SEMANTIC SEGMENTATION FOR AERIAL IMAGERY USING U-NETS

THESIS

Terrence J Yi, 2nd Lieutenant, USAF

AFIT-ENG-MS-20-M-075

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-MS-20-M-075

SEMANTIC SEGMENTATION FOR AERIAL IMAGERY USING U-NETS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Computer Engineering

Terrence J Yi, B.S.E.E.

2nd Lieutenant, USAF

March 19, 2020

AFIT-ENG-MS-20-M-075

SEMANTIC SEGMENTATION FOR AERIAL IMAGERY USING U-NETS

THESIS

Terrence J Yi, B.S.E.E.
2nd Lieutenant, USAF

Committee Membership:

Robert C Leishman, Ph.D
Chair

Joseph A Curro, Ph.D
Member

Clark N Taylor, Ph.D
Member

AFIT-ENG-MS-20-M-075

Abstract

In situations where global positioning systems are unavailable, alternative methods of localization must be implemented. A potential step to achieving this is semantic segmentation, or the ability for a model to output class labels by pixel. This research aims to utilize datasets of varying spatial resolutions and locations to train a fully convolutional neural network architecture called the U-Net to perform segmentations of aerial images. Variations of the U-Net architecture are implemented and compared to other existing models in order to determine the best in detecting buildings and roads. A final dataset will also be created combining two datasets to determine the ability of the U-Net to segment classes regardless of location. The final segmentation results will demonstrate the overall efficacy of semantic segmentation for different datasets for potential localization applications.

Table of Contents

Page

List of Figures

Figure Page

List of Tables

SEMANTIC SEGMENTATION FOR AERIAL IMAGERY USING U-NETS

I. Introduction

Modern navigation technology used in practical applications heavily relies on the use of the Global Positioning System (GPS) or Global Navigation Satellite System (GNSS) to achieve accurate locations. Though the advent of GPS has been conducive to many military and commercial advancements, this innovation has created a weakness in situations where it is not available. To combat these crucial situations, a need for image-aided navigation has been sought out [5]. Extensive research has led to many advancements in this area [5, 6, 7, 8, 9, 10]; many of the current state of the art advancements have come through machine learning technology applied to this application [11]. This thesis documents an attempt to perform semantic segmentation on aerial images to obtain important features for localization. A specific convolutional neural network (CNN) architecture known as the U-Net [3] is employed in the research and applied to existing and newly created datasets.

1.1 Problem Background

Current interest is high in surveillance, combat, and commercial technology that is able to localize in an area without the use of GPS. Having the robust ability to determine location makes a system much more valuable for armed forces.

A possible step that may help to improve localization tasks from an aerial perspective is the use of semantic segmentation [12, 5, 6]. Semantic Segmentation is the ability for a machine to classify features in a given image pixel-by-pixel to create a mask. The goal for the research is to input these masks into a feature detector to

determine significant features in an image that do not vary with temporal or seasonal changes. A U-Net architecture will be utilized to perform this semantic segmentation since it is the best performing method for aerial imagery [4, 13].

1.2 Research Objectives

Current efforts to achieve accurate aerial segmentation for the purposes listed are hindered due to a lack of data in various spatial resolutions and locations. This is likely due to the difficult and time consuming nature of creating labelled masks for numerous classes for a sufficient number of images with sufficient variations in environments. Baseline comparisons for developing higher performing models are also scarce based on a lack of data and a standardized dataset. A new labelled dataset will be created using images from public datasets to accurately assess the capability of the U-Net for this application and determine the efficacy of combining existing datasets. The final goal of this research is divided into the following two objectives:

- Develop a dataset using publicly accessible aerial imagery to create a larger dataset despite different spatial resolutions and locations.
- Create U-Nets that will be able to accurately determine the presence and location of buildings or roads in an image.

The classes specified for this instance of semantic segmentation includes roads and buildings, since these are the most prominent classes that can be found in public datasets. This data will then be given to separate U-Net models, one for each class. Each model will be trained on the labelled images and their corresponding ground truth masks. Ultimately, the U-Net will be used to provide a learning-based feature detector and localization approach to determine information on which classes can be used to localize a system regardless of temporal and seasonal variations.

The methodology in developing this initial data set consists of gathering datasets from four different sources and preprocessing the images to have uniform crops and spatial resolution for the input of the U-Nets. The U-Nets will be trained on each dataset individually and have their performance measured against other architectures. The built models will then be tested using the final dataset, which combines data from two sources. The U-Net built in Keras will use the mean intersection over union (mIoU) as the loss function based on prior research [14]. The VGG16 and ResNet50 architectures found in the Keras applications will also be tested for comparison. For all models, parameters such as initializers, optimizers, and testing image size are adjusted to provide an optimized model on the dataset for future feature extraction.

1.3 Limitations and Assumptions

The greatest limitations in achieving the research objectives is the amount of data available. Optimally this research would produce a network that is invariant to spatial resolution and also able to detect buildings and roads regardless of location. However, there currently does not exist enough data at varying spatial resolutions and locations for this to be a reasonable goal.

The lack of data also means that there are very few aerial imagery datasets which are used as a baseline for comparison. Comparing currently built models to previous work is limited to the datasets used by others, which are limited to single spatial resolutions and sometimes single locations. As more accurately labelled data becomes available, a finalized dataset for aerial imagery should be created to test numerous types of invariance to compare and assess overall model performance.

This thesis is organized as follows: Chapter II describes the basics of deep learning, current architectures for semantic segmentation, and results using U-Nets [3]. This

chapter also explains why this network architecture was selected for aerial localization. Chapter III discusses the processes used to create and train the U-Net and other model architectures for comparison, along with the creation of a new dataset. This chapter also includes the methodology of the optimization of the U-Net and the respective performance measures. Chapter IV provides the results of the final optimized U-Net and the other architectures used, and the overall performance in providing an accurate output mask of the input images. Finally, Chapter V discusses the conclusions drawn from the performance of the final model, and future improvements that can be made to ultimately aid in aerial visual navigation.

II. Background

This chapter provides a technical overview of deep learning along with background information on the semantic segmentation task. In doing so, convolutional neural networks (CNNs) and their use in this image classification problem will be reviewed, along with a variation of the CNN called the U-Net. Metrics and loss functions specific to this task will also be explored, along with their effect on the final segmentations versus conventional evaluation metrics for classification.

2.1 Artificial Intelligence

To accomplish the final task for localization in varying environmental conditions, a basic understanding of artificial intelligence (AI), machine learning, and deep learning is required. AI has been defined by Chollet as the effort to automate intellectual tasks normally performed by humans. While this definition includes both machine learning and deep learning, it also includes programs in which hard-coded rules are used to perform intellectual tasks, known as symbolic AI [1, 15].

2.1.1 Machine Learning

Machine learning requires the system to be trained rather than explicitly programmed using examples relevant to the given task. The machine learning system then determines statistical structure that allows it to learn meaningful representations of the input data. These learned representations can then allow the system to develop its own rules by which the task can be automated. There are four types of machine learning: supervised, unsupervised, self-supervised, and reinforcement. This thesis utilizes supervised learning, which requires three components:

- Input data points

- Examples of expected output
- Performance metrics

The performance measures used in a machine learning algorithm connect the input and the output by providing a feedback signal to correct the algorithm outputs to better match the expected output. This allows the system to learn from the given data and transforms the input into representations that perform the best based on a given performance metric [1, 15].

2.1.2 Deep Learning

Deep learning is a subset of machine learning which takes the machine learning approach further by learning many successive layers of representations of the data. These layers make up what is known as the depth of the model. Deep learning must be utilized using neural networks to learn the many different features and representations of the data. [1, 15].

2.2 Neural Networks

Neural networks are models that are structured with layers stacked on top of each other, with each stage successively filtering the input data to learn a final useful representation for some given task [1, 15]. The structure of a basic four layer neural network used for identifying hand-written digits from the MNIST dataset is shown in Figure 1.

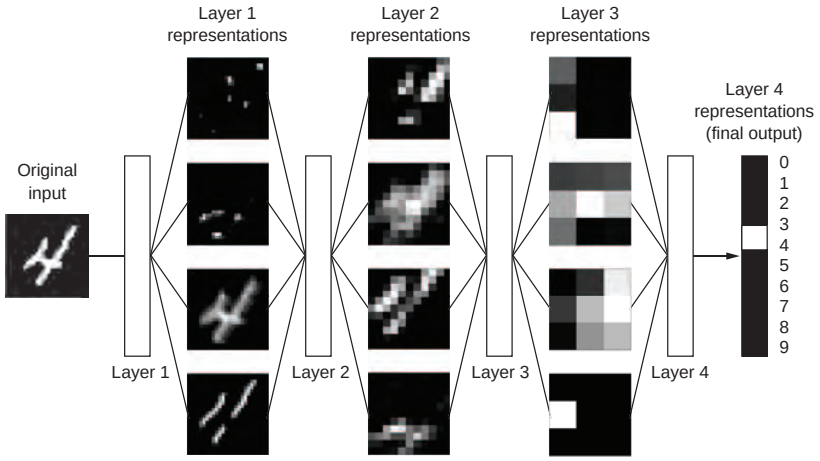

Figure 1: Four-Layer Neural Network for MNIST dataset [1]

Training a neural network involves four components:

- Layers
- Input data
- Loss function
- Optimizer

This multistage method of learning various data representations is the core principle of deep learning [1].

Layers can be defined as the fundamental data structure of a neural network. Each layer processes inputs and outputs in data containers called tensors. Layers also frequently have values called weights. The ultimate goal of training a neural network is to find the correct values for these weights, which specify what transformations the layer performs on the incoming data [1, 15]. This parameterization of the network layers is shown in Figure 2.

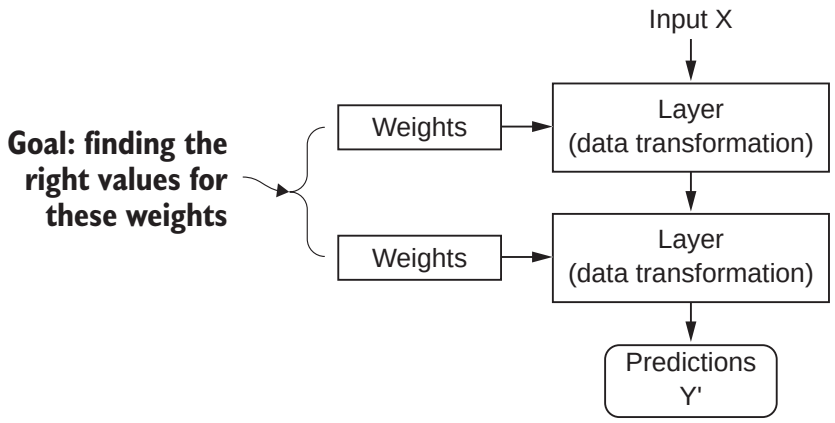

Figure 2: Parameterization of Deep Learning Layers based on Weight Values [1]

The difficulty in optimizing these networks for various tasks is the configuration of these weights, since there can exist tens of millions of parameters that depend on each other. To determine a weight configuration, a loss function is used as a measurement of how well the layer representations from the network output are performing compared to the expected output. This score is then used as a feedback signal for the optimizer to adjust the weights of the network layers using a backpropagation algorithm [1, 15].

The backpropagation algorithm starts with the final loss value and works from the top to the bottom layers to compute the contribution each parameter had in that value [1]. Stochastic gradient descent is the process of taking the derivative of the loss function at a local point in training to determine the adjustments required to minimize the loss and optimize the model to a defined evaluation metric [15].

The final depiction of a neural network training loop is shown in Figure 3. This training loop begins with the weights being randomly initialized, which leads to a high loss score since these initial weights are likely incorrect for representing the data in a useful manner. With each training batch, however, the network adjusts its weights and slowly begins to improve until the network has a minimal loss score that can

generalize to the test set [1, 15].

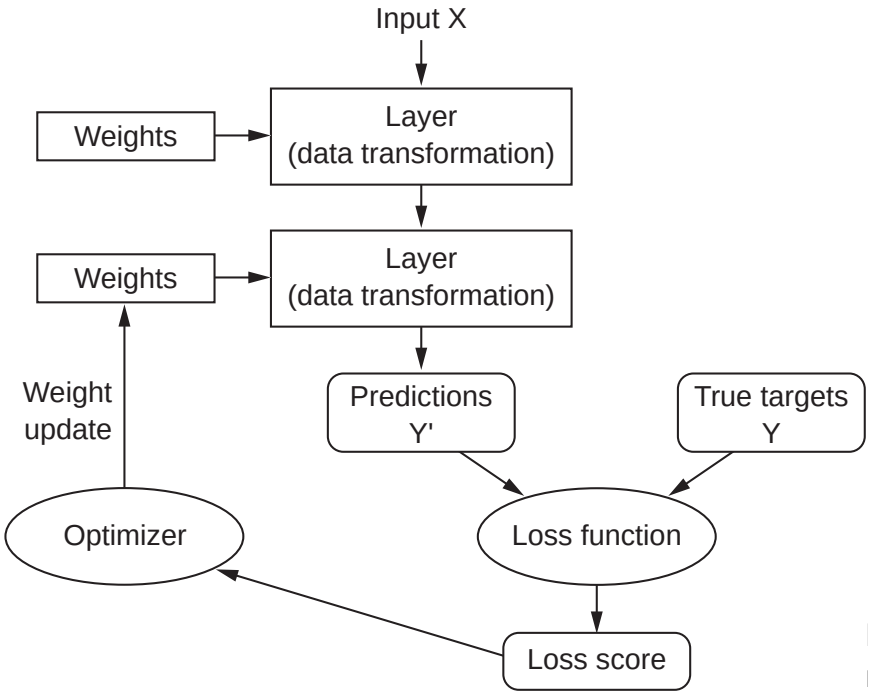

Figure 3: Training Loop for a Neural Network [1]

2.2.1 Dense Layers

Traditionally for datasets other than images, densely connected layers are used, also known as fully connected layers in which each neuron in one layer is connected to all neurons from the previous layer. However, these dense layers are only able to learn global patterns in their input feature space, whereas convolution layers are able to learn local patterns [1, 15]. This means that input images can be broken into features such as edges and textures which can be learned and are more useful than global patterns for classification purposes [1, 15].

2.3 Convolutional Neural Networks

CNNs are a type of neural network primarily used for computer vision tasks in deep learning. These networks are identical to typical neural networks, except that a convolution kernel is used in one or more of their layers instead of general matrix multiplication [15]. These types of networks are able to take image tensors as an input, determine which features in the image are important for classification or differentiation, and output these classifications [1, 16].

CNNs are used because of their ability to learn patterns that are translation invariant along with their spatial hierarchies using relevant filters. Learning translation invariant patterns means that after learning a pattern in one section of a given image, the network will be able to recognize that same pattern anywhere else in the image. This provides efficiency when processing images, leading to fewer training samples needed to learn representations that are able to be generalized [1].

Spatial hierarchies describe how earlier layers in the network will be able to learn small local patterns and increase into larger patterns consisting of these small patterns in the following layers. This also increases efficiency of the network to learn complex and abstract visual concepts. This also allows these networks to have a much higher accuracy when performing predictions of classes versus simply vectorizing a complex image with pixel dependencies throughout. Thus for any image classification task, CNNs must be utilized in order to track these important aspects and relationships between features [1].

2.3.1 Convolution

Convolution is an operation on two functions of a real-valued argument that outputs a new function. Let s(t) be the output estimation function based on time t, $x(a)$ be the input position function based on age of the measurement a, and $w(a)$ be the

weighting function which prioritizes recent measurements. The general formula for convolution using these defined functions from [15] is

$$s(t) = \int x(a)w(t-a)da. \tag{1}$$

For CNNs, the function $x(a)$ is referred to as the input, the weighting function $w(a)$ is called the kernel, and the output $s(t)$ is called the feature map [15]. The convolutions used in CNNs are performed over two dimensional tensors that are comprised of height, width, and color channels of the input images. These operations extract patches and perform transformations from the input to produce an output feature map with varying depth. The size of these extracted patches are one of the two key parameters for defining these convolutions. The second is the depth of the output feature map. This number represents the number of filters created by the layer which can encode various aspects from the input data [1].

2D Convolutions are calculated by sliding a square window of specified height and width over all pixels of the input feature map. The input feature map for images consists of three dimensions: height, width, and color bands, which generally correspond to red, green, and blue. Performing 2D convolution on these feature maps creates two dimensional patches of surrounding features. These patches are then transformed through a convolution kernel into a one dimensional vector representing the output depth. The vectors are then spatially reassembled into a two dimension output map, which corresponds to all locations in the input map [1]. The process of this convolution is shown in Figure 4.

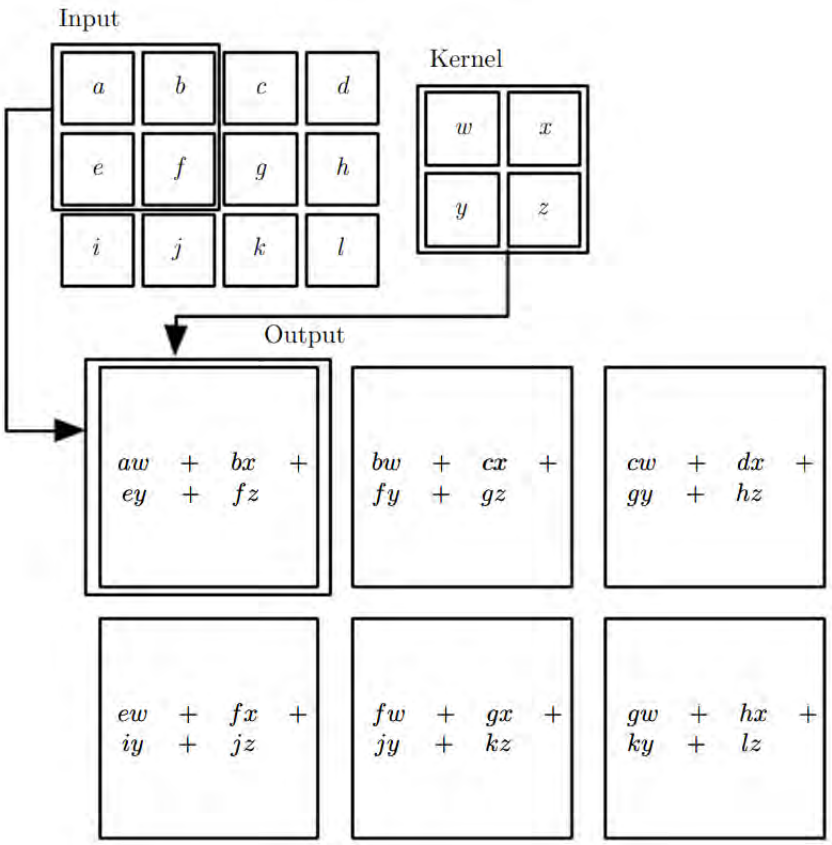

Figure 4: Visual Representation of Two Dimensional Convolution [1]

2.3.2 Strides

A factor that can contribute to a different output size in CNNs is called the stride of the convolution. Stride is a parameter of the convolution operation which defines the distance between patches that are extracted from the input feature map. With a stride of 2, the width and height of the output feature map are downsampled by a factor of 2 with no padding [1].

2.3.3 Max-Pooling

Max-Pooling is an operation used in CNNs to aggressively downsample feature maps [1, 15]. Downsampling is done for two reasons. First, downsampling reduces the

number of feature-map coefficients for the model to process, and thus can potentially prevent over-fitting to the training data. Second, it allows learning of the spatial hierarchies of filters by making successive convolution layers look at increasingly large windows [1].

Max-Pooling is done by extracting patches from the input feature maps and outputting the maximum value of each channel. Other methods of downsampling exist, such as Average-Pooling [1, 15]. However, max-pooling tends to perform better since for neural networks it is more valuable to learn the maximal presence of different features versus their average presence [1, 15].

2.4 Semantic Segmentation

Semantic Segmentation is a method of image classification in which each pixel of an image is associated with a corresponding class label [16, 5]. This image classification task currently has numerous applications such as biomedical image detection [3], autonomous driving [11], and navigation [9]. Semantic segmentation for aerial imagery is implemented in this paper for future use in a feature extractor to determine significant aspects of an image that may indicate invariance between times of day and season. Though fully convolutional networks (FCNs) can be used for this task, these networks require vast amounts of data to accurately learn image segmentations and localizations. Due to the time consuming nature of manually labelling satellite images with various classes, few datasets are publicly available with extensive pixel-wise labels. In order to combat both problems, FCNs and U-Nets are widely used for segmentation in aerial imagery [4, 13, 17].

