A Novel Gradient Difference Minimizing Loss for Image Semantic Segmentation and Classification using Attention U-Net architecture

Athrva Pandhare [1] **Abhinav Raghunathan** [2]

Abstract

In this project, we shall attempt to implement an Attention U-net (Oktay et al., 2018) for semantic segmentation of animal images. We will also implement a classification branch from the U-Net which will predict the label of the Images. Using the attention framework, we hope to be able to direct the neural network training in a way that it focuses mostly on the important parts of the image (which contain animals), and is able to correctly ignore the non-important parts. We expect the classifier to also perform well considering that it will also benefit from the attention framework implemented.

1. Introduction and Motivation

The U-Net is a popular architecture for image segmentation tasks, and has been widely used in medical image processing and segmentation(Ronneberger et al., 2015). The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization (Ronneberger et al., 2015). The architecture may be thought of as including an encoding region that captures the most important “semantic” features in an image, thereby reducing the dimensionality, and a decoding region, which decodes this low (relatively) dimensional encoding to produce a different representation of the image (in our case a segmentation map).

As an improvement to the original U-Net architecture, the Attention U-Net introduces a novel attention gate (AG) model that automatically learns to focus on target structures of varying shapes and sizes. Models trained with AGs implicitly learn to suppress irrelevant regions in an input image while highlighting salient features useful for a specific task (Oktay et al., 2018). Owing to the success of the Attention framework for image segmentation, it has found extensive use where precise localization is required.

A more specialized architecture called the Spatial Channel Attention U-Net(SCAU-Net) is an example of the attention based variations of the U-Net architecture. SCAU-Net has an encoder-decoder-style symmetrical structure integrated with spatial and channel attention as plug-and-play modules. The main idea is to enhance local related features and restrain irrelevant features at the spatial and channel levels(Zhao et al., 2020). The SCAU-Net has shown an improvement of 1% on the Dice index and of 1.5% on the Jaccard index for the gland dataset GlaS and CRAG(Zhao et al., 2020).

Attention U-Net architectures have also been tried with modified loss functions, for instance, Abraham et al.(Abraham & Khan, 2018) suggest the use of a novel focal tversky loss function for lesion segmentation. The network with this modified loss function showed an improvement of 25.7% on the BUS2017 dataset, which is a relatively sparse dataset with lesions occupying only about 4.84% of the total image area.

Attention U-Net inspired architectures have also found extensive use in non-medical image segmentation tasks, this is apparent from the work done by Chen et al.(Chen et al., 2021), who suggest the use of self-attention U-Nets for Segmentation of Building Rooftops in Optical Remote Sensing Images. Some related work advocating the use of Attention U-Net frameworks are done by Luo et al.(Luo et al., 2019), Li et al.(Li et al., 2020), Zhang et al.(Zhang et al., 2021)

2. Dataset

We used subset of the Oxford IIIT Dataset for traiing and inference. The total number of images used were 2550. The images were randomly (uniformly) smapled from the overall dataset to ensure that this subset contained all the classes (totalling 37).The Oxford IIIT dataset is a 37 category pet dataset with roughly 200 images for each class. The images have large variations in scale, pose and lighting. All images have an associated ground truth annotation of breed, head ROI, and pixel level trimap segmentation.
The reason for training on a subset of the complete dataset was primarily the computational cost involved. Furthermore,

[1] athrva@seas.upenn.edu (athrva) [2] abhirags@seas.upenn.edu (abhirags).

in this project we wanted to make, testing novel concepts a priority. Hence we tried to optimize our training time and also spent time interpreting the model and it's performance. We introduced novel ideas for both the training and interpretation.

3. Proposed Approach

The work done by Oktay et al.(Oktay et al., 2018), illustrates that the Attention based implementation of the U-Net performs better at segmentation tasks compared to the original U-Net implementation. The U-Net architecture uses skip connections that link the layers in the encoder region to the layers in the decoder region. The skip connections combine the spatial information from the downsampling path with the spatial information from the upsampling path to convey the important semantic spatial information which is required to reconstruct the structure of the image in the decoder section. However, this process also brings along the poor feature representation from the initial layers. To counter this Oktay et al.(Oktay et al., 2018) have proposed a network that uses "attention gates" to retain the good feature representation from the deeper layers and the good spatial information from the shallow layers. We shall be using the Attention U-Net for animal image semantic segmentation and classification.

In our study we shall attempt to treat this problem as an image generation task and instead of pixel-wise classification, we shall try to make a generative model that translates the input RGB images into their corresponding segmentation maps. For the classification part of our project, we shall take the logits from the end of the encoder section of our Attention U-Net, and use a global average pooling operation to get an encoding of the labels which when activated using a softmax layer will result in a one-hot probability distribution. The encoder captures the semantic information from the image and hence is the most suitable part of the overall architecture to siphon the logits from.

4. Methods

4.1. Data Pre-processing

The data made available in the Oxford IIIT dataset contains RGB images of animals, and their corresponding segmentation maps. In this project, the task can be thought of as image translation from a form $\mathcal{A}$, the RGB image, to a form $\mathcal{B}$, the segmentation map. The segmentation maps in the Oxford IIIT dataset are normalized to have pixel values between 0 and 1. Next the segmentation maps are converted into single channel images (makes image generation easier). The different classes in the images are depicted as different shades of gray. Figure 1 shows a sample of the input RGB images and their corresponding segmentation maps.

