NTU CE6190 Assignment 1: Semantic Segmentation using Attention U-Net

Seow Wei Liang
G2205192K
WEILIANG003@e.ntu.edu.sg

Abstract

In this paper, we will experiment with semantic image segmentation using a public dataset (CamVid). We will use an improved U-Net architecture, Attention U-Net, with a pretrained VGG19 encoder network for semantic segmentation with batch normalization. We conduct an analysis of the critical hyperparameters settings (e.g. learning rate, dropout and weight decay). To tackle the class imbalance problem in semantic segmentation, we have explored different loss functions such as cross entropy loss, tversky loss, focal tversky loss, focal loss, dice loss as well as a unified focal loss (focal loss and focal tversky loss) and found that unified focal loss works best for Attention U-Net Semantic segmentation. We performed an ablation study based on the Attention U-Net model architecture using a pretrained VGG19 encoder network. We compared different encoder architectures with frozen and unfrozen backbone and compare the results with VGG19 unfrozen backbone achieving best results. In addition, we also compared with our Attention U-Net implementation with other U-Net architectures (Vanilla U-Net, Residual U-Net and Residual Attention U-Net). We also compared the results between U-Net with the FCN-8, SegNet and DeepLabV3+ for semantic segmentation of CamVid dataset.

1. Introduction

This project focuses on semantic segmentation which is a subcategory of image segmentation. Image segmentation involves grouping or labeling similar regions or segments in an image on a pixel level. Image Segmentation can be divided into three main types which are semantic segmentation, instance segmentation and panoptic segmentation.

Semantic segmentation identifies both stuff and things within an image and predicts an unique class label for each image pixel based on multi-class classification. However, semantic segmentation predictions cannot differentiate instances of the same category within an image. In contrast to semantic segmentation, Instance segmentation typically deals with tasks related to countable things. It can detect each instance of a class present in an image and assigns it a different mask. Panoptic segmentation unifies both techniques from semantic segmentation and instance segmentation which is able to predict a semantic label for each pixel (due to semantic segmentation) and a unique instance identifier (due to instance segmentation).

In this paper, we will focus mainly on semantic image segmentation and we will use a public dataset, CamVid. We will use a modified U-Net architecture, Attention U-Net [12] using VGG19 [16] encoder network with batch normalization [8] for semantic segmentation. We will study the effect of different hyperparameter settings on the mean IoU and performed an ablation study on the attention U-Net. To tackle the class imbalance problem in semantic segmentation, we have explored different loss functions such as cross entropy loss, tversky loss [14], focal tversky loss [1], focal loss [10], dice loss [19] as well as a unified focal loss [17] (focal loss and focal tversky loss). We compared different encoder architectures with frozen and unfrozen backbone and compare the results. In addition, we also compared our Attention U-Net implementation with other U-Net architectures (Vanilla U-Net [13], Residual U-Net [18] and Residual Attention U-Net [11]). We also compared the results for U-Net based architecture with the Fully Convolutional Neural Network (FCN-8) [15], SegNet [2] and DeepLabV3+ [4]for Semantic Segmentation of CamVid dataset.

2. Related Work

Related work in this project includes U-Net Architecture, attention and class imbalance.

2.1. U-Net Architecture

The architecture of U-Net [13] consists of a downsampling path and a upsampling path. The downsampling path contains encoder layers that capture contextual information and reduce the spatial resolution of the input, while the upsampling path contains decoder layers that decode the encoded data with the help of information from downsampling path via skip connections to generate a segmentation map.

The purpose of downsampling path in U-Net is to capture increasingly abstract representations of the input image. The encoder layers perform convolutional operations to reduce the spatial resolution of the feature maps while simultaneously increasing the number of channels. On the other hand, the upsampling path decode the encoded input image and upsample the feature maps, while also performing convolutional operations. The skip connections from the downsampling path help to preserve the spatial information lost during downsampling of input image, which helps the decoder layers to locate the features more accurately during upsampling.

