Thesis

by Muhammad Haziq Faiz Mohd Ripin

Submission date: 10-Feb-2023 04:53PM (UTC+0800)
Submission ID: 2010788029
File name: thesis.pdf (11.51M)
Word count: 19389
Character count: 101620

CHAPTER 1

ACKNOWLEDGEMENTS

ABSTRACT

Semantic segmentation of satellite images is an important task as it has a wide range of practical applications, such as land cover mapping, poolution monitoring, deforestation detection and economic assessment. With the rise of deep learning, semantic segmentation models have been built using the convolutional neural network for many years. CNNs have proven to be really good at capturing local features but limited when capturing the global features. Recently, Transformers, a network that was originally designed for processing texts were adopted by the field of computer vision and has shown a lot of potential to computer vision task such as semantic segmentation. Unlike CNN, Transformer has the ability to capture both local and global features. For FYP 1, we utilized UNet-like Transformer (UNetFormer) that is made up of a CNN-based encoder and Transformer-based decoder for semantic segmentation of satellite images to get an initial result. We also trained a U-Net, a purely CNN-based encoder-decoder network using the same dataset for comparison purpose. The dataset chosen for this project is the LoveDA dataset which contains 5987 High Spatial Resolution

(HSR)images with 166768 annotated pixels from three different cities in China.

The U-Net model gave an mIoU of 33.6% and the UNetFormer gave an mIoU gave an mIoU of 72.5%. U-Net has only 2.4 million parameters compared to 11.2 million on the UNetFormer.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 2

INTRODUCTION

2.1 *Introduction to Semantic Segmentation*

The last few years have seen a massive surge in research regarding deep learning applications in computer vision with the most common one being object detection, where a network accept an image as an input and output either a single or multi class label. Typically, the position of detected objects are defined by rectangular coordinates that are represented by bounding boxes. However, in a lot of image processing task, such as in satellite images analysis the target output should include more accurate localization. The bounding box may have more than one objects inside it. To increase its localization, instead of assigning a set of labels to an image, semantic segmentation would label each pixel independently. After each pixel is labelled, a new image, called the mask will be produced with every pixels being coloured according to its label.

The emergence of the term "semantic segmentation" can be traced back to the 1970s (?, ?). At that time, this terminology was equivalent to non-semantic image segmentation but emphasized that the segmented regions must contain a "semantic meaning" hence where the semantic part of the word comes from. Semantic segmentation algo-

rithms calssify each pixel into a class. On the other hand, non-semantic segmentation algorithms try to detect consistent regions or region boundaries and it can be solved using many unsupervised algorithms.

In the 1990s, "object segmentation and recognition" is a two-class image segmentation problem which is to separate foreground objects from background. As the task is quite challenging, a relaxed two-class image segmentation problem: the sliding window object detection, was proposed to partition objects with bounding boxes. However, two-class image segmentation cannot tell what these segmented objects are. As a result, the generic sense of object detection was gradually extended to multi-class image labeling, which is the present definition of semantic segmentation, to tell both where and what the objects in the scene.

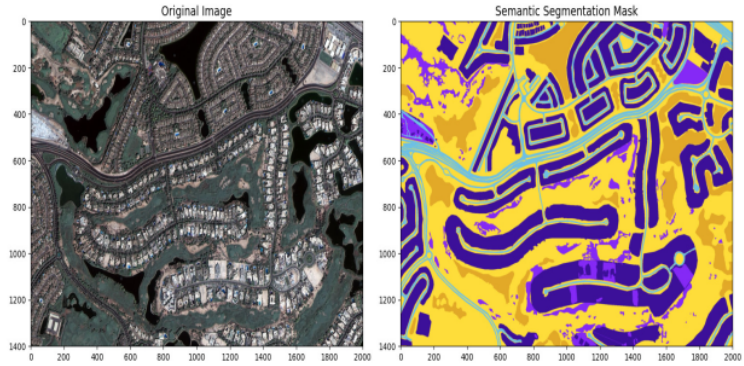

Figure 2.1: Semantic Segmentation of a Satellite Image

Figure 2.2: Object Detection of a Satellite Image

2.2 *Introduction to Satellite Images*

Satellite images are images of the Earth that are collected by observation satellites. Observation satellites are satellites that are designed to observe the Earth from orbit while equipped with sensors that measure a range of electromagnetic spectrum such as UV, visible, infrared, microwave, or radio.

