Masters Program in Geospatial Technologies

AUTOMATION OF ROAD FEATURE EXTRACTION FROM HIGH RESOLUTION IMAGES

Kanda Uda Heva Prasadi Thilanka Senadeera

Dissertation submitted in partial fulfilment of the requirements
for the Degree of *Master of Science in Geospatial Technologies*

NOVA IMS Information Management School
UNIVERSITAT JAUME·I
WESTFÄLISCHE WILHELMS-UNIVERSITÄT MÜNSTER

AUTOMATION OF ROAD FEATURE EXTRACTION FROM HIGH RESOLUTION IMAGES

Dissertation supervised by

Mauro Castelli, PhD

NOVA Information Management School (NOVA IMS),

Universidade Nova de Lisboa, Lisbon, Portugal

Dissertation co-supervised by

Filiberto Pla Bañón, PhD

Institute of New Imaging Technologies, Universitat Jaume I

Castellón de la Plana, Spain

Dissertation co-supervised by

Nuno Tiago Falcao Alpalhao, MSc.

NOVA Information Management School (NOVA IMS),

Universidade Nova de Lisboa, Lisbon, Portugal

February de 2021

DECLARATION OF ORIGINALITY

I declare that the work described in this document is my own and not from someone else. All the assistance I have received from other people is duly acknowledged and all the sources (published or not published) are referenced. This work has not been previously evaluated or submitted to NOVA Information Management School or elsewhere.

Lisbon, 17.02.2021

Kanda Uda Heva Prasadi Thilanka Senadeera

[the signed original has been archived by the NOVA IMS services]

ACKNOWLEDGMENTS

" True guidance is like a small torch in a dark forest it doesn't show everything once. But gives enough light for the next step to be safe."

Swami Vivekananda

I dedicated this paragraph to express my special thanks to the people who stood by my side for bringing this journey to a successful end.

Firstly, my deepest sense of gratitude is express to my supervisor Professor Dr. Mauro Castelli for his valuable guidance, positive encouragement, and continuous support throughout the study. It was a great opportunity and a proud privilege for me to be your student. Secondly, I would like to express my deepest thanks and sincere appreciation to my co-supervisors, Professor Dr. Filiberto Pla Bañón and Mr. Nuno Alpnaho, for their support and contribution. Additionally, I am equally thankful to Prof. Dr. Marco Painho for his continuous monitoring and positive encouragement throughout the thesis period. I would also like to convey a deep thank you to the Erasmus Mundus program for funding my Master of Science in Geospatial Technologies.

I would also be grateful to my family in Sri Lanka for their support in every aspect to keep me moving forward. Last not least, I am also thankful to my husband, Rasanka for letting me fly and be successful in life. Without your love, generosity, and support, this would not have been possible.

DEDICATION

I humbly offer this to all beloved Sri Lankans and Europeans account for my education with their taxes who become a catalyst for me to enlighten myself with knowledge and power.

විදු ඇසින් ආලෝකය ලබා ලොව මෙතෙක් නුදුටු මං පෙත් සොයා යාමට මා හට සවිය වුනු සදලුතලයේ සිට මහපොලව දක්වා ගත සිත වෙහෙසවන සුවහසක් ආදරණීය මිනිසුන් හට සෙනෙහසින් පුදන වගයී....!

Eu humildemente ofereço isso a todos os amados cingaleses e europeus responsáveis pela minha educação com seus impostos, que se tornam um catalisador para que eu me ilumine com conhecimento e poder.

AUTOMATION OF ROAD FEATURE EXTRACTION FROM HIGH-RESOLUTION IMAGES

ABSTRACT

The detection of road features from remotely sensed images has become a critical factor in maintaining a reliable and updated road network in a country to provide a base reference for transportation, emergency planning, and navigation. With the recent advances of convolutional neural networks in image processing, several publications are devoted to the development of a method for automatically extract roads from satellite images. However, a reliable feature extraction method has not yet been developed with the desired accuracy and precision, and always seems to be a proportionality between the accuracy and the complexity of these developed methods. The aim of this study was therefore to develop an accurate road extraction method without compromising computational efficiency. In this paper, a semantic segmentation neural network that combines the strengths of transfer learning and U-net architecture is proposed with a minimal network complexity. Further, post-processing based on morphological operations and regional properties of the extracted segments were used to remove the noises from the final output. The results have been compared with different automatic classification and segmentation methods and the results of the proposed method produced an F1 score of 0.83 and high accuracy of 95.57%, more accurate and precise than all the other models for the freely available Massachusetts dataset. Finally, the developed method stood superior to the preexisting methods in terms of performance measure and network complexity.

KEYWORDS

Convolutional neural networks

U-Net Image Segmentation architecture

Road extraction

Transfer learning

Morphological operations

ACRONYMS

ANN – Artificial Neural Network

CNN – Convolution Neural Network

DCNN - Deep Convolutional Neural Networks

FCN – Fuly Convolution Network

FFNN - Feed Forward Neural Network

GC-DCNN- Global context-based dilated convolutional neural network

GPU – Graphics Processing Unit

ISM – Image Segmentation Model

ML – Machine Learning

OSM – Open Street Maps

RAM – Random Access Memory

RELU- Rectified Linear Unit

RFC - Random Forest Classifier

SVMC - Support vector machine classifier

USA – United States of America

VGG16 - Visual Geometric Group

INDEX OF THE TEXT

Pág.

INDEX OF TABLES

INDEX OF FIGURES

1. INTRODUCTION

1.1 Overview of the study

The surface of the Earth can be divided into natural and artificial features via aerial imagery, and the ability to extract such features has played a pivotal role in the development and planning of nations (Shrestha and Vanneschi, 2018), (Wijesingha *et al.*, 2012). Road networks, an example of artificial features that can be extracted as artificial features in aerial imagery, provide a baseline reference for city, transportation, and emergency planning to database or resource management (Singh and Garg, 2013). Each of the applications requires a reliable road network dataset and with the rapidly changing human environment, these datasets need constant updating (Singh and Garg, 2013). In recent years, more attention has been paid to investigating newer and more robust methods to create or update existing road network databases (Wijesingha *et al.*, 2012), (Mahdianpari *et al.*, 2018). Historically, the primary method for developing or updating road network datasets relied on land surveys or digitization on scanned maps, however, those methods were more time and cost intensive. After a few decades of technical innovation, the availability of remote sensing images by means of artificial satellites or aircraft systems enables more information about features of the earth's surface to be collected at a low cost and with higher resolution in a short time. However, to make the remotely sensed imagery more meaningful for data extraction, it is crucial that advancements in information extraction methods continued to be pursued within the scientific community (F. F. Ahmadi, M.J.V. Zoej, H. Ebadi, 2008). Road extraction is one of the main tasks in the field of information extraction and in particular, it is a challenging task because of its complexity due to the availability of noise and occlusion in the satellite imagery and due to the different types of background in which they are located (Zhang, Liu and Wang, 2018). This difficult problem has been tackled in many different ways in the past and many road extraction algorithms have been proposed, such as global thresholding and morphological analysis (J. Wang *et al.*, 2016), texture and hypothesis testing (Bakhtiari, Abdollahi and Rezaeian, 2017), edge detection, support vector machine classification (SVM) and mathematical morphology (F.

F. Ahmadi, M.J.V. Zoej, H. Ebadi, 2008), as well as deep convolutional neural networks (Singh and Garg, 2013). Each of these analyses is based on either spectral information, spatial information, or both together (Singh and Garg, 2013), (F. F. Ahmadi, M.J.V. Zoej, H. Ebadi, 2008), (Bakhtiari, Abdollahi and Rezaeian, 2017). However, each of the algorithms mentioned above had its weaknesses and strengths; for instance, threshold-based road extraction can face problems with noise in their outputs due to the reflectivity values and artifacts (J. Wang *et al.*, 2016), or edge detection methods can often lead to decision-making problems when overlapping features are present (F. F. Ahmadi, M.J.V. Zoej, H. Ebadi, 2008). Because of these problems, methods focusing on spectral or spatial characteristics to extract road features continue to face challenges in their accuracies and most road extraction methods must undergo a large number of image preprocessing and post-processing steps before and after they are applied to an extraction algorithm or process. (Wijesingha *et al.*, 2012), (Alshaikhli, Liu, and Maruyama, 2019). This could adversely affect the development of the automatic feature extraction methods to delineate roads from high-resolution images. However, Deep Convolutional Neural Network (DCNN) architectures, inspired by artificial neural networks that follow the same process as in a biological neuronal system, have given comparatively better results for the road extraction projects due to their ability to effectively combine both spectral ranges and spatial information from remotely sensed images without image preprocessing and little post-processing (Shrestha and Vanneschi, 2018), (Wijesingha *et al.*, 2012). Moreover, DCNN performs incredibly better than the other established methods for this application because of the following factors.

1. They consist of a series of layers made of filters having weights and biases that learn directly from raw input which makes them more suitable in spatial data-related applications, as they can adjust to spatial heterogeneity.
2. In contrast to other methods, which rely on spatial and contextual features of roads that depend on the shape and neighboring objects, DCNNs are not based on assumptions about what roads or their surroundings look like. (Sirefelt Rickard, 2004)

3. The structure of the interconnected nonlinear neurons up to unlimited hidden layers allows an infinite number of neurons to be used for the process, which allows the use of a large amount of data to be involved to get the maximum degree of discrimination to obtain an almost perfect result for the system output. (Shrestha and Vanneschi, 2018)

Therefore, this project will aim to develop an automatic road extraction method using deep convolution neural networks.

1.1 Research Gap

With a sharp increase in the availability of digital imagery and the possibility of getting better results with the DCNN architectures, numerous publications on the subject of road extraction have been published over the past few decades. However, there seems to be a proportionality between the accuracy and the complexity of these developed methods. Most of them require a lot of processing power, computational time, and too much hardware (GPU, processor, and RAM) to get a better result. (Saifi, Singla and Nikita, 2020) Also, although the number of publications shows that the field of road extraction using CNN architectures is improving, reliable feature extraction has not yet been developed with the desired accuracy and precision. Therefore, this study aims to improve the relative accuracy of the extraction of road features in high-resolution images without compromising computational efficiency.

1.2 Research Objectives

The main aim of the research is to formulate a method of road extraction from high-resolution images that uses a deep learning approach based on a convolutional neural network. In order to achieve the main research goal, the following sub-goals are also addressed:

1. Review and assess the potential of the latest deep learning algorithms for automatic road extraction using high-resolution aerial / satellite images.
2. Compare several image classifiers and CNN based segmentation architectures based on accuracies to find out the best method to extract roads using high-resolution aerial/satellite imagery.

3. Enhance the accuracy of the chosen method by changing the chosen network architecture and applying transfer learning.

1.3 Thesis Organization

The thesis consists of 6 chapters, Chapter 1 describes the overview of the work including the research gap and the objectives considered in the project. Chapter 2 reviews the existing literature on road extraction methods using aerial/remote sensing imagery and explains the properties of road features that can be used as parameters for these methods. It also covers the CNN-based U-Net image segmentation architecture and its uses in the field of remote sensing and GIS. Chapter 3 presents the theoretical background of deep convolution neural networks and their architectures and also the concept of transfer learning in CNN applications. Chapter 4 describes the study area, the data sets, and the methodology and tools used for the research. Chapter 5 presents the results of the study while deals with the analysis and discussion of the results. Finally, Chapter 6 summarizes the conclusions by answering key research questions and recommendations for future work.

2. LITERATURE REVIEW

This chapter is intended to give a brief overview of existing literature in the field of road extraction. The first section of the chapter describes the basic knowledge about road extraction using remotely sensed images and is divided into two subsections as classical approaches and deep learning-based approaches. The next subsection compacts with the application of deep learning algorithms for semantic segmentation and reviews the U-Net architecture for semantic segmentation. Finally, the last subsection introduces the concept of transfer learning for deep learning.

2.1 Automatic road extraction using Remote Sensing Imagery

Collecting road data for use in geographic information systems (GIS), navigation, transportation, and emergency planning has played a major role in the advancements of human civilization and constantly requires updating due to the dynamic human environment (Alshaikhli, Liu and Maruyama, 2019). A great effort has been made to automate the task of creating and updating road network datasets through feature extraction methods using remotely sensed imagery (F. F. Ahmadi, M.J.V. Zoej, H. Ebadi, 2008). Many studies recognize the effectiveness of feature extraction in remotely sensed imagery as the development of these methods have been of great benefit to mapping applications, navigation systems, and computer vision (Alshaikhli, Liu and Maruyama, 2019), (F. F. Ahmadi, M.J.V. Zoej, H. Ebadi, 2008). In a meta-analysis by Mena (2003), various road extraction methods were described to explain the growth of the topic over the last 30 years (Mena, 2003). The author classified different methods by their parameters, as well as their advantages and disadvantages while providing an exhaustive review of the existing literature, allowing one to make inferences on which methods would serve best in certain circumstances (Mena, 2003). These extraction methods can be broadly divided into two classes, as classic approaches and deep learning approaches are described below.