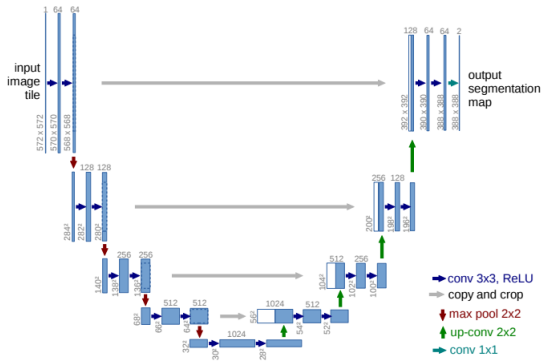

Figure 1. A schematic of U-Net Architecture

2.2. Hard vs Soft Attention

In the section, we explained two different types of attention, Hard versus Soft attention. Hard attention highlight relevant regions leveraging on image cropping or iterative region proposal. One region of image is computed at a time which makes it non-differentiable and requires techniques such as reinforcement learning for training. Due to its non-differentiable nature, standard backpropagation used in neural networks cannot be used and Monte Carlo sampling is required to compute the accuracy across various stages of backpropagation. On the other hand, soft attention focus on different parts of the image with relevant parts of the image having larger weights and irrelevant parts of the image with lower weights. In contrast to hard attention, soft attention can be trained with standard backpropagation allow the attention weights to be updated during training, thereby allowing the model to focus on the more relevant regions. In our study, Attention U-Net utilize soft attention that can be incorporated using Attention gates which will be elaborated subsequently.

2.3. Attention gates

Attention gates (AGs) proposed in [12] can be integrated to a standard CNN model to suppress irrelevant features in background regions of the image. It does not necessitate multiple model training and large number of model parameters. Attention coefficients calculated from Attention gates helps to focus on important image regions preserving only the activations relevant to the specific task, removing irrelevant feature responses. As shown in Fig. 2, Attention gates takes two inputs, a gating signal, g, from the lower decoder block with better feature representation and skip connection, x, from the encoder at the same level of the decoder with better spatial information. These two inputs are convolve using 1x1 convolution to achieve the same dimensions before combining them. The aligned weights from the two inputs gets relatively larger while unaligned weights diminished. The output is sent to the relu activation function followed by another convolution operation and a sigmoid activation function to obtain the weights. The weights are upsampled and multiplied element wise with the input x from the skip connection. This allows the decoder to be able to focus on relevant portions of the input originating from skip connection from the encoder.

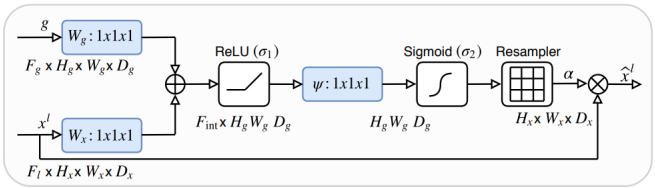

Figure 2. A schematic of Attention gate in Attention U-Net Architecture

2.4. Class Imbalance

Class imbalance refer to the problem of imbalance in the number of classes during model training which typically manifest in classification tasks. Image segmentation is particularly prone to class imbalance problem due to some classes having pixels that are under-represented in the training data. To tackle this problem, we explore different loss functions such as tversky loss, focal tversky loss, focal loss, dice loss and unified focal loss (both focal loss and focal tversky loss) to mitigate this class imbalance problem. We found that unified focal loss works best among all loss function explored.

3. Methodology

Methodology used in this project includes Attention U-Net model and loss function modifications to tackle class imbalance.

3.1. Attention U-Net

In attention U-Net [12] architecture as shown in Fig. 3, Attention gates (AGs) are incorporated into the standard

U-Net architecture to focus on important features that are passed through the skip connections in Attention U-Net. Information extracted from coarse scale is used in gating to filter out irrelevant and noisy responses in skip connections. This operation is performed immediately before the concatenation operation to merge only relevant activations. Additionally, AGs filter the neuron activations during the forward propagation as well as during the backpropagation. Gradients belonging to background regions are down weighted during the backpropagation. This allows model parameters in shallower layers to be updated based on relevant spatial regions to a given task.

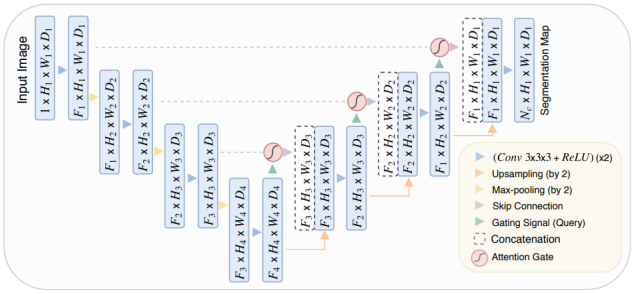

Figure 3. A schematic of Attention U-Net Architecture

3.2. Loss functions to tackle class imbalance

We explored different loss functions to tackle class imbalance. Loss functions can be classified into distribution-based losses, region-based losses and compound losses which will be explain in detail in this section. Distribution-based loss include cross entropy loss and focal loss. Region-based losses include dice loss, tverksy loss and focal tverksy loss. Compound loss method include the unified focal loss which merge both distribution-based loss and region-based loss methods.