2.4.1 Fully Convolutional Networks

2.5 Fully Convolutional Networks

One of the greatest advancements made in the task of image segmentation came in a variation of the CNN called the FCN [2]. The FCN differs from the conventional CNN by transforming the fully connected layers at the end of the CNN into convolution layers. This creates a network that computes a nonlinear filter for the output vectors in each layer. The final network is thus enabled to operate on an input of arbitrary size and produce an output of corresponding spatial dimensions. This also enables the classification network to output a heatmap of the desired object class. Further modifying the network by adding layers and a spatial loss produces an efficient machine for end-to-end dense learning [2]. An example of this transformation is shown in Figure 5.

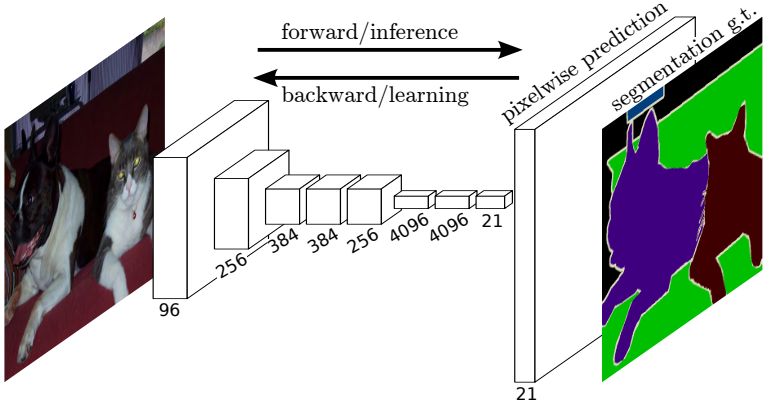

Figure 5: Fully Convolutional Neural Network making dense predictions for per-pixel tasks such as semantic segmentation [2]

The use of FCNs is not only more flexible with the ability to take various input sized images, but has also been proven to be more efficient through in-network upsampling for learning dense predictions [2]. The FCN is also able to retain the spatial

information of the input, which is crucial for the application of semantic segmentation since this task requires both localization and classification.

While a FCN is able to take an input image of any size, the output resolution is reduced using convolutions with no padding. These were introduced in order to keep filters small and computational requirements reasonable. The result is a coarse output with a reduced size by a factor equal to the pixel stride of the receptive field of the output units [2].

2.5.1 U-Nets

The U-Net architecture is a variation of a FCN, most often applied for the task of semantic segmentation. The original U-Net architecture was designed by Ronneberger et al. in 2015 to perform image segmentation and localization for biomedical use [3].

U-Nets are advantageous compared to typical CNNs due to their ability to provide localization in their output as well as classification. Localization in this case means each individual pixel in an image is labeled with a respective class. It is also advantageous compared to FCNs since U-Nets are able to work with few training images while also yielding more precise segmentations. This is done by creating upsampling layers that contain a large number of feature channels which enable the propagation of context information to higher resolution layers. U-Net segmentation networks are most valid to use in this particular application since they are the most easily scalable and sizeable FCN, avoid tradeoffs in localization and contextual information, and are widely used in state-of-the-art methods for semantic segmentation for satellite imagery [3, 4].

2.5.1.1 Network Architecture

The architecture of the U-Net consists of a contracting path at the input and an expanding path at the output. The basic architecture of this network is introduced in Ronneburger's paper, shown in Figure 6 [3].

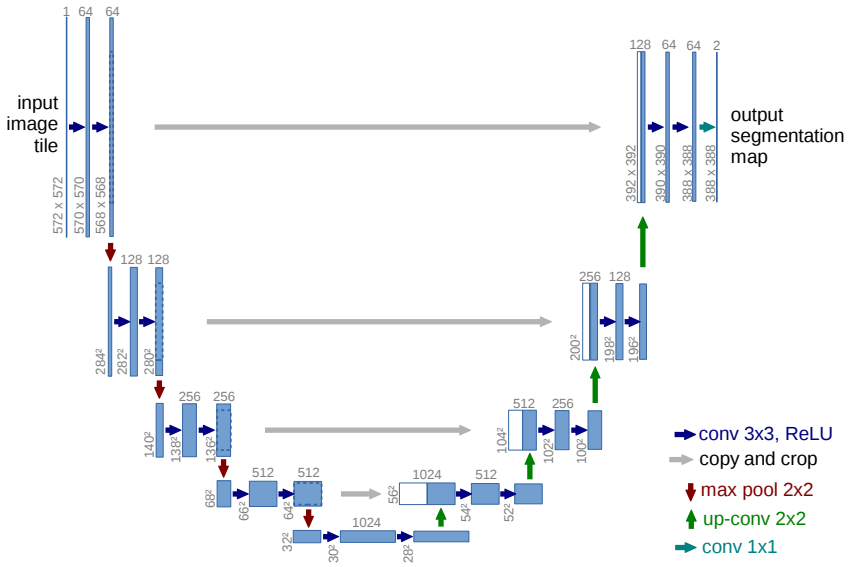

Figure 6: Architecture for the U-Net where each blue box corresponds to a multi-channel feature map with the number of channels at the top. The size of the image is shown at the lower left edge of each box, and the white boxes represent copied feature maps. The contracting path is found on the left side and the expanding path is on the right with arrows denoting concatenation of feature maps [3].

The contracting path of the U-Net follows the traditional architecture of a CNN, consisting of repeating 3x3 convolutions and max pooling operations with stride 2 for downsampling. These downsampling operations double the number of feature channels, and are then followed by upsampling the feature map and a 2x2 convolution for the expanding path. These operations halve the number of feature channels, and are then concatenated with the cropped feature map from the contracting path. This

is then followed by additional convolution layer to finally map each feature vector to the desired number of classes [3].

By using the contracting path, the U-Net is able to capture context of the image, while the symmetric expanding path enables precise localization. Localization is enabled by combining the high resolution features from the contracting path with the upsampled output. The upsampling portion of the model also has a large number of feature channels to allow the network to propagate context information to higher resolution layers [3].

2.5.1.2 Results in Aerial Imagery

Though initially intended for the purposes of biomedical image segmentation, the U-Net has been the primary architecture used for problems involving segmentation of aerial imagery [4, 13, 17]. An example of this is through the 2017 Kaggle competition entitled the "DSTL Satellite Imagery Feature Detection" challenge. This competition required participants to take high resolution aerial imagery and classify 10 different defined classes using pixel-wise segmentation. The dataset consisted of 57 images divided into 25 training images and 32 test images. All of the top entries in this competition used some variation of the U-Net due to its ability to combine low-level feature maps with higher-level ones, enabling precise localization [13].

U-Nets were also utilized in a paper written by Khalel and El-Saban on the automated pixel labeling for both the INRIA Aerial Imagery dataset and the Massachussetts buildings dataset [17]. In this instance, the authors cascade two U-Nets together, with the second one acting as a post-processor for the output of the first. By doing so, the stacked U-Net architecture outperforms state-of-the-art models on these datasets with a 74.55 mean intersection over union (mIoU) score and 96.05% pixel accuracy level for the INRIA Aerial Imagery dataset and a 0.9633 precision-recall breakeven

point for the Massachussetts buildings dataset. This further supports the importance and efficacy of U-Nets for both biomedical and aerial imagery [17].

2.5.2 Evaluation Metrics

To measure the performance of various network architectures for semantic segmentation, a proper metric that accurately describes the network's ability to identify a class is required. While other metrics will be explored, the primary metric used for this thesis will be the mIoU.

2.5.2.1 Pixel Accuracy

Pixel accuracy is the most basic method of testing overall accuracy for the classification of images. This measure is the result of calculating the percentage of pixels correctly identified from a given test image. Often times for the semantic segmentation task, however, this measure is not utilized.

Because many of the classes for various datasets are imbalanced, a network can be naively regarded as an excellent performing classifier if it was optimized for pixel accuracy. For example, take a network that classifies images between background and a defined object class. If a given image consists of 90% background, the network would predict every pixel as the background class and achieve a high accuracy without learning any segmentation [18].

2.5.2.2 Jaccard Index (Intersection over Union)

To combat the unreliable measure of accuracy from the previous method, a different performance measure is more commonly used called the Jaccard Index. This metric is also referred to as the intersection over union (IoU). This metric calculates the similarity between the predicted region and the ground-truth region for an object

present in a set of images. The equation used to calculate the IoU is shown in (2), where TP is true positive, FP is false positive, and FN is false negative between the prediction and the ground truth images. A corresponding diagram is also shown in Figure 7 [4, 18, 14].

$$\text{Jaccard} = \frac{TP}{TP + FP + FN} = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{2}$$

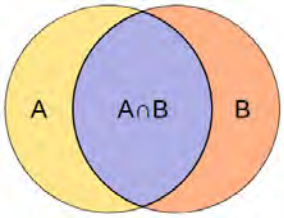

Figure 7: Visual representation of Jaccard Index, where A is the ground truth data and B is the output prediction for a single class [4].

The IoU equation is more accurate at evaluating a network since it more heavily punishes incorrect classifications and rewards only correct ones by not including true negatives as a factor in performance. For binary or multi-class segmentations, the average of each class IoU is taken, known as the mIoU. For this thesis, only one class is evaluated at a time: buildings or roads. Because of this, the IoU and mIoU values will be the same [4, 14, 18].

2.5.3 Loss Functions for Segmentation

Much like the evaluation metrics used to more accurately determine the efficacy of a model, different loss functions exist for this segmentation task. To achieve the best performance possible, the loss function should optimize the network to achieve a better IoU accuracy rather than the basic pixel accuracy. The widely used binary cross-entropy loss function will be explored for this application, but ultimately an

approximation of Equation (2) is used to develop the final loss function used in this research.

2.5.3.1 Binary Cross Entropy

Binary cross entropy, or log loss, is considered to be a baseline loss function for many classification and segmentation works [3, 16]. However, studies in [19] conclude that this loss is often used without considering more useful alternatives based on the given task. The formula for binary cross entropy is

$$\mathrm{L}_{BCE} = \sum_{x} -\left(T_x \log\left(P_x\right) + \left(1 - T_x\right) \log\left(1 - P_x\right)\right), \tag{3}$$

in which T is a single image of labels used as truth data, T_x is a single element of T, and P_x is a single element of the output prediction mask of the network.

Because this function still awards both true positives and false negatives, the resulting network becomes more prone to naive solutions such as classifying all pixels based on the majority. For this reason, it is likely more suitable to develop a loss function that directly incorporates the evaluation metric that is desired [14].

2.5.3.2 Intersection over Union as a Loss Function

It is important to note that the mIoU evaluation metric cannot be used as a loss function by itself since it is not differentiable. The calculation for IoU in (2) assumes that the output prediction mask consists of 1's and 0's. However, the actual output of neural networks is an array of probabilities representing the likelihood the network predicts a pixel belongs in a particular class. Because of this, the mIoU score cannot accurately be measured at the output of the network. To allow this, an approximation must be made using the output probability values [14]. This approximation is

$$\mathrm{I}oU' = \frac{|T * P|}{|T + P - (T * P)|} = \frac{I}{U}, \tag{4}$$

where T is the truth data for an image and P is the prediction mask of the same image, averaged throughout the entire set to create an IoU value between 0 and 1.

Because this new approximation uses arithmetic operations versus the set operations shown in (2), it is now differentiable and able to be used as a loss function [14]. Because the goal of a loss function is to be minimized, the final loss function used for semantic segmentation is

$$\mathrm{L}_{IoU} = 1 - IoU'. \tag{5}$$

Based on previous research and the architectures discussed in this chapter, the U-Net and some variations will be implemented for semantic segmentation in this thesis. To evaluate the final models, mIoU will be used as the baseline for all comparisons. Because of this evaluation metric, a loss function based on the mIoU will be used to maximize the final scores for each model. Chapter 3 will discuss the implementation of the models, the datasets used, and the required processing done for the input images and the final test sets.

III. Methodology

This chapter discusses the methods used in this research to achieve semantic segmentation. Section 3.1 explains the datasets used for this thesis and the necessary preprocessing required to optimize and run the models. Section 3.2 discusses the various model architectures used, along with the evaluation metrics and loss functions utilized to optimize the final models for comparison. Finally, section 3.3 will cover the processing for the final test sets and determining the overall accuracy for the outputs of the final models.

3.1 Dataset

Currently there is a lack of abundant aerial imagery data with pixel-wise labels for semantic segmentation. This is caused by the fact that labelling multiple classes of objects in high resolution images is an incredibly time-expensive task. This section will discuss an initial approach of manually hand-labeling data, the final datasets used, and the preprocessing required.

3.1.1 Initial Labeling

The dearth of accurately labeled aerial images resulted in an initial approach of creating a new dataset. This new dataset consisted of randomly augmented images from existing ANT Center flight test data around the area of Dayton, Ohio. The algorithm used for this data procuring process is described in [12]. Using this algorithm, 1000 images were randomly extracted from the bank of flight test data. Each of these images were then labeled using a Microsoft Surface tablet and the Image Labeler application within the computer vision toolbox in MATLAB. Three classes were labeled: buildings, roads, and clouds. An example of one of the many labeled

images is shown in Figure 8

Figure 8: Sample of initial hand-label approach, with streets labeled as lines in the images.

This initial approach was abandoned due to time-constraints and lack of precision. Other datasets were discovered that were shown to have higher accuracy of labels, though the number of labeled classes was less than the number in which this research was originally intended. This lack of data prevents a single network from being capable of classifying multiple key features in an image at once, such as bodies of water, buildings, roads, trees, cropland, etc. However, the varying geographic locations of roads and buildings found in the final datasets allows the final network to gain some invariance in detecting these classes with changes in environment, which can prove to be useful in future navigation research.

3.1.2 Existing Datasets

Currently existing datasets that can be applied for the semantic segmentation task for aerial imagery consist of binary classifications of single classes. While some exist

that contain multiple classes in a single image, the only ones that contain an adequate number of samples were images that classified roads and images that classified buildings. The datasets for buildings are then combined together to create a larger dataset and for the created models to gain some invariance on location of the images. The datasets for roads were not combined due to the large imbalance in number of samples.

3.1.2.1 INRIA Aerial Image Labeling Dataset

The INRIA Aerial Image Labeling dataset was created for achieving automated pixelwise labeling for aerial images. The images provided are three-band, colored orthorectified images, and cover 810 square kilometers of land, 405 for training and 405 for testing. The spatial resolution of these images are 0.3 meters per pixel, and the ground truth data for the classes are "building" and "not building" [8].

This dataset consists of different cities in the training and test sets. There are 36 images for each region, which have dimensions of 5000x5000 pixels and cover an area of 1500x1500 meters. The training set consists of images over Austin in Texas, Chicago in Illinois, Kitsap County in Washington, Western Tyrol in Austria, and Vienna in Austria, with the first five images of each city used as validation. The test set consists of images over Bellingham in Washington, Bloomington in Indiana, Innsbruck in Austria, San Francisco in California, and Eastern Tyrol in Austria. Because the test set analysis is only available through submission to the INRIA website, however, the validation data will be used for testing. Sample images of some of these different cities and their ground truth data are shown in Figure 9.

Figure 9: Sample of INRIA Dataset

The original intent of separating the cities this way was to make networks learn from the data provided in the training set and be scored on their ability to generalize to other regions of varying urban densities and architectures. This allows the network to achieve a degree of robustness to these variations, along with invariance to illumination conditions, urban landscape, and time of year [17, 8].

3.1.2.2 Massachusetts Dataset

The Massachusetts Dataset consists of two different datasets: one for the detection of roads and the other for the detection of buildings. The Massachusetts Buildings Dataset consists of 151 aerial images covering urban and suburban regions of Boston. Each image has a resolution of 1500x1500 pixels for an area of 2.25 square kilometers, making the spatial resolution of the dataset 1 meter per pixel. This dataset was randomly split into a training set of 137 images, a test set of 10 images, and a validation set of 4 images [20]. A sample of this dataset is shown in Figure 10.

Figure 10: Sample of Massachusetts Buildings Dataset

The Massachusetts Roads Dataset consists of 1171 aerial images of the entire state of Massachusetts. These images maintain the resolution of 1500x1500 and spatial resolution of 1 meter per pixel. The data was again randomly split by the author into a training set of 1108 images, a validation set of 14 images, and a test set of 49 images [20]. A sample of this dataset is shown in Figure 11.

Figure 11: Sample of Massachusetts Roads Dataset

3.1.2.3 Roadnet Dataset

The Roadnet Dataset was one of the first comprehensive datasets created for the road detection task, and contains images and masks of the classes "road" and "not road". These high resolution images are divided into 14 for training and 6 for testing. Though these images are of varying height and width, they are all captured with a spatial resolution of 0.21 meters per pixel covering 21 urban areas of Ottawa, Canada [21]. A sample of this dataset is shown in Figure 12.

Figure 12: Sample of Roadnet Dataset

3.1.3 Preprocessing

Due to current computational limitations, full sized high resolution imagery is unable to be input into a CNN and trained efficiently. This creates a need to preprocess the images from the datasets into smaller augmented patches in which the models train and test. The final test images are then stitched back together from the patches to create a final prediction mask. Spatial resolutions were also normalized for use in the final combined buildings dataset.

The resolution of the input image crops was chosen to be 224x224 pixels since this is a common resolution for semantic segmentation [13, 17, 1], and also because of the computational resources available. Image data generators were created for use with Keras, which yield batches of 32 randomly augmented 224x224 crops from the

high resolution images with each step in the epoch. The random augmentations used for this project include horizontal flips, vertical flips, and rotations. A fill mode of "mirror" was used, which takes the empty portions of the image caused by rotation and mirrors the actual image until those portions are filled. These augmentations are done to allow the model to learn from different perspectives, since aerial imagery is not always going to be uniform to one orientation. This also artificially generates more training data for the model to more confidently learn and generalize [1].

The INRIA and RoadNet datasets were tested using a 85-15 split for training and validation, which is used to check the performance of the model as it trains. The Massachusetts datasets, however, already had validation images provided, which were used instead.

The training and validation sets of the Massachusetts Buildings dataset were combined with the INRIA dataset to create the separate Buildings dataset. Before these datasets were added together, however, the INRIA dataset was resized to the same scale as the Massachusetts Buildings dataset so that both would have the same spatial resolution. This was done using the OpenCV library in Python to resize the INRIA images to a resolution of 1500x1500 for a matching spatial resolution of 1 m. Resizing was done since the datasets likely do not have enough data to allow the networks to learn different locations as well as different spatial resolutions. These training sets were then also divided using an 85-15 split.

3.2 Model Architectures

This section discusses the various model architectures used in this investigation. These models were tested and compared against other models from previous research to determine their ability to sufficiently learn from the datasets [8, 20, 21].

3.2.1 U-Net

The baseline U-Net architecture used for this thesis is mostly unchanged from the original model in [3]. The diagram for this architecture is shown in Figure 13. The contracting path of the network consists of repeating 3x3 convolutions followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. Each downsampling step doubles the number of features, which are then halved in the expanding path. This is done by performing upsampling followed by 2x2 convolutions, and concatenating the output with the corresponding feature map from the contracting path. The final layer uses a 1x1 convolution to map the feature vector to the number of classes, which is one for binary classification. The primary difference between the finalized U-Net architecture and the original is the addition of batch normalization between the convolutions and ReLU activation functions for faster convergence [17].

