COMP9517 Group Project team: puush

Chengpeng Yang
Faculty of Engineering
University of New South Wales
Kensington, Australia
z5455913@ad.unsw.edu.au

2nd Wenjing Wang
Faculty of Engingeering
University of New South Wales
Kensington, Australia
z5522320@ad.unsw.edu.au

3rd Yangfangyuan Zhao
Faculty of Engingeering
University of New South Wales
Kensington, Australia
z5543360@ad.unsw.edu.au

4th Zelong Hu
Faculty of Engineering
University of New South Wales
Kensington, Australia
z5507000@ad.unsw.edu.au

5th Huayang Xie
Faculty of Engineering
University of New South Wales
Kensington, Australia
z5393197@ad.unsw.edu.au

***Abstract*—Semantic segmentation in natural environments poses significant challenges for autonomous vehicles as there are many irregular and unstructured scenes. More adaptable and reliable solutions are required since traditional datasets and algorithms frequently fall short in addressing these difficulties. This study explores the use of ensemble methods to improve segmentation accuracy for autonomous vehicle navigation.**

1) **Use ensemble methods to enhance segmentation accuracy for autonomous vehicle navigation**
2) **Combine Fully Convolutional Networks (FCNs) and DeepLabV3 models, both with ResNet-50 backbones.**
3) **Employ the WildScenes dataset, featuring diverse and annotated natural scenes**
4) **Train FCN and DeepLabV3 models independently and integrate using averaging, voting or other techniques**
5) **Demonstrated the potential of ensemble methods when facing complex natural environments**

Implications:

1) ***Draws attention to how crucial model integration and data augmentation are to enhance the reliability of computer vision systems.***
2) ***Encourages the use of sophisticated segmentation methods for autonomous navigation in real-world application.***

I. Introduction

A. Background

Autonomous vehicles reason and navigate in highly irregular, natural, unstructured environments, a very challenging problem. In such an environment, perception and decision-making turn pretty complex due to the variation of terrain. On this note, fine-grained semantic segmentation forms a very important module within scene understanding, since it allows vehicles to recognise their surroundings by classifying every pixel into predefined classes. Therefore, a critical capability if autonomous systems are to ensure safety and efficiency.

B. Challenges

These are environments that thus pose challenges difficult to be addressed by perception systems. Current traditional datasets and algorithms often fail to meet all the complexities natural settings usually pose. For example, [1] showed that semantic segmentation in snowy weather conditions is challenging due to loss of accuracy conventional ways suffer from. Moreover, [2] report considerable performance variations of segmentation models across different environments, requiring more flexible solutions.

C. Recent Advancements

These have, in recent times, revolved around ensemble and hybrid approaches. [3] probed the use of superpixels in conjunction with colour and texture features, successively achieving better segmentation in unstructured terrains. Unlike above, [4] used semantic mapping to show its applicability in differentiation between different types of terrain, thus enhancing vehicle understanding of the environment. In addition, models such as SFNet-N and MAPC-Net, using the power of multiscale feature fusion and synthetic data, form an important milestone toward more reliable and accurate semantic segmentation within complex natural scenes [5], [6].

D. Objective of the Study

In this paper, we focus on improving the accuracy of semantic segmentation using an ensemble method so that autonomous vehicles could perform all tasks such as navigation in various natural scenes. In this paper, we will fuse two models: Fully Convolutional Network (FCN) and DeepLabv3. Both of these models clearly show their superiority in handling images at high resolution and encode strong robust features. This strategy improves the accuracy of pixel-level annotation results, enhancing the robustness and reliability of the model.

In particular, it will go a step further to demonstrate quantifiable improvements in segmentation performance measured with rigorous metrics such as Intersection over Union. The projected outcome of this study has a very high potential to affect knowledge regarding the state-of-the-art applicability of computer vision techniques in real-world settings and natural environments.

E. Dataset

The WildScenes dataset [7], published by CSIRO in 2023, is a key asset for 2D and 3D semantic segmentation research

in natural environments. We will deal only with the 2D image data in this project, which contains five sequences of images during traversals through Venman National Park and Karawatha Forest Park in Brisbane, Australia. This dataset contains a wide variety of natural scenes, changing vegetation, types of terrain, and lighting conditions, which are very well suited to developing robust segmentation models. The dataset includes images of 2,016 $\times$ 1,512 pixels each; there are 9,306 in total, all annotated. These annotations encompass a broad spectrum of natural elements and hence provide ground truth data useful for training and testing some segmentation algorithms.

