Deep-dive into U-Net and ResNet encoder/decoder pairs

and their application for Image Segmentation

Julien Y-Son Kuhn de Chizelle

jkuhndechizelle@gmail.com

Abstract

Mapping the two-dimensional extent of salt bodies in seismic data is a time-consuming task requiring many weeks to complete. Alleviating the workload is paramount to saving precious time and capital. The main objective of any image segmentation task is to locate and identify features of interest correctly. This paper focuses on applying the U-Net neural network architecture with varying ResNet encoder workflows to the Oxford IIIT Pet images - a dataset better adapted to notice prediction flaws. The lessons learned through this research provide a better understanding of mask segmentation algorithms and their detection potential for the Kaggle TGS Salt Seismic Challenge dataset. The accuracy, interaction over union (IoU), and layer visualization results amplify the long-standing adage that simpler is often better. The eight-layer depth U-Net encoder, trained through 10 epochs, far surpasses the prediction results from the ResNet encoders' 50- and 101-layer depths. Nonetheless, if one transfer learns the weight parameters from the larger Imagenet pet dataset to the Oxford IIIT Pet dataset, the results closely equal a standard 8-layer depth U-Net architecture.

Keywords: UNet, ResNet50, ResNet101, Computer Vision, Seismic Geophysics, Salt Diapirs, Oxford IIIT Pet Dataset, TGS Kaggle Salt Seismic Challenge, Interaction over Union, Image Segmentation.

Section 1 – Introduction

Seismic geophysics provide interpreters with the unique ability to visualize complex structural alterations happening beneath our feet. By studying the changes in wave amplitude reflections resulting from rock density contrasts, geophysicists can map lateral extents of layer strata, notice boundary discontinuities, and recognize features of interest (notably oil and gas, salt diapirs, and fault scarps) (Aminzadeh and Dasgupta, 2013). The introduction of big data and artificial intelligence has renewed interest in more efficient workflows. Industries are now investing considerable capital into automatic image segmentation algorithms based on the ever-growing need for faster project turnarounds. This paper focuses on the U-Net neural network architecture for image segmentation with varying ResNet and U-Net encoder frameworks. Preliminary algorithm searches apply the U-Net/ResNet pairs to the Oxford IIIT Pet Dataset. Later research will use the learned lessons to the Kaggle TGS Salt Seismic Challenge dataset. While the TGS Kaggle Seismic Challenge provides the dataset of interest, visualizing the flaws in the prediction requires a trained eye. The Oxford IIIT Pet dataset, with household pet images, is ideal for visualizing progress from layer outputs.

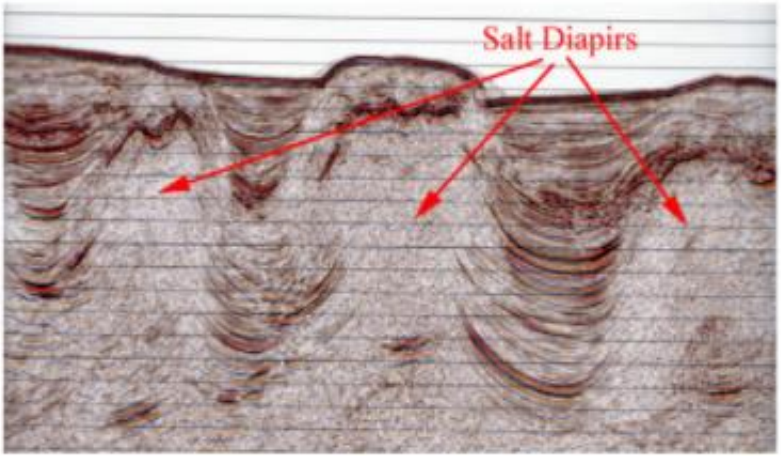


Figure 1: Example of salt diapir in seismic data (Mervine, 2011)

Section 2 – Literature Review

2,1 – Previous salt diapir feature detection algorithms

Identifying salt diapirs in seismic data has long intrigued machine learning researchers. Wang, Ma, and Yang (2018), for example, used a U-Net encoder-decoder architecture on a synthetic seismic dataset showing salt diapir signatures. While the results looked promising, the dataset was synthetic and idealistic. As presented in the Kaggle challenge, the salt diapir dataset provides a more realistic scenario by using real acquired data. Model predictions will be analyzed through accuracy percentages and Intersection over Union (IoU) to determine the overlap between the predicted and ground truth masks (Zeng, Jiang & Chen, 2018).

$$Intersection\ over\ Union = J(A,B) = \frac{A \cap B}{A \cup B}$$

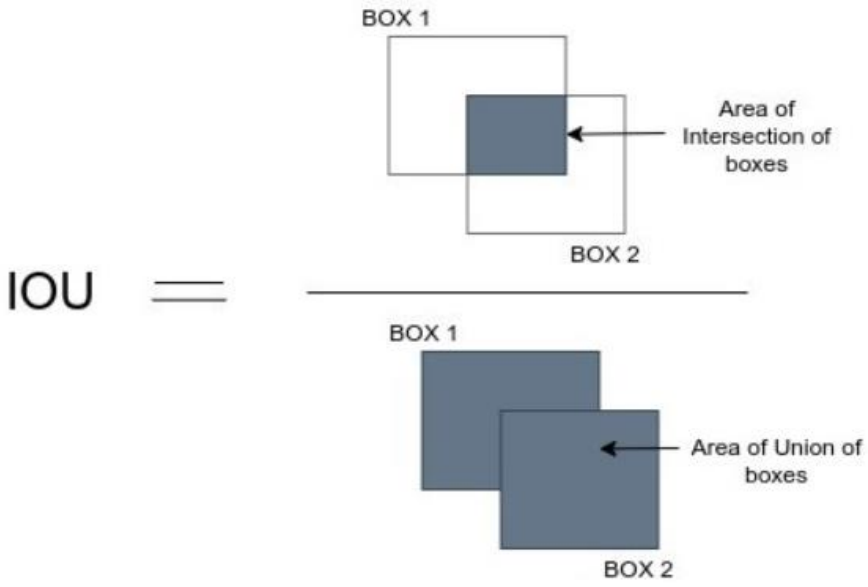


Figure 2.1 – Visual representation of Intersection over Union (Subramanyam, 2021)

As algorithm searches keep testing new approaches to salt body segmentation, authors often resort to a U-Net and ResNet pair architecture. A well-known example is based on the winning Kaggle salt segmentation algorithm by Babakhin et al. (2019). It is worth mentioning that, while the cited paper provides the most accurate salt identification model to date for this challenge, their architecture adds layers of complexity and computing time that go beyond the scope of this report. Nevertheless, giving a brief overview is essential. Therefore, the remainder of this paragraph provides a brief explanation of the Babakhin et al. (2019)'s algorithm. The authors first begin by building a preliminary self-supervised model to predict unlabeled datasets. The predicted labels are then used to improve the model further. Following self-supervised learning, the data is passed into a network inspired by the U-Net architecture using an ensemble of two models for the encoder: ResNet34 and ResNeXT50. The multiple cycle iterations across the two different models increase prediction accuracies. Additionally, the authors use an ensemble of attention mechanisms in the network and spatial Squeeze and Excitation modules. The model finally achieves a mean-average-precision (mAP) score of 0.90 on the private dataset. Other researchers (Islam, 2020; Milosavljevic, 2020) have also used a combination of ResNets and U-Nets with variations to the architecture.

2.2 – Understanding U-Nets:

The previous section laid out a brief overview of algorithms used for image segmentation of salt bodies. Let us now explain what some of the algorithms mean. The U-Net architecture's main objective is to locate features based on contextual information extracted from surrounding pixels (Ronneberger, Fisher & Brox, 2015). While not the first context algorithm proposed, the strength of the U-Net lies in its efficiency and straightforward architecture: a distinct U-shape consisting of a dimension-reducing encoder and an amplified representation decoder. The U-Net

model first uses a combination of max-pooling and convolutional layers with ReLU activations to target individual pixels. The model then up-samples with added convolutional layers that increase the dimensions, thereby adding context to the respective pixels using more feature channels. Furthermore, as nearby features may look similar (adding uncertainty to standard segmentation algorithms), the U-Net adds a weight loss to its layers to increase the distinction between label backgrounds.

