COMP9517 Group Project Report

Haiyue Yu
z5465699
UNSW
Sydney, Australia
haiyue.yu@student.unsw.edu.au

Hongyu Sun
z5497112
UNSW
Sydney, Australia
hongyu.sun2@student.unsw.edu.au

Xuetong Wang
z5431400
UNSW
Sydney, Australia
xuetong.wang1@student.unsw.edu.au

Yunlong Liu
z5493759
UNSW
Sydney, Australia
yunlong.liu@student.unsw.edu.au

Zishuo Wang
z5490411
UNSW
Sydney, Australia
zishuo.wang@student.unsw.edu.au

I. Introduction

Semantic segmentation is a basic task in computer vision that involves classifying images on a pixel-by-pixel basis. The technology is crucial for the development of self-driving vehicles as it ensures safe and efficient navigation of vehicles by accurately identifying and classifying elements in the environment. Although significant progress has been made in semantic segmentation in urban environments, however, semantic segmentation in natural environments faces more serious challenges due to its irregular and unstructured nature. Elements such as trees, water, dirt and gravel have a high degree of variability and complexity, requiring to adopt advanced segmentation technology.

The main purpose of this study is to develop and assess computer vision methods about semantic segmentation of images of natural environments. The evaluation was based on the recently released WildScenes dataset, which consist of 9306 two-dimensional images, each image has a resolution of 2016 x 1512 pixels. These images were taken in Venman National Park and Karawatha Forest Park in Brisbane, Australia. Each image is finely marked, providing a comprehensive basis for training and evaluating segmentation algorithms.

WildScenes data bring the challenges and opportunities for developing segmentation models, including lots of natural scenes that includes seasonal changes, diverse lighting conditions and extensive flora and fauna. The complexity of segmentation task would be increased because of these factors. Moreover, the enormous size of data requires more effective techniques to optimize the computational resources.

We aim to facilitate the development of the field of semantic segmentation in natural environment and provide some methods and insights for research and application in the future by this study.

II. Literature Review

The methods used in this research mainly include ResNet-50 and DeepLabV3 plus.

ResNet-50 is a convolutional neural network architecture mainly used for image classification tasks. As early as 2015, He et al. proposed a residual network that can improve speed and accuracy as a model of convolutional neural networks. ResNet-50 uses residual connections as a subtraction of learned features in the input of this layer based on ResNet [1]. In PyTorch, ResNet has 5 structures with different depths. In the first layer, three identical block structures are used so that you can enter the next layer without downsampling. In the second layer, after passing through the first block, the number of channels is first reduced by conv2, and then the size is reduced by conv2. Because of the size change, the input must be downsampled to the output. The next three layers do not need to be downsampled because they have the same structure. For the third and fourth layers, repeat the process for the second layer, except that the number of channels is increased and the output size is decreased. In the process of deepening the network layers, the feature extraction will gradually shift from simple features such as edges and textures to higher-level semantic features such as the shape and class of objects. In the research of Ikechukwu A V et al [2], it was shown that the accuracy of ResNet-50 is significantly better than the VGG-19 model, which shows the strong advantages of the ResNet-50 model in image segmentation.

DeepLabV3 [3] is a deep learning model for semantic image segmentation that uses convolutional neural networks (CNN) to extract image features and apply them to image segmentation tasks. In this experiment, an improved version of DeepLabV3, DeepLabV3 plus, will be used. As mentioned in the research by Liu M et al [4], DeepLabV3 plus introduces a decoder compared to DeepLabV3 that can compress low-level features and reduce the proportion of low-level features. In this experiment, the Atrous Spatial Pyramid Pooling (ASPP) module was combined with depth separable convolution [5] to better preserve the contextual multi-scale information of the input feature map. To save processing time and computational cost, the MobileNetV2 network pre-trained by the imageNet dataset was used as the basic network in this experiment.