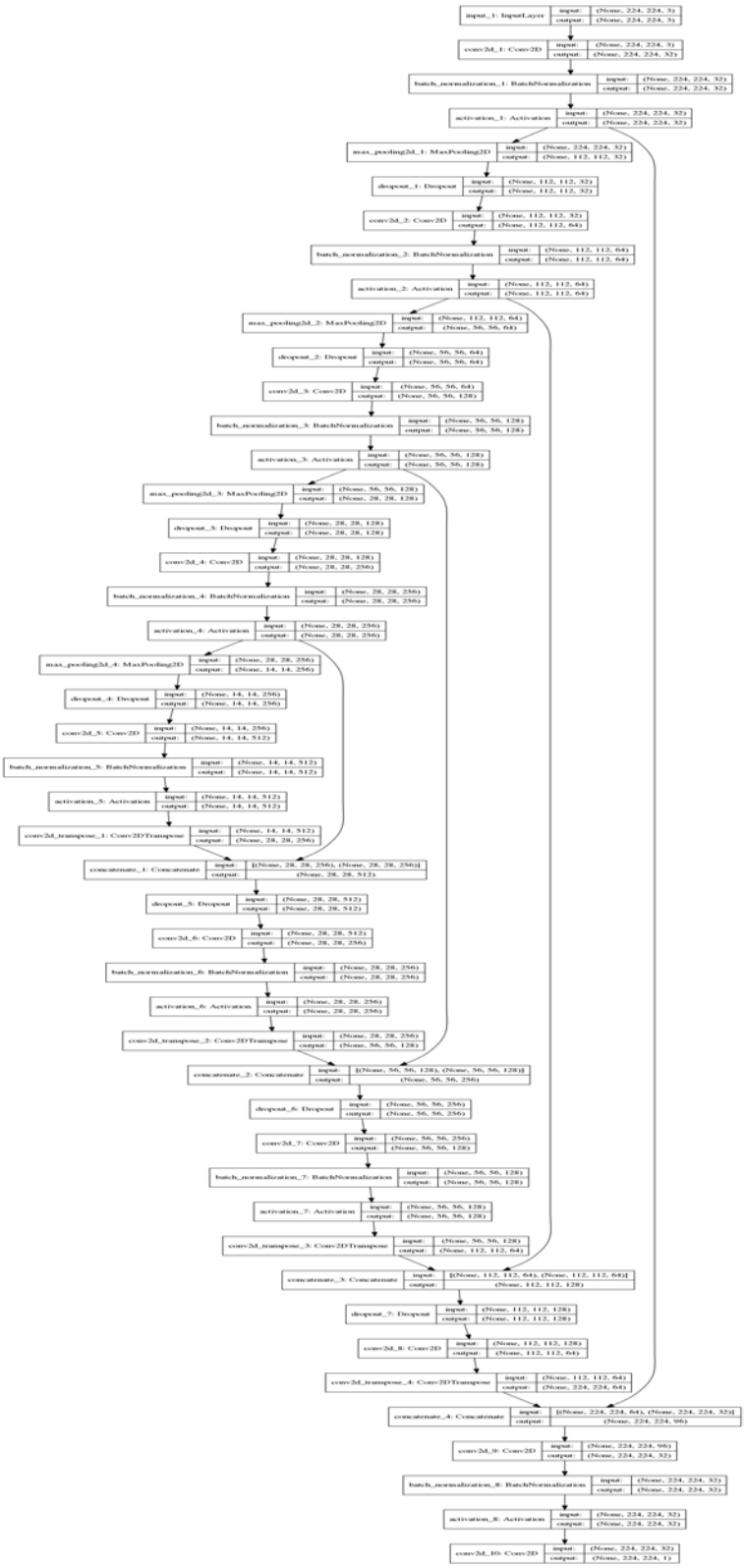

Figure 13: Final U-Net Model Architecture

This version of the U-Net will be used as the baseline model for all comparisons. It will also be compared against the findings in [17], which show an increase in classification performance by stacking two U-Nets. This means a second U-Net model will be compiled that consists of two simply stacked U-Net architectures. Another architecture consisting of three stacked U-Nets will also be created and tested to test the theory that simply stacking U-Nets in succession will lead to a drastic increase in performance.

All final models for this thesis utilize the Adam optimizer with Nestorov momentum based on its greater performance and use in other state-of-the-art U-Net models [22, 17]. All models also use the He uniform variance scaling initializer for the same reasons [23, 17].

3.2.2 Pre-trained Models

Within Keras various pretrained models widely used for segmentation are available for use. Two of the most frequently used of these available models are ResNet and VGGNet. These models are all pre-trained to identify over 1000 classes using the ImageNet dataset, which consists of various images. These models will be tested alongside the U-Net to determine their effectiveness in learning semantic segmentation of aerial imagery.

3.2.3 VGG16

The VGG-Net architecture was developed to determine the effect of increasing the depth of a network by adding small convolution filters. The study resulted in the VGG network that consisted of 16-19 weight layers with a state-of-the-art performance in large-scale image classification at the time [24]. This study confirmed the importance of depth in visual representations. The original design of the 16 layer model is shown

in Figure 14 [24].

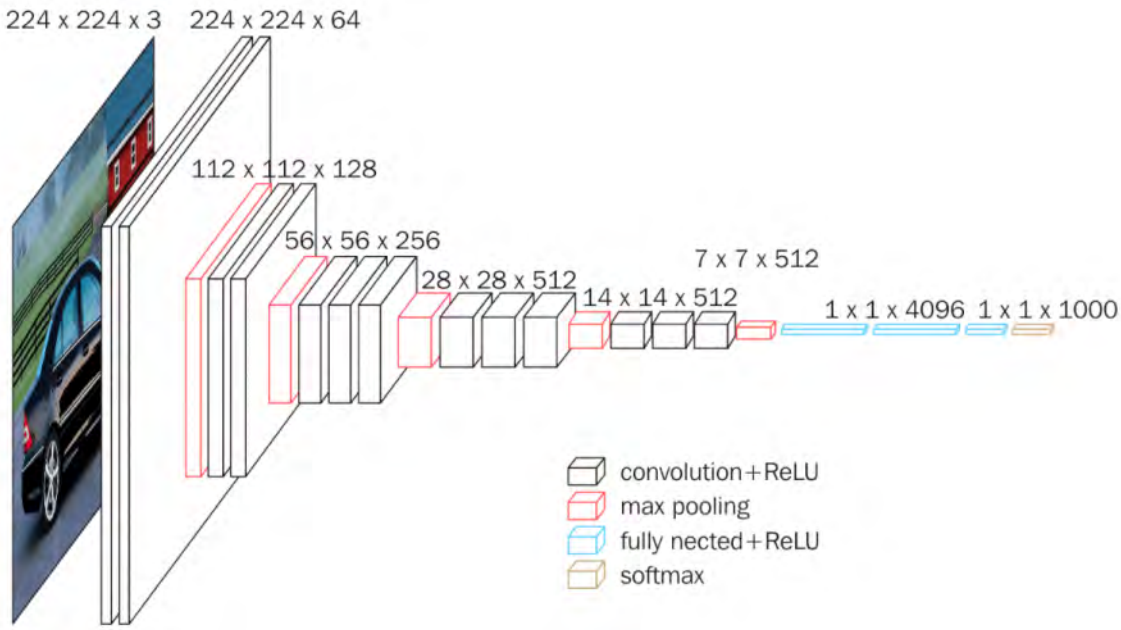

Figure 14: VGG16 Model Architecture

For all of these models, the expanding path of the U-Net shown in **??** is added to the end of the model to map the output to the desired output resolution of 224x224.

3.2.3.1 ResNet50

The ResNet architecture was developed for easier training and optimization through the use of residual functions which reference the layer inputs, rather than arbitrarily adding more layers hoping for higher accuracy. The original design of a 34 layer ResNet is shown in Figure 15 [25]. This thesis utilizes a 50 layer variation of this model entitled the ResNet50.

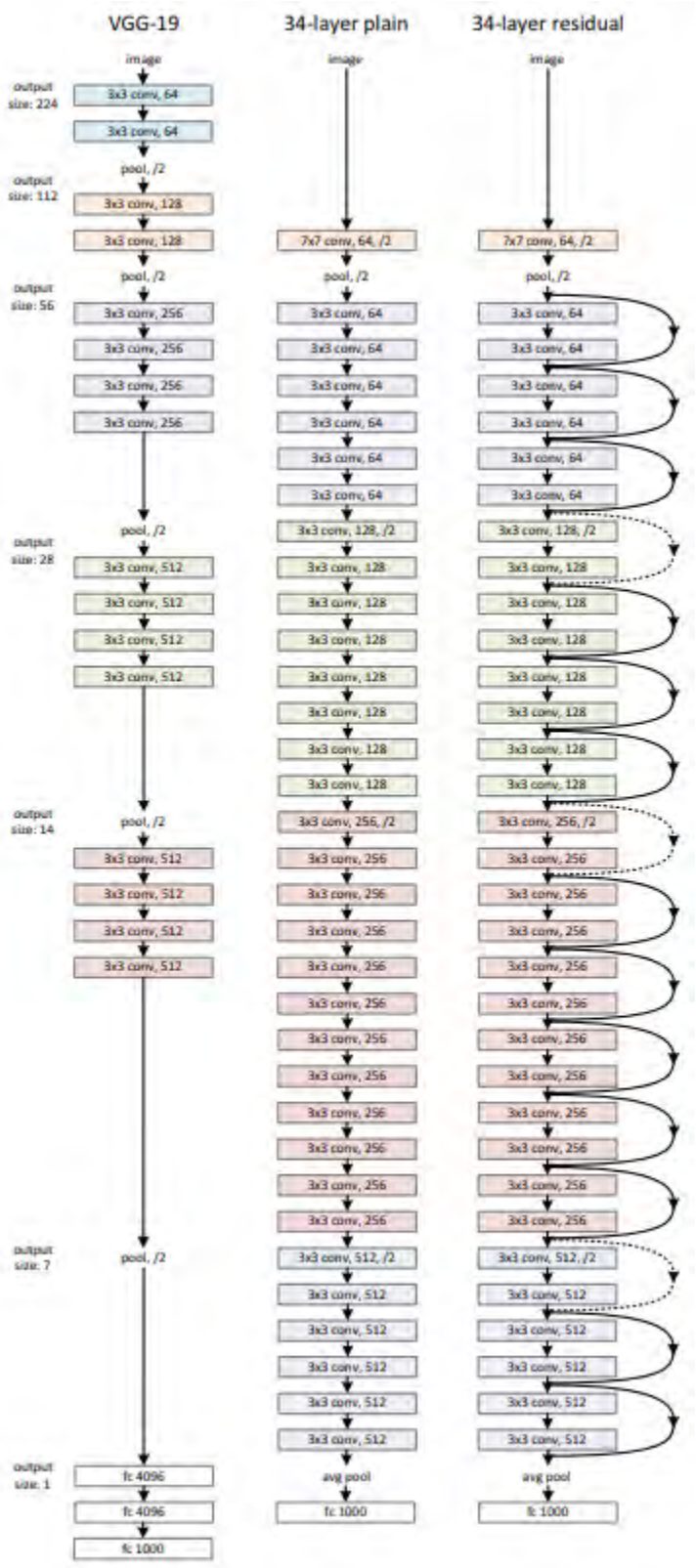

Figure 15: VGG-19 vs. Plain 34 Layer Model vs. ResNet Model Architecture

3.2.4 Evaluation Metrics

The evaluation metrics used during training are binary accuracy and mIoU, since these are the primary metrics used by the authors of the original datasets [8, 21]. The weights of the models with the lowest validation loss and high mIoU scores are used since these models are more likely to perform the best when predicting the final test set and because mIoU is more commonly used for semantic segmentation.

3.2.5 Loss Functions

The loss functions used for these experiments were the mIoU approximation (5) and binary cross-entropy (3). The mIoU approximation, also known as Jaccard loss, was implemented since the evaluation metrics for most of the datasets as well as many others in semantic segmentation use mIoU as a standard. Binary cross-entropy was also used as a comparison for some of the datasets.

3.3 Testing

To properly test the images against their existing truth data using these networks, additional processing is required to interpret the output of the networks and make accurate comparisons. This section will discuss the steps used to process the test set, along with the final metrics used for comparison of the created models and models from previous research.

3.3.1 Test-Set Processing

Because the models cannot be tested on the full high resolution images provided in the datasets, crops of the test images are used. These crops are then used to created the final high resolution prediction mask using stitches of the smaller output prediction masks. The size of these test crops can vary between the original size of

224x224 and 112x112 based on previous research indicating that testing on smaller crops may result in increased segmentation performance [13, 3]. Because the test images are not evenly divisible by these resolutions, an algorithm was developed to create evenly distributed overlapping patches. This was done to prevent cropping out large portions of the test image. The final patches of test data are taken by using (Algorithm 1) throughout the image.

Algorithm 1 Test Image Cropping

```
x = Image Width
y = Image Height
L = CropLength
xBottomBoundary = x/2 - floor((x/2)/L)
xTopBoundary = x - xBottomBoundary
yBottomBoundary = y/2 - floor((y/2)/L)
yTopBoundary = y - yBottomBoundary
CroppedImage = []
Counter = 0
for xIndex in (xBottomBoundary, xTopBoundary, L) do
    if xIndex + croplength < x
        CroppedImage[Counter] = Image[xIndex:xIndex+L]
        Counter++
    for yIndex in (yBottomBoundary, yTopBoundary, L) do
        if yIndex + croplength ¡ y
            CroppedImage[Counter] = Image[yIndex:yIndex+L]
            Counter++
return CroppedImage
```

By cropping the test images and masks using this algorithm, the entire image is utilized and the overlap is distributed throughout the edges of the image. Having this distributed overlap helps the final prediction since FCNs are known to vary their predictions on edges of the imagery. The prediction values of all of the crops are added together into a separate array and averaged to determine the final prediction mask for a given image. These values are then used against a threshold of 0.5 in order to create a definitive binary mask, which is then evaluated using the various evaluation metrics.

3.3.2 Metrics

The final evaluation metrics for comparison are based upon the metrics used in the original research papers for each dataset [8, 20, 21]. The INRIA dataset uses IoU and accuracy for each specified city as their measure of performance. The Massachusetts datasets use a lesser known metric called the precision-recall breakeven point. This is the point on the precision-recall curve for the model in which precision and recall are the same. The calculation of these values are shown in below where TP, FP, and FN are true positives, false positives, and false negatives respectively.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

Finally, the RoadNet dataset uses accuracy and IoU as well as precision and recall. The final U-Net model along with the other pre-trained models from Keras will be measured using the respective metrics for each of the four datasets, and with IoU and accuracy for the final combined U-Net Buildings dataset.

The four datasets tested for this research consist of the following: INRIA Aerial Imagery, Massachusetts Buildings and Roads, and RoadNet. Along with the four individual datasets, a final combined dataset is also created and tested using a combination of the INRIA Aerial Imagery and Massachusetts Buildings datasets. The baseline U-Net was created using most of the original architecture found in [3], along with the Jaccard loss function, Adam optimizer with Nestorov momentum, and He uniform initializer. Two other models with two cascaded U-Nets and three cascaded U-Nets were also created to test the findings in [17] that stacking U-Nets can improve model performance. Other previously established architectures known as the ResNet

and the VGGNet are also implemented for further comparison. Lastly, the final test sets were processed using Algorithm 1 in order to properly evaluate the models based on the training images. The results of these implementations are found in Chapter 4.

IV. Results and Analysis

This chapter discusses the overall results of each model and the ability of each to perform semantic segmentation the four datasets along with the final combined buildings dataset. The metrics used for this measurement will vary based on the ones used by the original authors, but will primarily be based on IoU and accuracy [8, 21, 20].

The sections in this chapter will cover the training and testing results of all five created models, and the following subsections will discuss the performance of the models on each of the five datasets for comparison. Along with the numerical results of each model, examples of the segmentation outputs for a common image will also be shown for visualization purposes.

All models were trained for 1000 epochs with 500 batches of 32 images per epoch. Based on the model architectures and the number of datasets used for this thesis, a large number of models and variations were trained. This required considerable computational resources and time, which meant some models were stopped early if training showed little to no improvement.

4.1 Training

The following sections discuss the training performance of each proposed model using accuracy, mean intersection over union (mIoU) and loss curves for each model and each dataset. Because the training curves of all three levels of the U-Net and the ResNet50 models were fairly similar, the graphs for the 2-Level and 3-Level U-Nets and the ResNet50 models are included in Appendix A.

4.1.1 INRIA Dataset

The training curves for each of the five models for the INRIA Buildings dataset are shown in Figure 16 - Figure 17

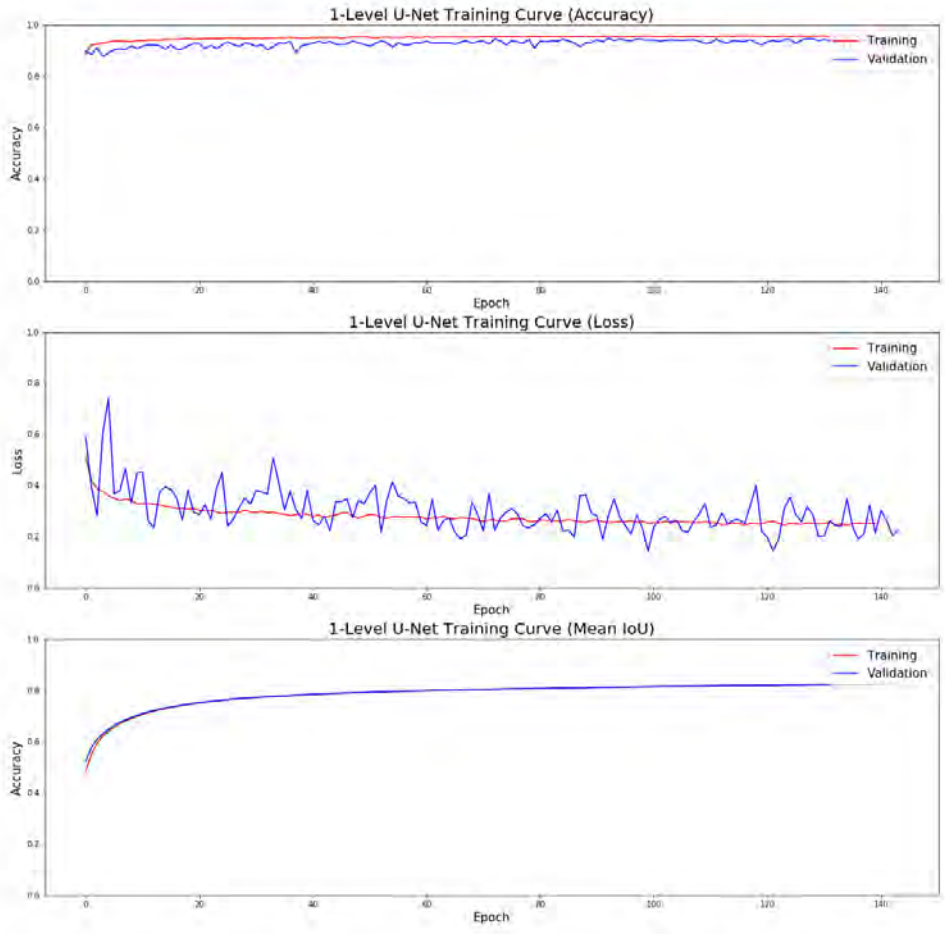

Figure 16: The 1-Level U-Net INRIA training curves. The validation loss follows the training loss despite the noise, indicating that the model is able to learn the dataset. The mIoU curve reinforces this by increasing as the loss curves decrease.

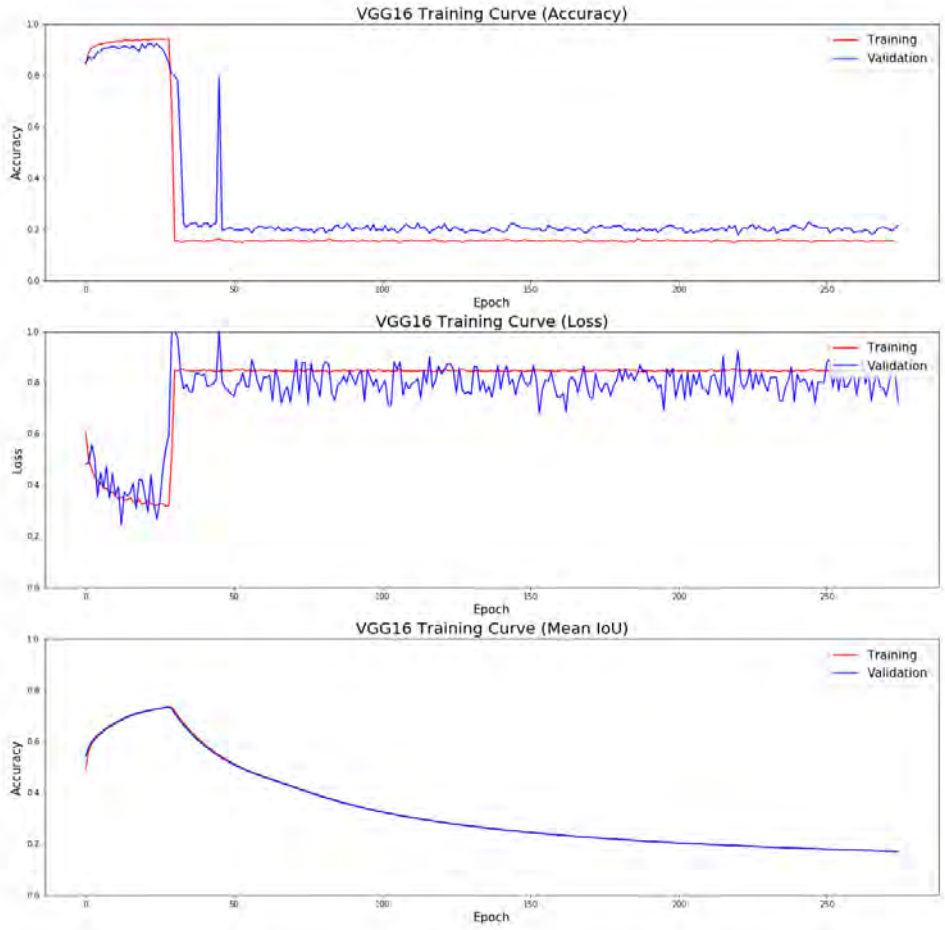

Figure 17: The VGG16 INRIA training curves. All three curves exhibit a drastic decrease in performance approximately at epoch 40.

