ISTANBUL TECHNICAL UNIVERSITY GRADUATE SCHOOL

AUTOMATIC BUILDING DETECTION USING VERY HIGH RESOLUTION SATELLITE IMAGERY

GEOMATICS ENGINEERING DESIGN II
GRADUATION PROJECT

Mehmet Niyazi KARACALAR

Department of Geomatic Engineering

JUNE 2022

ISTANBUL TECHNICAL UNIVERSITY GRADUATE SCHOOL

AUTOMATIC BUILDING DETECTION USING VERY HIGH RESOLUTION SATELLITE IMAGERY

GEOMATICS ENGINEERING DESIGN II

Mehmet Niyazi KARACALAR
(010170652)

Department of Geomatic Engineering

Thesis Advisor: Prof. Dr. Elif SERTEL

JUNE 2022

ISTANBUL TEKNİK ÜNİVERSİTESİ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK ÇÖZÜNÜRLÜKLÜ UYDU VERİLERİNDEN OTOMATİK BİNA BELİRLEME

TASARIM PROJESİ TEZİ

Mehmet Niyazi KARACALAR
(010170652)

Geomatik Mühendisliği Anabilim Dalı

Tez Danışmanı: Prof. Dr. Elif SERTEL

HAZİRAN 2022

Mehmet Niyazi KARACALAR, Geomatics Engineering Undergraduate Program student of İTU Graduate School student ID 010170652, successfully defended the thesis/dissertation entitled "AUTOMATIC BUILDING DETECTION USING VERY HIGH RESOLUTION SATELLITE IMAGERY", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Elif SERTEL** .....
İstanbul Technical University

Date of Submission : 10 June 2022
Date of Defense : 10 June 2022

To my family,

FOREWORD

We would like to thank our supervisor Prof. Dr. Elif Sertel for their help, contributions, and guidance during the course of the thesis. In addition, I would like to thank my family for their support.

June 2022 Mehmet Niyazi KARACALAR

TABLE OF CONTENTS

ABBREVIATIONS

CNN **:** Convolutional Neural Network
MB **:** Massachusetts Buildings
MSAW **:** Multi-Sensor All Weather
FCN **:** Fully Convolutional Network
SAR **:** Synthetic Aperture Radar
ReLU **:** Rectified Linear Unit
IoU **:** Intersection over union
RGB **:** Red Green Blue
ELU **:** Exponential Linear Unit
GPU **:** Graphical Processing Unit

LIST OF TABLES

Page

LIST OF FIGURES

Page

AUTOMATIC BUILDING DETECTION USING VERY HIGH RESOLUTION SATELLITE IMAGERY

SUMMARY

Building conclusions from remotely sensed images is a difficult and costly undertaking using traditional approaches. Knowing the location of buildings is critical in issues such as population estimation, city planning, and disaster management. The extraction of building inferences by manpower is a time-consuming and repeated job. It is vulnerable to major human-caused mistakes. Diversity in building geometries, variations in reflectance values, and comparable qualities with other objects remain weak with index-based and traditional machine learning algorithms, therefore building inferences are always an issue to address for remote sensing. Deep learning-based techniques, which have recently gained popularity, appear to be promising. Deep learning-based techniques began to be deployed in practice, leaving the theoretical realm, particularly in light of hardware and software improvements. The Unet and Deeplabv3+ architectures are the most often used for determining building inferences in the context of semantic segmentation. The goal of this research is to autonomously recognize buildings by combining Unet and Deeplabv3+ architectures with Resnet50, Resnet101, and Efficient-b1 encoders. The study made use of the Massachusetts building dataset. Six distinct models were trained in total. With U-Net and Deeplabv3+ architectures, Resnet50, Resnet101, and Efficient-b1 encoders, and the same hyperparameters, these models were extended to 80 epochs. The models trained with U-Net perform better; the model with the highest IoU score is 87.42 percent IoU, and it was trained with U-Net using the resnet101 encoder. Again, two additional models were trained using U-Net by experimenting with different encoders, and as a consequence of this training, resnet50 earned an IoU score of 86.66 percent, while the model utilizing the Efficient-b1 encoder achieved an IoU score of 86.65 percent. When we visually assess the model output achieved with the Efficient-b1 encoder, we observe that it is far too poor to deserve this score. This might be due to a mismatch between the model and the encoder. While Deeplabv3+ models performed worse than U-Net, the model trained with the resnet101 encoder earned IoU scores of 86.16 percent, resnet50 86.03 percent, and Efficient-b1 81.94 percent.

YÜKSEK ÇÖZÜNÜRLÜKLÜ UYDU VERİLERİNDEN OTOMATİK BİNA BELİRLEME

ÖZET

Uzaktan algılanmış görüntülerden sonuçlar elde etmek, geleneksel yaklaşımları kullanarak zor ve maliyetli bir iştir. Nüfus tahmini, şehir planlaması ve afet yönetimi gibi konularda binaların yerlerinin bilinmesi kritik öneme sahiptir. İnsan gücü ile bina çıkarımlarının çıkarılması zaman alan ve tekrarlanan bir iştir. İnsan kaynaklı büyük hatalara açıktır. İndeks tabanlı ve geleneksel makine öğrenimi algoritmalarında bina geometrilerindeki çeşitlilik, yansıma değerlerindeki farklılıklar ve diğer nesnelerle karşılaştırılabilir nitelikler zayıf kalır, bu nedenle bina çıkarımları her zaman uzaktan algılama için ele alınması gereken bir konudur. Son zamanlarda popülerlik kazanan derin öğrenme temelli teknikler umut verici görünmektedir. Derin öğrenmeye dayalı teknikler, özellikle donanım ve yazılım iyileştirmeleri ışığında teorik alanı terk ederek pratikte uygulanmaya başlandı. Unet ve Deeplabv3+ mimarileri, anlamsal bölümleme bağlamında bina çıkarımlarını belirlemek için en sık kullanılanlardır. Bu araştırmanın amacı, Unet ve Deeplabv3+ mimarilerini Resnet50, Resnet101 ve Efficient-b1 kodlayıcılarla birleştirerek binaları bağımsız olarak tanımaktır. Çalışma, Massachusetts bina veri setini kullandı. Toplamda altı farklı model eğitilmiştir. U-Net ve Deeplabv3+ mimarileri, Resnet50, Resnet101 ve Efficient-b1 kodlayıcıları ve aynı hiperparametreler ile bu modeller 80 çağa kadar genişletildi. U-Net ile eğitilen modeller daha iyi performans gösterir; En yüksek IoU puanına sahip model yüzde 87,42 IoU elde edilmiştir ve resnet101 kodlayıcı kullanılarak U-Net ile eğitilmiştir. Yine U-Net kullanılarak farklı kodlayıcılarla denemeler yapılarak iki ek model eğitilmiş ve bu eğitim sonucunda resnet50 yüzde 86,66 IoU puanı alırken, Efficient-b1 kodlayıcı kullanan model yüzde 86,65 IoU puanı elde etmiştir. . Efficient-b1 kodlayıcı ile elde edilen model çıktısını görsel olarak değerlendirdiğimizde, bu puanı hak etmek için çok zayıf olduğunu gözlemliyoruz. Bunun nedeni, model ve kodlayıcı arasındaki uyumsuzluk olabilir. Deeplabv3+ modelleri U-Net'ten daha kötü performans gösterirken, resnet101 kodlayıcı ile eğitilen model yüzde 86,16, resnet50 yüzde 86,03 ve Efficient-b1 yüzde 81,94 IoU skorlarıları elde edildi.

1. INTRODUCTION

The extraction of precise building objects from remote sensing data has become a significant issue that has gotten a lot of attention in recent years. Building information is critical in a lot of geospatial applications, like urban planning, natural disaster risk, and damage assessment, 3D city modeling, and environmental sciences. Developing imaging technologies and the increase in the resolution of satellite images force engineers to get the maximum efficiency from these images (Nahhas et al.,2018). As a result, new methods have to be applied in engineering studies. Deep learning models represent the last learning paradigm in machine learning and artificial intelligence (Emmert-Streib et al, 2020). Deep learning allows computational models with several processing layers to learn various degrees of abstraction for data representations. (LeCun et al., 2015). Because of factors such as irregular surfaces, diverse geometric forms, and reflectivity values, the partly or completely automatic process for developing determination techniques that have been used in the past have now given way to deep learning techniques. (K. Rastogi et al.,2020). Therefore, using deep learning will be a useful choice when performing automatic building detection with high-resolution satellite images. We should not only concentrate on classifying various images to acquire a thorough comprehension of them but also try to correctly estimate the ideas and locations of things contained in each image. (Zhao et al, 2019). Object detection has gained a lot of academic attention in recent years because of its tight association with video analysis and image understanding. (Zhao et al, 2019). With the fast advancement of deep learning, more powerful tools that can learn semantic, high-level, and deeper features are being offered to solve the issues that affect traditional systems. (Zhao et al, 2019). Within the scope of the design project, it is aimed to make detection of buildings with deep learning methods by using image segmentation with remotely sensed high-resolution satellite images.

Ayala et al (2021) used Sentinel-1 and Sentinel-2 data (at 10 m) together with OpenStreetMap to train deep learning models for building and road detection at 2.5 m

resolution. OpenStreetMap vector data can be rasterized to obtain the necessary resolution which makes this process possible. So, a few basic yet effective adjustments to the U-Net architecture they suggested in order to understand how to improve the resolution of the output masks while simultaneously semantically segmenting the input picture. As an outcome, produced mappings quadruplicate the input spatial resolution, narrowing the gap in building and road detection between satellite and aerial data. In their study, when only Sentinel-1 satellite data is used, 0.2376 IoU score for roads and 0.4704 IoU score for buildings are obtained. Fusion of Sentinel-1 and Sentinel-2 gives better results like 0.5549 IoU for buildings and 0.4415 IoU for roads.

An improved boundary-aware perceptual (BP) loss is proposed in Zhang et al. (2020) research to improve CNN models' building extraction ability. They use WHU aerial dataset and the INRIA dataset to verify the effectiveness and efficiency of the suggested BP loss. They found significant gains in performance in two samples of CNN architectures: PSPNet and UNet, which are extensively employed for pixel-wise labeling tasks. On the WHU aerial dataset and the INRIA dataset, UNet with ResNet101 obtains 90.78 percent and 76.62 percent IoU scores with BP loss, respectively, which are 1.47 percent and 1.04 percent higher than those only trained with the cross-entropy loss function.

Experiments by Wu et al (2018) on a very high-resolution aerial image dataset with approximately 17,000 structures across 18 km2 show that their technique works well in the building segmentation task. The suggested MC–FCN technique beats the traditional FCN method as well as the adaptive boosting method based on features obtained from the histogram of oriented gradients. MC–FCN improves the Jaccard index and kappa coefficient by 3.2 percent (0.833 vs. 0.807) and 2.2 percent (0.893 vs. 0.874) compared to the state-of-the-art U–Net model with just a 1.8 percent rise in model-training time. Furthermore, the sensitivity analysis reveals that limitations at different places have varying effects on the MC – FCN's performance.

In their paper, Liu et al. (2019) propose a new network called Deep Encoding Network (DE-Net) that is developed for the challenge and is based on several recently

introduced techniques in picture segmentation. DE-Net is made up of four modules: inception-style downsampling modules with a striding convolution layer and a max-pooling layer, encoding modules with six sequential residual blocks and a scaled exponential linear unit (SELU) activation function, compressing modules that reduce feature channels, and a densely upsampling module that allows the network to encode spatial information inside feature maps. Without pre-training, DE-Net achieves state-of-the-art recall, F1-Score, and IoU metrics on the WHU Building Dataset. In its own-built Suzhou Satellite Building Dataset, it also outperformed other segmentation networks. The results of the experiments show that DE-Net is effective in extracting buildings from aerial and satellite data. It also implies that, given adequate training data, creating and training a network from the ground up may be preferable to fine-tuning models pre-trained on datasets unrelated to building extraction.

Shrestha et al (2018) present a methodology for selecting a framework that blends high accuracy with minimal network complexity as one of the project's main goals. The exponential linear unit (ELU), a contemporary activation function, is used to improve the performance of the fully convolutional network (FCN), providing more accurate building prediction. Post-processing conditional random fields (CRFs) are added at the end of the adopted convolutional neural network (CNN) architecture to minimize noise (falsely identified buildings) and sharpen the borders of the buildings. The tests were carried out using aerial imagery of Massachusetts buildings. In terms of performance measurements such as the F1-score and IoU measure, their suggested framework surpassed the fully convolutional network (FCN), which is the existing baseline framework for semantic segmentation. In terms of performance evaluations and network complexity, the suggested technique outscored a pre-existing classifier for constructing extraction using the same dataset.

