for due to the quality of their results.

4.2 RELLIS-3D DATASET Rellis-3D is an extensive off-road dataset designed for semantic segmentation, containing 6234 labeled 1920x1200 images [18]. Figure 1 shows an example frame with all three image types. The raw camera image, the colored ground truth label image for visualization purposes, and the grayscale ground truth label IDs image. For this research, 70% of the data was used for the train set and the remaining 30% was used for the test set. Figure 2 shows the ontology of the Rellis-3D dataset. There are twenty class labels which consist of traversable areas like dirt, grass, asphalt, rubble and obstacles like bushes, trees, objects and poles. Because the dataset was recorded after a rainy day, there are very few dirt labels so this label was excluded in the study.

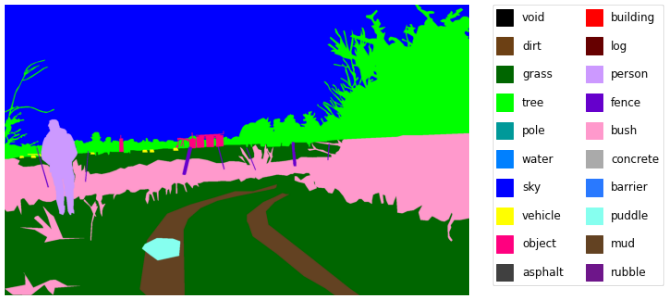

Figure 2: Colored ground truth image with class labels

4.3 IMPLEMENTATION Once SwiftNet was established for Cityscapes, the model was adjusted to run with the Rellis-3D dataset. Most notably the number of output channels were changed to match the number of classes in Rellis-3D, a custom dataset class for Rellis-3D was created, and the mean and standard deviation of Rellis-3D were calculated. The Cityscapes and Rellis-3D dataset follow the same training and evaluation protocol on a single Titan RTX GPU.

The code for this study was initially derived from the original SwiftNet repository found here: https://github.com/orsic/swiftnet. The modified code for this study can be found here:

https://github.com/cufctl/segmentation.

4.4 TRAINING The same training protocol as in [13] was followed. In detail, SwiftNet uses the Adam optimizer [33], the boundary-aware loss function, an initial learning rate of $4 \cdot 10^{-4}$ which decays with cosine annealing at a decay rate of 10^{-4}. Each image in the train set is augmented with random horizontal flipping [29], random scaling between 0.5-2.0, and random cropping of size 768x768. Finally, each image is standardized using the mean and standard deviation of the dataset. The batch size is set to 14 and the model is trained for 250 epochs. During training, the model saves the weights from the epoch with the highest accuracy. The weights with the highest accuracy are then used for evaluation.

4.5 EVALUATION For model evaluation, the Rellis-3D test set was used which contains 1,870 images. The model inference speed was the primary metric measured and the model accuracy was a secondary metric. When describing accuracy of a semantic segmentation model, the primary metric used is the Jaccard Index [34], also known as the intersection over union (IoU), shown in equation 1. The intersection and union rely on the true positive (TP), false positive (FP), and false negative (FN) values. Pixels labeled as void do not contribute to the score.

$$IoU_{class} = \frac{TP}{TP + FP + FN} \tag{1}$$

To describe the overall IoU, the mean IoU is used (mIoU) which is an average of all the class IoU scores. For timing the inference speed in PyTorch, a similar approach as in [13] was followed. Algorithm 1 shows the timing method assuming a batch size of 1 in frames per second (FPS).

Algorithm 1 Proposed timing approach in PyTorch

```
device = torch.device('cuda')
n = len(dataset_loader)
model.eval()
model.to(device)
torch.cuda.synchronize()
start_t = perf_counter()
with torch.inference_mode():
    for data, target in dataset_loader:
        data, target = data.to(device)
        outputs = model(data)
        _, pred = torch.max(outputs, 1)
        pred = pred.byte().cpu()
    torch.cuda.synchronize()
    end_t = perf_counter()
    FPS = n / (end_t - start_t)
```

During evaluation, standardization is the only image preprocessing done. The batch size is set to 1 to replicate a real-time processing system, which would segment one frame at a time.

