Semantic Segmentation of Natural Environments Using DeepLabV3 and SegFormerB0: A Comparative Study on the WildScenes Dataset

1 Qianyue Zhang
z5436473

2 Zhewen Zhu
z5369319

3 Tinghao Li
z5468081

4Yang He
z5458975

5 Hanbing Li
z5497007

***Abstract*—In the study of semantic segmentation of the natural environment,we use two different methods: the first called DeepLabV3 is a deep learning model based on convolutional neural network(CNN), the backbone network we chooses ResNet-50; The second is Segformer, a hybrid model based on Transformer, and Mit-B0 is selected for the backbone network. In this paper, we compare the training results of the two models by setting three evaluation indicators: Cross-Entropy Loss, Pixel Accurac(PA), and Mean Intersection over Union(mIoU). By analyzing the overall resolution and the resolution of each label, we explored the performance of each model in different categories, then we compared it with the existing literature to verify the training effect of our model under big data. Experimental results show that the both two models can complete the semantic segmentation task of the natural environment well. However, there are some differences in the recognition of individual labels. These experimental results are basically consistent with the literature description and model characteristics. In the future, we can further optimize the model through data augmentation, transfer learning, and skeleton network replacement to improve its performance in the semantic segmentation task of the natural environment.**

***Index Terms*—Semantic segmentation, DeepLabV3, SegFormerB0, Convolutional neural network(CNN), Transformer**

I. Introduction

In self-driving car research, the ability to safely and accurately navigate the natural environment is critical. This requires vehicles to be able to accurately identify various scenes and objects on the road. For example, when a car is driving on a road with more sand, it needs to drive more carefully. In contrast, on a smooth road, such care is not necessary. No matter what the setting, water should be avoided, but identifying individual objects in a natural setting is more challenging than in an urban setting, where there are often many irregular overlapping elements. Therefore, a thorough understanding of the scene is required.

In order to achieve the above mentioned goal we need to use semantic segmentation technique. Semantic segmentation technique assigns each pixel in the image to a predefined semantic label, thus converting the input image into a grid map. The grouping and analysis of 2D, 3D and video information is driven by this process. As a hot subject in the area of computer vision, semantic segmentation is closely associated with image classification and target detection. Entities and objects in the input data are identified by every mission in various ways, and various levels of data in the produced output is provided. Hence, deep insight and usage of semantic segmentation methods are necessary for realizing autonomous driving in natural environments.

This paper aims to compare different computer vision methods to achieve semantic segmentation of images in natural environment, and make some improvements on this basis. In recent years, convolutional neural networks (CNNS) have achieved remarkable success and are widely used in semantic segmentation tasks [1]. In this study, we used and tested two methods, DeepLabV3 and segformerB0. DeepLabV3 is based on convolutional neural networks, while SegFormerB0 combines CNN and converter architectures. By comparing the performance of the two models, we found the better one and found a scheme that is more suitable for semantic segmentation in the natural environment.

We used the WildScenes dataset, a recently released multi-modal dataset consisting of a sequence of five two-dimensional images recorded with an ordinary camera while traveling through Australia's Wenman National Park and Karawatha Forest Park. Each image in the dataset is manually labeled. The main advantages of the dataset are its large size (more than 20 km traversed in 6 months), suitable size (9,306 images and 12,148 point clouds), high two-dimensional resolution (2016×1512), and accurate 6-DOF positioning information. In addition, the content of the dataset involves traversing very dense and rugged terrain, which is helpful for studying autonomous systems that need to operate in remote, hard-to-reach locations.

However, we only used the technical part of the 2D vision-light image dataset because our limited equipment and time made it difficult to analyze and classify all semantic segmentation patterns (such as 3D point clouds, hyperspectral data, MRI, CT, etc.). In addition, we review the performance metrics used to evaluate the effectiveness of semantic segmentation.

II. Literature Review

A. Research Background

In this section, we summarize the advancements in semantic segmentation research. Semantic segmentation aims to classify each pixel in an image into a specific category and has broad applications in fields such as autonomous driving, medical image analysis, and remote sensing image processing.

Before the widespread adoption of deep learning, semantic segmentation primarily relied on traditional methods based on feature extraction and classifiers. Common feature extractors included SIFT, SURF, and HOG [2], while classifiers included Random Forest and Conditional Random Forest [3]. These methods are known as classical approaches, and their performance largely depended on the quality of manually designed features.

Since 2012, deep learning has developed rapidly, especially the maturity of convolutional neural network (CNN) technology, which has greatly improved the effect of semantic segmentation. The advantage of CNN is that it can automatically learn and extract image features, reducing the dependence on artificial design features. With the continuous optimization of CNN architecture, the mainstream method of semantic segmentation has gradually become a method based on deep learning.

In this case, we used two models, DeepLabV3 and SegFormerB0, both of which have shown excellent performance in various application scenarios.

B. Dataset

The WildScenes dataset, as provided, is a bi-modal dataset encompassing both 2D and 3D semantic annotations within natural environments. This dataset includes long-term, sequentially synchronized 2D images and 3D point cloud data, facilitating comprehensive analysis. Additionally, we provide precise 2D semantic labels obtained through meticulous human annotation [5].