JointNet, a unique neural network, is one way they suggest for meeting extraction needs for both roads and buildings. The suggested approach contributes to road and building extraction in three ways: it can extract huge items with a huge receptive field, in addition, to accurately extracting small objects. The network can successfully extract a diversity of ground objects, from road centerlines to large-scale buildings, by adjusting the loss function. This network module combines dense connectivity with

atrous convolution layers, keeping the dense connectivity pattern's efficiency while achieving a broad receptive field. To enhance road extraction, the suggested technique employs the focus loss function. The offered method is designed to function for both road and building extraction. Experimental results on three datasets verified the effectiveness of JointNet in information extraction of road and building objects (Zhang et. al., 2019).

In the study by Sariturk et al. (2020), the performance of two distinct convolutional neural network designs for object segmentation and classification on pictures (SegNet and Fully Convolutional Networks (FCN)) was compared. These two models were trained using the same training dataset and their performance was validated with the same test dataset. The results demonstrate that while there is no significant difference in accuracy between these two designs for creating segmentation, the FCN architecture is 1 percent more effective than SegNet. This condition, however, may change depending on the dataset utilized in the system's training.

Yang et al. (2020) offer the EANet, a novel end-to-end edge-aware network, and an edge-aware loss for obtaining correct buildings from aerial images in this paper. The architecture is made up of image segmentation networks and edge perception networks, which are in charge of building prediction and edge investigation, respectively. The Wuhan University (WHU) building benchmark and the International Society for Photogrammetry and Remote Sensing (ISPRS) Potsdam segmentation benchmark were used to analyze their method, which were discovered to achieve 90.19 percent and 93.33 percent intersection-over-union and best performance, respectively, without using additional datasets, data augmentation, or post-processing. The EANet is successful at extracting buildings from aerial images, showing that image segmentation quality may be enhanced by paying attention to edge details.

Using a multi-feature convolutional neural network (MFCNN) and morphological filtering, Xie et al (2020) offer a method for extracting buildings from high-

resolution images. There are two phases to their process. To produce pixel-level segmentation of the buildings, the MFCNN is utilized first, which consists of the residual connected unit, dilated perception unit, and pyramid aggregation unit. Second, morphological filtering is utilized to refine building boundaries, optimize building boundaries, and increase boundary regularity. They use datasets from Massachusetts and Inria that are chosen for experimental study. The proposed MFCNN's extraction results are compared to the results of existing deep learning models that have been frequently applied in recent years: FCN-8s, SegNet, and U-Net, all under identical experimental settings. The suggested model improves the F1-score by 3.31 percent –5.99 percent, the OA by 1.85 percent –3.07 percent, and IoU by 3.47 percent –8.82 percent on both datasets, according to with results. The proposed method is effective in extracting buildings from complicated situations, as seen by their results.

2. METHODOLOGY

2.1. Convolutional Neural Networks

The most extensively used deep learning architecture in computer vision is the Convolutional Neural Network. Using this network architecture, deep learning challenges may be realized over images. A traditional CNN design with convolution layers, pooling layers, and fully connected layers (Yamashita, Nishio, Do, & Togashi, 2018). Simply a CNN structure is as shown in figure 2.1.

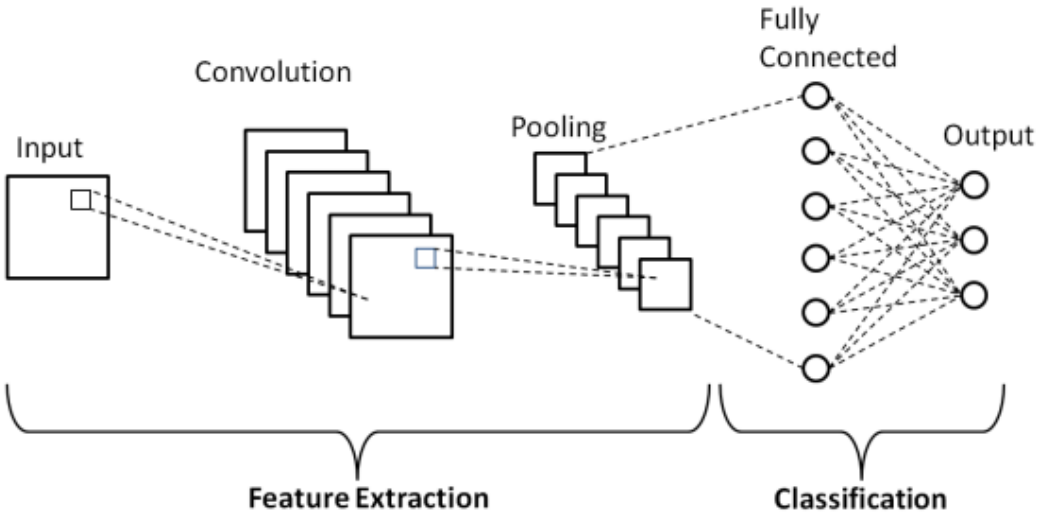

Figure 2. 1: Structure of convolutional neural networks (Phung, 2019)

The convolution layer is the process of emphasizing the desired features by applying filters of fxf size to the input image. Pooling is a sampling procedure that comes after the convolution layer and is used to reduce parameters. Its primary function is to lower the complexity of the successive layers by downsampling (Albawi, Mohammed, & Alzawi, 2017). The full link layer is frequently employed at the network's end to classify the results of the convolution and co-location layers. Convolution and jointing results are opened as a single vector, and each input is connected to neurons in the full link layers to be output to learning the weights (Yamashita et al. 2018). Figure 2.2 shows which hyperparameters are dealing with in which layer.

	Parameters	Hyperparameters
Convolution layer	Kernels	Kernel size, number of kernels, stride, padding, activation function
Pooling layer	None	Pooling method, filter size, stride, padding
Fully connected layer	Weights	Number of weights, activation function
Others		Model architecture, optimizer, learning rate, loss function, mini-batch size, epochs, regularization, weight initialization, dataset splitting

Figure 2. 2: A list of parameters and hyperparameters in a CNN (Yamashita et al., 2018).

2.1.1 Activation Functions

In multilayer neural networks, activation functions are employed for nonlinear transformation operations. The output of hidden layers is normalized using various activation functions in order to compute gradient descent in hidden layers. The output is transformed to a nonlinear value after matrix multiplication and the weight of neurons is calculated in linear functions in hidden layers. The most used activation functions are Sigmoid, Rectified Linear Units (ReLU), Leaky ReLU.

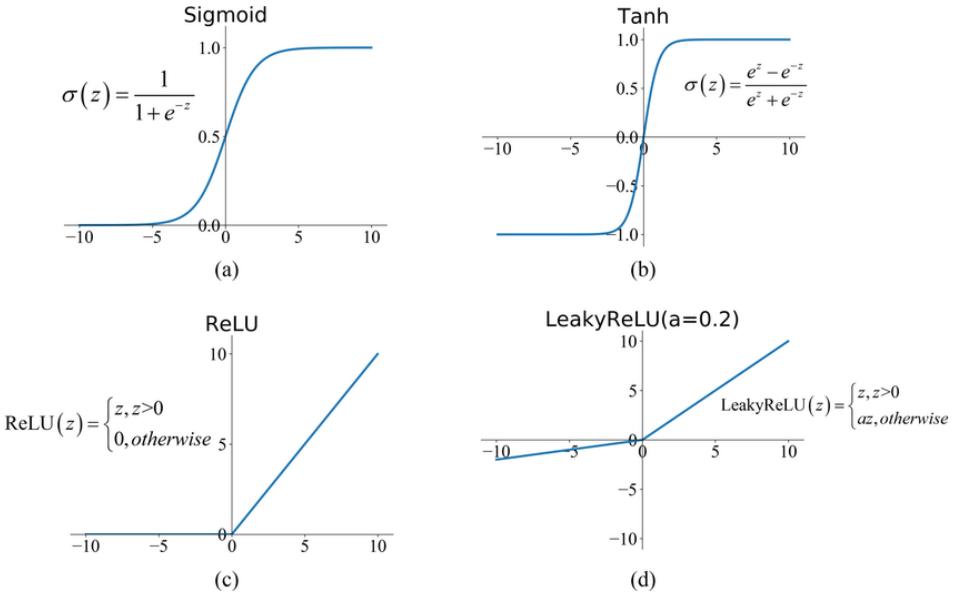

Figure 2. 3: Activation Functions which are commonly use (a) Sigmoid Function; (b) Tanh Function; (c) ReLu Function; (d) Leaky ReLu Function (Feng et al., 2019).

Looking at the linear unit functions in Figure 2.3, it is clear that each one attempts to tackle a distinct problem. They are all employed to prevent linearity and learning, and their distinguishability is their most significant attribute. However, ReLU always returns zero for negative values. Leakage ReLU has been presented as a partial solution. The ELU (Exponential Linear Unit) activation function is distinguished by the fact that it is differentiable for all values. Regardless matter whether technique is used, the extraction of data and desired qualities is critical in the selection of the activation function (Yamashita et al. 2018).

2.1.2 Loss Function

Loss functions compute the value difference between the values produced via estimating and the real values (Yamashita et al., 2018). This difference should ideally be minor (or even zero). As a result of the difference computed by the loss functions, the model may be optimized and improved. Therefore, loss functions are an important step that should be selected according to the model and data. The dice and Jaccard loss functions are two of the most common loss functions utilized in picture segmentation research (Bertels et al., 2019). Also, Bertels et al. (2015) and other researchers have demonstrated the importance of loss functions in pixel-based classification by testing L1 loss function, Jaccard loss function, Dice loss function, and Lovasz loss function on some medical segmentation datasets using different loss functions on UNet architecture (Zhang et al., 2020). Some researchers have constructed boundary-sensitive loss-based networks to improve the performance of CNN models over boundary domains in order to highlight the influence of a boundary-sensitive loss function on performance. Dice and Jaccard loss functions do not improve for border regions because they only evaluate pixel differences and neglect structural information disparities (Zhang et al., 2020).

2.1.3 Data Augmentation

Deep learning requires a large amount of training data in order to achieve effective learning. Obtaining high resolution images is challenging and expensive, especially when working with remote sensing images. By altering existing data, it is feasible to enhance the training set data and avoid issues like overfitting. It has been determined that the transformation in satellite and aerial photographs preserves shape, particularly when applied to things where distortions should be avoided, such as buildings (Buslaev et al., 2018). Cropping, rotations, reflections, and scaling are data augmentation procedures used on satellite and aerial photographs.

2.1.4 Evaluation Metrics

Evaluation metrics are required to compare outcomes and create relevant conclusions. The most commonly used metrics in testing models in this context are Accuracy, Recall, Precision, F1 score, and IoU metrics. True positive (TP), False positive (FP), True negative (TN), and False negative (FN) are used to calculate these values.
True positives (TP): Predicted positive and are actually positive.
False positives (FP): Predicted positive and are actually negative.
True negatives (TN): Predicted negative and are actually negative.
False negatives (FN): Predicted negative and are actually positive.
Accuracy: The percentage of correct predictions for the test data is known as accuracy. It is simple to compute simply dividing the number of right guesses by the total number of forecasts.

Accuracy = Correcty Predict / All Predictions

Precision: It is the percentage of the ratio of all achieved results to correctly achieved results.

Precision = TP / (TP+FP)

Recall: It is a measure of how many of the results that should be predicted as positive are actually detected as positive.

Recall = TP / (TP+FN)

F1 Score: It is the harmonic mean of sensitivity and sensitivity.

F1 Score = 2 x Precision x Recall / (Precision+Recall)

IoU: The ratio of the intersection and union of the predicted and exact values.

