

## **ABSTRACT**

Semantic segmentation of satellite imagery plays an important part in different applications like metropolitan preparation, ecological observing, and disaster management. This project investigates the use of profound learning strategies, especially U-Net Convolutional neural networks, for object detection in high-resolution satellite imagery. The review centres around distinguishing streets, structures, sunlight powered chargers, and vehicles, utilizing progressed convolutional brain network models. Key techniques incorporate picture handling, include extraction, and AI calculations. The examination considers the exhibition of the proposed approach in contrast to existing techniques, showing improved precision and productivity. Results feature the capability of deep learning in satellite image examination, preparing for further developed navigation and asset distribution in assorted areas. The discoveries of this study add to the developing assortment of exploration in satellite picture examination and proposition experiences into the future bearings of profound learning applications in remote detecting.

**Keywords:** Semantic segmentation, Satellite imagery, Deep learning, Convolutional neural networks, Object detection.

## **TABLE OF CONTENTS**

<b><u>S.No.</u></b>	<b><u>CHAPTER</u></b>	<b><u>TITLE</u></b>	<b><u>P.No.</u></b>
1.		ABSTRACT	4
2.		ACKNOWLEDGEMENT	5
3.		LIST OF TABLES, FIGURES & ACRONYMS	8
4.	1	INTRODUCTION	12
	1.1	PROBLEM BACKGROUND	14
	1.2	OBJECTIVES	15
	1.3	CHALLENGES	16
	1.4	BACKGROUND & LITERATURE SURVEY	16
	1.5	LIMITATIONS AND ASSUMPTIONS	18
	1.6	ORGANIZATION OF THE REPORT	19
5.	2	BACKGROUND	20
	2.1	ARTIFICIAL INTELLIGENCE	20
	2.1.1	MACHINE LEARNING	20
	2.1.2	DEEP LEARNING	21
	2.2	NEURAL NETWORKS	21
	2.3	CONVOLUTIONAL NEURAL NETWORKS	22
	2.4	SEMANTIC SEGMENTATION	23
	2.5	FULLY CONVOLUTIONAL NEURAL NETWORKS	23
	2.6	U-NETS	25
	2.6.1	RESULTS IN AERIAL IMAGERY	26
	2.6.2	EVALUATION METRICS	27
	2.6.2.1	JACCARD COEFFICIENT	27
	2.6.2.2	PIXEL ACCURACY	28
	2.6.3	LOSS FUNCTION FOR SEGMENTATION	28
	2.6.3.1	INTERSECTION OVER UNION AS LOSS FUNCTION	29
	2.6.3.2	BINARY CROSS ENTROPY	30

	2.7	RESNET-50	31
6.	3	METHODOLOGY	33
	3.1	DATASET	33
	3.2	PREPROCESSING	35
	3.3	MODEL ARCHITECTURE	36
	3.4	PRETRAINED MODELS	38
	3.4.1	RESNET-50	38
7.	4	RESULTS AND ANALYSES	40
	4.1	TRAINING THE MODEL	40
	4.1.1	U-NET TRAINING	41
	4.1.2	RESNET MODEL TRAINING	44
	4.2	TESTING THE MODEL	45
	4.2.1	U-NET MODEL PREDICTION	46
	4.2.2	RESNET-50 MODEL PREDICTION	48
	4.3	SUMMARY	49
8.	5	CONCLUSION & FUTURE WORK	51
9.	6	APPENDICES	52
	6.1	ABOUT THE DATASET	52
	6.2	ASSIGNING COLOURS TO THE LABELS	53
	6.3	TEST AND TRAIN DATASET RATIO	55
	6.4	IMAGE DIMENSIONS	55
	6.5	JACCARD COEFFICIENT	56
		IMPLEMENTATION	
	6.6	U-NET MODEL IMPLEMENTATION	56
	6.7	GENERATING LOSS FUNCTION	58
	6.8	MODEL SUMMARY	59
	6.9	TRAINING & VALIDATION LOSS CURVE	61
10.	8	REFERENCES	62

## **LIST OF TABLES**

<b><u>TABLE NO.</u></b>	<b><u>TITLE</u></b>	<b><u>PAGE NO.</u></b>
3.1	LABELS AND THEIR CORRESPONDING COLOURS	33

## **LIST OF FIGURES**

<b><u>FIG.NO.</u></b>	<b><u>TITLE</u></b>	<b><u>PAGE NO.</u></b>
2.1	A SIMPLE NEURAL NETWORK	22
2.2	THE CNN COMPONENTS	22
2.3	FULLY CONVOLUTIONAL NEURAL NETWORK	24
2.4	U-NET ARCHITECTURE	25
2.5	JACCARD COEFFICIENT UNION SET	27
2.6	IoU LOSS FUNCTION	29
2.7	ARCHITECTURE OF RESNET-50	32
3.1	ORIGINAL IMAGE VS. MASKED IMAGE	34
3.2	ORIGINAL IMAGE VS MASKED IMAGE	34
3.3	U-NET MODEL ARCHITECTURE	37
3.4	VGG-19 VS. PLAIN NETWORK VS. RESNET ARCHITECTURE	39
4.1	GRAPH OF JACCARD COEFFICIENT	41
4.2	TRAINING LOSS VS. VALIDATION LOSS	43
4.3	TRAINING IoU VS. VALIDATION IoU	43
4.4	RESNET-50 TRAINING CURVE (ACCURACY)	44
4.5	RESNET-50 TRAINING CURVE (LOSS)	45
4.6	RESNET-50 TRAINING CURVE (MEAN IoU)	45
4.7	TEST IMAGE VS. TEST LABEL VS. PREDICTION (U-NET)	47
4.8	TEST IMAGE VS. TEST LABEL VS. PREDICTION (RESNET-50)	48
6.1	FILE STRUCTURE OF DATASET	52
6.2	DATASET SUMMARY	53
6.3	LABELS AND COLOURS	54
6.4	FUNCTION ASSIGNING COLOURS TO LABELS	54

6.5	RGB ARRAYS	55
6.6	TRAIN AND TEST SIZE	55
6.7	IMAGE DIMENSIONS	56
6.8	JACCARD COEFFICIENT FUNCTION	56
6.9	U-NET MODEL IMPLEMENTATION	57
6.10	LOSS FUNCTION IMPLEMENTATION	58
6.11	MODEL SUMMARY	59
6.12	TRAINABLE VS. NON-TRAINABLE PARAMETERS	60
6.13	TRAINING & VALIDATION LOSS CURVE IMPLEMENTATION	61
6.14	TRAINING & VALIDATION IoU CURVE IMPLEMENTATION	61

## **LIST OF ACRONYMS**

<b><u>ACRONYM</u></b>	<b><u>FULL FORM</u></b>
CNN	CONVOLUTIONAL NEURAL NETWORK
DL	DEEP LEARNING
FCN	FULLY CONVOLUTIONAL NEURAL NETWORK
IoU	INTERSECTION OVER UNION
ML	MACHINE LEARNING
RESNET	RESIDUAL NETWORK
ReLU	RECTIFIED LINEAR UNIT
GPS	GLOBAL POSITIONING SYSTEM
GNSS	GLOBAL NAVIGATION SATELLITE SYSTEM
AI	ARTIFICIAL INTELLIGENCE
ANN	ARTIFICIAL NEURAL NETWORKS
MSE	MEAN SQUARED ERROR
SGD	STOCHASTIC GRADIENT DESCENT
VGG	VISUAL GEOMETRY GROUP

# **CHAPTER 1**

## **INTRODUCTION**

Deep learning is a beacon of both fascination and apprehension in the field of artificial intelligence. Its capacity to imitate the intricate human mind, from visual comprehension to speech synthesis, raises significant concerns regarding the boundaries that separate machine and consciousness. Deep learning is a convergence of scientific inquiry and technological innovation, drawing inspiration from medical and neurological studies to design algorithms that digitally mirror our cognitive faculties. It is rooted in the endeavour to replicate the cognitive processes of the human brain.

In the age of big data, the exponential growth of digital information has pushed deep learning, which is part of the larger field of artificial intelligence, to new heights. Its rise signifies a paradigm shift in decision-making and opens the door to autonomous, data-driven insights free from human limitations. Deep learning is a distinct scientific field that emerged from earlier epochs of cybernetics and connectionism, propelled by significant breakthroughs since the beginning of the 21st century.

The meticulous pixel-level delineation of objects in an image is the hallmark of semantic image segmentation, which represents the cutting edge of computer vision. Semantic segmentation goes beyond traditional labelling methods to produce high-resolution outputs enriched with contextual understanding by equipping machines with the ability to distinguish and classify visual data in a manner analogous to human perception. A new era of intelligent image analysis is being heralded thanks to this computational prowess, which is put to use in a variety of fields, including urban planning and medical diagnostics.

The integration of deep learning and semantic segmentation into satellite imagery signals revolutionary possibilities for image-aided navigation. The need for alternative solutions has prompted the investigation of image-based localization as a result of the widespread reliance on GPS navigation's vulnerability. Utilizing convolutional brain organizations, especially the U-Net design, analysts try to remove remarkable



elements from aeronautical pictures, making ready for vigorous restriction structures versatile to the constraints of conventional satellite route frameworks.

As a result, there is a potential for revolutionary advancements in navigation technology at the intersection of deep learning, semantic segmentation, and satellite imagery. Researchers commenced on a journey to redefine the boundaries of spatial awareness and propel the evolution of navigation systems into uncharted territories of precision and reliability by using the power of artificial intelligence to extract actionable insights from vast troves of visual data.

The need for sophisticated image analysis methods grows stronger as the demand for accurate, real-time geospatial data grows in a variety of fields like urban planning, disaster response, and environmental monitoring. In this endeavour, semantic segmentation of satellite imagery emerges as a linchpin, making it possible to extract the nuanced spatial features necessary for making informed decisions. Stakeholders are given granular insights into the dynamics shaping our planet's surface thanks to semantic segmentation, which distinguishes between various land cover types, infrastructure components, and natural phenomena.

Semantic segmentation algorithms have also improved as a result of the convergence of machine learning and remote sensing, which has facilitated the creation of extensive and varied datasets. The abundance of annotated data sources has democratized access to training data, fostering innovation and collaboration within the research community. These sources range from publicly accessible satellite imagery repositories to proprietary datasets curated by governmental agencies and private businesses. Geospatial engineering, the earth sciences, and computer science all benefit from this proliferation of data-driven insights, which not only speeds up the development of algorithms but also encourages interdisciplinary synergies.

The incorporation of semantic segmentation into image-aided localization holds promise for enhancing positioning systems' resilience and robustness in the context of navigation technology. Localization algorithms can add rich environmental cues to conventional GPS or GNSS position estimates by utilizing the semantic context encoded in aerial imagery. Image-aided localization provides a complementary solution for ensuring continuous and dependable positioning in a variety of operating

environments, whether traversing urban canyons where satellite signals are obscured or remote regions devoid of infrastructure.

## **1.1 PROBLEM BACKGROUND**

The growth of satellite imagery and advancements in deep learning in recent years have sparked a growing interest in using aerial segmentation for a variety of purposes, including urban planning and disaster response. However, despite significant progress in the field, several persistent obstacles prevent aerial segmentation techniques from being widely used and effective.

One of the essential deterrents is the shortage of top calibre, commented on datasets enveloping different spatial goals and geological areas. Making marked covers for various classes, like streets and structures, is a work serious and tedious cycle, frequently requiring careful manual comment. As a result, it's possible that the current datasets lack the breadth and depth required to train robust segmentation models that can be applied to a variety of urban landscapes and environmental contexts. The comparison and benchmarking of segmentation algorithms is further complicated by the absence of standardized evaluation benchmarks.

Gauge correlations for creating higher-performing models are in many cases scant, preventing progress and blocking the distinguishing proof of best practices in model engineering and preparing methodologies. Also, the adequacy of airborne division models can be undermined by dataset inclinations and restricted speculation capacities. Models prepared on homogeneous datasets may battle to adjust to assorted genuine situations, prompting less than ideal execution and untrustworthy outcomes in functional settings. Comprehensive datasets, improved model architectures, and solid evaluation frameworks are all necessary to address these issues. Researchers can unlock the full potential of aerial segmentation for a wide range of applications, including infrastructure management and disaster response, urban planning, and environmental monitoring.

## **1.2 OBJECTIVES**

The precise segmentation of aerial imagery for a variety of important applications is the primary objective of this research project. The first goal is to create a comprehensive dataset by making use of aerial imagery that is freely available online. This will help us get around the problems caused by different spatial resolutions and locations. This dataset aims to provide a solid foundation for training and evaluating semantic segmentation models, including variations in environmental contexts and urban landscapes, by combining images from various sources.

Implementing U-Net architectures that are specifically designed to detect and distinguish prominent features like roads and buildings in aerial imagery is the second objective. The research aims to streamline the model training process and improve performance for real-world applications where the identification of infrastructure elements is essential by defining these classes as the primary focus of semantic segmentation. To produce high-fidelity segmentation results, each U-Net model will undergo rigorous training on the annotated dataset, integrating spatial patterns and contextual cues.

The research aims to investigate the possibility of combining existing datasets to expand the training corpus and improve the model's capacity to generalize across a variety of temporal and environmental conditions. The study aims to improve the semantic segmentation framework's robustness and adaptability by utilizing the collective insights gleaned from various datasets. This will guarantee consistent performance across various geographic regions and seasonal dynamics.

The ultimate objective of this research project is to use the trained U-Net models as learning-based feature detectors and localization tools that can identify and locate key classes like roads and buildings regardless of seasonal and temporal changes. The research aims to improve the resilience and accuracy of image-aided navigation systems in a variety of operational contexts by utilizing the power of deep learning and semantic segmentation.

### **1.3 CHALLENGES**

1. *Manual Inspection:* Experts would need to physically examine vast amounts of satellite imagery to identify objects and patterns. This process is extremely time-consuming and practically unfeasible, especially considering the enormous geographic areas that require monitoring. The manual effort required would be immense, leading to significant delays in data analysis and decision-making.
2. *Human Error:* Manual assessment is susceptible to human error, which can lead to mistakes and inconsistencies. These errors might result in missed detections or false identifications, compromising the effectiveness of monitoring efforts. Inconsistent evaluations due to human fatigue or oversight could further degrade the reliability of the results.
3. *Limited Coverage:* The labour-intensive nature of manual assessment would limit the extent of geographic coverage, creating potential gaps in monitoring critical areas. These gaps could delay the detection of illegal activities, environmental hazards, or other crucial events, thereby hindering timely intervention and response.
4. *Resource Constraints:* The limited availability of trained analysts would hamper the ability to conduct timely and comprehensive monitoring of satellite imagery. This constraint would be particularly challenging during emergencies or situations requiring rapid response, as the demand for immediate and accurate analysis would far exceed the available human resources.