5 RESULTS AND DISCUSSION

The following section presents unique results for the inference speed and mIoU on the Rellis-3D dataset. Although this research focuses on off-road environments, Cityscapes results will be shown and compared to the original SwiftNet paper [13]. This can serve as replication and verification of results.

5.1 RELLIS-3D Using SwiftNet's multi-scale architecture on the Rellis-3D dataset, an average inference speed of 24 FPS with an mIoU of 77.9% was achieved. These results are encouraging due to the difficulty of segmenting unstructured terrain. Similarly, in the IoU class breakdown, Table 1, traversable and non-traversable regions obtain higher accuracy than smaller objects. Autonomous UGVs are interested in traversing off-road terrain and more likely to drive over small objects like logs and poles rather than avoid them. In this scenario, higher inference speed is favored over higher accuracy. Figure 3 shows a SwiftNet segmented image from the Rellis-3D dataset.

Class	IoU (%)
grass	91.83
tree	90.04
pole	42.15
water	81.48
sky	97.54
vehicle	67.30
object	72.73
asphalt	86.08
building	65.08
log	61.79
person	92.52
fence	65.61
bush	85.09
concrete	91.24
barrier	87.63
puddle	80.96
mud	65.46
rubble	77.87

Table 1: Individual class IoU from the Rellis-3D dataset

The inference speed achieved on Rellis-3D is lower by about 10 FPS than what was achieved in the original SwiftNet study on the Cityscapes dataset; this achieved an inference speed of 34 FPS on a GTX 1080Ti GPU with the multi-scale architecture. There are several reasons this study achieved lower than expected inference speed compared to the original study. First, the Rellis-3D images are 1920x1200 which is a total of $1920 * 1200 = 2304000$ pixels. The Cityscapes images are 2048x1024 which is a total of $2048 * 1024 = 2097152$, making the Rellis-3D

images more computationally expensive. However, in this study, when timing SwiftNet on the Cityscapes dataset there is minimal difference in inference speed between datasets. Second, the inference speed is timed at the start of the forward pass after the model and tensor is transferred to the GPU, and the timing ends after the predicted classes for each pixel is transferred back to the CPU. If the timing ends after the predicted classes but before the GPU to CPU transfer, the inference speed increases to 30 FPS. Even though this study obtained a lower inference speed than the original SwiftNet work, it still obtains high inference speed and can be a very viable DNN in perception systems for autonomous UGVs. In terms of accuracy, this study achieved a higher mIoU score than the original SwiftNet work on the Cityscapes dataset, which is summarized in Table 3. Two major reasons for this could be that Cityscapes uses 19 labels while Rellis-3D only uses 18, and Rellis-3D has more images in its train set than Cityscapes.

5.2 CITYSCAPES In this research, the Cityscapes dataset was used mainly to implement SwiftNet and attempt to replicate the original research. The focus remains on off-road environments. Thus, this section is only briefly covered. The Cityscapes dataset contains 5000 images for semantic scene understanding in urban environments[16]. It contains a train set of 2975 images, a validation of set 500 images, and a test of 1525 images. The test set labels are private and an mIoU score on the test set can only be obtained by submitting a model to the evaluation server. For this reason, the current study evaluates the trained SwiftNet model on the validation set. In this study, SwiftNet attains 26.9 FPS with an mIoU of 73.5%. Like Rellis-3D, Cityscapes produces slightly lower FPS than the original study. Additionally, the mIoU appears slightly lower than the original, which may be attributed to the stochastic nature of the neural network training process. Table 2 shows the individual IoU score for each label and Table 3 summarizes the metrics between the current and original study.

6 CONCLUSION

With the continued effort to integrate higher levels of autonomy in self-driving cars, research in autonomous UGVs falls behind. The evaluation of state-of-the-art DNNs in off-road terrain helps alleviate these gaps. In the context of perception, semantic segmentation provides a better understanding of the surroundings than object detection but at a higher computation complexity. SwiftNet demonstrates strong semantic segmentation results in terms of both inference speed and accuracy in less structured environments. The Titan RTX GPU used to train and evaluate SwiftNet limits this study because UGVs typically use edge devices, like a Nvidia Jetson, for video processing which would contain a less powerful GPU. Future work deploys and evaluates SwiftNet on a Nvidia Jetson AGX Xavier with model quantization [35] and optimization through TensorRT due to the limited

Figure 3: Example frame segmented with SwiftNet from the Rellis-3D test set: original image (top), predicted segmented image (middle), and ground truth segmented label (bottom).

hardware resources on edge devices. Another limitation lies in the unstructured environment itself. DNNs struggle to generalize on different environments other than their training data. Another future work area evaluates the trained SwiftNet model on video frames of different off-road terrain.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, 2015.