IoU = TP / (TP+FP+FN)

2.1.5 Semantic Segmentations

A semantic segmentation classifies every pixel in an image, resulting in a class-separated picture. Semantic segmentation is the process of grouping classes on an image that pertain to the same item (Thoma, 2016). In terms of semantic segmentation, different occurrences of the same item on the picture are identical. For example, if the image contains three similar items, the pixels of these three objects are assigned the same class name. Semantic segmentation is the process of applying semantic tags to each pixel in a picture (Papandreou et al., 2015). Figure 2.4 below is an example of Semantic Segmentation

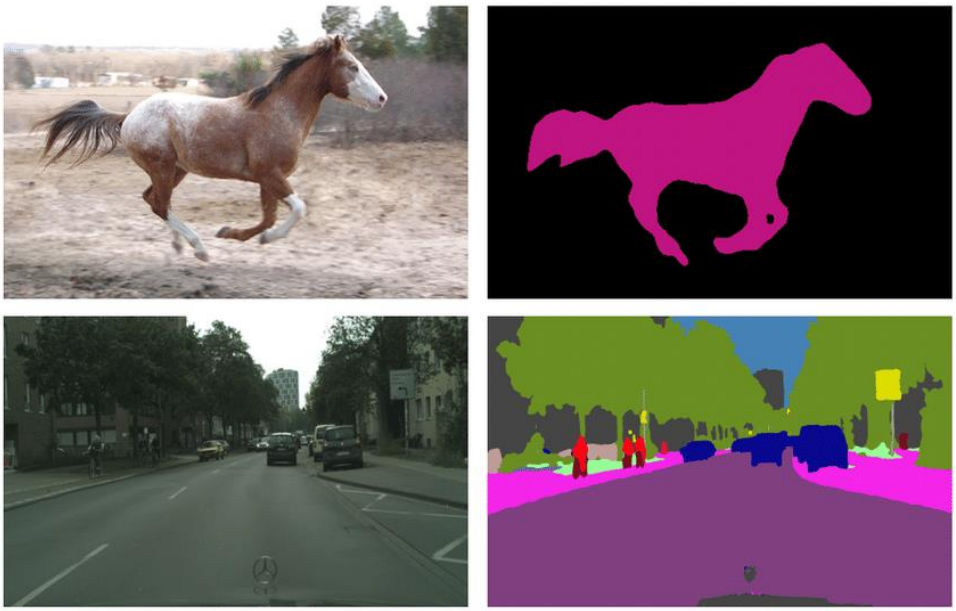

Figure 2. 4: Exaample of Semantic Segmantation (Artacho et al. , 2019).

2.1.6 Number of Epoch

Each feature in the data collection is processed once using model parameters throughout an epoch. This procedure ensures that the mistake is reduced and that it is performed several times. Deep learning calculates the best weight values to solve a problem step by step. As a result, performance will be low in the initial epoch and will improve as the number of epochs grows. However, at a certain stage, our model's learning pace will slow. Furthermore, training a model might take a long period; some models require training for days or months. This is obvious given that the most

fundamental distinction between deep learning and machine learning is the quantity of data used. As a result, extra hyper factors are used to try to shorten the training process.

2.2 CNN Models

Many CNN models, such as U-Net, DeepLabv3+, SegNet, Residual Network (Res-Net), and Fully Convolutional Network, are utilized in applications such as object extraction with semantic segmentation from satellite pictures (FCN). Deeplabv3+ and U-Net models were used within the scope of the study.

2.2.1 U-Net

The U-Net architecture is a semantic segmentation architecture published by Ronneberger et al. in an article for biomedical purposes in 2015. U-Net is commonly used in projects involving pixel-based segmentation. One of the most important features of this network structure is to prevent max-pooling, which disrupts precise positioning. The architecture is divided into two stages. The encoder stage, like in standard convolutional neural networks, comprises of convolution and max pooling layers (CNN). This layer has three 3x3 convolution layers, two corrected linear unit (ReLu) activation functions, and two max-pooling layers with two 2x2 sized strides for down sampling. At each downsampling, the feature channels are doubled. The decoder stage, which utilizes transposed convolutions for exact placement, is the second step. Up-sampling is accomplished by the use of transposed convolutions. Each convolution layer is 3x3 in size, and it is followed by the ReLU activation function. This is known as an end-to-end fully convolutional network, and it contains no density layers and just convolutional layers, allowing it to take any size image as input.

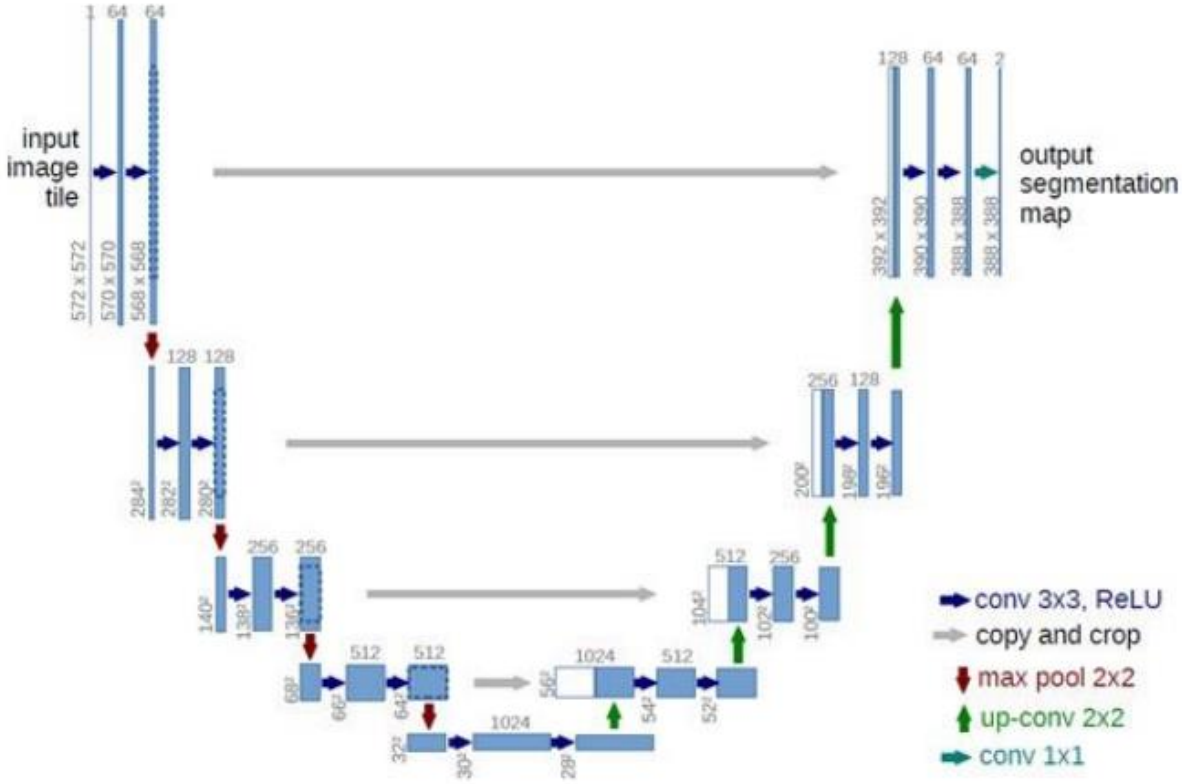

Figure 2. 5: U-net Architecture (Ronneberger, 2015).

2.2.2 DeepLabv3+

DeepLab is a deep learning model for segmenting semantic images. A ResNet-101 design with a Spatial Pyramid Pooling Module is included in the DeepLabv3 architecture. The encoder component of the structure generates multi-scale information by employing atrous convolution on various portions (Chen et al., 2018). Deeplabv3 makes advantage of Atrous convolution is a useful method for increasing the explicit size of things. Deep convolutional neural networks calculate it in order to recognize multi-scale output information (Chen et al., 2018).

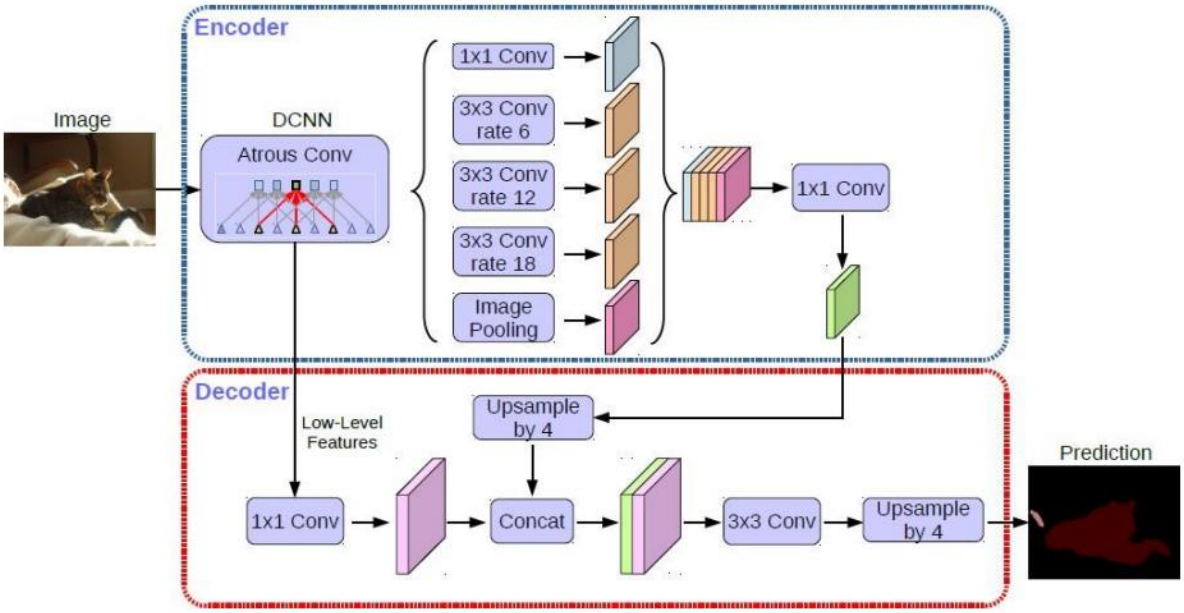

Figure 2. 6: DeepLabv3+ structure (Chen et al., 2018).

2.3 Google Colab

It is a cloud-based online solution derived by Google, which signifies cooperation integrating jupyter notebook infrastructure. It is recommended for machine learning and deep learning studies due to the free GPU capacity it provides and its support for the Python programming language. Nvidia provides customers with K80, T4, P4, and P100 GPUs, according to the Google colab official site. Within the scope of the design project, Google colab was regularly employed. Because it is a Google service, it simplifies access to data sets through its interface with Google Drive. Google colab notebooks include widely used Python libraries by default and allow the library to be installed using pip, Python's official library installation method, if necessary. Uncertain GPU use restrictions, on the other hand, prevent scheduling, particularly in processes containing vast quantities of data and hence requiring long processing periods, and can cause epochs to be interrupted. For increased GPU use restrictions, Google colab suggests adopting the commercial version named Google colab pro.

3.DATASETS

3.1.Massachusetts Buildings Dataset

Volodymyr Mnih provided the dataset featuring Massachusetts building footprints for the first time in his PhD thesis titled "Machine Learning for Aerial Image Labeling," which was released in 2013. Images feature bands of red, green, and blue. Each image is 1500 by 1500 pixels in size and equates to around 2.25 square kilometers. This dataset covers a total area of 340 square kilometers. The real-world equivalent of each image's pixel edge in the Massachusetts collection is 1 meter. The OpenStreetMap project provided the reference labels.

Figure 3. 1: Sample of Aerial image of training data of Massachusetts dataset and mask image containing building footprints

The dataset has been divided into three sections: train, validation, and test. There are 137 aerial images in the train set. There are four images in the validation set. There are ten images in the test set. The OpenStreetMap project's corresponding reference labels for building footprints are imported into the dataset. The omission noise in the dataset is around 5% and below. The dataset mostly consists of various-sized structures, detached residences, and garages.

Data Sets	Imagery	Area	Spatial Resolution	Region
Train Set	137 Image	308.5 km²	1 meter	Boston City
Validation Set	4 Image	9 km²	1 meter	Boston City
Test Set	10 Image	22.5 km²	1 meter	Boston City

Table 3. 1: Massachusetts Buildings Dataset Information Table.

3.2.INRIA Aerial Image Labelling Dataset

This dataset contains orthorectified, colored data with a spatial resolution of 30 cm and adequate spatial accuracy for two semantic classes (Building and non-building classes). The images are aerial images, not satellite images. This data set has an unusual property in that the test and training sets are totally different. Because it comprises cities, it is possible to calculate how the suggested labeling approach generalizes for every given city. This dataset has gained notoriety and popularity by supplying data for several contests and research projects, allowing diverse methodologies to be tried.