### **1.4 BACKGROUND AND LITERATURE SURVEY**

A similar research paper and project was carried out by Rivera, Antonio & Rolando et al., [14] titled “Satellite Image Using Image Processing and Machine Learning Techniques: applications to agriculture, environment and mining”. This study explores the classification of satellite image categories employing image processing and machine learning methodologies. Satellite imagery serves as a important research tool, enabling continuous monitoring of Earth's surface through multi-temporal data,

offering extensive coverage over vast geographical regions worldwide. Notably, satellite imagery remains unaffected by sunlight or atmospheric conditions, rendering it an ideal source for remote sensing applications. The primary aim of this investigation is to employ image analysis and classification techniques to identify and categorize satellite images into distinct groups. A dataset comprising 50 sample images was utilized for this purpose. Various features including colour-based, statistical, and texture features were extracted using wavelet transform techniques. Machine learning algorithms, specifically the KNN classifier, were then employed to classify these images into predefined categories, with a training-testing split of 70-30%. The effectiveness of this approach was evaluated using a confusion matrix, demonstrating its capability to accurately classify satellite images. Notably, the db7 wavelet achieved a high accuracy rate of 95% when combined with the KNN classifier, surpassing alternative methods for satellite image characterization.

We also referred to the paper titled “Semantic Segmentation of Aerial Images with an Ensemble of CNSS” by Dimitris & Wegner et al., [15]. This research presents a deep learning method for semantic segmentation of high-resolution aerial images. Deep neural networks offer the advantage of learning directly from raw images, eliminating the need for manual feature engineering. In recent years, deep convolutional neural networks (CNNs) have gained popularity for various image analysis tasks, including visual recognition and object detection. However, traditional CNNs are not well-suited for per-pixel semantic segmentation as they tend to aggregate information across larger image regions, making it difficult to discern individual pixel contributions. Recently, advancements such as deconvolutional network layers and Fully Convolutional Networks (FCNs) have addressed this limitation, enabling precise pixel-level segmentation. We propose an FCN architecture that takes both intensity and range data as input, utilizing aggressive deconvolution and using early network layers to achieve pixelwise classification at full resolution. We discuss the design considerations and complexities involved in such a network and demonstrate its efficacy through ensemble learning, achieving excellent results on challenging datasets such as the ISPRS semantic labelling benchmark using only raw data inputs.

To make the model more precise and error prone, we took help of the research conducted by N. Subraja and D. Venkateshkar, [11] "Satellite Image Segmentation

using Modified U-Net Convolutional Networks”. Object detection in satellite imagery has become increasingly prominent in recent years, playing a pivotal role across various applications. With the widespread success of deep learning techniques in computer vision, their application has extended to satellite imagery for tasks such as object identification, tracking, classification, and semantic segmentation. Despite numerous studies examining object detection in satellite imagery, this review focuses on the latest advancements in this field specifically utilizing deep learning methodologies. The paper discusses the detection of roads, buildings, solar panels, and vehicles employing Modified U-Net Convolutional networks, demonstrating improved accuracy compared to previous methods.

## **1.5 LIMITATIONS AND ASSUMPTIONS**

The project faces several limitations that affect its ability to fully achieve its research objectives. Chief among these limitations is the scarcity of diverse and annotated aerial imagery datasets, which constrains the scope and generalization potential of the developed models. Ideally, the research aims to produce a network capable of invariant spatial resolution and adept at detecting buildings and roads across various locations. However, the current lack of sufficient data at different spatial resolutions and geographical locales renders this goal unrealistic. Additionally, the dearth of data limits the availability of baseline datasets for comparison with existing models. Consequently, assessing the performance of the proposed approach is hindered by the absence of standardized benchmarks. Furthermore, reliance on publicly accessible datasets introduces the risk of biases or constraints in terms of spatial resolutions, geographic coverage, and class distributions, potentially impacting the efficacy of the developed models. Despite these limitations, the project assumes that using existing datasets and augmenting them with publicly available imagery will enhance the diversity and richness of the training data, thereby improving model robustness and generalization. Additionally, the research anticipates that advancements in aerial imagery collection technologies and annotation methods will lead to the availability of more accurately labelled datasets in the future, facilitating the creation of standardized benchmarks for evaluation. Ultimately, the project assumes that despite these inherent limitations, the developed models will offer valuable insights and contribute to advancements in image-aided navigation systems.

## **1.6 ORGANIZATION OF THE REPORT**

The remaining chapters of the report are given as follows:

Chapter 2: Background

Chapter 3: Methodology

Chapter 4: Results and Analysis

Chapter 5: Conclusion

Chapter 6: Appendices

## **CHAPTER 2**

### **BACKGROUND**

This section gives a specialized outline of profound advancing alongside foundation data on the semantic segmentation task. In doing as such, convolutional neural networks (CNNs) and their utilization in this picture order issue will be evaluated, alongside a variety of the CNN called the U-Net. Measurements and misfortune capabilities intended for this undertaking will likewise be investigated, alongside their impact on the last divisions versus regular assessment measurements for arrangement.

#### **2.1 ARTIFICIAL INTELLIGENCE**

Artificial intelligence alludes to the reproduction of human insight by a framework or a machine. The objective of man-made intelligence is to foster a machine that can think like people and copy human ways of behaving, including seeing, thinking, picking up, arranging, foreseeing, etc. One of the most important characteristics that sets humans apart from animals is intelligence. With the endless event of modern upheavals, a rising number of kinds of machine types persistently supplant human work from varying backgrounds, and the up-and-coming substitution of HR by machine insight is the following enormous test to be survived. Various researchers are zeroing in on the field of simulated intelligence, and this makes the exploration in the field of man-made intelligence rich and different. Search algorithms, knowledge graphs, natural language processing, expert systems, evolution algorithms, machine learning (ML), deep learning (DL), and other areas of AI research are just a few examples.[1]

##### **2.1.1 MACHINE LEARNING**

Machine Learning is a science which was found and created as a subfield of Artificial Intelligence during the 1950s. The initial steps of machine learning returns to the 1950s however there were no huge investigates and advancements on this science. In any case, during the 1990s, the explores on this field restarted, created and have reached right up to the present day. A science will work on more later on. The



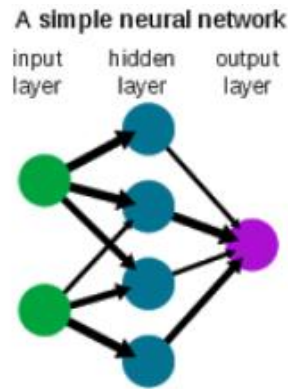
difficulty of analysing and processing the rapidly expanding data is the cause of this development. ML depends on the guideline of finding the best model for the new information among the past information because of this rising information. Consequently, ML explores will happen in lined up with the rising information. This exploration incorporates the historical backdrop of ML, the strategies utilized in ML, its application fields, and the investigates on this field. The point of this study is to communicate the information on ML, which has become exceptionally famous these days, and its applications to the analysts.[2]

### **2.1.2 DEEP LEARNING**

Deep learning is a procedure of ML in Artificial Intelligence. Deep learning in a refined "AI" calculation that far outperforms a significant parcel of its precursors in its abilities to see picture. In the field of machine learning and example recognition, deep learning is currently a very dynamic area of research. It has expanded epic victories in an extensive zone of uses, for instance, discourse acknowledgment, PC vision and normal language handling and various industry thing. neural network is utilized to execute the ML or to plan smart machines.[3]

### **2.2 NEURAL NETWORKS**

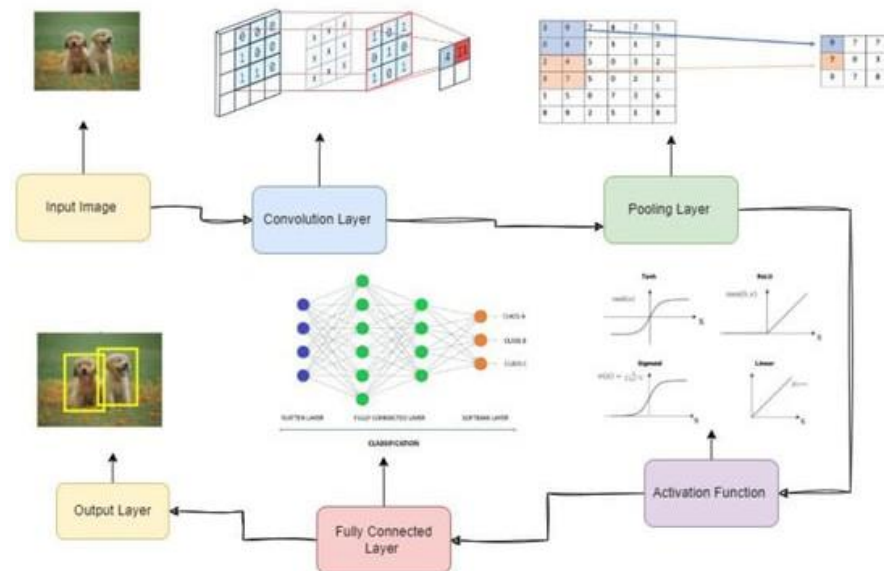
An Artificial Neural Network (ANN) is a data handling worldview that is roused by the way biological nervous systems, like the mind, process data. The information processing system's novel structure is the paradigm's most important feature. It is made out of countless profoundly interconnected handling components (neurons) working as one to take care of explicit issues. ANNs, similar to individuals, advance as a visual cue. Through a learning process, an ANN is set up for a specific use, like pattern recognition or data classification. Learning in organic frameworks includes acclimations to the synaptic associations that exist between the neurons. This is valid for ANNs also.[4]



**Figure 2.1 A Simple Neural Network**

## **2.3 CONVOLUTIONAL NEURAL NETWORKS**

Convolutional neural networks (CNNs) are artificial intelligence systems based on multi-layer neural networks that can identify, recognize, and classify objects as well as detect and segment objects in images. In fact, CNN or ConvNet is a popular discriminative deep learning architecture that could be learned directly from the input object without the obligation for human feature extraction.



**Figure 2.2 The CNN Components [8]**

This network is frequently used in visual identification, medical image analysis, image segmentation, NLP, and many other applications since it is specifically designed to deal with a range of 2D shapes. It is more effective than a regular network since it can automatically identify key elements from the input without the need for human participation.

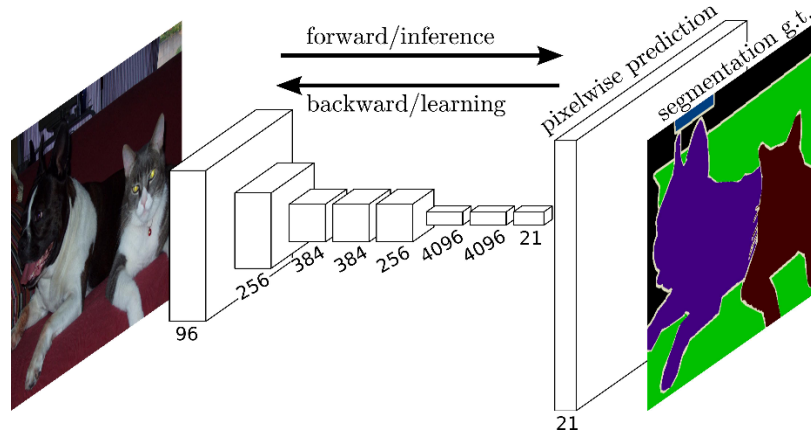
## **2.4 SEMANTIC SEGMENTATION**

Image segmentation has been one of the most challenging issues in computer vision for the past three decades. Picture division is not quite the same as picture grouping or article acknowledgment in that it isn't important to understand what the visual ideas or articles are in advance. To be more specific, an object classification will only be able to classify things that it has specific labels for, like a horse, a car, a house, or a dog. An ideal picture division calculation will likewise fragment obscure articles, that is to say, objects which are new or obscure. There are various applications where picture divisions could be utilized to work on existing calculations from social legacy conservation to picture duplicate location to satellite symbolism investigation to on-the-fly visual inquiry and human-PC connection. Having access to segmentations would make it possible to approach the issue on a semantic level in each of these applications. For instance, in satisfied based picture recovery, each picture could be sectioned as it is added to the data set. At the point when a question is handled, it very well may be fragmented and permit the client to inquiry for comparable sections in the data set — e.g., track down each of the bikes in the data set. In human-PC collaboration, all aspects of every video casing would be divided so the client could cooperate at a better level with different people and items in the climate. With regards to an air terminal, for instance, the security group is ordinarily inspired by any unattended stuff, some of which could hold hazardous materials. It would be gainful to make questions for all articles which were abandoned by a human.[5]

## **2.5 FULLY CONVOLUTIONAL NEURAL NETWORKS**

Fully Convolutional Neural Networks (FCNs) represent a seminal advancement in the field of computer vision, offering a powerful framework for pixel-level prediction tasks such as semantic segmentation and image-to-image translation. Unlike traditional convolutional neural networks (CNNs) that are designed for tasks like classification and object detection, FCNs are specifically tailored to handle tasks that require spatially dense outputs, making them well-suited for tasks where every pixel in an input image needs to be classified or transformed.

At the heart of FCNs lies their fully convolutional nature, which enables them to accept input images of arbitrary sizes and produce output maps of corresponding dimensions. This design choice contrasts with conventional neural networks that rely on fully connected layers, which inherently constrain input sizes. By replacing fully connected layers with convolutional layers, FCNs preserve spatial information throughout the network, allowing for end-to-end learning and prediction at the pixel level.