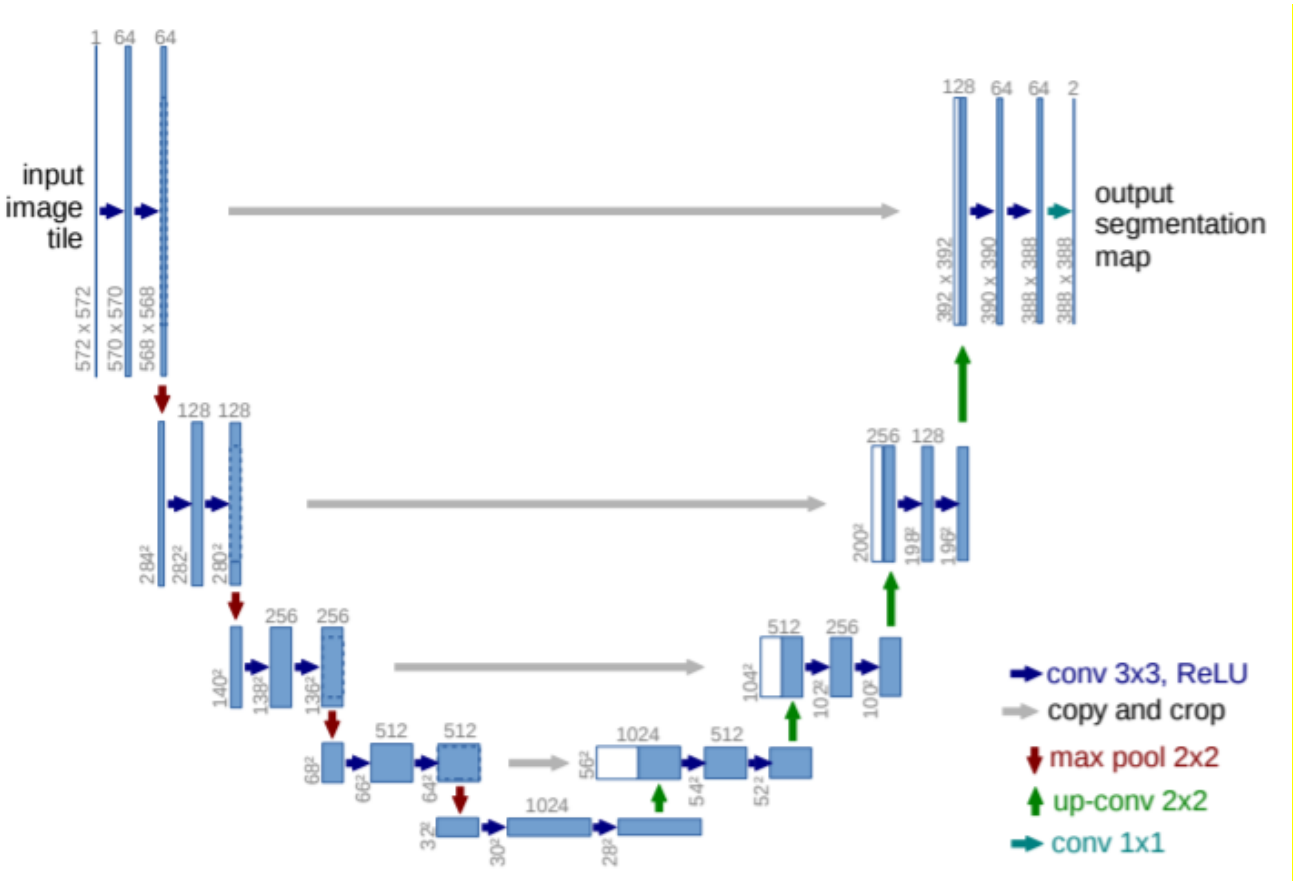


Figure 2.2 – U-Net architectures from Ronneberger, Fisher & Brox (2015)

2.3 – Understanding ResNets

Although U-Nets by themselves offer impressive results, ResNet architectures give the model the ability to add even more depth (better feature extraction) while at the same time reducing the number of layers to arrive at the desired result. By comparing a ResNet layer to its equivalent VGG16 architecture, He et al. (2016) show that ResNets have significantly fewer layers to achieve a similar result. A common misconception suggests that increasing depth in neural networks leads to better results. Unfortunately, that is not always the case.

He et al. (2015) shows that solely stacking layers lead to higher test and train error percentages due to vanishing/exploding gradients and an accuracy stagnation followed by degradation as more layers are added. He et al. (2015) addresses this issue by adding a deep residual learning framework with a new layer block defined by: F(x) + x. The "x" term, referring to the difference between the actual and predicted result, is added as an additional learning method.

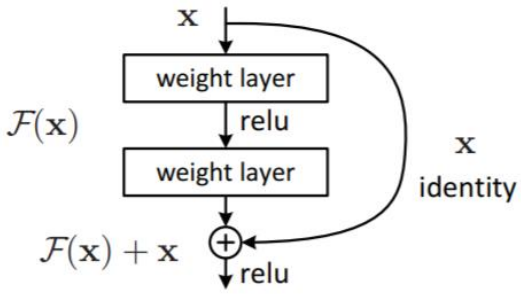


Figure 2.3 – Residual Learning Framework

The 'x' term, also known as 'skip connection layers' or 'identity functions', accounts for what may degrade the accuracy. Furthermore, adding a residual 'x' identity function ensures that "the signal could be *directly* propagated from one unit to any other units, in both forward and backward passes", increasing algorithm optimization (He et al., 2016). Mathematical proofs show the importance of the identity function by adding a lambda term to the equation previously proposed by He et al. (2015) to break apart the identity function, and W representing weights,

$$x_{l+1} = \lambda_l x_l + \mathcal{F}(x_l, W_l)$$

$$x_L = \left(\prod_{i=l}^{L-1} \lambda_i\right) x_l + \sum_{i=l}^{L-1} \hat{\mathcal{F}}(x_i, W_i)$$

Taking into account the backpropagation derivatives where ε is the loss function:

$$\frac{\delta \varepsilon}{\delta x_l} = \frac{\delta \varepsilon}{\delta x_L} (\left(\prod_{i=l}^{L-1} \lambda_i\right) x_l + \frac{\delta}{\delta x_l} \sum_{i=l}^{L-1} \hat{\mathcal{F}}(x_i, W_i))$$

In the above equation, if λ_i is larger than 1, then the factor can be exponentially large. Similarly, if λ_i is smaller than 1, then the factor can be exponentially small and tend to 0, prohibiting the use of the shortcut function and leading to optimization problems (He et al., 2016). As such, identity functions are key to reducing unwanted exponential factors and therefore result in lower training loss and error. From result observations, the ResNet architecture lowers error rates with increasing depth.

Since its introduction to the world of image localization in 2015, researchers have actively attempted to improve the architecture of the ResNet model. A first improvement added a ResNet layer to learn information from other parallel ResNet frameworks (Targ et al., 2016). Other authors added multi-level shortcuts by adding smaller shortcut skips within the larger ones (Zhang et al., 2016). As such, these multi-layered architectures can learn both large and small-scale residuals. Finally, other authors have reduced the depth and increased the width of the residual networks by adding more convolutional layers per blocks (Zagoruyko and Komodakis, 2016). This addition adds more feature planes and increases filter sizes in convolutional layers.

Assigning the number of residual blocks is strategic. Li et al. (2017) shows that models get more constrained as the number of residual layers increases and instead advocates for using a

bottleneck approach. From the ResNeck equation, adding more shortcut-residual terms adds more dependency upon previous layers:

$$y = \prod_{r=1}^{R}(\prod_{l=1}^{L_r-1} W^{r,l} + S^r)\, x = Wx$$

with r-th residual unit, L_r layers, S^r shortcut connections, W weights, and:

$if\ b \geq a, \prod_{i=a}^{b} W^i$ = $W^b W^{(b-1)} \ldots W^{(a+1)} W^a$, otherwise b denotes an identity matrix.

Furthermore, by taking into account the Hessian matrix of the saddle point curve, Li et al. (2017) explains that bottleneck skipping of n = 2 leads to a strict saddle point of:

$$H = \begin{bmatrix} \mathbf{0} & A^T & & & \\ A & \mathbf{0} & & & \\ & & \mathbf{0} & A^T & \\ & & A & \mathbf{0} & \\ & & & & \ddots \end{bmatrix}$$

and as such a depth-invariant structure. The depth-invariance can also be seen when the loss surface is taken into account. Taking another value of n, on the other hand, does not lead to depth-invariance. A skipping of n ≥ 3 leads to a Hessian matrix of 0, and that of n = 1 leads to a Hessian matrix represented below, showing a more complicated depth structure:

$$H = \begin{bmatrix} B & A^T & A^T & \cdots & A^T \\ A & B & A^T & \cdots & A^T \\ A & A & B & & \vdots \\ \vdots & \vdots & & \ddots & A^T \\ A & A & \cdots & A & B \end{bmatrix}$$

A bottleneck skipping of n = 2 is, therefore, the ideal skip for a ResNet model.

Section 3 – Data

This research focuses on the Oxford IIIT Pet dataset (Parkhi et al., 2012) and the TGS Kaggle Salt Identification Challenge dataset (TGS, 2018).

3.1 – The Oxford-IIIT Pet dataset

The Oxford-IIIT Pet dataset consists of 7349 colored images divided into 3680 training images and 3669 testing images. The data consists of animal images and associated animal masks, transformed into NumPy arrays of size (180, 180, 3). As model overfitting is vital, the testing dataset is further split into a testing dataset (90% of initial) and an unseen dataset (10% of initial). As the testing dataset plays a role in training the model through iterative epochs, the unseen dataset will provide a less biased accuracy. The updated Oxford Pet dataset consists of 3680 training images, 3302 testing images, and 367 unseen images.

Figure 3.1: Animal image and its associated animal mask

3.2 – The TGS Kaggle Salt Challenge dataset

The Kaggle Salt Challenge dataset consists of 101 x 101-pixel representations of seismic segy data and is split into train and test folders. The train folder consists of 4000 grayscale seismic images and their associated black-and-white salt localization masks. The test folder consists of 18000 grayscale images without their related segmentation masks. As the author of this research does not have access to the Kaggle labels for the test dataset masks, the focus will solely be on the

training dataset. Similar to the Oxford pet dataset, the dataset is split into training, testing, and unseen folders with similar percentages divided.

Figure 3.2: Seismic image and its associated salt diapir mask

Section 4 – Methods

Understanding the architecture behind U-Nets and ResNets is key to advancing feature segmentation algorithms. This paper focuses on the visually more straightforward Oxford IIT Pet dataset as a preliminary workflow to understand salt diapir detection in seismic images (Kapoor, 2021). Lessons learned from performances on this dataset will inspire models on the more visually complex seismic dataset.

Ten different models are run on the pet dataset, listed in table 1, each run with ten epochs and three-epoch patience for accuracy metrics for callback stopping. These models test the accuracies based on three neural network architectures with a U-Net decoder: a standard U-Net encoder, a ResNet-50 encoder, and a ResNet-101 encoder. The purpose of these architectures is to increase the depth of the neural network progressively.