C. Related Technologies

1) DeepLabV3

a) Atrous Convolution

Also known as dilated convolution, it grows the receptive field of convolutional filters without growing the number of coefficients or the amount of computation. By introducing holes between the filter weights, multi-scale data can be captured effectively by atrous convolution.

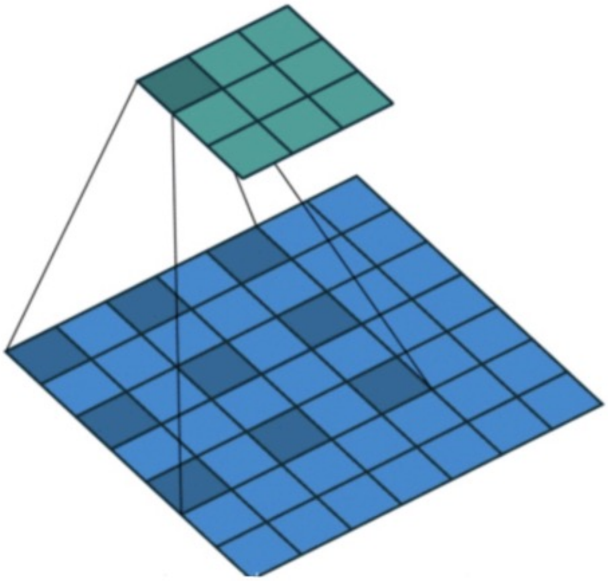


Fig. 1. Empty convolution picture interpretation

b) Atrous Spatial Pyramid Pooling (ASPP)

With parallel atrous convolutions with different sampling rates, the Atrous Spatial Pyramid Pooling (ASPP) module captures multi-scale data, improving the model's capacity of perceiving objects of different sizes [8].

c) ResNet Backbone

DeepLabv3 uses ResNet-50 as its backbone network, further improving feature extraction capabilities. The model structure of DeepLabV3 is shown in Figure 2.

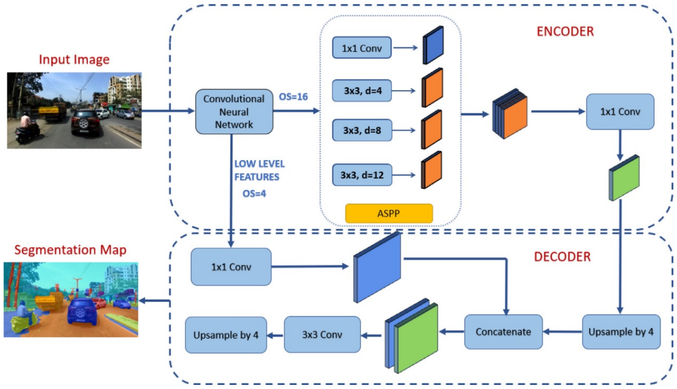


Fig. 2. DeepLabV3 model

DeepLab is a set of deep learning models the Google R&D team developed for semantic segmentation mission with the introduction of atrous convolution to improve the characteristic extraction ability as its main characteristic. The DeepLab series has evolved over some iterations, gradually improving the model structure and segmentation efficiency, with DeepLabV3 as a great version. DeepLabV3 keeps computational performance by comparing with past versions and greatly improves multi-scale object perception and refines detail processing.

DeepLabV3 has shown excellent efficiency in semantic segmentation missions. Excellent multi-scale characteristic extraction and segmentation precision are enabled by its atrous convolution and ASPP module.

2) SegFormerB0

As a recent semantic segmentation model on basis of the Transformer structure, Segformer combines the strong characteristics of Transformers with creative design to improve segmentation efficiency. The model structure of SegFormer is shown in Figure 3 [4].

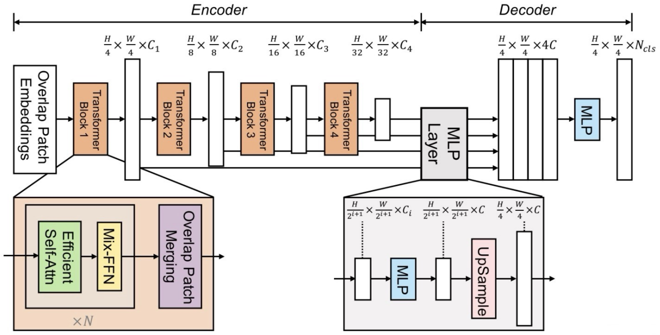

Encoder
Decoder
$\frac{H}{4}\times\frac{W}{4}\times C_1$
$\frac{H}{8}\times\frac{W}{8}\times C_2$
$\frac{H}{16}\times\frac{W}{16}\times C_3$
$\frac{H}{32}\times\frac{W}{32}\times C_4$
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{4}\times\frac{W}{4}\times N_{cls}$
Overlap Patch Embeddings
Transformer Block 1
Transformer Block 2
Transformer Block 3
Transformer Block 4
MLP Layer
MLP
Efficient Self-Attn
Mix-FFN
Overlap Patch Merging
$\times N$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C_i$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C$
$\frac{H}{4}\times\frac{W}{4}\times C$
MLP
UpSample

Fig. 3. SegFormer model

The Segformer series includes various versions, among which SegformerB0 is a lightweight base version. As a lightweight model in the Segformer series, SegformerB0 shows great merit in computational performance and multi-scale feature extraction. Through improving the model structure, SegformerB0 greatly decreases computational intricacy and ensures efficiency, so that it is highly useful on resource-constrained tools. Besides, Positional data is introduced by SegformerB0 through convolutional layers, decreasing the computational overhead related to positional encoding, and improving performace in processing high-resolution images. MiT (Mix Transformer encoder) is adopted, which usefully captures multi-scale data in images, so that it performs better in complicated scenes.