Train Regions	Train Tiles	Test Regions	Test Tiles	Region Areas
Austin	36	Bellingham	36	81 km²
Chicago	36	San Francisco	36	81 km²
Kitsap County	36	Bloomington	36	81 km²
Vienna	36	Innsbruck	36	81 km²
Austria	36	East Tyrol	36	81 km²

Table 3. 2: INRIA Aerial Image Labelling Dataset (Maggiori et al., 2017).

The Inria dataset is 5000 × 5000 pixels in size and comprises 36 color (RGB) images of Austin, Chicago, Kitsap County, Western Tyrol, Vienna, Bellingham, San Francisco, Bloomington, Innsbruck, and East Tyrol. This collection, which includes 180 images, covers an area of 810 square kilometers. The reference data is made up of information equivalent to 255 (8-bit images) building 0 in 180 single channel images, but not building class. Official geographic information system agencies, cadastral agencies, and open source official cadastre organizations provided the polygon vectors in the data collection.

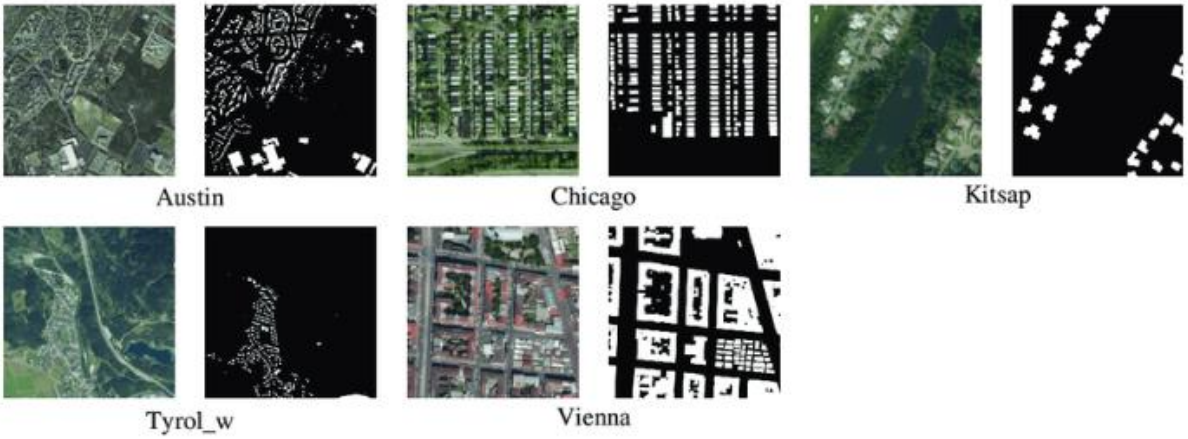

Figure 3. 2: Examples and their ground truth of the Inria dataset (Ma et al., 2020).

4.STUDY

Within the scope of the geomatics engineering design project, using the methods described in chapter 2 and using the datasets introduced in chapter 5, a building detection study was carried out from high resolution aerial images. In this study, Massachusetts Bilding Dataset was used, 6 different models were trained and tested. For this, U-Net and Deeplabv3+ models were combined with ResNet50, ResNet101 and Efficinet-b1 Encoders.

4.1.Augmented Images and Masks

The MB dataset was exposed to three different augmentation functions. There are three of them: horizontal flip, vertical flip, and random rotation. The appropriate labels of the MB dataset are used to generate one hot encoding mask with and without building features.

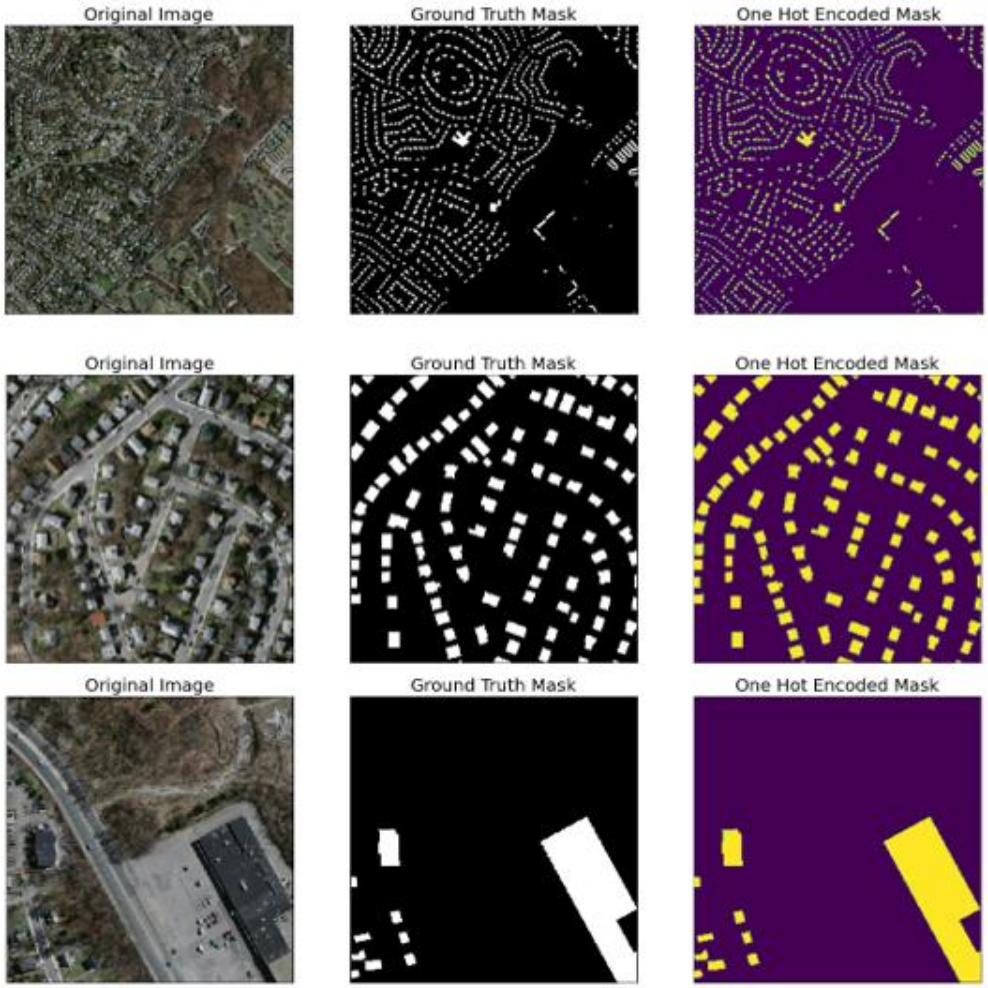

Figure 4. 1: Sample of the original image, Ground truth mask, and one hot encoded mask.

4.2.Training and Testing

The study employs MB dataset train, validation, and test sets. The train set has 137 image, the validation set contains four images, and the test set contains ten images. Pavel Yakubovskiy's Segmentation Models package for Pytorch is used to create the Unet and Deeplabv3+ models. As encoders, Resnet50, Resnet101, and Efficient-b1 are utilized with imagenet weights. The sigmoid function is used to define the activation function. The hyperparameters used in the study are given in the table below.

Model	Batch Size	Epoch	Learning Rate	Optimizer	Activation Function	Loss Function	Score	Encoder
U-Net Model	16	10	0.0001	Adam	Sigmoid	Dice Loss	IoU	Resnet50
U-Net Model	32	80	0.0001	Adam	Sigmoid	Dice Loss	IoU	Resnet101
U-Net Model	32	80	0.0001	Adam	Sigmoid	Dice Loss	IoU	Efficient-b1
Deep Labv3+	32	80	0.0001	Adam	Sigmoid	Dice Loss	IoU	Resnet50
Deep Labv3+	32	80	0.0001	Adam	Sigmoid	Dice Loss	IoU	Resnet101
Deep Labv3+	32	80	0.0001	Adam	Sigmoid	Dice Loss	IoU	Efficient-b1

Table 4. 1: The Study's Hyperparameters

4.3. Results

A building detection study was conducted using high resolution aerial images as part of the geomatics engineering design project. The Massachusetts dataset was used to train six distinct models. The Unet and Deeplabv3+ models are built with Pytorch. Resnet50, Resnet101, and Efficient-b1 are used as encoders.

Before to model prediction, the input MB dataset is divided into three sets: train, validation, and test. The training phase of the model is finished using train and validation sets. We train our model in each iteration of the training process and take measurements with our test set at the conclusion of each 80 epoch.

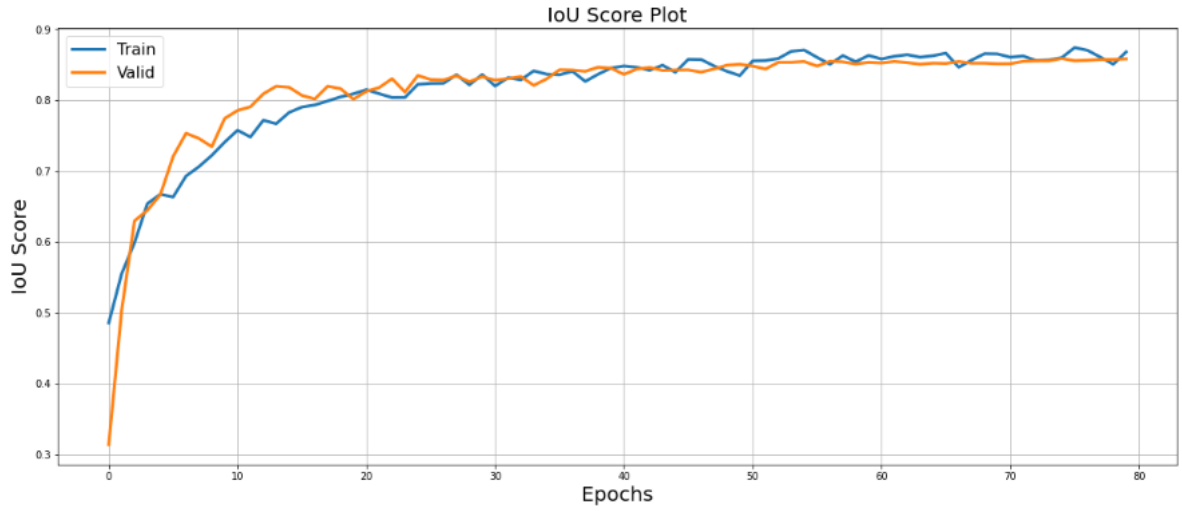

Figure 4. 2: Change in IoU by epoch number of U-Net model with Resnet101 encoder Massachutsetts training and validation score.

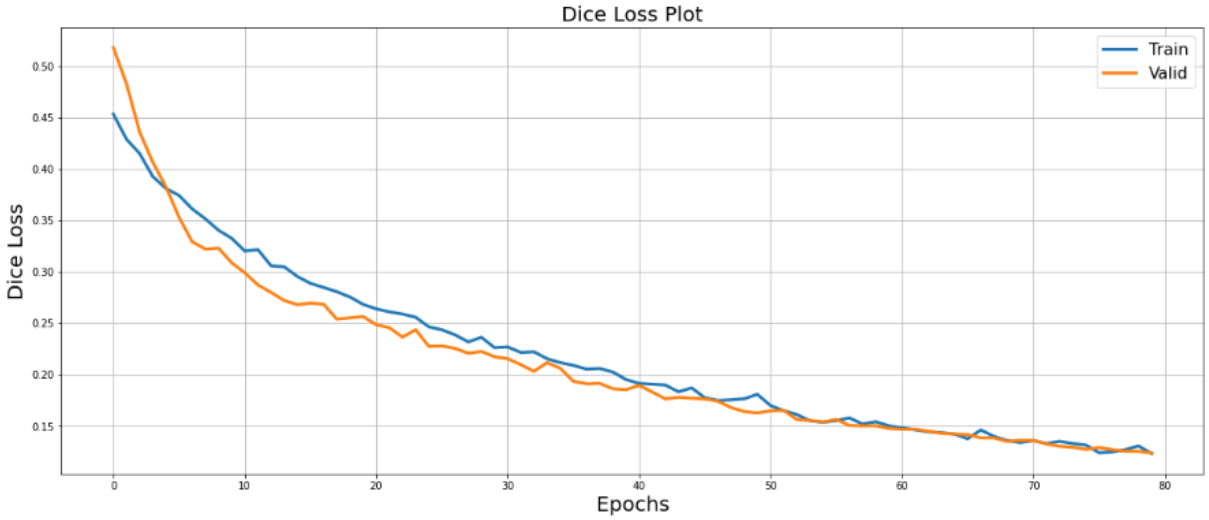

Figure 4. 3: Change in Dice Loss by epoch number of U-Net model with Resnet101 encoder Massachutsetts training and validation score.