There are 3 types of resolution that one should consider when working with satellite images. Namely the resolutions are the spatial resolution, spectral resolution and temporal resolution.

Spatial resolution refers to the smallest feature that is displayed by an image. In most datasets it is usually represented as a single numerical value representing one side of a square pixel. For example, a spatial resolution of 10m means that a single pixel represents an area of 10 m^2.

Spectral resolution refers to the extent of the sensors on the satellite to detect and measure wavelengths on the electromagnetic spectrum. The finer the spectral resolution, the narrower the wavelength range for a particular channel or band. Images with high spectral resolution is important in computer vision because classes such as rock types and soil types would require an analysis at a much finer spectrum to distinguish them.

Spectral Bands	**Wavelength μm**	**Description**
Band 1	400 - 450	Least absorbed by water, and will be very useful in bathymetric studies.
Band 2	450 -510	Provides good penetration of water.
Band 3	510 - 580	Ideal for calculating plant vigor.
Band 4	585 - 625	Detects the "yellowness" of particular vegetation.
Band 5	630 - 690	Better focused on the absorption of red light.
Band 6	705 - 745	Centered strategically at the onset of the high reflectivity portion of vegetation response
Band 7	770 - 895	Effectively separates water bodies from vegetation, identifies types of vegetation and also discriminates between soil types
Band 8	860 - 1040	Overlaps Band 7 but is less affected by atmospheric influence.

Table 2.1: Different purposes of spectral bands of satellite images

Lastly, temporal resolution refers to the time period between capturing two consecutive images of the same surface area. In some literature it is also called the satellite revisit period. An image with a higher temporal resolution has a lower time period between two consecutive images. For example an image with a temporal resolution of two days means that a satellite will capture an image of the same area every two days. For some satellites, a constellation of satellites are used to increase temporal resolution. As an

example the SENTINEL-2 mission actually use 2 satellites with each having a revisit period of 10 days making the temporal resolution to be effectively 5 days. Temporal resolutions are important to detect changes that occur during a specified time period.

2.3 Applications of Semantic Segmentation of Satellite Images

1. **Land Cover Mapping**

   Land cover mapping is the process of constructing a cover map that provide information about the Earth's surface cover pattern and land use. Such example of information provided are vegetation index and soil index. Covers maps are important for agricultural monitoring, public policy development and urban planning. Land cover mapping utilizes semantic segmentation of satellite images. Before the advances of deep learning, land cover mapping relies on traditional semantic segmentation techniques such Support Vector Machines and Random Decision Forest (?, ?). However, semantic segmentation requires a huge number of features to distinguish huge variations of land patterns. Traditional methods that only rely on low-level spectral and spatial resolution have been proven to be less optimal than its deep learning alternative due to the latter's ability to extract multilevel and multi-scale features(?, ?).

2. **Water Bodies Detection**

   Semantic segmentation of satellite images has long been used in detecting bod-

ies of water such as lakes and natural springs in areas where water is a scarce resource. One of the most popular traditional technique is normalized difference water index (NDWI). This technique heavily relies on IR band and measure the reflectance characteristics of water. This technique is very susceptive to noise and quite complex to develop and deploy. However, deep learning has been proven to be more reliable than NDWI, a CNN based network reached an accuracy of 99.86% (?, ?).