Compared to other semantic segmentation models such as PSPNet, SegNet, and UNet, the primary advantage of SegformerB0 lies in its high computational efficiency and lightweight design. PSPNet relies on pyramid pooling to capture multi-scale information, SegNet uses an encoder-decoder structure to recover high-resolution segmentation images, and UNet integrates features from different levels through skip connections. In contrast, SegformerB0's design, featuring no positional encoding and utilizing Overlap Patch Merging technology, allows it to maintain efficient feature extraction and processing without losing resolution. Experimental results show that SegformerB0 performs excellently on the ADE20K and Cityscapes datasets, achieving a good balance between computational efficiency and performance.

In summary, SegFormerB0 excels in semantic segmentation tasks due to its innovative architecture design and efficient feature extraction capabilities, offering significant advantages in multi-scale feature extraction and high-resolution segmentation while maintaining computational efficiency.

III. Method

On basis of our dataset analysis [5], it was discovered that it includes a lot of high-resolution images, capturing various natural scenes under various weather conditions. Hence, DeepLabv3 and SegformerB0 was chosen to fully utilize these feature. The sections below will detail the information preprocessing procedures and interpret the merits of these two models, and the rationale for their selection.

A. Data preprocessing

To achieve uniform class distributions, we processed the dataset as follows.

First, we labeled the images based on the corresponding RGB values and consolidated all images into a directory named "sample." Within the sample directory, there are two subdirectories: images and labels, with a one-to-one correspondence between the files in these subdirectories. Next, we generated a master CSV file based on these images, which includes the image ID, the path to the image in the images directory, and the path to the corresponding labeled image in the labels directory.In the second step, we employed stratified sampling to extract 50% of the dataset and generated a corresponding CSV file. In the third step, based on the stratified sampling results from the second step, we further divided the data into training, validation, and test sets in a 7:0.5:2.5 ratio, creating separate CSV files for each subset. Additionally, these CSV files included a binary column for the labels, where 0 indicates the absence of a label and 1 indicates the presence of a label.

This processing ensures that the dataset has uniform class distributions, facilitating subsequent model training and evaluation.

Figure 4 shows the statistical count of each label before sampling, Figure 5 shows the count after sampling, and Figure 6 displays the code output results after sampling. From these three figures, it is evident that our sampling process ensured that each.

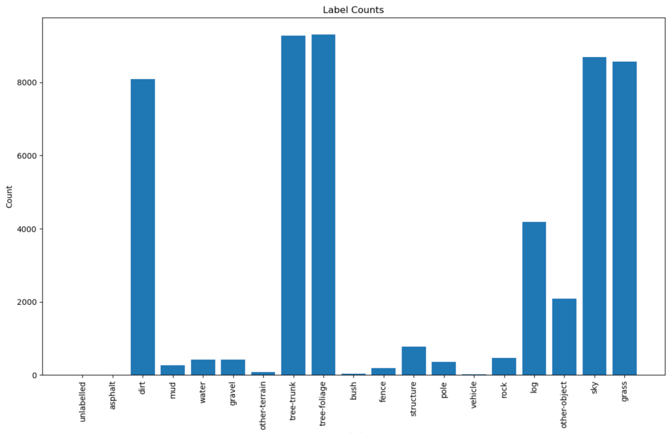


Fig. 4. labels original

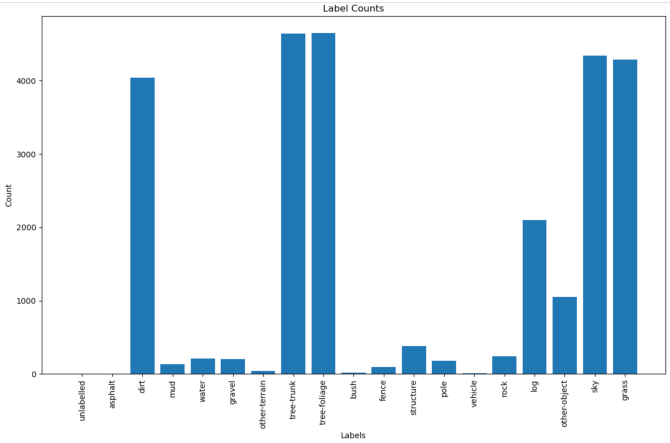
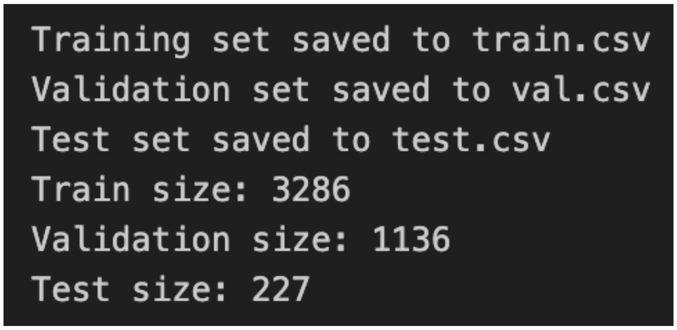


