RE-IMPLEMENTING DEEPLABV3+

Alex Wang
Northeastern University
wang.alex2@northeastern.edu

11 December 2020

ABSTRACT

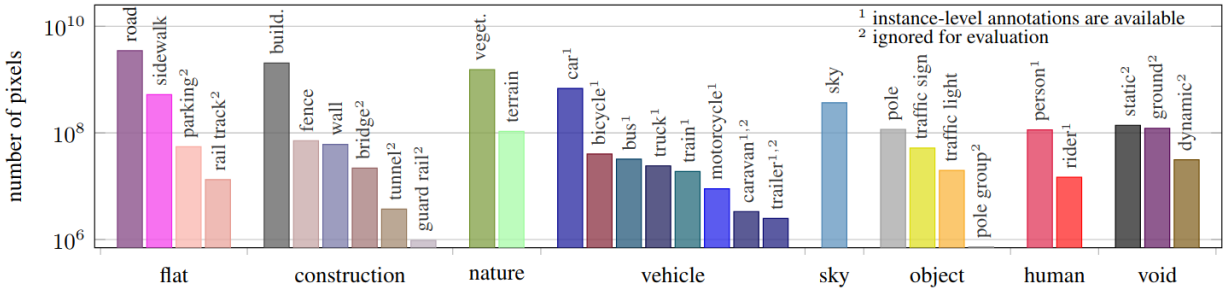
This work re-implemented DeepLabv3+ and attempted to recreate performance on the Cityscapes semantic segmentation dataset. The backbone is a modified ResNet-50 with atrous convolution and multi-grid rates using an output stride of 16. Atrous Spatial Pyramid Pooling and the decoder were implemented according to their descriptions in DeepLabv3 and DeepLabv3+, respectively. The re-implemented model used 3475 finely labeled Cityscapes images for training and validation (2975 for training and 500 for validation). The training protocol used random scaling, random left-right flipping, random crops of size 512, and a polynomial learning rate policy. The re-implementation achieved a mean intersection over union of 56.15 on the Cityscapes validation set.

1 Introduction

Autonomous vehicles use cameras and deep neural networks to develop an understanding of the surrounding environment. It is important for the vehicle to recognize the road, street signs, traffics lights, cars, bicycles, pedestrians, and so on. A task that can help provide such understanding of the environment is semantic segmentation, as it is used to classify each pixel of a given image to a specific label. The goal of this work is to re-implement DeepLabv3+, a former state of the art semantic segmentation model, and apply it to the Cityscapes dataset of complex urban street scenes.

1.1 Cityscapes Dataset

Cityscapes is a dataset of outdoor street scenes in 50 different cities in Germany. It contains thirty classes to label, which are grouped into eight different categories: flat, construction, nature, vehicle, sky, object, human, and void. Only 19 categories are considered in the evaluation since some categories rarely occur and others are not as relevant from an application standpoint. The distribution of pixels per class is shown below, and it can be seen that the dataset is dominated by roads, sidewalks, buildings, vegetation, cars, and the sky. Classes that are ignored for evaluation are labeled as such [1]:



The dataset contains 5000 finely labeled images and 20,000 coarsely labeled images with a height of 1024 pixels and width of 2048 pixels. The coarsely labeled images are only provided as extra training instances. The finely labeled images have been pre-split into 2975 training images, 500 validation images, and 1525 test sets. The split ensures an equal share of images from cities of different sizes, cities from different geographic regions, and images taken during different seasons. The splits are also done at the city level (*i*.e. all im-

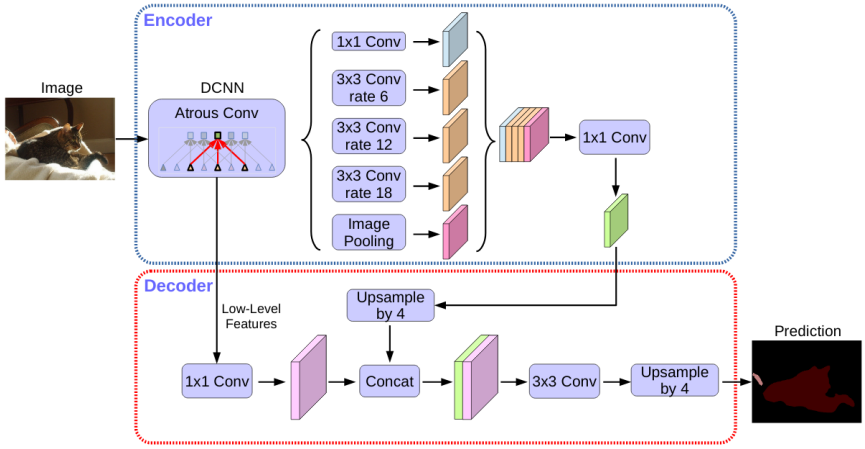
ages from Hamburg are contained in the train split). The labels for the test images are withheld and test set evaluation can only be done when submitting to Cityscapes' evaluation server. Therefore, this project only considers the validation metrics for comparison. An example of an image and its label is shown below [1]:

1.2 Metrics

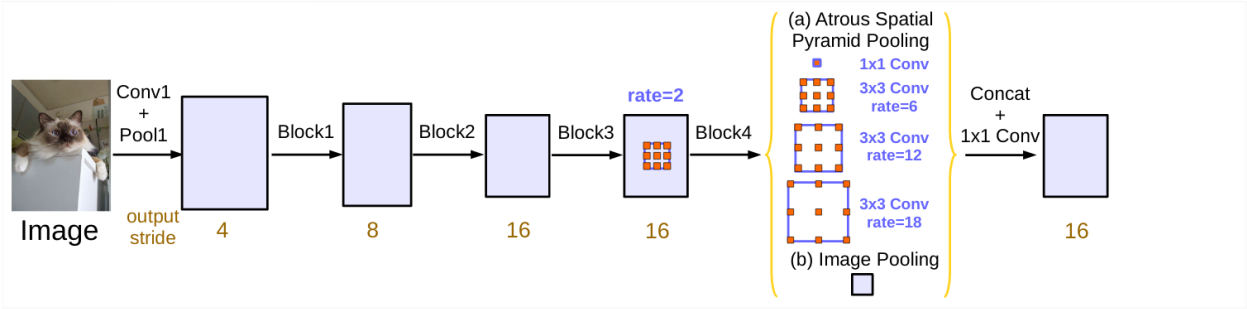
The main metric used to evaluate performance on semantic segmentation tasks like Cityscapes is Mean Intersection-over-Union (mIoU). For each class, this is calculated as $IoU = \frac{TP}{TP+FP+FN}$ where the true positives, false positives, and false negatives are calculated by pixel over the whole test set. The numerator is the number of correctly classified pixels, and the denominator is the number of pixels in either the true label or predicted label. Border pixels and void pixels are not considered in the metric. The mean is then taken to arrive at the mIoU [1]. The mIoU metric can be skewed towards larger objects, and the Instance-level Intersection over Union (iIoU) metric is used to multiply true positives and false negatives by the ratio of the class' average instance size to the size of the respective ground truth instance. Unfortunately, this metric is not as commonly reported, so this project evaluates performance based on the standard mIoU metric.

1.3 DeepLabv3 and DeepLabv3+

DeepLabv3+ is a former state of the art model for semantic segmentation. This is a fully convolutional neural network with an encoder-decoder structure. The purpose of the encoder is to learn contextual information at different resolutions, while the purpose of the decoder is to learn object boundaries. This model mainly consists of three components: the backbone, the Atrous Spatial Pyramid Pooling (ASPP) unit, and the decoder. The backbone extracts features from the images, the ASPP captures contextual information at multiple scales, and the decoder regains spatial information from two inputs: 1) low level features from an intermediate layer of the backbone at output stride 4, and 2) the output feature maps from the ASPP at output stride 16 or 8 depending on the computational budget (output stride 8 is more expensive). This general architecture is shown below [2]:



While DeepLabv3+ uses a modified aligned Xception model with atrous separable convolution as the backbone, this implementation uses the modified ResNet-50 backbone specified by the previous model design, DeepLabv3. DeepLabv3 is the encoder of DeepLabv3+, and directly up-samples the encoder output instead of using a decoder module like DeepLabv3+. The following shows the modified ResNet-50 backbone with the ASPP [3]:



DeepLabv3 modifies the ResNet-50 architecture [4] by using a stride of 2 at block1 instead of a stride of 1, and it uses atrous convolution in block4 with stride 1, base rate 2, and multi-grid rates 1, 2, and 4. This means that within block4, the three 3x3 convolutional layers will have dilation rates and paddings of 2, 4, and 8, respectively. This allows the network to widen its field of view and maintain its depth and output stride without impacting the number of parameters required [3].