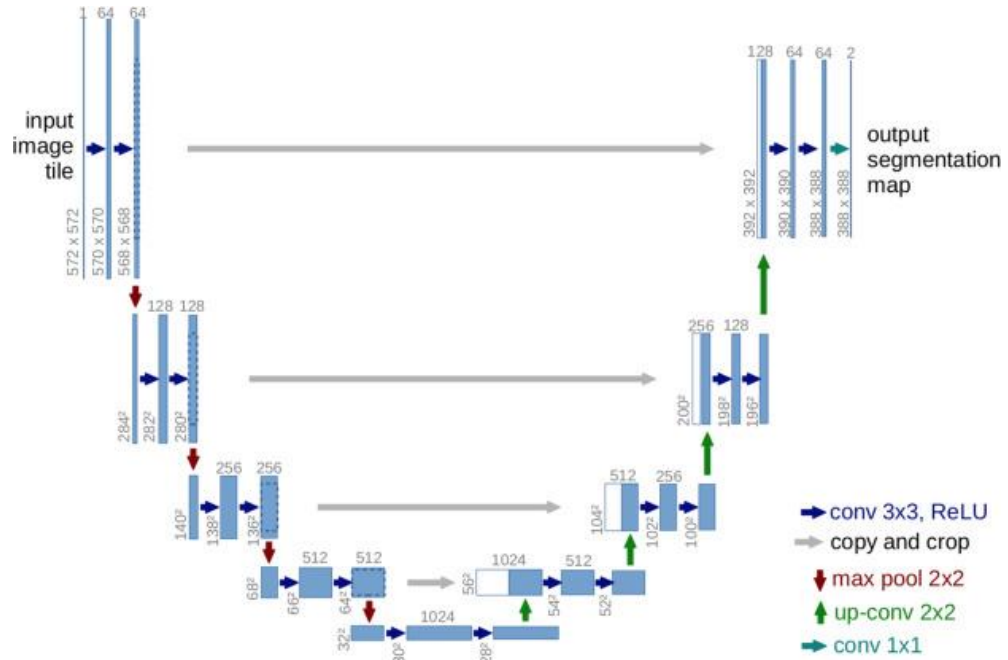


**Figure 2.3 Fully Convolutional Neural Network [16]**

The architecture of FCNs typically comprises an encoder-decoder structure, wherein the encoder extracts hierarchical features from the input image through a series of convolutional and pooling layers. These features capture increasingly abstract representations of the input image, enabling the network to learn rich feature representations. The decoder, on the other hand, up samples the feature maps to generate dense pixel-wise predictions. This up sampling process is often achieved through transposed convolutions or other up sampling techniques, which increase spatial resolution while preserving spatial relationships learned by the encoder. FCNs offer a flexible and modular architecture that can be adapted and extended to various tasks and domains. Researchers have developed numerous variants and extensions of FCNs to address specific challenges and applications, including instance segmentation, depth estimation, and image-to-image translation. These advancements underscore the versatility and utility of FCNs as a foundational framework for tackling diverse computer vision tasks, paving the way for further innovations in the field.

## 2.6 U-NETS

The U-Net architecture represents a notable variant of Fully Convolutional Networks (FCNs), primarily tailored for semantic segmentation tasks. Initially introduced by Ronneberger et al. in 2015 for biomedical image segmentation and localization, U-Nets offer distinctive advantages over conventional CNNs. Notably, they excel in providing both classification and localization outputs simultaneously, with localization referring to the labelling of individual pixels in an image with their respective classes. This capability is particularly advantageous for tasks requiring precise delineation of object boundaries. U-Nets demonstrate efficacy even with limited training data, yielding more refined segmentations compared to FCNs.



**Figure 2.4 U-Net Architecture [9]**

The architecture of U-Nets comprises a contracting path and an expanding path, reminiscent of traditional CNN designs. The contracting path consists of successive 3x3 convolutions and max-pooling operations with a stride of 2 for down sampling. These operations enhance the network's capacity to capture contextual information by progressively reducing spatial dimensions while increasing the number of feature channels. Conversely, the expanding path involves up sampling the feature map and applying 2x2 convolutions, followed by concatenation with the corresponding feature map from the contracting path. This fusion of high-resolution features from the contracting path with the up sampled output enables precise localization, a crucial aspect of semantic segmentation.

Unlike earlier approaches that suffer from drawbacks such as slow processing due to patch-based operations or compromises between localization accuracy and contextual information, U-Nets offer a more elegant solution. By replacing pooling operators with up sampling operators in successive layers, U-Nets incrementally increase the resolution of the output, facilitating both localization and contextual understanding. This seamless integration of contracting and expanding paths enables the network to effectively capture image context while preserving localization accuracy, crucial for tasks like semantic segmentation of satellite imagery.

U-Nets emerge as a highly versatile and efficient architecture for semantic segmentation tasks, particularly well-suited for applications involving aerial imagery. Their ability to balance context propagation and precise localization makes them a cornerstone in state-of-the-art methods for semantic segmentation, offering significant potential for advancing image analysis in various domains.[6][7]

### **2.6.1 RESULTS IN AERIAL IMAGERY**

The U-Net has been the primary architecture utilized for problems involving the segmentation of aerial imagery, despite the fact that it was initially intended for use in biomedical image segmentation. An illustration of this is through the 2017 Kaggle competition entitled the "DSTL Satellite Symbolism Element Discovery" challenge. Participants in this competition were required to use pixel-wise segmentation to classify ten distinct defined classes from high-resolution aerial photographs. There were 57 images in the dataset, with 25 training images and 32 test images. Due to its ability to combine lower-level feature maps with higher-level ones, which enables precise localization, all of the top entries in this competition utilized some form of the U-Net. U-Nets were likewise used in a paper composed by Khalel and El-Saban on the computerized pixel naming for both the INRIA Ethereal Symbolism dataset and the Massachusetts structures dataset. The authors cascade two U-Nets together in this instance, with the second serving as a post-processor for the first's output. Thusly, the stacked U-Net engineering beats cutting edge models on these datasets with a 74.55 mean convergence over association (mIoU) score and 96.05%-pixel exactness level for the INRIA Flying Symbolism dataset and a 0.9633 precision-recall breakeven.

## 2.6.2 EVALUATION METRICS

A proper metric that accurately describes the network's capacity to identify a class is required in order to measure the performance of various network architectures for semantic segmentation. The mIoU will serve as the primary metric, although other metrics will be investigated.

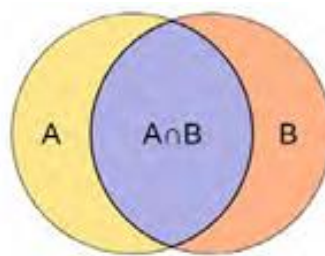
### 2.6.2.1 JACCARD COEFFICIENT

The Jaccard Similarity is a method for determining the similarity between two sets or a measure of the similarity between two asymmetric binary vectors. It is a common measure of proximity that is used to calculate how similar two things, like text documents, are. The index is between 0 and 1. Range more like 1 method greater closeness in two arrangements of information.

It is denoted by J and it is likewise alluded as Jaccard File, Jaccard Coefficient, Jaccard Divergence, and Jaccard Distance. It is utilized frequently in Text Mining, E-Commerce, Recommendation Systems, and other Data Science and Machine Learning applications. It is calculated by the formula:

**Jaccard Similarity** = (number of observations in both sets) / (number in either set) or mathematically,

$$J(A, B) = |A \cap B| / |A \cup B|$$



**Figure 2.5 Jaccard Coefficient Union Set**

**A- Ground Truth Data**

**B- Output Prediction**

The Jaccard Similarity Index of two datasets will be 1 if they share exactly the same members, and it will be 0 if there are no shared members. Jaccard Similarity will tell us how many features in the dataset are similar to each other.[10]

### **2.6.2.2 PIXEL ACCURACY**

An alternative measurement to assess a semantic division is to just report the percent of pixels in the picture which were accurately characterized. The pixel precision is regularly announced for each class independently as well as universally across all classes. While considering the per-class pixel exactness we're basically assessing a double veil; a true positive address a pixel that is accurately anticipated to have a place with the given class (as per the objective cover) though a true negative address a pixel that is accurately distinguished as not having a place with the given class.

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

This measurement can some of the time give deluding results when the class portrayal is little inside the picture, as the action will be one-sided in chiefly revealing how well we distinguish negative case (i.e. where the class is absent).

### **2.6.3 LOSS FUNCTION FOR SEGMENTATION**

In deep learning, the loss function serves as a critical component for training neural networks by quantifying the disparity between predicted outputs and ground truth labels. Commonly used in tasks like classification, regression, and segmentation, the loss function calculates the degree of error or deviation between predicted and actual values. For example, in classification tasks, cross-entropy loss measures the dissimilarity between predicted class probabilities and true class labels. Meanwhile, for regression tasks, mean squared error (MSE) or mean absolute error (MAE) are often employed to assess the disparity between predicted and true continuous values. Choosing an appropriate loss function is crucial as it directly impacts the optimization process and the ultimate performance of the model.

The choice of a suitable loss function depends on the specific characteristics of the task at hand and the desired properties of the model output. For instance, tasks requiring probabilistic interpretations may benefit from using log-likelihood-based loss functions such as negative log-likelihood loss. In contrast, tasks involving



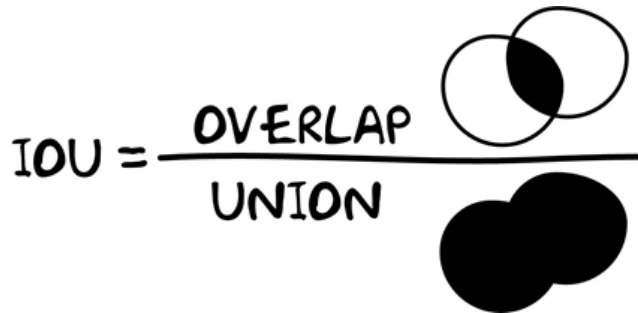
structured prediction, such as semantic segmentation, often employ specialized loss functions like Dice loss or Jaccard loss, which measure the similarity between predicted and ground truth segmentation masks. Ultimately, the selection of an appropriate loss function is a crucial consideration in the design and training of neural networks, directly influencing the model's ability to learn meaningful representations and make accurate predictions.

### **2.6.3.1 INTERSECTION OVER UNION AS A LOSS FUNCTION**

Intersection over Union (IoU) is a widely used evaluation metric for assessing the performance of semantic segmentation models. However, it cannot be directly utilized as a loss function due to its non-differentiable nature. The standard IoU calculation assumes binary output prediction masks consisting of ones and zeros. In reality, neural networks output probability arrays representing the likelihood of pixels belonging to specific classes. Consequently, an approximation of IoU, denoted as IoU', is necessary to make it differentiable and suitable for use as a loss function. IoU' is computed using probabilities and arithmetic operations, ensuring differentiability and enabling its incorporation into the training process.

$$IoU' = \frac{|T * P|}{|T + P - (T * P)|} = \frac{I}{U'}$$

IoU' is calculated as the ratio of the intersection of the ground truth (T) and predicted (P) masks to their union, averaged across the entire dataset. By using arithmetic operations instead of set operations, IoU' becomes differentiable, making it compatible with backpropagation during training.



**Figure 2.6 IoU Loss Function [12]**

The loss function based on IoU', denoted as LIoU, is formulated to be minimized during training. Specifically, LIoU is defined as 1 minus IoU', aiming to maximize the IoU value and thereby improve the segmentation accuracy of the model.

$$L_{IoU} = 1 - IoU'$$

In this project, the U-Net architecture and its variations are implemented for semantic segmentation tasks. The performance of these models is evaluated using the mean Intersection over Union (mIoU) metric, which serves as the baseline for all comparisons. To optimize the final models based on mIoU scores, a loss function derived from mIoU is utilized during training. Chapter 3 of the thesis will dive into the implementation details of the models, the datasets employed, and the preprocessing steps undertaken to prepare the input images and final test sets. This comprehensive approach ensures a thorough understanding and rigorous evaluation of the semantic segmentation models developed in the project.

### **2.6.3.2 BINARY CROSS ENTROPY**

Binary cross entropy loss, also known as log loss, stands as a fundamental choice for classification and segmentation tasks, serving as a baseline loss function in many studies. However, recent investigations, as highlighted in studies, have raised concerns regarding its indiscriminate usage without exploring more task-specific alternatives. The inherent nature of binary cross entropy loss, which equally penalizes both true positives and false negatives, poses challenges. This characteristic makes the resulting network susceptible to simplistic strategies, such as classifying all pixels based on the majority class, rather than discerning nuanced distinctions between classes. The formula is

$$L_{BCE} = \sum_x - (T_x \log(P_x) + (1 - T_x) \log(1 - P_x))$$

in which T is a single image of labels used as truth data,  $T_x$  is a single element of T, and  $P_x$  is a single element of the output prediction mask of the network. Given these limitations, there is a growing recognition of the necessity to develop loss functions that align more closely with the specific evaluation metrics desired for the task at hand. By incorporating task-specific evaluation metrics directly into the loss function,

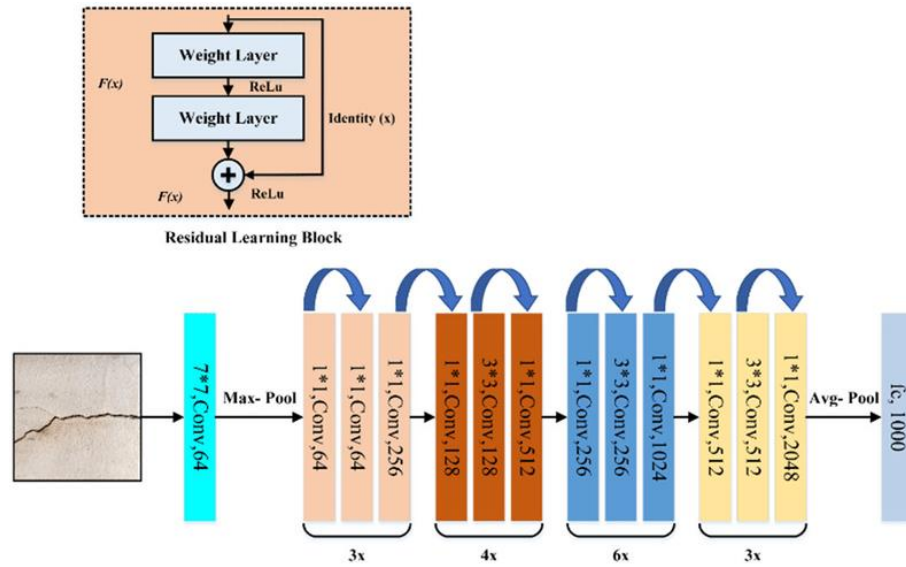
it becomes possible to steer the training process towards optimizing model performance based on relevant criteria. This approach not only enhances the effectiveness of the training process but also promotes the development of models that are better suited to the intricacies of the target task.