Fig. 5. labels split



```
Training set saved to train.csv
Validation set saved to val.csv
Test set saved to test.csv
Train size: 3286
Validation size: 1136
Test size: 227
```

Fig. 6. split result

B. DeepLabV3

DeepLabv3 integrates the merits of standard convolution and atrous convolution, so that it is especially useful in dealing with high-resolution images and complicated scenes, which are characteristics of our diverse and high-resolution dataset.

Firstly, adopting atrous convolution, DeepLabv3 extends the receptive field of the convolutional kernels without growing the computational burden. Atrous convolution inserts gaps between the factors of the convolutional kernel to achieve this, so that the kernel covers a bigger image area without growing the number of coefficients [6]. This ability is key to capturing the complicated contextual data discovered in natural environments, improving the model's capacity of perceiving details at different scales, enhancing segmentation precision.

Secondly, an Atrous Spatial Pyramid Pooling (ASPP) module is adopted by DeepLabv3 [7], fusing multi-scale characteristics through parallel atrous convolutions with various sampling rates. The ASPP module usefully extracts data at various scales by enabling the model, growng the accuracy of object recognition across different sizes. This multi-scale feature fusion ability is especially significant for precisely segmenting the rich natural scenes in our dataset.

Besides, ResNet-50 is adopted as its backbone network by DeepLabv3. While providing great feature extraction abilities, ResNet-50's deep residual network structure usefully solves the vanishing gradient issue in deep networks by skip connections. With the 50-layer depth of ResNet-50, the model captures rich and complicated characteristics, so that it is highly useful for high-resolution image processing and precise segmentation missions.

According to experimental outcomes on the Cityscapes dataset, DeepLabv3 realizes an impressive 78.83% average Intersection over Union (mIoU), displaying its excellent efficiency in semantic segmentation. To sum up, DeepLabv3, with its extended receptive field, multi-scale feature fusion ability, and powerful feature extraction backbone network, is an ideal option for dealing with high-resolution and diverse natural environment images. Precise semantic segmentation outcomes are provided without greatly growing computational costs, so that it is highly appropriate for our study dataset.

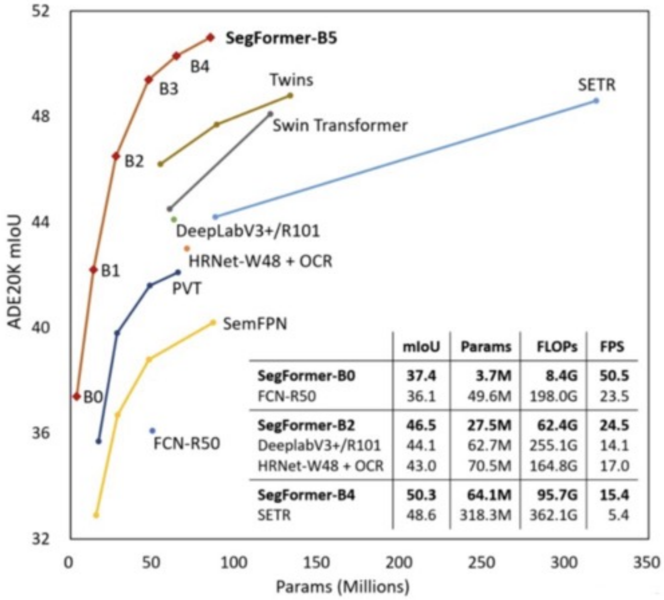
C. SegFormerB0

Diverse object data in images is usefully captured by the hybrid Transformer encoder (MiT) in Segformer through hierarchical feature extraction. To be specific, the MiT encoder adopts the dynamic attention system of the Transformer structure, so that it extracts rich image data at various feature levels. This hierarchical feature extraction approach provides the MiT encoder with a great merit in dealing with complicated scenes and diverse objects. By paying attention to both local and global features within an image, the MiT encoder can conduct more accurate semantic segmentation and object recognition. This ability comes from the Transformer's self-attention system, so that the model dynamically selects and focuses on significant areas of the image across various feature layers, capturing multi-scale image data. While enhancing the model's feature representation, it enhances its generalization capacity and robustness in complicated scenes.

Unlike traditional Transformers, Segformer introduces positional information through convolutional layers, thereby reducing the computational overhead associated with positional encoding. Traditional Transformers rely on positional encoding to introduce positional information for each element in the input sequence to preserve their order. However, this positional encoding typically adds extra computational complexity.