2.2 Classical approaches for road extraction

Roads in remotely sensed images can be identified based on their features, which can be summarized as geometric, spectral, structural, topological, and functional aspects (W. Wang *et al.*, 2016). Generally, they appear as stripes with a high ratio between their width and length, and the spectral values of a road change slowly along its path and suddenly changes at the edges of the road (J. Wang *et al.*, 2016). The classical approaches for road extraction have mainly used these features as fundamental properties to detect and extract the roads from other features in the digital images (Mena, 2003). Many researchers have used spectral feature information, such as a combination of adaptive global thresholding and morphological operations (Singh and Garg, 2013), Support vector machine classifier (Bakhtiari, Abdollahi and Rezaeian, 2017), principal component analysis (PCA) (Talal *et al.*, 2014). Image segmentation using gradient and edge detection filters (Hormese and Saravanan, 2016) to extract roads and similarly, the spatial properties of the road segments have also been used in some methods, such as the Hough transform, which uses the shape of the features to identify the incomplete instances of the objects and snakes, the deformable lines that adapt to features of interests such as roads. (Sirefelt Rickard, 2004), (Bakhtiari, Abdollahi and Rezaeian, 2017) Additionally, J.Wang used a knowledge-based approach that utilized the previously derived information about the brightness, aspect ratio, and rectangularity of the road segments to develop a hypothesis model (J. Wang *et al.*, 2016). The performance of these classical approaches are depending on the illumination condition, type of surface material, and presence of disturbing objects yielded a much higher classification accuracy than direct application (J. Wang *et al.*, 2016),(Pasquali, Iannelli and Dell'Acqua, 2019).

2.3 Convolutional neural networks for road extraction

With the advent of artificial neural networks inspired by human biological neurons, they have become a popular tool for analyzing data for various applications. The state of the art in the field of object extraction was later considerably improved with the introduction of convolutional neural networks, in which it is explicitly assumed that the inputs are images. (Shrestha and

Vanneschi, 2018),(Boyagoda and Da Silva, 2020), (Wijesingha *et al.*, 2012). The ability to effectively combine both spectral ranges and spatial information from remotely sensed images without image preprocessing and little post-processing (Shrestha and Vanneschi, 2018), (Wijesingha et al., 2012) makes CNN dominant in the field of road extraction. This subsection discusses the main contributions of the CNN networks to the road extraction projects, highlighting the algorithms used and the architectures developed.

In 2013, Minh developed a CNN-based method to automatically extract features such as streets, buildings, and trees directly from digital images. In his method, a new approach called patch-based CNN was introduced into the semantic segmentation family, in which image patches with a size of 64 * 64 are used as input to the CNN after applying principal component analysis, which reduces the dimensions and increases the interpretability of the images, that allows the creation of uncorrelated variables for the CNN input. (Mnih, 2013). Former research (Alshehhi *et al.*, 2017) used a modified patch-based CNN by replacing fully connected layers with global average poolings and also introducing an enhanced post-processing step to extracts roads and buildings from high-resolution images. In this work as the post-processing, a simple linear interactive clustering that measures the compactness and asymmetry of the extracted features was used to filter out the misclassified regions (Alshehhi *et al.*, 2017). Wijesinghe et al. (2012) extracted road networks in suburban and rural areas from high-resolution which consisted of a two-part methodology: a self-organizing supervised learning neural network for road feature extraction and a typical pattern recognition neural network for comparing performances (Wijesingha *et al.*, 2012). The results showed an accuracy of 70 percent when compared with the existing road network dataset of the same area (Wijesingha *et al.*, 2012).

Buslaev (Buslaev *et al.*, 2018) evaluated the performance of a new CNN model consisted of a ResNet-34 encoder and a decoder extracted from vanilla U-Net architecture on Digital Globe's satellite dataset. In this research, they also used the Jaccard index (intersection over union) for the evaluation matrix to the training phase with the binary cross-entropy to improve the performance (Buslaev *et al.*, 2018). Due to the occlusions and the complex backgrounds of the images, these methods produced low accuracy in some images. Thus, to

minimize those drawbacks, in 2020, Lan developed a new approach called global context-based dilated convolutional neural network (GC-DCNN) (Lan *et al.*, 2020). The structure was similar to the U-net image segmentation architecture and pooling layers were replaces with pyramid pooling module and dice coefficient loss is used for the loss function in the training phase. And his method produced better results over the aforementioned problems (Lan *et al.*, 2020). Alshaikhli, Liu, and Maruyama (2019) more recently proposed a new road extraction method by use of deep convolutional neural networks (DCNN) and presented a new model for encoding the Deep CNN through residual blocks and U-Net (Alshaikhli, Liu and Maruyama, 2019). With these methods, the authors were able to a better result in terms of image prediction when compared to other top models (Alshaikhli, Liu and Maruyama, 2019). Based on the literature review, the approach to be used in this study for detecting road features in remotely sensed images will be Convolution neural networks as proposed by Alshaikhli, Liu and Maruyama (2019) and Wijesinghe et al. (2012) (Alshaikhli, Liu and Maruyama, 2019), (Wijesingha *et al.*, 2012).

2.4 Convolutional neural networks for semantic segmentation

Image segmentation can divide a digital image into smaller subsets of pixels by grouping those with similarity in color, texture, or intensity to tackling the complexity is known as image segmentation (Buslaev *et al.*, 2018). When these subdivisions are raised to the pixel level, the segmentation process is explicitly called semantic segmentation. In recent years, there have been several attempts to develop an efficient architecture for pixel-by-pixel semantic segmentation (Buslaev et al., 2018). Among them, fully convolution neural networks (FCN) can be identified as one of the successful deep learning CNN, which has been produced by replacing the fully connected layers with convolutions which are working as feature extractors for the segmentation process. This can aid in producing feature maps for the image segmentation problems that can be upsampled to obtain an image with the same image dimensions as the input image (Buslaev et al., 2018). The recent trends in image segmentation with deep neural networks are discussed below.

Almost all semantic segmentation architectures consist of two main steps, downsampling to capture the contextual information of the digital image to determine WHAT objects are present and upsampling to restore them to find out WHERE those objects can be detected (Saifi, Singla and Nikita, 2020) Different segmentation architectures have been developed in the past by modifying how these architectures use downsampling and upsampling functions to detect and locate the different objects. For instance, FCN 8, FCN 16, FCN 32, three separate FCN models follow the same up and downsampling method with different skip connections and final convolution layer. The skip connections of the method reduce the overfitting of the model, reasoned to enhance accuracy. Khan, (Khan *et al.*, 2020) evaluates several deep neural network models for semantic segmentation in 2020, in his method he worked with four selected convolutional neural network architectures (FCN, SegNet, U-Net, and DeepLabV3+), and class weight balancing was used to avoid the effect of unbalancing the number of pixels in background and prostate. The results of the research conclude that the best results of 92.58% accuracy were given by the DEepLab V3+ model (Khan *et al.*, 2020). Research by (Noh et al., 2015) proposed a novel image segmentation algorithm that consisted of two parts: convolution and deconvolution networks, where the convolution network was adopted from the pre-trained VGG-16 model and the deconvolution network includes unpooling, deconvolution - and rectification layers to get the final segmentation map. A study conducted by (Badrinarayanan, Kendall and Cipolla, 2017) developed a novel decoder-encoder architecture followed by a final pixel-wise classification layer for semantic segmentation called SegNet. The encoder of this architecture consisted of 16 Convolution layers, which are built based on VGG 16 and the method produced an outstanding performance for both road scenes and SUN RGB-D indoor scene segmentation tasks (Badrinarayanan, Kendall and Cipolla, 2017).

2.5 U net image segmentation architecture for semantic segmentation

The U-Net image segmentation architecture is an asymmetrical FCN (Fully Convolutional Network) that was introduced by Ronneberger in 2015 and

consists of a contracting and expanding path combined with an intermediate bridge. In contrast to other image segmentation models, U-Net consists of intermediate concatenation connections, which make it possible to transfer information directly from low to high levels, whereby more precise segmentation results can be achieved with just a few training images (Ronneberger, Fischer and Brox, 2015), (Barrile and Bilotta, 2016), (Saifi, Singla and Nikita, 2020). It has been widely used in medical image segmentation applications, and there are also some contributions in the road extraction field. Abderrahim evaluated the performance of the U-Net architecture via three segmentation models (FCN, RSRCNN, SegNet) for Minh's data set gives the highest accuracy of 97.7% compared to the other methods. (Abderrahim, Abderrahim and Rida, 2020). In this method, data augmentation is used to improve the accuracy and precision of the segmented image (Abderrahim, Abderrahim and Rida, 2020). The road extraction method proposed by Zhang improved the U net architecture by applying residual learning into the encoder part instead of using plain neural units (Zhang, Liu and Wang, 2018). In this method, the skip connections within a residual unit, that were innate through residual learning and skip connections between low and high levels of the network, that were innate by U-Net architecture, were used at the same time to minimize information degradation allows obtaining better accuracy for the road extraction (Zhang, Liu and Wang, 2018). Later a study conducted by Alshaikhli modified Zhang's deep residual U-Net model by applying plain convolution layers, produced more enhanced outputs for the road extraction challenges (Alshaikhli, Liu and Maruyama, 2019). Compared to the other state of the art in road extraction methods, U-Net-based methods led to more accurate results (Zhang et al., 2018). By examining the trend and the results of previous work in the field of road extraction, it is confirmed that CNN, which is developed based on a U-Net architecture, can perform more accurately and precisely. Therefore, a U-Net network architecture was selected for the study.

2.6 Transfer learning for deep convolution neural networks

Because of the structure of the interconnected nonlinear neurons into unlimited hidden layers, an infinite number of neurons can be used for a CNN model, which could result in a gigantic number of training parameters for a particular CNN model. Therefore, training such a model required a considerable amount of training data and also a lot of computing power (Shrestha and Vanneschi, 2018). The most recent developments in machine learning therefore tried to transfer the knowledge of one model (source domain) to another similar model (target domain) (Huang, Pan and Lei, 2017). This process of transferring weights and biases of a pre-trained model to another specific model is known as transfer learning and is an efficient way to train a CNN model with a small number of training patterns and less processing time and power (Huang, Pan and Lei, 2017). Research conducted by Xie sought to develop a CNN model to map poverty in Uganda. Due to the lack of reliable data to train the network, they used transfer learning to train the model with the pre-trained VGG 16 model, which has facilitated an increase in the accuracy of the extraction model from 0.63 to 0.76 (Xie *et al.*, 2016). Therefore, in this study, the concept of transfer learning is used to train the developed U-Net network using a pre-trained VGG 16 model to further improve the accuracy of the work.

3. THEORETICAL BACKGROUND

This chapter provides a background overview of the theories that were involved in the study. The first section describes the concepts of artificial neural networks in detail. The second section covers the CNN architecture, training approaches, and hyperparameters tuned for optimal performance, and the last section gives a brief explanation of the architectures of U-Net and VGG 16 models.

3.1 Artificial neural networks

As computer science has advanced, scientists have attempted to build computer software that simulates the human mind and performs an intelligent task (Alshaikhli, Liu and Maruyama, 2019). As a result, unused innovation which is called Artificial Neural Networks (ANN) that are motivated by biological neurons has been presented to the family of statistical learning algorithms (Alshaikhli, Liu and Maruyama, 2019) (Khan *et al.*, 2020). A biological neuron receives signals from dendrites, processes them within a cell body, and finally transmits the processed information to the brain via an axon and vice versa. Artificial Neural Networks follow the same process as the biological neurons in which they receive the signals from input layers, process them within the neuron, and finally produce an output that represents the processed information from the ANN (Shrestha and Vanneschi, 2018).

An ANN is made up of multiple perceptron (non-linearly connected neurons), most often arranged in layers so that each neuron is connected to the other neurons in the previous layer, as shown in Figure 3.2, which is called the feed-forward neuron Networks (FFNN). Therefore, each perceptron in a neural network receives a set of input signals $(x_1, x_2, x_3, \dots)$ from the neurons of the previous layer or the environment, which are then fed in via connections with weights to calculate the weighted sum plus a bias value (b) $(l = w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + \cdots + b)$ in order to obtain the activation value. Then the activation value is "squeezed" by an activation function to determine the output value $z, z = f(l)$ which is the input for the next neuron or the final output of the network (Sirefelt Rickard, 2004), (Ayo and Da Silva, 2020), (Boyagoda and Da Silva, 2020).