The most important characteristic that was analyzed during training for all datasets was the validation loss over each epoch. The validation loss curves roughly follow the training loss, indicating that the models are able to learn the dataset. This is also shown by the increase in mIoU as the loss curves decrease. This is also a common occurrence in the training curves presented in the other 4 datasets. It is also impor-

tant to note that the biggest difference between these models shown is in the VGG16 architecture. After approximately 40 epochs the model immediately drops off with a drastic decrease in accuracy and mIoU, and a large increase in loss. Reducing the learning rate at this point may have been able to improve the final performance as the epochs increased. Overall it seems that most of the models except for the VGG16 had a decent capacity to learn the INRIA dataset.

4.1.2 Massachusetts Buildings Dataset

The training curves for the models used for the Massachusetts Buildings dataset are shown in Figure 18 - Figure 19. The 2-level U-Net, 3-level U-Net, and ResNet50 training curves can be found in Appendix A. Much like the INRIA dataset, the validation loss curves roughly follow the training loss. Most of the models appear to have the capability of learning the dataset, except for the VGG16 model, which drastically decreased in performance again around epoch 60.

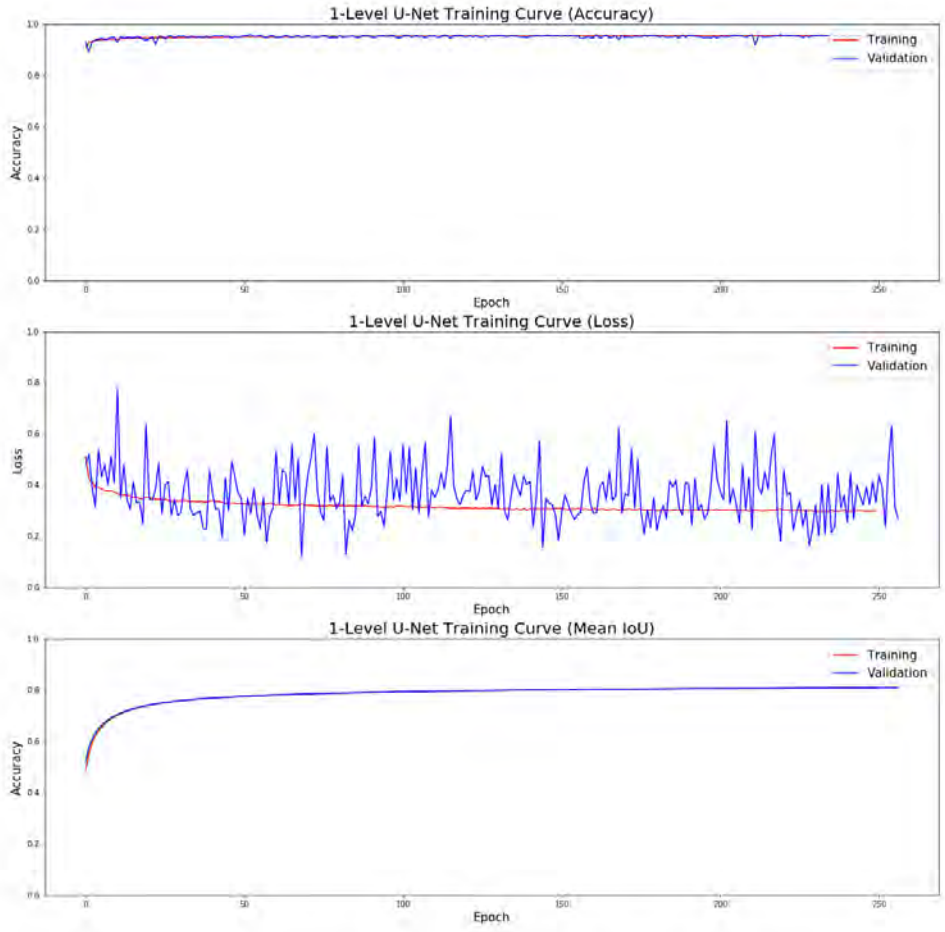

Figure 18: The 1-Level U-Net Massachusetts Buildings training curves. The variance of the validation loss curve against the training curve indicates that the validation set is likely poor in assessing generalization of the model.

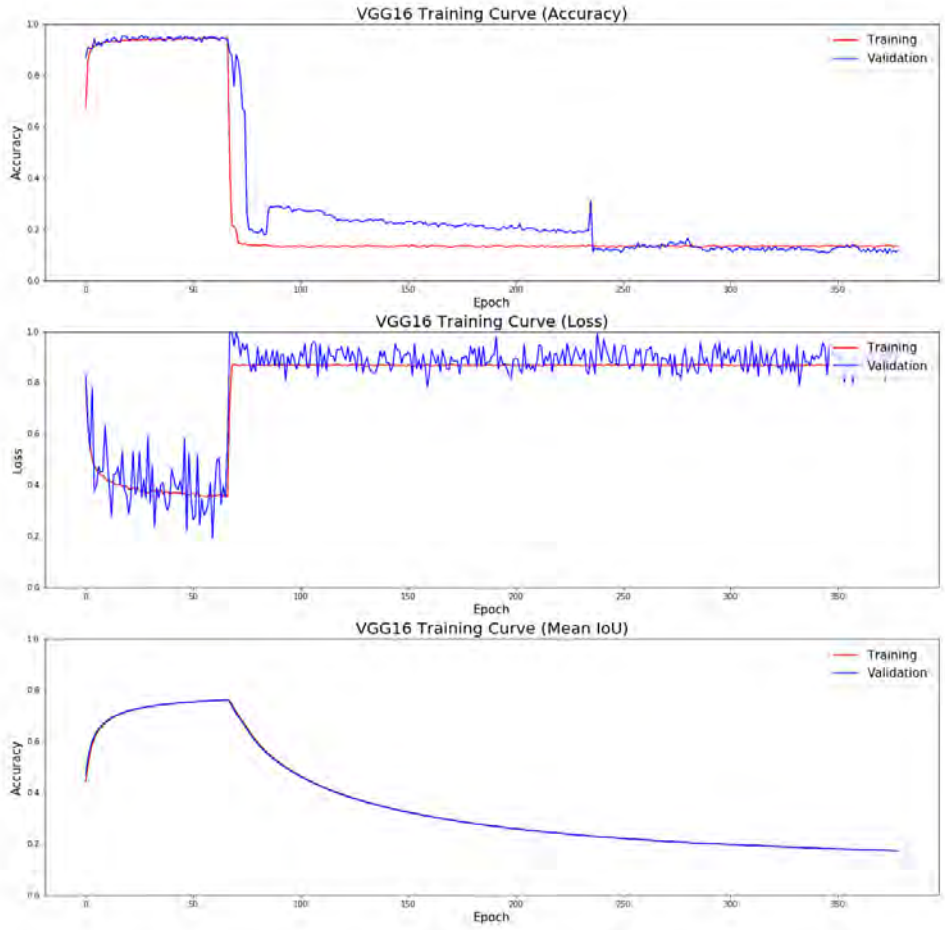

Figure 19: The VGG16 Massachusetts Buildings training curves. The VGG16 model begins to fail after approximately 60 epochs.

4.1.3 Massachusetts Roads Dataset

The training curves for the models used for the Massachusetts Roads dataset are shown in Figure 20 - Figure 21. The 2-level U-Net, 3-level U-Net, and ResNet50 training curves can be found in Appendix A.

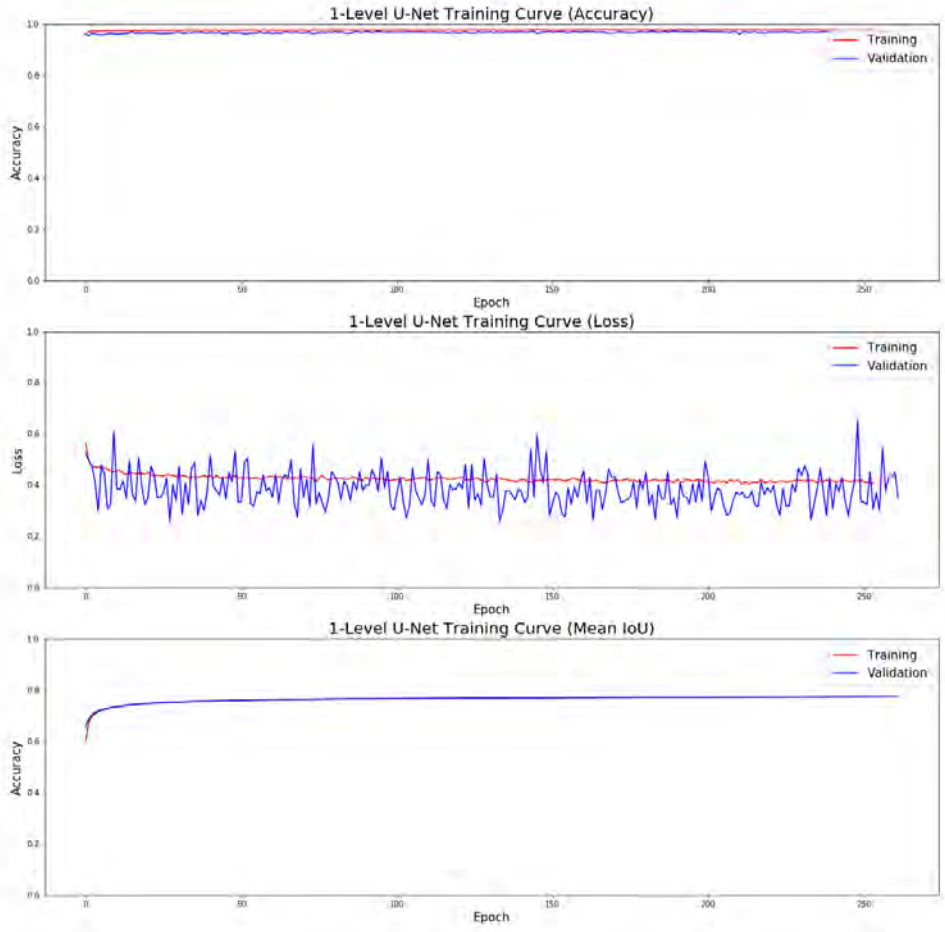

Figure 20: The 1-Level U-Net Massachusetts Roads training curves. The loss curves may indicate that the validation data may be poor at assessing model performance.

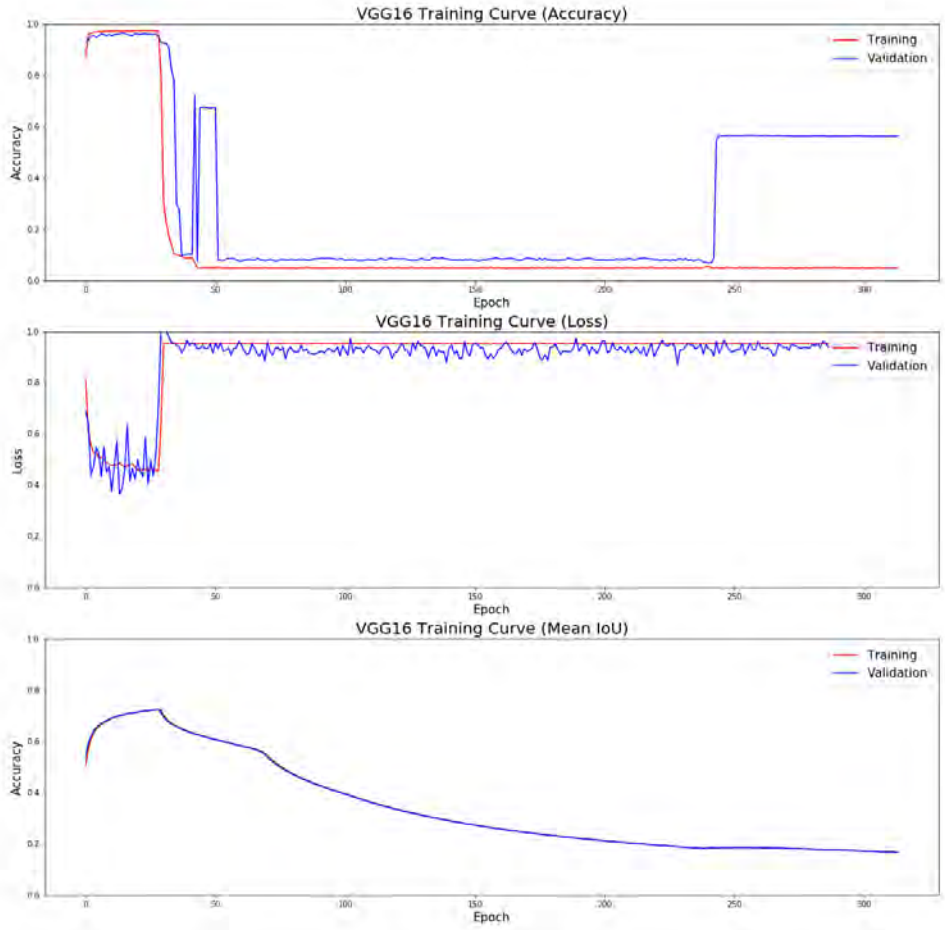

Figure 21: The VGG16 Massachusetts Roads training curves. The VGG16 model begins to fail after approximately 40 epochs.

The validation loss curves shown for the Massachusetts Roads dataset again follows roughly follows the training curve and shows significant noise. However, the validation loss is also consistently below the training loss. The VGG16 model again begins to fail after approximately 40 epochs.

4.1.3.1 RoadNet

The training curves for the models used for the RoadNet dataset are shown in Figure 22 - Figure 23. The 2-level U-Net, 3-level U-Net, and ResNet50 training curves can be found in Appendix A. The loss curves overall for the RoadNet dataset appear to be less noisy and have a better fit with most models. All of the models appear to train fairly well to the data based on the decreasing loss scores except for VGG16, which again falters around epoch 40.

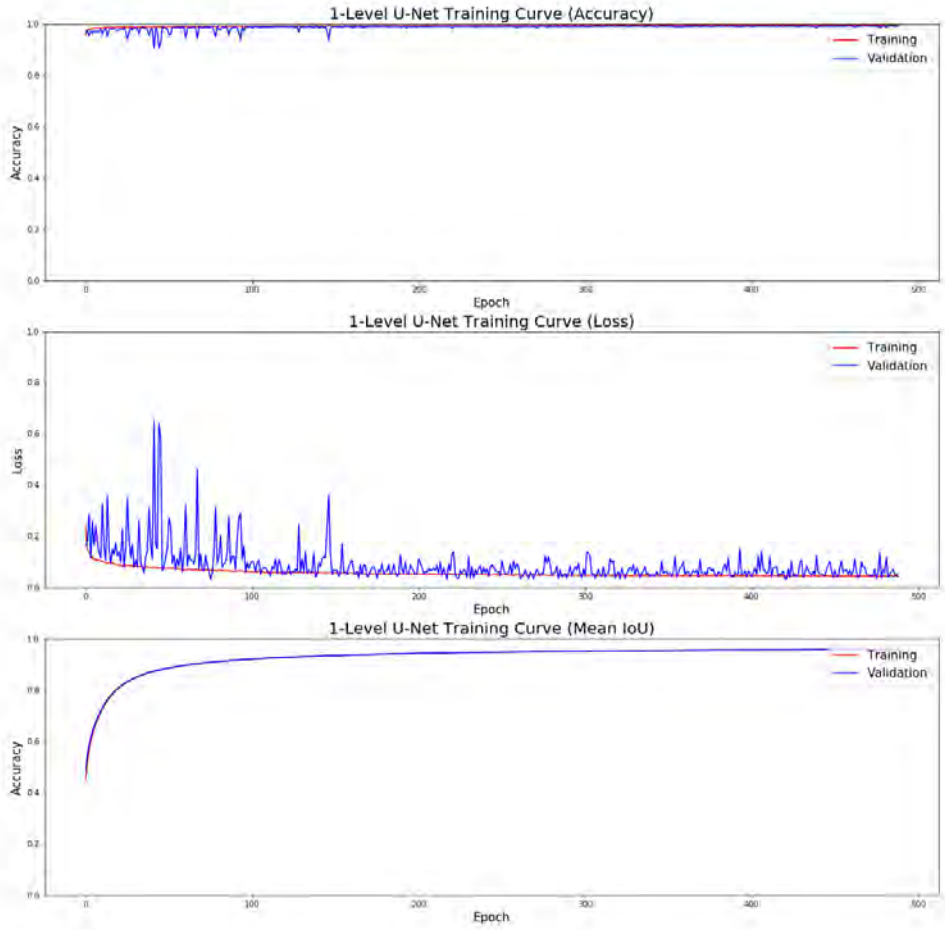

Figure 22: The 1-Level U-Net RoadNet training curves. The validation loss curves for this dataset exhibit less noise compared to those found in other datasets.

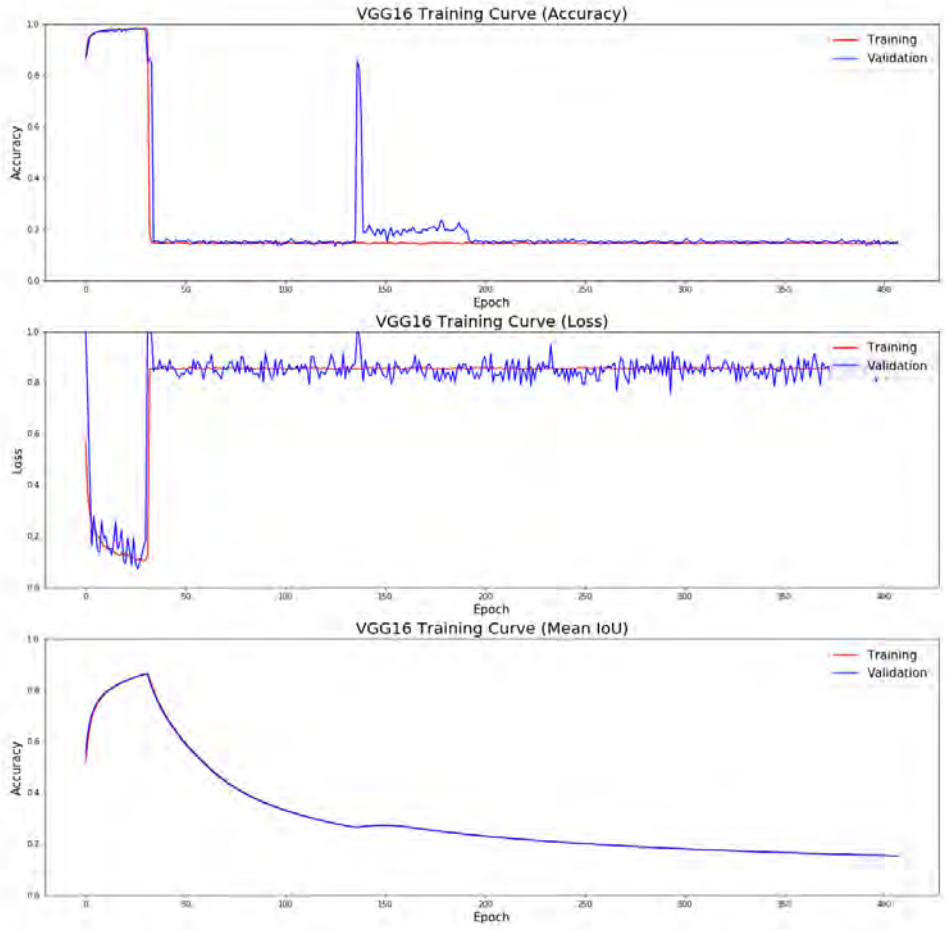

Figure 23: The VGG16 RoadNet training curves. The VGG16 model begins to fail after approximately 40 epochs.