In the context of semantic segmentation, where pixel-wise classification accuracy is crucial, the choice of an appropriate loss function is paramount. While binary cross entropy loss provides a straightforward and widely adopted option, its tendency to encourage simplistic solutions underscores the importance of exploring alternative loss formulations tailored to the nuances of semantic segmentation tasks. By integrating task-specific evaluation metrics into the loss function design, researchers can pave the way for more nuanced and effective solutions that better capture the complexities of semantic segmentation tasks.

## **2.7 RESNET-50**

For benchmark, we have used ResNet-50. The ResNet family of architectures revolutionized the field of convolutional neural networks (CNNs) with its introduction of residual connections, which addressed the challenge of training very deep networks without encountering the vanishing gradient problem. The original ResNet model, ResNet-34, consisted of 34 weighted layers and marked a significant departure from previous architectures like VGGNet by incorporating shortcut connections.

These shortcut connections enable the network to bypass certain layers, effectively converting a conventional network into a residual network. Unlike VGGNet, where each convolutional layer has a  $3 \times 3$  filter, ResNet employs fewer filters, resulting in a less complex architecture. Despite its simplicity, ResNet achieves remarkable efficiency; for instance, a 34-layer ResNet achieves a performance of 3.6 billion FLOPs, significantly faster than VGGNet with 19.6 billion FLOPs.



**Figure 2.7 Architecture of ResNet-50[13]**

The design principles underlying ResNet are straightforward yet effective. Firstly, the number of filters in each layer remains constant, depending on the size of the output feature map. Secondly, if the size of the feature map is halved through operations like pooling, the number of filters is doubled to maintain the time complexity of each layer. These principles ensure that the network maintains a balance between depth and computational efficiency, facilitating the training of increasingly deep architectures.

ResNet-50, a variant of the original ResNet architecture, further extends this paradigm by introducing residual blocks with 50 weighted layers. This deeper architecture enhances the model's representational capacity, enabling it to capture more intricate patterns and features in the input data. Despite its increased depth, ResNet-50 continues to leverage the benefits of residual connections, ensuring stable training and improved performance on a wide range of computer vision tasks.

In summary, ResNet-50 stands as a testament to the power of residual connections in enabling the training of deep neural networks. By overcoming the challenges associated with vanishing gradients, ResNet-50 and its predecessors have significantly advanced the state-of-the-art in image classification, object detection, and semantic segmentation, among other tasks, making them indispensable tools in the arsenal of modern machine learning practitioners.







## **CHAPTER 3**

### **METHODOLOGY**

This section talks about the techniques utilized in this exploration to accomplish semantic division. Section 3.1 makes sense of the datasets utilized for this postulation and the important preprocessing expected to upgrade and run the models. Section 3.2 examines the different model structures utilized, alongside the assessment measurements and misfortune capabilities used to advance the last models for examination. At last, Section 3.3 will cover the handling for the last test sets and deciding the general precision for the results of the last models.

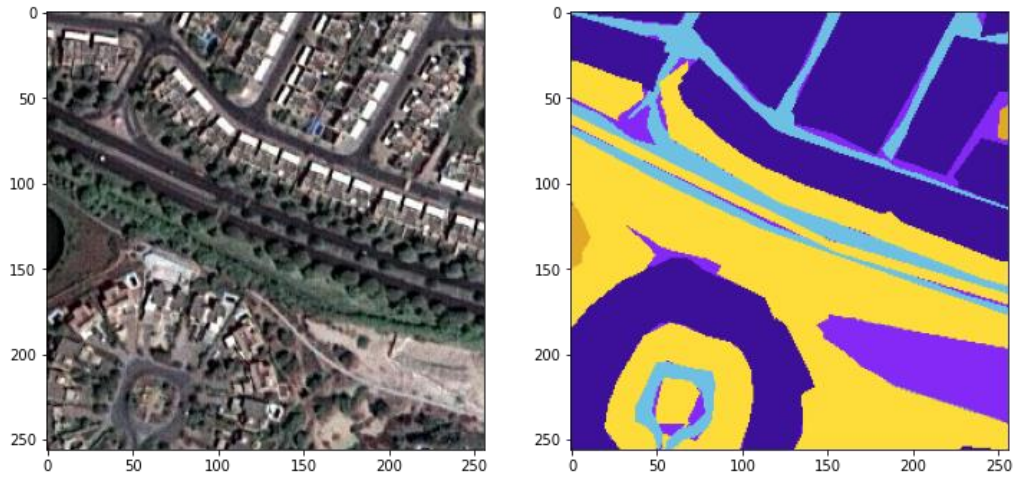
#### **3.1 DATASET**

The dataset utilized for training and validation in this project is the "Semantic Segmentation of Aerial Imagery," an openly accessible dataset published by Humans in the Loop in collaboration with the Mohammed Bin Rashid Space Centre in Dubai, UAE. Comprising 72 high-resolution satellite images of Dubai, each image is accompanied by its corresponding semantic segmentation masks. These masks delineate the spatial distribution of various land cover classes within the images, including water bodies, land areas, roads, buildings, vegetation, and a category labelled as unlabelled.

Name	R	G	B	Color
Building	60	16	152	
Land	132	41	246	
Road	110	193	228	
Vegetation	254	221	58	
Water	226	169	41	
Unlabeled	155	155	155	

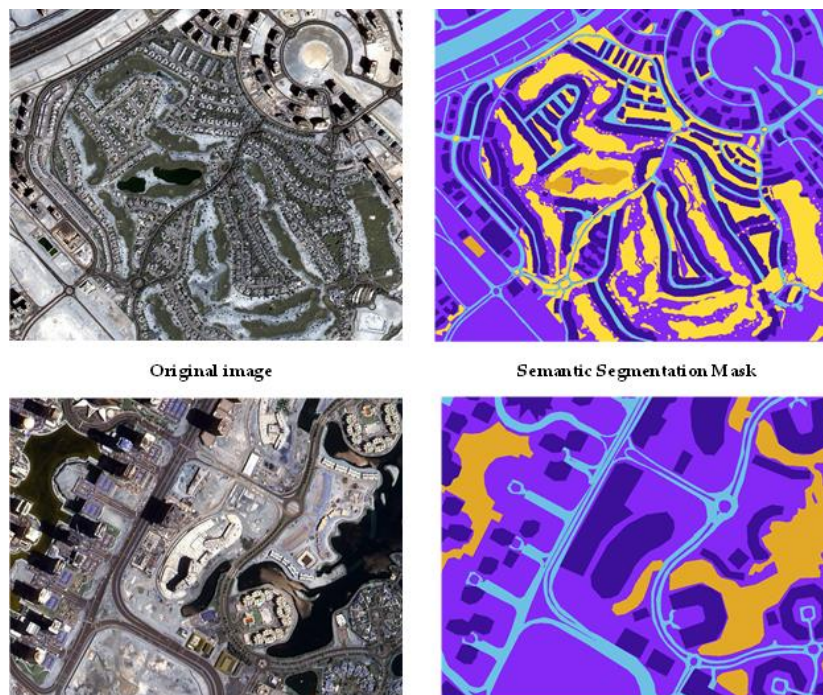
**Table 3.1 The labels and their corresponding colours**

A distinctive feature of this dataset is its segmentation into six distinct classes, enabling fine-grained analysis of land cover patterns and urban morphology within the urban landscape of Dubai. The segmentation masks provide detailed information about the spatial extent and distribution of each class, facilitating the training and evaluation of semantic segmentation models.



**Figure 3.1 Original Image vs Masked Image**

Notably, the segmentation of the satellite images was conducted by trainees affiliated with the Roia Foundation in Syria, underscoring the collaborative nature of this dataset creation process. This collaboration not only enhances the accessibility of the dataset but also contributes to capacity-building efforts in regions where expertise in image analysis and annotation may be limited.



**Figure 3.2 Original Image vs Masked Image**

For the purposes of this project, the training and validation subsets of the dataset consist of 65 and 7 images, respectively. This division ensures the availability of a sufficient number of samples for model training while reserving a separate subset for model validation and performance evaluation. By using this dataset, the project aims to develop and evaluate semantic segmentation models capable of accurately delineating land cover classes within satellite imagery, with a specific focus on roads and buildings for navigation-related applications.

### **3.2 PREPROCESSING**

Preprocessing plays a crucial role in ensuring the accuracy and effectiveness of computer vision systems, which encompass a wide spectrum of complexity and functionality. Typically, these systems comprise several key components aimed at acquiring, processing, and extracting meaningful information from visual data. Image acquisition serves as the foundational step, wherein data is gathered from various sensors such as cameras, light sensors, distance sensors, and radiographic devices. Depending on the sensor type, the resulting images may range from 2D to 3D representations or sequential image sequences, with each pixel encoding diverse physical measurements such as light intensity levels, absorption, reflection of electromagnetic waves, or distance.

Preprocessing is indispensable for preparing the acquired data for subsequent analysis by computer vision algorithms. This preprocessing phase involves several essential tasks aimed at enhancing the quality and utility of the input data. Firstly, it encompasses adjustments to the resolution and clarity of the images, ensuring consistency and correctness in the image coordinate system. Secondly, noise reduction techniques are applied to mitigate unwanted distortions or artifacts that may arise from sensor imperfections or environmental factors. By minimizing noise, the integrity of the data is preserved, thereby reducing the likelihood of false information being incorporated into subsequent analyses. Additionally, preprocessing aims to increase the variance within the data, thereby enhancing the discernibility of relevant features and patterns. By amplifying variations in the data, preprocessing ensures that

the desired information can be effectively extracted by subsequent computer vision algorithms.

Following preprocessing, the next critical step in the computer vision pipeline is feature extraction, wherein meaningful attributes or landmarks are identified and characterized within the pre-processed image data. Feature extraction encompasses a range of techniques aimed at capturing salient aspects of the visual content, such as colour, shape, texture, and spatial relationships. These features are crucial for subsequent analysis and interpretation tasks, facilitating tasks such as object recognition, classification, and scene understanding. Features extracted from the images are often categorized into global features, which encompass overarching characteristics such as colour distributions and overall shape properties, and local features, which capture more fine-grained details and spatial relationships within specific regions of interest.

### **3.3 MODEL ARCHITECTURE**

In our project, the U-Net architecture serves as the cornerstone for semantic segmentation of aerial imagery. The baseline U-Net architecture employed remains largely faithful to the original model proposed by Ronneberger et al. The contracting path of the network entails a series of 3x3 convolutions followed by rectified linear unit (ReLU) activation functions, facilitating feature extraction and nonlinear transformations. Subsequently, a 2x2 max pooling operation with a stride of 2 is applied for down sampling, effectively reducing spatial dimensions and increasing feature richness. Each down sampling step results in a doubling of features, which are subsequently halved in the expanding path through up sampling followed by 2x2 convolutions. Notably, the output from the expanding path is concatenated with the corresponding feature map from the contracting path, enabling the fusion of high-resolution features with contextual information. To ensure faster convergence and improved performance, batch normalization is incorporated between convolutions and ReLU activation functions. This modification enhances the stability of the training process and mitigates issues related to internal covariate shift. The final layer of the U-Net architecture utilizes a 1x1 convolution to map the feature vector to the desired number of classes, facilitating semantic segmentation.



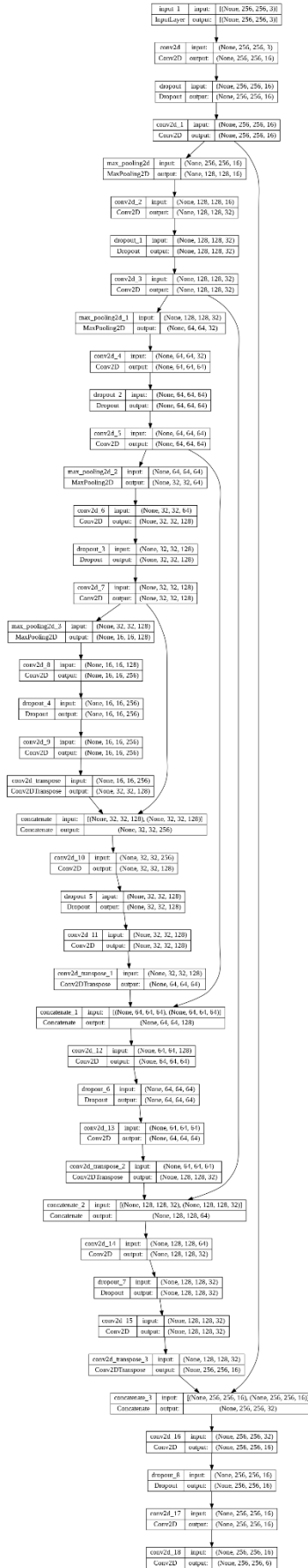


Figure 3.3 U-Net Model Architecture

In our project, the U-Net architecture serves as the baseline model for all comparisons. Additionally, we explore variations of the baseline model proposed in previous studies, including stacking multiple U-Nets to assess the impact on classification performance. Specifically, we consider configurations where two and three U-Nets are sequentially stacked, aiming to ascertain whether such stacking leads to a substantial enhancement in performance. The choice of optimizer and initializer is crucial for model training, with the Adam optimizer and He uniform variance scaling initializer selected for their proven effectiveness and widespread adoption in state-of-the-art U-Net models. By using the robustness and versatility of the U-Net architecture, our project endeavours to develop accurate and efficient semantic segmentation models for analysing aerial imagery, with a focus on applications such as road and building detection for navigation purposes.

### **3.4 PRETRAINED MODELS**

In Keras, a popular deep learning library, several pre-trained models are readily accessible for segmentation tasks, including ResNet and VGGNet. These models have been extensively trained on the ImageNet dataset, encompassing over 1000 classes, enabling them to capture a wide range of visual concepts and features. Using pre-trained weights from these models allows for efficient transfer learning, where the learned representations can be fine-tuned for specific segmentation tasks. In the context of semantic segmentation of aerial imagery, ResNet along with the U-Net architecture, will be evaluated to ascertain their efficacy and suitability for the task at hand. This comparative analysis aims to identify the most effective model for accurately delineating semantic features in aerial images, thereby advancing the capabilities of computer vision systems in this domain.