LITERATURE REVIEW

Relevant Techniques

A. Fully convolutional Networks(FCN)

1) Overview

According to Long et al. (2015), it is among the first deep learning models created especially for semantic segmentation. FCNs replaced fully connected layers with convolutional layers to transform conventional CNNs (convolutional neural networks) into end-to-end segmentation models. It can easily generate detailed, pixel-by-pixel predictions from input photos.

2) Architecture and Methodology

- Replace FC layers: to use convolutional layers instead of fully-connected layers
- Upsampling: To make sure that every pixel in the output matches to a pixel in the original picture, it then employed transposed convolution layers to upsample the coarse feature and map back to the original size of the image.
- Skip Connections: Segmentation may be improved by combining features from various network levels by integrating semantic information from deep layers with geographical information obtained through skipping connections.

3) Advantages

The ability to gather and predict pixel-by-pixel labelling made them appropriate for plenty of uses, including intelligent driving and medical imaging.

4) Limitations

- Coarse Predictions: The decrease of spatial resolution in deeper layers is linked with coarse predictions.
- Inability of multi-scale text: Incapable of to successfully capture multi-scale context

B. DeepLab v3

1) Overview

It is a semantic segmentation method that has been altered from Deeplabv2, its previous iteration. In terms of semantic picture segmentation today, it is also regarded as a SoTA (state-of-the-art) model. This method is very useful for complicated segmentation problems since it tackles the difficulties of keeping spatial resolution and collecting multi-scale context.

2) Architecture and Methodology

- Backbone Network: The model uses a backbone network for extracting features from the input image. Typically, the backbone choices include ResNet. To maintain a spatial resolution, atrous convolutions are introduced in following phases.
- Atrous Convolution: This convolution introduces dilations into the kernel. The method allowed the network to control the field-of-view of each filter with less redundancy including computational costs or unnecessary parameters.
- Atrous Spatial Pyramid Pooling(ASPP): ASPP builds a pyramid of filters by applying parallel atrous convolutions with different dilation rates, which collect multi-scale contextual information.
- Batch Normalisation: This is employed throughout the whole architecture of Deeplabv3. The training process is stabilised and accelerated by lowering the internal covariate shift.
- Output Layer: The pixel-wise classification map is generated by the convolutional layer, which is the last layer in DeepLabV3. From a preset set of classes, a label is assigned to each pixel in the input image.

3) Advantages

- Reserving essential information: Able to efficiently gather multi-scale contextual information whilst maintaining spatial features
- Adaptability: Able to be adapted to various backbone networks, making it benefit from the strengths of different architectures.

4) Limitations

- Longer training times: this model takes a longer time and a significant amount of computational resources to train, especially on big datasets
- Need for larger training datasets: In order to achieve optimal performance, it often requires large datasets.
- Overfitting Risks: The model has a high capacity that may rise the possibility of overfitting especially in cases of training on smaller datasets.

C. ResNet-50

1) Overview

The Residual Networks(ResNet) architecture is one of the most commonly used deep learning models for imaging recognition applications. It is developed by He et al., solving the issue of disappearing gradients in very deep networks by introducing the notion of residual learning. For ResNet-50 in particular, it has 50 layers and balanced complexity with performance, making it an inevitable choice for various computer vision applications.

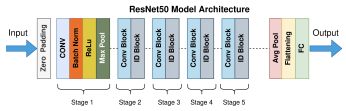

Fig. 1. ResNet-50 Model Architecture Model

2) Architecture and Methodology

- Residual Blocks: Instead of learning unreferenced functions, a residual block enables the model to learn residual functions concerning the layer inputs. Shortcut connections that elide one or more levels are used to accomplish this.

Bottleneck Design: In its residual blocks, the design reduced the number of parameters and computational complexity. Each block consists of 3 layers:

1) 1x1 Convolution: Reduces the number of channels of the input.
2) 3x3 Convolution: Processes the features with the reduced dimension.
3) 1x1 Convolution: Restores the dimension back to the original number of channels.

- Layers: It is composed of multiple stages, each with a stack of residual blocks whilst each stage increases the depth and complexity of the learned features:

1) Stage 1: 7x7 convolutional layer followed by a max pooling layer
2) Stage 2: 3 bottleneck blocks, each with 256 filters
3) Stage 3: 4 bottleneck blocks, each with 512 filters
4) Stage 4: 6 bottleneck blocks, each with 1024 filters
5) Stage 5: 3 bottleneck blocks, each with 2048 filters

- Activation Functions: ReLU(Rectified Linear Unit) activation functions are used after each convolutional and batch normalisation layer to introduce non-linearity.
- Glocal Average Pooling: It employs global average pooling to reduce the risk of overfitting along with the reduction of parameters.
- Final Classification layer: A dense layer follows the glocal average pooling layer to produce the final scores.