4.1.3.2 Combined Buildings

The final training curves for the models used for the newly created buildings dataset are shown in Figure 24 - Figure 25. The 2-level U-Net, 3-level U-Net, and ResNet50 training curves can be found in Appendix A.

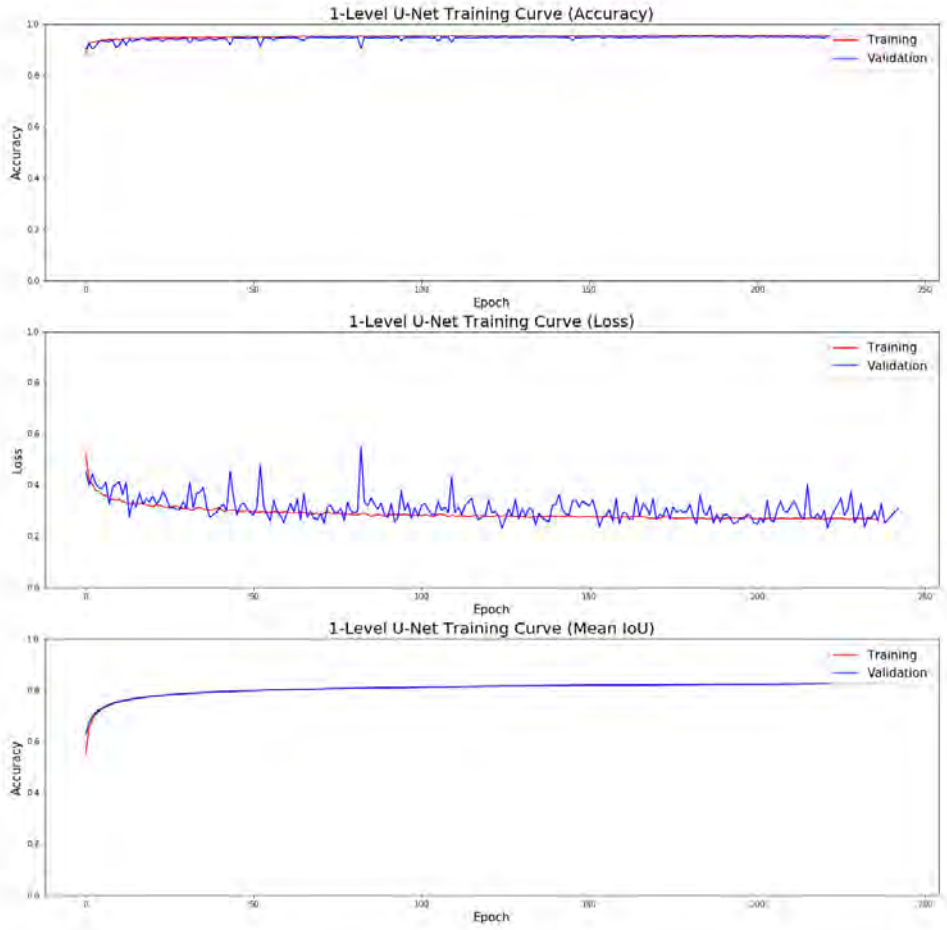

Figure 24: The 1-Level U-Net Buildings training curves. The validation loss curve is not nearly as noisy as those found in the Massachusetts Buildings dataset.

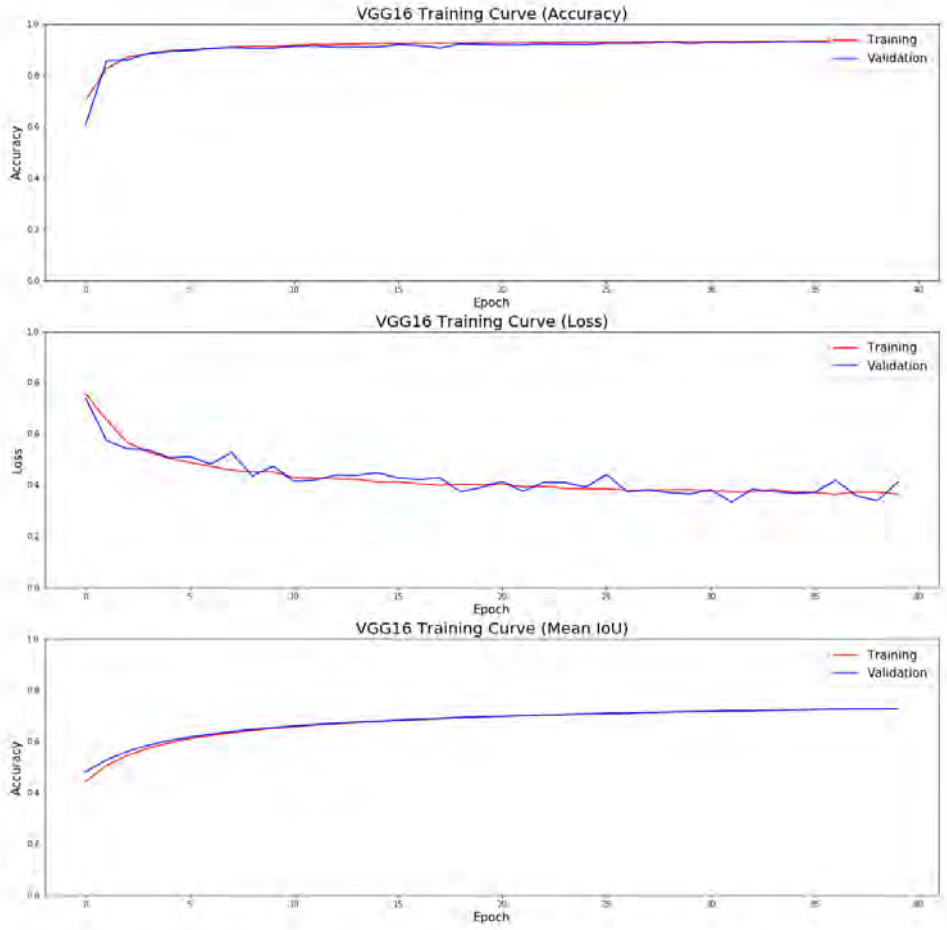

Figure 25: The VGG16 Buildings training curves. This model was not able to be tested extensively, but based on performance on other datasets, the model would have likely failed with more epochs.

The loss curves after combining the INRIA and Massachusetts Buildings datasets are similar to those found in the original INRIA dataset versus the larger discrepancies in the original Massachusetts dataset. The VGG16 model was unable to be tested extensively for this dataset due to time limitations, but would likely have drastically

decreased in performance after a few more epochs due to its inability to learn the previous four datasets.

Overall based on the training curves presented, most of the models have an adequate capacity to learn the various datasets presented. The final results and comparisons are determined using the final predefined test sets to obtain the highest performing model.

4.2 Test

The output prediction masks of the tested models are compared alongside the results of other research to determine the efficacy of the proposed architectures. The prediction masks of the proposed architectures are also plotted alongside the truth data to visualize the results of the segmentations. The test data used for the Massachusetts Buildings, Massachusetts Roads, and RoadNet datasets were predefined by the authors [20, 21]. The validation set defined by the author was used as the test set for the INRIA Buildings dataset [8]. The final test data for the combined buildings dataset was created by combining the test data from the Massachusetts Buildings dataset and the validation data from the INRIA Buildings dataset. [8, 20]. The final mIoU calculation was found using Equation (2) with the output prediction mask and truth data.

4.2.1 INRIA Dataset

Because the INRIA Dataset uses binary accuracy and mIoU as metrics [8], the predictions of the models will be calculated in the same manner. The results of the predictions for each model and other existing models are shown in Table 1

None of the proposed model architectures were able to achieve a higher mIoU than any presented by Khalel et al [17]. The 1-Level U-Net, however, was able

to outperform all of the models introduced by the authors of the INRIA dataset [8]. The results from this particular dataset show a decrease in performance with the addition of U-Net levels, contrary to what was found in previous research [17]. This may be attributed to a drastic increase in complexity in the model for a less complex dataset. The ResNet50 is also shown to be drastically worse in producing accurate prediction masks against the other models. The VGG16 model with the lowest validation loss was unable to output valid values when predicting on the INRIA test set. The prediction masks for each model along with the truth data for an example image is shown in Figure 27.

Method	Metric	Austin	Chicago	Kitsap Co.	West Tyrol	Vienna	Overall
FCN [8]	mIoU	47.66	53.62	33.70	46.86	60.60	53.82
	Acc.	92.22	88.59	98.58	95.83	88.72	92.79
Skip [8]	mIoU	57.87	61.13	46.43	54.91	70.51	62.97
	Acc.	93.85	90.54	98.84	96.47	91.48	94.24
MLP [8]	mIoU	61.20	61.30	51.50	57.95	72.13	64.67
	Acc.	94.20	90.43	98.92	96.66	91.87	94.42
SegNet (Single-Loss) [17]	mIoU	76.49	66.77	72.69	66.35	76.25	72.57
	Acc.	93.12	99.24	97.79	91.58	96.55	95.66
SegNet + MultiTask-Loss [17]	mIoU	76.76	67.06	**73.30**	66.91	76.68	73.00
	Acc.	93.21	**99.25**	97.84	91.71	**96.61**	95.73
2-Levels U-Net + Aug. [17]	mIoU	**77.29**	**68.52**	72.84	**75.38**	**78.72**	**74.55**
	Acc.	**96.69**	92.40	**99.25**	**98.11**	93.79	**96.05**
Resnet50	mIoU	41.68	54.03	44.00	50.88	55.82	51.24
	Acc.	85.52	85.92	98.57	95.76	83.27	89.81
1-Level U-Net	mIoU	68.91	64.83	51.57	65.90	75.17	69.10
	Acc.	95.42	91.09	98.88	97.16	92.70	95.05
2-Level U-Net	mIoU	68.14	61.89	52.10	62.71	71.37	66.25
	Acc.	95.15	90.02	98.86	96.66	91.55	94.44
3-Level U-Net	mIoU	64.35	60.23	50.21	61.16	71.94	64.99
	Acc.	94.37	89.09	98.73	96.31	91.34	93.97

Table 1: INRIA Dataset Results

The segmentations provided by each of the U-Nets appears to perform well in densely crowded areas of the image. However, in large areas without buildings the networks seem to exhibit slight edge artifacts, despite testing on smaller images, which was previously found to help prevent such effects [13]. The U-Nets also seem to struggle with predicting perfect edges as well, since the building predictions are almost

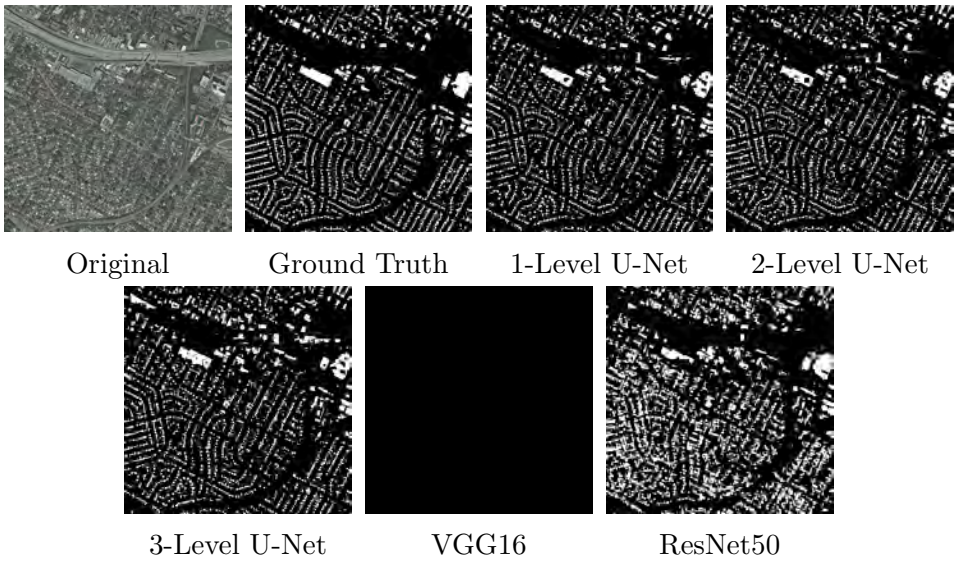

Figure 27: Output Prediction Masks for INRIA Dataset

never perfectly rectangular. The ResNet50 predictions suffered from the boundary effects found in patch predictions, and was more likely to use a blanket prediction for larger areas of the image rather than precisely segmenting the more crowded areas of the image. The VGG16 network with the lowest validation loss was unable to produce any meaningful outputs in its prediction process.

4.2.2 Massachusetts Dataset (Buildings)

The Massachusetts dataset results are calculated using the precision-recall breakeven point [20]. The results of the predictions for each model and other existing models is shown in Table 2

Mean intersection over union and accuracy metrics were added to the Massachusetts datasets for future methods of comparison and to also compare between the created models. The models created for this thesis were unable to achieve a higher precision-recall breakeven point score against the 2-Level U-Net with data augmentation and all other tested methods, including the original model presented by Mnih et al [20]. The results for this dataset also slightly contradict those found in the INRIA dataset,

Method	Precision-recall breakeven point	mIoU	Accuracy
Mnih et al. [20]	0.9211	-	-
Saito et al. [26]	0.9230	-	-
Marcu et al. [7]	0.9423	-	-
2-Level U-Net + Aug [17]	**0.9633**	-	-
Resnet	0.5542	39.40	82.38
1-Level U-Net	0.8355	71.75	93.90
2-Level U-Net	0.8328	71.36	93.76
3-Level U-Net	0.8479	**73.60**	**94.32**

Table 2: Massachusetts Buildings Dataset Results

since mIoU and accuracy now appear to increase with the addition of two U-Net stages. The ResNet50 is consistent against the INRIA Buildings dataset with a much lower mIoU and accuracy compared to the U-Nets. The VGG16 model was again unable to produce meaningful output predictions for this dataset. The prediction masks for each model along with the truth data for an example image is shown in Figure 28.

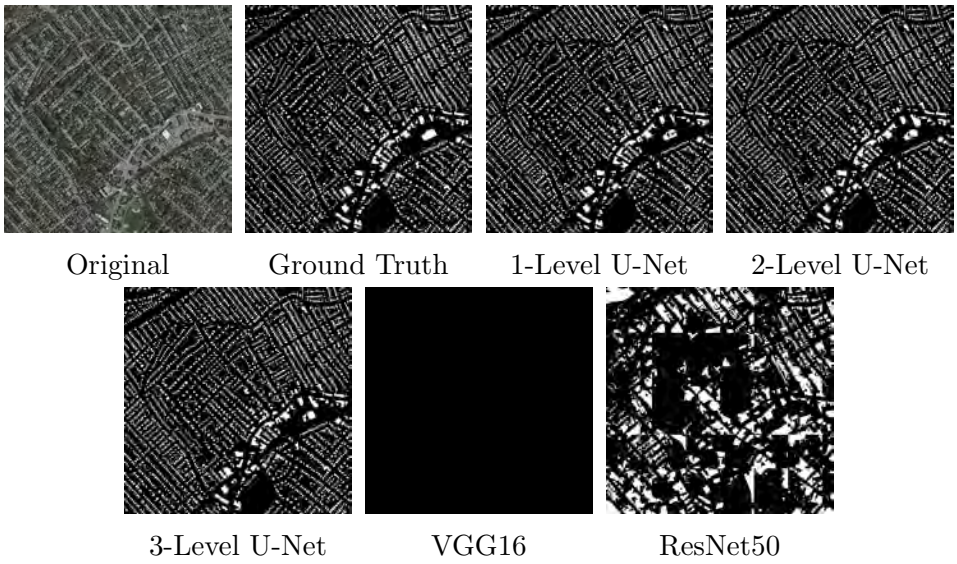

Figure 28: Output Prediction Masks for Massachusetts Buildings Dataset

The output masks appear to follow the same general pattern found in the INRIA masks. Though both are not particularly excellent at detecting edges, the 3-Level

U-Net appears to have an overall sharper prediction mask for each building than the lower level ones. The slight edge artifacts, however, are still present in all models in areas with a lower density of buildings. The ResNet50 model again struggles the most with edge artifacts and exhibits poor precision in each prediction patch.

4.2.3 Massachusetts Dataset (Roads)

The results of the predictions for each model and other existing models are shown in Table 3. Using the precision-recall breakeven metric, the final post-processing network introduced by Mnih et al. performed significantly better than those proposed in this thesis. Other methods using this dataset, however, utilized the traditional metrics of mIoU and accuracy [27]. All of the U-Nets trained for this thesis were able to provide a higher mIoU than the U-Net model presented in [27], however, the U-Nets for this thesis were greatly outperformed by the alternative methods found in [27]. The 2-Level U-Net was able to achieve the highest overall accuracy, but for evaluating the performance of useful predictions this metric means much less, since a higher accuracy can simply mean more "not road" predictions.

Method	Precision-recall breakeven point	mIoU	Accuracy
Neural net [20]	0.8873	-	-
CRF [20]	0.8904	-	-
Post-processing net [20]	**0.9006**	-	-
U-Net [27]	-	59.85	95.66
LinkNet [27]	-	67.92	96.78
DeepLabv3+ [27]	-	66.72	96.67
D-LinkNet [27]	-	67.03	96.75
ResNet34 + ASPP + LinkNet [27]	-	**68.78**	96.72
VGG16	0.3291	5.19	95.39
Resnet	0.3505	8.12	95.45
1-Level U-Net	0.8354	61.73	97.81
2-Level U-Net	0.7663	62.15	**97.83**
3-Level U-Net	0.7614	61.57	97.80

Table 3: Massachusetts Roads Dataset Results

The prediction mask for each model along with the truth data for an example image is shown in Figure 29. Because the number of road pixels in these images are much smaller than background pixels, it is slightly more difficult to determine patterns in the output predictions. The U-Nets again appear to have slight edge artifacts based on the lack of continuous lines between prediction patches for each detected road. The VGG16 network was able to produce valid outputs for this dataset, but was only able to mostly predict every pixel as background to achieve a high accuracy. The ResNet50 also performed poorly, likely due to the artifact effect found in the previous dataset predictions. The results of this dataset also highlight the important of each stage of the network; while the ResNet50 on its own performed poorly, the smaller ResNet34 combined with alternative encoders and decoders was able to achieve state-of-the-art results [27].

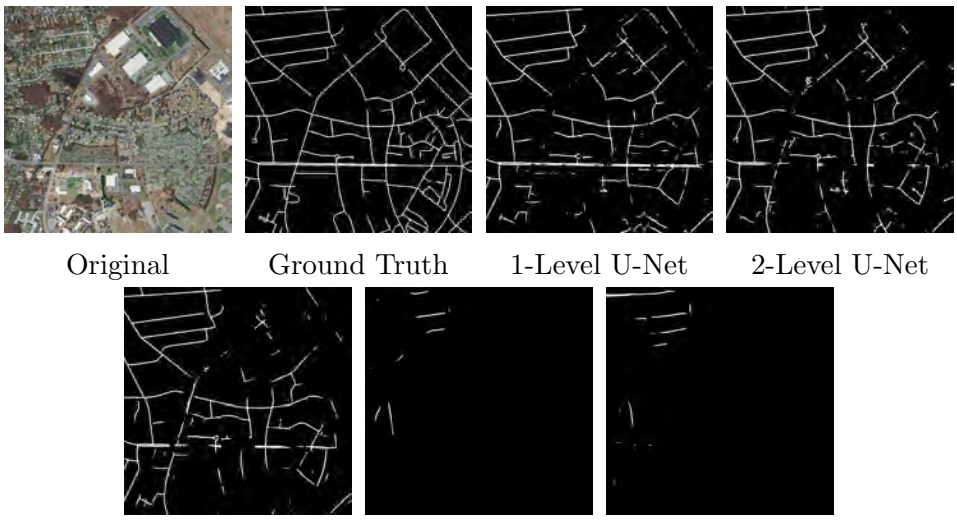

Figure 29: Output Prediction Masks for Massachusetts Roads Dataset

4.2.4 RoadNet Dataset

The RoadNet results were compared using mIoU and accuracy [21]. The results of the predictions for each model and other existing models are shown in Table 4.