Each of these architectures will furthermore test various input images and weight parameters. The first variability, data augmentation, uses horizontal flips to add more dataset images to the training set. The second variability, Imagenet weights for lower layer depths (base

weights), increases the accuracy of feature extraction based on a pre-trained model with a more extensive training dataset (Dwivedi, 2019; Chollet, 2018).

Table 1: models run	
Model number	Model parameters
1	U-Net encoder / U-Net decoder
2	U-Net encoder with data augmentation / U-Net decoder
3	ResNet-50 encoder / U-Net decoder
4	ResNet-50 encoder with data augmentation / U-Net decoder
5	ResNet-50 encoder with Imagenet pre-trained weights / U-Net decoder
6	ResNet-50 encoder with Imagenet pre-trained weights and data augmentation / U-Net decoder
7	ResNet-101 encoder / U-Net decoder
8	ResNet-101 encoder with data augmentation / U-Net decoder
9	RestNet-101 encoder with Imagenet pre-trained weights / U-Net decoder
10	ResNet-101 encoder with Imagenet pre-trained weights and data augmentation / U-Net decoder

The validity of result predictions is based on accuracy and Intersection over Union (IoU) metrics for the train dataset, test dataset, and unseen dataset. Visual representations of each model, as well as the accuracy and losses metrics through each epoch iteration, are displayed in the appendix section, all of them showing the same three randomly selected images from the unseen dataset.

Section 5 – Results

Table 2: Model results					
Model	Train Accuracy	Test Accuracy	Train IoU	Test IoU	Unseen IoU
Oxford Pet Dataset					
Basic U-Net	0.87	0.86	0.89	0.87	0.87
Basic U-Net with data augmentation	0.86	0.87	0.89	0.88	0.89
U-Net/ResNet-50	0.88	0.77	0.81	0.81	0.80
U-Net/ResNet-50 with data augmentation	0.84	0.66	0.75	0.75	0.74
U-Net/ResNet-50 with imagenet pre-trained weights	0.92	0.87	0.90	0.89	0.89
U-Net/ResNet-50 with data augmentation and imagenet pre-trained weights	0.92	0.87	0.91	0.89	0.89
U-Net/ResNet-101	0.60	0.58	0.70	0.70	0.70
U-Net/ResNet-101 with data augmentation	0.59	0.58	0.70	0.70	0.70
U-Net/ResNet-101 with imagenet pre-trained weights	0.60	0.59	0.71	0.71	0.70
U-Net/ResNet-101 with data augmentation and imagenet pre-trained weights	0.60	0.59	0.71	0.70	0.70

Section 6 – Analysis and Interpretation

Interesting patterns are extracted from the results table and the visualizations shown in the appendix section. First, studying the performance of the various neural network architectures (U-Net, ResNet-50, and ResNet-101 encoders), result values show that performance decreases with increasing depth. While the dataset accuracies and Interaction over Union (IoU) metrics show the same trends, the unbiased IoU shows the results clearly – dropping from 0.87 for U-Net, 0.80 for ResNet-50, and 0.70 for ResNet-101. As seen in the appendix section for each model, predictive versus actual mask images further corroborate these metrics. Similarly, train and validation set accuracy curves per epochs in figure 6.1 show significantly less overfitting with a shallower depth

U-Net encoder than the deeper ResNet encoder models. Furthermore, table 2 shows that increasing the size of the training dataset through data augmentation algorithms and using previously defined imagenet weights increase the prediction accuracies of the given models. It is additionally worth noting that the imagenet weights increase the prediction accuracy of our models.

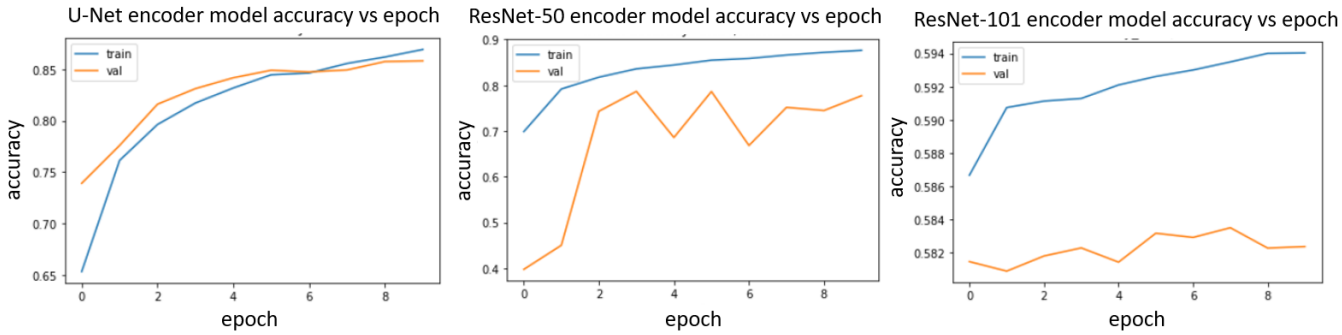


Figure 6.1: accuracy epoch curves

Analyzing the first hidden layer in each model gives a potential clue as to the performance differences. Interestingly, the U-Net encoder model uses a first convolutional layer with 64 filters, significantly increasing the depth with which the model extracts features. On the other hand, the ResNet layers only extract three filters for its first convolutional layer, reducing the extraction potential of its first and possibly critically significant first layer. Figure 6.2 shows an unseen dataset image's visual feature extraction with the respective model summary indicating the filter size below.

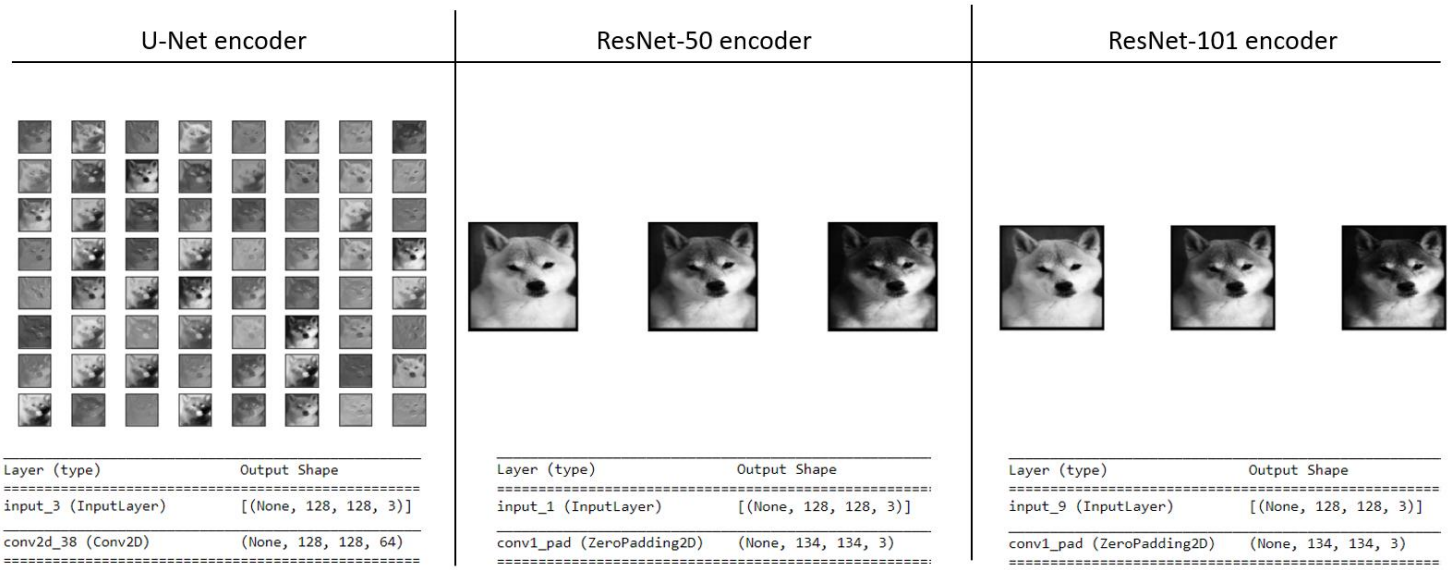


Figure 6.2: First hidden layer comparison between models with U-Net encoder, ResNet-50 encoder, and ResNet-101 encoder.

Section 7 – Conclusions

7.1 – Summary

Image segmentation requires a high-performing algorithm capable of extracting relevant feature characteristics and outputting a prediction mask with an Interaction over Union (IoU) ratio mask as close as possible to one. Applying image segmentation to a Kaggle salt diapir dataset provided by TGS requires a trained eye for prediction validation. As such, understanding the potential of various algorithms on an easier dataset is necessary. The Oxford IIIT- Pet dataset provides such a dataset with the use of household pets as images. This paper analyzed ten models altering neural network depths (U-Net, ResNet-50, and ResNet-101 encoder), input images (through the use of data augmentation), and parameter weights (through imagenet weights trained on a more extensive dataset). Results demonstrate that the depth of the neural network does not necessarily constitute a better predictive algorithm. Furthermore, the first hidden network's filter size may also play a role in determining predictive capabilities. Finally, data augmentation and imagenet weights both play a role in increasing prediction accuracies.

7.2 – Direction for future work

While this paper provides a brief introduction to the application of U-Nets and ResNets as applied to the Oxford IIIT Pet dataset with applicability for seismic datasets, future studies are needed to clarify some of the key concepts.

First, analysis shows that shallower neural network depths with Imagenet weights lead to more accurate predictions. A model that reflects these conditions needs to run on the Kaggle TGS seismic dataset, with results analyzed for automation potential. Furthermore, future research must test shallower ResNet models, notably ResNet-18 and ResNeXt-50, to fully understand the impact of the ResNet architecture on image segmentation.