3. **Soil Erosion Detection**

   Understanding soil attributes is an important step for the construction industry. As most construction projects require excavation ((e.g.,piping, laying foundation, tunneling), soil attributes could affect the entire excavation process concerning scheduling, resource planning, procurement, claim resolution, and safety considerations. Soil classification is a process of categorizing soil based on similar attributes. The traditional involves on-site sampling and analysing the samples in a laboratory which is very time consuming and expensive. Thanks to deep learning ability to utilizes high resolution images, a CNN based network was developed to classify soil based on semantic segmentation of satellite images (?, ?).

4. **Flood Detection and Assessment System**

   Due to climate change, flooding has quickly becoming one of the most destructive and frequent type of natural disaster, and this trend is expected to continue.

Detecting flood prior of its occurrence has been vital to save lives and minimizes financial loss. On top of that, a lot of agencies around the world require a system to assess the total destruction from flooding. The assessment method is usually done manually using aerial images. Semantic segmentation has been widely used as a tool to aid in the process of designing and deploying accurate flood detection and post-flood assessment system (?, ?).

2.4 Problem Statement

1. Most of the models reviewed are either not trained using satellite images or the dataset have very low resolution and dimensions.
2. Most of the semantic segmentation models reviewed are proposed before 2022 and they use CNN as Vision Transformer is a new development.
3. Vision Transformer models are too large hence making it unsuitable for real-world applications.

2.5 Project Objectives

1. Identify suitable datasets of satellite images that will be used for training and validation. Multiple datasets will be evaluated and a new dataset would be constructed if necessary. The dataset must contain a minimal class imbalanced.

2. Identify and train a semantic segmentation model with a Transformer-based encoder-decoder model to perform using chosen dataset to get a baseline result. We would also train a CNN model using the same dataset to get a comparison.

3. Identify the appropriate steps that can be taken to compress the Transformer model while maintaining its performance.

2.6 *Project Scope*

The scope of the dataset used in this project will be limited to satellite images. The chosen dataset must have a spatial resolution that is small enough to avoid any losses of information. On top of that, the dataset must be bigger than 240x240 pixels. The dataset must have been taken by a satellite. There is no restriction on the type of bands that the dataset can have.

The scope of the network proposed in this project must be one based on Vision Transformer. The Vision Transformer network must be able to perform semantic segmentation task of satellite images. The performance of the proposed vision transformer network shall be compared to the previous works trained on the same dataset.

2.7 *Chapter Organization*

This report is organized into 6 chapters. Chapter 1 would provide a brief introduction to satellite images and semantic segmentation, the applications of semantic segmentation of satellite images and the problems that this project aims to solve. Chapter 2

review the current literature that aims have similiar objectives to this project. It also contain a brief introduction to the semantic segmentation techniques before the advance of deep learning. Chapter 3 explains the fundamental theories to be understood before designing and implementing the model. Chapter 4 presents the details of the project pipelines with clear distinctions of work to be done. Chapter 5 shows the implementation plan for both FYP1 and FYP2 including milestones as well as preliminary results of experiments. Lastly, Chapter 6 provides a summary of work done in FYP1 with findings, future directions, and problems encountered.

CHAPTER 3

LITERATURE REVIEW

This chapter would cover the literature review part of this project. The first section would elaborate on the datasets evaluated, including the data exploratory analysis of the datasets. The second and third sections would include a brief introduction to traditional methods semantic segmentation and their limitations respectively. The fourth sections would serve as a literature review of semantic segmentation using Convolutional Neural Networks. The fifth and last section would include the literature review of semantic segmentation using transformers and its advantages.

