Deep Learning models for 2D Semantic Segmentation in Natural Environments

Xuanzhi Liu
CSE, UNSW Sydney

Fang Chen
CSE, UNSW Sydney

Zhenghao Li
CSE, UNSW Sydney

Lin Pang
CSE, UNSW Sydney

Siyuan Yang
CSE, UNSW Sydney

***Abstract*—With the growth of Deep Learning and Computer Vision, Autonomous driving has become a hot research topic. Semantic segmentation plays a key role in the understanding of vehicle camera input images which assigns a label to each pixel in the image indicating which class the pixel belongs to. There have been many semantic segmentation researches in the direction of autonomous driving, but the scenarios of these researches are basically on urban roads. In this report, we implement and compare the segmentation accuracy of three models—DeepLabV3, DeepLabV3+, and PSPNet—on a natural environment dataset. We compare the performance of these three models on the mIOU and Accuracy metrics, and also summarise the features of their prediction results. At the end of the report, we present several directions for future work. Our trained weights can download from here: Google Drive**

I. Introduction

A. Semantic Segmentation

Semantic segmentation, together with image classification and object detection, are known as the three main tasks of computer vision. In deep learning area, it was first mentioned by Jonathan Long et al. [1] in 2015. The model uses many convolutional layers [2] to process the input images and output the class of each pixel in image, which is called 'mask'. During the past few years, there are many semantic segmentation models have been created, such as U-Net [3], which is mainly used for biomedical images segmentation, and DeepLab Series [4], which is mainly used in autonomous driving.

B. Autonomous Driving

In autonomous driving, semantic segmentation is able to recognise the different types of scenarios and objects the vehicle may encounter along the way. Compared with driving in urban roads and segment the input images from camera, doing segmentation in natural environments may bring more challenges to the vehicle. For example, when driving in a forest, they should detect where is the mud or puddle and keep away from them. There are also many tall trees and grass in the forest, the camera of the vehicle should segment them precisely and find the right road to drive.

C. Natural Environments Dataset

Kavisha Vidanapathirana et al. [5] published a dataset of natural environments in Australian forests. In the 2D parts of the dataset, it includes 9306 images about natural environments and their labeled mask in five different routes at different season. They manually annotated semantic segmentations for every sampled 2D image in their dataset, dividing the observed scene into a collection of different natural-scene classes. Figure 1. shows some samples of the dataset.

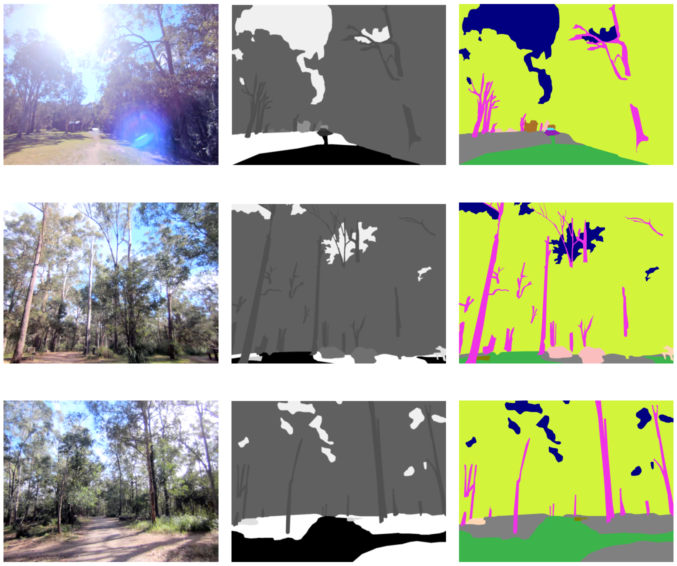
(a) Original Image (b) Gray-scale Mask (c) Color-map Mask

Fig. 1: Three samples of the WildScenes dataset. From left to right is the original image, the gray-scale mask of image and the color-map mask of image

D. Our Work

In this report, we compare three deep learning networks-DeepLabV3 [4], DeepLabV3+ [6], PSPNet [7]-on the natural environments dataset to test them whether can finish the semantic segmentation task in autonomous driving. We randomly select many images from the original dataset and divide them into training-set, validate-set and test-set in ratio. We use the pre-implemented models and pre-trained weights based on ImageNet [8] from Pytorch [9], Torchvision and Segmentation-Models-Pytorch [10]. We use torchmetrics [11] to calculate the metrics.