The inverse structure and linear bottleneck structure of MobileNetV2 can extract image features more effectively and help reduce convolution computations [6]. In the research of Chen, Y. et al. [7] it was proposed that MobileNetV2, as a lightweight encoder model, can achieve efficient performance based on a smaller number of layers, parameters, and computation amount. Therefore, the combination of DeepLabV3 plus and MobileNetV2 is very suitable for semantic segmentation applications on mobile devices.

III. Method

A. *DeepLabV3+*

The implementation of this method refers to the deepLabV3 plus model [8] of user bubbliiiing on Github, and its complex code is simplified and improved.

*1) **Data Preparation**:* This dataset uses all data from the WildScenes2D dataset. In the preprocessing part, all images are first converted to RGB mode. The original image is rescaled and padded if necessary to maintain the aspect ratio of the image and to facilitate the subsequent training. To reduce computational complexity, the image is scaled to 512×512 for training.

*2) **Model Architecture**:* For this method, MobileNetV2 is used as the segmentation framework of the backbone network, and pre-trained weights are used to accelerate convergence and improve performance. In terms of model configuration, the image features are classified into 19 different classes according to the classification given in the paper, and the image is adjusted to the size of 512×512 for training. The downsampling factor can be selected as 8 or 16. For fast training, the downsampling factor is set to 16 here.

In this experiment, convolution was used to perform a two-dimensional convolution operation to extract the spatial features of the image. BatchNorm2d was used as the output of the two-dimensional convolution layer to accelerate neural network training and improve the stability of the model through batch normalization. Finally, the ReLU6 activation function is used to limit the output value to between 0 and 6. At the same time, the InvertedResidual module is used as the core module in MobileNetV2. By increasing the width of the input channel, then applying a depth convolution, and then shrinking it back. Add a ”shortcut” connection between the input and the output to implement the residual connection. Stacked through multiple Inverted Residual modules, and finally classified through a fully connected layer.

This experiment also combined the ASPP module to extract features from different receptive fields through multiple convolution branches (1×1 and 3×3) and convolution operations with different expansion rates (6, 12, 18). Then a global average pooling operation is used to obtain global features, restored to the original size through interpolation, and fused with multi-scale features. Finally, feature fusion is completed, the feature maps of all branches are spliced, and the final multi-scale feature representation is obtained through 1x1 convolution fusion.

*3) **Optimizer**:* Cross-entropy loss is used as the primary loss function, with Dice loss and Focal loss as auxiliary loss functions to handle class imbalance. Dice loss measures the similarity between predicted and ground truth labels, enhancing segmentation performance. Focal loss helps the model focus on hard-to-classify samples.

The optimizer of choice is Stochastic Gradient Descent (SGD) with a momentum of 0.9 and a weight decay of 1e-4. A cosine annealing strategy is used to adjust the learning rate during training. The learning rate starts high and gradually decreases as training progresses. The cosine annealing strategy includes warm-up and no-augmentation phases, making the learning rate changes smoother. After a certain number of iterations, the learning rate decreases proportionally.

*4) **Model Training**:* In the first 50 epochs, only the newly added layers are trained. After 50 epochs, the backbone network is unfrozen, and all layers participate in training. At this point, each batch contains 4 images to avoid memory issues. The model is distributed across multiple GPUs to accelerate training. During training, models are periodically saved to the logs folder, and the best-performing model based on validation loss is saved separately.

B. *ResNet-50*

The implementation of this method references the mask2former_r50 model implemented by K. Vidanapathirana et al [9]. But to demonstrate differentiation as well as to explore the feasibility of the model, the Mask2FormerHead as well as the Transformer decoder were not used, and the various parameters of the model were readjusted. The pre-trained ResNet-50 model from PyTorch [10] was mainly used for the implementation.

*1) **Data processing and Transformation**:* It was prepared to use all the data in the dataset for training. However, considering the time and computer performance, the original 2,016 x 1,512 pixels image and label were scaled down to 256 x 192 pixels in equal proportions, and then the image was transformed into a Tensor for input to the deep learning model for computation, and the label was transformed into a LongTensor for the computation of the loss function. These preprocessing can effectively improve the convergence speed and stability of the training process.