#### **3.4.1 RESNET-50**

The ResNet architecture introduced a groundbreaking approach to training deep neural networks by using residual functions that directly reference the inputs of previous layers. This innovative design mitigates the vanishing gradient problem, making it easier to optimize deeper networks. The original ResNet design, depicted in Figure 3.1, comprised 34 layers and laid the foundation for subsequent variants. In this thesis, the

ResNet50 model, an extended variation with 50 layers, is utilized to harness the increased representational capacity and learn more intricate features from the data. By using ResNet50, the aim is to enhance the model's performance in tasks such as image classification, object detection, and semantic segmentation, among others.



**Figure 3.4 VGG19 vs Plain Network with 34 parameter layers vs Residual network with 34 parameter layers [17]**

## **CHAPTER 4**

### **RESULTS AND ANALYSES**

This chapter presents a comprehensive analysis of the performance of each model developed in this study for semantic segmentation across four distinct datasets, including the amalgamated buildings dataset. Evaluation metrics, primarily based on Intersection over Union (IoU) and accuracy [8, 21, 20], are employed to gauge the efficacy of each model in delineating land cover classes within the satellite imagery. The training and testing outcomes of all five models are thoroughly examined, with subsequent subsections delving into the specific performance of each model on the individual datasets for comparative assessment. Alongside quantitative metrics, qualitative insights are provided through visual representations of segmentation outputs for a standardized image, aiding in the interpretation of model performance. Notably, all models underwent training for 1000 epochs with 500 batches of 32 images per epoch, necessitating significant computational resources and time. Consequently, certain models were terminated prematurely if training yielded negligible improvements, given the expansive range of model architectures and dataset variations explored in this thesis.

#### **4.1 TRAINING THE MODEL**

Training a machine learning model involves iteratively optimizing its parameters to minimize a chosen loss function, thereby improving its ability to make accurate predictions on unseen data. This process typically begins by initializing the model's parameters randomly or with pre-trained weights. Next, the model is presented with training data, consisting of input features and their corresponding target labels.

During training, the model's predictions are compared to the ground truth labels using the chosen loss function, which quantifies the disparity between predicted and actual values. Optimization algorithms such as stochastic gradient descent (SGD) or its variants adjust the model's parameters in the direction that reduces the loss, guided by gradients computed using backpropagation.

As training progresses through multiple iterations or epochs, the model learns to make better predictions by updating its parameters based on the observed discrepancies

between predictions and ground truth labels. The training process continues until a predefined stopping criterion is met, such as reaching a certain number of epochs or achieving satisfactory performance on a validation dataset.

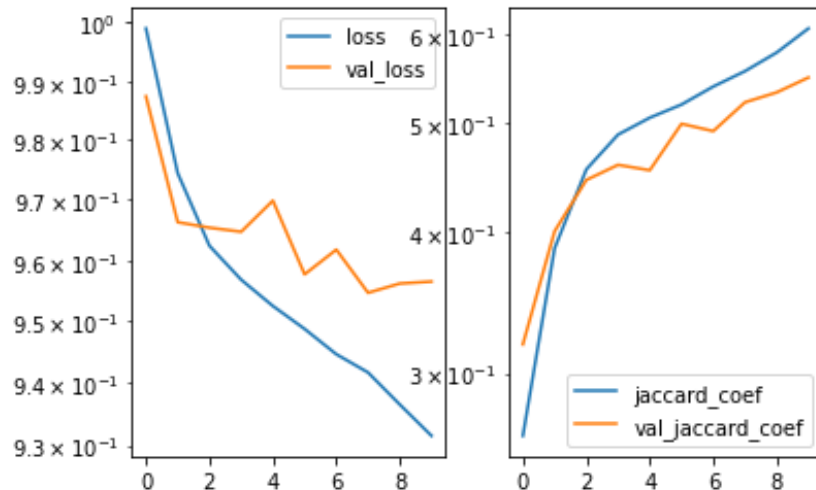
Once training is complete, the trained model can be evaluated on a separate test dataset to assess its performance on unseen examples. This evaluation provides insights into the model's ability to generalize to new data and informs decisions about its deployment in real-world applications.

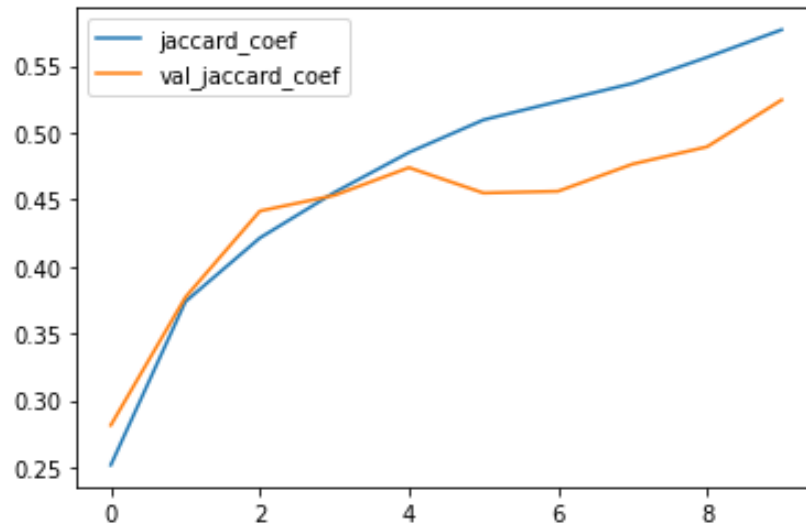
#### **4.1.1 U-NET TRAINING**

During the training phase of the U-Net model, several performance metrics are monitored to assess the convergence and effectiveness of the training process. Key metrics include training loss, validation loss, and accuracy, which provide insights into the model's learning dynamics and generalization ability.

Graphs of the Jaccard coefficient over epochs provide insights into U-Net's segmentation accuracy and convergence. Rising trends signify improving accuracy, while discrepancies between training and validation curves may indicate overfitting or underfitting. These graphs aid in model comparison and decision-making for architecture, hyperparameters, and training strategies.

Jaccard coefficient graphs offer practical insights into U-Net's effectiveness for semantic segmentation. Researchers can optimize model performance by analysing trends in these graphs, facilitating comparison between models and guiding decisions for improved satellite image analysis. By identifying patterns in Jaccard coefficient curves, researchers gain valuable understanding of model behaviour and can refine training strategies to enhance segmentation accuracy.





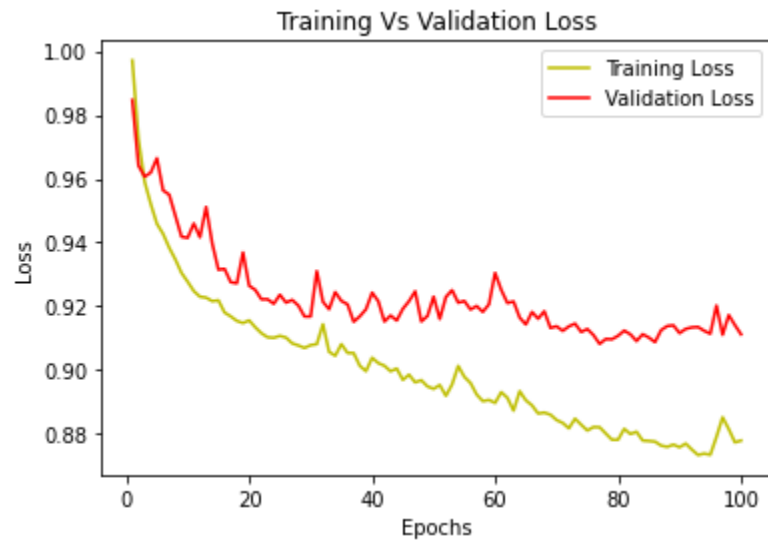
**Figure 4.1 Graph of Jaccard Coefficient**

The training loss, typically computed as the average loss over each batch of training data, reflects how well the model is fitting the training data. Concurrently, the validation loss, computed similarly but on a separate validation dataset, serves as a proxy for the model's generalization performance on unseen data. Monitoring the trend of both training and validation losses over epochs provides valuable insights into the model's training dynamics, including potential overfitting or underfitting.

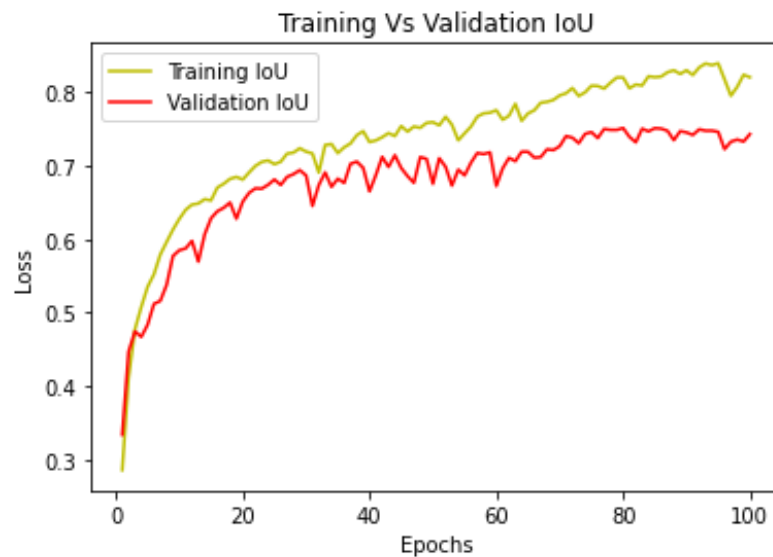
With loss metrics, accuracy and Intersection over Union (IoU) scores are commonly used to evaluate the segmentation performance of the model. Accuracy measures the percentage of correctly classified pixels, while IoU computes the overlap between predicted and ground truth segmentation masks. These metrics offer a more nuanced understanding of the model's segmentation performance across different land cover classes and can be visualized over epochs to track performance improvements or stagnation.

Graphs depicting the training and validation loss curves over epochs are instrumental in visualizing the convergence behaviour of the U-Net model. A typical graph would show a decreasing trend in both training and validation losses, indicating the model's ability to learn and generalize from the training data. Discrepancies between training and validation loss curves may suggest issues such as overfitting or inadequate model capacity.

Graphical representations of accuracy and IoU scores over epochs provide insights into the segmentation performance of the model across different phases of training. Ideally, these curves exhibit an increasing trend, indicative of the model's improving ability to accurately delineate land cover classes in satellite imagery.



**Figure 4.2 Training Loss vs Validation Loss**



**Figure 4.3 Training IoU vs. Validation IoU**

### 4.1.2 RESNET MODEL TRAINING

Training a ResNet model typically involves several key steps aimed at optimizing its performance and achieving high accuracy on the given task. Initially, the model's architecture, such as ResNet50, is selected based on the requirements of the specific problem. The model is then initialized with pre-trained weights, often obtained from training on large-scale datasets like ImageNet, which enables the network to capture general features from diverse images.

During training, the ResNet model undergoes an iterative process where it learns to minimize a chosen loss function by adjusting its parameters using optimization techniques like stochastic gradient descent (SGD) or its variants. The loss function used for training ResNet is typically selected based on the task at hand, such as cross-entropy loss for classification tasks or mean squared error (MSE) for regression tasks. As training progresses, the model's performance is monitored using metrics such as accuracy, which measures the proportion of correctly classified samples, or the loss value, which quantifies the disparity between predicted and true labels. These metrics are computed on a separate validation dataset to assess the model's generalization ability and prevent overfitting.

To further improve performance, techniques like learning rate scheduling, data augmentation, and regularization may be employed during training. Learning rate scheduling adjusts the learning rate over time to facilitate faster convergence and prevent oscillations, while data augmentation introduces variations to the training data to increase its diversity and robustness. Regularization techniques like dropout or weight decay help prevent overfitting by penalizing overly complex models.

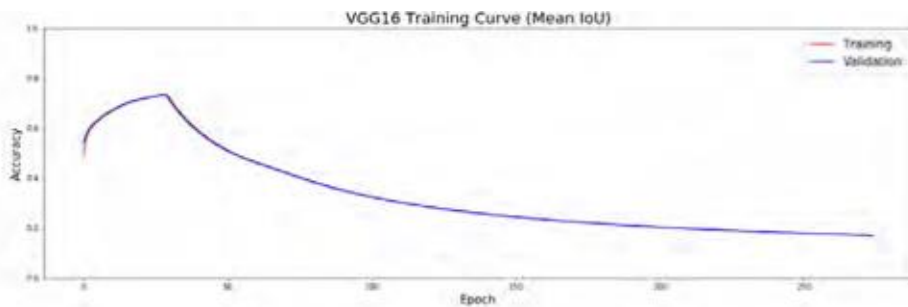


**Figure 4.4 RESNET-50 Training Curve (Accuracy)**





**Figure 4.5 RESNET-50 Training Curve (Loss)**



**Figure 4.6 RESNET-50 Training Curve (Mean IoU)**

Throughout the training process, hyperparameters such as batch size, learning rate, and optimizer choice are fine-tuned through experimentation to optimize performance. Once the model achieves satisfactory performance on the validation dataset, it is evaluated on a separate test dataset to assess its generalization to unseen data. Training a ResNet model involves careful selection of architecture, initialization with pre-trained weights, iterative optimization using appropriate loss functions and optimization techniques, and fine-tuning of hyperparameters to achieve high accuracy and performance on the target task.

## **4.2 TESTING THE MODEL**

Testing a trained machine learning model involves assessing its performance on unseen data to evaluate its ability to generalize beyond the training set. This process typically begins by feeding the test data into the trained model and obtaining predictions for the input samples. These predictions are then compared to the ground truth labels or targets associated with the test data.

Metrics such as accuracy, precision, recall, F1-score, or area under the receiver operating characteristic curve (ROC-AUC) are computed to quantify the model's performance on the test set. These metrics provide insights into different aspects of the model's behaviour, such as its overall correctness, ability to correctly identify positive cases (precision), ability to capture all positive cases (recall), and the balance between precision and recall (F1-score).

Testing the model serves as a critical step in the machine learning pipeline, providing validation of the model's effectiveness and guiding further iterations or improvements. It ensures that the model's performance meets the desired criteria and instils confidence in its ability to make accurate predictions in real-world scenarios.