All architectures introduced with the RoadNet dataset by Liu et al. perform significantly better in segmenting roads than any of the proposed U-Net architectures. The difference in performance between the U-Nets created for this thesis and the U-Net created by Liu et al. for road segmentation is likely due to the different preprocessing methods used for inputting images into the network, dividing the validation set from the training set, and the implementation of the different loss functions [21]. The prediction masks for each model along with the truth data for an example image is shown in Figure 30. The performance of the U-Nets in this dataset is greatly improved over the Massachusetts Roads dataset, which is shown by the detection of most of the roads found in the ground truth data. However, the models still lack the ability to consistently label continuous roads between image patches. The ResNet50 model rarely predicted a positive class, and those that were predicted still showed signs of edge artifacts.

Method	mIoU	Accuracy
FCN8s [21]	91.9	98.1
FCN8s+ [21]	91.6	98.0
SegNet [21]	85.3	96.3
SegNet+ [21]	87.7	97.0
UNet [21]	88.7	97.3
UNet+ [21]	91.1	97.9
CasNet [21]	91.1	97.9
CasNet+ [21]	92.1	**98.2**
RoadNet+ [21]	**92.4**	**98.2**
ResNet50	0.8	87.5
1-Level U-Net	81.5	97.5
2-Level U-Net	82.2	97.5
3-Level U-Net	83.9	97.8

Table 4: RoadNet Dataset Results (+ refers to the inclusion of the loss function by Liu et al.)

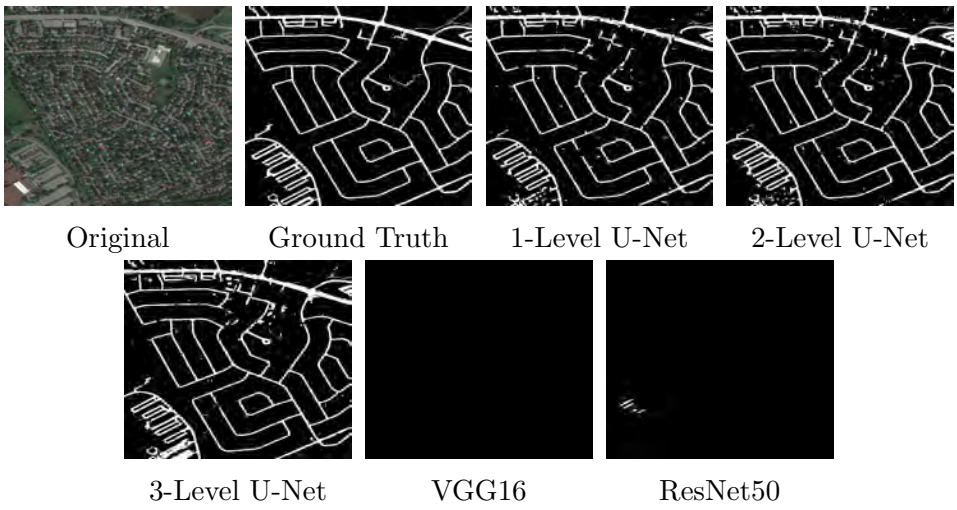

Figure 30: Output Prediction Masks for RoadNet Dataset

4.2.5 Combined Buildings Dataset

The final results for the newly created Buildings dataset utilize the conventional metrics of mIoU, shown in Table 5. The result of combining the two buildings datasets was surprisingly a much lower mIoU and accuracy score than either of the two datasets individually. Even with the scaling of the INRIA images to match the spatial resolution of the Massachusetts Buildings dataset, none of the models seem to have the ability to learn location invariance. This is most likely due to the lack of data that have the same spatial resolution. By having a sufficient number of aerial images of the same spatial resolution, potential artifacts detected by the models from preprocessing can likely be avoided. This can be seen by comparing the output prediction results of the Massachusetts Buildings and INRIA portions of the dataset, which are shown in Figure 31 and Figure 32. Both figures use the same original image from Figure 28 and Figure 27 for comparison.

The U-Net models for this dataset have similar segmentation masks found in the original Massachusetts Buildings dataset. The biggest difference found is that with the addition of the INRIA dataset, the ResNet50 and VGG16 models were finally

Methods	mIoU	Accuracy
VGG16	**50.33**	88.91
ResNet50	35.03	86.80
1-Level U-Net	37.20	81.37
2-Level U-Net	46.09	87.65
3-Level U-Net	46.21	**89.76**

Table 5: Combined Buildings Dataset Results

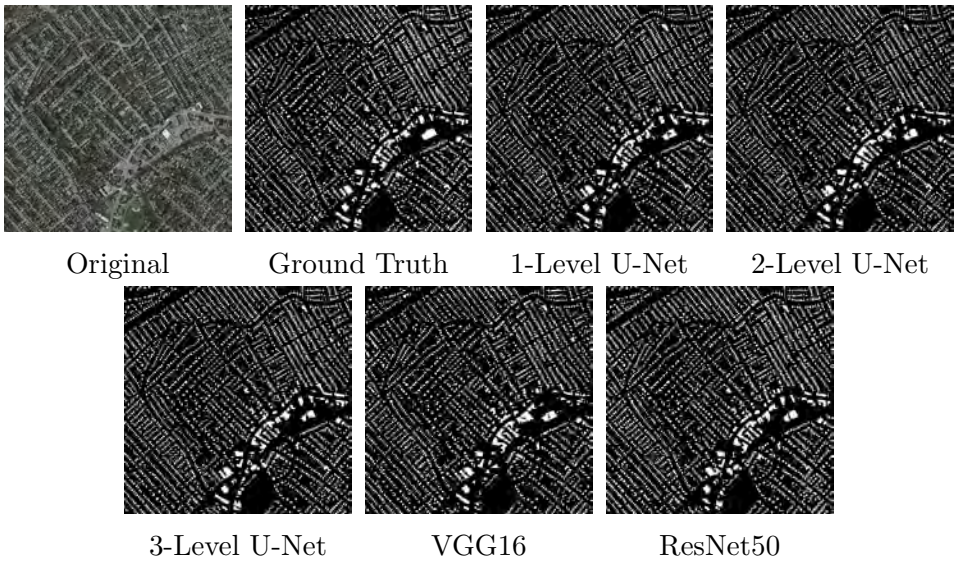

Figure 31: Output prediction masks for the final buildings dataset for an image originally from the Massachusetts Buildings dataset.

able to output decent predictions, with the VGG16 network performing the best. The performance on the INRIA dataset however greatly differs for the U-Nets likely due to the effects of resizing the original image. The outputs of all models resulted in a choppy final image with poor predictions occurring at the edges of the output patches.

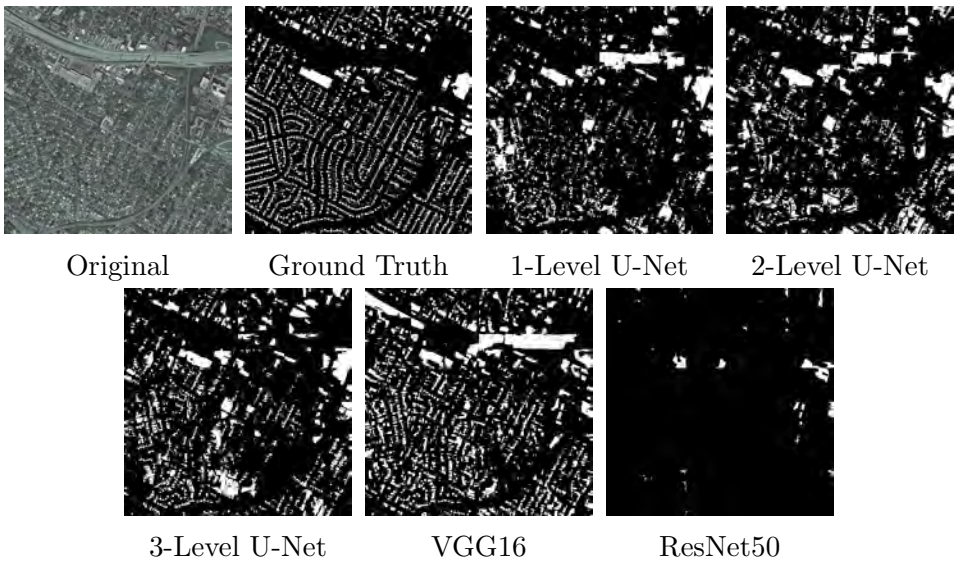

Figure 32: Output prediction masks for the final buildings data for an image originally from the INRIA Buildings dataset. The discrepancy in segmentation quality can likely be attributed to artifacts introduced to the input image after resizing.

4.3 Summary

The results of both building and road datasets varied greatly. Based on these tests there seems to be no definitive evidence that the addition of U-Net levels leads to an increase in segmentation performance. The mIoU values found in Table 4 and Table 5 seem to support this theory with a consistent increase with additional levels of U-Nets, however, Table 1 indicates a consistent decrease. The results in Table 2 show an initial decrease with the 2-level U-Net and an the greatest overall mIoU with the 3-level U-Net. Lastly, the results in Table 3 show an initial increase with the 2-level U-Net and the lowest mIoU value for the 3-level U-Net. Because these variations in performance do not appear to be linked to additional U-Net levels or the datasets used, the value of adding levels is inconclusive.

This testing also shows that U-Nets are not necessarily an optimal or universal solution for segmenting any class from an aerial perspective. Out of all of the models compared, the only model that was able to outperform prior research was the 2-

level U-Net model in Table 3 for the least valuable metric in semantic segmentation, accuracy. Rather than arbitrarily implementing U-Nets or adding U-Net levels for minimal gains in performance, it is likely more effective to develop specific encoders and decoders based on the dataset. In the future a universal network for roads and buildings may be developed and implemented, but with the current lack of labeled data and baselines for comparison, it is difficult to determine if U-Nets are the future for all segmentation.

V. Conclusion

This research aimed to train variations of the U-Net on multiple datasets, as well as combine two publicly available datasets for use in semantic segmentation. This new dataset, however, proved to be inconclusive in allowing networks to learn invariances between the two. Though these networks were unable to achieve state-of-the-art results and perform successful classifications on the combined dataset, the performance of these models can serve as a baseline for others at Air Force Institute of Technology (AFIT) attempting to perform these segmentations.

5.1 Performance

Despite the fact that the proposed architectures were outperformed by existing ones, the output segmentations of the U-Nets performed fairly well for both buildings and roads. While designing model architectures specifically for a given dataset will lead to higher mean intersection over union (mIoU) and accuracy scores, the U-Net has proven to be a useful model to establish a baseline for comparison. Since few models are able to output decent prediction masks for vastly different classes, the U-Net will likely be consistently used in the future as a benchmark for this task.

5.2 Hyperparameters

Due to the large number of models trained for this thesis, the only hyperparameters modified from the original U-Net created were the initializer, optimizer, and loss function. The original models trained on the datasets used the Glorot Uniform initializer and the Adam optimizer based on previous research with this model. These were adjusted based on further research to the He Uniform initializer and Adam optimizer with Nesterov momentum, which improved the overall output predictions [22, 23, 13].

All of the models were also trained using binary cross-entropy as a loss function to determine whether performance is truly enhanced with the Jaccard loss function. The outputs of these models were significantly worse, confirming previous research [14, 19]

5.3 Edge Predictions

Edge predictions are a common concern in performing semantic segmentation for high resolution images [13]. As shown in the results of this study, convolutional neural networks (CNNs) tend to lose accuracy the further from the center the predictions are made, which can result in predictions that look abnormal at the edges. Because of this, some authors utilize a cropping layer to the end of the network [13]. The test images used for this thesis were instead overlapped at the edges of the high resolution images to slightly mitigate this issue. A sliding window method could also be utilized in which predictions are made throughout the image and averaged for a final prediction mask, at the cost of drastically increased training time.

5.4 Future Work

In the future the performance of these models can likely be improved with a larger batch size and smaller image patches. Previous research found that a larger batch size is more important in improving the U-Net model versus a larger receptive field [13]. With either smaller image crops for training or more powerful resources to handle larger batches, the results found in this thesis could be drastically improved. The possibility of implementing generative adversarial networks may also be of value. These networks have shown to be versatile in many applications and may also have potential in semantic segmentation for aerial imagery. Finally, while the use of U-Nets has shown to be a competent benchmark for static images, utilization of these models in real-time applications with the Autonomy and Navigation Technology (ANT)

Center would also be valuable.

Appendix A. Additional Training Curves

1.1 INRIA Dataset

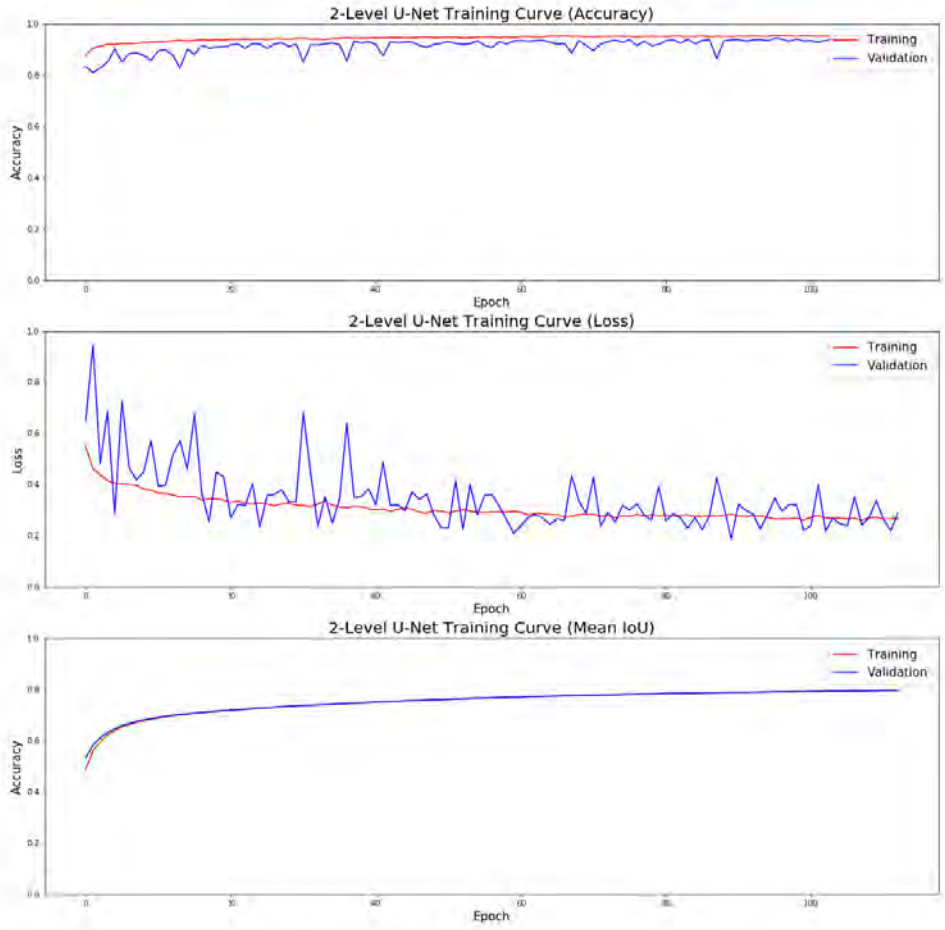

Figure 34: 2-Level U-Net INRIA Training Curves

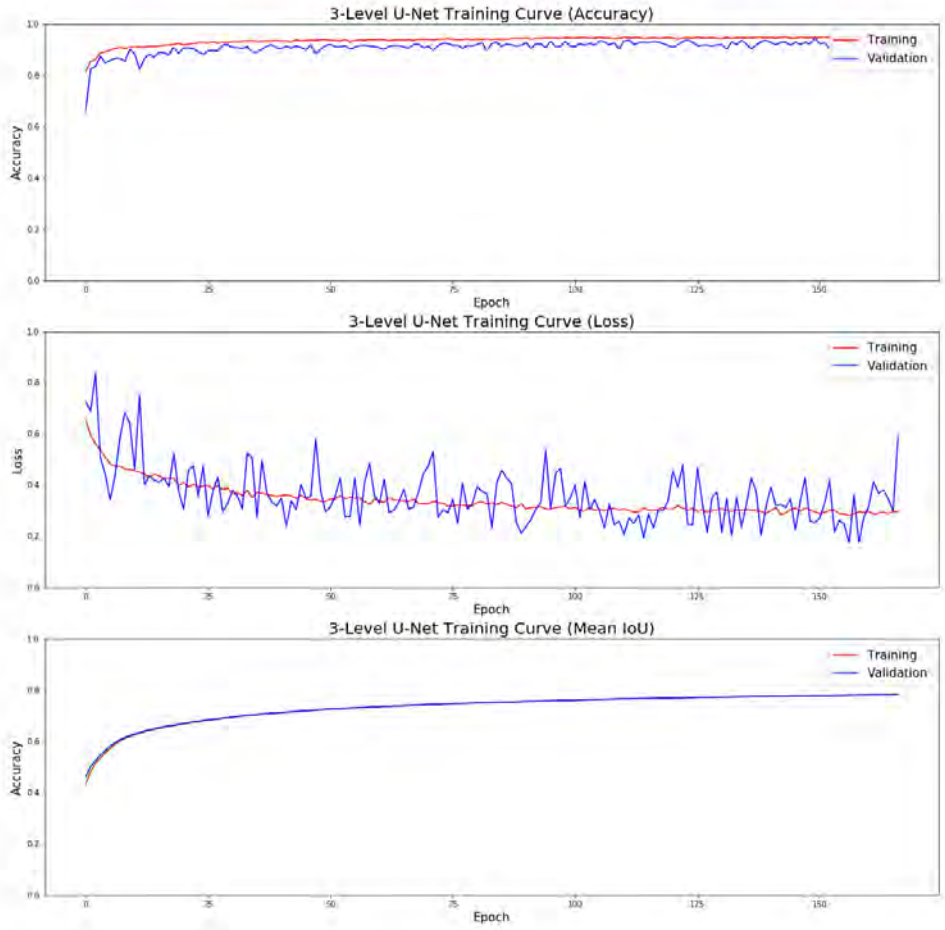

Figure 35: 3-Level U-Net INRIA Training Curves

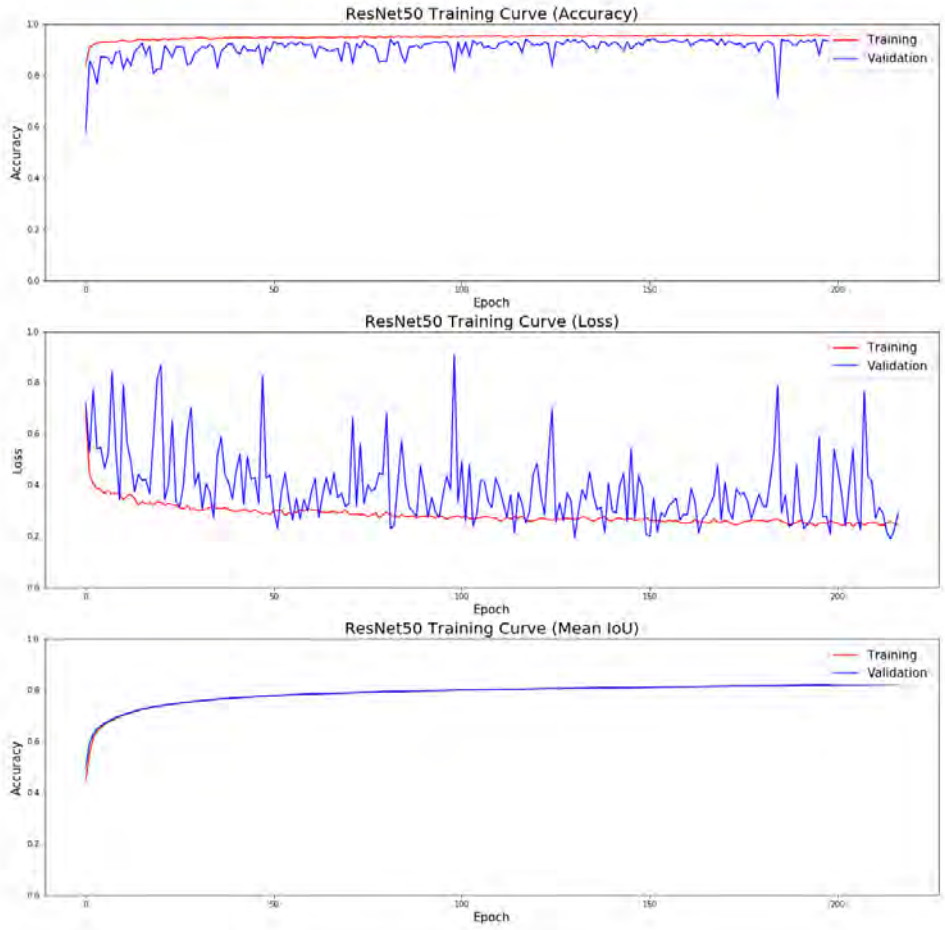

Figure 36: The ResNet50 INRIA training curves. On average, the training loss was below the noisy validation loss, leading to the conclusion that the U-Net may have had a better fit to the data.