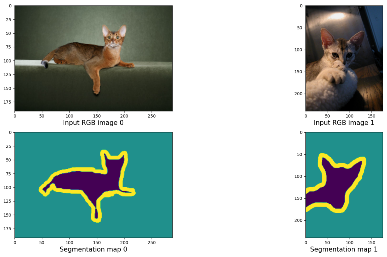

Figure 1. Sample RGB Images and Segmentation Maps

4.2. The Attention U-Net : Conceptual Architecture

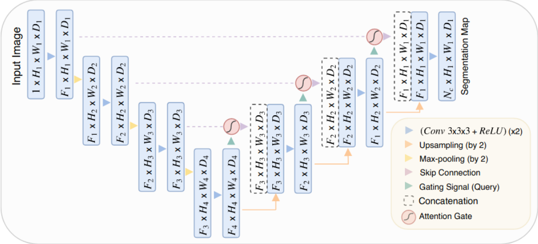
Input Image
$1 \times H_1 \times W_1 \times D_1$
$F_1 \times H_1 \times W_1 \times D_1$
$F_1 \times H_2 \times W_2 \times D_2$
$F_2 \times H_2 \times W_2 \times D_2$
$F_2 \times H_3 \times W_3 \times D_3$
$F_3 \times H_3 \times W_3 \times D_3$
$F_3 \times H_4 \times W_4 \times D_4$
$F_4 \times H_4 \times W_4 \times D_4$
$F_3 \times H_3 \times W_3 \times D_3$
$F_3 \times H_3 \times W_3 \times D_3$
$F_2 \times H_3 \times W_3 \times D_3$
$F_2 \times H_2 \times W_2 \times D_2$
$F_2 \times H_2 \times W_2 \times D_2$
$F_1 \times H_2 \times W_2 \times D_2$
$F_1 \times H_1 \times W_1 \times D_1$
$F_1 \times H_1 \times W_1 \times D_1$
$N_c \times H_1 \times W_1 \times D_1$
Segmentation Map
(Conv 3x3x3 + ReLU) (x2)
Upsampling (by 2)
Max-pooling (by 2)
Skip Connection
Gating Signal (Query)
Concatenation
Attention Gate

Figure 2. Sample RGB Images and Segmentation Maps

Figure 2. shows the conceptual architecture of the Attention U-Net along with its components (notice the use of the attention gate). An RGB image is fed into the network which first encodes the image into a feature rich embedding and then decodes it to produce the final image. Skip connections between the encoder section and the decoder section are added to transfer the spatial representation of the image and additionally attention gates are added that take the input from the skip connections and the previous deep layers to combine the good spatial information from the initial layers and the rich feature representation of the deep layers.

4.3. Components of the Loss function

In this section we define the loss functions used for training the Attention U-Net.

4.3.1. Some conventional Loss functions

Commonly used loss functions for supervised image segmentation include L1 and L2 loss for penalized regression, and Berhu which combines L1 and L2 loss functions (Table 1) (Ming et al., 2021). In these equations, d represents the estimated segmentation map and d* represents the ground truth segmentation map.

$$L_1\left(d, d^*\right) = \qquad L_1\left(d, d^*\right) = \frac{1}{N} \sum_{i=1}^{N} ||d_i - d_i^*\|_1 \quad (1)$$

Furthermore, we define the L_2 loss as follows,

$$L_2\left(d,d^*\right) = L_2\left(d,d^*\right) = \frac{1}{N}\sum_{i=1}^{N}||d_i - d_i^*||_2^2 \tag{2}$$

As a note to the reader, a commonly combination of the L_1 and the L_2 loss functions is the BerHu loss function, defined as follows:

$$L_{Berhu}\left(d,d^*\right) = \left\{ \begin{array}{ll} |d-d^*| & \text{if } |d-d^*| \le c \\ \frac{|d-d^*|^2+c^2}{2c} & \text{if } |d-d^*| > c \end{array} \right. \tag{3}$$

We also define the cross-entropy loss here, which is formulated below.

$$CE(\boldsymbol{y},\hat{\boldsymbol{y}}) = -\sum_{i=1}^{\mathrm{N}_c} y_i \log(\hat{y}_i) \tag{4}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is the ground truth and $y \in \mathbb{R}^n$ is the prediction.