#### **4.2.1 U-NET MODEL PREDICTION**

This section showcases a series of images depicting the original satellite imagery, corresponding ground truth labels, and predictions generated by the trained U-Net model on the test dataset. Each set of images provides a visual representation of the model's performance in semantic segmentation, illustrating its ability to accurately classify different land cover classes within the satellite images.

##### **Original Satellite Imagery:**

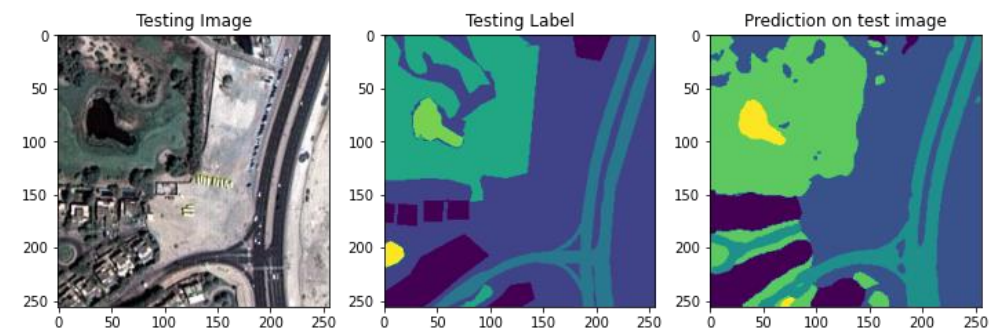
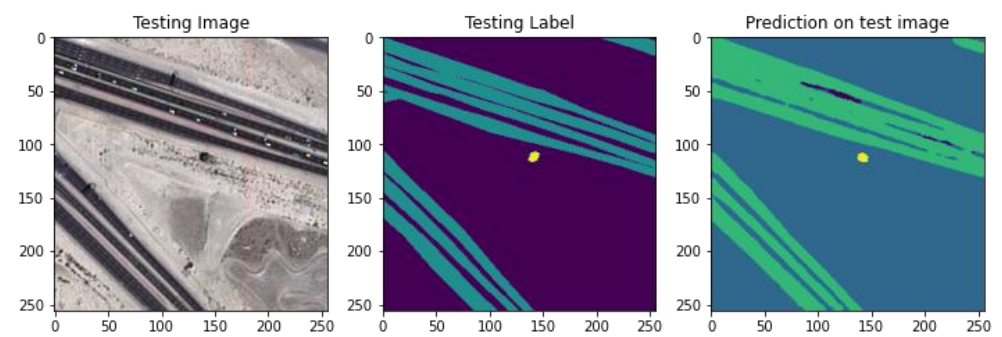
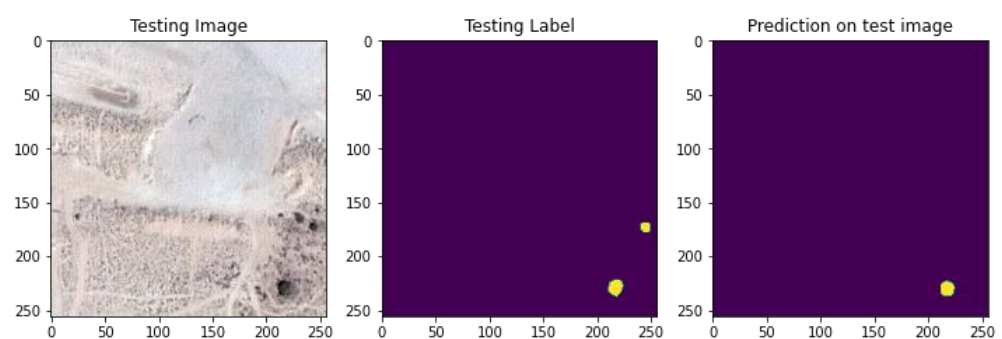
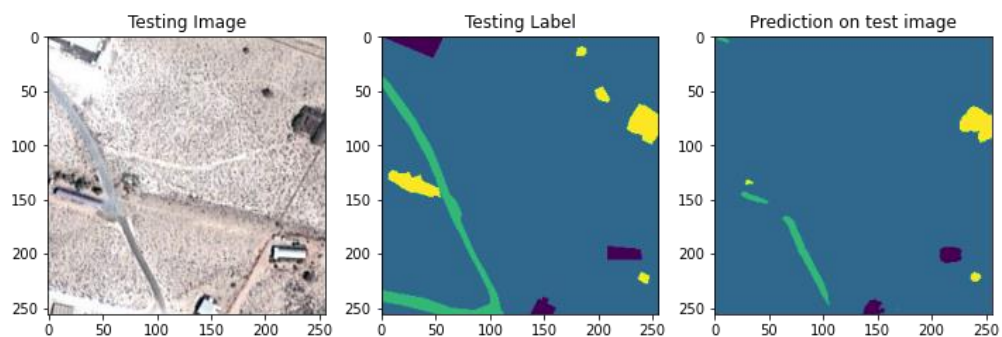
The original satellite images serve as the input data for the semantic segmentation task. These high-resolution images capture various geographical features and land cover types, ranging from urban areas and vegetation to water bodies and infrastructure.

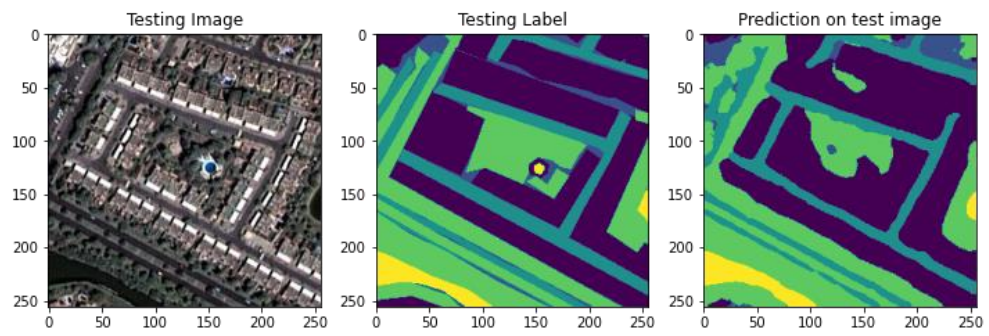
##### **Ground Truth Labels:**

The ground truth label images depict manually annotated segmentation masks, where each pixel is assigned a specific class label corresponding to the land cover type it represents. These labels serve as the reference standard for evaluating the accuracy of the model's predictions.

##### **Model Predictions:**

The model predictions showcase the segmentation masks generated by the trained U-Net model on the test dataset. These predictions highlight the model's ability to automatically classify pixels within the satellite images into different land cover classes, providing insights into its segmentation accuracy and performance.

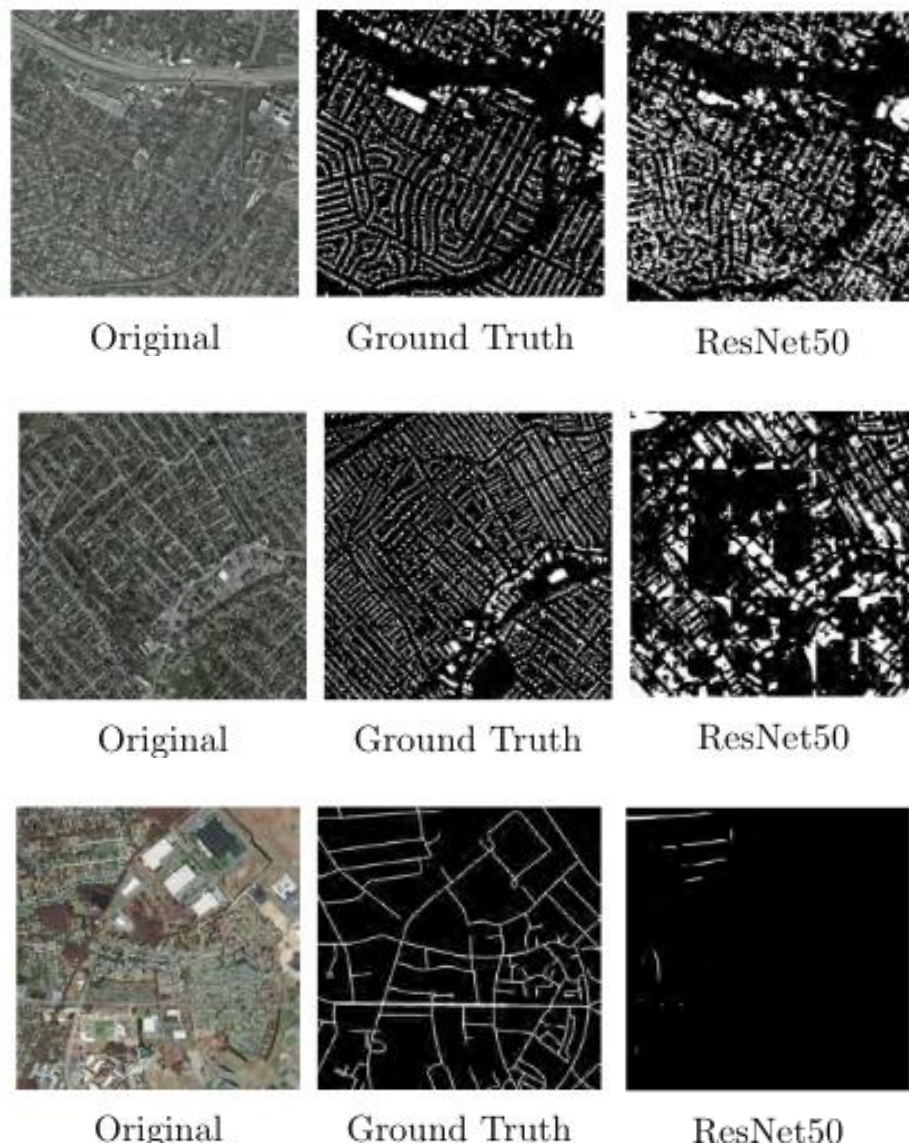




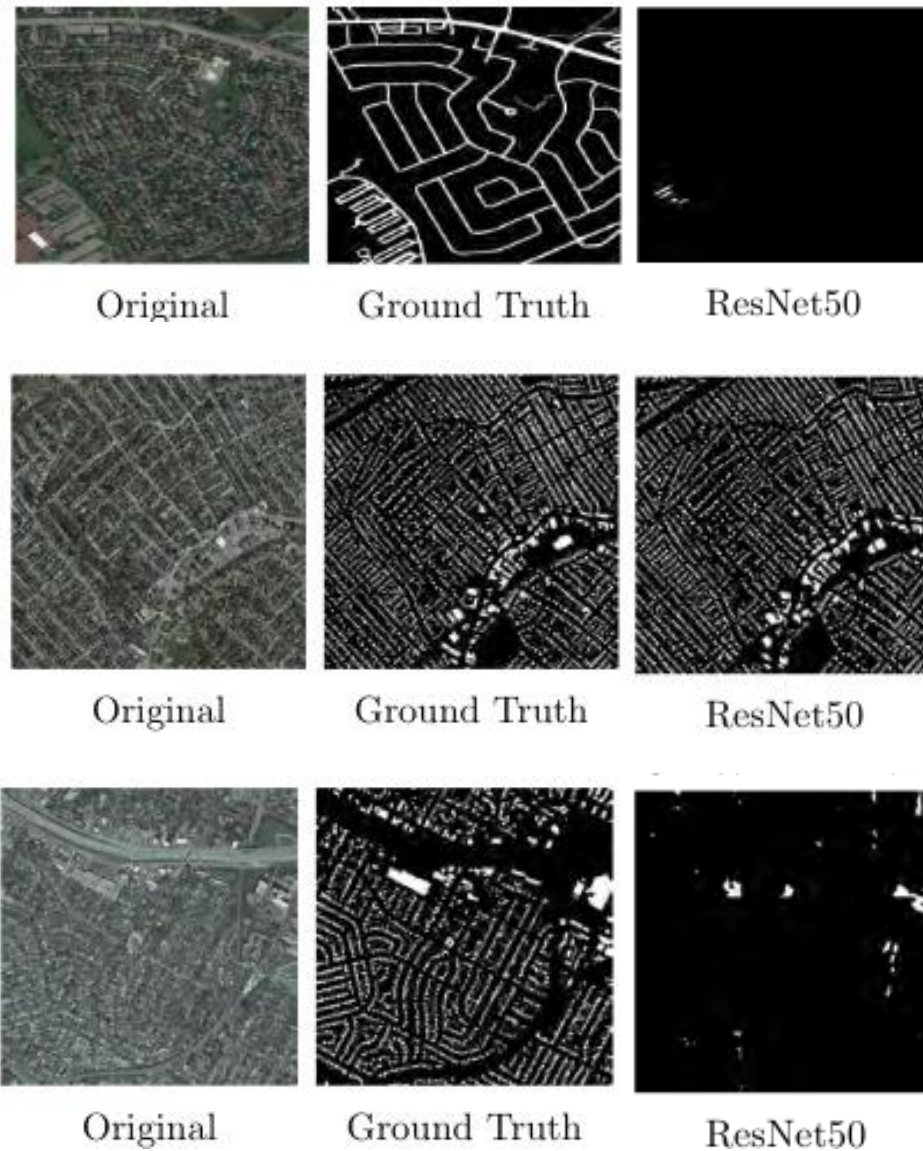
**Figures 4.7 Test Image vs. Test Label vs. Prediction (U-Net)**

### **4.2.2 RESNET-50 MODEL PREDICTION**

Following are the prediction made by ResNet-50 Model.







**Figures 4.8 Test Image vs. Test Label vs. Prediction (RESNET-50)**

### **4.3 SUMMARY**

The results obtained from testing the U-Net models on both building and road datasets exhibited considerable variability. Despite expectations of improved segmentation performance with the addition of U-Net levels, the findings suggest otherwise. While the mean Intersection over Union (mIoU) values initially indicate a consistent increase with additional U-Net levels, contradicting results are observed, where a consistent decrease is noted.

Further analysis reveals inconsistent trends across different U-Net levels and datasets. For instance, an initial decrease with the 2-level U-Net followed by the highest overall mIoU with the 3-level U-Net. An initial increase with the 2-level U-Net and the lowest mIoU value with the 3-level U-Net. These variations in performance, seemingly unrelated to additional U-Net levels or dataset characteristics, render the value of adding levels inconclusive.