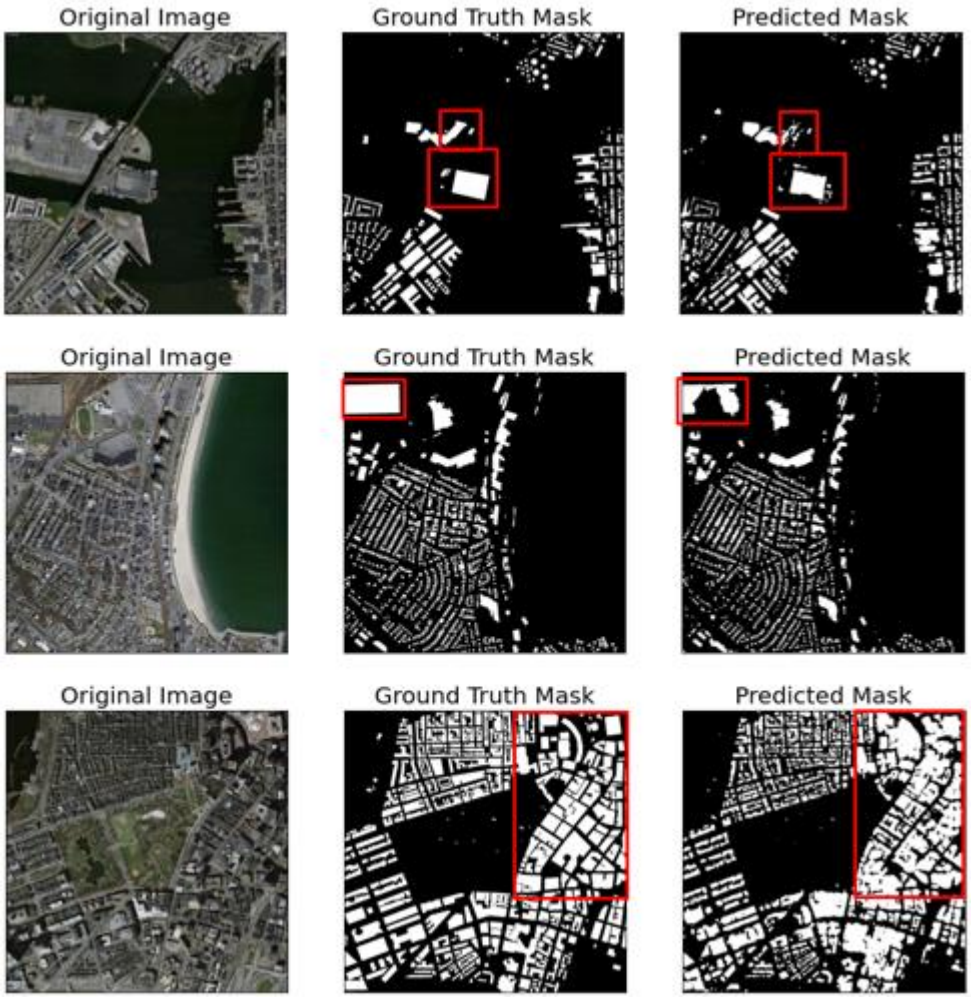

Figure 4. 4: Prediction result of test data trained by U-Net model with Resnet101 encoder

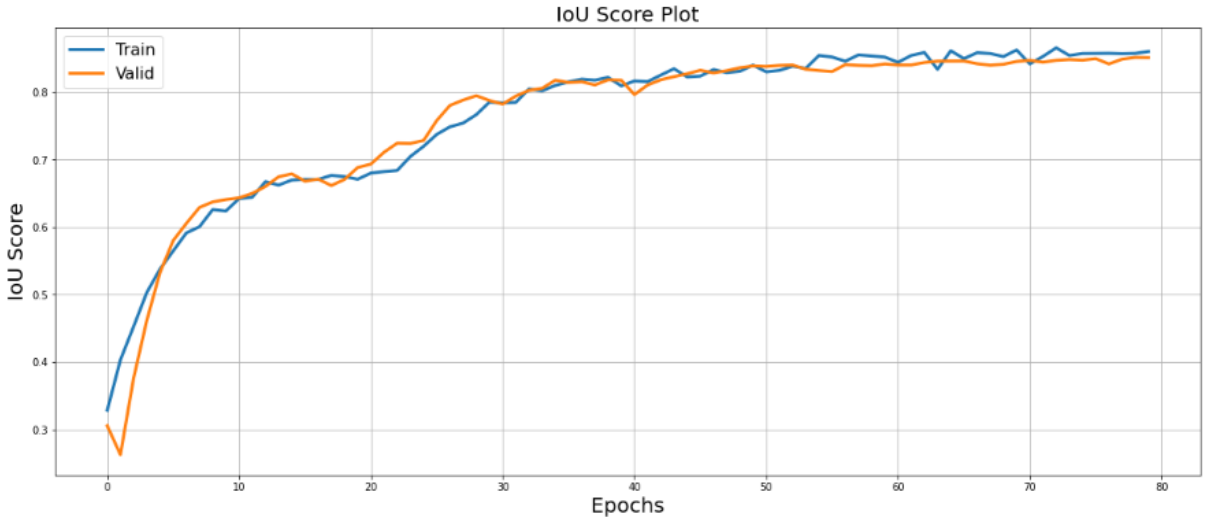

Figure 4. 5: Change in IoU by epoch number of U-Net model with Resnet50 encoder Massachutsetts training and validation score.

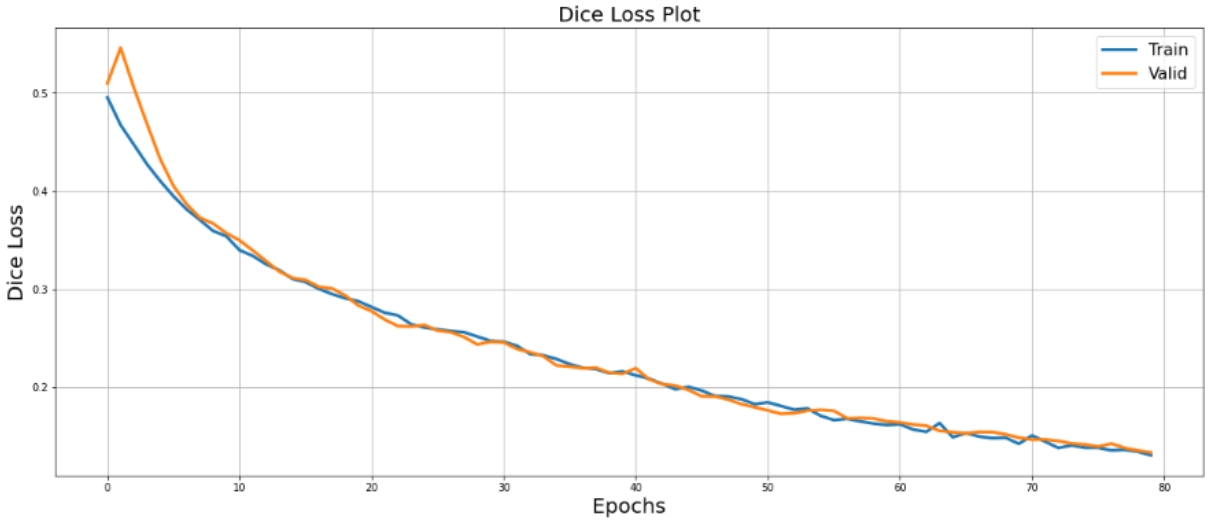

Figure 4. 6: Change in Dice Loss by epoch number of U-Net model with Resnet50 encoder Massachutsetts training and validation score.

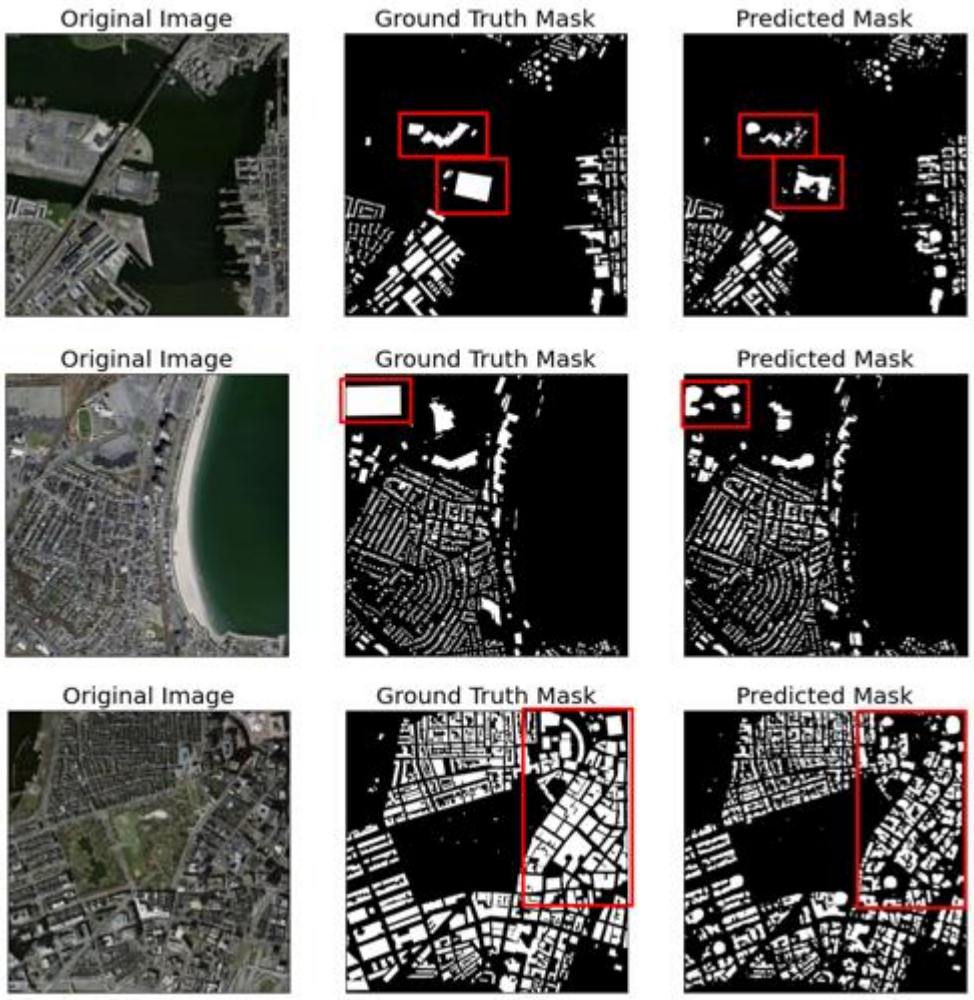

Figure 4. 7: Prediction result of test data trained by U-Net model with Resnet50 encoder

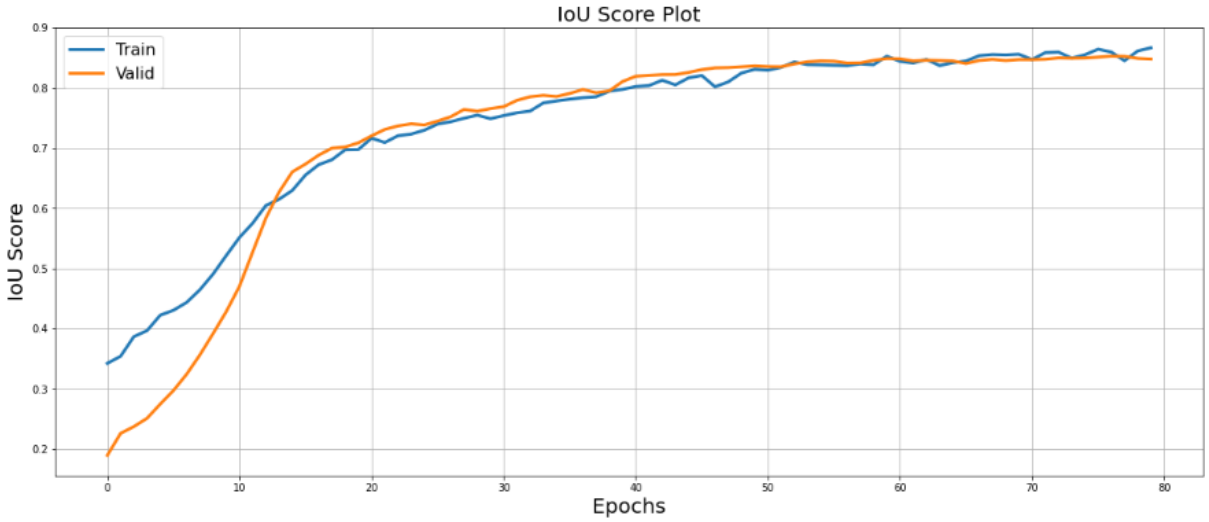

Figure 4. 8: Change in IoU by epoch number of U-Net odel with Efficient-b1 encoder Massachutsetts training and validation score.