*2) **Modelling**:* ResNet-50 is a deep convolutional neural network with powerful feature extraction capabilities. The choice to use pre-trained models can leverage migration learning. Migration learning involves using pre-trained models on large datasets and fine-tuning them for specific tasks. This approach has been shown to significantly improve performance, especially when the target is small [11].

For the ResNet-50 model, his last two layers (The fully connected layer and the pooling layer) were removed instead adding two new convolutional layers (‘conv1’ and ‘conv2’) for generating segmentation results. The fully connected layer is used for classification tasks, mapping features to class-specific probability distributions. For image segmentation tasks, we need to classify each pixel, not the whole image. Therefore, the fully connected layer is not applicable for pixel level tasks. Pooling layers reduce the feature map to a single vector, which is appropriate for classification tasks, but for segmentation tasks we need to maintain the spatial dimension of the feature map in order to make predictions for each pixel. Convolutional layers are able to preserve the features map's spatial information. The conv1, this layer downscales the output of the last layer of ResNet-50 from 2048 channels to 512 channels, reducing the number of parameters and computational complexity while maintaining sufficient feature information. And the conv2, this layer downscales the number of channels of the feature map from 512 to the number of classes to generate the final segmentation result. And in the segmentation task, the output segmentation mask needs to be of the same size as the input image, so it is finally adjusted into to generate pixel-level predictions using F.interpolate.

*3) **Optimizer and loss function:*** K. Vidanapathirana et al's model's loss function uses a combination of loss functions, including cross-entropy loss, Dice loss, etc., and they used the AdamW optimizer, set different learning rate multipliers for different modules, and used the PolyLR learning rate scheduling strategy.

After researching, we finally decided to use the cross-entropy loss function as well as the AdamW optimizer and set the StepLR learning rate scheduler. The AdamW optimizer combines adaptive learning rate and independent weight decay to provide a stable optimization process. The cross-entropy loss function performs well in classification tasks and is an effective measure of the difference between the predicted probability distribution and the true distribution.

*4) **Training and validation:*** For the training of the model, instead of using the MMsegmentation profile, forward propagation is used to calculate the predicted output of the model, and back propagation is used to calculate the gradient of the loss function over the model parameters, so that the optimization algorithm can be used to update the model parameters and make the prediction results of the model closer to the real values. Finally, the model is evaluated by calculating the validation loss through the model's performance on the validation set.

IV. Evaluation and Results

A. DeepLabV3+

*1) **Data Preprocessing+:*** In order to ensure consistency in the colour channels, the input image is converted to RGB format. The model is configured to classify 19 different classes, and images are resized to 512×512 pixels for training, in order to fit the dimensions of the model, with any necessary adjustments made to maintain the aspect ratio. Furthermore, the input image is normalised by scaling the pixel values to the range [0, 1].

During the preliminary training phase, the model's backbone network is maintained in a fixed state, with each batch comprising eight images. Following the unfreezing of the backbone, each batch comprises four images. To enhance the efficiency of the training process, the data loading is conducted in parallel by four worker processes.

*2) **Model Architecture:*** In this experiment, we utilise the DeepLabV3+ model to perform the semantic segmentation task. The DeepLabV3+ model incorporates a Cavity Space Pyramid (ASPP) module that captures multi-scale contextual information, thereby enhancing the accuracy of the segmentation process. The model structure is capable of effectively expanding the perceptual field through cavity convolution while maintaining high resolution, thereby improving the target recognition rate. Furthermore, the DeepLabV3+ and MobileNetV2 models were integrated to leverage the strengths of both. The DeepLabV3+ model offers robust segmentation capabilities, while the MobileNetV2 model, as a lightweight and efficient convolutional neural network, can function as a backbone (feature extractor) to perform semantic segmentation tasks with enhanced efficiency, particularly on mobile or embedded devices with constrained computational resources.