4.3.2. The Label Loss (Classification Loss)

Our classification task is a classic example of multi-class classification, where the network needs to assign on of the 37 classes to any given image. Since we used a composite loss function which included a weighted summation of many loss components, we decided to use the L_1 loss between the predicted probability distribution of labels and the actual ground truth labels. This loss function for the labels is as follows:

$$L_l = \frac{1}{N}\sum_{n=1}^{N}||y_n - \frac{e^{f_i(x_n)}}{\sum_{j=1}^{N_c} e^{f_j(x_n)}}||_1 \tag{5}$$

4.3.3. The Gradient Loss

Consider that $\mathcal{R}$ is the input RGB image of dimensions $(M \times N)$ and d^* is the corresponding ground truth segmentation map of the same dimensions.
In general, we define a convolution operation using a kernel $\mathcal{K}$ (of dimensions $(m \times n)$ as,

$$\mathcal{O}_{i,j} = \sum_{k=1}^{M}\sum_{l=1}^{M} d^*(i+k-1, j+l-1) \times \mathcal{K}(k,l) = d^* \circledast \mathcal{K} \tag{6}$$

where i runs from 1 to $M-m+1$ and j runs from 1 to $N-n+1$. Let g_x and g_y be the horizontal and vertical gradient kernels respectively. Then the horizontal and vertical gradient of the ground truth segmentation map is given by,

$$\begin{aligned} G_x &= d^* \circledast g_x \\ G_y &= d^* \circledast g_y \end{aligned} \tag{7}$$

Similarly, the x and y gradient of the generated segmentation map d is given by,

$$\begin{aligned} F_x &= d \circledast g_x \\ F_y &= d \circledast g_y \end{aligned} \tag{8}$$

Now, we define the bounded gradient image of the ground truth segmentation-map d^*, and the generated segmentation map d as G_{xy} and F_{xy} respectively.

$$\begin{aligned} G_{x,y} &= S\left\{\frac{G_x^2}{\max(G_x^2)+\epsilon} + \frac{G_y^2}{\max(G_y^2)+\epsilon}\right\} \\ F_{x,y} &= S\left\{\frac{F_x^2}{\max(F_x^2)+\epsilon} + \frac{F_y^2}{\max(F_y^2)+\epsilon}\right\} \end{aligned} \tag{9}$$

where S is the sigmoid function and ϵ is a small positive real number. Finally, the gradient loss may be formulated as follows,

$$\nabla_{loss} = \sum_{i=1}^{M-m+1}\sum_{j=1}^{N-n+1} ||G_{x,y}| - |F_{x,y}|| \tag{10}$$

4.3.4. Formulating the Total Loss

Let us define the total loss as $\mathcal{G}(d,d^*,y,y^*)$. We define $a_1 \in \mathbb{R}$ (representing the weights assigned to each of the components in the total loss), L_1 denoting the L_1 loss and L_2 denoting the L_2 loss. Then we can define the total loss as,

$$\begin{aligned} \mathcal{G}(d,d^*,y,y^*) &= \nabla_{loss} \times (L_1(d,d^*) + L_2(d,d^*)) \\ &\quad \times (1 + CE(d,d^*)) + a_1 L_l \end{aligned} \tag{11}$$

The total loss can therefore be viewed as a composite loss function that attempts to minimize multiple aspects of the error between the ground truth segmentation map and the generated segmentation map.

5. Interpretation of the Total Loss function

In this section we define the loss functions used for training the GAN. A composite loss function was used for training. The loss function consisted of the following components.

1. The *Gradient loss*, which is a novel implementation. Adding this component of the loss ensures that the edges are conserved in the generated images. Based on our training process, it sped up the training process significantly. It is in general seen that using loss functions like L_1 or L_2 alone in training generative frameworks can result in blurred looking images, that don't necessarily preserve the sharpness of the edges. Adding the gradient loss penalizes "blurness" and encourages the model to be more confident in it's predictions.

2. L_1 *Loss* between the ground truth segmentation map and the generated segmentation map. This component of the loss minimizes the ”difference” between the ground truth segmentation map and the generated segmentation map.

3. L_2 *Loss* between the generated segmentation map and the ground truth segmentation map. Adding this loss speeds up the initial learning process when there is a large difference between the generated segmentation map and the ground truth segmentation map.

4. *Cross-entropy loss* between ground truth segmentation map and the generated segmentation map. This loss has the additional effect of penalizing the incorrect pixels. It was found that this loss sped up the initial training process.

6. Results and Discussion

In this section we demonstrate the results obtained during the training process. We shall also discuss the various metrics tracked during training and attempt to explain the patterns therein.

We shall begin with the training losses. During the training process, we recorded the L_1 loss, L_2 loss, the Gradient loss and the Label loss. Figure (3) shows the decrease in the different components of the loss function throughout the training process. We see that the training loss decreases rapidly initially and then reachs an inflection point at epoch 43. Thereafter, the decrease in the loss is gradual. Figure (4) shows the gradient loss for the training dataset. We observe a similar trend in the gradient loss. The exception here being that there is another increase in the error decay at around epoch 155, after which the error decay rate increase again.