To further improve computational efficiency, Segformer implements a dynamic pruning strategy. This technique prunes non-informative neurons based on specific input instances, significantly reducing computational load without sacrificing accuracy. This advantage is particularly evident for high-resolution datasets like WildScenes. Segformer achieves an excellent balance between performance and computational efficiency. One of the goals of its lightweight design is to reduce computational complexity, making it deployable on low-power devices.

Experimental results on the ADE20K dataset show that Segformer achieves a new state-of-the-art performance with 51.0% mIoU [8], significantly improving efficiency, as shown in Figure 7. All results were obtained using a single model and single-scale inference, demonstrating Segformer's outstanding performance and efficiency in semantic segmentation tasks. Although the experimental results indicate that SegFormerB5 achieves the highest mIoU among the SegFormer models, it requires high-performance hardware and longer processing times. Therefore, we chose SegFormerB0 for our experiments due to these constraints.



	mIoU	Params	FLOPs	FPS
SegFormer-B0	**37.4**	**3.7M**	**8.4G**	**50.5**
FCN-R50	36.1	49.6M	198.0G	23.5
SegFormer-B2	**46.5**	**27.5M**	**62.4G**	**24.5**
DeeplabV3+/R101	44.1	62.7M	255.1G	14.1
HRNet-W48 + OCR	43.0	70.5M	164.8G	17.0
SegFormer-B4	**50.3**	**64.1M**	**95.7G**	**15.4**
SETR	48.6	318.3M	362.1G	5.4

Fig. 7. mIoU comparison of each model

To sum up, Segformer MiT-B0, with its cutting-edge hybrid Transformer encoder, convolutional positional encoding, and dynamic pruning strategy, offers high-precision semantic segmentation outcomes and maintains computational performance. This makes it highly appropriate for our study dataset, involving high-resolution and diverse natural scenes.

IV. Experimental Results

A. Environment Setup and Results

Our computing environment is made up of a computer configured with an NVIDIA GeForce RTX 3060 graphics card. Our experiment was first downloaded to our local computer a 2D portion of the WildScenes dataset, which is made up of V-01, V-02, V-03, K-01, and K-03. The vscode software platform connected to a local GPU is used to implement and test our code. According to our tests, this setup is feasible, and the outcomes display that our configuration usefully backs data processing and model training.

During the experiment, we kept using vscode for data processing and model training. Data loading, preprocessing, model training and validation are included in the concrete procedures. All operations are carried out in the local GPU environment, which ensures the efficient utilization of computing resources and the reliability of experimental results. Finally, a series of experiments validate the feasibility of our approach, showing that our setup and method can efficiently process and analyze 2D portions of the WildScenes dataset.

Through this experimental process, we confirmed the feasibility of our experimental method, which laid the foundation for the subsequent development of our experiment.

B. Parameter setting

1) DeepLabV3

Firstly, we define ASPP(Atlas Space Pyramid Pool) module to capture multi-scale context information by processing different sensory fields through multi-scale Atlas convolution. Next, we define the DeepLabv3 class, which includes a ResNet-50 backbone for feature extraction, an ASPP module for multi-scale feature processing, and a 1x1 convolution classifier for generating the final segmentation results. We resized the image to 256x256 pixels and set the batch size of the data loader to 8 to maximize GPU utilization. Finally, by adjusting the learning rate and the number of iterations, we find that the training effect is best when the learning rate is 0.0001 and the number of epochs is 25.

2) SegFormer

First, we define attention mechanism classes to capture remote dependencies in images, including multi-head self-attention. We then implement a multi-layer perceptron (MLP) module for further feature processing, consisting of two fully connected layers and an activation function (GELU). The Block class combines an attention mechanism and an MLP module to handle features, including the normalization, attention, and MLP layers. The SegFormerBackbone class implements the backbone network, which consists of four encoders, each consisting of a convolutional layer and multiple Block units, for multi-scale feature extraction. The segformer0 class implements the complete SegFormer architecture by upsampling the final output to the target size (256x256) via bilinear interpolation. We set the batch size of the data loader to 8 to maximize GPU utilization. Finally, by adjusting the learning rate and the number of iterations, we find that the training effect is best when the learning rate is 0.0001 and the number of epochs is 15.

C. Setting of evaluation criteria

The performance of our models was evaluated using metrics commonly used in image processing. These indicators include: Cross-entropy loss, pixel accuracy (PA), and average cross-linking (mIoU). For the detailed evaluation of labels, only average cross-linking (mIoU) is used. The following is the calculation and description of these three evaluation indicators.

1) Cross-Entropy Loss

Cross-Entropy Loss is a commonly used loss function, particularly suitable for classification tasks. In image segmentation tasks, it measures the difference between the predicted probability distribution and the true label distribution. By minimizing Cross-Entropy Loss, the model can learn to make more accurate class predictions. The formula is:

$$\text{Loss} = -\sum_{i=1}^{N} y_i \log(\hat{y_i}) \tag{1}$$

Where y_i are the true labels, and $\hat{y_i}$ are the predicted probabilities by the model.