3) Advantages

- Ease of Training: It results in a faster convergence and a better performance as the residual connections helps to avoid disappearing gradient problem.
- Balanced Complexity: The architecture strikes a balance between the model complexity and computational feasibility by providing a better performance in accessing only the shallower networks when avoiding accessing the deeper variants.
- Fine-tuning: It is a nice choice for transfer learning as it can be fine-tuned for specific tasks with smaller datasets.

4) Limitations

- Complexity: Implementing and fine-tuning of ResNet-50 can be a rather complicated process as the users need to have a good understanding of the optimisation techniques.
- Memory Usage: As ResNet-50 requires high memory consumption, this can be a challenge when processing high-resolution images.

D. Ensemble

1) Overview

The Ensemble models is a technique using a combination of two or more individual model in order to achieve a higher accuracy and to improve overall performance. In semantic segmentation in particular, this technique can help to mitigate the weaknesses of using one single model.

2) Strategy

- Averaging: The predictions produced by the chosen models are averaged in order to decrease errors.
- Stacking: A meta-model is trained to combine the predictions from all models chosen to ascertain for the optimal during integration.
- Voting: From all selected models, we will use the one with the majority vote to determine the class label for each label.

3) Advantages

- Improved Accuracy: In most cases, the ensemble approach can obtain a better segmentation accuracy. As each model uses a different approach when extracting features, the ensemble approach may lead to a more reliable result because it is a combination.
- Enhanced Generalisation: The ensemble approach helps to prevent overfitting and improve the generalisation.
- Robustness: It is more resilient to potential variations. One model's forecast may make up for the other models' inability to properly segment a specific region, leading to more trustworthy and consistent performance.

4) Limitations

- Higher Memory Usage: Ensemble models often require more memory for storing data and processing segmentation tasks.
- Computational Complexity: When working with limited processing resources, the technique might be an obstacle because it increases computational needs.
- Inference Speed: The inference speed is relatively slower than comparing with using just one single model because the ensemble approach requires to generate and combine predictions from all models selected in the ensemble approach.

Necessary Background

A. Intersection over Union(IoU)

- Evaluation Metrics

IoU is a frequently used metric for evaluating the effectiveness of object detection and segmentation. The metric finds out the difference between the ground truth segmentation and the anticipated segmentation using a quantitative analysis between the two. Thus, the IoU metric can give us a relatively readable indication on how our models performed.

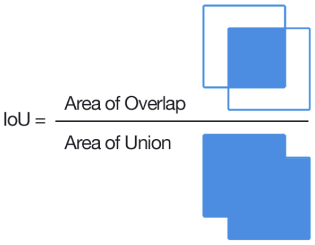

Fig. 2. IoU

B. Data Augmentation

- Importance of Data Augmentation

In computer vision, data augmentation is a critical technique for improving model performance, especially when working with smaller datasets. It includes generating additional training samples by applying different transformations to the original data. This enhances the model's resilience to various situations and it helps to reduce overfitting.

METHOD

1) FCN

F. Introduction

Fully Convolutional Network is one of the better known image segmentation models that can process input images of arbitrary size and output corresponding segmented images. Its innovative design is to replace the fully connected layers in traditional convolutional neural networks with convolutional layers, which in turn enables accurate pixel-level image segmentation. This innovation enables the model to learn and predict more effectively.

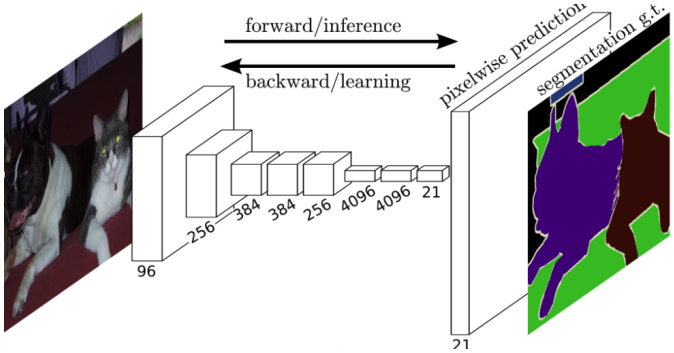

Fig. 3. FCN model explained

Model Modifications

In order to improve the performance of the FCN model in this project, we used a new classification header. This custom header consists of multiple convolutional layers, batch normalisation, ReLU activation function and Dropout, which changes the 2048 feature channels of the convolutional layer to 512, and we let the model use another convolutional layer to reduce the feature channels to the number of categories we want to target, in order to achieve the direct output of the category prediction for each pixel that we want to achieve. After our series of operations the model is able to extract features from the image and perform the classification task more efficiently.