1.2 Massachusetts Buildings Dataset

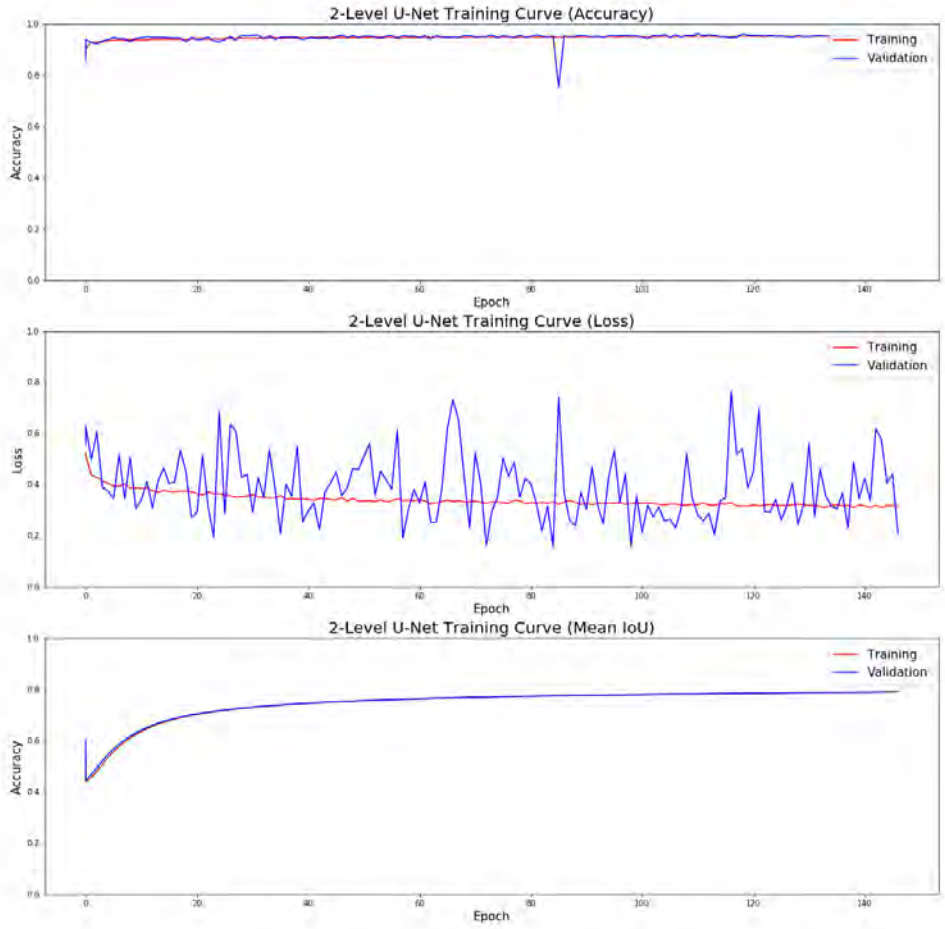

Figure 37: 2-Level U-Net Massachusetts Buildings Training Curves

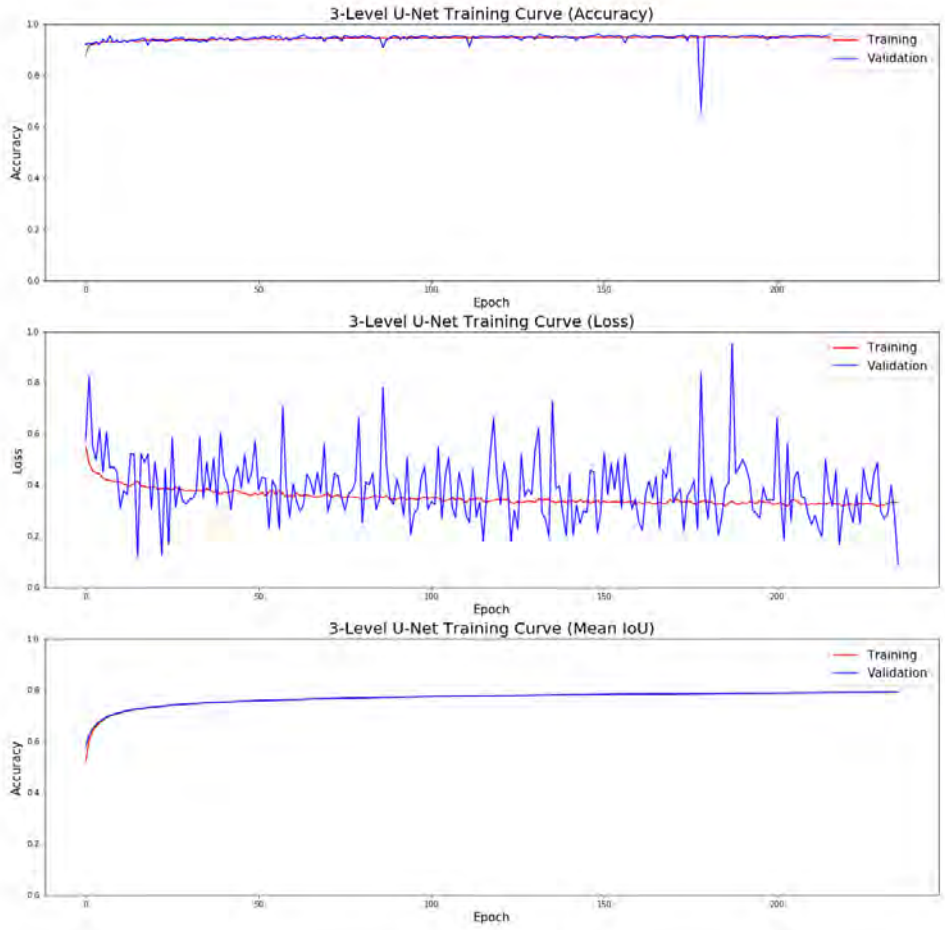

Figure 38: 3-Level U-Net Massachusetts Buildings Training Curves

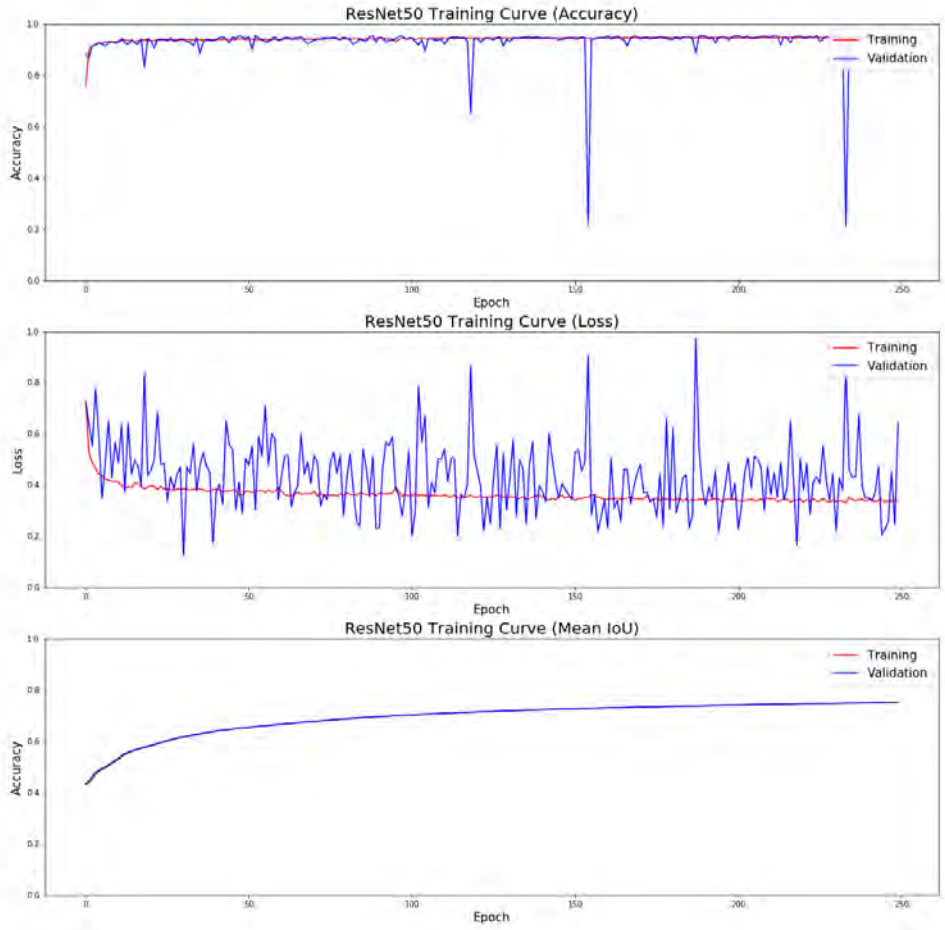

Figure 39: The ResNet50 Massachusetts Buildings training curves. The curves exhibit generally the same pattern as the U-Net models, but with larger variance in the validation loss curve against the training loss curve.