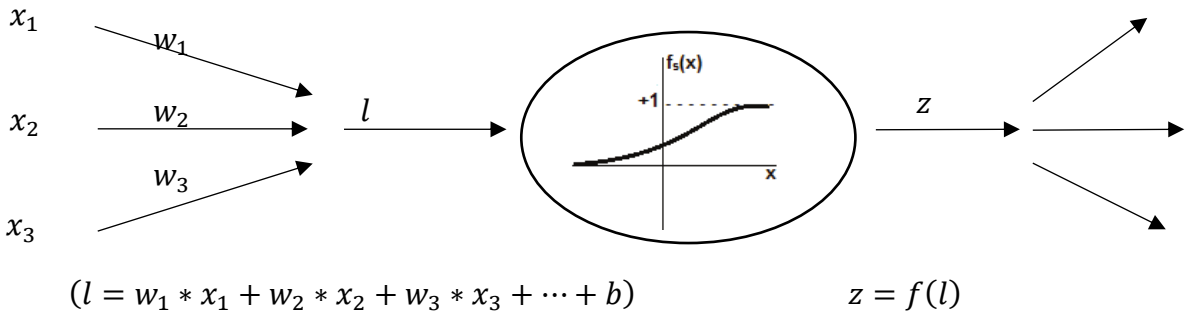


Figure 3.1: Functionality of a perceptron

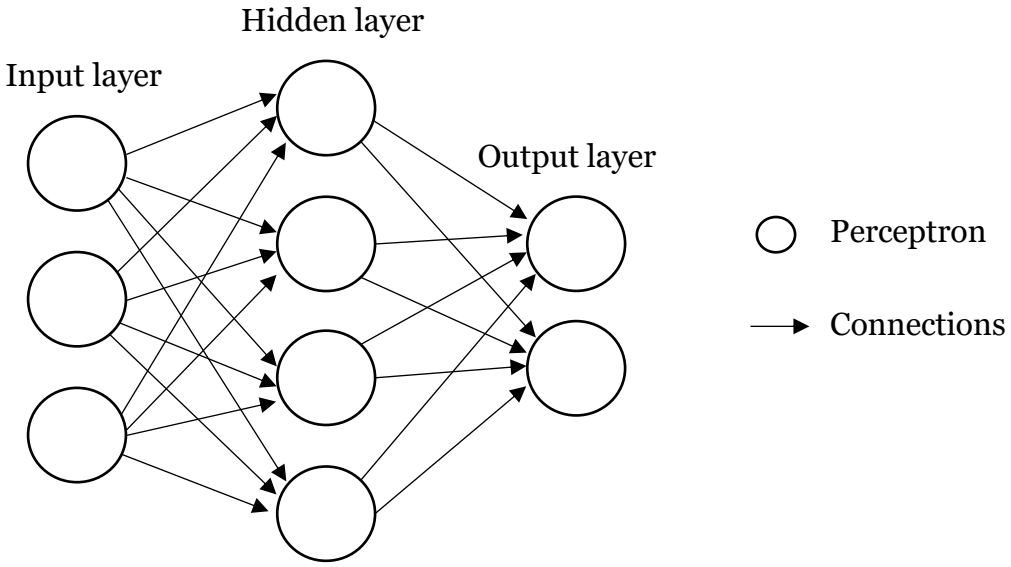


Figure 3.2: Structure of an artificial neural network

When then FFNN consists of more than one layer stacked with each other, it is said to be a deep neural network (Sirefelt Rickard, 2004). To empower an artificial neural network to generate the desired output, regardless of whether it is an FFNN or a Deep NN, the weights and biases of each neuron should be determined by a learning algorithm using a set of training data (inputs and their corresponding labels) before it uses for any application. This enables the artificial neural networks to learn from experience and to make predictions for unknown future operations. If the structure of the data fed to the ANN is images, this particular branch of artificial neural networks is called CNN (Ayo and Da Silva, 2020).

3.2 Convolution neural networks, training and hyperparameters

3.2.1 Convolution neural networks (CNN)

Convolution neural networks (CNN) have been a straightforward approach in the field of computer vision for many applications such as object recognition (Boyagoda and Da Silva, 2020), feature extraction (Saifi, Singla and Nikita, 2020), image classification (Ayo and Da Silva, 2020), semantic segmentation (Tran and Le, 2019), and character recognition. The CNNs are a form of ANN that has been expressly developed to detect objects in the images (Ayo and Da Silva, 2020). Therefore, the input layers of the CN network are made up of neurons that accept three-dimensional responses correspond to the image width, height, and the number of spectral bands (usually 3 for R, G, B, channels) (Boyagoda and Da Silva, 2020). As shown in figure 3.3, traditionally, the structure of the CNN is mainly composed of a convolution layer followed by a fully connected layer, which is alternatively stacked, and the main components of the convolution layer are composed of a convolution, an activation layer, and a pooling layer (Alshehhi *et al.*, 2017). Various CNN architectures have been proposed for image segmentation in the past, e.g., B. Seg Net and Google Net, Alex Net, etc.

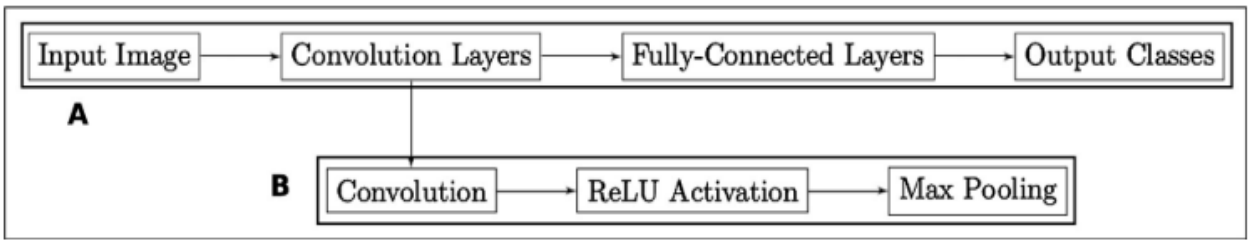


Figure 3.3: The standard architecture of the CNN, A: Main components B: Components of the Convolution layer (Alshehhi et al., 2017)

The convolutional layers in the CNN capture the contextual information of the digital image and generate a new image called a feature map, formed by computing the dot product between the coefficients of the spatial filter and the pixel values of the image at each position in the image. After completing a full forward pass across the width and height of the image, the resulting feature map is passed through a nonlinear activation function, which squeezed the values in

the feature map according to a certain function (e.g., ReLU, tanh, etc.). Then after the altered feature map is passed through a pooling operation to reduce the dimensions and the complexity of the feature map via a pre-defined function (Maximum, Average, Minimum, etc.). It is generally done by moving a filter across the width and height of the input with a specified stride size (usually 2) while taking the maximum within the filter which is called Max pooling. Then finally a fully connected layer, which is comprising of neurons, in which each neuron is connected to all others in the previous layer is used to getting a meaningful network output. Once an image has been completely passed through CNN, it predicts the class of the object in the image as the output. In the case of a road extraction application, the final output layer defines whether or not each pixel of the image represents a road (Shrestha and Vanneschi, 2018), (Boyagoda and Da Silva, 2020), (Ayo and Da Silva, 2020), (Sirefelt Rickard, 2004), (Alshehhi *et al.*, 2017), (Wulamu *et al.*, 2019).

3.2.2 Training approach

Once a CNN architecture has been developed, it should train on a range of known data before using it for the desired application. For this step, it is important to have a set of accurate training data that represents the inputs into the CNN as well as the desired labels for the outputs. The training process of the CNN modifies the weights and biases of the convolutions in such a way that a given set of inputs achieve their desired output and this process consists of three steps: namely forward computation, loss optimization, back-propagation and parameter updating. Forward computation returns the class labels for input images as a probability to belongs to a certain class, then loss optimization, optimize the probability scores by adjusting the weights and biases of the convolutions which have been trained over the network. Finally, backpropagation gradually updates the weight and biases of the whole networks using error surface derivatives (Shrestha and Vanneschi, 2018).

3.2.3 Hyperparameters

Hyperparameters are the parameters that must be set before the training process, e.g. Learning rate, number of epochs, weight and bias initializations, etc. (Sirefelt Rickard, 2004). There are three ways of setting these parameters.

1. Manual: the basic method where initial parameters are set by hand, considering the prior knowledge of the application, or predicting the values.
2. Search algorithms: provides the feasible ranges and combinations of parameters to train the network for the optimal solutions.
3. Automatic approach: Create an automatic method to initialize the parameters for an optimal solution (Shrestha and Vanneschi, 2018).

In this study, all three methods were used to set the hyperparameters, initial weights, and biases of the network were initialized with the kernel initializer "he_normal" of the Keras working environment which assigns a random value from a normal distribution, centered on 0. The learning rate was initially set to 0.0001 and gradually reduced with the number of epochs using an automatic function. The number of iterations was initially set at 100 and the early stop principle was used to stop the training process as soon as the validation set's performance stopped increasing.

3.3 U Net network architecture

The following subsection describes the U-Net convolutional neural networks that Ronneberger proposed in 2015 for biomedical image segmentation (Ronneberger, Fischer and Brox, 2015). It has been recognized as one of the highly successful CNN architectures for segmenting different medical images in the field of cardiology and neurology (Abderrahim, Abderrahim and Rida, 2020). The U-Net network architecture is shown in Figure 3.4 and is essentially similar to the letter “U” in the English alphabet. The architecture mainly consists of three modules, an encoder (contraction path), a decoder (expansion path), and a bridge in between them to connect the output of the encoder to the decoder (Zhang, Liu and Wang, 2018) and a total of 23 convolution layers are used in all three modules. The contraction path consists of repeated convolution blocks made up of a 3 * 3 convolution layer followed by a ReLu activation layer and a 2 * 2 max-pooling layer. within each contraction block, the number of feature maps is doubled, and the size of the feature map is halved. The network expansion path consists of a sequence of up sampling convolutional layers that double the size of the feature map and halve the

number of feature maps. Additionally, these up sampling layers combine the high-level features of the image objects using intermediate concatenations. Finally, the output layer uses a 1 * 1 size convolution layer to output the segmented image in the same dimensions as the input image (Abderrahim, Abderrahim and Rida, 2020), (Zhang, Liu and Wang, 2018), (Ronneberger, Fischer and Brox, 2015). The symmetrical structure and the process of combining the properties of the image objects at high and low level through concatenation make the U-Net structure unique from the other available segmentation models (Tran and Le, 2019).

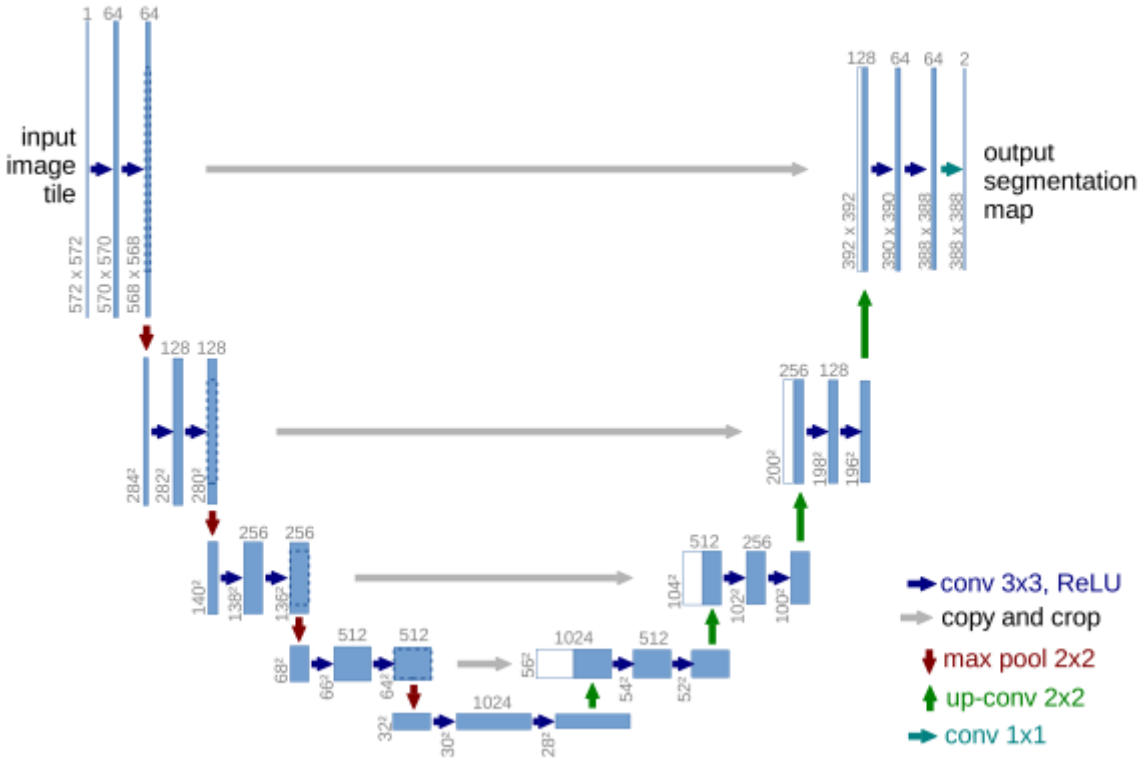


Figure 3.4: U-net architecture (example for 32x32 pixels in the lowest resolution) (Ronneberger, Fischer and Brox, 2015)

3.4 VGG 16 pre-trained model

VGG 16 is a convolutional neural network model that has been proposed by K. Simonyan and A. Zisserman in 2015 (Simonyan and Zisserman, 2015). This model has two different architectures as VGG 16 and VGG 19, VGG 16 network configuration consist of 16 layers and 19 layers in the VGG 19. It has been

recognized as an extremely successful feature extractor in the field of image segmentation (Boyagoda and Da Silva, 2020). The following figure 3.5 shows the architecture of the VGG-16, it consisted of five repeated convolution layers (3*3 filter size) and each of them is followed by a max-pooling layer that is performed over a 2×2-pixel window, with stride 2.

Figure 3.5: VGG 16 network architecture

The convolutional blocks are then followed by three fully connected layers; The first two each have 4096 channels and the last layer is a Softmax layer. This VGG 16 model has already been trained on a data set of over 14 million images from 1000 classes (ImageNet data set) and has achieved a test accuracy of 92.5% (Simonyan and Zisserman, 2015). Weights of this trained model are used in this research project to increase the accuracy of the developed U-Net architecture. The main reason that makes the VGG 16 most suitable in this project is that the repeated convolutions of both models i.e. VGG 16 and the U-Net has the same initial formation (filter size (3*3) and the size (2*2) and stride (2) of the max-pooling layer).