*a) **Convolutional Layers:*** The DeepLabV3+ model incorporates the ASPP module, which comprises a sequence of parallel convolutional layers with varying expansion rates. The initial convolution layer in the ASPP module is characterised by a 1x1 convolution kernel and no extension. The remaining three convolution layers feature convolution kernel sizes of 3x3 and expansion rates of 6, 12 and 18, respectively. Furthermore, a global averaging pooling layer is incorporated, followed by a 1x1 convolution layer for resizing feature maps.

*b) **Upsampling:*** The output of the ASPP module is subjected to a final 1x1 convolution layer, which generates a segmentation graph comprising the requisite number of classes. Subsequently, bilinear interpolation is employed to upsample the feature map, thereby ensuring that the segmentation output is properly aligned with the input dimensions. The upsampling factor is set to 16, thereby maintaining a balance between accuracy and computational efficiency.

*3) **Training:***

*a) **Loss Function:*** The Cross-Entropy Loss is employed in the context of multi-class classification tasks, where it is deemed an appropriate methodology.

*b) **Optimizer:*** The Stochastic Gradient Descent (SGD) optimiser is employed.

*c) **Learning Rate Scheduler:*** The cosine annealing learning rate scheduler is employed, with an initial learning rate of 0.007 and a minimum learning rate of 0.00007, momentum of 0.9, and weight decay of 0.0001.

*d) **Gradient Descent:*** Gradient descent is performed at each step to prevent the accumulation of excessively large gradients, with the maximum norm set to 1.0.

*e) **Number of Epochs:*** As shown in Table 1, the model was trained for a total of 100 epochs. During the initial 50 epochs, the weights of the MobileNet backbone were maintained at a constant value, while only the final few layers of DeepLabv3+ underwent training. Subsequently, all backbone weights were unfrozen, and the entire model continued to be trained. At the conclusion of each calendar element, the training loss was calculated and recorded.

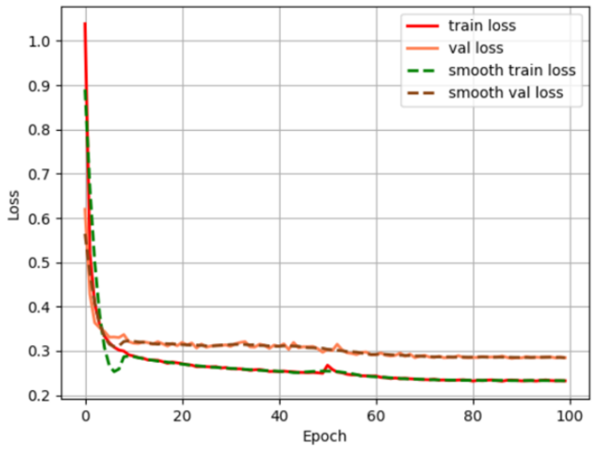

Fig. 1. train and val loss of DeepLabV3+

4) ***Experimental Results:*** At the outset of the process (epoch 0), both the training and validation losses are elevated, reaching approximately 1.0. This is a typical phenomenon at the outset of the training process, when the model has not yet been optimised. In the initial stages of training, there is a notable reduction in both the training and validation losses. This suggests that the model rapidly acquires pertinent patterns from the data during the initial phase of training. Following approximately 10 epochs, the rate of decline in both losses decelerates, and a state of stability is reached. This indicates that the model is approaching an optimal set of weights. Throughout the training process, the training loss is consistently lower than the validation loss. This is to be expected, given that the model is directly optimised to minimise the training loss. The validation loss is higher, but the trend is similar, indicating that the model generalises reasonably well to unseen data. The smoothed loss curves (smooth train loss and smooth val loss) facilitate the visualisation of the overall trend, eliminating the noise caused by batch-to-batch fluctuations. The smoothed curves demonstrate a discernible decline, thereby corroborating the assertion that the model is acquiring knowledge in a progressively efficacious manner. From approximately epoch 20 to epoch 100, both the training and validation losses reach a plateau. This indicates that the model has reached a point where further training does not result in a significant reduction in loss. This indicates that the model is approaching its optimal performance within the current training setup. The lack of a significant divergence between the training and validation losses indicates that the model is not exhibiting signs of overfitting.