2) Pixel Accuracy

Pixel Accuracy (PA) is one of the basic metrics for measuring the performance of image segmentation models. It represents the ratio of correctly predicted pixels to the total number of pixels. High Pixel Accuracy indicates that the model can make correct predictions on most pixels. The formula is:

$$\text{PA} = \frac{\sum_i (TP_i + FP_i + FN_i)}{\sum_i TP_i} \tag{2}$$

Where TP is True Positive, FP is False Positive, and FN is False Negative.

3) Mean Intersection over Union

Mean Intersection over Union (mIoU) is a commonly used evaluation metric in image segmentation tasks. It calculates the overlap between the predicted segmentation and the ground truth segmentation, measuring the overall segmentation performance across different classes. Higher mIoU indicates better segmentation performance across different classes. The formula is:

$$\begin{aligned} \text{IoU} &= \frac{TP}{TP + FP + FN} && (3) \\ \text{mIoU} &= \frac{1}{C} \sum_{i=1}^{C} \frac{I_o U_i}{U_i} && (4) \end{aligned}$$

Where C is the number of classes.

D. Experimental result

These are the results of our experiment. Figure 8 shows the semantic segmentation image of DeepLabV3, and Figure 11 shows the semantic segmentation image of SegFormerB0. Figure 9 presents the changing trend of Loss and mIoU during training and verification for DeepLabV3, while Figure 12 shows the same for SegFormerB0. Figure 10 illustrates the IoU value of each class for DeepLabV3, and Figure 13 presents the IoU value of each class for SegFormerB0.

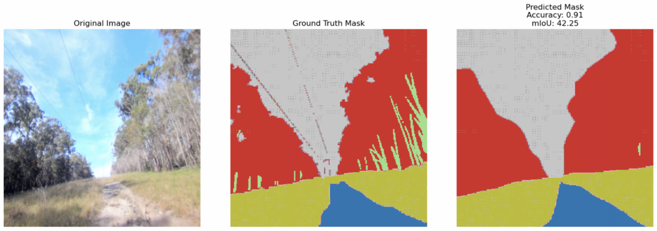


Fig. 8. DeepLabV3 semantic segmentation results

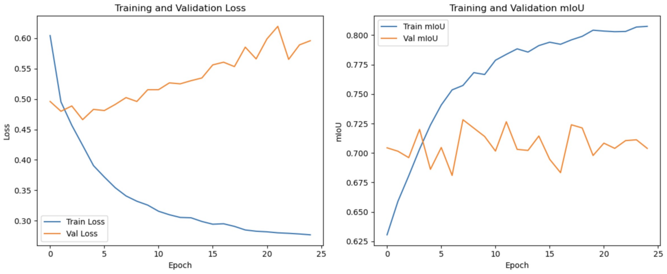
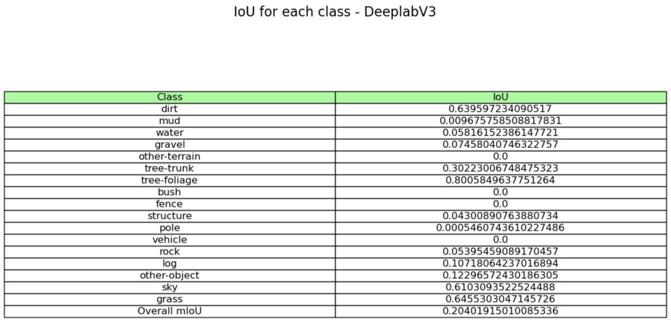


Fig. 9. DeepLabV3 result



IoU for each class - DeeplabV3

Class	IoU
dirt	0.639597234090517
mud	0.009675758508817831
water	0.05816152386147721
gravel	0.07458040746322757
other-terrain	0.0
tree-trunk	0.30223006748475323
tree-foliage	0.8005849637751264
bush	0.0
fence	0.0
structure	0.04300890763880734
pole	0.0005460743610227486
vehicle	0.0
rock	0.05395459089170457
log	0.10718064237016894
other-object	0.12296572430186305
sky	0.6103093522524488
grass	0.6455303047145726
Overall mIoU	0.20401915010085336