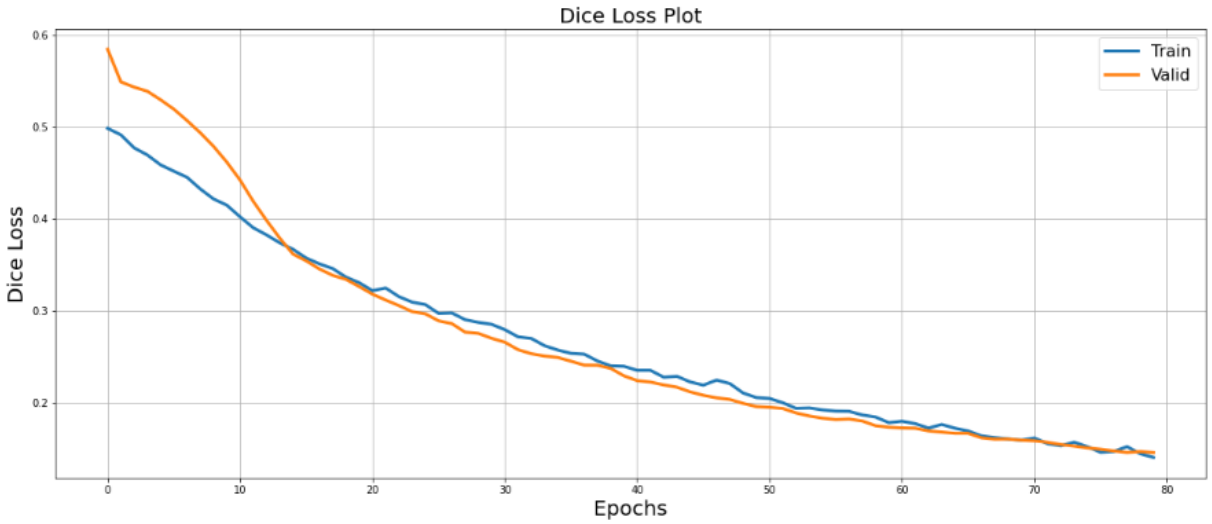

Figure 4. 9: Change in Dice Loss by epoch number of U-Net odel with Efficient-b1 encoder Massachutsetts training and validation score.

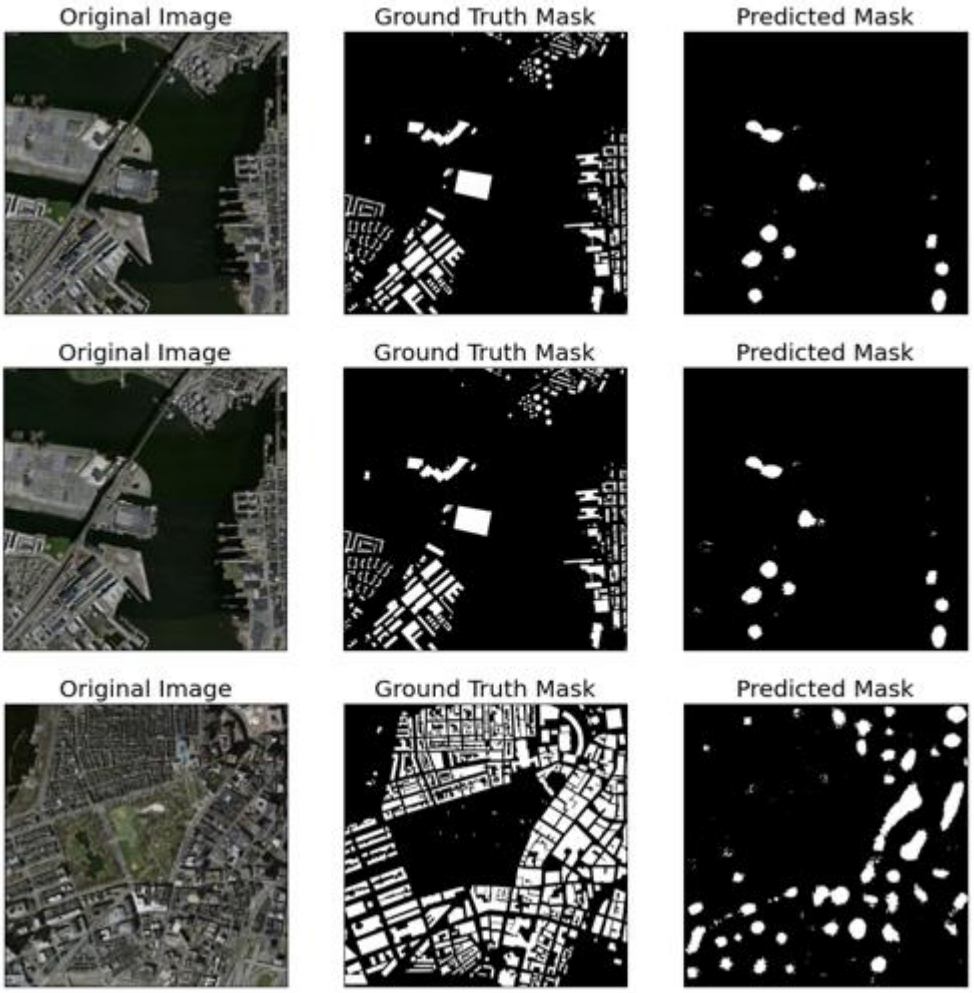

Figure 4. 10: Prediction result of test data trained by U-Net model with Efficient-b1 encoder

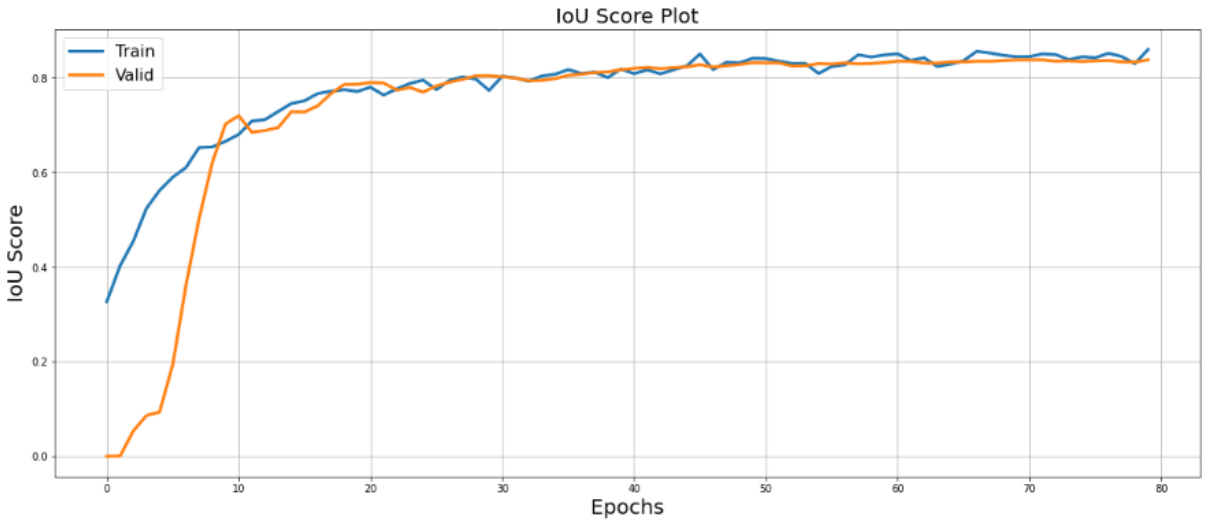

Figure 4. 11: Change in IoU by epoch number of Deeplabv3+ model with Resnet50 encoder Massachutsetts training and validation score.

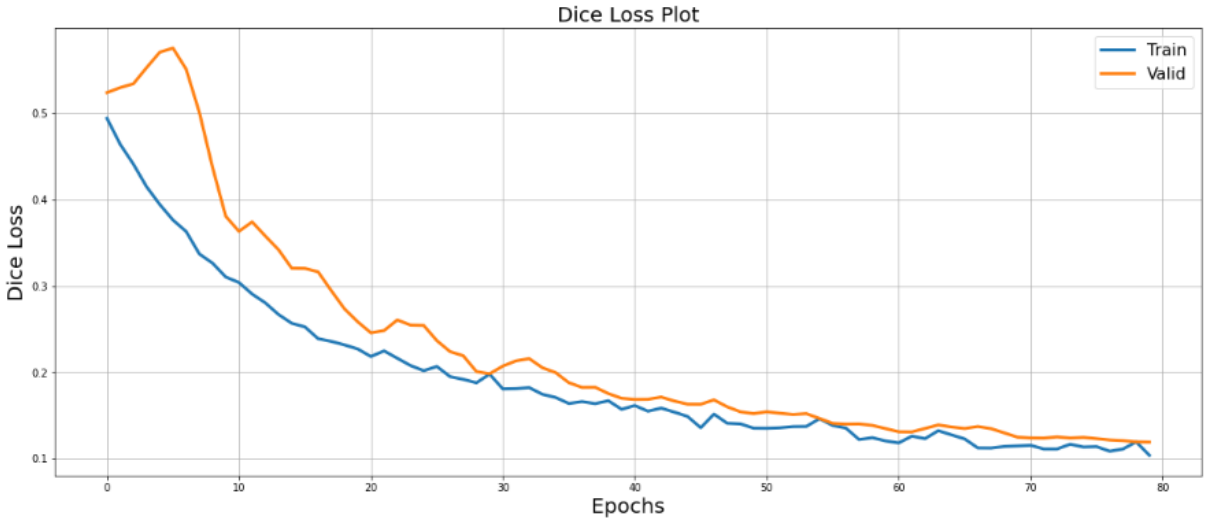

Figure 4. 12: Change in Dice Loss by epoch number of Deeplabv3+ model with Resnet50 encoder Massachutsetts training and validation score.