a) ***IoU:*** The relatively high IoU values for the classes 'tree-foliage' (0.8419), 'dirt' (0.6502), 'sky' (0.5851), and 'grass' (0.5721) indicate that the model performs well in segmenting these classes. The classes ”gravel” (0.0036) and ”bush” (0.1239) exhibit markedly low IoU values, indicating a notable deficiency in the model's performance with respect to these classes. Some classes, including ”mud,” ”other-terrain,” and ”fence,” exhibit an IoU value of 0.0, indicating that the model is unable to accurately segment these classes. The mean IoU for all classes was 0.37, providing a measure of the overall performance of the model. Although the model still exhibits a deficiency in performance relative to certain classes, the average IoU suggests that it demonstrated a moderate to robust performance across the majority of classes.

TABLE I
IoU OF CLASSES ON DEEPLABV3+

Class	IoU
bush	0.1239
dirt	0.6502
fence	0.0
grass	0.5721
gravel	0.0036
log	0.1331
mud	0.0
other-object	0.1201
other-terrain	0.0
rock	0.1290
sky	0.5851
structure	0.3183
tree-foliage	0.8419
tree-trunk	0.4919
water	0.2906

b) ***Visual Results:*** As shown in Figure 2, the efficacy of the model can be evaluated by visualising the segmentation outcomes of the test samples and comparing the predictions with the actual circumstances.

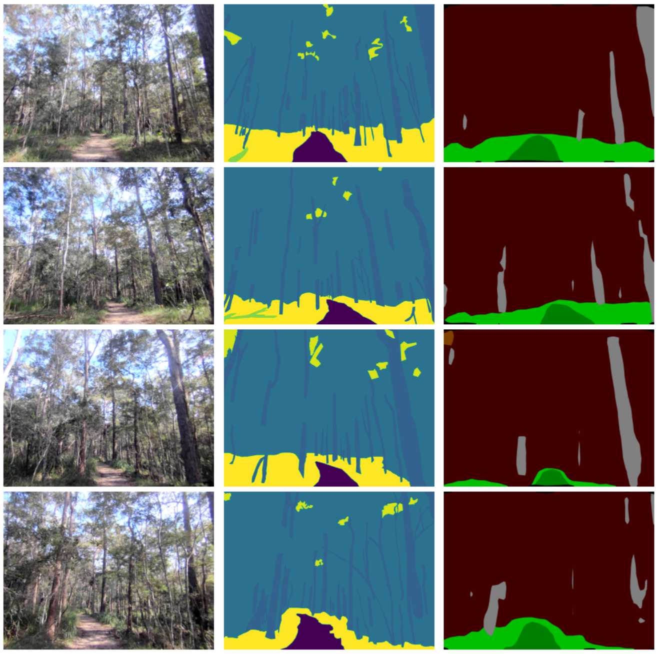

Fig. 2. Predictions results of DeepLabV3+

B. *ResNet-50*

*1) **Data Preprocessing**:* The images were resized to a uniform size of 256x192 pixels in order to ensure consistency and reduce the computational load. The images were transformed into PyTorch tensors utilising the ToTensor() method, which normalises the pixel values to the range [0, 1].

The labels were resized to 256x192 pixels and converted to integer-type tensors, thus aligning with the dimensions of the image and enabling the calculation of loss.

*2) **Model Architecture** :* The experiment employs a pre-trained ResNet-50 model as the foundation for its methodology. The final fully connected layer is eliminated, and two convolutional layers are incorporated to modify the output channels for the segmentation task.