Reason for Modification

The traditional FCN model still lacks the ability to process features relative to our modified model, although it also has the ability to process complex backgrounds or multiple categories of images. So we set up a "MyHead" classifier, which enables the model to learn the complex features in the task more efficiently, thus improving the accuracy of image segmentation and the operation efficiency of the model.

Advantage

The improved classifier by the team has made the FCN model more versatile for many image segmentation tasks, be it forests or kiosks in the dataset or anything else. Moreover, batch normalization and Dropout have been fitted into the model to ensure better generalization and avoid overfitting. We also use a pre-trained model as the base, which will reduce our computation resources and thus increase our efficiency.

2) DeeplabV3

Introduction

Deeplabv3 is a state-of-the-art convolutional neural network architecture specifically designed for image semantic segmentation, and one of its core is Atrous Convolution, which enables the expansion of the perceptual domain to obtain a wider range of contextual information without decreasing the resolution of the image. So Deeplabv3 is more suitable for this project. Deeplabv3 also includes an improved Atrous Spatial Pyramid Pooling (ASPP) module, which allows the model to handle image inputs at different scales.

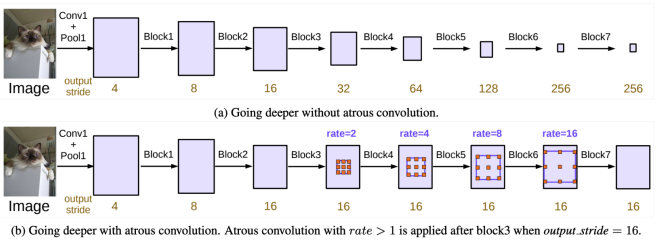

Fig. 4. DeepLab v3 model explained: Cascaded modules without and with atrous convolution

Modification of the Model

To meet the specific application requirements of this project, we made significant modifications to the DeepLabV3 model. Firstly, we replaced the classifier head of the model and adjusted it from its original design, which was oriented towards generic classification tasks, to a task-specific multi-class classification setup. This was achieved by introducing a new `DeepLabHead`, configured with 2048 input channels and a predefined number of output categories, ensuring that the

model can perform accurate segmentation directly on specific categories.

Reason for the Modification

Considering the specificity of different image segmentation tasks—for example, different types of cellular images or satellite images featuring various landforms may require distinguishing more detailed categories—the traditional DeepLabV3 output category settings may not suffice. Therefore, by customizing the classification head, we allow the model to be more flexible and adaptable to various specific application scenarios, while maintaining its efficient learning capabilities and delivering accurate segmentation results.

Advantage

Utilizing the pre-trained ResNet50 as a base, DeepLabV3 not only inherits the powerful feature extraction capability of the deep network but also effectively extends the sensory field through the atrous convolution technique. This enables the model to recognize and process more complex image structures.In addition, the pre-training model can effectively reduce the training cycle, because it does not have to start training from scratch every time. It significantly reduces the computational and time costs of the project and improves the efficiency.

3) Integration Model

Introduction

We have developed a novel integrated model that combines the advantages of two powerful segmentation models, FCN and DeepLabV3. The integrated model can effectively improve the accuracy and robustness of the image segmentation task, and this integrated model combines the advantages of two independent models to provide more accurate predictions for the project to some extent.

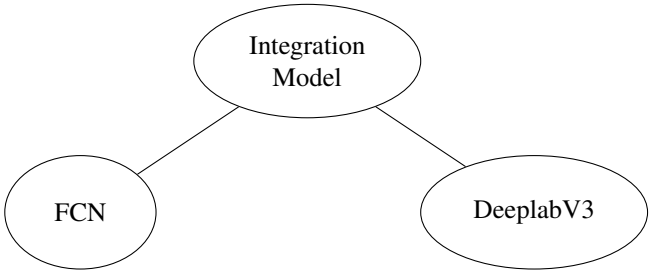

Fig. 5. Integration Schematic

Modification of the Model

In the integrated model, we fused the outputs of the FCN and DeepLabV3 models. by summing the output feature maps of the two models. Subsequently, the fused features are mapped to the desired number of categories by a 1x1 convolutional layer, thus allowing the model to directly classify directly by combining features.