4. DATA AND METHOD

This chapter contains a description of the dataset used and the methodology followed for the study. The first subsection reviews the properties of the data set and the second section covers each step of the methodology; preprocessing and data preparation steps, method selection, model optimization, hyperparameter selection, and transfer learning with the predefined VGG 16 model. The next subsection provides an overview of the software, hardware, and tools utilized to implement the proposed method, and the comparison measures used in the study are described in the last section.

4.1 Data Description

In this experimental project, a freely available road dataset prepared by Mnih was used (Mnih, 2013). The dataset consists of ariel photographs covering the entire state of Massachusetts in the United States, and their respective ground truth raster images (labels), which were created using open street maps (OSM). The dimensions of the image were initially 1500 * 1500 pixels and comprised

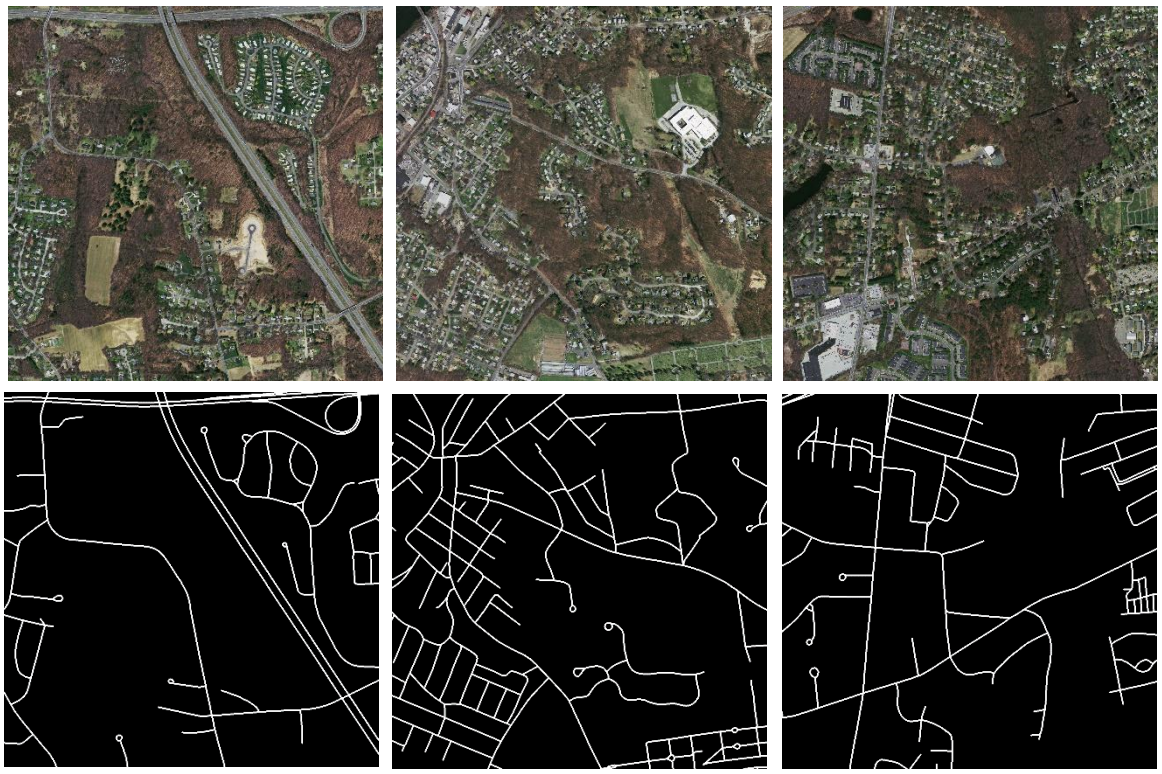


Figure 4.1: Three samples of Massachusetts data set a) Aerial Images, b) their respective ground truth images from OSM

of three spectral bands (Red, Green, and Blue) cover 2.25 square kilometers targeted to develop images with a spatial resolution of 1 meter. The sample of a data set is shown in figure 4.1.

4.2 Method

In this thesis, a new method for delineating roads from high-resolution images was proposed and this subsection describes the methodological framework of the study in detail. The main outline of the procedure is shown in Figure 4.2, which comprises five main steps: data preprocessing, method selection, model development and hyperparameter selection, transfer learning, and post-processing. All six steps of the method were coded utilizing Python programming language and the corresponding libraries that are installed in the Anaconda working environment. The implemented codes can be found at this link.

4.2.1 Data pre-processing

Data preprocessing is vital in deep learning applications (Shrestha and Vanneschi, 2018). The main purpose of data preprocessing is to transform or encode the data set used in the application so that the characteristics of the data can be identified and easily interpreted by the learning algorithm (Sirefelt Rickard, 2004). Therefore, the first step in the method was to convert the data set into a suitable format that was best fitted to the network. Since the developed method attempting to delineate roads using deep CNN without compromising the computational efficiency, the first step of the image preprocessing was to reduce the dimensions of the images to preserve the memory while the CNN algorithm is running. So, in this step, the dimensions of the images were lowered from 1500 pixels to 300 pixels by dividing an original image into 25 components.
As a result, as shown in Table 1, the number of images is increased by 25 times. Then the images were arbitrarily split into three separate parts to be used for training, validation, and testing purposes.

	Training	Validation	Testing
No. of original images (1500*1500-pixel dimensions)	360	40	120
Images with 300*300 pixels image dimensions	9000	1000	3000

Table 4.1: Number of randomly distributed Training, validation, and testing images before and after image cropping.

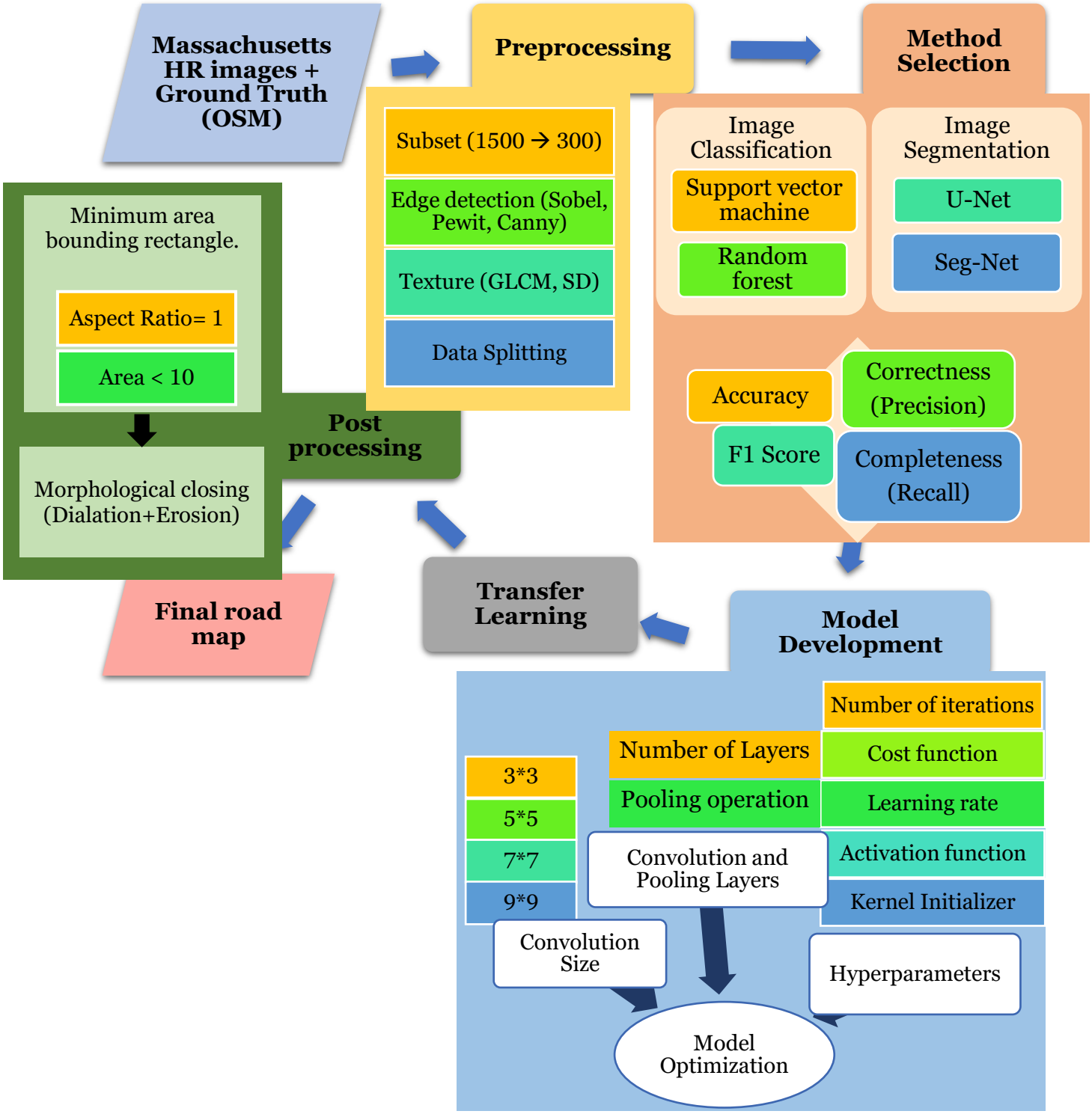


Figure 4.2: Methodological framework of the study

Ground truth labels of the respective images were also passed through the same steps to reduce the dimension to be reconciled with the input data. The training data set together with its respective ground truth labels were used to train the developed methods and the validation set was used to validate the best possible model during the training process and finally, the test data set was used to evaluate the models through comparative measures. As the next step in the image preprocessing, the texture bands and the outputs of the edge detection filters for the original images were derived to quantify the spatial variability of the neighborhood. The dissimilarity feature of the grayscale coexistence matrix and the standard deviation of the spectral values in the images across a 3 * 3 filter was used as the texture measures. Similarly, three readily available edge detectors, Canny, Sobel, and Prewitt, were used for edge detection. these all five output bands were used as the secondary measures for the classification algorithms.

4.2.2 Method Selection

At the beginning of the method selection process, four feature extraction algorithms were chosen due to their balance of accuracy and network complexity as tested in the previous literature. These four methods included two image classifiers, namely, a support vector machine classifier, a random forest classifier, and two image segmentation architectures as U-Net image segmentation architecture and SegNet image segmentation architecture. After implementing the methods using Python's executable code, the performance of each method was assessed using four comparative measures: Accuracy, Precision, F1 Score, and Recall. Then, the method which produced the highest F1 score was selected as the best model to proceed with the research. Detail explanation of comparison measures is written in section 4.4.

4.2.3 Model development and hyperparameter selection.

In this step, A new model based on U-Net image segmentation architecture was developed and the sensitivity to the hyperparameters on the designed CNN was tested. This experiment was mostly built on top of the TensorFlow learning platform working in the python environment and additionally, the Keras application programming interface was also used to reduce the cognitive load.

During the model development, the number of the convolutional and pooling layers in the architecture, the convolution size, and the operation of the pooling layer in the algorithm was changed sequentially to find the best model and in the case of hyperparameters, the learning rate, the number of iterations, the kernel initializer, the activation function, and the cost function have also been optimized in order to train the network with the lowest losses for validation while protecting the network from overfitting.

4.2.4 Transfer learning

The concept of transfer learning was used in this step, to transfer the knowledge from a trained VGG-16 model to enhance the accuracy of the designed method. As visualized in figure 4.3, the first 13 layers, 10 convolutions, and 3 pooling layers of the contraction path (encoder) were set as nontrainable to replace the weights and biases of the layers with the VGG 16 ideals.

```
0 input_4 - False
1 block1_conv1 - False
2 block1_conv2 - False
3 block1_pool - False
4 block2_conv1 - False
5 block2_conv2 - False
6 block2_pool - False
7 block3_conv1 - False
8 block3_conv2 - False
9 block3_conv3 - False
10 block3_pool - False
11 block4_conv1 - False
12 block4_conv2 - False
13 block4_conv3 - False
14 conv2d_transpose_9 - True
15 concatenate_9 - True
16 conv2d_21 - True
17 dropout_9 - True
18 conv2d_22 - True
19 conv2d_transpose_10 - True
20 concatenate_10 - True
21 conv2d_23 - True
22 dropout_10 - True
23 conv2d_24 - True
24 conv2d_transpose_11 - True
25 concatenate_11 - True
26 conv2d_25 - True
27 dropout_11 - True
28 conv2d_26 - True
```

Figure 4.3: Trainable and no trainable layers of the developed U-Net based road extraction model

4.2.5 Post-processing

The final step of the methodology was to enhance the visual interpretation of the final road map by applying post-processing strategies. After extracting the roads with the developed U-Net model, the segmented output consists of noise due to the spectral similarities that exist between streets and other man-made structures, particularly buildings. The post-processing technique developed in this step combines the morphological operations and the factors calculated on the minimally bounded rectangular box (MBRB). Specifically, the first step was to draw the minimal bounding rectangle for the objects in the segmented image to better describe the shape features of the polygons. Then the features with an aspect ratio of 1 and areas of less than 10 pixels were removed to overcome the negative influences of the buildings. Then, the basic operations of the mathematical morphology include dilation and erosion were used to further remove the noises and fill the gaps

between the road segments. The image closing that is dilation ($\oplus$) followed by erosion ($\ominus$) is described as follows.

$$A \cdot B = (A \oplus B) \ominus B$$

Where A is the binary image and B represents the structuring element. The structuring element proposed in (Talal *et al.*, 2014) was used for the study due to its high performance and effectiveness. consequently, roads are detected more completely.