*a) **Convolutional Layers**:* The first convolutional layer reduces the channels from 2048 to 512, and the second convolutional layer adjusts the channels from 512 to the number of classes.

*b) **Upsampling**:* Bilinear interpolation is used to resize the output feature map to the target size (256x192).

*3) **Training**:*

*a) **Loss Function**:* Cross-Entropy Loss is used, suitable for multi-class classification tasks.

*b) **Optimizer**:* The AdamW optimiser is employed with a learning rate of 0.0001 and a weight decay of 0.01.

*c) **Learning Rate Scheduler**:* The StepLR scheduler is employed, whereby the learning rate is multiplied by a factor of 0.1 at each of 10 epochs.

*d) **Gradient Clipping**:* Gradient clipping is performed at each stage of the back propagation process to prevent the accumulation of excessive gradients, with the maximum norm set to 1.0.

*e) **Number of Epochs**:* The model is trained for 30 epochs, with the training loss calculated and printed after each epoch.

*4) **Evaluation Method**:*

*a) **Validation Process**:* Following the conclusion of each epoch, the model's validation loss is determined through the utilisation of the validation set.

*b) **IoU Calculation**:* As shown in Table 3, the model's performance is evaluated on the test set by calculating the IoU (Intersection over Union) for each class and the overall mean IoU. The IoU is a metric that quantifies the degree of overlap between the predicted results and the ground truth. Higher values of IoU indicate superior performance.

*5) **Experimental Results**:* By plotting the training and validation loss curves, it is possible to observe the manner in which the model converges. The results demonstrate that the training loss exhibits a gradual decline, indicative of the model's ability to learn. Conversely, the validation loss initially decreases before subsequently increasing, suggesting that the model may be approaching a point of overfitting.

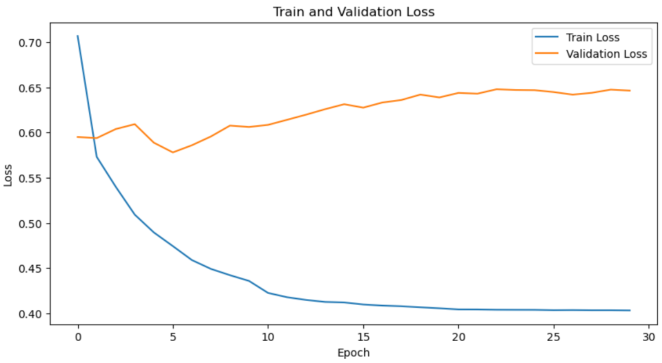

Fig. 3. train and val loss of ResNet-50

*a) **IoU**:* The IoU metric is calculated for each class, with enhanced performance observed in select classes (e.g., tree-foliage and dirt) and diminished performance observed in rock and other-terrain. The mean IoU for all classes is 0.1865, which provides an indication of the overall performance of the model.

TABLE II
IoU OF CLASSES ON RESNET-50

Class	IoU
bush	0.0911
dirt	0.6460
fence	0.0004
grass	0.5325
gravel	0.0038
log	0.0605
mud	0.0001
other-object	0.0437
other-terrain	0.0
rock	0.0
sky	0.4179
structure	0.0290
tree-foliage	0.7726
tree-trunk	0.1575
water	0.0424
Mean IoU	0.1865

*b) **Visual Results**:* As shown in Figure 4, the efficacy of the model can be evaluated by visualising the segmentation outcomes of the test samples and comparing the predictions with the actual circumstances.

V. Discussion

A. *DeepLabV3+*

From the training and validation loss curves, it can be seen that the DeepLabV3+MobileNetV2 combination converges rapidly within the first 20 epochs, and the loss value drops rapidly to around 0.3, and remains relatively stable throughout the training process.

The difference between the validation loss and the training loss is very small, indicating that there is no obvious overfitting phenomenon in the model during the training process.

The mIoU curve shows that the performance of the model improves rapidly at the beginning of the training and stabilizes at the later stage. The mIoU of the model reaches about 20 after 50 epochs, which indicates that the model performs well in semantic segmentation.