Fig. 10. DeepLabV3 class IoU

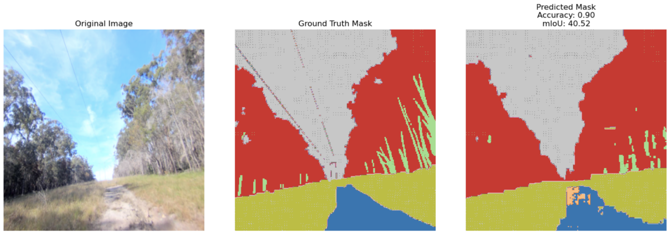


Fig. 11. SegFormerB0 semantic segmentation results

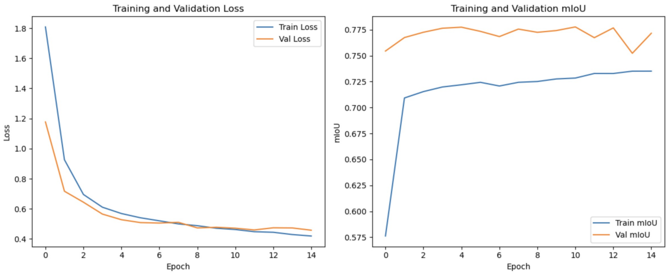
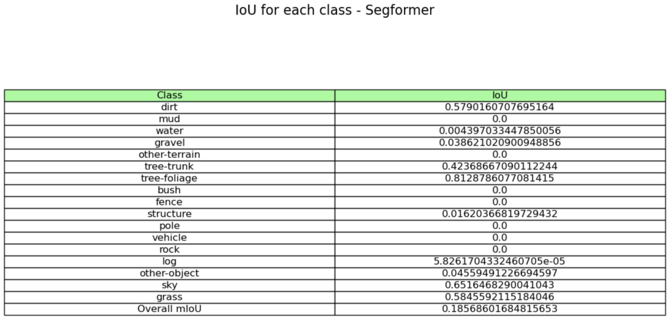


Fig. 12. SegFormerB0 result


IoU for each class - Segformer

Class	IoU
dirt	0.5790160707695164
mud	0.0
water	0.004397033447850056
gravel	0.038621020900948856
other-terrain	0.0
tree-trunk	0.42368667090112244
tree-foliage	0.8128786077081415
bush	0.0
fence	0.0
structure	0.01620366819729432
pole	0.0
vehicle	0.0
rock	0.0
log	5.8261704332460705e-05
other-object	0.04559491226694597
sky	0.6516468290041043
grass	0.5845592115184046
Overall mIoU	0.18568601684815653

Fig. 13. SegFormer class IoU

V. Discussion

A. DeepLabV3

1) Result analyse

In Figure 9 we show the loss and mIoU trends of DeepLabV3 during training and validation.

The figure on the left shows changes in losses during training and validation. We observed that the training loss decreased steadily from the initial value of 0.6 and gradually stabilized, indicating that the model was constantly learning and optimizing. However, in the middle period of training, the validation loss showed a certain fluctuation, which indicated that the model may have overfitting phenomenon. The figure on the right shows how mIoU changes during training and validation. mIoU starts at 0.65 and keeps increasing, eventually stabilizing at around 0.80. This shows that the segmentation performance is improved during the training process and reaches a very high level at the end. However, validation mIoU shows volatility, showing an upward trend but not stabilizing at the highest position. It can be proved that our model does exist overfitting phenomenon.

Figure 10 shows the IoU results by class and the overall mIoU in DeepLabV3. The overall mIoU was 0.20. The IoU of tree-leaf is as high as 0.80, which indicates that tree-leaf is excellent in leaf segmentation. In addition, Dirt, Grass and Sky occupations also performed well, with IoU values above 0.60. However, some categories performed poorly, such as MUDs with an IoU of just 0.01, and seven categories with an IoU of 0, indicating a lack of successful segmentation in these categories.

2) Advantages

It is clear from the trends in loss and mIoU that the model performs well on training data and excelled in some categories. This shows that the model has the ability to efficiently capture and segment high-resolution details. Although the overall performance is not ideal, the strong performance in specific classes proves the potential of the model in handling complex scenarios and detailed segmentation.

3) Disadvantages

During training, the model showed signs of fitting, possibly due to insufficient training data. Future experiments could increase the sample size from 50 percent to 80 percent or more to alleviate this problem. In addition, the parameters can be adjusted to improve the results. By fine-tuning the parameters, better experimental results can be obtained.

B. SegFormerB0

1) Result analyse

Figure 12 shows SegFormerB0 losses and mIoU trends during the training and validation phases.

The figure on the left shows the loss over time. Both training and validation losses dropped sharply at the beginning of training, indicating that learning and optimization worked well at an early stage. Then the loss tends to be stable, indicating that the model has reached a convergence point. The figure on the right depicts trends in mIoU during the training and validation phases. Training mIoU rose rapidly from the beginning of training and stabilized at about 0.72, indicating good learning performance in image segmentation tasks. Although the verified mIoU has a certain fluctuation, it is stable at about 0.75, indicating good generalization ability and a balance between learning and generalization.

In addition, the IoU results for each class and the overall mIoU for segformer0 are shown in Figure 13. The overall mIoU was 0.1857. The IoU of leaves was the highest (0.8129), indicating excellent performance in leaf segmentation. Sky and Dirt also performed well, with iou of 0.6515 and 0.5790 respectively. However, some classes perform poorly, such as Mud, which has an IoU of 0.0, and several others also have an IoU of 0.0, indicating that the model is almost completely unable to split these classes.

2) Advantages

The trend of losses and mIoU shows that the model does not exhibit a fit and performs well in terms of generalization, consistency and convergence. The IoU results of individual classes also show that the model performs well in the segmentation of some specific classes with stability and reliability.

3) Disadvantages

The relatively low initial mIoU that increases over multiple epochs indicates that the model requires multiple epochs to achieve optimal performance. This may indicate the need for better initial parameter Settings or further fine-tuning. The poor performance of some classes may be due to the insufficient sample size of these particular classes, and increasing the sample size of these classes may lead to better results.

By solving these problems, the overall performance of the model can be further improved, especially when classes are segmented with fewer samples.