Second, the winning Kaggle algorithm by Babhakin et al. (2019) was not thoroughly dissected. Delving deeper into layer visualizations to better comprehend what each of the two models predicts is of interest to developing future algorithms.

Third, the U-Net architecture of an encoder-decoder pair, using a ResNet architecture for the encoder, has been used in many algorithms tackling the Kaggle salt segmentation challenge. However, this challenge is based on 2-dimensional images from pixel patches without the complete seismic segy data. The advantage of the entire 3-dimensional seismic dataset provides more spatial relationships between images of salt bodies in 3-dimensions. As Shi, Wu, and Fomel (2019) show, it appears that a simple U-Net architecture without a ResNet specified encoder can lead to promising results. Similarly, other authors have used simple U-Nets on fault detection challenges in 3-dimensional seismic data (Wu et al., 2019). Further studies into the usefulness of ResNet versus U-Net architectures for 3-dimensional data are therefore warranted.

Section 8 - References

Aminzadeh, Fred, and Shivaji N. Dasgupta. 2013. "Fundamentals of Petroleum Geophysics." *Developments in Petroleum Science* 60: 37-92. https://doi.org/10.1016/B978-0-444-50662-7.00003-2

Babakhin, Yauhen, Artsiom Sanakoyeu, and Hirotoshi Kitamura. 2019. "Semi-Supervised Segmentation of Salt Bodies in Seismic Images using an Ensemble of Convolutional Neural Networks." *arXiv: 1904.04445 [cs.CV]*, https://arxiv.org/pdf/1904.04445.pdf.

Chollet, Francois. 2018. "Deep Learning for computer vision." In *Deep Learning with Python,* 119 – 177. New York: Manning Publications.

Dwivedi, Priya. 2019. "Understanding and Coding a ResNet in Keras." towardsdatascience. Accessed October 26, 2021. https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. "Deep Residual Learning for Image Recognition." *arXiv: 1512.03385 [cs.CV]*, https://arxiv.org/pdf/1512.03385.pdf.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. "Identity Mappings in Deep Residual Networks." *arXiv: 1603.05027 [cs.CV]*, https://arxiv.org/pdf/1603.05027.pdf.

Islam, Muhammad Saif ul. 2020. "Using deep learning based methods to classify salt bodies in seismic images." *Journal of Applied Geophysics* 178: 1-9.

Kapoor, Lohit. 2021. "Semantic Image Segmentation using UNet." Medium.com. Accessed October 27, 2021. https://medium.com/geekculture/semantic-image-segmentation-using-unet-28dbc247d63e

Kumar, Resham Sundar. 2021. "VGG16 vs ResNet50 when used as encoder in UNET." Medium.com. Accessed October 25, 2021. https://medium.com/@resham.sundar/vgg16-

vs-resnet50-when-used-as-encoder-in-unet-e93c95ee14a2

Li, Sihan, Jiantao Jiao, Yanjun Han, and Tsachy Weissman. 2017. "Demystifying ResNet." *arXiv: 1611.01186 [cs.NE]*, https://arxiv.org/pdf/1611.01186.pdf.

Mervine, Evelyn. 2011. "Geology Word of the Week: H is for Halokinesis." American Geophysical Union Blogosphere, Accessed October 24, 2011. https://blogs.agu.org/georneys/2011/07/20/geology-word-of-the-week-h-is-for-halokinesis/

Milosavljevic, Aleksandar. 2020. "Identification of Salt Deposits on Seismic Images Using Deep Learning Method for Semantic Segmentation." *International Journal of Geo-Information* 9(1), no. 24: 1-16. https://doi.org/10.3390/ijgi9010024

Parkhi, Omkar M, Andrea Vedaldi, Andrew Zisserman, and C.V. Jawahar. 2012. "The Oxford-IIIT Pet Dataset." IEEE Conference on Computer Vision and Pattern Recognition. Accessed October 10, 2021. https://www.robots.ox.ac.uk/~vgg/data/pets/

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. "U-Net: Convolutional Networks for Biomedical Image Segmentation." *arXiv: 1505.04597 [cs.CV]*, https://arxiv.org/pdf/1505.04597.pdf.

Shi, Yunzhi, Xinming Wu, and Sergey Fomel. 2019. "SaltSeg: Automatic 3D salt segmentation using a deep convolution neural network." In *SEG Interpretation* 7, no. 3: SE113 – SE122. https://doi.org/10.1190/INT-2018-0235.1.

Subramanyam, Vineeth S. 2021. "IOU (Intersection over Union)." Medium.com. Accessed November 7, 2021. https://medium.com/analytics-vidhya/iou-intersection-over-union-705a39e7acef

Targ, Sasha, Diogo Almeida, and Kevin Lyman. 2016. "Resnet in resnet: Generalizing residual

architectures." *arXiv: 1603.08029 [cs.LG]*, https://arxiv.org/pdf/1603.08029.pdf.

Tomlinson Geophysical Services Inc. (TGS). 2018. "Kaggle: TGS salt identification challenge." Kaggle. Accessed October 10, 2021. https://www.kaggle.com/c/tgs-salt-identification-challenge.

Wang, Wenlong, Jianwei Ma, and Fangshu Yang. 2018. "Automatic Salt Detection with Machine Learning." *EAGE annual 80th conference+exhibition*. doi:10.3997/2214-4609.201800917.

Wu, Xinming, Luming Liang, Yunzhi Shi, and Sergey Fomel. 2019. "FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation." *Geophysics* 84, no.3: IM35 – IM45. doi: 10.1190/geo2018-0646.1

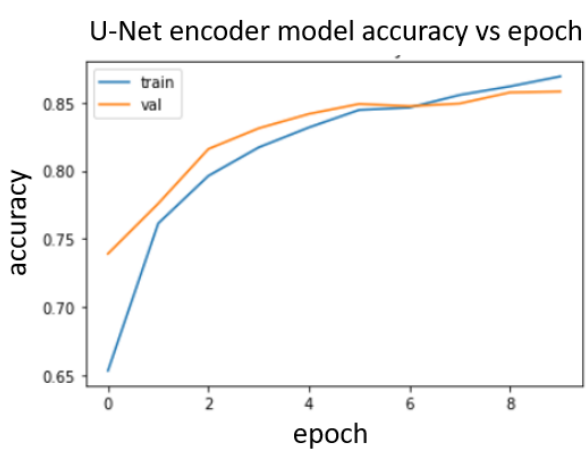
Zagoruyko, Sergey, and Nikos Komodakis. 2017. "Wide Residual Networks." *arXiv: 1605.07146 [cs.CV]*, https://arxiv.org/pdf/1605.07146.pdf.

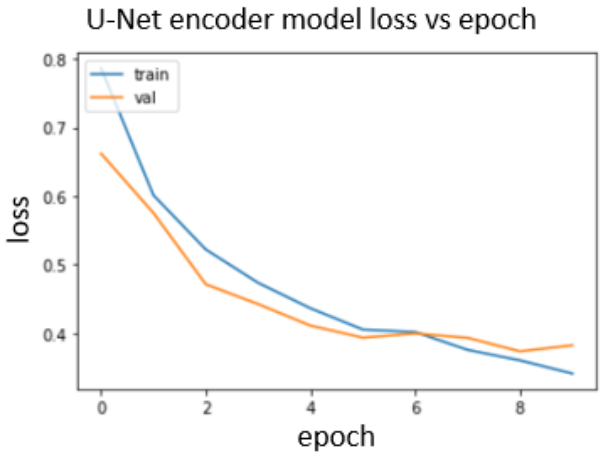
Zeng, Yu, Kebei Jiang, and Jie Chen. 2018. "Automatic Seismic Salt Interpretation with Deep Convolutional Neural Networks." *arXiv: 1812.01101 [physics.geo-ph]*, https://arxiv.org/pdf/1812.01101.pdf.

Zhang, Ke, Miao Sun, Tony Han, Xingfang Yuan, Liru Guo, and Tao Liu. 2016. "Residual networks of residual networks: Multilevel residual networks." *arXiv: 1608.02908 [cs.CV]*, https://arxiv.org/pdf/1608.02908.pdf.