Unfortunately, this implementation differs from this explanation because a stride of 1 was used in block1 instead of a stride of 2, and a stride of 2 was used in block3 instead of a stride of 1 as specified in the original ResNet paper [4].

The output of block4 is then passed through four parallel branches in the ASPP that captures contextual information at different atrous rates, and a fifth branch that does global average pooling followed by bi-linear up-sampling. The output of the five branches are concatenated and then sent through a 1x1 convolution to learn the context that is most important [3].

Lastly, the DeepLabv3+ decoder uses a 1x1 convolution to reduce the low-level features to 48 channels, concatenates this with the upsampled encoder output, further processes this with two 3x3 convolutional layers, uses a 1x1 convolutional layer to get a feature map of logits for each class, and finally upsamples this by four to produce the final prediction mask [2].

1.4 Benchmark Performance

The expected performance for DeepLabv3+ with a ResNet-50 backbone on the Cityscapes validation set is 76.6 mIoU [5].

2 Experimental Setup

Table 1: Training Hyper-parameter Choices (Experiment 2)

Hyper-parameter	**Re-implementation**	**Reference**
Pretraining	No	No
Backbone	ResNet-50	ResNet-50
Training Set	Train Fine	Train Extra
Optimizer	SGD	SGD
Weight Decay	0.0005	0.0005
Momentum	0.9	0.9
Initial Learning Rate	0.007	0.007 or 0.01
End Learning Rate	0.001	0.001
Learning Rate Policy	Poly (power = 0.9)	Poly (power = 0.9)
Decay Steps	90,000	90,000
Train Iterations	120,000	90,000
Batch Size	4	16
Batch Normalization Fine-Tuning	No	Yes
Initial Rescale	0.5	1.0 (None)
Random Scaling	0.5 to 2.0	0.5 to 2.0
Random Crop	Yes	Unknown
Crop Size	512	513
Random Horizontal Flipping	Yes	Yes

Two experiments were run with the finely annotated Cityscapes training and validation sets. Both experiments were trained end to end using cross entropy loss and batch sizes of 4. The hardware used was a RTX 2070 SUPER with 8GB of memory.

The first experiment resized the images and annotations from (1024, 2048) to (512, 512) and used the Adam optimizer with a learning rate of 0.001. Note that this is a resize and not a crop. Training was done using early stopping and a patience of 8 epochs.

The second experiment attempted to mimic the training protocol as described in DeepLabv3 and DeepLabv3+ [2]. This experiment ran for 120,000 iterations (DeepLabv3+ ran their model with 90k iterations but train and validation loss had not yet converged at 90,000 iterations), employed the polynomial learning rate policy with an initial learning rate of 0.007 and power of 0.9, used a crop size of 512 (DeepLabv3+ used a crop size of 769, the benchmark implementation used a crop size of 513), randomly scaled the images from 0.5 to 2.0 times, and used random horizontal flipping. DeepLabv3 and DeepLabv3+ did not mention what optimizer was used, but the original DeepLab mentioned using Stochastic Gradient Descent with momentum of 0.9 and weight decay of 0.0005 [6]. This implementation also initially resized the images and annotations from (1024, 2048) to (512, 1024) in an attempt to offset the fact that DeepLabv3+ used a crop size of 769, but this was not done for the benchmark training protocol. The validation set scaled images down from (1024, 2048) and took a center crop of size 512.

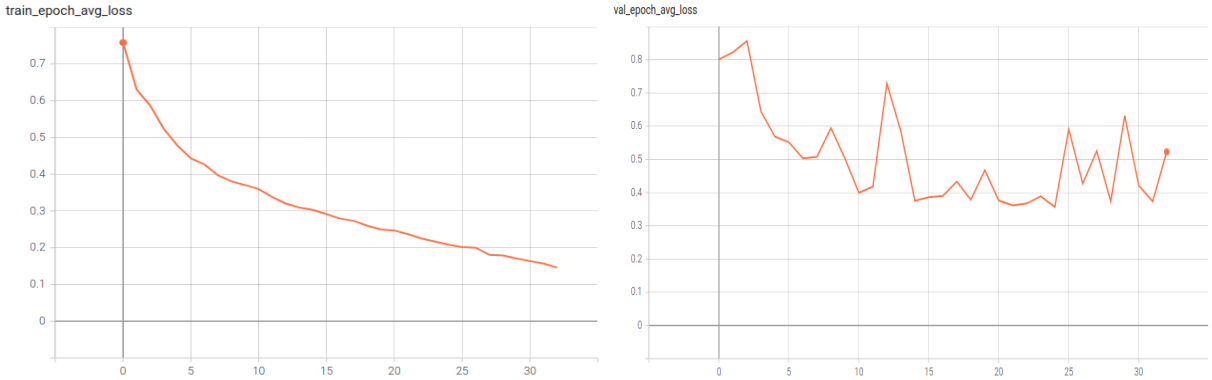
The full set of training hyper-parameters is presented in Table 1 with the differences highlighted in red.