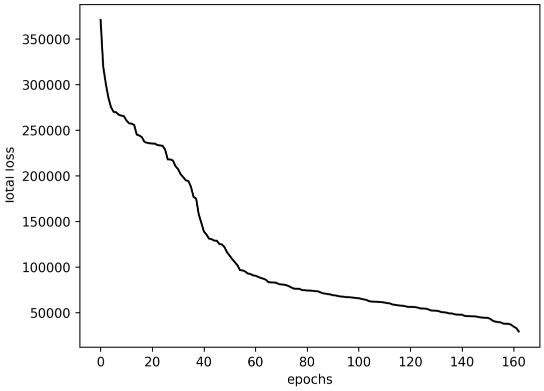

Figure 3. Total loss on the training dataset

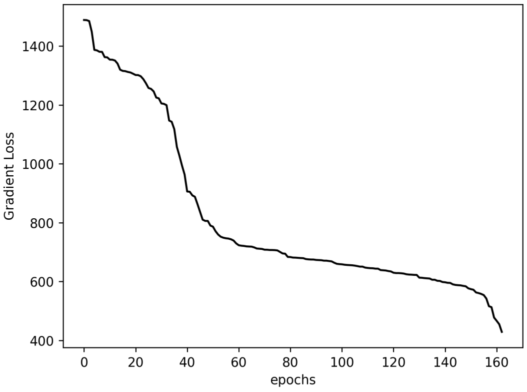

Figure 4. Gradient loss on the training dataset

We now investigate the L_1 and L_2 losses on the training dataset. Figures (5),(6) show the L_1 and L_2 loss during the training process. The L_1 loss seems to follow a near linear downward trend throughout the training process. Generally, it is seen in generative networks that most of the loss decrease happens in the first few epochs, after which, the loss decay is gradual. The linear downward trend observed here may signify that adding the Gradient loss in the loss formulation may have a local-optima avoiding effect. The trend followed by the L_2 is more of a conventional trend, where we seen a steep and sharp decline in the loss in the initial epochs, and thereafter a plateauing of the loss with decrease following a small negative slope. Finally, we now infer trends from the label loss which is an indication of the classification accuracy. Figure (7) shows the label/Classification loss on the training dataset. We see that the initial decrease in the classification loss is very gradual. This can be attributed to the fact the that in the initial epochs, the gradient loss, L_1 and L_2 losses dominate. Eventually, at epoch 150, we see a sharp decline in the classification loss.

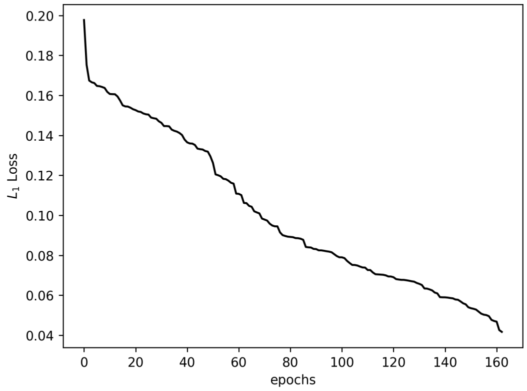

Figure 5. L_1 on the training dataset

We can make similar observations on the validation loss

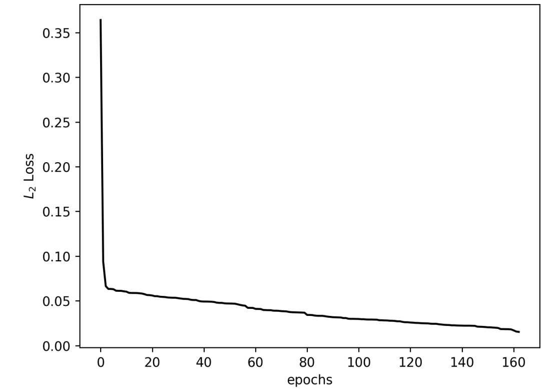

Figure 6. L_2 on the training dataset

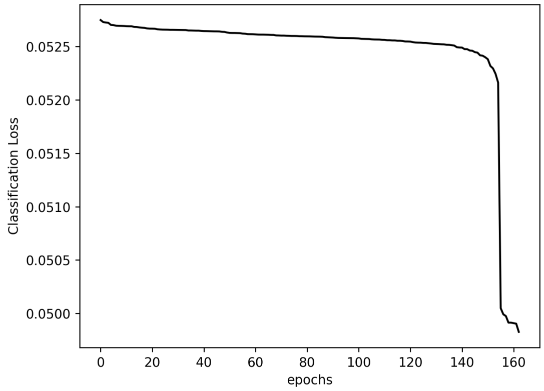

Figure 7. Classification/Label loss on the training dataset

curves. The validation loss was also recorded throughout the training process. The validation loss is *ideally* thought to give an unbiased estimate of the performance of the model. Figure (8) shows the total loss curve on the validation dataset. We see a similar trend in the total validation loss as was seen in the total training loss. The exception here is that there exists a step-like decrease in the total validation loss. We see that the validation loss settles at around epoch 60. Figure (9) show the Gradient loss for the validation dataset. The gradient loss on the validation dataset also show the same trend as was seen in the training dataset. We see a step-like decrease in the loss, with an accelerated decrease in the loss occurring after epoch 150. Figures (10) show the L_1 for the validation dataset. We see that the L_1 loss again follows a near-linear trend. Interestingly, we see that the slope of decrease does not seem to flatten toward the end of the training. This seems to indicate that additional training can result in a further decrease in the loss. Finally, figure (11) shows the classification loss on the validation dataset. The trend here is similar to the trend in the training dataset, the initial decline in the classification loss is gradual, which then becomes steep at around epoch 150. Overall, we see that the designed loss pipeline seems to minimize all the components on the total loss cumulatively as well as individually. Figure (12) shows the progression of the generated image throughout the training process. It can be seen that we were able to obtain good results.