II. Literature Review

A. Convolutional Neural Networks

1) LeNet: LeCun et al. [12] used several convolutional layer, pooling layer to form a Deep Learning model and implemented it to recognise hand-write numbers in 1998. This is one of the earliest applications of neural networks. However, with the increasing difficulty of the task, LeNet can not finish the recognition of these tasks with high accuracy because its structure is too simple.

2) AlexNet: AlexNet is one of the most successful networks in the early of computer vision area. It was created by Krizhevsky et al. [13] in 2012. It uses convolutional layer, maximum pooling layer and fully connected layer to form the structure of network, and uses RELU [14] activate function and dropout [15] to avoid over fitting, which made a breakthrough in ImageNet classification challenge. However, AlexNet's task is focus on image classification, it can not performance well in object detection and semantic segmentation area.

B. Semantic Segmentation

1) FCN: Fully Convolutional Networks (FCN) was created by Jonathan et al. [1] in 2015. It introduced convolutional neural networks to semantic segmentation area by removing the fully connected layer at the end of the network with convolutional layer, and using upsampling to restore the resolution of image, which inspired other subsequent work. However, similar with the LeNet, it can not finish some complex segmentation tasks because of its simple network structure.

2) U-Net: U-Net was created by Ronneberger et al. [3] in 2015. It uses encoder-decoder to extract features and uses skip connections to connect corresponding layer, which can process the edge information in high accuracy. However, U-Net is widely used in segment biomedical images, it can not performance as well as the following networks we will introduce in autonomous driving area.

C. Semantic Segmentation in Autonomous driving

1) DeepLab Series: DeepLab is one of the most popular semantic segmentation models in autonomous driving area, which was created by LC Chen et al. [4] [16] [6] in 2017. In DeepLab, authors uses Atrous Convolution [4] to expand the receptive fields so it can capture more contextual information without increasing the computation. It also uses ASPP [16] and CRF [4] to improve the segmentation accuracy. In this report, we will compare the accuracy of DeepLabV3 and DeepLabV3+ on the natural environments dataset, and analysis why they can segment pixels in high accuracy.

2) PSPNet: H Zhao et al. [7] created PSPNet in 2017. It firstly uses a pretrained model, such as resnet [17], to extract the feature of the image, and uses Pyramid Pooling Module to capture muti-scale contextual information, then fuses the multi-scale features and finally input them into a classifier for prediction. In this report, we will compare the performance of PSPNet and DeepLab net on the natural environment dataset and try to explain why they work good or bad.

III. Methods

A. Split the Dataset

There are 9306 images and their corresponding masks in the 2D dataset. Due to the GPU constraint, we can not use all of them to train the dataset. So we randomly select a part of them for our training. The dataset includes five folders, for each folder, we randomly select 300, 150 50 images to training-set, validate-set and test-set, respectively. So we have 1500, 750, 250 images in training-set, validate-set and test-set, which is the ratio of 6:3:1.

B. Image pre-processing

1) Original images: For each input image, we resize it into 512 * 512 and expand it to tensor. We do not use normalization to process the images because we want to preserve more features of the image.

2) Masks: There are 19 classes in this natural environments dataset, which is labeled from 0 to 18, 0 is the background. The value of each pixel in mask indicates the class it belongs to. So the first step we need to remapping the labels into range 0 to 15 because three classes have been deleted in the paper of WildScenes [5]. After remapping, we resize the mask into 512 * 512 which is same as the original images and expand it to tensor.

C. ResNet

ResNet [17] is one of the most famous network structures today. Its main idea is using 'Residual Connection' and 'Shortcut Connection' to add the inputs directly to the outputs of the following layers, which can solve the gradient vanishing and gradient explosion problem. With Resnet, we can train very deep networks. ResNet has many differents structures, such as resent50, resnet101, resnext [18], and so on. Resnet can be used directly as a network structure for image classification or as a backbone for other networks for feature extraction.

D. DeepLabV3

1) Resnet Backbone: DeepLabV3 uses ResNet backbone to extract the features of input images. This feature map will be used in Atrous Convolution and ASPP parts.

2) Atrous Convolution: One of the main idea in DeepLab is that it created Atrous Convolution. Inserting some 0 values to the traditional convolution kernels, which can expand the view of the convolution but doesn't increase the volume of computation because we do not care the weights of zero.