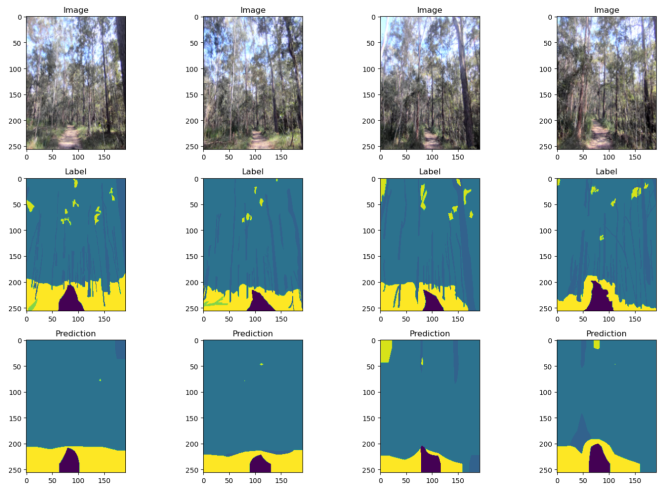

Fig. 4. Predictions results of ResNet-50

*1) **Advantages**:*

*a) **Fast convergence**:* The model's loss decreases rapidly within the first 20 epochs of training, indicating that the combined method is able to learn effective features quickly.

*b) **Strong generalization ability**:* The difference between the validation loss and the training loss is small, which indicates that the model has good generalization ability and performs consistently on both the training and validation sets.

*c) **Efficient feature extraction**:* MobileNetV2, as a lightweight feature extraction network, is able to extract more effective features while ensuring computational efficiency.

*2) **Disadvantages**:*

*a) **Large initial loss fluctuations**:* In the early stages of training, the loss value fluctuates a lot, which may be caused by unstable initial weights or high learning rate.

*b) **Slow improvement of mIoU in later stages**:* In the middle and later stages of training, the improvement of mIoU slows down significantly, indicating that it is difficult to further improve the model after reaching a certain performance.

B. *ResNet-50*

From the training and validation loss curves, the training loss and validation loss of ResNet-50 drop faster in the early stage of training, but the validation loss rebounds and fluctuates in the later stage, indicating that the model may have some overfitting phenomenon.

From the mIoU curve of ResNet-50, we can see that the performance of the model improves faster in the early stage of training, but in the middle and late stage of training, the mIoU improves slower, and the final mIoU value is slightly lower than the mIoU value of DeepLabV3+MobileNetV2.

From the classes' IoU analysis, ResNet-50 performs better on some class (e.g., tree-foliage, grass, dirt) but worse on others (e.g., rock, other-terrain), which suggests that the model's segmentation ability varies greatly across classes.

*1) **Advantages**:*

*a) **Better initial performance**:* In the early stages of training, the loss value of ResNet-50 decreases faster, which indicates that the model is able to learn some basic features quickly.

*b) **Good segmentation on some classes:*** In the class IoU analysis, ResNet-50 has better segmentation on some classes (e.g., tree-foliage, grass, dirt), which indicates that the model is more capable of recognizing these classes.

*2) **Disadvantages**:*

*a) **Large fluctuations in validation loss**:* In the middle and late stages of training, there are large fluctuations in validation loss, indicating that the model is overfitting during training.

*b) **Large difference in performance between classes**:* The model's segmentation performance for different classes varies widely, and the IoU values for some classes are very low, which indicates that the model has a weak ability to segment these classes.

VI. Conclusion

A. *DeepLabV3+*

The combined approach performs well in the semantic segmentation task with fast convergence and strong generalization. The loss of the model decreases rapidly at the beginning of training, and mIoU increases steadily throughout the training process. However, the large fluctuation in loss at the beginning of training and the slow increase in mIoU at the later stage suggests that the model may need a more stable training process at the beginning, and that further improvement becomes difficult after a certain performance is achieved.