The integrated model is effective in reducing the dependence of a single model on specific features, while the decision-making process can be diversified by adjusting the weights, thus improving the final accuracy.

Strengths

The integrated model enhances its own adaptability to various segmentation scenarios by integrating features learnt from different models. At the same time, the integration strategy provides a degree of fault tolerance, which means that one of the components in the integrated model performs poorly under certain conditions and the model maintains a degree of high performance. This fault tolerance relies on the outputs of the other models to reduce the degree of impact, thereby stabilising and improving overall performance.

EXPERIMENTAL RESULT

Experiment Overview

In this experiment, we used three different learning models, FCN, DeepLabV3, and ensemble learning strategies. Next, we will analyze their training results one by one.

Experimental Design

In this experiment, we used three different learning models, FCN, DeepLabV3, and ensemble learning strategies. Next, we will analyze their training results one by one.

Deep Learning Architectures

When designing the experiment, we chose the two most popular and commonly used models, FCN and DeepLabV3, both of which are based on the ResNet50 model to solve the problem of semantic image segmentation. Finally, on this basis, we integrated the training results of the two models to build an innovative integrated model that combines the advantages of FCN and DeepLabV3 to improve the performance of the model.

The DeepLabV3 model is initialized with a pre-trained ResNet50, which can make the model's extraction ability stronger. We set the classifier head to "DeepLabHead" with 2048 input channels and dynamically set it to output "num_classes" to better handle a large number of data labels.This model was trained using the Adam optimizer with a learning rate of 1×10^{-4} and weight decay of 1×10^{-6}, over 10 epochs. Training and validation processes were managed using a custom 'ModelTrainValid' class, designed to facilitate efficient model evaluation.

The FCN model also utilized a pretrained ResNet50 backbone, adapted with a specialized head for image segmentation that reduces feature channels to an intermediate dimension before classification. Parallelly, an ensemble model was developed by integrating the FCN and DeepLabV3 outputs before final classification. This ensemble approach uses a convolution layer to combine and refine features from both base models, aiming to enhance segmentation accuracy.

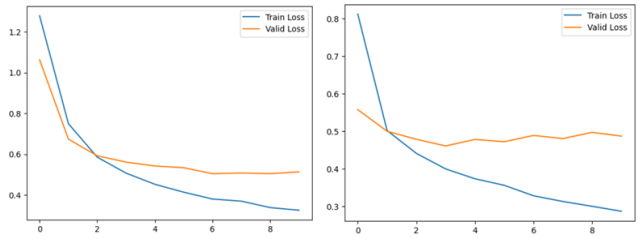

Fig. 6. Figure x Loss Trends of FCN in Two Phase

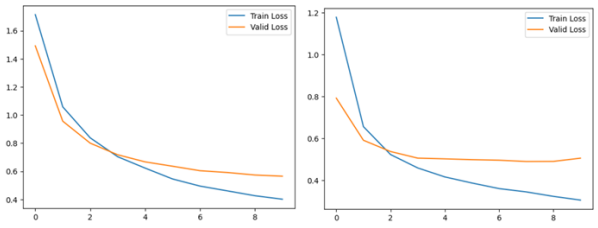

Fig. 8. Figure x Segmentation Performance of FCN in Two Phase

Result and Performance Analysis

FCN and DeepLabV3 Model Performance

Phase One: Training Loss demonstrated a sharp decline, indicating rapid learning, and stabilized as epochs increased. Validation Loss began close to the training loss but plateaued, suggesting the model's satisfactory generalization from training data to unseen validation data.

Phase Two: Training Loss in Phase Two followed a similar sharp decrease but leveled off at a lower value compared to Phase One, suggesting enhanced learning efficiency and model optimization. Validation Loss showed less fluctuation in Phase Two and maintained a closer gap with the training loss, indicating improved model consistency and generalization ability.