Categorical Cross-Entropy loss is traditionally used in classification tasks. As the name implies, the basis of this is Entropy. In statistics, entropy refers to the disorder of the system. It is represented in Eq. (1) with p_i as the probability of event A happened. It is number of bits that we need to encode the information of event A that happened. Cross entropy is defined as number of bits we need to encode the information with an encoding scheme under the assumption that the probability distribution is P but actual distribution is Q. It is represented in Eq. (2) with Y_i as ground truth label and p_i is the prediction probability. Cross entropy loss is also used in semantic segmentation but does not take into account the class imbalance.

$$Entropy = -\sum_{i=1}^{i=n} p_i log_2(p_i) \tag{1}$$

$$Cross\_entropy = -\sum_{i=1}^{i=n} Y_i log_2(p_i) \tag{2}$$

Focal loss [10] is a modification of the cross entropy loss which lowers the loss for the easy examples and focus the training on the hard examples. Focal loss as shown in Eq. (3) has two hyperparameters α and γ. α is the class weights of each label.

$$FL = -\sum_{i=1}^{i=n} \alpha_i (i - p_i)^{\gamma} log_b(p_i) \tag{3}$$

The Dice coefficient in Eq. (4) is commonly used metric in computer vision community to calculate the similarity between two images, A and B. Dice coefficient is two times the intersection between the ground truth and the predicted mask, divided by the sum of the ground truth and the predicted mask. Subsequently, it has also been adapted as loss function known as Dice Loss [19] as shown in Eq. (5).

$$Dice = \frac{2 * |A \cap B|}{|A| + |B|} \tag{4}$$

$$Dice\_loss = 1 - Dice \tag{5}$$

Tversky index (TI) also known as a generalization of Dice coefficient. As shown in Eq. (6), it is weighted by the constants α and β that penalise False Positives (FP) and False Negatives (FN) respectively. In Eq. (6), TP refers to the True Positive. Taking 1 minus Tversky index will yield the Tversky loss [14] as shown in Eq. (7) similar to dice loss.

$$TI = \frac{TP}{TP + \alpha FN + \beta FP} \tag{6}$$

$$Tversky\_loss = 1 - TI \tag{7}$$

Similar to Focal Loss, which focuses on hard example by down-weighting easy/common ones. Focal Tversky loss [1] borrow the idea from focal loss and also tries to learn hard-examples such as those with small ROIs(region of interest) with the help of γ coefficient as shown in Eq. (8).

$$FTL = \sum_{c} (1 - TI_c)^{\gamma} \tag{8}$$

Unified focal loss [17] shown in Eq. (9) aims to combine both focal loss and Focal Tversky loss to improve segmentation results with weights given to Focal loss and Focal Tverksy loss based on their importance. In our experiments, we consider equal importance to Focal loss and Focal Tversky loss of α=0.5 and β=0.5.

$$UFL = \alpha FL + \beta FTL \tag{9}$$

4. Experiments

In this section, we will explain the dataset used, implementation details, discuss the effects of different optimization and regularization hyperparameters on our model performance, perform an ablation study to observe the effects of different techniques used.

4.1. Dataset

The dataset used in this challenge is CamVid [3]. CamVid (Cambridge-driving Labeled Video Database) is a road/driving scene dataset which captured as five video sequences with a 960×720 resolution camera mounted on the dashboard of a car. The original dataset consist of 32 classes but we utilized the dataset version with 12 classes consisting of 'Sky', 'Building', 'Pole', 'Road', 'Pavement', 'Tree', 'SignSymbol', 'Fence', 'Car', 'Pedestrian', 'Bicyclist' and 'Void'.

4.2. Implementation details

We implement our Attention U-Net by utilizing pretrained VGG19 using ImageNet [5] as the encoder for the model architecture and freeze the pretrained VGG19 layers. We use Adam [9] optimizer due to the adaptive learning rates by computing running average of the gradient and running average of the squared gradients. Adam is used with learning rate of 1e-4, weight decay of 5e-5, dropout of 0.01 used in decoder. We utilize a batch size of 8 and number of training epochs of 100. We use a learning rate scheduler to reduce the learning rate by a factor=0.1 on plateau by monitoring the validation accuracy with patience=10, min_lr=1e-6. We also use early stopping by monitoring the validation accuracy with patience=20 to prevent overfitting. We utilized the unified focal loss with equal weightings for focal tversky loss and focal loss. We used mean IoU described in Eq. (11) as a metric to evaluate the performance of the validation and test set. It is computed using True positive (TP), False Positive (FP) and False negative (FN) of each class to obtain the IoU for each class IoU_c in Eq. (10). The average of the IoU_c results in the mean IoU in Eq. (11).