4.3 Tools and Hardware

The entire model described in subsection 4.2 has been implemented using open source software and packages. The implemented codes executed on a laptop with an Intel Core i3 CPU running at 2.0 GHz and 8 GB of RAM, can be found in this GitHub repository. The resources used are described below.

Anaconda: Anaconda is a popular open-source distribution that comes with over 200 automatically installed packages and is aiming to simplify the python package management and deployment by providing suitable working environments for Windows, Linux, and macOS operating systems.

Python: Python is a programming language that supports the development of logical code to work with multiple approaches to programming, including structured (especially procedural), object-oriented, and functional programming for end users. The python version of 3.7 installed on Anaconda was used in the study.

Spyder: Spyder is an integrated development environment (IDE) that supports programming in the python language for scientific studies, which is already packed with frequently used dependencies as NumPy, NumPy, SciPy, Matplotlib, and pandas. In this study, code development and visualization were carried out using the Spyder IDE.

TensorFlow with Keras: TensorFlow is an open-source library designed for in-depth neural network training and interference for machine learning applications developed by the Google Brain team. It can be installed as a GPU or CPU deployment, depending on the requirements and hardware used for the study. Lightweight CPU deployment was used in this study to create the code in a way that is able to execute with the least hardware requirement. Keras is

the official high-level API that facilitates the reduction of the cognitive load and supports multiple backends as Theano, CNTK, etc.
In addition to the aforementioned, various python libraries and tools as Skit Learn, Geopandas, Pillows, OpenCV were also used for the implementation when it is required.

4.4 Comparison measures

This final subsection focuses on describing the accuracy assessment process that was carried out to validate the performance of the model and for the method selection phase discussed in section 4.2.2. The accuracy assessment aims to identify and quantify the errors by comparing the pixels or polygons from a segmented map with the known reference data set called the ground truth or image labels (Itza Alejandra *et al.*, 2020). Four comparison measures; Accuracy, Precision, Recall, and F1 Score were used in this study. Since road extraction is viewed as a binary problem consisting of pixels representing roads and non-roads in output segmented images or labeled known images, there are four possible states for the confusion matrix as true positives, true negatives, false positives, and false negatives. They can be defined as follows,

1. True positives (TP): Number of correctly classified target pixels (roads)
2. True negatives (TN): Number of incorrectly classified target pixels (roads)
3. False positives (FP): Number of correctly classified background pixels (non-roads)
4. False negatives (FN): Number of incorrectly classified background pixels (non-roads)

The following formulas were used to define the comparative dimensions.

1. Precision (correctness): The ratio between the true positives and all positives captured by the model, namely true and false positives.

$$Precision = \frac{TP}{TP + FP}$$

2. Recall (completeness): measures the proportion of correctly classified target pixels to all true target pixels.

$$Recall = \frac{TP}{TP + FN}$$

3. Accuracy: Is refers to the ratio between the number of correctly classified pixels (true positives) to the total number of pixels, that is the sum of true positives, true negatives, false negatives, and false positives.

$$Accuracy = \frac{TP}{TP + TN + FP + FN}$$

4. F1 Score: defines the harmonic mean of precision and recall, is mainly used to assess the accuracy of an unbalanced data set where the number of target pixels and background pixels are different in amounts (Shrestha and Vanneschi, 2018).

$$F1\ Score = \frac{2 * precision * recall}{precision + recall}$$

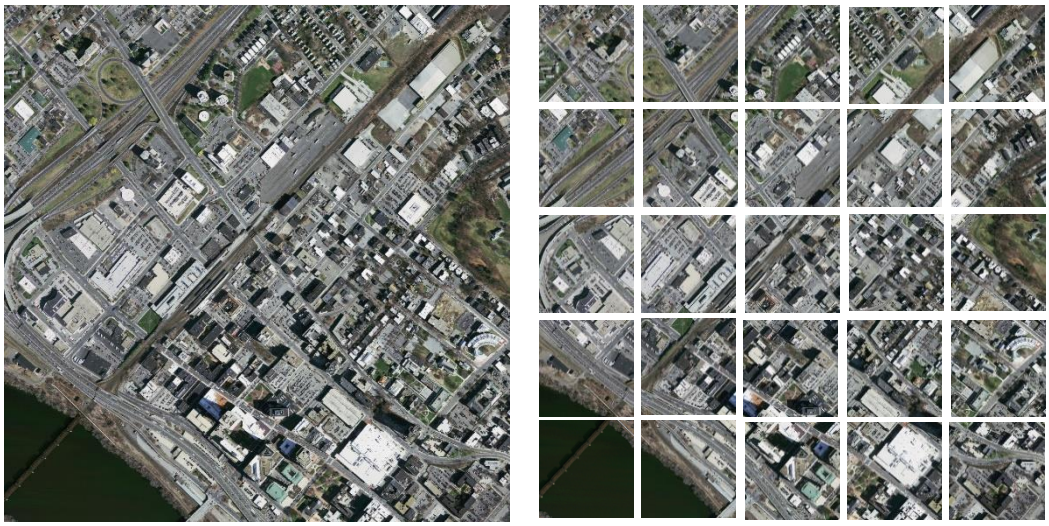
Additionally, the computing time and the network complexity were also taken into account for the comparative approach in order to obtain a model with less complexity.

5. RESULTS AND DISCUSSION

This section is devoted to visualizing and discuss the results of the experimental study. The first two sections display the output of the preprocessing steps and method selections. The subsequent section presents the output obtained during the model development and hyperparameter selection. The results obtained after applying transfer learning and post-processing are described in the last two sections.

5.1 Image Preprocessing

This section summarizes the outputs of the steps described in section 4.2.1. Figure 5.1 shows a sample of original images from the Massachusetts road data set and the corresponding 25 image tiles after the images were cropped to 300 * 300 pixels.

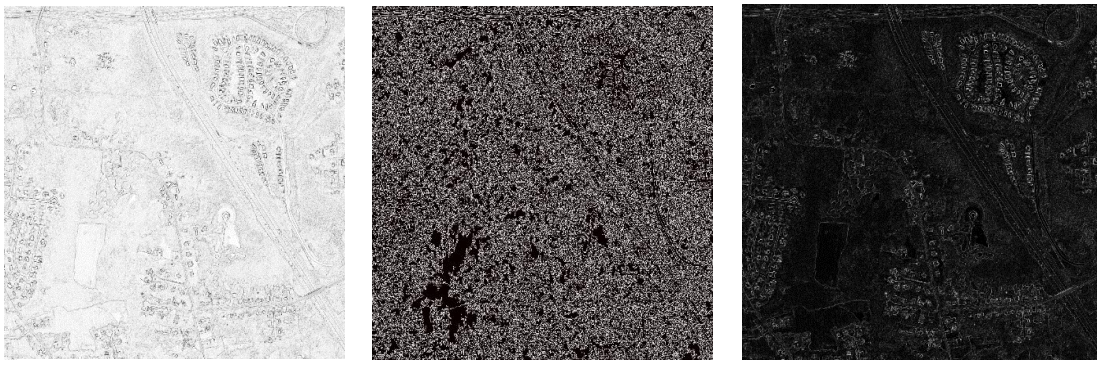


(a) *(b)*

Figure 5.1:One sample image from Massachusetts road data set (a) original image (b) after reducing the image dimensions in to 300*300 pixels

Image cropping resulted in reducing the capacity of the digital images thus allowing to save the memory while executing the codes for training the models. The sample outputs of the edge detection filters and textures are shown in Figure 5.2. Edge detection filters capture the rapid changes and discontinuities in the spectral values of digital images. They are often used in image

classification to enhance the rate of change in the spectral value along the edges of the streets (Sirefelt Rickard, 2004).



Sobel edge detector **Canny edge detector** Prewitt edge detector

Standard deviation **Gaussian filter**

Figure 5.2: Output bands of edge detection filters and textures

The classification models were catalysts using the second-order information bands above. The individual contribution of each band to the final classified output is shown in Table 1. The Prewitt and Sobel edge detection bands mark the classification with the height contribution, and the Canny edge detector contributes the least to the process.

Input band	Individual Contribution
Prewitt edge detection	0.282562
Sobel edge detection	0.282240
Original Image	0.181503
Gaussian filter	0.165455
STD texture	0.081202
Canny edge detection	0.007037

Table 5.1: Feature importance of Random forest classifier

5.2 Method Selection

A sample of 250 images was used to evaluate all four models by providing the same initial states. The outputs of the Random Forest Classifier (RF) for the training and testing images are shown in Figure 5.3.

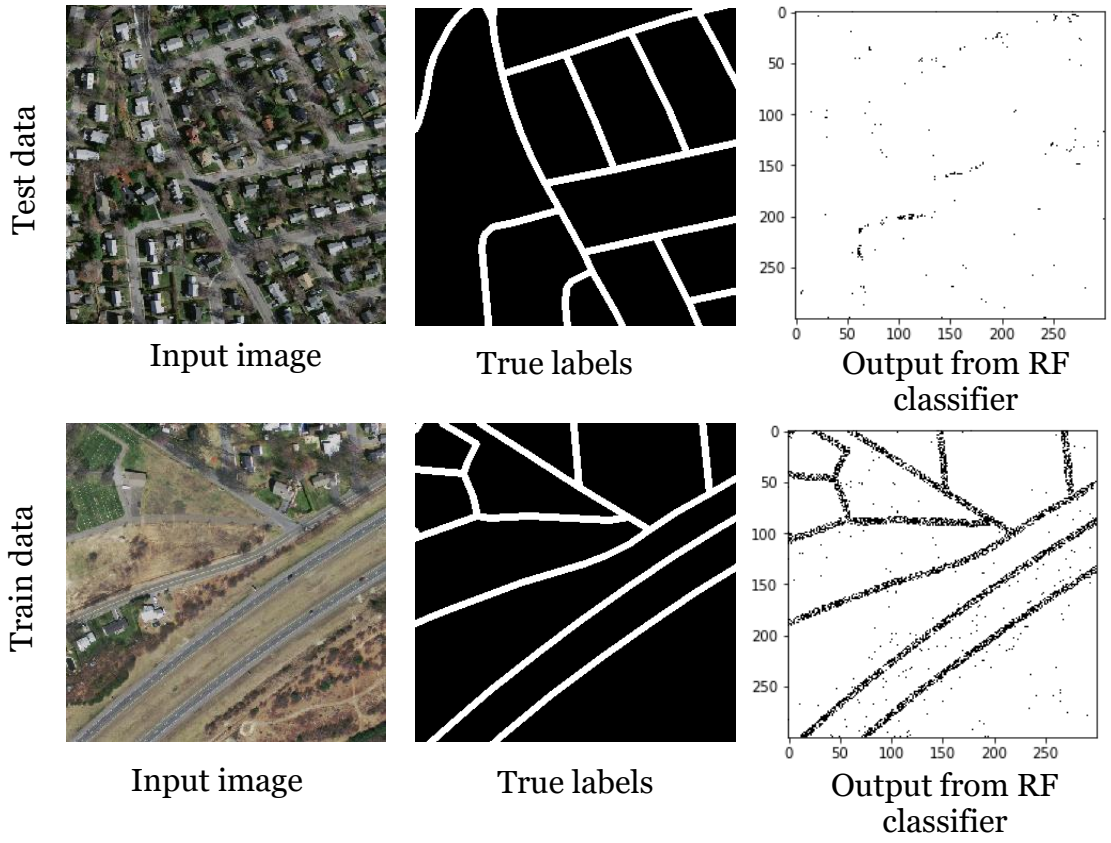


Figure 5.3: A sample input image its corresponding image labels and output from random forest classifier for train and test data sets

The outputs of the Support vector machine classifier (SVM) for the training and testing images are shown in Figure 5.4 and 5.5.

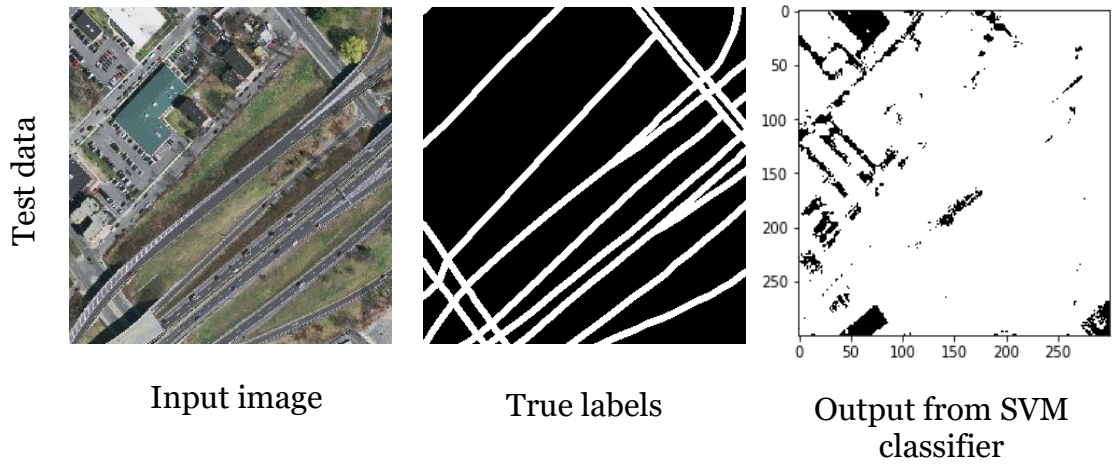


Figure 5.4: A sample input image its corresponding image labels and output from support vector machine classifier for test data set

Figure 5.5: A sample input image its corresponding image labels and output from support vector machine classifier for train data set

	Support vector machine classifier		Random forest classifier	
	Train data	Test data	Train data	Test data
Accuracy	0.53	0.75	0.97	0.88
Precision	0.11	0.20	0.94	0.63
Recall	0.70	0.07	0.62	0.02
F1 score	0.19	0.11	0.75	0.05

Table 5.2: Accuracy score values for support vector machine classifier and random forest classifier.