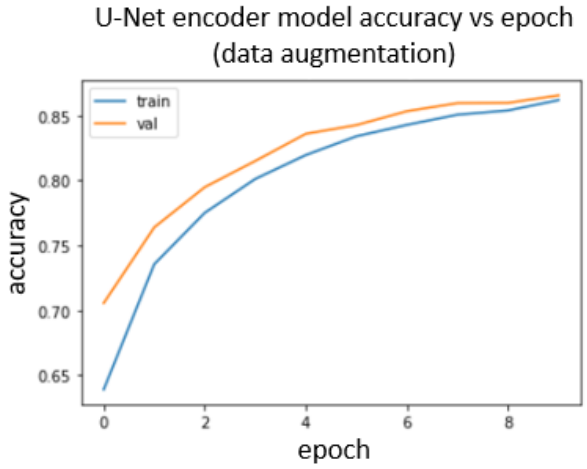
Section 9 – Appendix

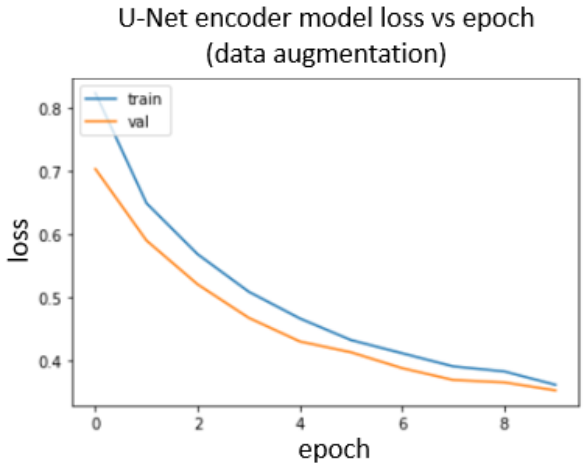
9.1 – Basic U-Net on Oxford dataset

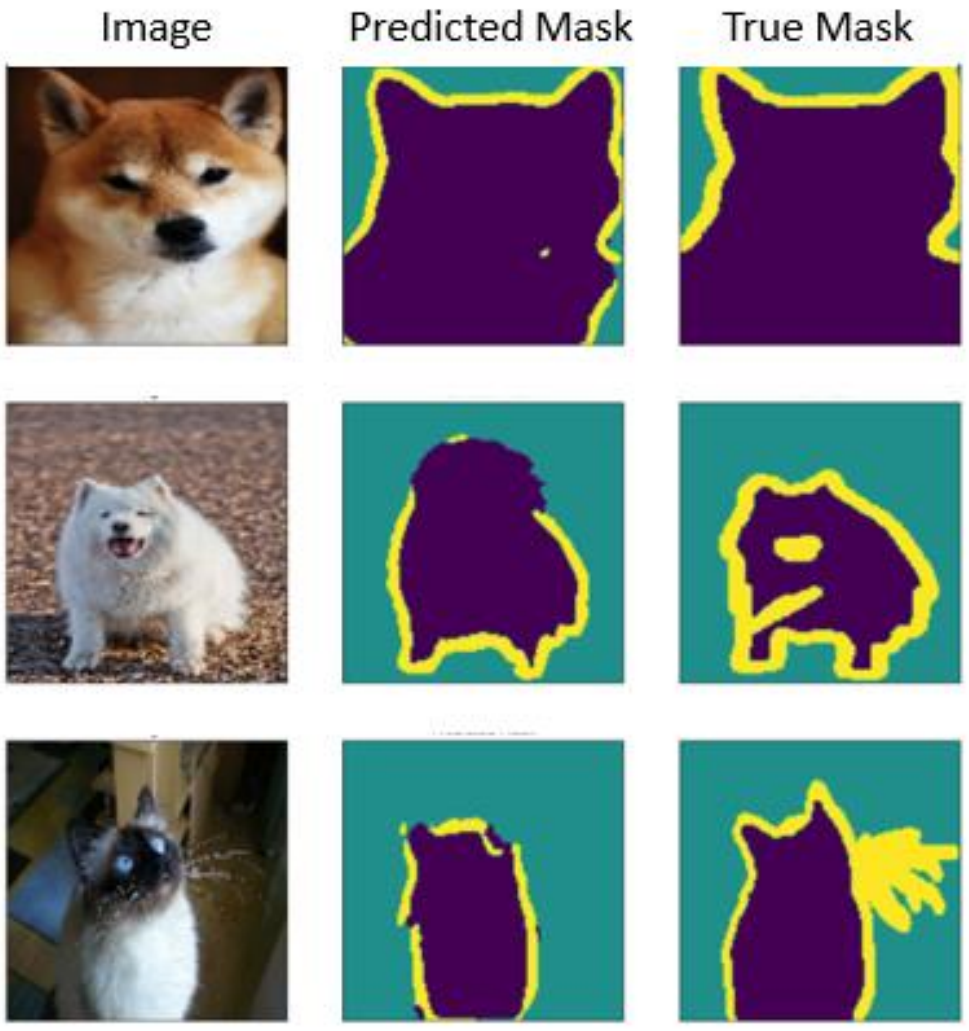




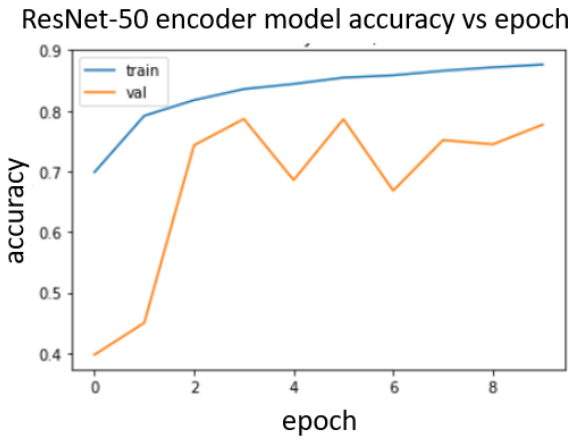
9.2 – Basic U-Net with data augmentation on Oxford dataset

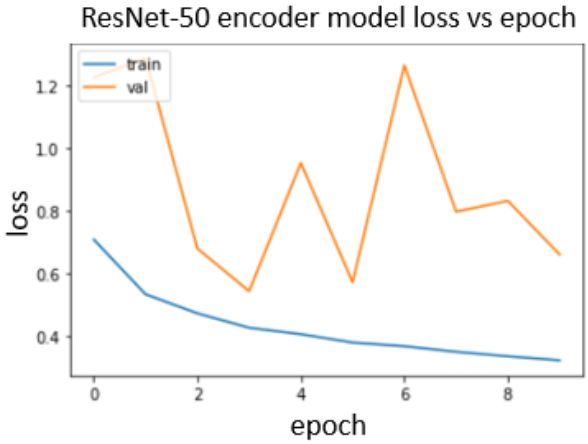


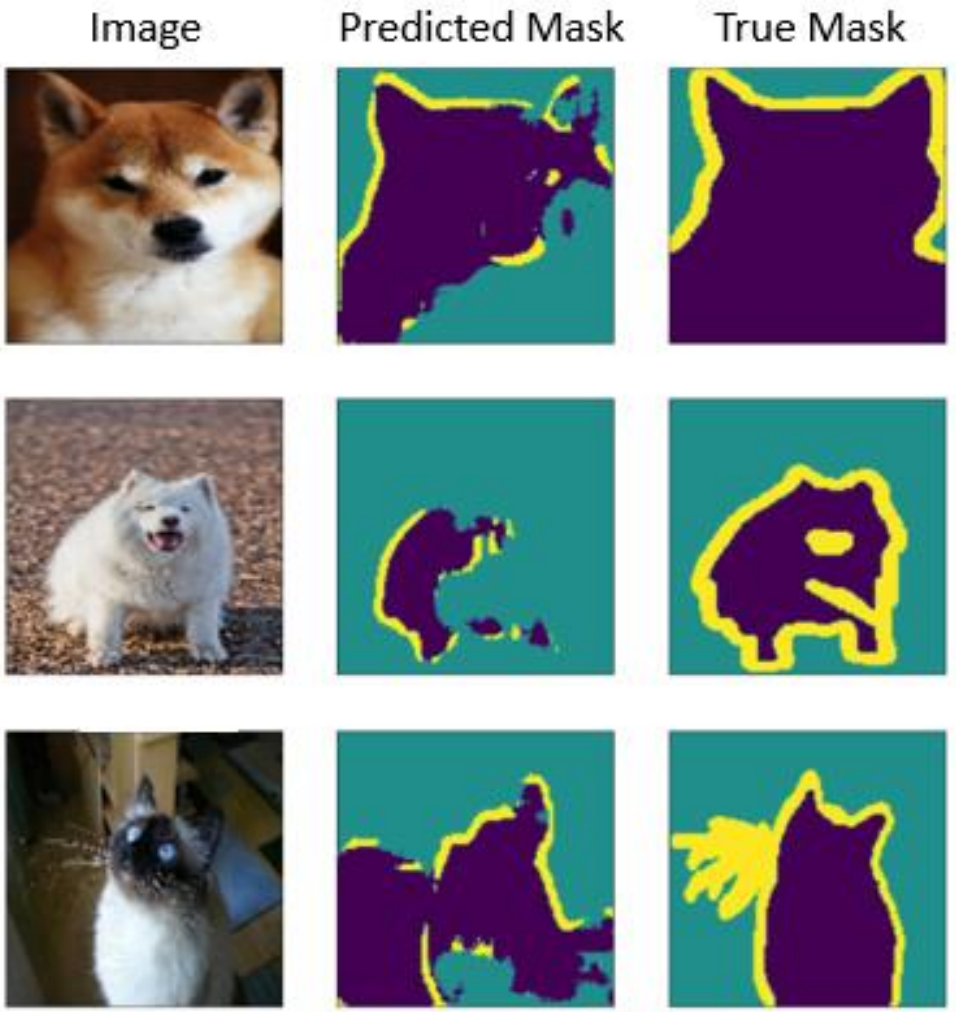




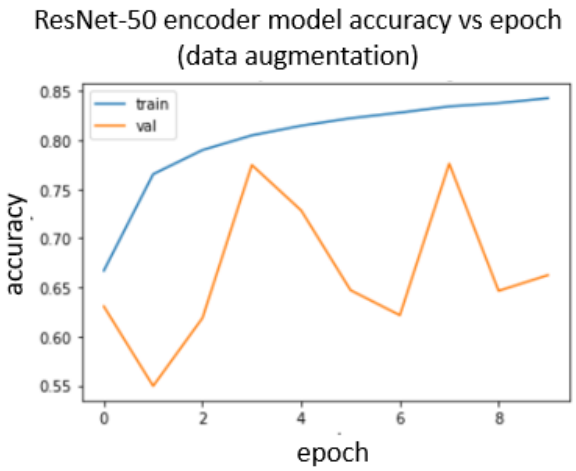
9.3 – ResNet50 encoder on Oxford dataset