3) Atrous Spatial Pyramid Pooling: ASPP part uses different size atrous convolution to capture multi-scale contextual information and merge them together for subsequent segmenatation task. After ASPP part, the feature map will contain multiple features of different scales.

E. DeepLabV3+

1) Decoder: Based on DeepLabV3, DeepLabV3+ add an decoder. Encoder part uses backbone (such as ResNet or Xception) to extract the feture map, decoder part is used to recover the details and boundary information in segmented images.

2) Improved ASPP: In ASPP part of DeepLabV3+, it uses more atrous convolution in different sizes, and implements some optimizations in feature merging.

F. PSPNet

1) ResNet Backbone: PSPNet uses ResNet structure to extract the features of input images. The output feature maps have 2048 channels, which contains different levels and scales image features.

2) PPM: PSPNet designs a Pyramid Pooling Module (PPM). It contains 3 parts: Multi scale pooling, upsampling and feature convergence. Using different size (such as 1x1, 3x3, 6x6) pooling layers to process the feature map, and upsampling the processed feature map to original size. Then, merged original feature map and processed feature map to get a integrated feature map with multiscale information.

G. Hyper Parameters

1) Epochs: 60

2) Learn Rate: $1e-4$

3) Optimizer: Adam

4) Criterion: CrossentropyLoss

5) Patience: 10

We use early stop to avoid overfitting. If the validate loss does not go down in 10 continuous epochs, the training will be stopped.

H. Metrics

We calculate the mIoU, mRecall and mAccuracy of the validate-set and test-set.

TABLE I: Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

1) IoU: IoU is a widely used metric in semantic segmentation area. It is is defined as the ratio of the intersection of the predicted and actual positive regions to their union.

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{1}$$

2) Recall: Recall is the ratio of true positive observations to the total actual positives:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

3) Accuracy: Accuracy is the ratio of true results (both true positives and true negatives) among the total number of cases examined:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3}$$

In a semantic segmentation task, we mainly focus on the performance of IoU because it can balance the impact of the various classes and prevent certain common classes from dominating the results of evaluations.

IV. Experimental Results

In order to improve the training speed, we upload the dataset to Google Drive and use A100 GPU provided by Colab to train the model. For each model, we save the best weights with the lowest validation loss and use it to test.

A. Training Curves

Figure 2. Figure 3. and Figure 4. in the appendix show the training loss, validation loss, validation mIoU, mRecall and mAccuracy of three models.

B. Test the models on test-set

Table II. shows the result of these three model's mIoU, mRecall and mAccuracy on the test set.

TABLE II: Performance Metrics of Different Models on Test-set

Model	mIoU (%)	mRecall (%)	mAccuracy (%)
DeepLabV3	15.98	21.29	22.31
DeepLabV3+	16.36	21.78	22.38
PSPNet	15.01	20.35	21.92

C. Prediction Results

Figure 5. Figure 6. and Figure 7. in the appendix show the predictions of the three models on the same five images.

V. Discussion

A. DeepLabV3 vs DeepLabV3+

DeepLabV3+ performs better than DeepLabV3 on all three metrics, this is because that DeepLabV3+ uses an encoder-decoder structure to capture the contextual information of images. Compared with Figure 5.,Figure 6. shows that DeepLabV3+ can segment more details of the images. Its predicted masks look like more roughly, especially for some tiny branches and tree trunks, DeepLabV3+ can segment them but DeepLabV3 can not do it.

However, both of these two models face over fitting problems. DeepLabV3+'s over fitting is a little more obvious, with a large error between the train loss and the validation loss. We think this is because that the existing structures of the networks are not able to learn new parameters after several epochs. The curves of mIoU and mAccuracy remain stable, with no significant increase, which also supports our idea. It means that the structure of DeepLab should be improved to finish this natural environments segmentation task.

B. DeepLabV3 vs PSPNet

PSPNet did not perform as well as DeepLab on all three evaluation metrics, all the evaluation metrics are lower that DeepLab, which means that using PSPNet to segment the natural environments dataset is not a good idea. We think this is because that DeepLabV3 uses ASPP with Atrous Convolution and PSPNet uses PPM models to capture the features of input image. PSPNet's method is not as good as DeepLab's method in terms of detail processing. From Figure

7., we can see that the masks of branches and tree trunks look much wider than the actual mask, which shows that PSPNet can not process edges information as good as DeepLab series.

C. IoU on Each Class

Table III. shows the result of 15 classes segmented by three models. DeepLabV3+ performs better in segmentation the class 'dirt' and 'mud', DeepLabV3 performs better in segmentation the class 'water'.