We use 367 training, 233 test images and 101 validation images from the CamVid dataset for our experiments. Input image size is (224, 244, 3). Data augmentation was performed using keras image data generator on the training set with rotation_range=5, width_shift_range=0.1, zoom_range=0.1, height_shift_range=0.1, horizontal_flip=True, rescale=1./255. We use batch normalization [8] to speed up convergence by reducing internal covariate shift. Data whitening through batch norm can lower the condition number of the feature covariance matrix and improves the speed of convergence.

We will also use Fig. 4 as a basis to analyse the results of our experiments.

$$IoU_c = \frac{\#TP_c}{\#TP_c + \#FP_c + \#FN_c} \tag{10}$$

$$mean\_IoU = \frac{1}{c}\sum_c IoU_c \tag{11}$$

Figure 4. Test Image for Experimental Analysis

4.3. Effect of Learning Rate

In this section, we use the model with hyperparameters described in implementation details and tweak the learning rates. We explored different learning rates of 1e-3, 1e-4, 1e-5 and record the best mean_iou score for the validation set and test set. Learning rate of 1e-4 gives the best result as shown in the Tab. 1 and Fig. 5 which obtain the highest mean IoU of 0.5639 and 0.5046 for the validation and test set respectively. Having too low a learning rate will make the model reach the optima slower. Having too high a learning rate may cause the model to overshoot near the optima.

Learning Rate	Validation	Test
1e-3	0.5018	0.4504
1e-4	0.5639	0.5046
1e-5	0.4538	0.3953

Table 1. Effect of learning rate on mean IoU score

4.4. Effect of Dropout

In this section, we use the model with hyperparameters described in implementation details and tweak the dropout probability. We explored different dropout probability of 0.001, 0.01, 0.1 and record the best mean_iou score for the validation set and test set. As shown in Tab. 2 and Fig. 6, dropout of 0.01 gives the best result as shown in the table which obtain the highest mean IoU of 0.5639 and 0.5046

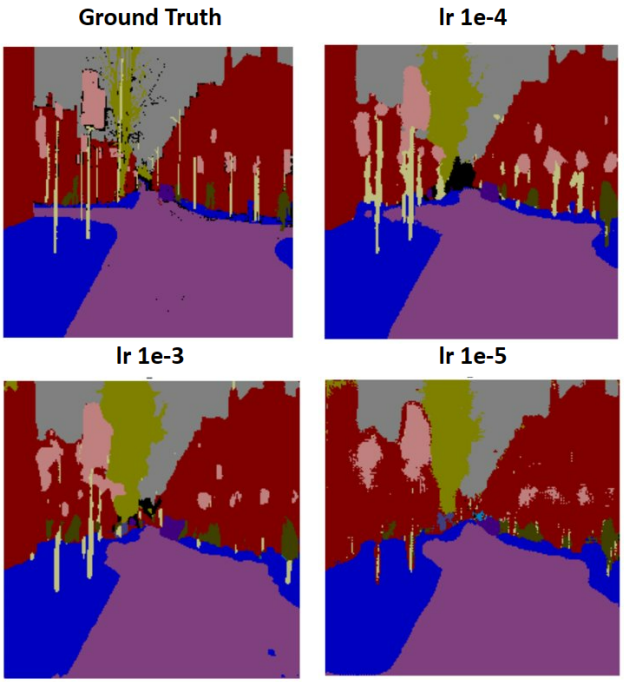

Figure 5. Effect of learning rate on CamVid segmentation

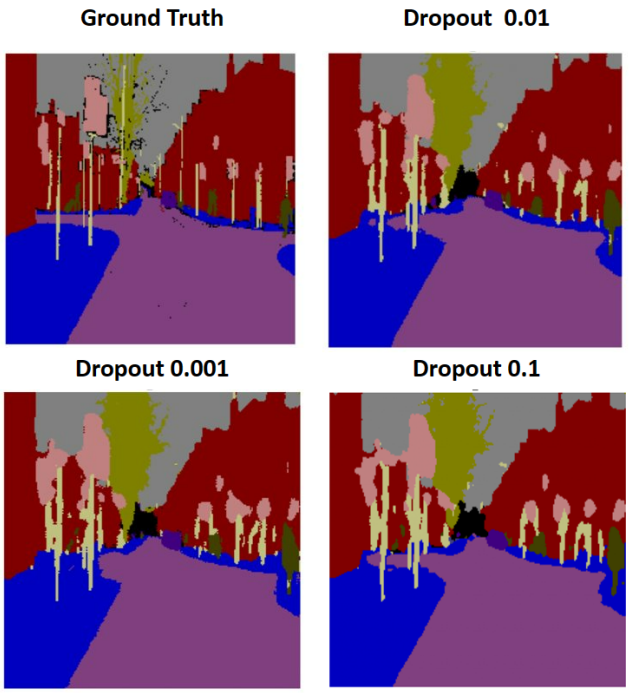

Figure 6. Effect of dropout on CamVid segmentation

for the validation and test set respectively. Having too high a dropout may result in the model having too little nodes activated resulting in a simple model with lower accuracy. Conversely, too low a dropout does not provide the model with sufficient regularization resulting in model overfitting.