Based on the accuracy assessment results above, both classifiers are not well suited to the problem. The random forest classifier tends to produce an over-fitted output resulting in very accurate results being produced for the training set and poor performance for the test set. The outputs of the U- Net image segmentation model (U-Net ISM) for the training and testing images are shown in Figures 5.6 and 5.7.

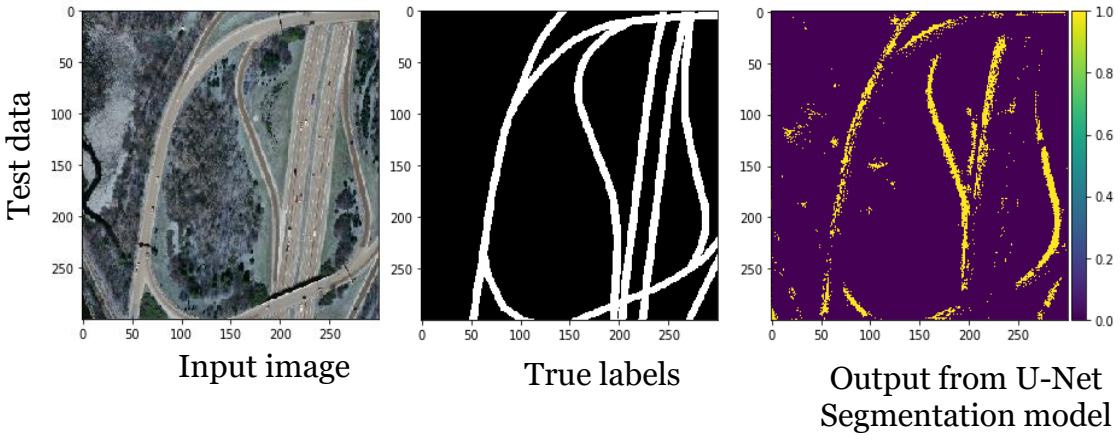


Figure 5.6: A sample input image its corresponding image labels and output from U-Net Image Segmentation model for test data set

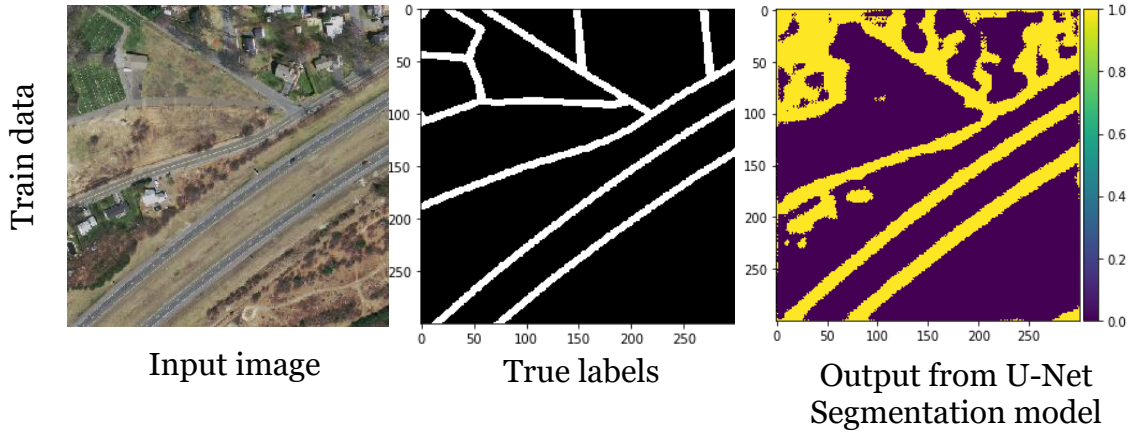


Figure 5.7: A sample input image its corresponding image labels and output from U-Net Image Segmentation model for train data set

The outputs of the Seg- Net image segmentation model for the training and testing images are shown in Figure 5.8.

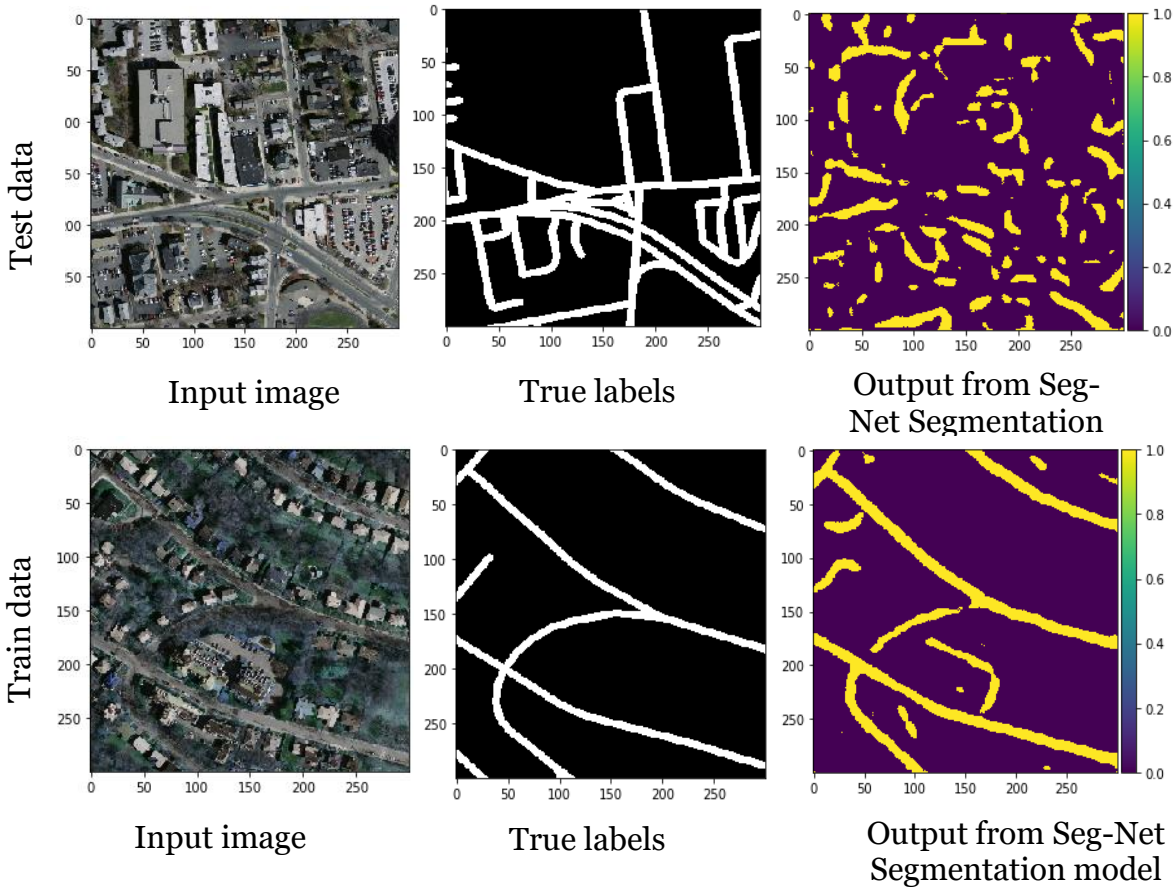


Figure 5.8: A sample input image its corresponding image labels and output from Seg-Net Image Segmentation model for test and Train data sets

	The U-Net image segmentation model		The seg-Net image segmentation model	
	Train data	Test data	Train data	Test data
Accuracy	0.7440	0.8677	0.8368	0.7185
Precision	0.3336	0.7819	0.2364	0.0175
Recall	0.4529	0.3531	0.1742	0.1766
F1 score	0.3842	0.4865	0.2006	0.0749

Table 5.3: Accuracy score values for U-Net and Seg-Net Image Segmentation models

The prediction accuracies for the U-Net image segmentation model were higher than for the Seg-Net image segmentation model and were good enough to comfortably continue the study with the architecture.

5.3 Model development and hyperparameter selection

5.3.1 Implementation

This section presents the results of the model development process performed by changing the size of the convolution, the number of convolution and pooling layers of the U-Net model, as well as the hyperparameters used for training the CNN.

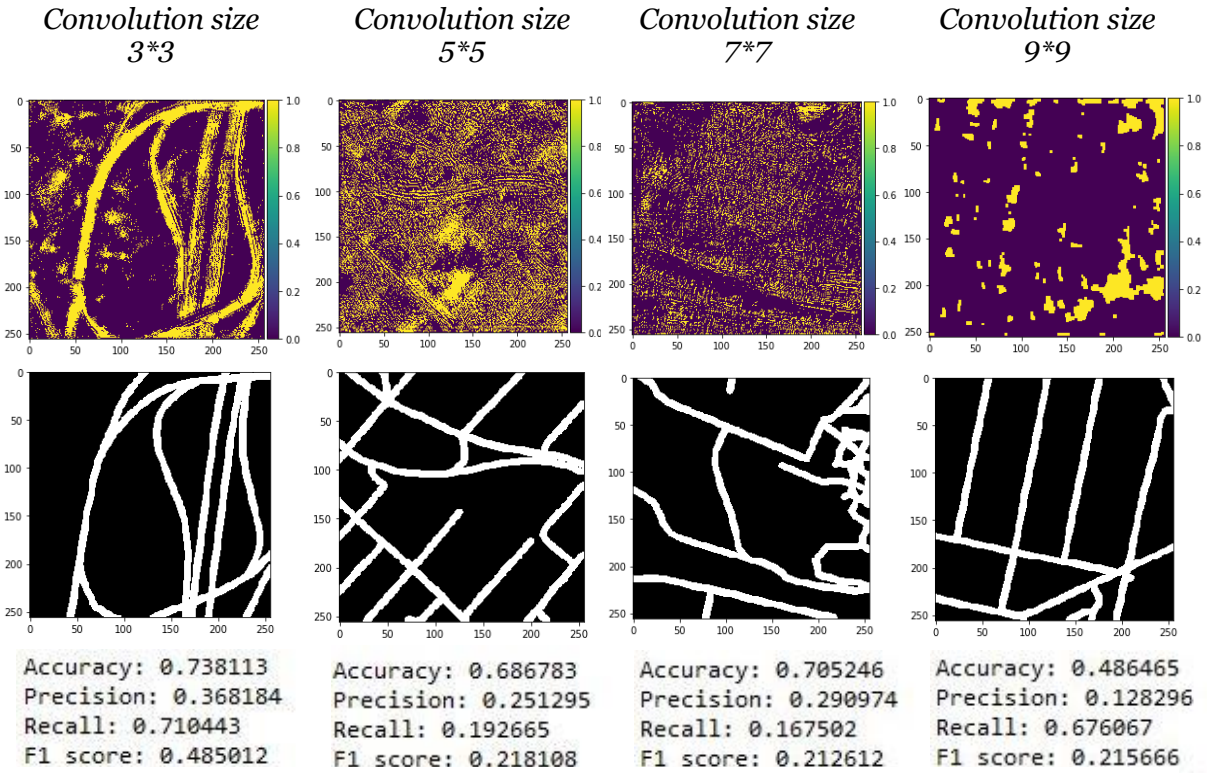


Figure 5.9: Classification performance of the U-Net ISM for different convolution sizes from 3*3, 5*5,7*7, and 9*9

Figure 5. 9 illustrates the classification performance of the U-Net for different convolution sizes from 3*3, 5*5,7*7, and 9*9. With an increase of convolution size from 3 to 9 there is a decrease of F1 score starting from 0.4850 to 0.2156. The highest F1 score of 0.4850 was observed for convolution size 3*3, was selected for the study. The original U-Net architecture (referred to as U-Net (9) in this study) comprised 9 CNN blocks, with a total of 8 blocks for the encoder path and the decoder path (4 blocks each), and the remaining block for the bridge between encoder and decoder. Similarly, U-Net (5) and U-Net (7) consist of 5 and 7 blocks, respectively. The number of CNN blocks indicates the complexity of the model and the time required for training, as the number of parameters in the model varies. The results observed in U-Net (5), (7), (9), and (11) are shown in Figure 5.10.