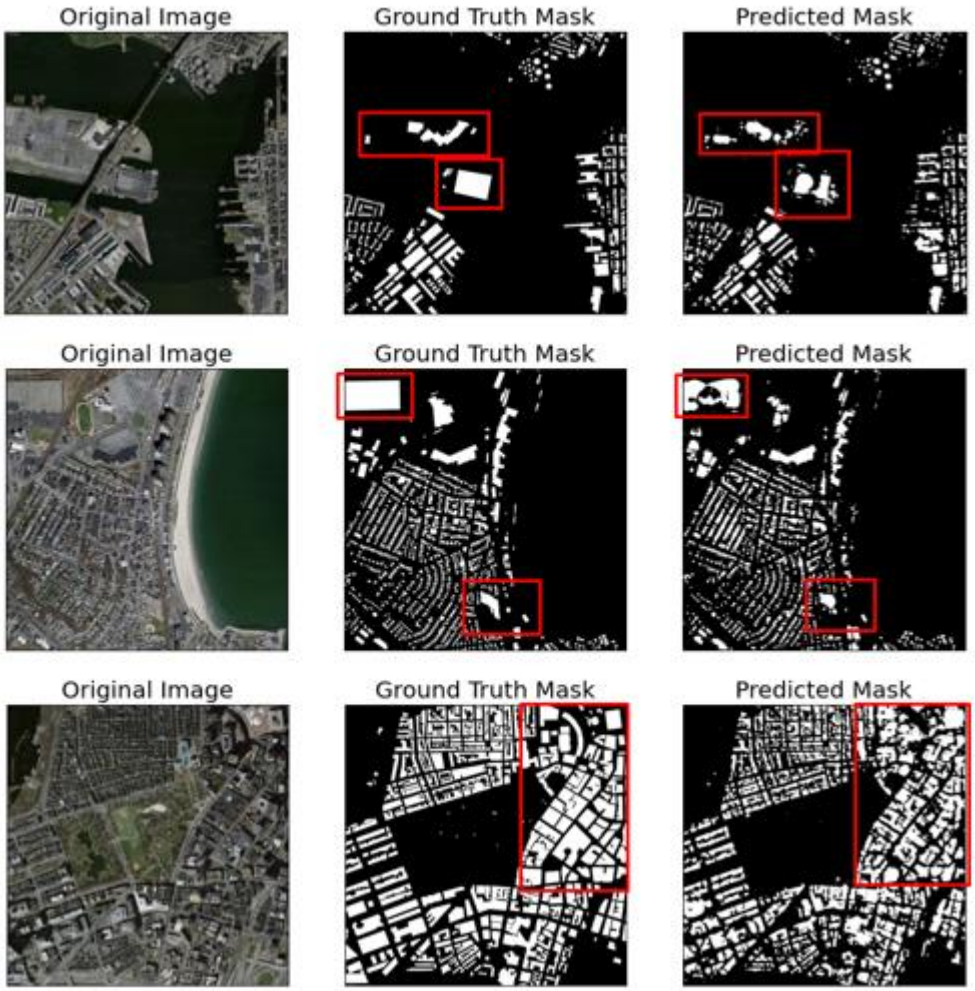

Figure 4. 13: Prediction result of test data trained by Deeplabv3+ model with Resnet50 encoder

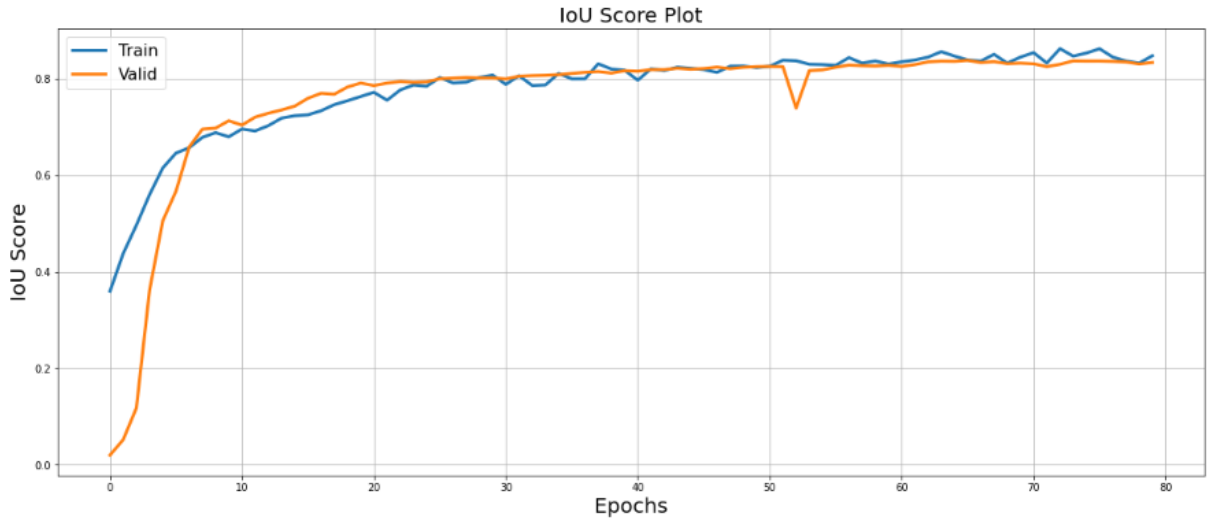

Figure 4. 14: Change in IoU by epoch number of Deeplabv3+ model with Resnet101 encoder Massachutsetts training and validation score.

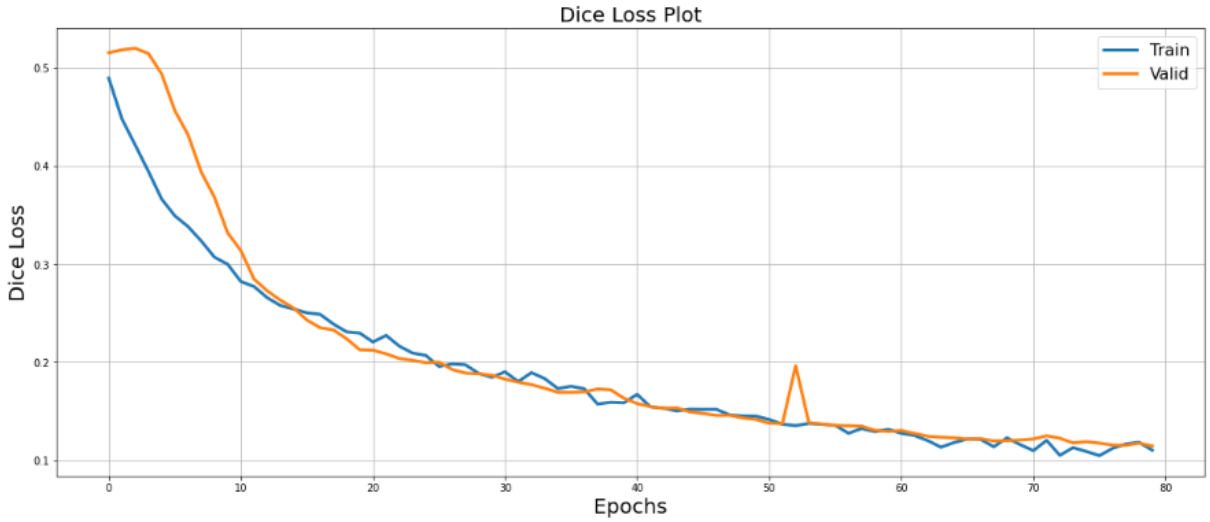

Figure 4. 15: Change in Dice Loss by epoch number of Deeplabv3+ model with Resnet101 encoder Massachutsetts training and validation score.

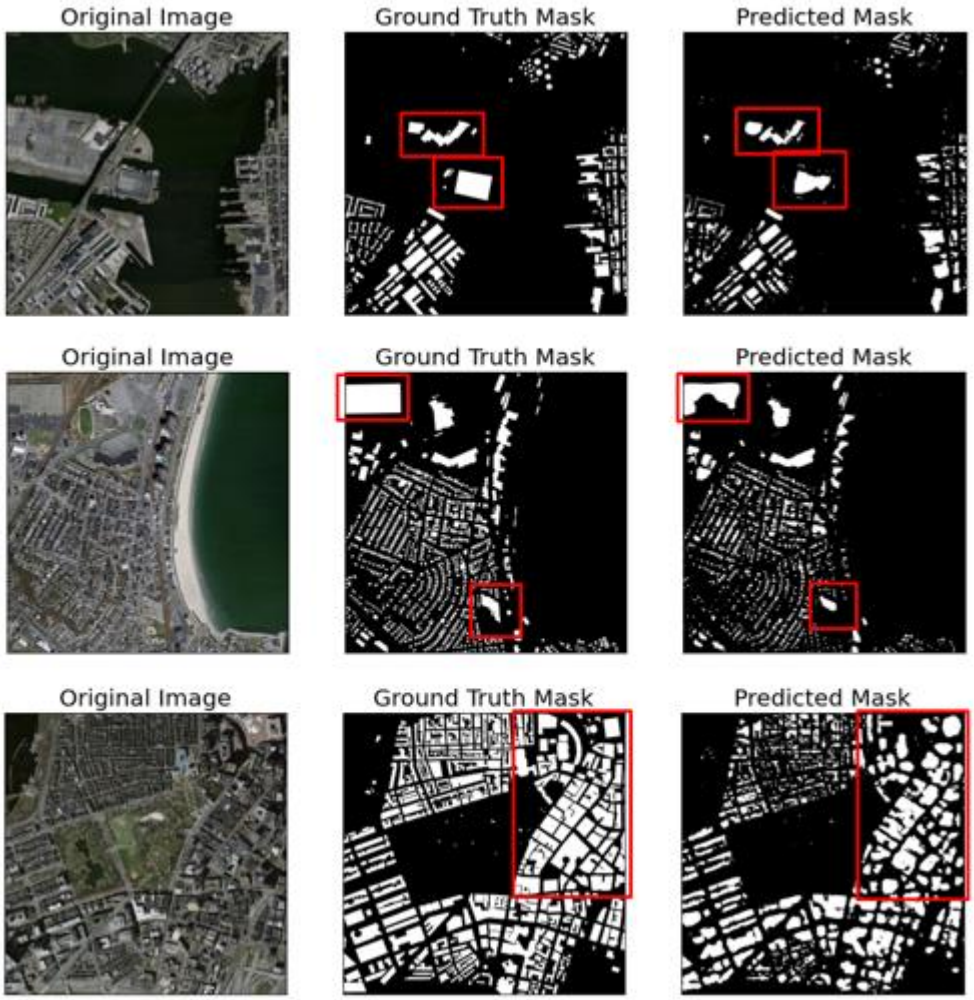

Figure 4. 16: Prediction result of test data trained by Deeplabv3+ model with Resnet101 encoder

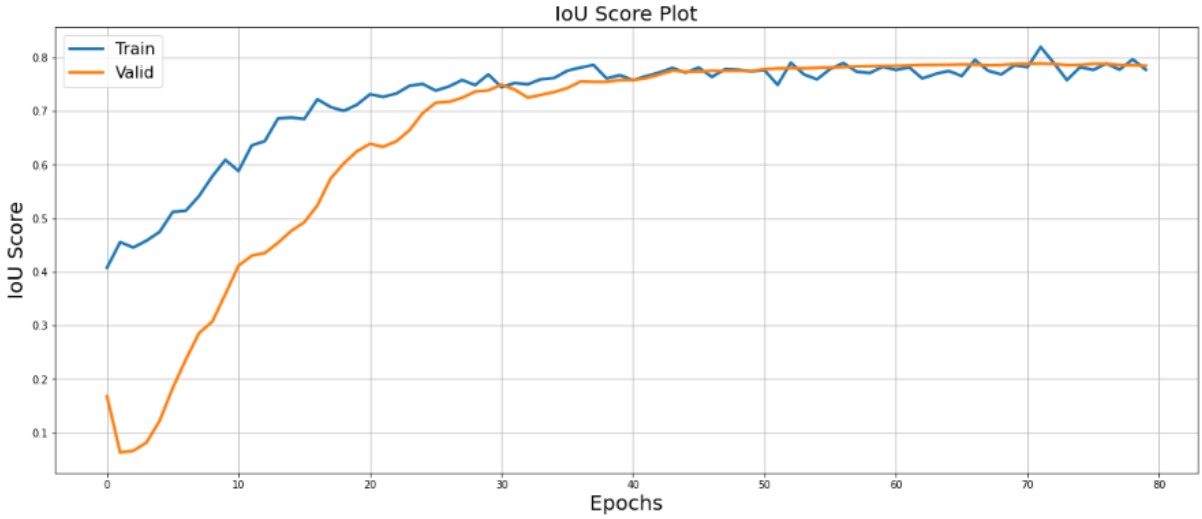

Figure 4. 17: Change in IoU by epoch number of Deeplabv3+ model with Efficient-b1 encoder Massachutsetts training and validation score.

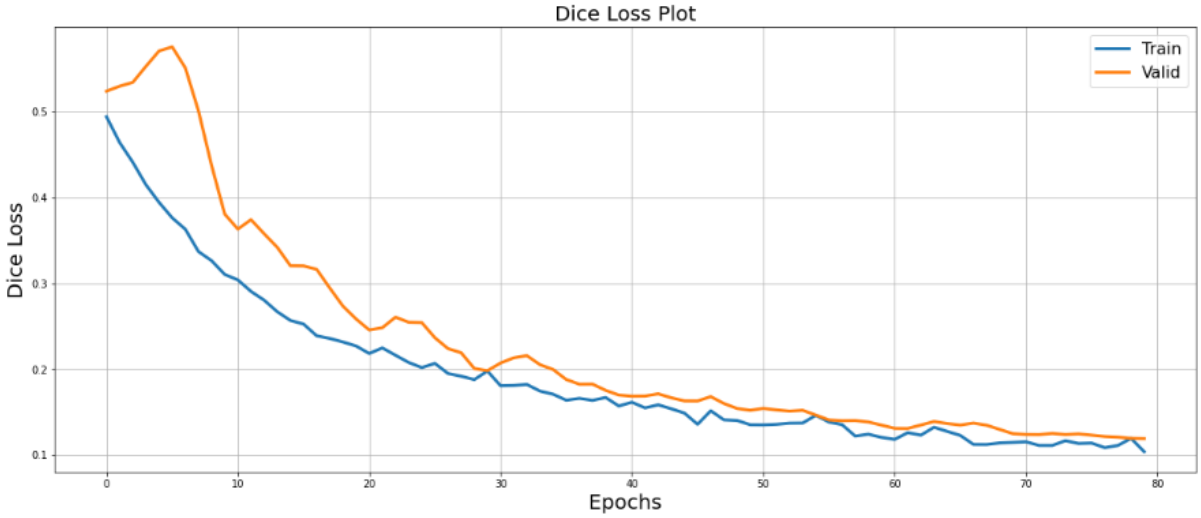

Figure 4. 18: Change in Dice Loss by epoch number of Deeplabv3+ model with Efficient-b1 encoder Massachutsetts training and validation score.