TABLE III: IoU Comparison of Three Models for Each Class

Class	DeepLabV3	DeepLabV3+	PSPNet
unlabelled	0.6600	0.6672	0.6428
dirt	0.4831	0.5093	0.4295
mud	0.7192	0.7280	0.6908
water	0.2114	0.1994	0.1392
gravel	0.1675	0.1677	0.1682
other-terrain	0.2533	0.2516	0.2554
tree-trunk	0.0534	0.0766	0.0520
tree-foliage	0.0095	0.0182	0.0161
bush	0	0	0.0006
fence	0	0.0001	0.0066
structure	0	0	0
rock	0	0	0
log	0	0	0
other-object	0	0	0
sky	0	0	0
grass	0	0	0
Average	0.1598	0.1636	0.1501

VI. Conclusion

In this project, we train three deep learning neural networks-DeepLabV3, DeepLabV3+, PSPNet-on a natural environments dataset for semantic segmentation task in autonomous driving. We compaere their mIoU, mRecall and mAccuracy on the test-set. We analysis the reason why the model performs good or bad. According to our experimental results, DeepLabV3+ performs the best in these three models. For our future work, we will focus on more semantic segmentation models like Segformer [19] or UperNet [20]. We will test their performance on this dataset and find some ways to improve their accuracy.

VII. Acknowledgements

In our work, we use the models and pretrained weights for Pytorch, Torchvision and Segmentation-Models-Pytorch, we use torchmetrics to calculate some metrics of our models. We declare herewith and express our high appreciation to their works.

References

[1] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[2] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE, 2010.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. 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, 2017.

[5] Kavisha Vidanapathirana, Joshua Knights, Stephen Hausler, Mark Cox, Milad Ramezani, Jason Jooste, Ethan Griffiths, Shaheer Mohamed, Sridha Sridharan, Clinton Fookes, and Peyman Moghadam. Wildscenes: A benchmark for 2d and 3d semantic segmentation in large-scale natural environments, 2023.

[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[7] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[10] Pavel Iakubovskii. Segmentation models pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2019.

[11] Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh Jha, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. Torchmetrics-measuring reproducibility in pytorch. *Journal of Open Source Software*, 7(70):4101, 2022.

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[14] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[15] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[16] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[19] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[20] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.