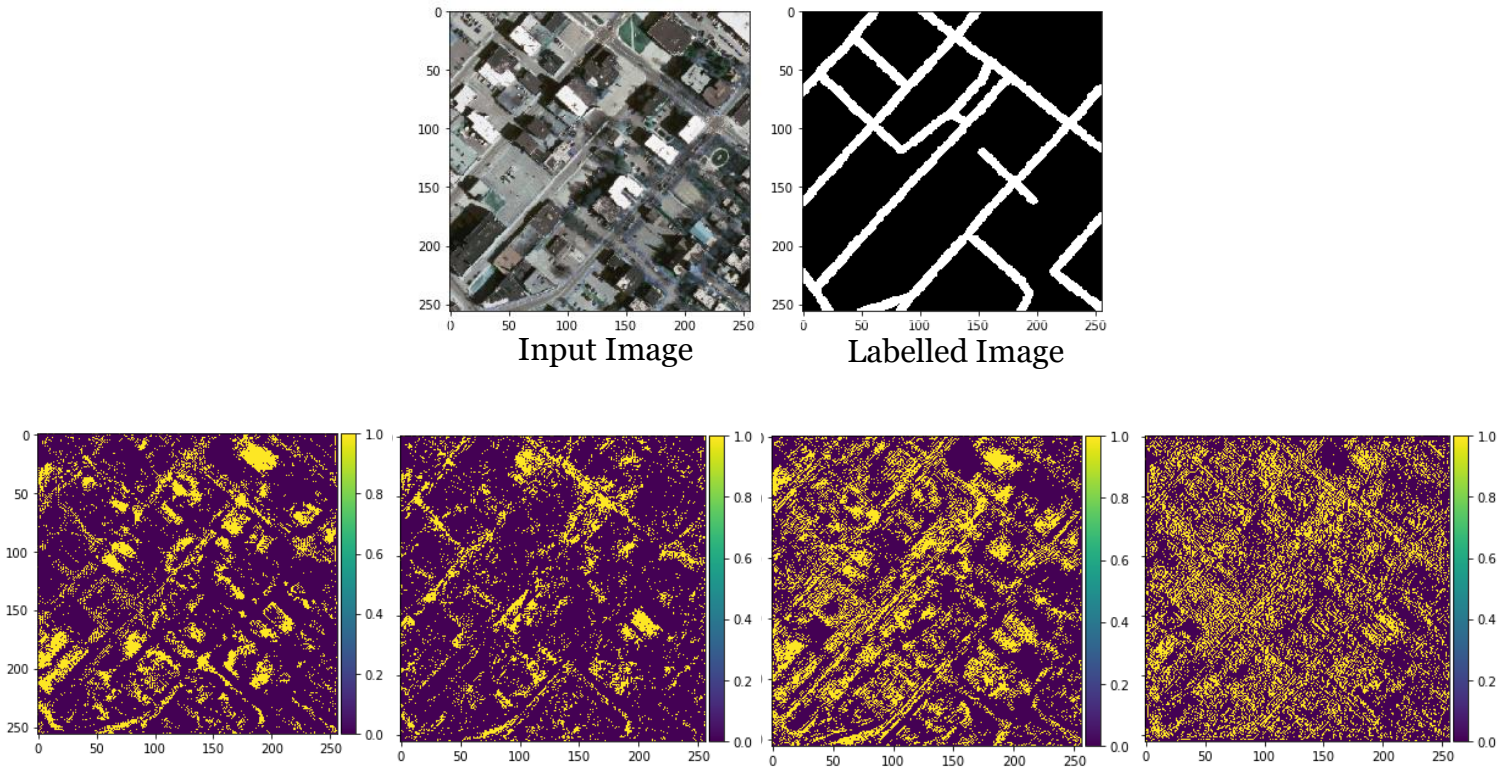


Figure 5.10: Classification performance and computing time of the U-Net ISM for different numbers of CNN blocks

In figure 5.10 it is observable that, as the number of blocks increases, the training time has also increased. The highest performance was observed with U-Net (5). Therefore, the U-Net architecture consisting of 5 blocks, 2 each for encoder and decoder, was selected as the optimized model which balanced complexity and performance.

5.3.2 Hyperparameters

Hyperparameters are the parameters that must be set before the training process, e.g., Learning rate, number of epochs, weight, and bias initializations, etc. (Sirefelt Rickard, 2004). This study used initial hyperparameters as described in Table 5.2 and the selection process was based on the performance of the previous image segmentation studies.

Parameter	Initial State	Remark
Number of Iterations	1000	The process was stopped early when there was slight progress on the validation dataset
Cost function	Binary cross entropy (Boyagoda and Da Silva, 2020)	
Activation function	Rectified Linear Unit (Abderrahim, Abderrahim and Rida, 2020)	
Learning rate	0.0001	Reduced with the iterations (Zhang, Liu and Wang, 2018)
Initial weights	he_normal (Keras)	a random value from a normal distribution centered on 0

Table 5.4: Hyperparameters assigned for CNN training.

5.4 Transfer Learning

The final network architecture is illustrated in figure 5.11, which is consisting of 11 convolution layers, 2 pooling layers, 2 convolution transpose layers, and 2 concatenations.

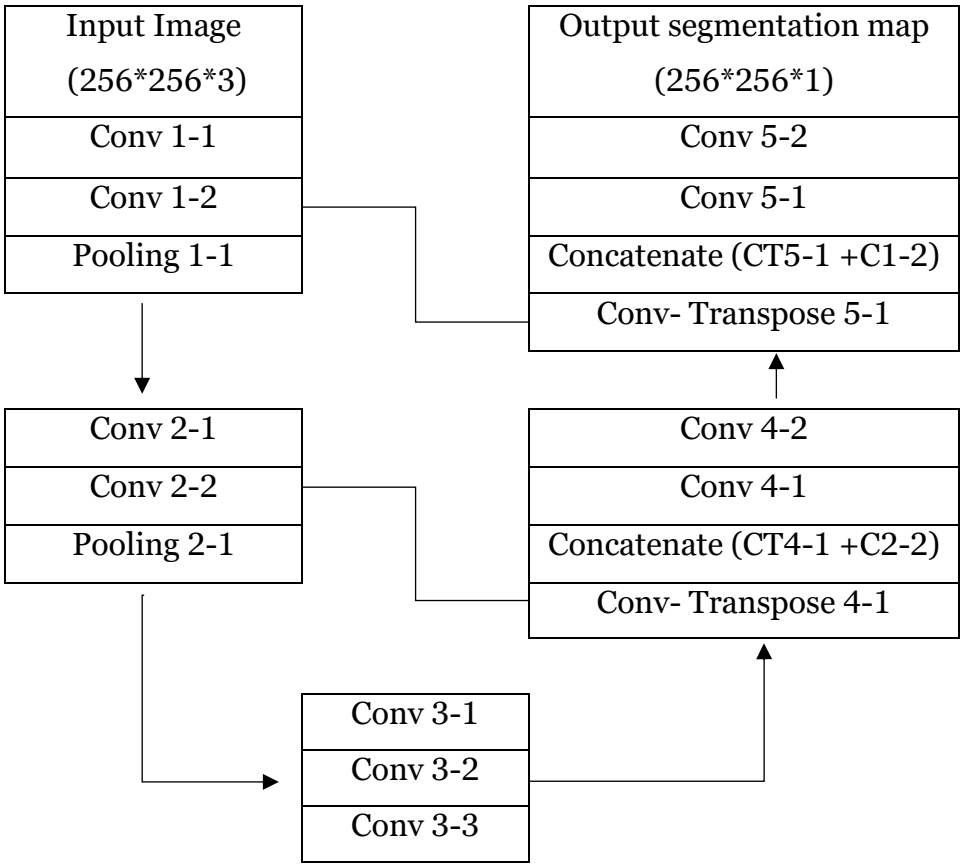


Figure 5.11: Implemented U Net based CNN model for road extraction.

Deep learning models often achieve increased accuracy with a transfer learning approach. Figure 5.12 visualizes the segmented output from the developed network trained by the VGG16 pre-trained feature extractor.

The final proposed model shows clear results with a high F1 score after transfer training, with less noise and breaks in the road segment. However, the second example visualized in the figure shows that the low F1 score is about 0.1241. We conclude that this is due to the inaccuracy of the labeled data in the source image and that the model could recognize the road if it were not in the source data. Also, we noticed few noisy objects as indicated in the red circles, mainly because of building roofs and parking slots having similar spectral properties as road features. Additionally, there were also eroded road segments, as shown in blue rectangles due to the background clutter, such as trees and shadows. However, the quantitative and visual analysis of the segmented output shows a

good improvement in the prediction results, which is further passed through post-processing steps to improve the visual interpretation.

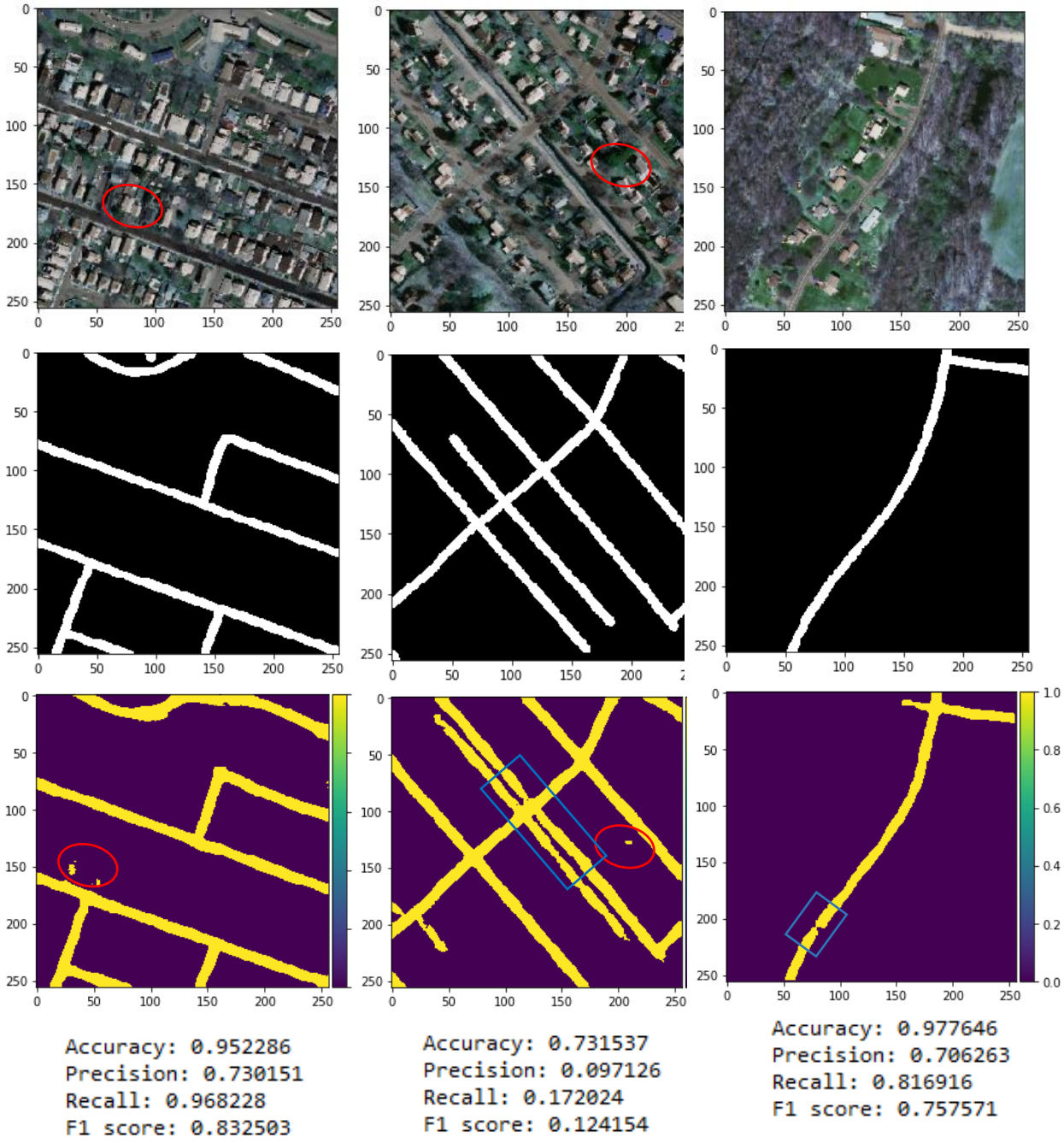
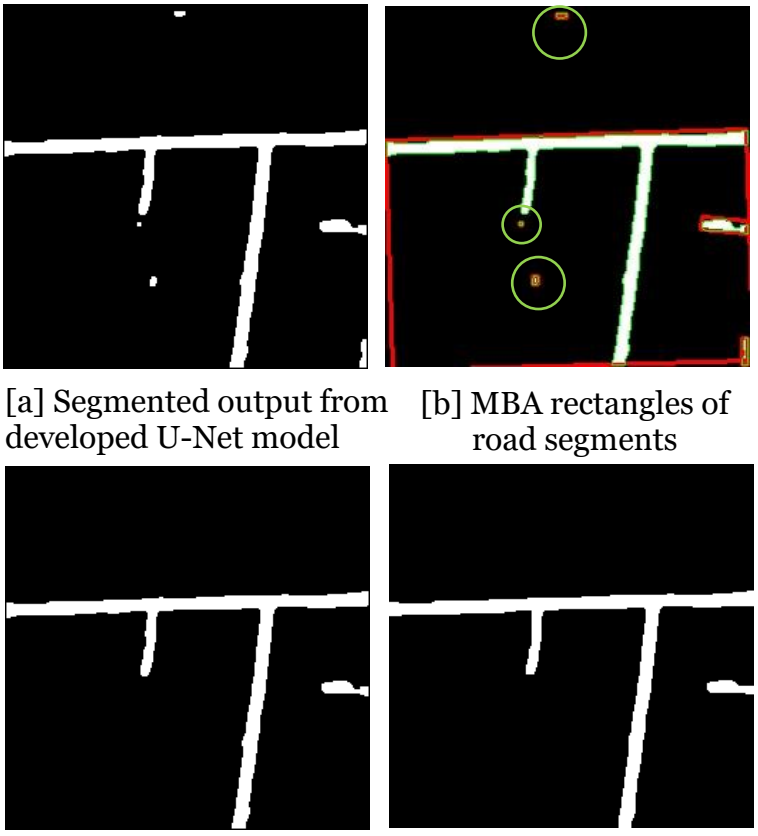


Figure 5.12: Three sample input images with the corresponding image names and outputs from the implemented CNN model after applying the transfer learning

5.5 Post-processing

Minimum Boundary Area Rectangle (MBAR) of the extracted features (see Figure 5.13. [b]) was used to remove the noises indicated with red circles in Figure 5.9. then the extracted shape features, the area, and the aspect ratio were used to eliminate the nonroad segments as shown in figure 5.13.[c]. finally, after applying the morphological opening to fill the eroded segments the final output from the developed U-Net model is visualized in figure 5.13.[d]. the output is very close to its respective known labels and the F1 score of the output was 0. 8320 with 0. 9693 recall. The results discussed in this chapter conclude the work in the next chapter, in which the goals presented in chapter 1 are checked in accordance with our results and the discussion from the experiments and analyses carried out.