VI. Conclusion

A. Experimental Procedure

1) Dataset Processing

We annotated and organized the images in detail, and generated a new CSV file to facilitate hierarchical sampling of the data. Through this processing method, a uniform class distribution of the dataset is ensured, which is helpful for subsequent model training and evaluation. When dividing the dataset, we divide the dataset into three parts according to the requirements: the training set, the validation set, and the test set. The above methods are used to ensure that the goal of uniform class distribution is achieved.

2) Model Architecture Selection

In the choice of model architecture, we decided to use two different 2D semantic segmentation methods: the DeepLabv3 (Chen et al. (2019)) and Segformer (Xie et al. (2021)) variants of the Resnet-50 backbone. In order to select the most suitable Segformer skeleton, we performed a single image test. Although the originally planned SegFormer maximum model (SegFormer-B5) was not possible due to device performance limitations, we chose SegFormer-B2, which is a compromise between performance and time. However, in order to optimize training time and resources, we ultimately decided to use the lightest B0 architecture. In addition, we partially tested the B2 and B5 models on the test set to speculate on the performance of the larger model for a more accurate assessment of SegFormer.

3) Model Training and Testing

After determining the evaluation indexes as Cross-Entropy Loss, Pixel Accuracy (PA) and Mean Intersection over Union (mIoU), we trained and tested the two models on three datasets. Each model undergoes detailed training steps on each dataset to ensure the reliability and consistency of the results. During training, we continuously adjust the hyperparameters to achieve the best model performance. By regularly saving and evaluating the intermediate results of the model, it is ensured that problems in the training process can be found and changed as soon as possible, and the causes can be easily analyzed.

B. Results Analysis

1) Overall Comparison

For the overall test results, we compared the two models. Through the output data and images, we can intuitively determine which model performs better. Further, we compare these results with those in related papers to evaluate whether the model's performance meets the training expectations under the big data sample. In particular, we only selected the lowest performance skeleton MIt-B0 on the Segformer framework, and we also performed single-image training processing on MIt-B2 and MIt-B5, and the results showed that our model showed a stepwise growth in all three evaluation criteria, which verified the generalization ability of the model on different datasets.

2) Detailed Comparison

We also generated a detailed evaluation of 15 labels in the 2D image in the two models, and only used the uniform intersection and union ratio (mIoU) as an index, and analyzed the resolution ability of the two models on different types of objects in detail. We compared these results with those in the relevant papers and found that our assessment was highly consistent with the results in the papers. This makes it easier to select suitable models for different geomorphological environments in the future, and also provides a direction for future work.

C. Future Work

In view of the different abilities of the model when recognizing different objects, we plan to try the following optimization methods: data augmentation, transfer learning, and skeleton replacement. Through data augmentation, the model's ability to recognize various objects in the natural environment is increased. Then use the existing model knowledge to improve the performance of the new model. In the case of higher demand, replace the skeleton with higher performance to improve the overall recognition effect. All of the above are new directions and new ideas for our future work.

VII. Member contribution

Our group ensures that all members were involved in the discussion and completion of each section.

A. Report

Hanbing Li and Yang He completed the introduction, literature review, and methods sections of the report.

Tinghao Li, Qianyue Zhang, and Zhewen Zhu completed the experimental results, discussion, and conclusion sections.

B. Code

Model 1 was completed by Hanbing Li and Yang He.

Model 2 was completed by Tinghao Li, Qianyue Zhang, and Zhewen Zhu.

After each training session, each group member suggested for adjustments based on the model's performance and proceeded with the next round of modifications.

C. Video

Each section of the PowerPoint and video presentation is prepared separately and then compiled together at the end. The division of sections for the video presentation does not mean that members only participated in their assigned parts.

References

[1] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[2] Routray, S., Ray, A. K., & Mishra, C. (2017, February). Analysis of various image feature extraction methods against noisy image: SIFT, SURF and HOG. In 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT) (pp. 1-5). IEEE.

[3] Speiser, J. L., Miller, M. E., Tooze, J., & Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. Expert systems with applications, 134, 93-101.

[4] Bai, H., Mao, H., & Nair, D. (2022, May). Dynamically pruning segformer for efficient semantic segmentation. In ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 3298-3302). IEEE.

[5] Vidanapathirana, K., Knights, J., Hausler, S., Cox, M., Ramezani, M., Jooste, J., ... & Moghadam, P. (2023). WildScenes: A Benchmark for 2D and 3D Semantic Segmentation in Large-scale Natural Environments. arXiv preprint arXiv:2312.15364.

[6] Yurtkulu, S. C., Şahin, Y. H., & Unal, G. (2019, April). Semantic segmentation with extended DeepLabv3 architecture. In 2019 27th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

[7] Baslamisli, A. S., Groenestege, T. T., Das, P., Le, H. A., Karaoglu, S., & Gevers, T. (2018). Joint learning of intrinsic images and semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 286-302).

[8] Nguyen, T. T., Dinh, S. V., Quang, N. T., & Binh, H. T. T. (2017, November). Semantic segmentation of objects from airborne imagery. In 2017 Fourth Asian Conference on Defence Technology-Japan (ACDT) (pp. 1-6). IEEE.