The testing underscores that U-Nets may not universally optimize segmentation from an aerial perspective. The comparison of models indicates that only the 2-level U-Net model surpassed previous research in one metric, accuracy, albeit minimally. Consequently, rather than indiscriminately adopting U-Nets or incorporating additional levels for marginal performance gains, a more effective approach may involve developing tailored encoders and decoders specific to the dataset. While the potential for a universal network for roads and buildings exists in the future, the current scarcity of labelled data and benchmarks impedes definitive conclusions regarding the future role of U-Nets in segmentation tasks.

## **CHAPTER 5**

### **CONCLUSION & FUTURE WORK**

This research commenced on training variations of the U-Net architecture across multiple datasets while also amalgamating two publicly available datasets to facilitate semantic segmentation. While the amalgamated dataset didn't yield conclusive results for enabling networks to discern invariances between datasets, the models' performance serves as a foundational reference point for future endeavours. Despite being outperformed by existing architectures, the U-Net showcased commendable performance in segmenting buildings and roads, solidifying its status as a reliable benchmark for comparative analyses.

The refinement of hyperparameters, including alterations to initializers and optimizers, led to discernible enhancements in output predictions, underscoring the significance of meticulous model configuration adjustments. Challenges pertaining to edge predictions, especially in high-resolution imagery, underscored the necessity for innovative solutions like the overlap of test images or the adoption of sliding window techniques to mitigate such issues effectively.

Future avenues of exploration encompass the exploration of larger batch sizes and smaller image patches to potentially amplify model performance, alongside delving into the integration of generative adversarial networks to foster superior segmentation outcomes. Exploring the integration of U-Nets into real-time applications across diverse domains could yield significant advancements, offering the potential to enhance decision-making processes and operational efficiency in various contexts.

In essence, this research lays a comprehensive foundation for ongoing endeavours aimed at refining U-Net architectures to address complex real-world challenges in semantic segmentation and autonomous navigation, positioning these methodologies as pivotal tools in the arsenal of contemporary machine learning and computer vision practitioners.

## **CHAPTER 6**

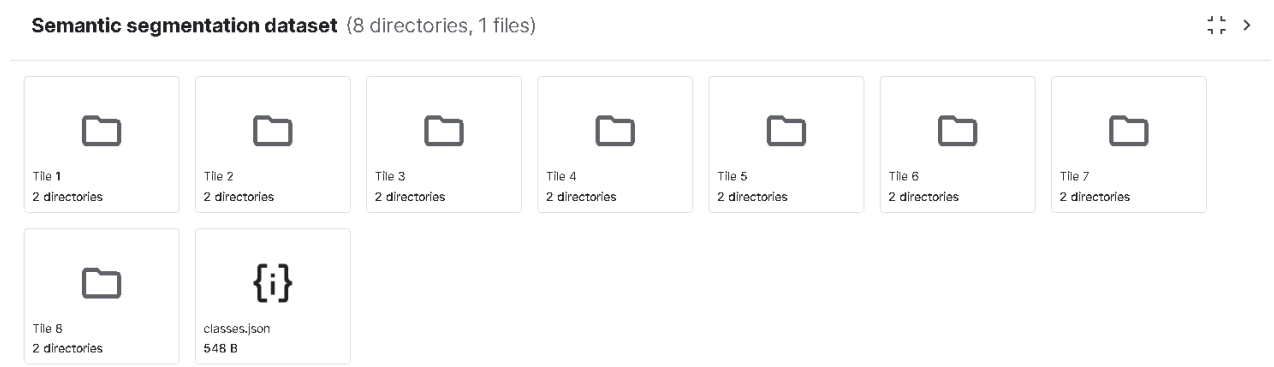
## **APPENDICES**

### **6.1 ABOUT THE DATASET**

The dataset used in this project is an official dataset available in Kaggle from the Mohammed Bin Rashid Space Centre (MBRSC) in Dubai, UAE. This dataset comprises high-resolution satellite images capturing various landscapes and urban areas within Dubai. It includes diverse categories such as buildings, roads, vegetation, water bodies, and other land cover types. The dataset is particularly valuable for its detailed and accurate imagery, making it ideal for tasks such as land cover classification, object detection, and segmentation using advanced deep learning models like U-Net. The high quality and comprehensive coverage of this dataset enable robust analysis and application in various remote sensing and urban planning projects. The link to the dataset is given below:

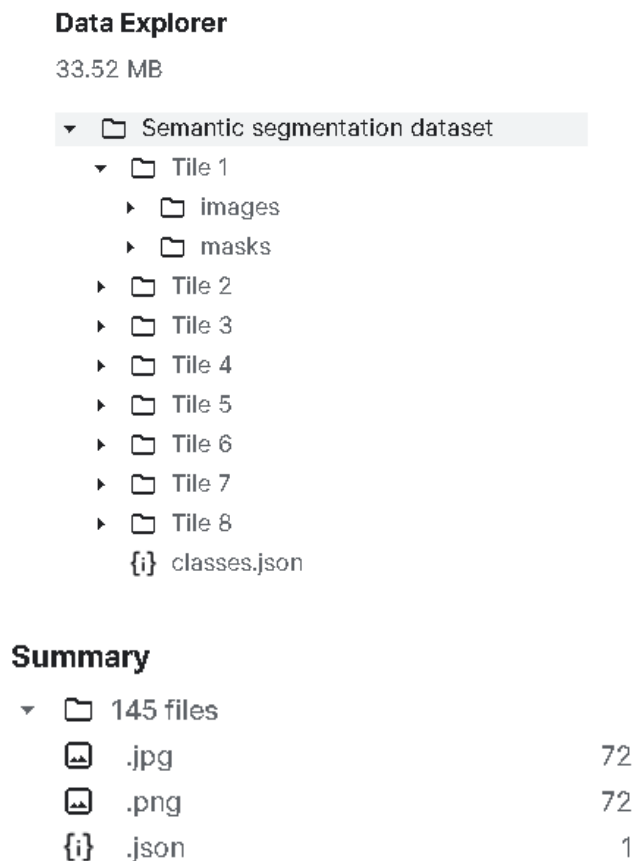
<https://www.kaggle.com/datasets/humansintheloop/semantic-segmentation-of-aerial-imagery>

Following is the summary of the dataset from Kaggle:



**Figure 6.1 File Structure of Dataset**





**Figure 6.2 Dataset Summary**

## **6.2 ASSIGNING COLOURS TO THE LABELS**

The following code converts hexadecimal colour codes, commonly used in web design, into RGB (Red, Green, Blue) values and stores them as NumPy arrays for six different land cover classes. For each class, the code removes the leading hash symbol from the hex code, then slices the string into its red, green, and blue components. Each component is converted from hex to a decimal integer, and these integers are collected into a tuple, which is then converted into a NumPy array. This process is repeated for the classes "building", "land", "road", "vegetation", "water", and "unlabelled", and the resulting RGB arrays are printed to the console.

```

class_building = '#3C1098'
class_building = class_building.lstrip('#')
class_building = np.array(tuple(int(class_building[i:i+2], 16) for i in (0,2,4)))
print(class_building)

class_land = '#8429F6'
class_land = class_land.lstrip('#')
class_land = np.array(tuple(int(class_land[i:i+2], 16) for i in (0,2,4)))
print(class_land)

class_road = '#6EC1E4'
class_road = class_road.lstrip('#')
class_road = np.array(tuple(int(class_road[i:i+2], 16) for i in (0,2,4)))
print(class_road)

class_vegetation = '#FEDD3A'
class_vegetation = class_vegetation.lstrip('#')
class_vegetation = np.array(tuple(int(class_vegetation[i:i+2], 16) for i in (0,2,4)))
print(class_vegetation)

class_water = '#E2A929'
class_water = class_water.lstrip('#')
class_water = np.array(tuple(int(class_water[i:i+2], 16) for i in (0,2,4)))
print(class_water)

class_unlabeled = '#9B9B9B'
class_unlabeled = class_unlabeled.lstrip('#')
class_unlabeled = np.array(tuple(int(class_unlabeled[i:i+2], 16) for i in (0,2,4)))
print(class_unlabeled)

```

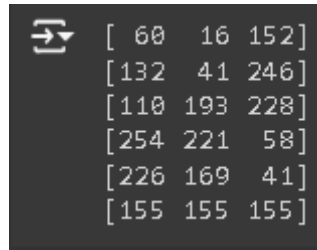
**Figure 6.3 Labels and Colours**

```

[ ] def rgb_to_label(label):
    label_segment = np.zeros(label.shape, dtype=np.uint8)
    label_segment[np.all(label == class_water, axis=-1)] = 0
    label_segment[np.all(label == class_land, axis=-1)] = 1
    label_segment[np.all(label == class_road, axis=-1)] = 2
    label_segment[np.all(label == class_building, axis=-1)] = 3
    label_segment[np.all(label == class_vegetation, axis=-1)] = 4
    label_segment[np.all(label == class_unlabeled, axis=-1)] = 5
    #print(label_segment)
    label_segment = label_segment[:, :, 0]
    #print(label_segment)
    return label_segment

```

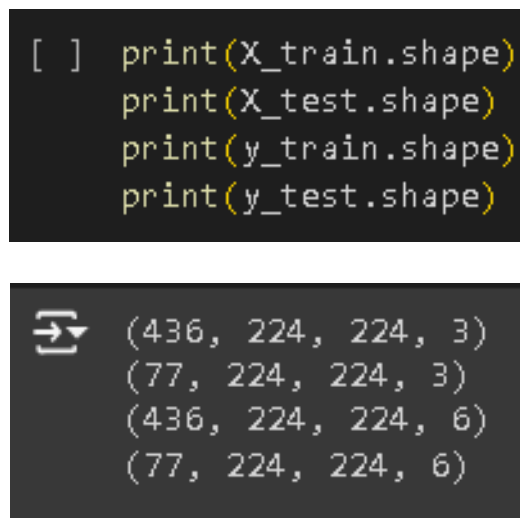
**Figure 6.4 Function assigning colours to labels**



**Figure 6.5 RGB arrays**

### **6.3 TEST AND TRAIN DATASET RATIO**

The dataset is divided into training and testing sets, with 436 images and corresponding masks for training, and 77 images and corresponding masks for testing. Each image is 224x224 pixels in size with 3 colour channels (RGB), and each mask has 6 channels, likely representing different classes for segmentation purposes. This division ensures that our model can be trained on a substantial number of samples and then evaluated on a separate set to assess its performance.



**Figure 6.6 Train & Test size**

### **6.4 IMAGE DIMENSIONS**

Each image in the dataset has a shape of 224x224 pixels with 3 channels representing the RGB colour space. The corresponding masks for these images have the same spatial dimensions (224x224 pixels) but contain 6 channels, each representing a different class for segmentation purposes. This structure is essential for training and evaluating deep learning models for image segmentation tasks.

```
[ ] print(image_height)
    print(image_width)
    print(image_channels)
    print(total_classes)

⇒ 224
   224
   3
   6
```

**Figure 6.7 Image Dimensions**

## **6.5 JACCARD COEFFICIENT IMPLEMENTATION**

The `jaccard_coef` function computes the Jaccard coefficient (IoU) between the true and predicted segmentation masks. It flattens the input tensors, calculates the intersection of the predicted and true values, and then uses this intersection to compute the Jaccard coefficient. The coefficient provides a measure of how similar the predicted segmentation is to the ground truth, with a value ranging from 0 to 1, where 1 indicates perfect overlap. The use of a smoothing factor (1.0) helps to avoid division by zero errors.

```
[ ] def jaccard_coef(y_true, y_pred):
    y_true_flatten = K.flatten(y_true)
    y_pred_flatten = K.flatten(y_pred)
    intersection = K.sum(y_true_flatten * y_pred_flatten)
    final_coef_value = (intersection + 1.0) / (K.sum(y_true_flatten) + K.sum(y_pred_flatten) - intersection + 1.0)
    return final_coef_value
```

**Figure 6.8 Jaccard Coefficient Function**

## **6.6 U-NET MODEL IMPLEMENTATION**

The `multi_unet_model` function constructs a U-Net model tailored for multi-class image segmentation tasks. It starts with an input layer and follows the typical U-Net architecture, consisting of a contracting path (encoder) that downsamples the input using convolutional and max-pooling layers, and an expansive path (decoder) that upsamples the features using transposed convolutions and concatenates them with corresponding layers from the encoder. The model includes dropout layers to prevent

overfitting and uses ReLU activations with He-normal initialization for the convolutional layers. The final output layer uses a softmax activation function to produce class probabilities for each pixel, catering to `n\_classes` segmentation classes. The constructed model is then returned.

```
def multi_unet_model(n_classes=5, image_height=256, image_width=256, image_channels=1):

    inputs = Input((image_height, image_width, image_channels))

    source_input = inputs

    c1 = Conv2D(16, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(source_input)
    c1 = Dropout(0.2)(c1)
    c1 = Conv2D(16, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c1)
    p1 = MaxPooling2D((2,2))(c1)

    c2 = Conv2D(32, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(p1)
    c2 = Dropout(0.2)(c2)
    c2 = Conv2D(32, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c2)
    p2 = MaxPooling2D((2,2))(c2)

    c3 = Conv2D(64, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(p2)
    c3 = Dropout(0.2)(c3)
    c3 = Conv2D(64, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c3)
    p3 = MaxPooling2D((2,2))(c3)

    c4 = Conv2D(128, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(p3)
    c4 = Dropout(0.2)(c4)
    c4 = Conv2D(128, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c4)
    p4 = MaxPooling2D((2,2))(c4)

    c5 = Conv2D(256, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(p4)
    c5 = Dropout(0.2)(c5)
    c5 = Conv2D(256, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c5)

    u6 = Conv2DTranspose(128, (2,2), strides=(2,2), padding="same")(c5)
    u6 = concatenate([u6, c4])
    c6 = Conv2D(128, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(u6)
    c6 = Dropout(0.2)(c6)
    c6 = Conv2D(128, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c6)

    u7 = Conv2DTranspose(64, (2,2), strides=(2,2), padding="same")(c6)
    u7 = concatenate([u7, c3])
    c7 = Conv2D(64, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(u7)
    c7 = Dropout(0.2)(c7)
    c7 = Conv2D(64, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c7)

    u8 = Conv2DTranspose(32, (2,2), strides=(2,2), padding="same")(c7)
    u8 = concatenate([u8, c2])
    c8 = Conv2D(32, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(u8)
    c8 = Dropout(0.2)(c8)
    c8 = Conv2D(32, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c8)

    u9 = Conv2DTranspose(16, (2,2), strides=(2,2), padding="same")(c8)
    u9 = concatenate([u9, c1], axis=3)
    c9 = Conv2D(16, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(u9)
    c9 = Dropout(0.2)(c9)
    c9 = Conv2D(16, (3,3), activation="relu", kernel_initializer="he_normal", padding="same")(c9)

    outputs = Conv2D(n_classes, (1,1), activation="softmax")(c9)

    model = Model(inputs=[inputs], outputs=[outputs])
    return model
```