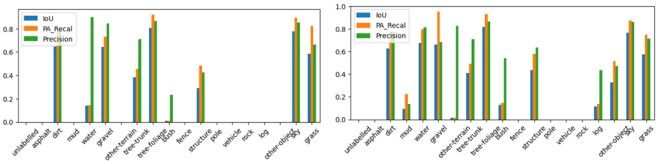

Fig. 7. Figure x Segmentation Performance of FCN in Two Phase

High-Performing Classes

Classes such as 'tree-foliage', 'water', and 'sky' showed notable improvements in IoU and precision from Phase One to Phase Two, particularly 'tree-foliage' which maintained high performance, suggesting the model's improved capability in capturing complex features with more training data or refined model tuning.

Moderately Performing Classes

Classes like 'gravel', 'other-terrain', and 'grass' exhibited improvements in both metrics, indicating better segmentation capability and classification accuracy.

Poorly Performing Classes

Classes such as 'pole', 'vehicle', and 'fence' struggled significantly with very low IoU and PA_Recall, particularly for 'pole' and 'vehicle'. While still underperforming, there was a slight improvement in these metrics for 'fence', showing perfect precision but still low recall, suggesting the model can identify these objects when they are present but often fails to detect them.

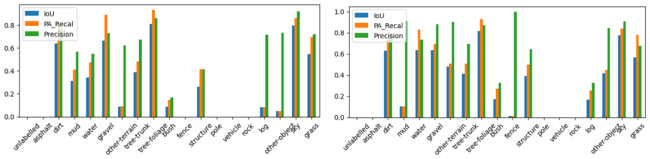

Fig. 9. Figure x Segmentation Performance of Deeplab3 in Two Phase

DeepLabV3 Specific Analysis

Phase One and Two: Both training and validation losses decrease significantly at the beginning, with the training loss tapering to a lower steady state compared to the validation loss, which suggests initial rapid learning followed by stabilization. The validation loss eventually plateaus, indicating the model's capability to generalize, though slight overfitting may be suggested by the divergence from the training loss.

Performance in Key Classes

Tree Foliage and Sky consistently show high IoU and Precision in both phases, with slight improvements in Phase Two. This indicates that DeepLabV3 effectively captures and segments these dominant features across different datasets.

Water and Gravel show substantial improvements in all metrics from Phase One to Phase Two, which could be due to enhanced model training or more representative data in the latter phase. Poorly Performing Classes like Pole, Vehicle, and Rock remain challenging, with minimal to no detection in both phases. This could indicate inherent limitations in the model's architecture in capturing such features or insufficient training examples for these categories.

Classes with Declined Performance: Some classes such as Dirt and Mud show a slight decline in performance metrics from Phase One to Phase Two. This might be due to the model overfitting on other more dominant features in the dataset or variations in the class balance between the two datasets.

Ensemble Model

Finally, we integrated the previously trained FCN and Deeplabv3 models, and also conducted two-stage experiments on the fused model, which facilitated our analysis of the robustness of the fused model.

From the results, the first-stage training loss decreased rapidly in the initial stage, indicating that the integrated model effectively learned from the dataset. However, the validation

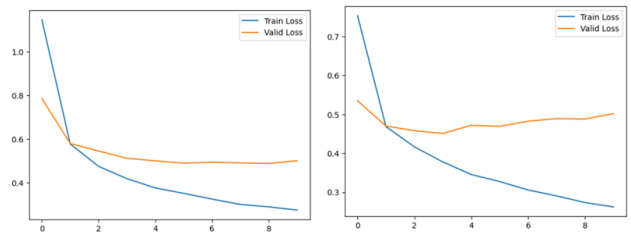

Fig. 10. Figure x Segmentation Performance of Deeplab3 in Two Phase

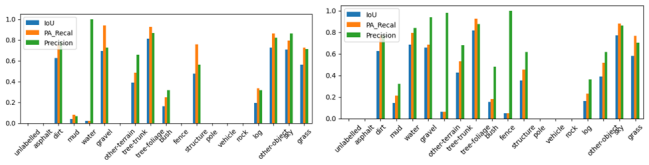

Fig. 11. Figure x Segmentation Performance of Ensemble Model in Two Phase

loss stabilized at a higher value relative to the training loss in the second epoch, indicating that the model learned overfitting.

The second stage showed a similar trend, with the training loss initially dropping sharply, but both the training and validation losses reached a lower steady-state value than the first stage.

Improvement in Key Classes

- **Dirt, Gravel, and Tree Foliage** have shown significant improvement in IoU and Precision from Phase One to Phase Two. Particularly, Gravel and Tree Foliage maintained high performance metrics, demonstrating the ensemble's ability to consistently segment these complex textures across phases.
- **Water** displayed a remarkable improvement in IoU, making it one of the best-segmented classes in Phase Two. This improvement reflects the ensemble's enhanced capability to differentiate water from other similar textures.