For the final evaluation on the validation set after training, the predictions are made using the entire (1024, 2048) image.

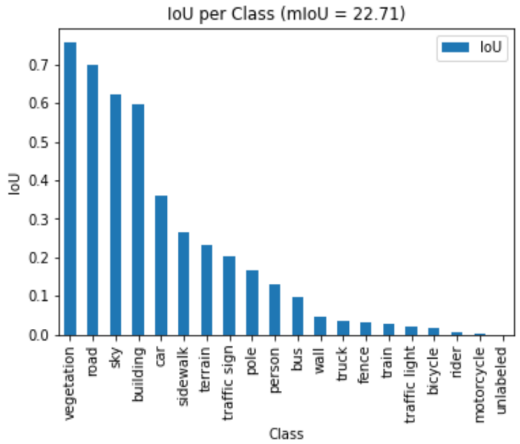
3 Results

3.1 Experiment 1

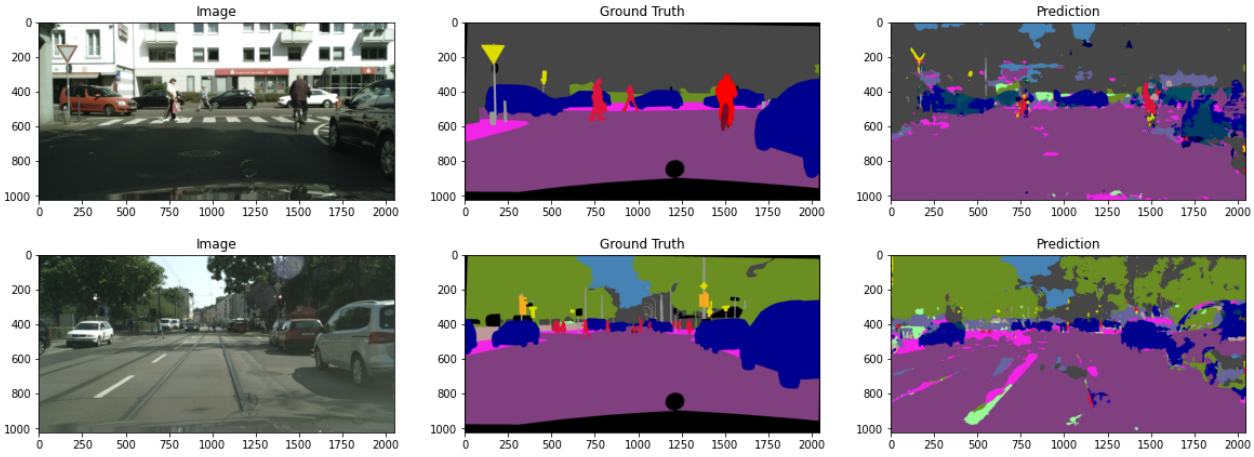
The following are the training and validation loss curves from experiment 1:



The mIoU on the validation set with image size (1024, 2048) was 22.71 for this experiment. It is clear that the resizing of the images at training time did not translate well to the final evaluation.