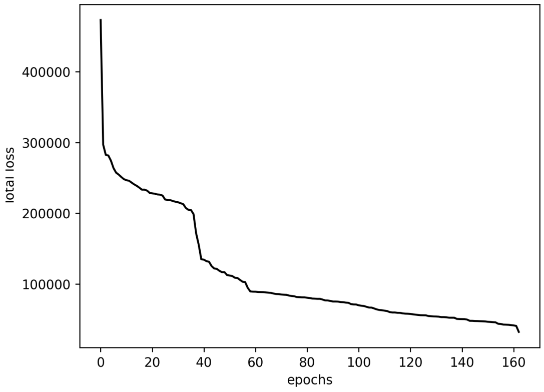

Figure 8. Total loss on the validation dataset

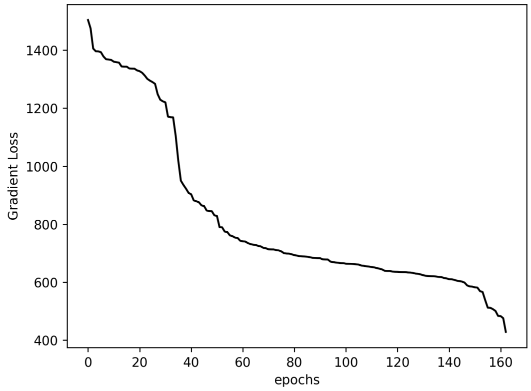

Figure 9. Gradient loss on the validation dataset

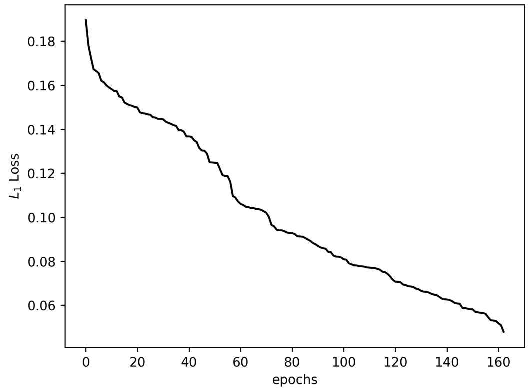

Figure 10. L_1 loss on the validation dataset

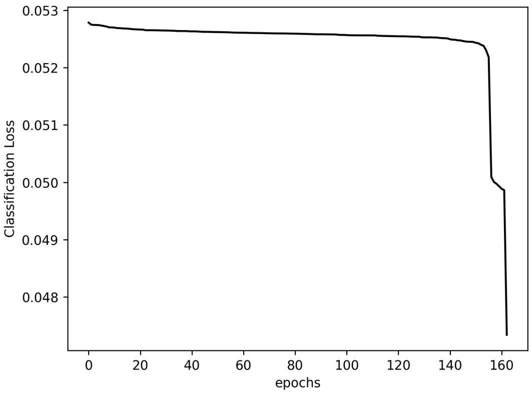

Figure 11. Classification/Label loss on the validation dataset

7. Evaluating the quality of the encoded logits

We attempted to evaluate the quality of the encoded logits from the lowermost layer of the encoder. Doing this gives us an indication of the performance of the model in segregation the various species of animals. Furthermore we are also able see the difference in activation given inputs of different classes.
Originally, the logits were of 37 × 16 × 16 shape. This is a very high dimensional representation of the data. We therefore decided to reduce the dimensionality of the encodings before proceeding forward. The following is our pipeline for interpreting the model.

7.1. Principal Component Analysis on the Encoded Logits

We used Principal Component Analysis (PCA) to reduce the dimensionality of the encoded logits. PCA is a form of dimensionality reduction framework that estimates the directions of maximum variance in the data. PCA can be estimated using eigenvalue decomposition or using singular value decomposition (SVD). In our case we used the SVD variant of PCA. PCA may also be thought of as a linear autoencoder. The input to PCA is the data itself (in our case the encoded logits) and the output as the principal components.
It was found that the first 900 principal components explained 97% of the variance of the encoded logits. We were therefore successful in decreasing the dimensionality of the encoded logits from 9472 to 900 (a 10.5 times reduction). We then used the reduced logits for our further analysis.

7.2. Gaussian Mixture Models with the EM Algorithm

In order to show that the Attention U-Net we developed responded to different inputs differently (and is thus capable of class distinctions), we used the Gaussian Mixture Model (GMM) clustering using the Expectation Maximization (EM) algorithm. We would cluster the reduced logits (from PCA) and indicate the respective inputs that were the source of the encodings. In this case the clusters formed will indicate the activation of the Attention U-Net architecture to different inputs.