Classes with Variable Performance

- **Bush and Other-Object** classes saw fluctuating IoU and Precision. This variability might be due to the ensemble's differing strategies in phase adaptation or variability in class representation between the two datasets.
- **Fence and Pole** remain challenging, with minimal improvement in IoU or Precision, suggesting that these features are still hard for the ensemble to detect, possibly due to their thin and linear nature which might be underrepresented in the training data.

DISCUSSION

A. Discussion on FCN

1) Overview of the model

In this project the FCN model is optimised on demand to suit the needs of the image segmentation task. The core change is the introduction of a custom classifier called MyHead, which is designed to replace the end classification layer in the standard FCN model. The main purpose is to enhance the model's adaptability and performance for specific problems.

2) Custom Classifier Header: The model uses a class of `MyHead`, a custom neural network module, which is used to replace the original classifier of the FCN model. This classifier contains the following layers:

- Convolutional layer: first a convolutional layer is used with 512 output channels, 3x3 convolutional kernels with a step size of 1 and a padding of 1. This layer is designed to extract richer features from the feature map.
- Batch Normalisation Layer: followed by a batch normalisation layer to speed up the training process and improve the generalisation of the model.
- Activation Layer: activation functions are used to introduce non-linearities that help the network to learn complex patterns.
- Dropout layer: a dropout ratio of 0.1 was used to reduce overfitting.
- Output Convolutional Layer: the last convolutional layer converts 512 channels into `num_classes` of channels, each representing the prediction results of one class.

3) Model Training Configuration

- Optimiser Configuration: the Adam optimiser was used with the learning rate set to 1×10^{-4} and the weights decayed to 1×10^{-6}, which helps to better tune the network weights and avoid overfitting.

4) Challenge Discussion and Optimisation Strategies

- Computational complexity: As convolutional operations are generally computationally intensive, they lead to low efficiency and require optimisation of computational strategies or utilisation of high-performance computing resources.
- Overfitting problem: Although Dropout has been used, the model still faces overfitting problems under limited data conditions. It can be solved by further data enhancement or introducing more sophisticated regularisation techniques.

B. Discussion on Deeplabv3

1) Overview of the model

In this project, the pre-trained `deeplabv3_resnet50` is used as a base, this structure integrates ResNet50, further enhances the sensory field using the atrous convolution technique, and optimises the model performance by modifying the classifier part of the network through DeepLabHead to accommodate a specific number of output categories.

2) Model Training Configuration

The model configuration uses the Adam optimiser with the learning rate set to 1×10^{-4} and weight decay to 1×10^{-6}. This helps to balance the need for rapid convergence of the model during learning with the need to prevent overfitting.

3) Challenge Discussion and Optimisation Strategies

- Computational complexity: while atrous convolution improves the sensory field of the model, it also increases the computational burden. Optimisation strategies include using more efficient hardware, performing model pruning, or training with mixed precision (more efficient hardware was used in this experiment).
- Hyperparameter tuning: the effectiveness and efficiency of model training can be improved by further adjusting the learning rate, batch size or optimiser parameters.

C. Discussion of FCN's Integration Model with Deeplabv3

1) Overview of the model

The integrated model in the project is optimised for the image segmentation task by integrating FCN and Deeplabv3, two segmentation models based on ResNet50 and pre-trained. Both models use a modified classifier header:

- The FCN model uses FCNHead with 2048 input channels and 128 output channels, which is the convolutional layer used to generate predictions from the feature map.
- The Deeplabv3 model uses DeepLabHead, also with 2048 input channels and 128 output channels, which employs atrous convolution to improve resolution and the ability to capture contextual information.

2) Integrated Strategy

The integrated model generates the final predicted output by summing the outputs of FCN and Deeplabv3 and then further processing the fused feature maps through a convolutional layer (nn.Conv2d). This approach takes advantage of the complementary strengths of the two models in capturing different features of the image and obtains more accurate results than a single model through integration.

3) Model Training Configuration

The model is configured with an Adam optimiser, with the learning rate set to 1×10^{-4} and the weight decay to 1×10^{-6}. This helps to optimise the training process, balancing the speed of learning with the model's ability to generalise.

4) Training process

The ModelTrainValid class is used to manage the training, validation, and testing of the model, ensuring that the model is trained efficiently on multiple batches of data and that the performance is progressively optimised through multiple cycles of learning. The process not only focuses on loss reduction but also monitors the model's performance on validation data to adjust the training strategy and avoid overfitting.