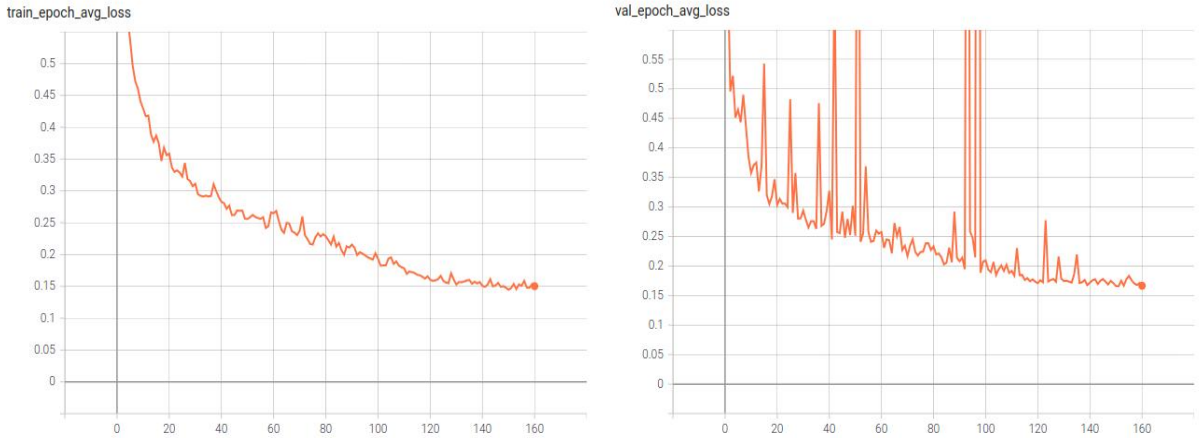
The following are some example predictions from inference:



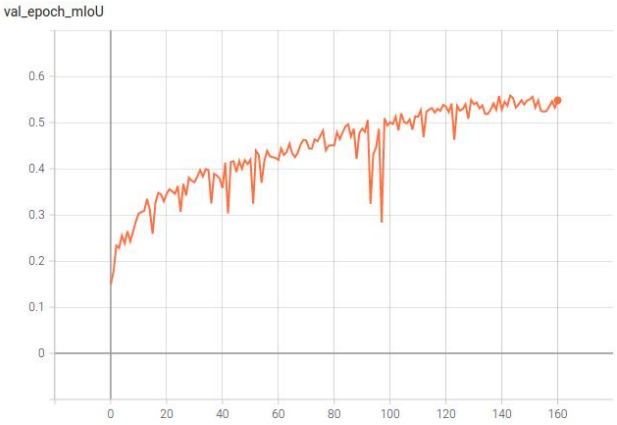
This model has started to learn larger classes like the road, buildings, and vegetation, but it has not yet learned to classify smaller objects like vehicles, humans, and street signs.

3.2 Experiment 2

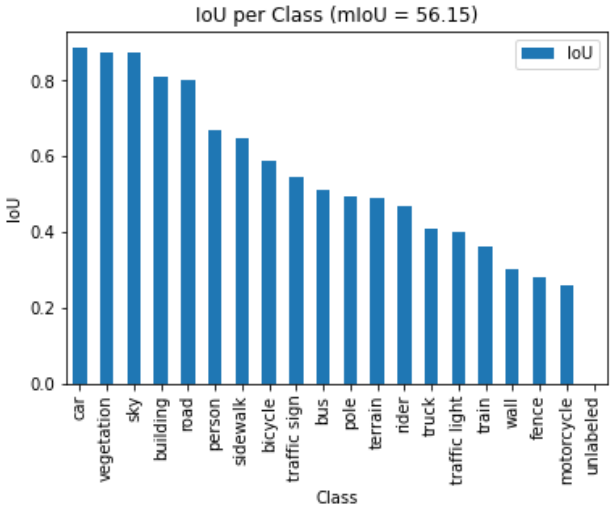
The following are loss curves from experiment 2, which took about 24 hours to run 160 epochs:



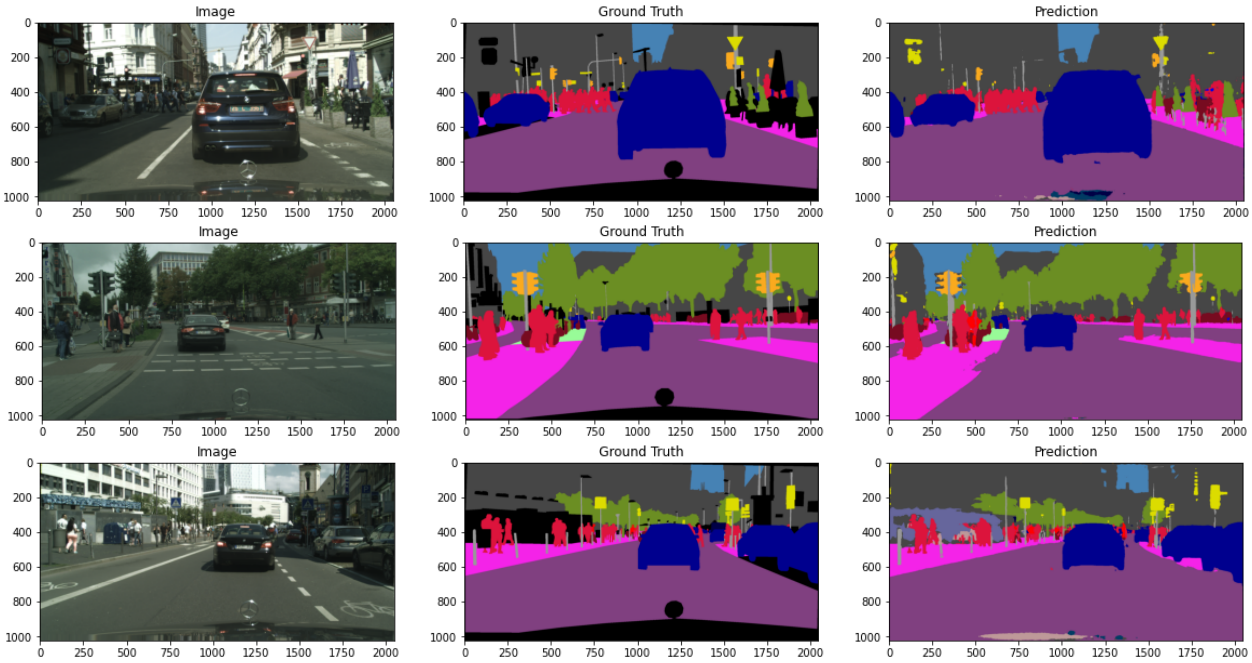
The validation mIoU reached a peak of 55.85 mIoU during training at epoch 143.



The mIoU on the validation set with image size (1024, 2048) was 56.15. This is still quite far off from the cited performance of 76.6 mIoU with the same model architecture. The IoU is generally better for classes that appear more frequently in the Cityscapes dataset.



The following are some example predictions from inference:



The predictions are a lot more detailed than experiment 1. The predicted segmentation mask is better at recognizing smaller instances of people and road signs. The object boundaries of larger objects seem to be quite accurate.

4 Conclusion and Future Work

This re-implementation of DeepLabv3+ achieved a mIoU of 56.15 on the Cityscapes validation set. The benchmark performance with the same backbone is cited as achieving a mIoU of 76.6. There are many possible reasons for why this implementation failed to produce similar results.

The first possibility is that the difference in backbone architecture caused worse performance. This implementation of the modified ResNet-50 followed the original ResNet architecture by using a stride of 1 in the first block [4]. However,

DeepLabv3 applied a stride of two to this block [3]. This means that this implementation returns low level features of output stride 4 after being processed through a ResNet block, while the DeepLabv3 architecture returns low level features before being processed through any bottleneck blocks. The extra processing through the bottleneck blocks could have caused a loss of spatial information due to further convolutional layers, but skip connections could have helped the model recover such information.

The second possibility is that the difference in batch size during training greatly reduced model performance. DeepLabv3 demonstrates the effect of batch size on performance for the PASCAL VOC 2012 validation set, as using a batch size of 4 achieved a mIoU of 64.43, a batch size of 8 achieved a mIoU of 75.76, and a batch size of 16 achieved a mIoU of 77.21 [3]. DeepLabv3+ also fine-tunes the batch normalization parameters, and higher batch sizes can help this process. Therefore, a similar effect on the Cityscapes dataset can also be expected.

Another possible explanation for worse performance is the size of the training set. This work only used the 2975 finely labeled training images, while DeepLabv3+ also used the 20,000 coarsely labeled. It could be beneficial to incorporate these images into training as well to help diversify the dataset.

There is a lot of potential future work that this project did not address. Apart from addressing the issues discussed, it would be worthwhile to investigate more types of data augmentation, metrics like iIoU and dice score, other loss functions like region mutual information loss, and other backbones like Xception.

References

[1] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *arXiv:1802.02611*, 2018.

[3] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv:1706.05587*, 2017.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *arXiv:1512.03385*, 2015.

[5] C. Kamann and C. Rother. Benchmarking the Robustness of Semantic Segmentation Models. *arXiv:1908.05005*, 2020.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs *arXiv:1606:00915*, 2017.