Dropout	Validation	Test
0.001	0.5509	0.4827
0.01	0.5639	0.5046
0.1	0.5581	0.4914

Table 2. Effect of dropout on mean IoU score

4.5. Effect of Weight Decay

In this section, we used the model with hyperparameters described in implementation details and tweak the weight decay using an Adam optimizer. We explored different weight decay of 5e-4, 5e-5, 5e-6 for all tasks and record the best mean_iou score for the validation set and test set. As shown in Tab. 3 and Fig. 7, Weight decay of 5e-5 gives the best result as shown in the table which obtain the highest mean IoU of 0.5639 and 0.5046 for the validation and test set respectively. Having too high a weight decay may result in the model having smaller weights with poor predictive accuracy. Conversely, too low a weight decay does not provide the model with sufficient regularization and resulting in model overfitting.

Weight decay	Validation	Test
5e-4	0.5526	0.4918
5e-5	0.5639	0.5046
5e-6	0.5538	0.4786

Table 3. Effect of weight decay on mean IoU score

4.6. Effect of different loss functions

In this section, we used the model with hyperparameters described in implementation details and experiment with different loss functions to tackle the problem of class imbalance. For Tversky loss, we set α=0.7, β=0.3. For Focal Tversky loss, we set α=0.7, β=0.3, γ=0.75. For Focal loss, we set α=[0.125, 0.125, 2.0, 0.125, 0.25, 0.25, 1.0, 1.0, 0.25, 1.0, 1.0, 0.25] representing the class weights and γ=1.

We experimented with both distribution based loss and region based loss and unified losses which combines both distribution based and region based loss and compare the results in Tab. 4 and Fig. 8. Among the distribution based loss experimented which are cross entropy loss and focal loss, focal loss performed better with mean IoU of 0.5378 and 0.4749 for validation and test set respectively compared to the cross entropy loss with mean IoU of 0.4893 and 0.4270

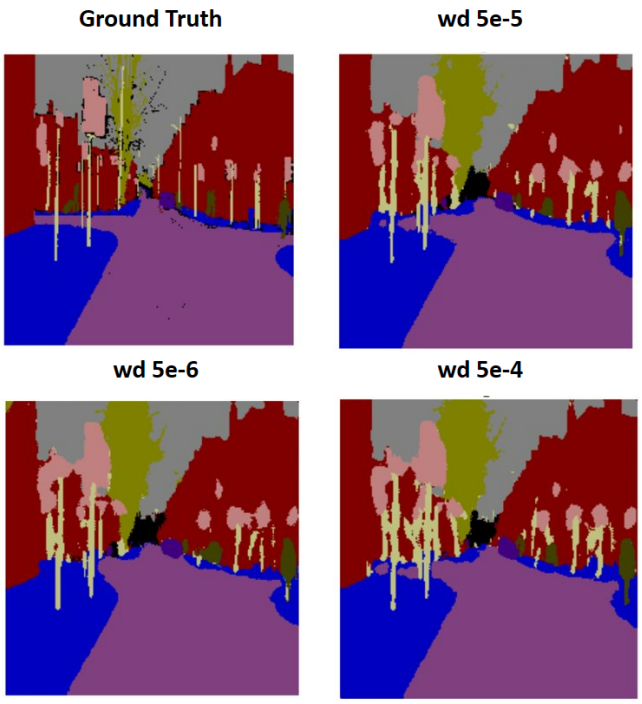

Figure 7. Effect of weight decay on CamVid segmentation

Loss function	Validation	Test
Dice loss	0.5384	0.4842
Tversky loss (α=0.7, β=0.3)	0.3784	0.3495
Focal Tversky loss (α=0.7, β=0.3, γ=0.75)	0.3800	0.3484
Cross Entropy loss	0.4893	0.4270
Focal loss	0.5378	0.4749
Unified Focal loss	0.5639	0.5046