Appendix

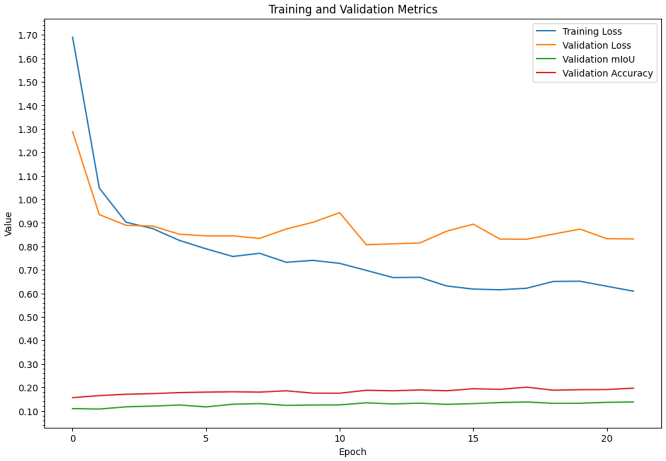

Fig. 2: Training Loss Curve for DeepLabV3

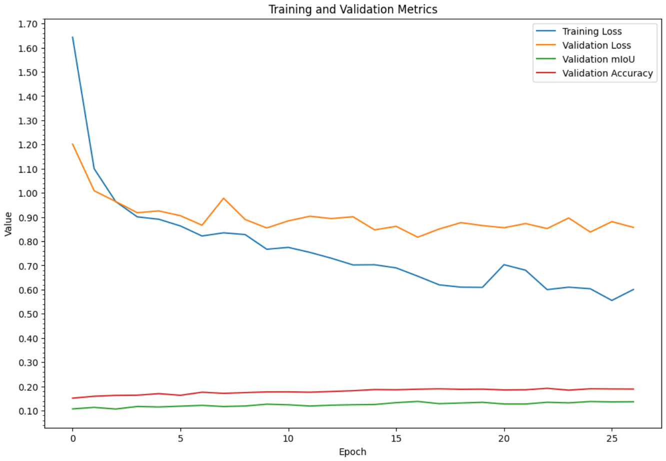

Fig. 3: Training Loss Curve for DeepLabV3+

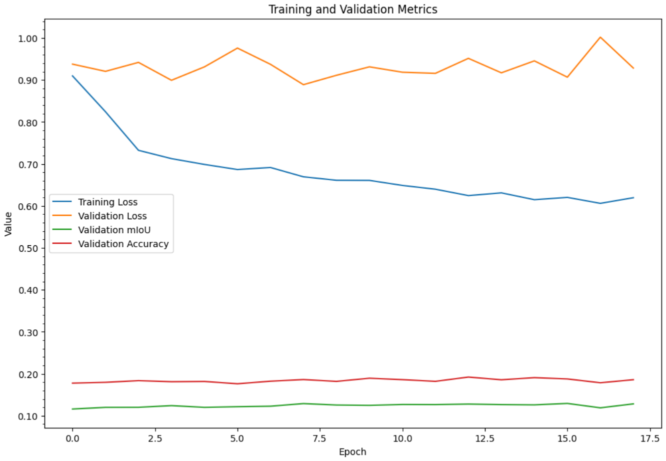

Fig. 4: Training Loss Curve for PSPNet