7.2.1. Selecting the number of clusters for GMM

In order to carry out a meaningful GMM clustering and by extension a meaningful model activation analysis, we need to select the right number of GMM clusters. An incorrect choice of the clusters can lead to a bias in our interpretation of the model's activations to different inputs.
For this task we used two important metrics **Akaike information criterion (AIC)** and the **Bayesian information criterion (BIC)**. For a detailed analysis of these criteria, the reader is referred to the work done by (Stoica & Selen, 2004) and (Claeskens & Hjort, 2008). In our analysis, we varied the number of clusters in the GMM model and estimated the AIC and BIC. Figures (13),(14) show the plots for AIC and BIC. We further defined a new criterion for the selection of optimal number of clusters, which we define as the **Combined Information Criterion (CIC)**.

$$CIC = \frac{2 \times AIC \times BIC}{AIC + BIC} \tag{12}$$

Figures (13),(14),(15) show the plots for AIC, BIC and CIC, for various tested GMM models with different number of clusters. Note that a lower value of AIC and BIC is more preferable. We see that the AIC reaches it's lowermost point at `number of clusters = 5`, and thereafter rises. Therefore according to the AIC plot, we see that the optimal number of clusters is 5. Consider figure (15) which shows the plot for the BIC criterion. We see that the BIC criterion is minimum at `number of clusters =`

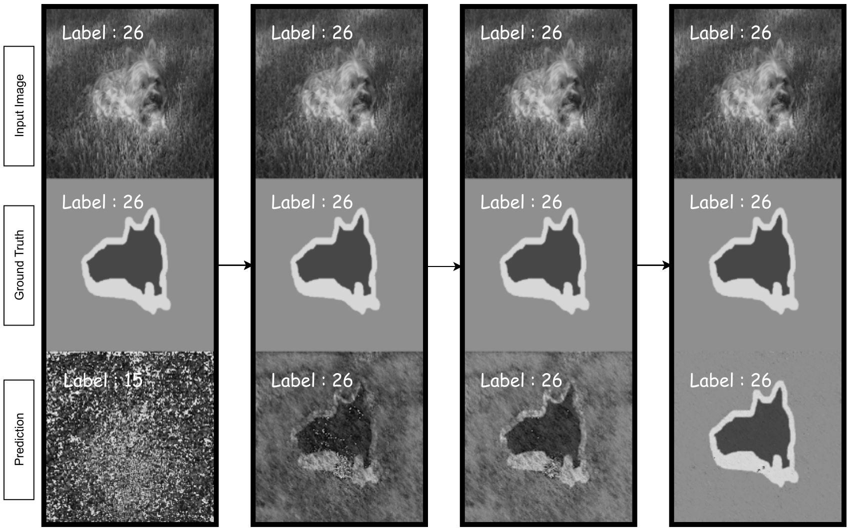

Figure 12. Progression of Generated Image during the Training Process

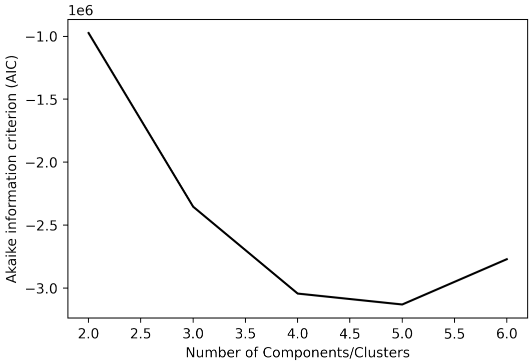

Figure 13. Akaike information criterion (AIC) vs Number of clusters

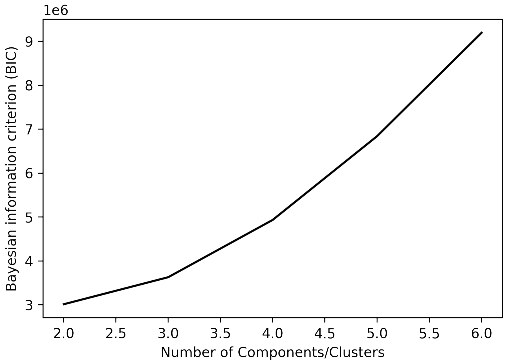

Figure 14. Bayesian information criterion (BIC) vs Number of clusters

`2`. In order to obtain a number of clusters that minimizes both the AIC and BIC, we define the CIC criterion, which by definition gives us the cluster number which minimizes the AIC and BIC criterion. From figure (15) we see that the optimal number of clusters (`Optimal number of clusters = 4`) using the CIC criterion.

7.2.2. Results of GMM Clustering

In this section, we demonstrate the results of the GMM clustering (with number of clusters = 4) on the PCA reduced logits. Figure (16) shows the results of GMM clustering. We observe that the model is able to produce different activations for different classes. We can also see that the clusters are not very well defined, this essentially hints at the difficulty of learning encodings which can,

1. Generate Segmentation maps when decoded.
2. Reflect the class/label difference of the inputs.

Consider figure (17), which shows one sample of inputs from each of the clusters of the reduced logits. We can see that the animals corresponding to each of the clusters are visually very different. This is to be expected, since the activation of the model for each of these animals is semantically different.
From this analysis, we have proven the following

1. The model is capable of distinguishing between very similar looking animals

2. The inner-most encoded logits of the model are capturing the classes successfully, and are also capturing enough spatial and feature representation to be able to decode and generate a segmentation map.