Table 4. Effect of different loss functions on mean IoU score

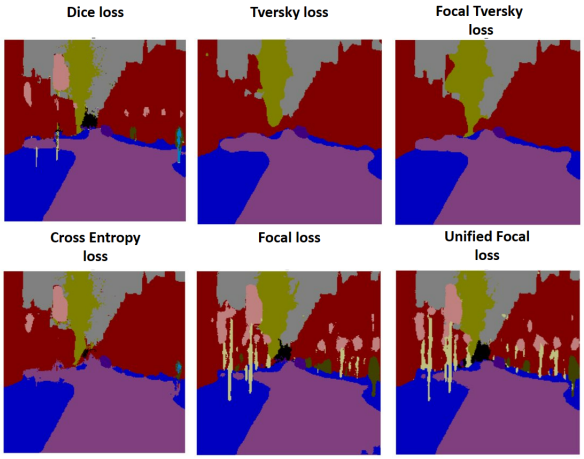

Figure 8. Effect of loss functions on CamVid segmentation

for validation and test set respectively which shows that the class weights applied and the focal parameter gamma helps to alleviate class imbalance and also focus the learning on the hard examples by down weighting the loss of the easy examples.

For the region based losses, Tversky loss and Focal Tversky loss does not perform as well compared to Dice loss which achieved mean IoU of 0.5384 and 0.4842 for validation and test set respectively. This results show that having unequal weights to penalize the false positive and false negative does not work as well compared to equal weightings for both false positive and false negative in dice loss.

Lastly, by combining both Focal loss (Region-based) and Focal Tverksy loss (Distribution-based) with equal weightings for both losses as a Unified Focal loss, we achieved the best performance of 0.5639 and 0.5046 for validation and test set respectively.

4.7. Ablation study

In this section, we used model with hyperparameters described in implementation details and performed an ablation study by removing the implementation of attention, batch normalization and VGG19 pretrained encoder network incrementally to study the effects on the mean IoU as shown in Tab. 5. With the removal of attention, mean IoU drops from 0.5639 to 0.5356 for the validation set and 0.5046 to 0.4665 for the test set. This shows that soft attention implemented in the Attention U-Net decoder plays an important role and helps to focus on important details originating from the skip connections from the encoder. Removing batch normalization results in mean IoU to fall from 0.5356 to 0.5182 for the validation set and 0.4665 to 0.4613 for the test set which shows batch norm has a lower impact on the mean IoU compared to attention but still helps to speed up convergence. Removing VGG19 pretrained encoder also results in a smaller drop in mean IoU for validation and test set compared to removing attention but larger drop compared to removing batch normalization as shown in Tab. 5. This proves that pretrained VGG19 weights on ImageNet plays an important role in better performance in mean IoU.

4.8. Comparison of different encoder architectures for Attention U-Net

In this section, we compared different encoder architectures (VGG16 [16], VGG19 [16], ResNet50 [6], DenseNet121 [7]) on the performance on the CamVid dataset with frozen and unfrozen encoder backbone and compare the results in Tab. 6. Overall, unfreezing the encoder backbone generally produce better results across

Techniques	Validation	Test
All	0.5639	0.5046
All-Attention (U-Net)	0.5356	0.4665
All-Attention-BN	0.5182	0.4613
All-Attention -BN-VGG19	0.4919	0.4482

Table 5. Ablation study for different techniques on mean IoU score

VGG16, VGG19, ResNet50, DenseNet121 which allow more adaptation of pretrained model on the training set. Comparing VGG16 and VGG19 encoders with encoder backbone unfrozen, VGG19 produce a better performance in mean IoU of 0.6237 and 0.5933 in the validation and test set respectively. This could be attributed to VGG19 having more pretrained layers compared to VGG16 and can learn better features. ResNet50 encoder with unfrozen backbone, results in better mean IoU of 0.6238 and 0.5883 in the validation and test set respectively compared to VGG16 but lower than VGG19 for the test set. This can be attributed to the skip connections in ResNet that helps to learn features from deeper layers without vanishing gradients. DenseNet121 (Unfrozen backbone) which connects every convolution layer to every other convolution layer in the same stage of network performed worst than VGG16 for the validation set but better than VGG16 for test set which indicate its performance is similar to VGG16 (Unfrozen backbone).