B. *ResNet-50*

ResNet-50 performs well in the initial training and has strong segmentation ability for some classes. However, the model suffers from overfitting in the middle and late stages, with large fluctuations in validation loss and large differences in segmentation performance for different classes.

The very low IoU values for certain classes indicate that the model has a weak segmentation ability for these classes and needs further improvement.

C. *Future work*

*1) **Optimize the initial weights and learning rate**:* To reduce the loss fluctuation at the beginning of training, a more stable initial weight setting and appropriate learning rate adjustment strategy can be adopted.

*2) **Enhance the generalization ability of the model:*** By adding data enhancement and regularization methods, the generalization ability of the model can be improved and the overfitting phenomenon can be reduced.

*3) **Enhance the generalization ability of the model:*** Improvement of class segmentation performance: To address the problem of poor segmentation performance of certain classes, consider using class balancing techniques or targeted data enhancement methods to improve the segmentation performance of these classes.

*4) **Enhance the generalization ability of the model**:* Further enhancement of mIoU: After the model reaches a certain performance, we can try to use a more complex model structure or combine with multi-model integration methods to further improve the segmentation performance of the model.

References

[1] Deepak Theckedath and Rajendra R. Sedamkar. "Detecting Affect States Using VGG16, ResNet50 and SE-ResNet50 Networks". In: *SN Computer Science* 1.2 (2020), p. 79. DOI: 10.1007/s42979-020-0114-9. URL: https://doi.org/10.1007/s42979-020-0114-9.

[2] A. Victor Ikechukwu et al. "ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images". In: *Global Transitions Proceedings* 2.2 (2021). International Conference on Computing System and its Applications (ICCSA- 2021), pp. 375–381. ISSN: 2666-285X. DOI: https://doi.org/10.1016/j.gltp.2021.08.027. URL: https://www.sciencedirect.com/science/article/pii/S2666285X21000558.

[3] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[4] Man Liu et al. "Comparison of multi-source satellite images for classifying marsh vegetation using DeepLabV3 Plus deep learning algorithm". In: *Ecological Indicators* 125 (2021), p. 107562. ISSN: 1470-160X. DOI: https://doi.org/10.1016/j.ecolind.2021.107562. URL: https://www.sciencedirect.com/science/article/pii/S1470160X21002272.

[5] Yating Wu et al. "Ultrasound Image Segmentation Method for Thyroid Nodules Using ASPP Fusion Features". In: *IEEE Access* 8 (2020), pp. 172457–172466. DOI: 10.1109/ACCESS.2020.3022249.

[6] Bhakti Baheti et al. "Semantic scene segmentation in unstructured environment with modified DeepLabV3+". In: *Pattern Recognition Letters* 138 (2020), pp. 223–229. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2020.07.029. URL: https://www.sciencedirect.com/science/article/pii/S0167865520302750.

[7] Chongjin Chen et al. "Mining Better Samples and Semantic Consistency for Contrast Learning in Forest Semantic Segmentation". In: *2022 41st Chinese Control Conference (CCC)*. 2022, pp. 6202–6207. DOI: 10.23919/CCC55666.2022.9902405.

[8] bubbliiiing. *DeepLabv3+ PyTorch Implementation*. Accessed: 2024-07-20. 2020. URL: https://github.com/bubbliiiing/deeplabv3-plus-pytorch.

[9] CSIRO Robotics. *mask2former_r50_2xb20-80k_wildscenes_standard-512x512.py*. *WildScenes GitHub repository*. Accessed: 2024-07-20. July 2024. URL: https://github.com/csiro-robotics/WildScenes/blob/main/wildscenes/configs/mask2former/mask2former_r50_2xb20-80k_wildscenes_standard-512x512.py.

[10] PyTorch Team. *ResNet50*. *PyTorch documentation*. Accessed: 2024-07-20. July 2024. URL: https://pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html.

[11] Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning". In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359. DOI: 10.1109/TKDE.2009.191.