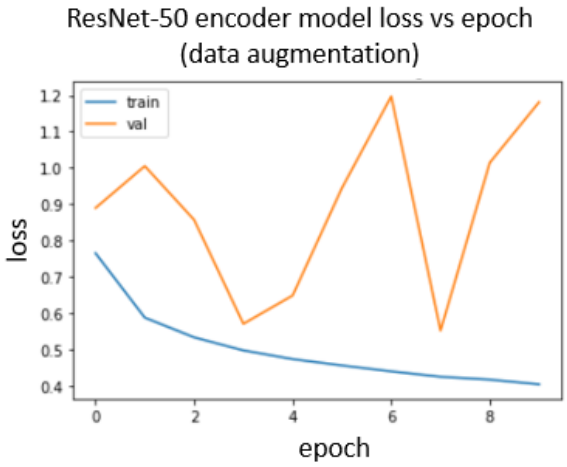


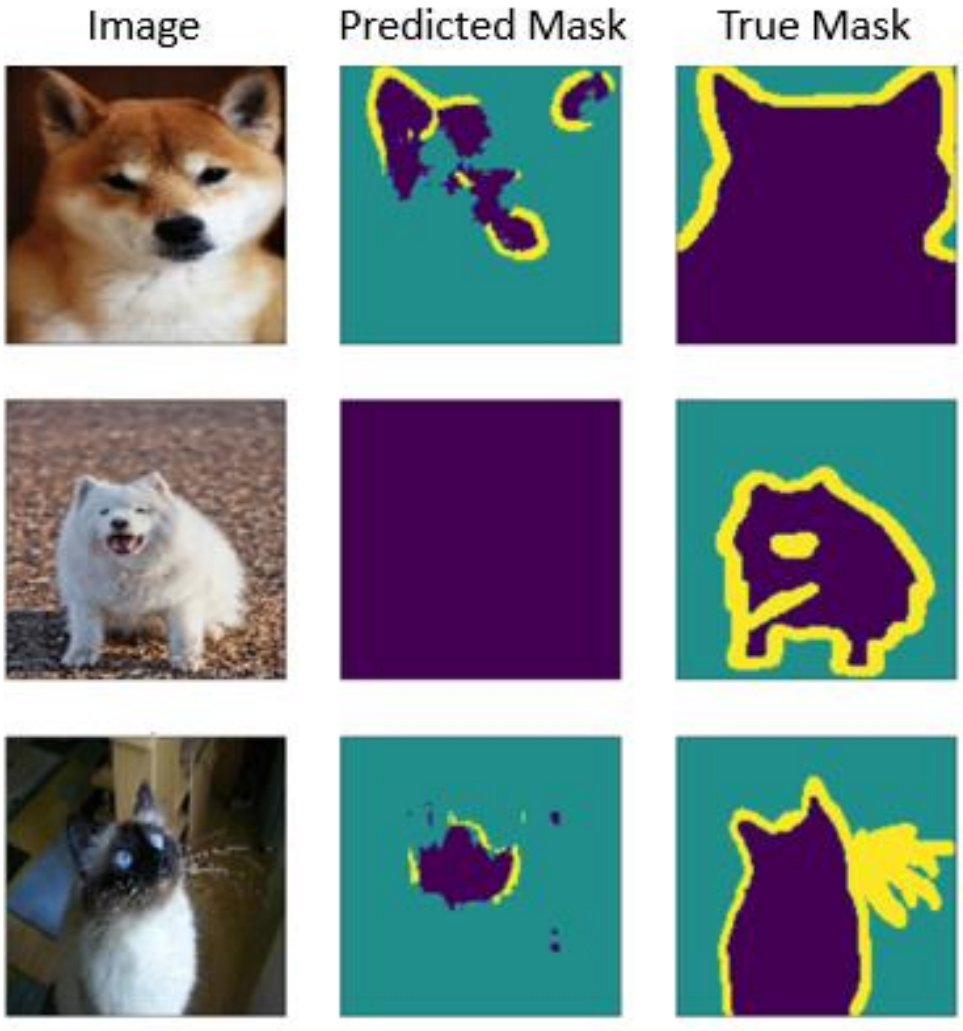




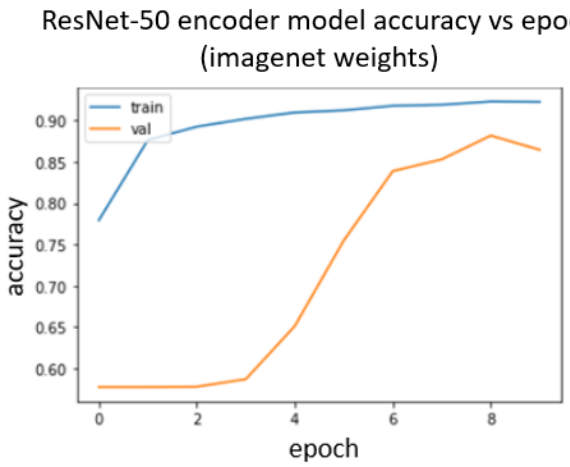
9.4 – ResNet50 encoder with data augmentation on Oxford dataset

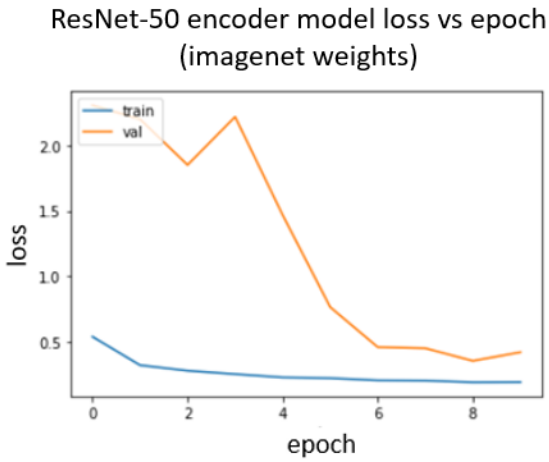


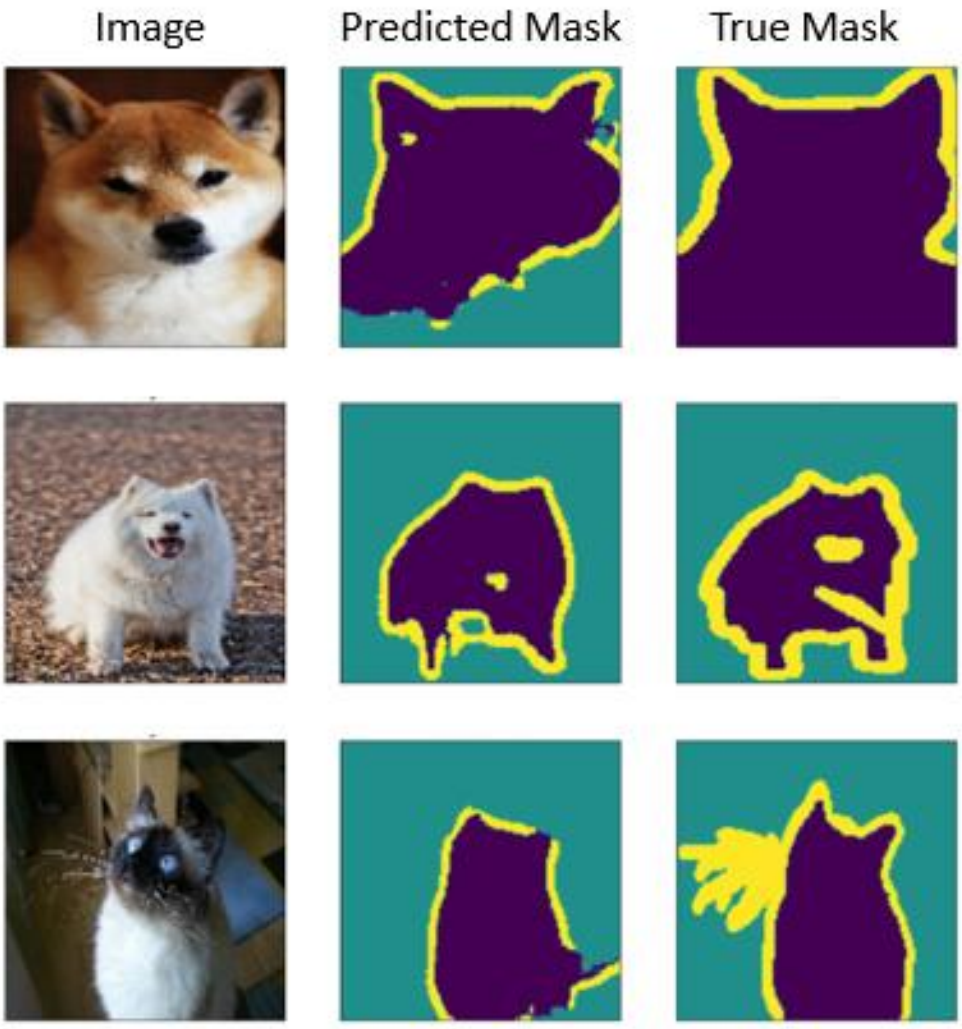




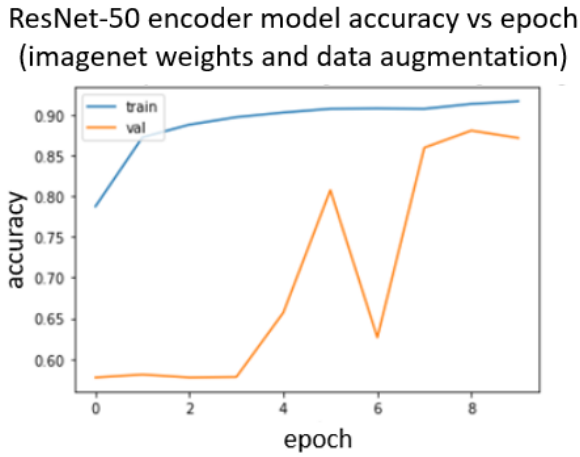
9.5 – ResNet50 encoder with imagenet weights on Oxford dataset