1.3 Massachusetts Roads Dataset

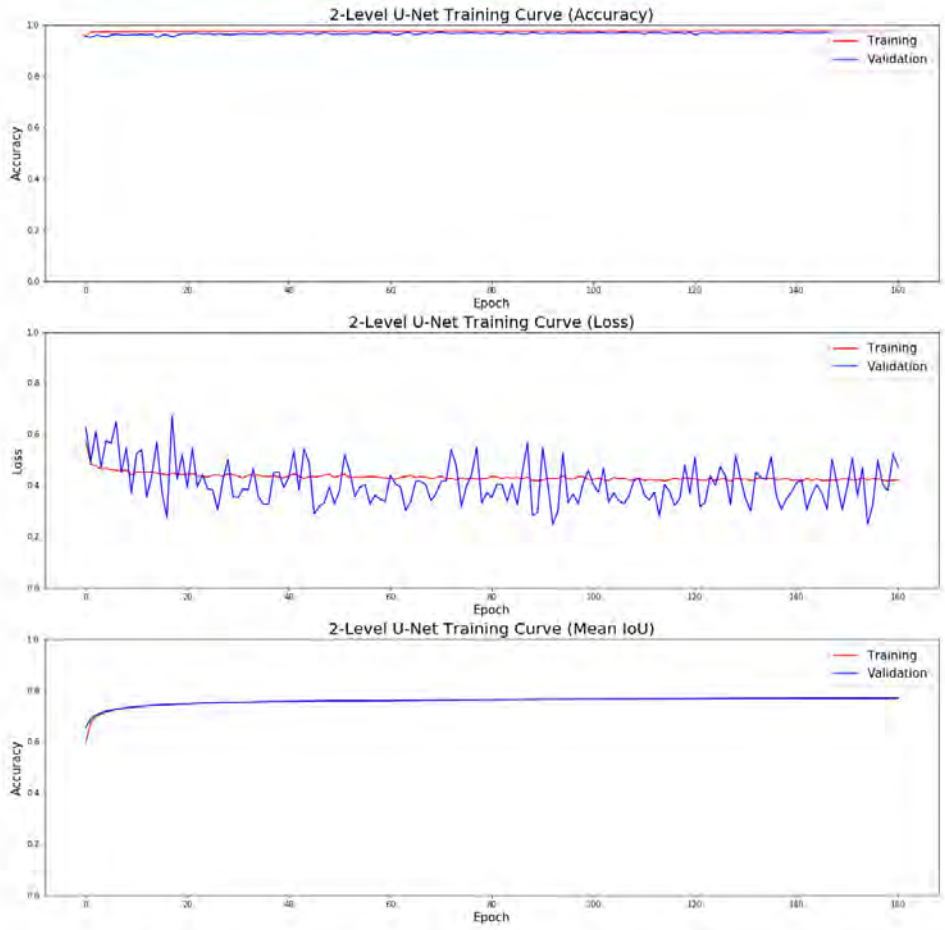

Figure 40: 2-Level U-Net Massachusetts Roads Training Curves

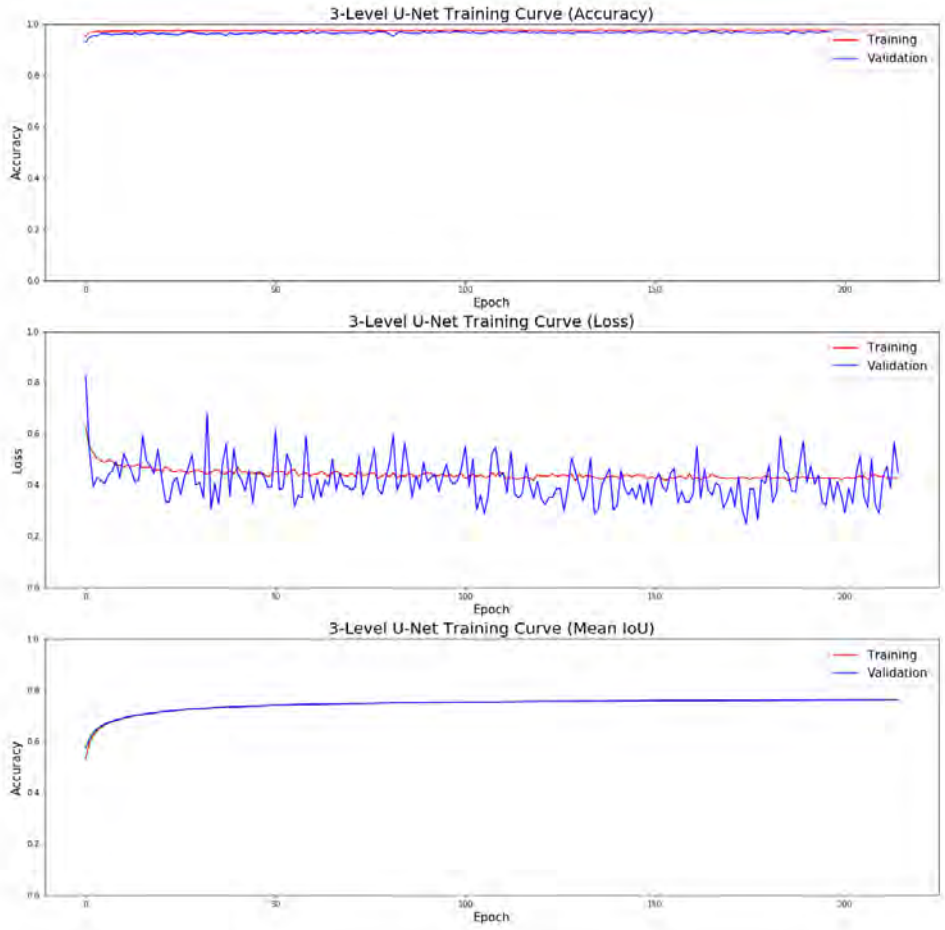

Figure 41: 3-Level U-Net Massachusetts Roads Training Curves

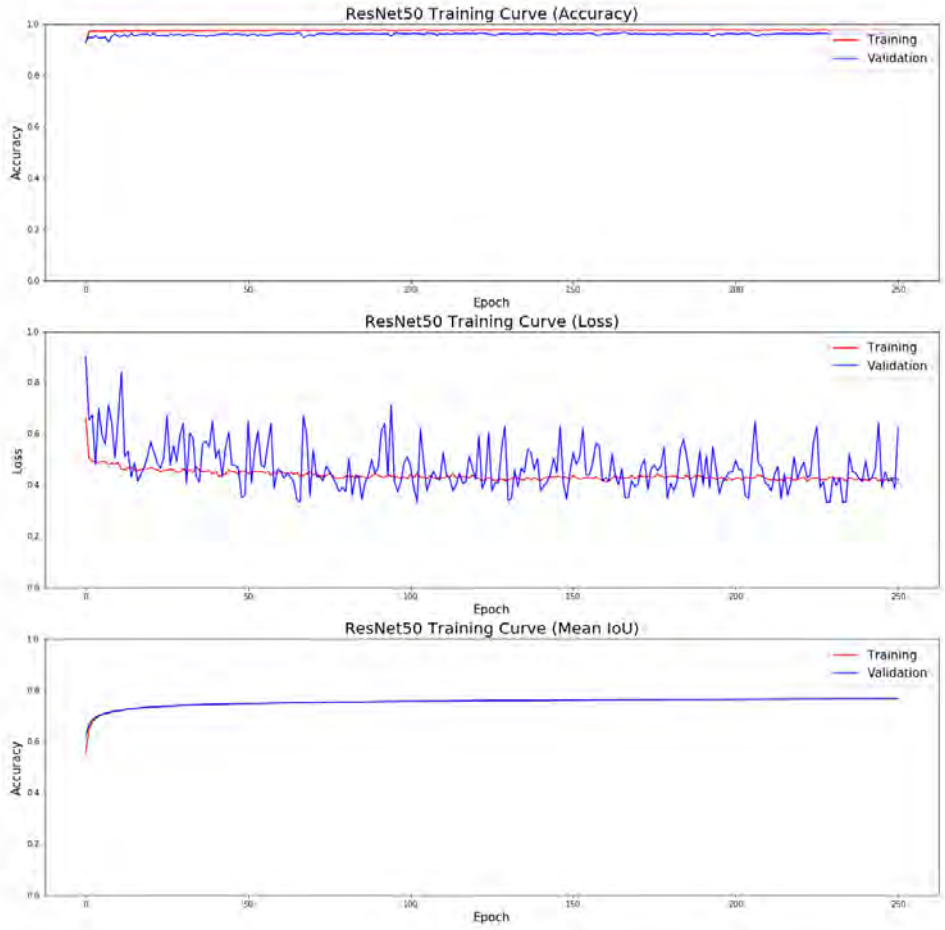

Figure 42: ResNet50 Massachusetts Roads Training Curves

1.4 Roads Dataset

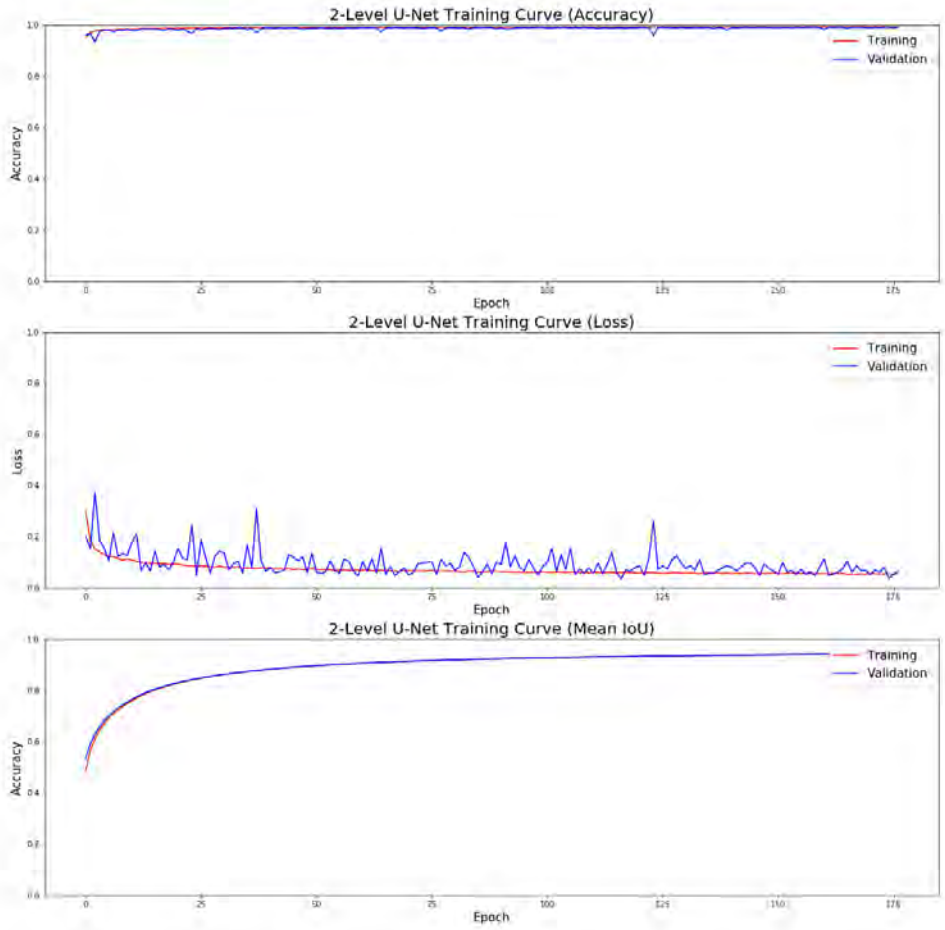

Figure 43: 2-Level U-Net RoadNet Training Curves

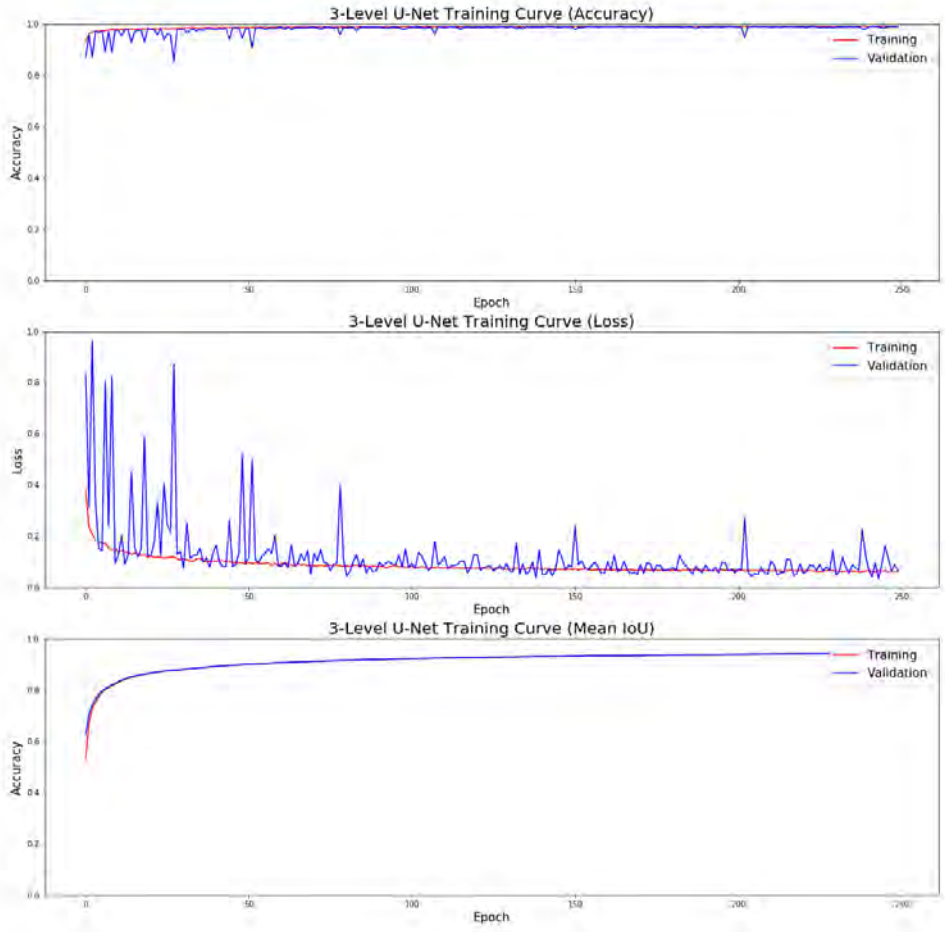

Figure 44: 3-Level U-Net RoadNet Training Curves

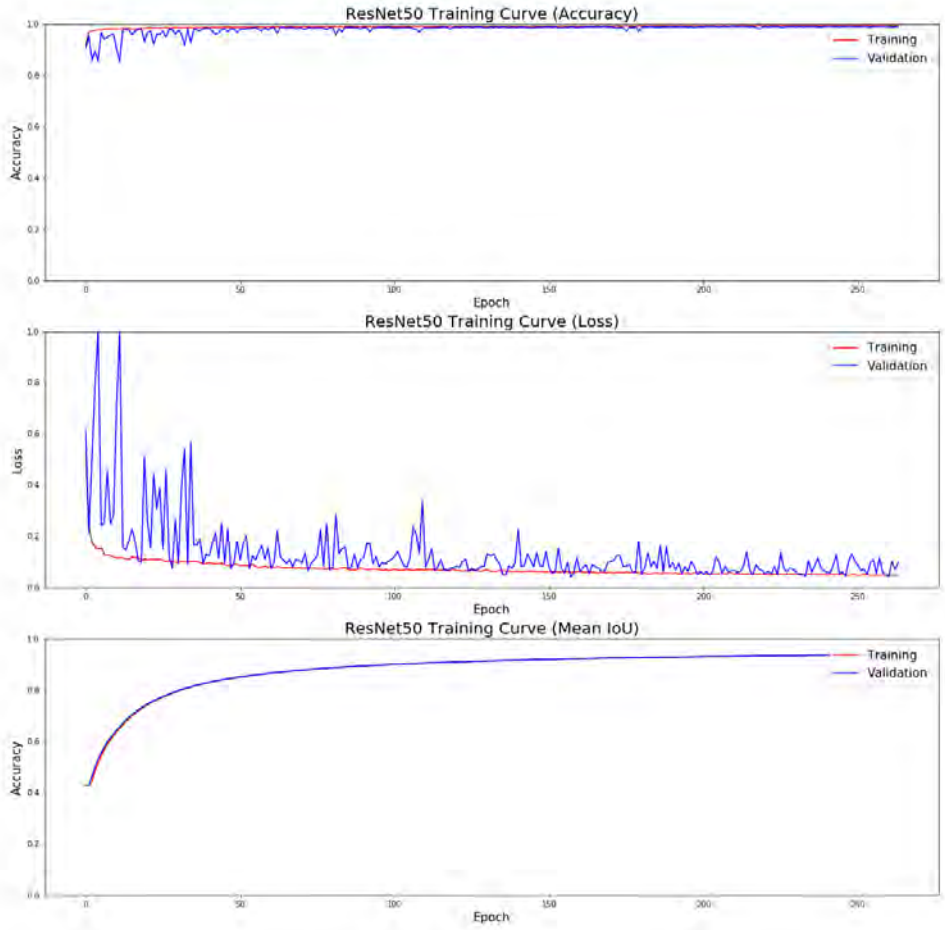

Figure 45: ResNet50 RoadNet Training Curves

1.5 Combined Buildings Dataset

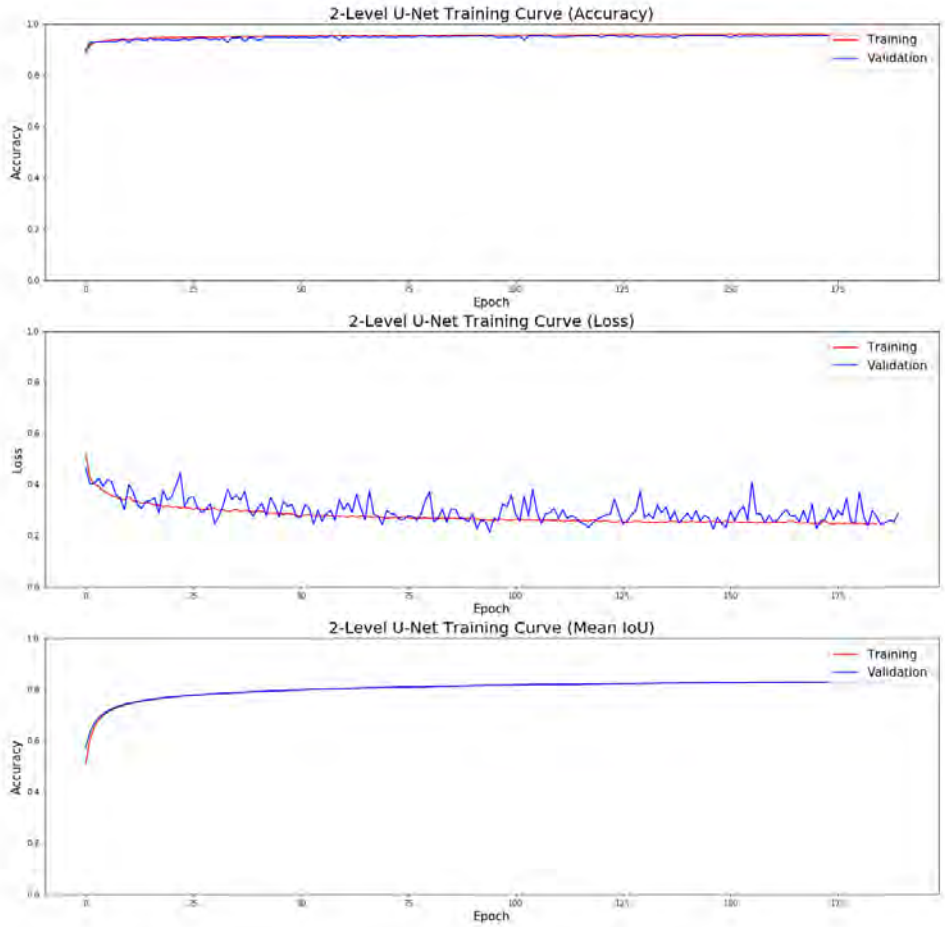

Figure 46: 2-Level U-Net Buildings Training Curves

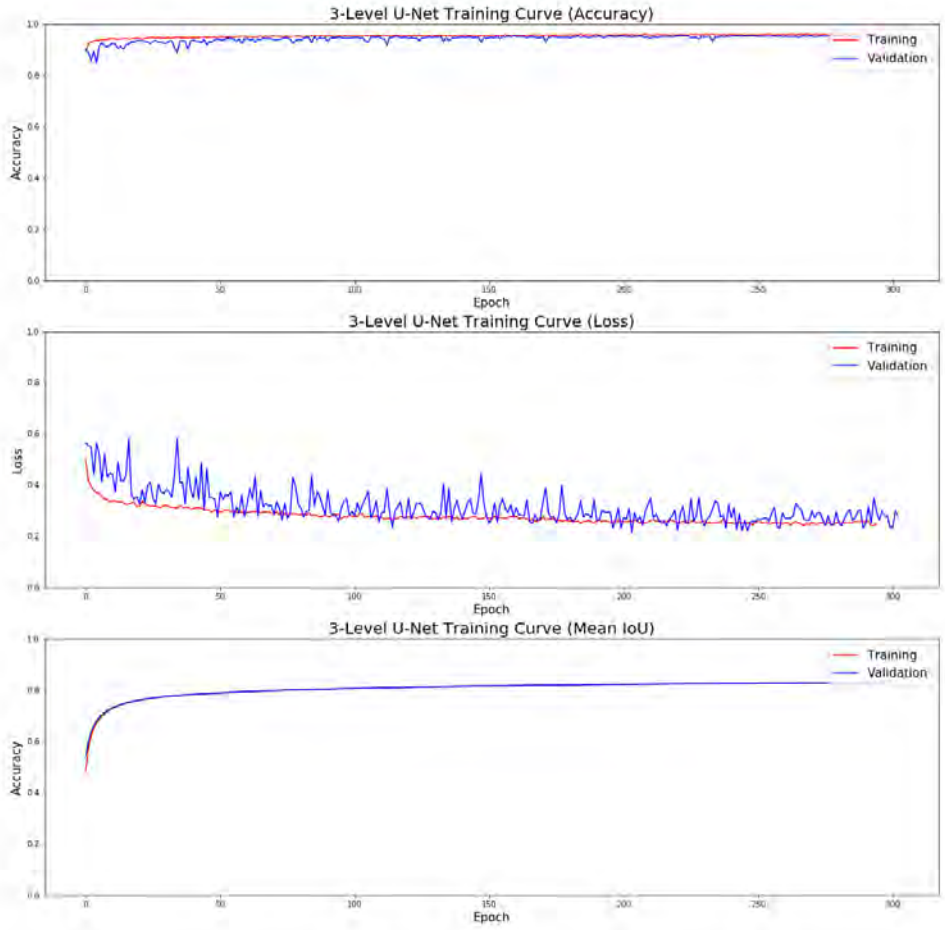

Figure 47: 3-Level U-Net Buildings Training Curves

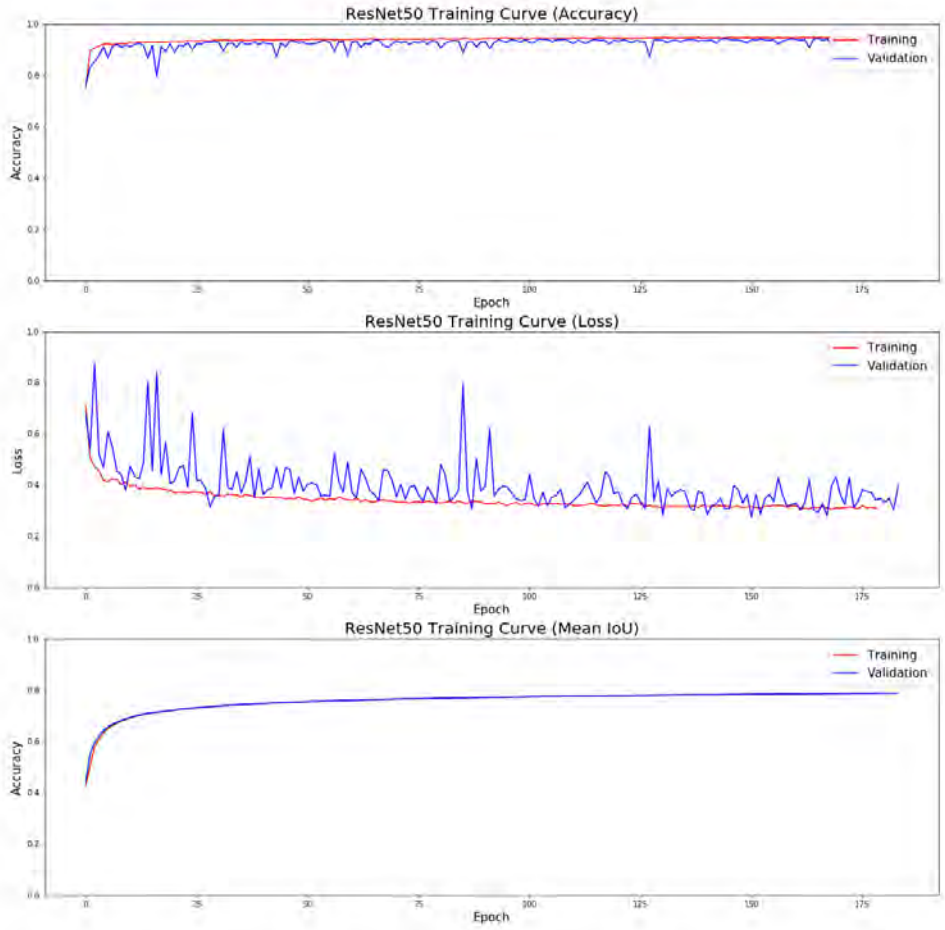

Figure 48: ResNet50 Buildings Training Curves

Bibliography

1. F. Chollet. *Deep Learning with Python*. Manning, Shelter Island, NY, USA, 1st edition, 2017.

2. Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

3. Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

4. Defence Science and Technology Laboratory. Dstl satellite imagery feature detection, Accessed: Feb. 06, 2019. [Online]. Available: https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection/.

5. Qasim Nadeem. Semantic segmentation , urban navigation , and research directions. 2018.

6. Yao Yeboah, Cai Yanguang, Wei Wu, and Zeyad Farisi. Semantic scene segmentation for indoor robot navigation via deep learning. In *Proceedings of the 3rd International Conference on Robotics, Control and Automation*, ICRCA 18, page 112118, New York, NY, USA, 2018. Association for Computing Machinery.

7. Alina Marcu. A local-global approach to semantic segmentation in aerial images. 07 2016.

8. Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017.

9. Abhinav Valada, Johan Vertens, Ankit Dhall, and Wolfram Burgard. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. pages 4644–4651, 05 2017.

10. Zuxuan Wu, Xin Wang, Joseph Gonzalez, Tom Goldstein, and Larry S. Davis. Ace: Adapting to changing environments for semantic segmentation. *ArXiv*, abs/1904.06268, 2019.

11. Lifeng An, Xinyu Zhang, Hongbo Gao, and Yuchao Liu. Semantic segmentation-aided visual odometry for urban autonomous driving. *International Journal of Advanced Robotic Systems*, 14(5):1729881417735667, 2017.

12. Jedidiah M. Berhold. Convolutional neural network architecture study for aerial visual localization. Master's thesis, Air Force Institute of Technology, 2019.

13. Vladimir Iglovikov, Sergey Mushinskiy, and Vladimir Osin. Satellite imagery feature detection using deep convolutional neural network: A kaggle competition. 06 2017.

14. Floris van Beers. Using intersection over union loss to improve binary image segmentation. 2018.

15. I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning (Adaptive Computation and Machine Learning Series*. MIT Press, Cambridge, MA, USA, 2016.

16. P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1451–1460, March 2018.

17. Andrew Khalel and Motaz El Saban. Automatic pixelwise object labeling for aerial imagery using stacked u-nets. 2018.

18. Md Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *Advances in Visual Computing*, pages 234–244, Cham, 2016. Springer International Publishing.

19. Katarzyna Janocha and Wojciech Czarnecki. On loss functions for deep neural networks in classification. *Schedae Informaticae*, 25, 02 2017.

20. Volodymyr Mnih. *Machine Learning for Aerial Image Labeling.* PhD thesis, University of Toronto, 2013.

21. Yahui Liu, Jian Yao, Xiaohu Lu, Menghan Xia, Xingbo Wang, and Yuan Liu. Roadnet: Learning to comprehensively analyze road networks in complex urban scenes from high-resolution remotely sensed images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(4):2043–2056, 2019.

22. Timothy Dozat. Incorporating nesterov momentum into adam.

23. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV 15, page 10261034, USA, 2015. IEEE Computer Society.

24. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

25. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 06 2016.

26. Shunta Saito and Yoshimitsu Aoki. Building and road detection from large aerial imagery. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 9405. SPIE, 2015.

27. Aziguli Wulamu, Zuxian Shi, Dezheng Zhang, and Zheyu He. Multiscale road extraction in remote sensing images. In *Comp. Int. and Neurosc.*, 2019.

Acronyms

AFIT Air Force Institute of Technology. 62

AI artificial intelligence. 5

ANT Autonomy and Navigation Technology. 63

CNN convolutional neural network. 1, 5, 9, 10, 11, 12, 13, 15, 16, 27, 63

FCN fully convolutional network. 13, 14, 15, 35

GNSS Global Navigation Satellite System. 1

GPS Global Positioning System. 1

IoU intersection over union. 18, 19, 20, 21, 35, 36, 38

mIoU mean intersection over union. viii, 3, 17, 18, 19, 20, 21, 34, 38, 40, 51, 53, 55, 56, 57, 60, 62

ReLU rectified linear unit. 28

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
19–03–2020	Master's Thesis	Sept 2018 — Mar 2020

4. TITLE AND SUBTITLE

Semantic Segmentation of Aerial Imagery using U-Nets

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Yi, Terrence J, 2d Lt

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-MS-20-M-075

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Intentionally Left Blank

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

In situations where global positioning systems are unavailable, alternative methods of localization must be implemented. A potential step to achieving this is semantic segmentation, or the ability for a model to output class labels by pixel. This research aims to utilize datasets of varying spatial resolutions and locations to train a fully convolutional neural network architecture called the U-Net to perform segmentations of aerial images. Variations of the U-Net architecture are implemented and compared to other existing models in order to determine the best in detecting buildings and roads. A final dataset will also be created combining two datasets to determine the ability of the U-Net to segment classes regardless of location. The final segmentation results will demonstrate the overall efficacy of semantic segmentation for different datasets for potential localization applications.

15. SUBJECT TERMS

Semantic Segmentation, Image Segmentation, U-Nets, Convolution Neural Networks, Image-Aided Navigation

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Robert C. Leishman, AFIT/ENG
U	U	U	UU	85	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636; robert.leishman@afit.edu