[2] J. Redmon and A. Farhadi, “Yolov3: An incremental

Class	IoU (%)
road	95.10
sidewalk	79.40
building	91.30
wall	50.10
fence	49.11
pole	61.84
taffic light	67.91
traff sign	74.96
vegetation	91.98
terrain	58.64
sky	94.10
person	80.76
rider	61.50
car	94.17
truck	67.58
bus	82.33
train	65.41
motorcycle	54.81
bicycle	75.93

Table 2: Individual class IoU from the Cityscapes dataset

Study	Dataset	FPS	mIoU (%)
Current study	Rellis-3D	24.5	77.9
Current study	Cityscapes	26.9	73.5
Original study	Cityscapes	34.0	75.9

Table 3: Inference speed and mIoU comparison of this work and the original work [13]

improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[3] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[4] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: a survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 315–329, 2020.

[5] A. Taeihagh and H. S. M. Lim, "Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks," *Transport reviews*, vol. 39, no. 1, pp. 103–128, 2019.

[6] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[8] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE winter conference on applications of computer vision (WACV)*, pp. 1451–1460, Ieee, 2018.

[9] M. Hofmarcher, T. Unterthiner, J. Arjona-Medina, G. Klambauer, S. Hochreiter, and B. Nessler, "Visual scene understanding for autonomous driving using semantic segmentation," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 285–296, Springer, 2019.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[11] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing*, vol. 493, pp. 626–646, 2022.

[12] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[13] M. Oršić and S. Šegvić, "Efficient semantic segmentation with pyramidal fusion," *Pattern Recognition*, vol. 110, p. 107611, 2021.

[14] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 325–341, 2018.

[15] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz, *et al.*, "Self-driving cars: A survey," *Expert Systems with Applications*, vol. 165, p. 113816, 2021.

[16] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

[17] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5000–5007, IEEE, 2019.

[18] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," in *2021 IEEE international conference on robotics and automation (ICRA)*, pp. 1110–1116, IEEE, 2021.

[19] K. Viswanath, K. Singh, P. Jiang, P. Sujit, and S. Saripalli, "Offseg: A semantic segmentation framework for off-road driving," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pp. 354–359, IEEE, 2021.

[20] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, "Stanley: The robot that won the darpa grand challenge," in *The 2005 DARPA grand challenge*, pp. 1–43, Springer, 2007.

[21] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Applied Soft Computing*, vol. 70, pp. 41–65, 2018.

[22] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *European conference on computer vision*, pp. 44–57, Springer, 2008.

[23] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1209–1218, 2018.

[24] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," tech. rep., Stanford, 2006.

[25] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[26] S. Sharma, J. E. Ball, B. Tang, D. W. Carruth, M. Doude, and M. A. Islam, "Semantic segmentation with transfer learning for off-road autonomous driving," *Sensors*, vol. 19, no. 11, p. 2577, 2019.

[27] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, pp. 1520–1528, 2015.

[28] A. Valada, G. L. Oliveira, T. Brox, and W. Burgard, "Deep multispectral semantic scene understanding of forested environments using multimodal fusion," in *International symposium on experimental robotics*, pp. 465–477, Springer, 2016.

[29] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[30] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

[31] M. Zhen, J. Wang, L. Zhou, T. Fang, and L. Quan, "Learning fully dense neural networks for image semantic segmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9283–9290, 2019.

[32] M. Orsic, I. Kreso, P. Bevandic, and S. Segvic, "In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12607–12616, 2019.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[34] M. Everingham, S. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International journal of computer vision*, vol. 111, no. 1, pp. 98–136, 2015.

[35] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.