[a] Segmented output from developed U-Net model

[b] MBA rectangles of road segments

[c] After removing areas < 10 or Aspect ratio =1

[d] Final output

Figure 5.13: Outputs of post processing steps

6. CONCLUSIONS AND RECOMMENDATIONS

This chapter summarizes the conclusion of the study described in accordance with the goals of the study. Also, it includes the limitations and future recommendations to further improve the results of the study.

6.1 Conclusions

The main aim of the research was to formulate a method of road extraction from high-resolution images that uses a deep learning approach based on a convolutional neural network. For this, First, existing CNN-based image segmentation architectures and classification algorithms were reviewed and two image segmentation architectures as U-Net image segmentation architectures and SegNet Net image segmentation architectures and two image classification algorithms were selected for the study, mainly due to their efficiency and accurate performance from the past studies. The best model among these four options was then selected based on an experimental study conducted on a freely available Massachusetts dataset developed by (Mnih, 2013). Because of the superior results of the U-Net Net image segmentation architectures compared to the other methods, this one was chosen to continue the study by changing the original network architecture and hyperparameters of CNN training. Subsequently, the developed model is further improved by transfer learning using an already existing VGG16 function extractor trained on the ImageNet data set. Finally, the segmented output of the model was enhanced by the post-processing strategies developed using parameters measured on minimally bounded rectangles covering the extracted road segments and morphological operations. The method developed in the study was superior to the elaborated application and was performed with an F1 value of 0.8230 and a recall of 0.9693. In addition, the network architecture was less complex than the original U-Net architecture and the processing time was comparatively shorter than the other three methods used in the study.

The study's minor objectives are discussed below with an explanation.

1. Review and assess the potential of the latest algorithms for automatic road extraction using high-resolution aerial / satellite images.

In chapter 2, the latest algorithms and models developed to extract the road features were discussed. Among the available methods, Deep Convolutional Neural Networks was chosen for the study because of its various advantages: independence from the spatial and contextual features of roads, adaptability to spatial heterogeneity, allowing a large amount of data for the process, and the ability to use data without preprocessing steps. Then some recent improvements of DCNN, their architectures, and the concepts of transfer learning were used for potential improvements.

2. Compare several image classifiers and CNN based segmentation architectures based on accuracies to find out the best method to extract roads using high-resolution aerial/satellite imagery.

Based on the previous studies, two image classification algorithms based on deep learning and two image segmentation architectures were selected for performance comparison through accuracy ratings using F1 score, recall, precision, visual inspection, and computation time taken during the training. Subsection 5.2 shows the results of the performance evaluations where the U Net image segmentation architecture produced the best results for road extraction.

3. Enhance the accuracy of the chosen method by changing the chosen network architecture and applying transfer learning.

To determine the optimal structure of the U-Net image segmentation architecture, various design experiments were carried out, as shown in Section 5.3. The effects of varying the folding size, the number of folding and pooling layers used in the model, and the hyperparameters of CNN training, were experimented within the process. The model is shown in Figure 5.9, which consists of 10 convolutions, 2 pooling operations, 2 convolution transposition layers, and two concatenation layers was implemented in the process. Additionally, transfer learning by the pre-trained feature extractor VGG16 was used to further increase accuracy up to an accuracy of 0.9557. When comparing the results achieved with and without transfer learning, as shown in Figures 5.8 and 5.9, it can be concluded that transfer learning has significantly improved the implemented model. Additionally, this model is much lighter than the original U-Net structure which consists of a total of 23 convolution layers and

it is free in terms of GPU memory requirement in machine learning. In the end, the thesis presented a CNN based deep learning approach that follows the U-Net image segmentation architecture for automatic road extraction.

6.2 Limitations and Recommendations

During the execution of this experiment, some directions for future development emerged, which are summarized below:

1. It is still possible to further improve the network's performance by applying data augmentation steps which are resulting in to increase in the amount of data for the training process.
2. It is recommended to use undistorted comparison matrices such as AUROC (Area Under the Receiver Operating Characteristic Curve) in the statistics because of the unbalanced ratios of the pixels belong to the road and non-road features.
3. The current study assigns the hyperparameters that ensure satisfactory performance in previous studies. However, an investigation should be conducted to optimize the hyperparameters for the CNN training as done in (Shrestha and Vanneschi, 2018).
4. Many cases have been observed where the freely downloaded road labels did not exactly match the high-resolution color images, resulting in a decrease in the accuracy of the test data and training the model with unsuitable features that do not represent roads. Therefore, it is recommended to perform a visual inspection and possible corrections to the label images before using them for the application.
5. Explorations of new post-processing methods are also recommended for future works.

7. BIBLIOGRAPHIC REFERENCES

Abderrahim, N. Y. Q., Abderrahim, S. and Rida, A. (2020) 'Road segmentation using u-net architecture', *Proceedings - 2020 IEEE International Conference of Moroccan Geomatics, MORGEO 2020*, pp. 0–3. doi: 10.1109/Morgeo49228.2020.9121887.

Alshaikhli, T., Liu, W. and Maruyama, Y. (2019) 'Automated method of road extraction from aerial images using a deep convolutional neural network', *Applied Sciences (Switzerland)*, 9(22). doi: 10.3390/app9224825.

Alshehhi, R. *et al.* (2017) 'Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks', *ISPRS Journal of Photogrammetry and Remote Sensing*, 130, pp. 139–149. doi: 10.1016/j.isprsjprs.2017.05.002.

Ayo, B. and Da Silva, J. (2020) *Integrating Openstreetmap Data and Sentinel-2 Imagery for Classifying and Monitoring Informal Settlements*. New university of Lisbon. Available at: https://run.unl.pt/bitstream/10362/93641/1/TGEO0221.pdf.

Badrinarayanan, V., Kendall, A. and Cipolla, R. (2017) 'SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), pp. 2481–2495. doi: 10.1109/TPAMI.2016.2644615.

Bakhtiari, H. R. R., Abdollahi, A. and Rezaeian, H. (2017) 'Semi automatic road extraction from digital images', *Egyptian Journal of Remote Sensing and Space Science*. National Authority for Remote Sensing and Space Sciences, 20(1), pp. 117–123. doi: 10.1016/j.ejrs.2017.03.001.

Barrile, V. and Bilotta, G. (2016) 'Fast Extraction of Roads for Emergencies with Segmentation of Satellite Imagery', *Procedia - Social and Behavioral Sciences*, 223, pp. 903–908. doi: 10.1016/j.sbspro.2016.05.313.

Boyagoda, E. M. R. C. L. . and Da Silva, J. (2020) *Object Detection for Single Tree Species*. New university of Lisbon. Available at: http://hdl.handle.net/10362/93643.

Buslaev, A. *et al.* (2018) 'Fully convolutional network for automatic road extraction from

satellite imagery', *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018-June, pp. 197–200. doi: 10.1109/CVPRW.2018.00035.

F. F. Ahmadi, M.J.V. Zoej, H. Ebadi, M. M. (2008) 'Road Extraction from HIgh resolution Satellite Images Usind Image Processing Algorithms and CAD- Based Environmnets facilities', *Jornal of Applied Sciences 8*, 17(2008), pp. 2975–2982.

Hormese, J. and Saravanan, C. (2016) 'Automated Road Extraction From High Resolution Satellite Images', *Procedia Technology*, 24, pp. 1460–1467. doi: 10.1016/j.protcy.2016.05.180.

Huang, Z., Pan, Z. and Lei, B. (2017) 'Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data', *Remote Sensing*, 9(9), pp. 1–21. doi: 10.3390/rs9090907.

Itza Alejandra *et al.* (2020) *Landcover and Crop Type Classification*. University of Nova, Lisbon, Portugal.

Khan, Z. *et al.* (2020) 'Evaluation of deep neural networks for semantic segmentation of prostate in T2W MRI', *Sensors (Switzerland)*, 20(11), pp. 1–17. doi: 10.3390/s20113183.

Lan, M. *et al.* (2020) 'Global context based automatic road segmentation via dilated convolutional neural network', *Information Sciences*, 535, pp. 156–171. doi: 10.1016/j.ins.2020.05.062.

Mahdianpari, M. *et al.* (2018) 'Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery', *Remote Sensing*, 10(7). doi: 10.3390/rs10071119.

Mena, J. B. (2003) 'State of the art on automatic road extraction for GIS update: A novel classification', *Pattern Recognition Letters*, 24(16), pp. 3037–3058. doi: 10.1016/S0167-8655(03)00164-8.

Mnih, V. (2013) 'Machine Learning for Aerial Image Labeling', *PhD Thesis*, p. 109.

Pasquali, G., Iannelli, G. C. and Dell'Acqua, F. (2019) 'Building footprint extraction from multispectral, spaceborne earth observation datasets using a structurally optimized U-

Net convolutional neural network', *Remote Sensing*, 11(23), pp. 1–20. doi: 10.3390/rs11232803.

Ronneberger, O., Fischer, P. and Brox, T. (2015) 'U-net: Convolutional networks for biomedical image segmentation', *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9351, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

Saifi, M. Y., Singla, J. and Nikita (2020) 'Deep Learning based Framework for Semantic Segmentation of Satellite Images', *Proceedings of the 4th International Conference on Computing Methodologies and Communication, ICCMC 2020*, (June), pp. 369–374. doi: 10.1109/ICCMC48092.2020.ICCMC-00069.

Shrestha, S. and Vanneschi, L. (2018) *IMPROVED FULLY CONVOLUTIONAL NETWORK WITH CONDITIONAL RANDOM FIELD FOR BUILDING EXTRACTION*

Sanjeevan Shrestha IMPROVED FULLY CONVOLUTIONAL NETWORK WITH CONDITIONAL RANDOM FIELD FOR. New university of Lisbon, Portugal.

Simonyan, K. and Zisserman, A. (2015) 'Very deep convolutional networks for large-scale image recognition', *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14.

Singh, P. P. and Garg, R. D. (2013) 'Automatic Road Extraction from High Resolution Satellite Image using Adaptive Global Thresholding and Morphological Operations', *Journal of the Indian Society of Remote Sensing*, 41(3), pp. 631–640. doi: 10.1007/s12524-012-0241-4.

Sirefelt Rickard (2004) *Semi-automatic road extraction from aerial images*. CHALMERS UNIVERSITY OF TECHNOLOGY. doi: 10.1117/12.508365.

Talal, T. M. . *et al.* (2014) 'Road Extraction from High Resolution Satellite Images by Morphological Direction Filtering and Length Filtering Road Extraction from High Resolution Satellite Images by Morphological Direction Filtering and Length Filtering', in *18th International Conference on Computer Theory and Applications*.

Tran, L. A. and Le, M. H. (2019) 'Robust u-net-based road lane markings detection for

autonomous driving', *Proceedings of 2019 International Conference on System Science and Engineering, ICSSE 2019*, (July), pp. 62–66. doi: 10.1109/ICSSE.2019.8823532.

Wang, J. *et al.* (2016) 'A new approach to urban road extraction using high-resolution aerial image', *ISPRS International Journal of Geo-Information*, 5(7), pp. 1–12. doi: 10.3390/ijgi5070114.

Wang, W. *et al.* (2016) 'A review of road extraction from remote sensing images', *Journal of Traffic and Transportation Engineering (English Edition)*, 3(3), pp. 271–282. doi: 10.1016/j.jtte.2016.05.005.

Wijesingha, J. S. J. *et al.* (2012) 'Automatic road feature extraction from high resolution satellite images using LVQ neural networks', *33rd Asian Conference on Remote Sensing 2012, ACRS 2012*, 1(November), pp. 169–175.

Wulamu, A. *et al.* (2019) 'Multiscale Road Extraction in Remote Sensing Images', *Computational Intelligence and Neuroscience*, 2019. doi: 10.1155/2019/2373798.

Xie, M. *et al.* (2016) 'Transfer learning from deep features for remote sensing and poverty mapping', *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 3929–3935.

Zhang, Z., Liu, Q. and Wang, Y. (2018) 'Road Extraction by Deep Residual U-Net', *IEEE Geoscience and Remote Sensing Letters*, 15(5), pp. 749–753. doi: 10.1109/LGRS.2018.2802944.

Masters Program in **Geospatial Technologies**

Supported by:
Education and Culture
ERASMUS MUNDUS