Figure 6.9 U-Net model implementation

## 6.7 GENERATING LOSS FUNCTION

This code segment generates a custom loss function for multi-class image segmentation, prioritizing the Dice loss but incorporating the Focal loss to address class imbalance and focus on hard examples. The class weights ensure balanced contribution from each class to the loss computation, promoting robustness and accuracy in the segmentation model.

```

v Generating Loss Function

• dice loss > Focal Loss > Total Loss
• Total Loss = (Dice loss + (1*Focal Loss))

[ ] weights = [0.1666, 0.1666, 0.1666, 0.1666, 0.1666, 0.1666]

[ ] dice_loss = sm.losses.DiceLoss(class_weights = weights)

[ ] focal_loss = sm.losses.CategoricalFocalLoss()

[ ] total_loss = dice_loss + (1 * focal_loss)
```

**Figure 6.10 Loss Function Implementation**

## 6.8 MODEL SUMMARY

Model: "model"			
Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 224, 224, 3)]	0	[]
conv2d (Conv2D)	(None, 224, 224, 16)	448	['input_1[0][0]']
dropout (Dropout)	(None, 224, 224, 16)	0	['conv2d[0][0]']
conv2d_1 (Conv2D)	(None, 224, 224, 16)	2320	['dropout[0][0]']
max_pooling2d (MaxPooling2D)	(None, 112, 112, 16)	0	['conv2d_1[0][0]']
conv2d_2 (Conv2D)	(None, 112, 112, 32)	4640	['max_pooling2d[0][0]']
dropout_1 (Dropout)	(None, 112, 112, 32)	0	['conv2d_2[0][0]']
conv2d_3 (Conv2D)	(None, 112, 112, 32)	9248	['dropout_1[0][0]']
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 32)	0	['conv2d_3[0][0]']
conv2d_4 (Conv2D)	(None, 56, 56, 64)	18496	['max_pooling2d_1[0][0]']
dropout_2 (Dropout)	(None, 56, 56, 64)	0	['conv2d_4[0][0]']
conv2d_5 (Conv2D)	(None, 56, 56, 64)	36928	['dropout_2[0][0]']
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 64)	0	['conv2d_5[0][0]']
conv2d_6 (Conv2D)	(None, 28, 28, 128)	73856	['max_pooling2d_2[0][0]']
dropout_3 (Dropout)	(None, 28, 28, 128)	0	['conv2d_6[0][0]']
conv2d_7 (Conv2D)	(None, 28, 28, 128)	147584	['dropout_3[0][0]']
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 128)	0	['conv2d_7[0][0]']
conv2d_8 (Conv2D)	(None, 14, 14, 256)	295168	['max_pooling2d_3[0][0]']
dropout_4 (Dropout)	(None, 14, 14, 256)	0	['conv2d_8[0][0]']
conv2d_9 (Conv2D)	(None, 14, 14, 256)	590080	['dropout_4[0][0]']
conv2d_transpose (Conv2DTranspose)	(None, 28, 28, 128)	131200	['conv2d_9[0][0]']
concatenate (Concatenate)	(None, 28, 28, 256)	0	['conv2d_transpose[0][0]', 'conv2d_7[0][0]']
conv2d_10 (Conv2D)	(None, 28, 28, 128)	295040	['concatenate[0][0]']
dropout_5 (Dropout)	(None, 28, 28, 128)	0	['conv2d_10[0][0]']
conv2d_11 (Conv2D)	(None, 28, 28, 128)	147584	['dropout_5[0][0]']

conv2d_transpose_1 (Conv2D Transpose)	(None, 56, 56, 64)	32832	['conv2d_11[0][0]']
concatenate_1 (Concatenate )	(None, 56, 56, 128)	0	['conv2d_transpose_1[0][0]', 'conv2d_5[0][0]']
conv2d_12 (Conv2D)	(None, 56, 56, 64)	73792	['concatenate_1[0][0]']
dropout_6 (Dropout)	(None, 56, 56, 64)	0	['conv2d_12[0][0]']
conv2d_13 (Conv2D)	(None, 56, 56, 64)	36928	['dropout_6[0][0]']
conv2d_transpose_2 (Conv2D Transpose)	(None, 112, 112, 32)	8224	['conv2d_13[0][0]']
concatenate_2 (Concatenate )	(None, 112, 112, 64)	0	['conv2d_transpose_2[0][0]', 'conv2d_3[0][0]']
conv2d_14 (Conv2D)	(None, 112, 112, 32)	18464	['concatenate_2[0][0]']
dropout_7 (Dropout)	(None, 112, 112, 32)	0	['conv2d_14[0][0]']
conv2d_15 (Conv2D)	(None, 112, 112, 32)	9248	['dropout_7[0][0]']
conv2d_transpose_3 (Conv2D Transpose)	(None, 224, 224, 16)	2064	['conv2d_15[0][0]']
concatenate_3 (Concatenate )	(None, 224, 224, 32)	0	['conv2d_transpose_3[0][0]', 'conv2d_1[0][0]']
conv2d_16 (Conv2D)	(None, 224, 224, 16)	4624	['concatenate_3[0][0]']
dropout_8 (Dropout)	(None, 224, 224, 16)	0	['conv2d_16[0][0]']
conv2d_17 (Conv2D)	(None, 224, 224, 16)	2320	['dropout_8[0][0]']
conv2d_18 (Conv2D)	(None, 224, 224, 6)	102	['conv2d_17[0][0]']

**Figure 6.11 Model Summary**

Trainable Parameters vs. Non-Trainable Parameters:

```
=====
Total params: 1941190 (7.41 MB)
Trainable params: 1941190 (7.41 MB)
Non-trainable params: 0 (0.00 Byte)
```

**Figure 6.12 Trainable vs. Non-Trainable Parameters**



## 6.9 TRAINING AND VALIDATION LOSS CURVE

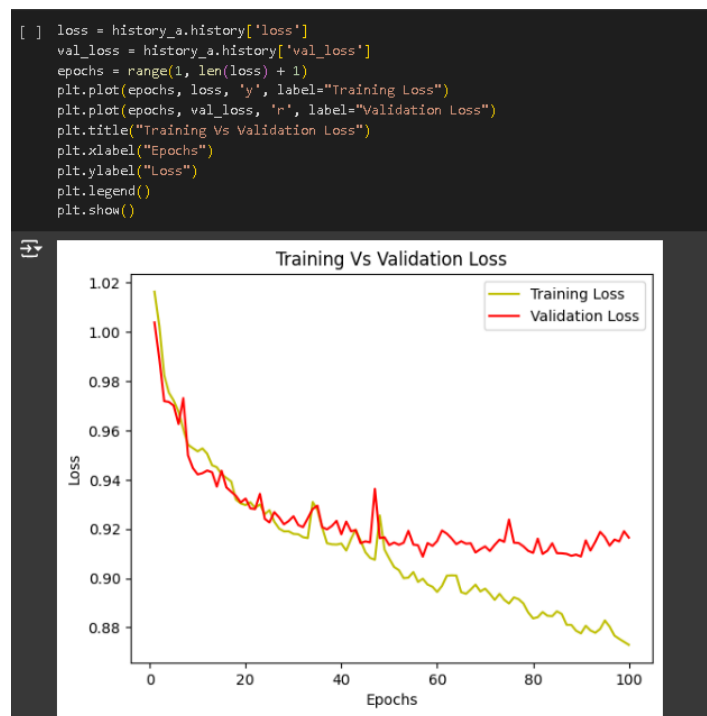


Figure 6.13 Training & Validation Loss Curve Implementation

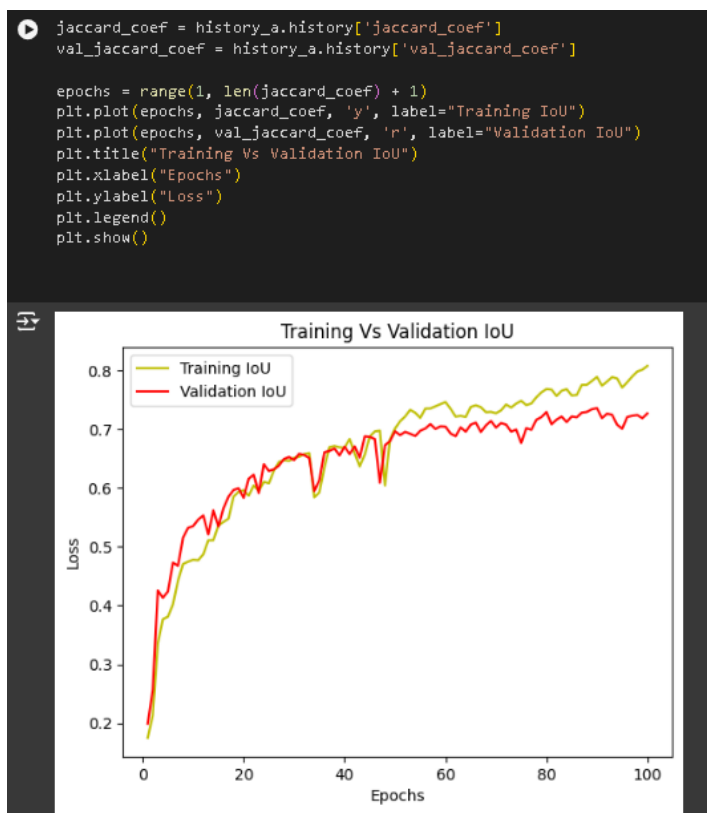


Figure 6.14 Training & Validation IoU Curve Implementation

## **CHAPTER 7**

### **REFERENCES**

- [1] Xu Y, Liu X, Cao X, et al. Artificial intelligence: A powerful paradigm for scientific research. *Innovation (Camb)*. 2021;2(4):100179. Published 2021 Oct 28. doi:10.1016/j.xinn.2021.100179
- [2] Çelik, Özer. (2018). A Research on Machine Learning Methods and Its Applications. 10.31681/jetol.457046.
- [3] Mishra, Chandrahas & Gupta, D.. (2017). Deep Machine Learning and Neural Networks: An Overview. *IAES International Journal of Artificial Intelligence (IJ-AI)*. 6. 66. 10.11591/ijai.v6.i2.pp66-73.
- [4] Sonali M, Maind, Priyanka Wankar M. *International Journal on Recent and Innovation Trends in Computing and Communication Research Paper on Basic of Artificial Neural Network.*; 2014.  
<https://ijritcc.org/index.php/ijritcc/article/download/2920/2920/2895>
- [5] Guo Y, Liu Y, Georgiou T, Lew MS. A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*. 2017;7(2):87-93. doi:<https://doi.org/10.1007/s13735-017-0141-z>
- [6] N. Siddique, S. Paheding, C. P. Elkin and V. Devabhaktuni, "U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications," in *IEEE Access*, vol. 9, pp. 82031-82057, 2021, doi: 10.1109/ACCESS.2021.3086020.
- [7] Ronneberger, O., Fischer, P., Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. MICCAI 2015. Lecture Notes in Computer Science(), vol 9351. Springer, Cham. [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
- [8] Taye MM. Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions. *Computation*. 2023; 11(3):52. <https://doi.org/10.3390/computation11030052>
- [9] Oğuz, Abdulhalık & Ertugrul, Omer. (2023). Introduction to deep learning and diagnosis in medicine. 10.1016/B978-0-323-96129-5.00003-2.

- [10] Jadeja M. Jaccard Similarity Made Simple: A Beginner's Guide to Data Comparison. Medium. Published July 10, 2023.  
<https://medium.com/@mayurdhvajsinhjadeja/jaccard-similarity-34e2c15fb524>
  
- [11] N. Subraja and D. Venkatasekhar, "Satellite Image Segmentation using Modified U-Net Convolutional Networks," 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2022, pp. 1706-1713, doi: 10.1109/ICSCDS53736.2022.9760787.
  
- [12] Paranjape C. Satellite imagery segmentation using U-NET. Medium. Published July 17, 2023. Accessed May 18, 2024.  
<https://pub.aimind.so/satellite-imagery-segmentation-using-u-net-4ec7f265ddbe>
  
- [13] Ali L, Alnajjar F, Jassmi HA, Gocho M, Khan W, Serhani MA. Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures. *Sensors*. 2021;21(5):1688. doi:<https://doi.org/10.3390/s21051688>
  
- [14] Rivera, Antonio & Rolando, Julio & Cabezas, Coello & Ronald, Emer & Solórzano, Rosales & Andrés, Carlos & Herrera, Mancheno & Guillermo, Gregory & Andrade, Cuesta & Elena, Carmen & Cabrera, Mantilla. (2023). Satellite Image Using Image Processing and Machine Learning Techniques: applications to agriculture, environment and mining.. *European Chemical Bulletin*. 1718-1734. 10.31838/ecb/2023.12.1.256.
  
- [15] Marmanis, Dimitris & Wegner, Jan & Galliani, Silvano & Schindler, Konrad & Datcu, Mihai & Stilla, Uwe. (2016). Semantic Segmentation of Aerial Images with an Ensemble of CNSS.
  
- [16] Trestioreanu, Lucian. (2018). Holographic Visualisation of Radiology Data and Automated Machine Learning-based Medical Image Segmentation.
  
- [17] Cortés-Ferre, Luis & Gutiérrez-Naranjo, Miguel & Egea-Guerrero, Juan & Pérez-Sánchez, Soledad & Balcerzyk, Marcin. (2023). Deep Learning Applied to Intracranial Hemorrhage Detection. *Journal of Imaging*. 9. 37. 10.3390/jimaging9020037.