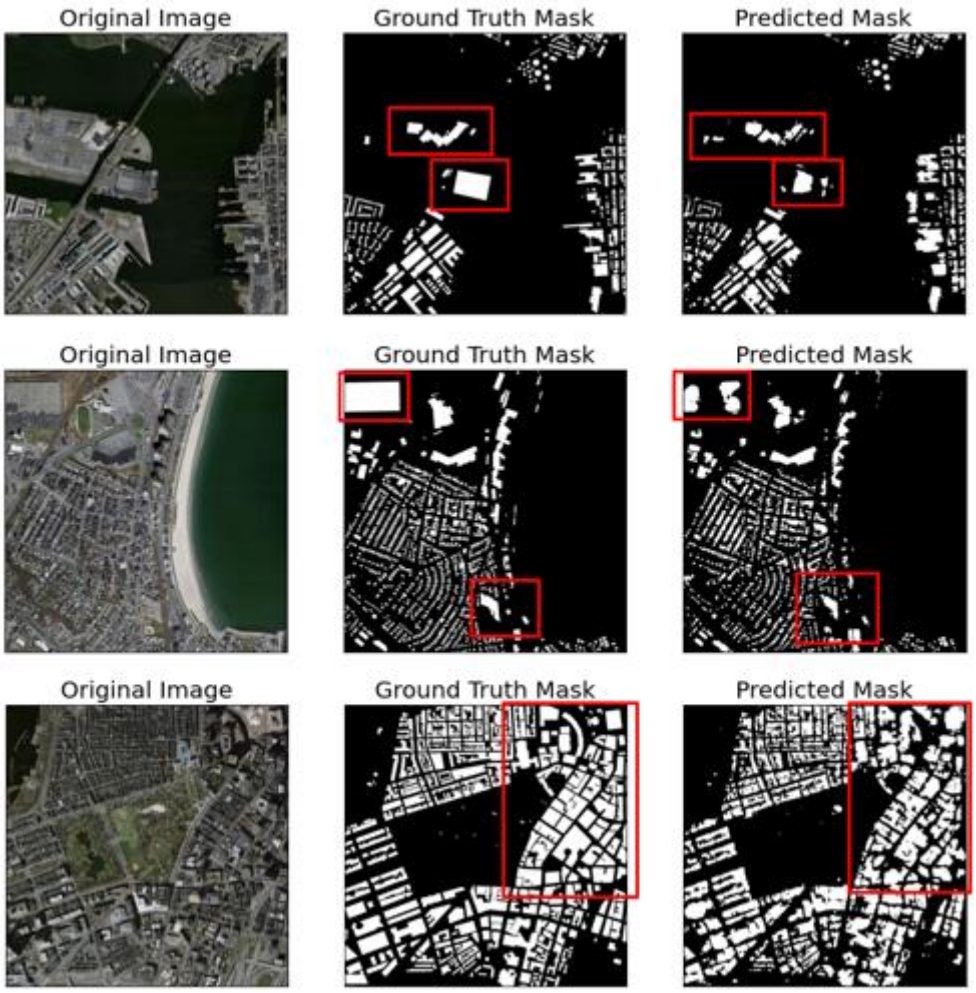

Figure 4. 19: Prediction result of test data trained by Deeplabv3+ model with Efficient-b1 encoder

Model	Epoch	Encoder	Google Colab Train Time	Train IuO Results (%)	Validation IuO Results (%)	Test IuO Results (%)
U-Net	80	ResNet50	8h 3min 27s	86.66	84.74	80.40
U-Net	80	Resnet101	9h 19min 32s	87.42	85.56	81.70
U-Net	80	Efficient-b1	5h 40min 53s	86.65	84.78	80.91
DeepLabv3+	80	ResNet50	5h 58min 46s	86.03	83.83	79.30
DeepLabv3+	80	ResNet101	10h 48min 2s	86.16	82.92	78.87
DeepLabv3+	80	Efficient-b1	5h 38min 27s	81.94	78.87	70.21

Table 4. 2: Performance of the Trained models

Figure 4. 20: Results of Models

After each epoch, we can observe the IoU performance and Dice Loss function presented by the validation and train datasets of each of the six models trained previously. For assessment, the IoU metric was used. The IoU measure is a critical assessment statistic during the decision-making process, particularly in segmentation research. The IoU measure is widely utilized in the literature and in most investigations. The IoU score is a performance measuring metric that is often employed in the category categorization problem of objects (Rahman, Wang, 2016).

When we look at the values in Table 4, we see that the best one of the trained models is the one trained using the resnet101 encoder and the U-Net model, but for the evaluation of the model, the images should be examined and interpreted as well as the scores. When we look at the images given, we can reach the same result. Even though there are some differences between the images, we see that the best result is obtained from the U-Net trained model using the 87.42% IoU result and the resnet101 encoder. However, although the IoU score is not excessive in the model trained using the U-Net and Efficent-b1 encoder when we look at the masks formed as a result of the test, we see that the results are very bad. Increasing the number of epochs or incorporating new datasets into training can increase the accuracy achieved.

5. CONCLUSION AND RECOMMENDATIONS

In this study, U-Net and DeepLabv3+ CNN models with Res-Net(50), Res-Net(101), and Efficient-b1 encoders are employed for building segmentation with high-resolution aerial images from the Massachusetts Buildings dataset. The Massachusetts Buildings dataset is organized into three parts: train, validation, and test. These datasets were used to train and test the model. By comparing the IoU score and Dice loss, the link between these sets was discovered. The model's prediction masks and ground truth labels have been vectorized into two classes: building and not building. While the models trained with U-Net show better results, we see that the model with the highest IoU score is 87.42% IoU result and the model trained with U-Net using the resnet101 encoder. Again using U-Net, 2 more models were trained by trying different encoders, and as a result of these trainings, resnet50 achieved an IoU score of 86.66%, while the model using the Efficient-b1 encoder achieved an IoU score of 86.65%. When we visually interpret the model result obtained using the Efficient-b1 encoder, we see that the result is too bad to deserve this score. This may be due to an incompatibility between the model used and the encoder. While models trained with Deeplabv3+ showed lower performance than U-Net, the model trained with resnet101 encoder achieved IoU scores of 86.16%, resnet50 86.03%, and Efficient-b1 81.94%. Apart from the hyperparameters and regularization approaches used to train the model, other parameters and methods may provide varied accuracy outcomes. This score must be used just for this study because the performance of the models might vary depending on the datasets and methods used. This study demonstrated that adequate hyperparameter selection and optimization, as well as regularization strategies, had an influence on model performance.

REFERENCES

Albawi S., Mohammed T. A., Alzawi S., (2017). Understanding of a Convolutional Neural Network. 10.1109/ICEngTechnol.2017.8308186.

Bertels J., vd., (2019). Optimizing the Dice Score and Jaccard Index for Medical Image Segmentation: Theory & Practice. LNCS 11765, Springer Nature Switzerland AG 2019. doi: 10.1007/978-3-030-32245-8_11

Buslaev A., Iglovikov V., Khvedchenya E., Parinov A., Druzhinin M., Kalinin A., (2018). Albumentations: Fast and Flexible Image Augmentations. Information.

Chen L., Papandreou G., Kokkinos I., Murphy K, Yuille A. L., (2017). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. IEEE Transactions on Pattern Analysis and Machine Intelligence. PP. 10.1109/TPAMI.2017.2699184.

Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S. and Dehmer, M., (2020). An Introductory Review of Deep Learning for Prediction Models With Big Data. Frontiers in Artificial Intelligence, 3. https://doi:10.3389/frai.2020.00004

LeCun, Y., Bengio, Y. and Hinton, G., (2015). Deep learning. Nature, 521(7553), pp.436-444. https://doi.org/10.1038/nature14539

Liu, Y., Zhou, J., Qi, W., Li, X., Gross, L., Shao, Q., Zhao, Z., Ni, L., Fan, X. and Li, Z., (2020). ARC-Net: An Efficient Network for Building Extraction From High-Resolution Aerial Images. IEEE Access, 8, pp.154997-155010. https:// 10.1109/ACCESS.2020.3015701

Ma, J., Wu, L., Tang, X., Liu, F., Zhang, X., & Jiao, L. (2020). Building extraction of aerial images by a global and multi-scale encoder-decoder network. Remote Sensing, 12(15), 2350. https://doi.org/10.3390/rs12152350

Maggiori E., Tarabalka Y., Charpiat G., Alliez P., (2017). Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark, IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, ss. 3226-3229

Nahhas, F., Shafri, H., Sameen, M., Pradhan, B. and, Mansor S., (2018). Deep Learning Approach for Building Detection Using LiDAR–Orthophoto Fusion. Journal of Sensors, 2018, pp.1-12. https://doi.org/10.1155/2018/7212307

Papandreou G., Chen L., Murphy K., Yuille A., (2015). Weakly and SemiSupervised Learning of a DCNN for Semantic Image Segmentation.

Rahman, M., Wang Y., (2016). Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation. 10072. 234-244. 10.1007/978-3-319- 50835-1_22.

Rastogi K., Bodani P., & Sharma S. A. (2020). Automatic building footprint extraction from very high-resolution imagery using deep learning techniques. Geocarto International, 0(0), 1–13. https://doi.org/10.1080/10106049.2020.1778100

Sariturk, B., Bayram, B., Duran, Z., & Seker, D. Z. (2020). Feature Extraction From Satellite Images Using Segnet and Fully Convolutional Networks (Fcn). International Journal of Engineering and Geosciences, October. https://doi.org/10.26833/ijeg.645426

Shrestha, S. and Vanneschi, L., (2018). Improved Fully Convolutional Network with Conditional Random Fields for Building Extraction. Remote Sensing, 10(7), p.1135. https://doi:10.3390/rs10071135

Thoma, M., (2016). A Survey of Semantic Segmentation. ArXiv, abs/1602.06541.

Wu, G., Shao, X., Guo, Z., Chen, Q., Yuan, W., Shi, X., Xu, Y. and Shibasaki, R., (2018). Automatic Building Segmentation of Aerial Imagery Using Multi-Constraint Fully Convolutional Networks. Remote Sensing, 10(3), p.407. https://doi:10.3390/rs10030407

Xie, Y., Zhu, J., Cao, Y., Feng, D., Hu, M., Li, W., Zhang, Y. and Fu, L., (2020). Refined Extraction Of Building Outlines From High-Resolution Remote Sensing Imagery Based on a Multifeature Convolutional Neural Network and Morphological Filtering. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, pp.1842-1855. https://10.1109/JSTARS.2020.2991391

Yamashita R., Nishio M., Do R. K. G., Togashi K., (2018). Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, ss.611–629. https://doi.org/10.1007/s13244-018-0639-9

Yang G., Zhang Q., Zhang G., (2020). EANet: Edge-Aware Network for the Extraction of Buildings from Aerial Images. Remote Sens., 12, 2161

Zhang, Y., Gong, W., Sun, J., & Li, W., (2019). Web-Net: A novel nest networks with ultra-hierarchical sampling for building extraction from aerial imageries. Remote Sensing, 11(16). https://doi.org/10.3390/rs11161897

Zhang, Y., Li, W., Gong, W., Wang, Z., & Sun, J., (2020). An improved boundary-aware perceptual loss for building extraction from VHR images. Remote Sensing, 12(7). https://doi.org/10.3390/rs12071195

Zhao, Z., Zheng, P., Xu, S. and Wu, X., (2019). Object Detection With Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems, 30(11), pp.3212-3232