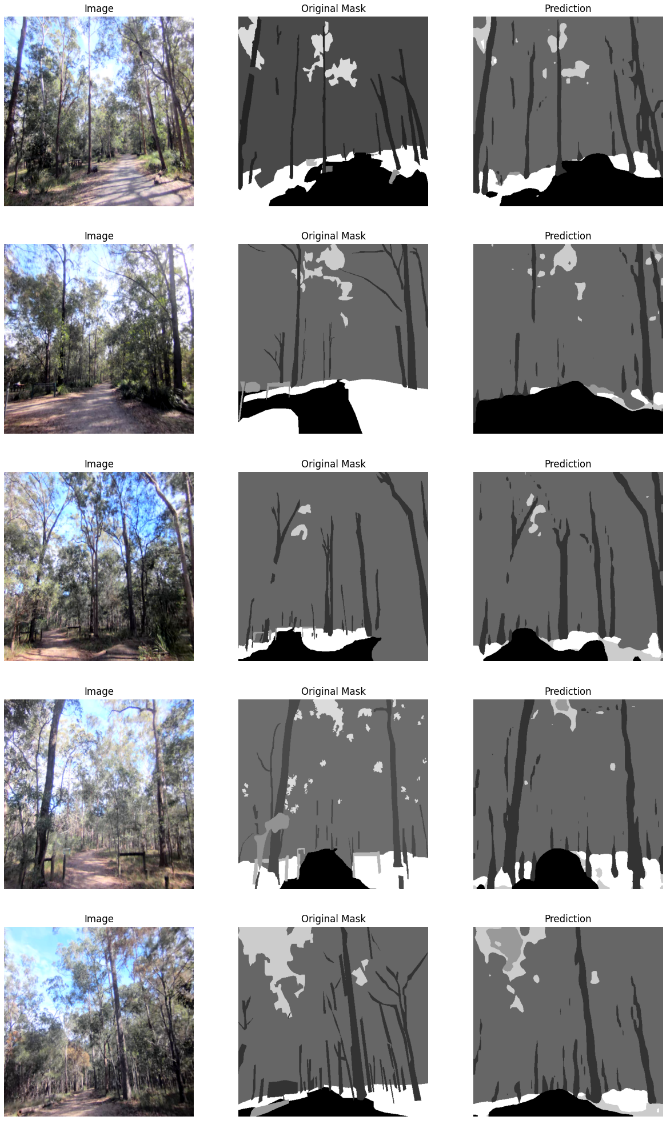

Fig. 5: Prediction Results for DeepLabV3

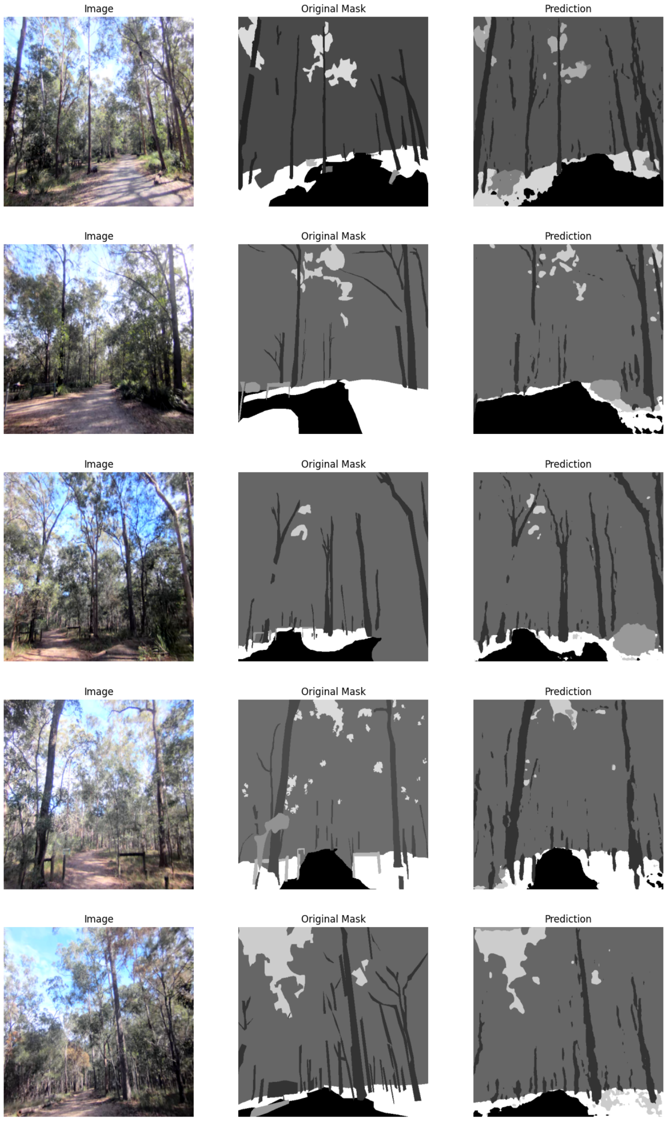

Fig. 6: Prediction Results for DeepLabV3+

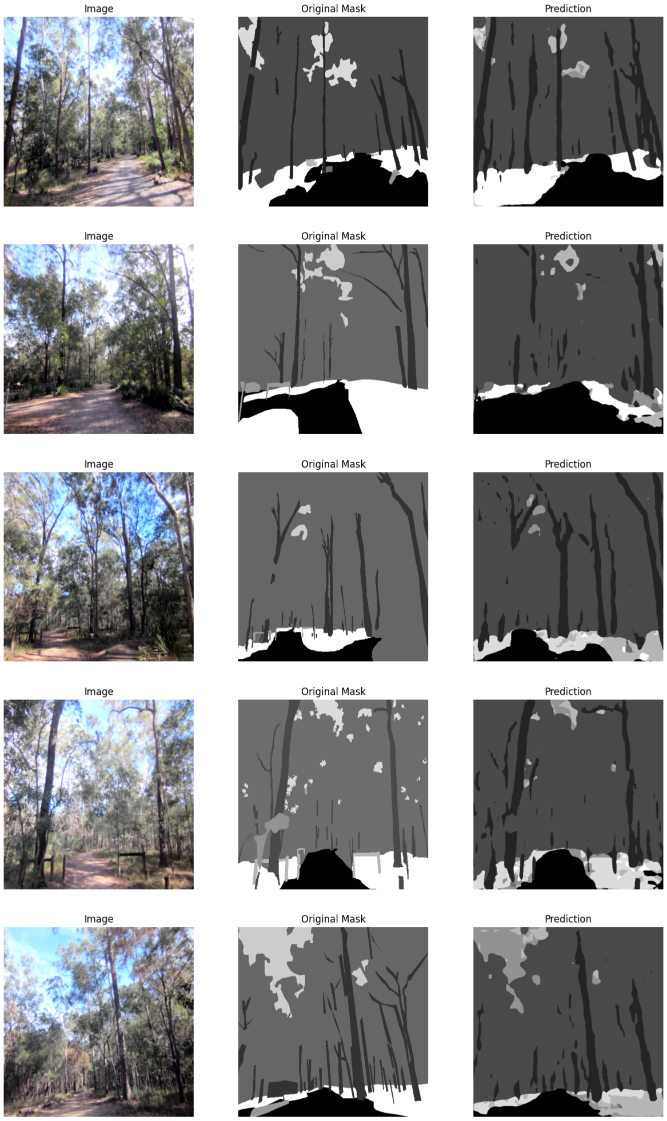

Fig. 7: Prediction Results for PSPNet