Encoder	Validation	Test
VGG16	0.6008	0.5429
VGG16 unfrozen	0.6164	0.5715
VGG19	0.5639	0.5046
VGG19 unfrozen	0.6237	0.5933
ResNet50	0.4187	0.3746
ResNet50 unfrozen	0.6238	0.5883
DenseNet121	0.5493	0.5230
DenseNet121 unfrozen	0.5970	0.5771

Table 6. Effect of different encoder architectures on mean IoU score

4.9. Comparison of different U-Net architectures

In this section, we compare the results of different U-Net based architecture (vanilla U-Net [13], residual U-Net [18], Attention U-Net [12] and residual attention U-Net [11]) with VGG19 encoder backbone unfrozen using the CamVid dataset in Tab. 7 and Fig. 9. Overall, VGG19-Attention U-Net performed best with mean IoU of 0.6237 on validation set and 0.5933 on the test set. Without attention gates in the decoder, VGG19 U-Net performed worse with mean IoU of 0.5893 and 0.5706 for validation and test set respectively which shows that Attention plays an important role in better performance. VGG19 Residual U-Net achieves a better mean IoU than VGG19 U-Net as shown in Tab. 7 with residual connection added for each layer to connect the input and output of same layer aiding the flow of information. However, VGG19 Residual Attention U-Net performed worst which indicates that using Residual connections concurrently with attention gates in U-Net may not work as well with semantic segmentation of CamVid dataset. Experiments with other dataset may be needed for VGG19 Residual Attention U-Net which we will leave for future work.

Model	Validation	Test
VGG19-U-Net	0.5893	0.5706
VGG19-Res U-Net	0.6032	0.5697
VGG19-Attention U-Net	0.6237	0.5933
VGG19-Res Attention U-Net	0.4817	0.4403

Table 7. Comparison of different U-Net architectures with backbone unfreezed on mean IoU score

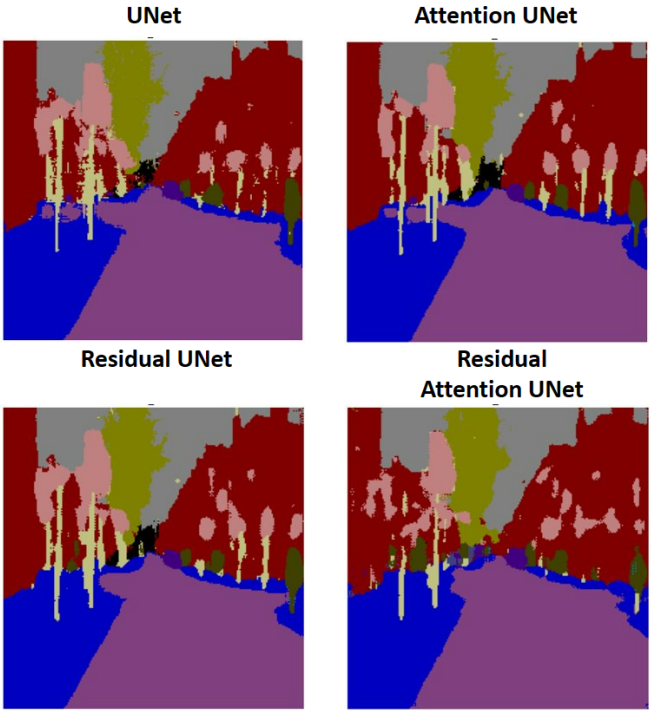

Figure 9. Effect of different U-Net with backbone unfrozen on CamVid segmentation

4.10. Comparison of other model architectures

We compare U-Net based architectures using VGG19 encoder with VGG19-FCN-8 [15] VGG19-Segnet [2] and VGG19-DeepLabV3+ [4] for the semantic segmentation of the CamVid dataset in Tab. 8 and Fig. 10. We use the vanilla U-Net without Attention for a fairer comparison and we do not use batch normalization in this study as the training for DeepLabV3+ is unstable with batch normalization with our batch size of 8. VGG19-U-Net perform best due to the skip connections that connect multiple resolution features from the encoder to decoder with mean IoU of 0.5893 and 0.5706 for validation and test set respectively only losing out to VGG19-DeepLabV3+ for the test set. VGG19-FCN-8 did not perform as well compared to VGG19-U-Net due to its simpler model architecture compared to U-Net with mean IoU of 0.5033 and 0.4725 for validation and test set respectively. VGG19-DeepLabV3+ which utilize atrous convolution in cascade or in parallel to capture multi-scale context with different dilations rates and Atrous Spatial Pyramid Pooling module augmented with image-level features encoding global context achieved better results than VGG-U-Net for test set with mean IoU of 0.5807 but performed worse than VGG-U-Net with the Validation set with mean IoU of 0.5886. This shows that both VGG-U-Net and VGG19-DeepLabV3+ perform equally well in terms of mean IoU. VGG19-SegNet which transfers pooling indices from encoder to decoder for segmentation performs better compared to VGG19-FCN8 but worse than VGG19-U-Net and VGG19-DeepLabV3+ with mean IoU of 0.5419 and 0.5147 for validation and test set respectively.