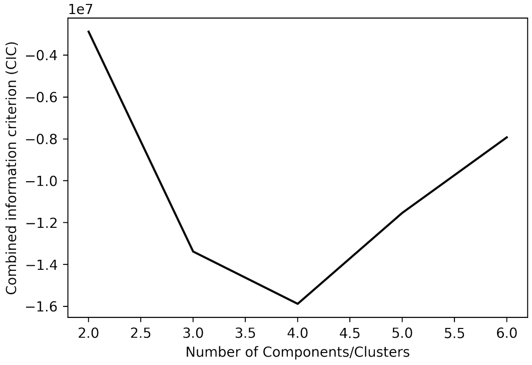

Figure 15. Combined information criterion (CIC) vs Number of clusters

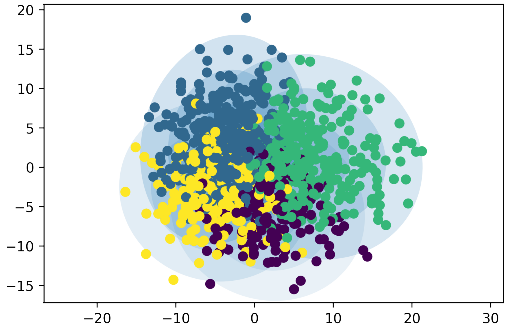

Figure 16. Results of GMM clustering of the Reduced Logits (x-axis : Principal component 1, y-axis : principal component 2)

8. Conclusion and Remarks

In this project we successfully implemented the Attention U-Net model for image semantic segmentation and classification. In the process we introduced novel ideas like the **Gradient loss** and the **Combined Information Criterion**. The former is a gradient error minimizing loss function that encourages a model to preserve the edges of the generated images. It was found that adding this loss function, potentially, has the benefit of pulling the model out of local optima. This loss function was used in conjunction with more conventional loss functions like L_1 loss and L_2 loss. We further did a model interpretation study to verify that our model was capable to produce appropriate activations for semantically different inputs. For this we first extracted the encoded logits from the model and conducted a Principal Component Analysis (PCA) on the model for dimensionality reduction. We found that we were able to reduce dimensionality by more than 10.5 times. These dimensionally reduced logits, which we call **reduced logits**, were then fed to a Gaussian Mixture Model (GMM) for clustering. In order to get the optimal number of clusters, we introduced another novel idea, the **Combined Information Criterion (CIC)** which is a combination of the **Akaike information criterion (AIC)** and the **Bayesian information criterion (BIC)**. We were able to show that the higher dimensional representation of encodings for semantically different inputs lived sufficiently far in higher dimensional space.

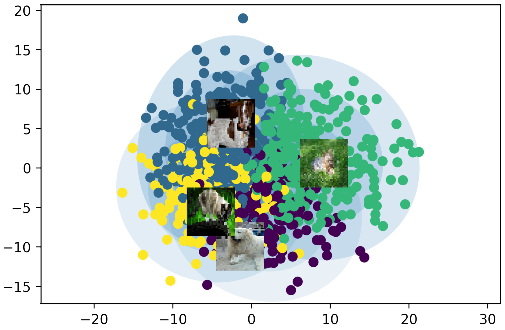

Figure 17. Results of GMM clustering of the Reduced Logits (x-axis : Principal component 1, y-axis : principal component 2)

9. Scope for Improvements

The following improvements can be made on the model and the training pipeline :

1. The model can further be trained on the complete dataset and it's performance can be calculated.
2. Different configurations of the Gradient Loss may be tested and their performances can be compared.
3. A different method of model interpretation can be implemented for example, Local Interpretable Model-Agnostic Explanations (LIME) and Shapley values (SHAP).

10. Timeline

The timeline that we decided for our project is shown in figure (18). We were able to complete all planned aspects of the project as well as introduce some novel ideas.

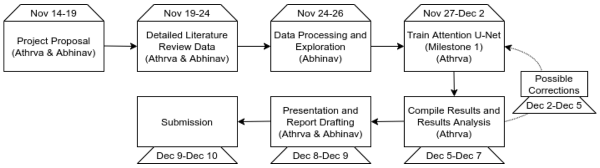
Nov 14-19
Project Proposal (Athrva & Abhinav)
Nov 19-24
Detailed Literature Review Data (Athrva & Abhinav)
Nov 24-26
Data Processing and Exploration (Abhinav)
Nov 27-Dec 2
Train Attention U-Net (Milestone 1) (Athrva)
Possible Corrections
Dec 2-Dec 5
Compile Results and Results Analysis (Athrva)
Dec 5-Dec 7
Presentation and Report Drafting (Athrva & Abhinav)
Dec 8-Dec 9
Submission
Dec 9-Dec 10

Figure 18. Project Timeline

Extra : Network Architecture

See the Attention U-Net architecture shown in figure (19), the green nodes depict the outputs of the model. The first green node is at the end of the encoder and represents the logits used for classification the second green node is at the end of the decoder and represents the predicted single channel segmentation map.

References

Abraham, N. and Khan, N. M. A novel focal tversky loss function with improved attention u-net for lesion segmentation. *CoRR*, abs/1810.07842, 2018. URL http://arxiv.org/abs/1810.07842.

Chen, Z., Li, D., Fan, W., Guan, H., Wang, C., and Li, J. Self-attention in reconstruction bias u-net for semantic segmentation of building rooftops in optical remote sensing images. *Remote Sensing*, 13(13), 2021. ISSN 2072-4292. doi: 10.3390/rs13132524. URL https://www.mdpi.com/2072-4292/13/13/2524.

Claeskens, G. and Hjort, N. L. *Model Selection and Model Averaging*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2008. doi: 10.1017/CBO9780511790485.