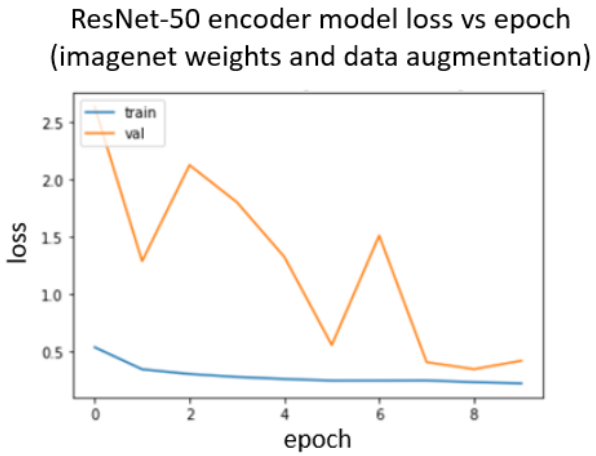


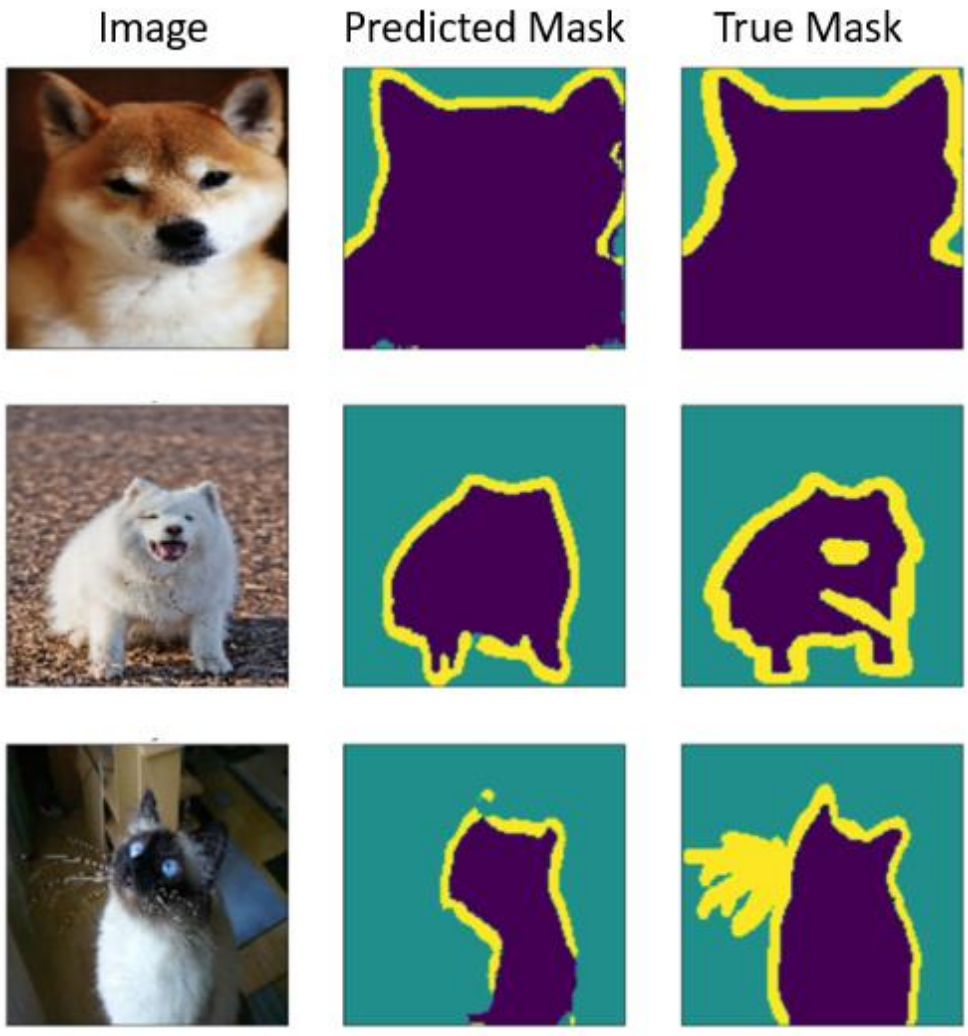




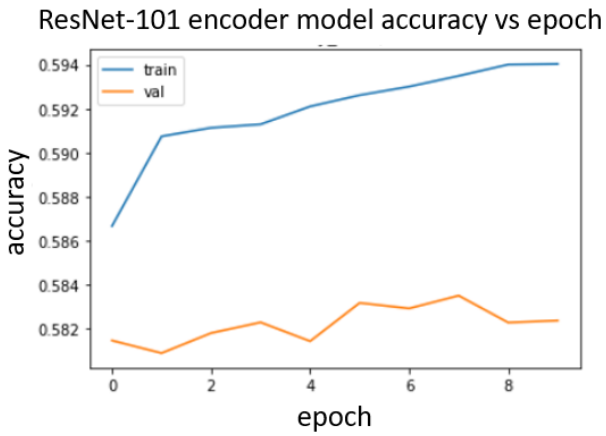
9.6 – ResNet50 encoder with imagenet weights and data augmentation on Oxford dataset

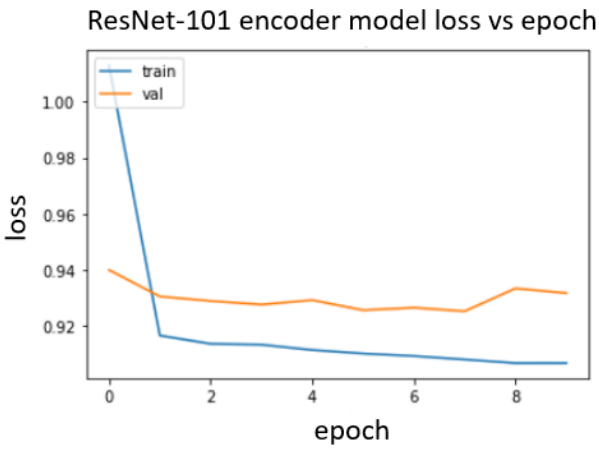


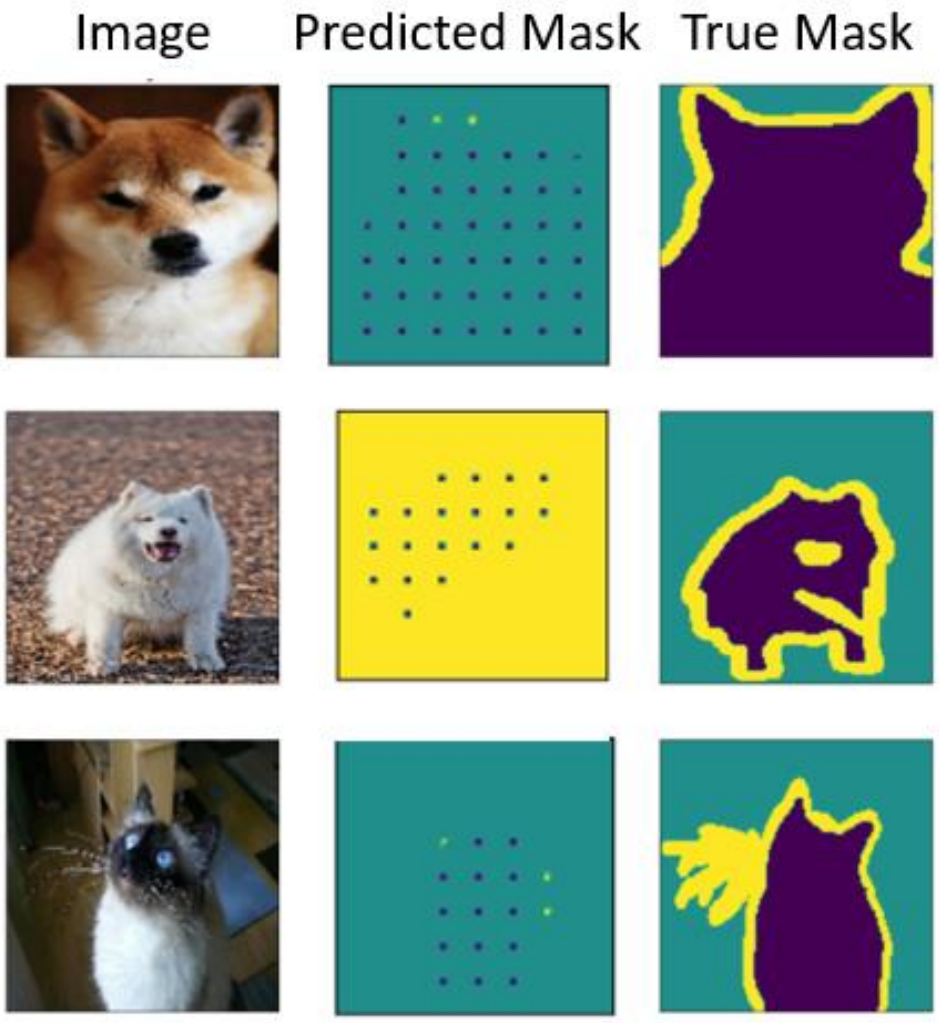




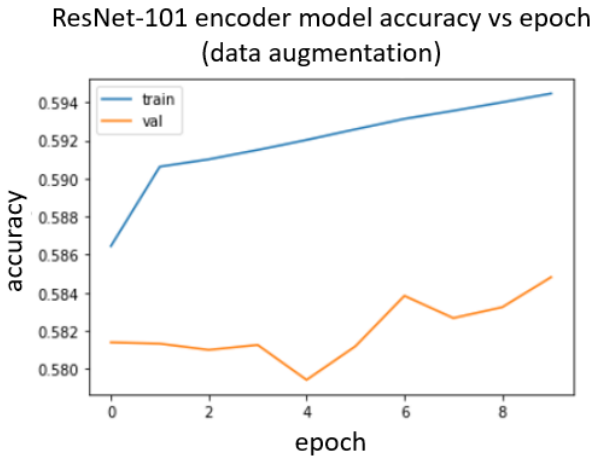
9.7 – ResNet101 encoder on Oxford dataset