5) Challenges and optimisation strategies

- Integration complexity: While model integration can improve performance, it also increases model complexity and computational requirements. Optimising computational resources and efficiency is the key to achieving fast training.
- Hyperparameter tuning: reasonable hyperparameter selection has a significant impact on model performance. More experiments may be needed to find the optimal learning rate and weight decay.

CONCLUSION

In this paper, we detail the semantic segmentation task in the project and explore various ways to improve the performance of semantic segmentation by comparing different models. We selected and processed the K-01 and K-03 subsets of the WildScenes dataset for counting and loading image and labeled data, implemented a custom dataset class, `K13Dataset`, for data preprocessing and enhancement. We used two models - the FCN model with ResNet50 as the backbone network and the DeepLabv3 model - and developed a combination of the two as a new third integrated model, `EnsembleModel`. We trained all three models, recording loss curves and validation. In model evaluation, we analyzed in detail the performance of each model on the test set, calculated IoU, PA_Recall, and Precision for each category, and compared the differences in model performance through bar graph visualization.

Limitations

Our project still has some shortcomings, such as the size and usefulness of the dataset. In order for self-driving vehicles to navigate safely and accurately in natural environments, we need a more diverse dataset. Currently, there are more samples in some categories and fewer samples in others, which may lead to poor learning of the model on some categories, as evidenced by lower IoU, PA_Recall, and Precision metrics for some categories, especially when dealing with rare or difficult-to-segment categories (e.g., fence, pole, vehicle). In addition, although the integrated model improves in performance, the computational overhead is large, which limits its efficiency in practical applications.

Future Directions

In the future, we plan to expand and diversify our dataset, specifically by including rare categories such as railings and vehicles, which are currently lacking in our experimental data. This will help improve the model's performance in these categories. Additionally, we will explore more complex and effective ensemble strategies, such as weighted feature fusion and multimodal feature integration, to fully leverage the strengths of each base model. We will also enhance the application of regularisation techniques to prevent model overfitting and consider other evaluation metrics, such as the F1-score, for a more comprehensive performance assessment. Most importantly, we will test the model in more realistic scenarios, such as different lighting conditions, weather conditions, and natural environments, to ensure the model's robustness and practical applicability. Through these improvements, we aim to further enhance the model's practical value, providing a safer and more reliable comprehensive solution for autonomous driving in natural environments.

REFERENCES

[1] Z. Pan, T. Emaru, A. Ravankar, and Y. Kobayashi, "Applying semantic segmentation to autonomous cars in the snowy environment," 2020. [Online]. Available: https://arxiv.org/abs/2007.12869

[2] W. Zhou, J. S. Berrio, S. Worrall, and E. Nebot, "Automated evaluation of semantic segmentation robustness for autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, p. 1951–1963, May 2020. [Online]. Available: http://dx.doi.org/10.1109/TITS.2019.2909066

[3] I. Gheorghe, W. Li, T. Popham, and K. J. Burnham, "Superpixel based semantic segmentation for assistance in varying terrain driving conditions," in *Progress in Systems Engineering*, H. Selvaraj, D. Zydek, and G. Chmaj, Eds. Cham: Springer International Publishing, 2015, pp. 691–698.

[4] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Proceedings of 11th International Conference on Field and Service Robotics (FSR '17)*, September 2017, pp. 335 – 350.

[5] H. Wang, Y. Chen, Y. Cai, L. Chen, Y. Li, M. A. Sotelo, and Z. Li, "Sfnet-n: An improved sfnet algorithm for semantic segmentation of low-light autonomous driving road scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21 405–21 417, 2022.

[6] X. Zhou, Y. Feng, X. Li, Z. Zhu, and Y. Hu, "Off-road environment semantic segmentation for autonomous vehicles based on multi-scale feature fusion," *World Electric Vehicle Journal*, vol. 14, no. 10, p. 291, 2023.

[7] K. Vidanapathirana, J. Knights, S. Hausler, M. Cox, M. Ramezani, J. Jooste, E. Griffiths, S. Mohamed, S. Sridharan, C. Fookes, and P. Moghadam, "Wildscenes: A benchmark for 2d and 3d semantic segmentation in large-scale natural environments," 2023.

[8] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017. [Online]. Available: http://arxiv.org/abs/1706.05587

[9] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1605.06211, 2016. [Online]. Available: http://arxiv.org/abs/1605.06211

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.