Model	Validation	Test
VGG19-U-Net	0.5893	0.5706
VGG19-FCN-8	0.5033	0.4725
VGG19-Segnet	0.5419	0.5147
VGG19-DeepLabV3+	0.5886	0.5807

Table 8. Effect of other model architectures with unfrozen backbone on mean IoU score

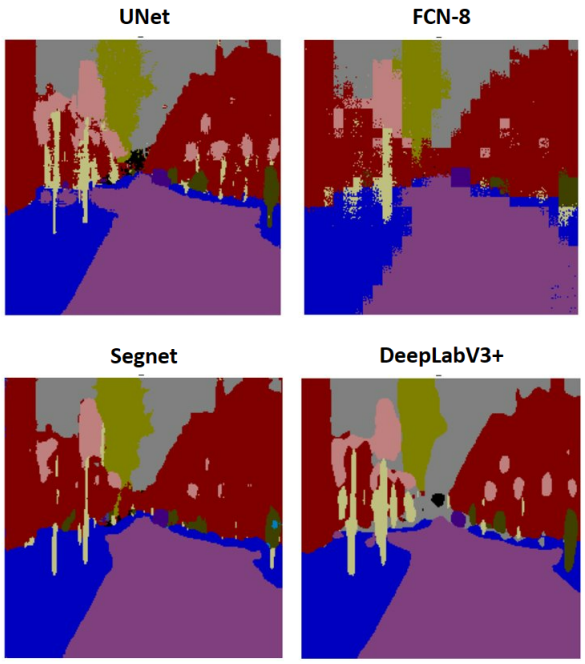

Figure 10. Effect of different models with backbone unfreezed on CamVid segmentation

5. Conclusion

In conclusion, this study extensively explored semantic image segmentation utilizing the CamVid dataset. The utilization of an enhanced U-Net framework, the Attention U-Net, integrated with a pretrained VGG19 encoder network alongside batch normalization was investigated. A comprehensive analysis of crucial hyperparameters like learning rate, dropout, and weight decay was conducted. Addressing the challenge of class imbalance, various loss functions such as cross entropy, Tversky, focal Tversky, focal, dice, and a unified focal loss were examined, revealing that the unified focal loss significantly outperforms others in the context of Attention U-Net Semantic segmentation. Moreover, an ablation study centered on the Attention U-Net model architecture employing a pretrained VGG19 encoder network was performed. Comparisons were drawn among different encoder architectures with both frozen and unfrozen backbones, with the VGG19 unfrozen backbone demonstrating the most favorable outcomes. Furthermore, a comparative analysis between the implemented Attention U-Net and other U-Net variants (Vanilla U-Net, Residual U-Net, and Residual Attention U-Net) was conducted. Additionally, a comparison of the results achieved by U-Net with FCN-8, SegNet, and DeepLabV3+ for the semantic segmentation of the CamVid dataset was executed.

6. Acknowledgement

I wish to extend my heartfelt gratitude to Prof. Lin Guosheng for his exceptional support and guidance during the duration of this course. Prof. Lin's unwavering commitment to orchestrating this project and sharing invaluable knowledge has played a pivotal role in enhancing our learning journey. I deeply appreciate his dedication to fostering our academic development.

References

[1] Nabila Abraham and Naimul Mefraz Khan. A novel focal tversky loss function with improved attention u-net for lesion segmentation, 2018. 1, 3

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation, 2016. 1, 8

[3] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009. Video-based Object and Event Analysis. 4

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018. 1, 8

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 4

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 6

[7] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018. 6

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 1, 4

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 4

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. 1, 3

[11] Zhen-Liang Ni, Gui-Bin Bian, Xiao-Hu Zhou, Zeng-Guang Hou, Xiao-Liang Xie, Chen Wang, Yan-Jie Zhou, Rui-Qi Li, and Zhen Li. Raunet: Residual attention u-net for semantic segmentation of cataract surgical instruments, 2019. 1, 7

[12] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention u-net: Learning where to look for the pancreas, 2018. 1, 2, 7

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 1, 7

[14] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks, 2017. 1, 3

[15] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2016. 1, 8

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 1, 6

[17] Michael Yeung, Evis Sala, Carola-Bibiane Schönlieb, and Leonardo Rundo. Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation, 2021. 1, 3

[18] Zhengxin Zhang, Qingjie Liu, and Yunhong Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, May 2018. 1, 7

[19] Rongjian Zhao, Buyue Qian, Xianli Zhang, Yang Li, Rong Wei, Yang Liu, and Yinggang Pan. Rethinking dice loss for medical image segmentation. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 851–860, 2020. 1, 3