Goceri, E. Challenges and recent solutions for image segmentation in the era of deep learning. In *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 1–6, 2019. doi: 10.1109/IPTA.2019.8936087.

Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., and Tu, Z. Deeply-supervised nets, 2014.

Li, C., Tan, Y., Chen, W., Luo, X., He, Y., Gao, Y., and Li, F. Anu-net: Attention-based nested u-net to exploit full resolution features for medical image segmentation. *Computers Graphics*, 90:11–20, 2020. ISSN 0097-8493. doi: https://doi.org/10.1016/j.cag.2020.05.003. URL https://www.sciencedirect.com/science/article/pii/S0097849320300546.

Luo, Z., Zhang, Y., Zhou, L., Zhang, B., Luo, J., and Wu, H. Micro-vessel image segmentation based on the adunet model. *IEEE Access*, 7:143402–143411, 2019. doi: 10.1109/ACCESS.2019.2945556.

Ming, Y., Meng, X., Fan, C., and Yu, H. Deep learning for monocular depth estimation: A review. *Neurocomputing*, 438:14–33, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.12.089. URL https://www.sciencedirect.com/science/article/pii/S0925231220320014.

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M. C. H., Heinrich, M. P., Misawa, K., Mori, K., McDonagh, S. G., Hammerla, N. Y., Kainz, B., Glocker, B., and Rueckert, D. Attention u-net: Learning where to look for the pancreas. *CoRR*, abs/1804.03999, 2018. URL http://arxiv.org/abs/1804.03999.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

Stoica, P. and Selen, Y. Model-order selection: a review of information criterion rules. *IEEE Signal Processing Magazine*, 21(4):36–47, 2004. doi: 10.1109/MSP.2004.1311138.

Zhang, B., Mu, H., Gao, M., Ni, H., Chen, J., Yang, H., and Qi, D. A novel multi-scale attention pfe-unet for forest image segmentation. *Forests*, 12(7), 2021. ISSN 1999-4907. doi: 10.3390/f12070937. URL https://www.mdpi.com/1999-4907/12/7/937.

Zhao, P., Zhang, J., Fang, W., and Deng, S. Scau-net: Spatial-channel attention u-net for gland segmentation. *Frontiers in Bioengineering and Biotechnology*, 8, 2020. doi: 10.3389/fbioe.2020.00670.

11. Acknowledgement

The authors especially thank the University of Pennsylvania CIS 581 Computer Vision and Computational Photography instructor, Prof. Jianbo Shi and the TAs, for their advice and support throughout this project and the semester.

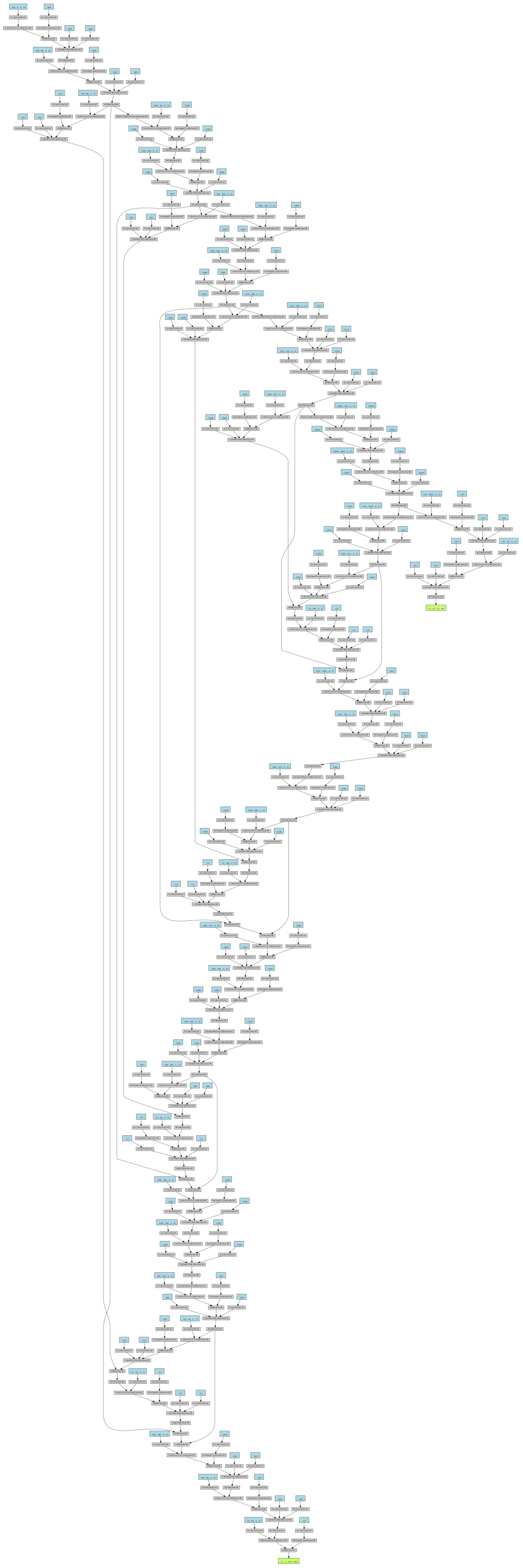

Figure 19. Extra : Full Attention U-Net Architecture