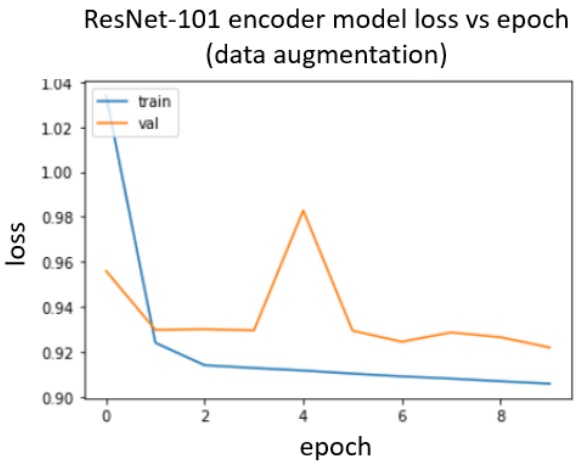


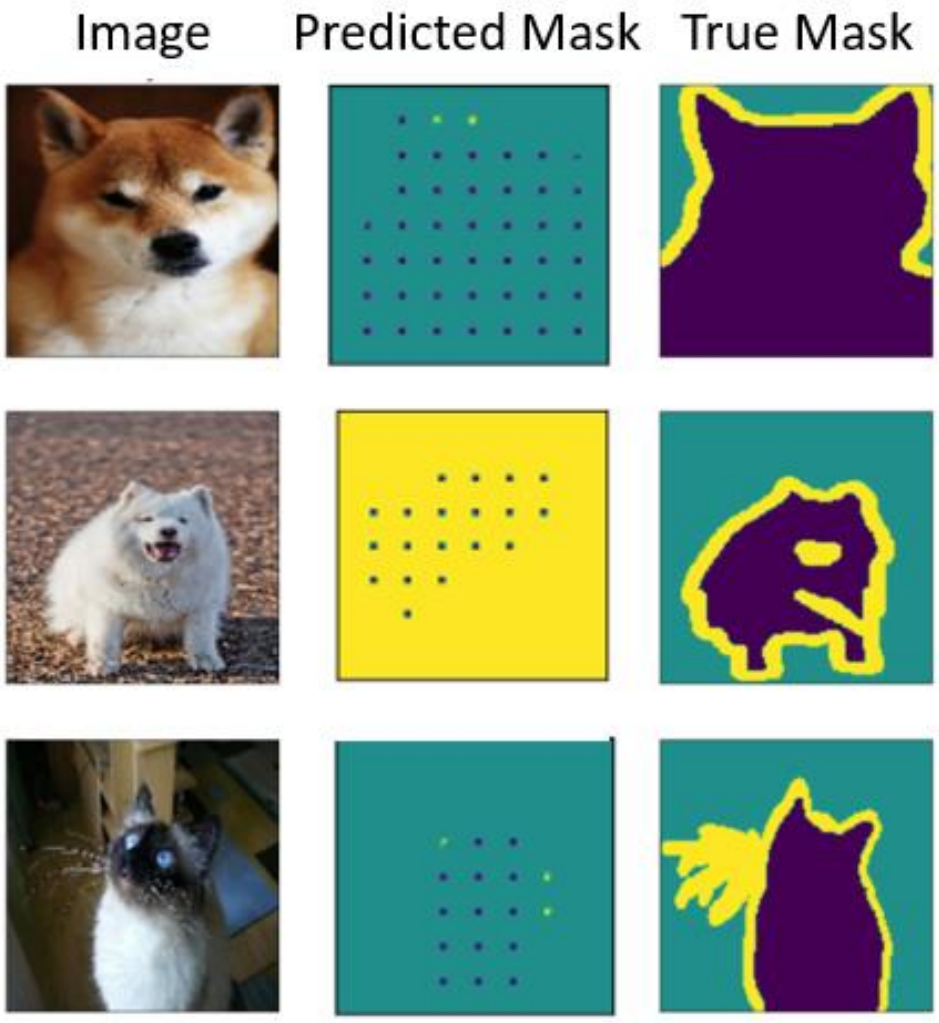




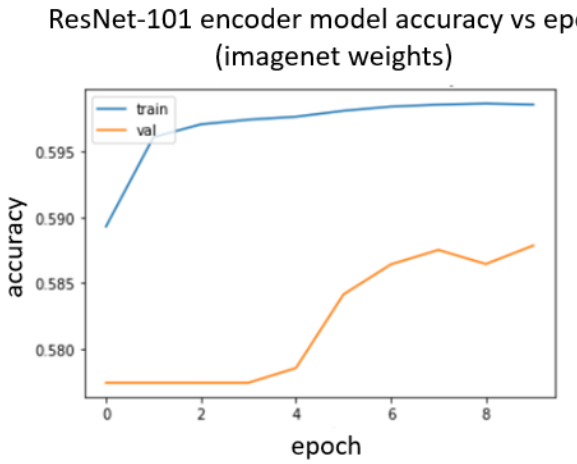
9.8 – ResNet101 encoder with data augmentation on Oxford dataset

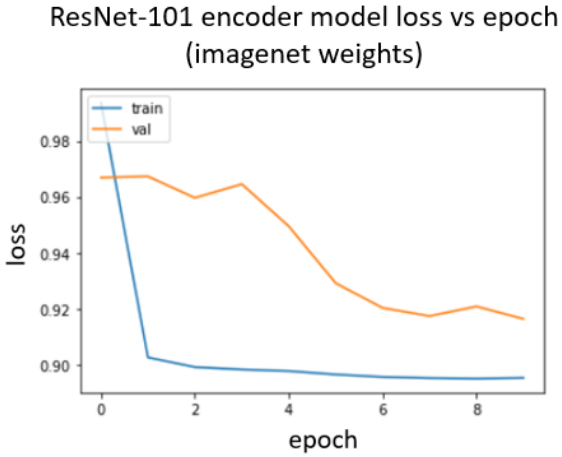


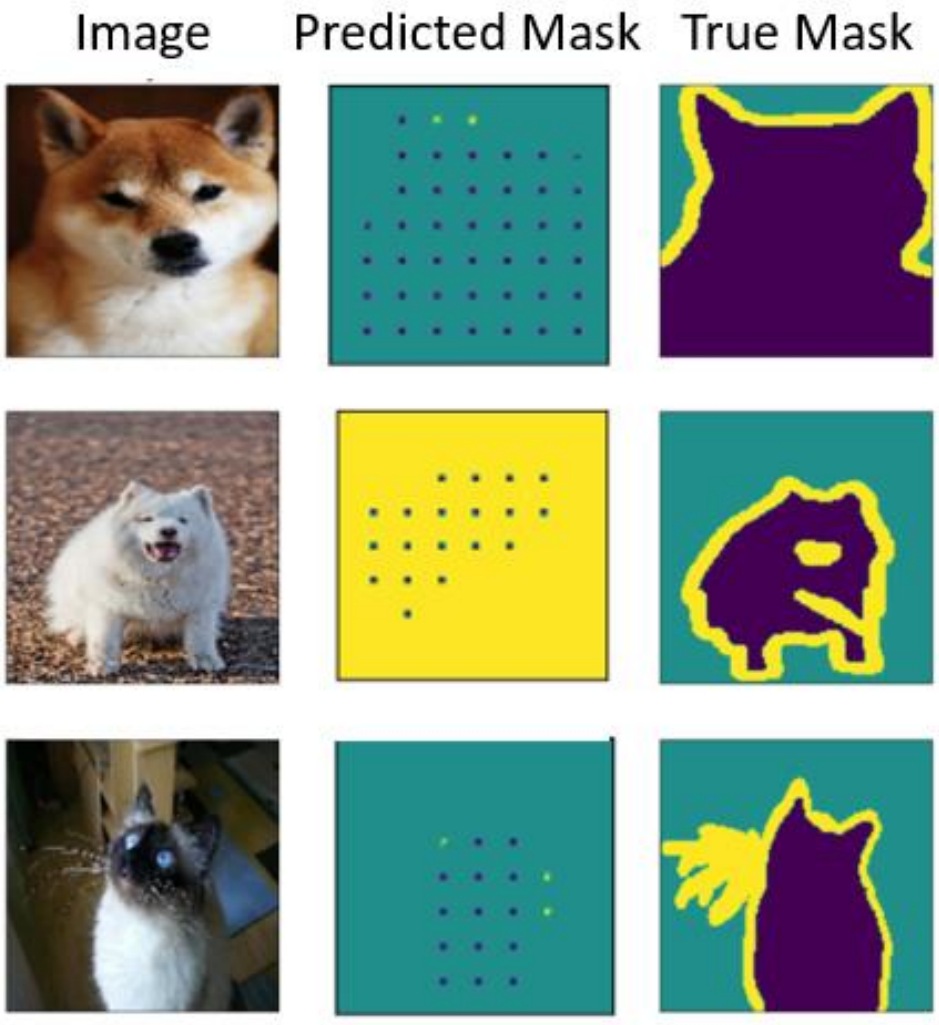




9.9 – ResNet101 encoder with imagenet weights on Oxford dataset







9.10 – ResNet101 encoder with imagenet weights and data augmentation on Oxford dataset