3.1 Satellite Images Datasets

Dataset	Source	# Samples	# Classes	Size (px)	Res (m)	Band
Benin Cashew Plantation	Airbus Pléiades	70	6	1,122x1,186	10	MSI
Cloud Cover Detection	Sentinel-2	22,728	2	512x512	10	MSI
Kenya Crop Trade	Sentinel-2	4,688	7	3,035x2,016	10	MSI
Deep Globe Land Cover	DigitalGlobe +Vivid	803	7	2,448x2,448	0.5	RGB
DFC2022	Aerial	3,981	15	2,000x2,000	0.5	RGB
ETCI 2021 Flood Prediction	Sentinel-1	66,810	2	256x256	5–20	SAR
GID-15	Gaofen-2	150	15	6,800x7,200	3	RGB
LandCover.ai	Aerial	10,674	5	512x512	0.25–0.5	RGB
LoveDA	Google Earth	5,987	7	1,024x1,024	0.3	RGB
Potsdam	Aerial	38	6	4,000x4,000	0.02	RGB
Vaihingen	Aerial	33	6	1,281–3,816	0.09	RGB
SEN12MS	Sentinel-1/2, MODIS	180,662	33	256x256	10	SAR, MSI

Table 3.1: Datasets Reviewed for Semantic Segmentation Task

Table **??** shows the list of dataset that were evaluated and considered for this project. Each of the dataset is made for semantic segmentation task meaning that each pixel in the dataset is labelled. All of the datasets except for Potsdam and Vaihingen can be downloaded using the TorchGeo library.

The Potsdam and Vaihingen datasets (?, ?) are datasets made specifically for urban semantic segmentation used in the 2D Semantic Labeling Contest - Potsdam and 2D Semantic Labeling Contest - Vaihingen respectively. Both of the datasets are available upon request here. Although the images are not taken by satellites, a lot of literature reviewed such as (?, ?), (?, ?), (?, ?) and (?, ?) train their semantic segmentation model using these datasets thus we think these merit a bit of discussions. Vaihingen dataset is composed of 33 orthorectified image tiles acquired by a near NIR-RGB drone camera, over the town of Vaihingen, Germany. The average size of the images is 20494 x 20064 pixels with a spatial resolution of 9 cm. Potsdam dataset is composed of 38 orthorectified image tiles acquired over the town of Potsdam using the same NIR-RGB drone camera.. The average size of the tiles is also 20494 x 20064 pixels but with a smaller spatial resolution of 5 cm. Images in both datasets are accompanied by a digital surface model (DSM) representing the absolute height of pixels. There are a total of 6 classes in each dataset: impervious surfaces (roads, concrete surfaces), buildings, low vegetation, trees, cars and clutters (represents uncategorizable land covers). Figure **??** and **??** shows a pair of an image and its corresponding mask from the Potsdam dataset while Figure **??** and **??** show the ones from Vaihingen dataset.

The LoveDA dataset (?, ?) contains 5987 High Spatial Resolution (HSR) images with 166768 annotated pixels from three different cities in China. The images are obtained from Google Earth. The images are divided into two sub-categories: urban and rural. There are nine urban areas selected from different economically developed districts, which are all densely populated. The other nine rural areas were selected from undeveloped districts. The spatial resolution is 0.3 m, with red, green, and blue bands. After geometric registration and pre-processing, each area is covered by 1024 × 1024 images, without overlap. LoveDA dataset contains a total of 7 classes: building, road, water, agriculture, barren, forest and background. Figure **??** and **??** shows a pair of an image and its corresponding mask from LoveDA dataset.

Figure 3.1: An Image From LoveDa Dataset (?, ?)

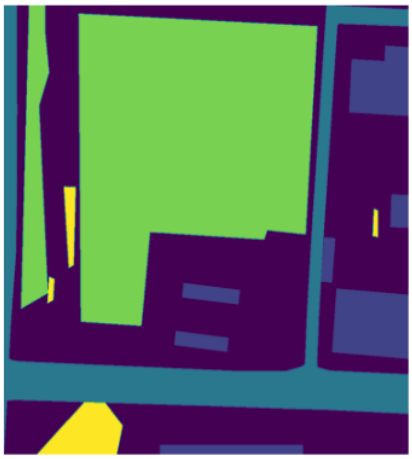

Figure 3.2: A Mask From LoveDa Dataset (?, ?)

The GID-5 dataset (?, ?) contains 120 6800 x 7200 HSR images with 5 classes. The classes are built-up, farmland, meadow, farmland and water which are pixel-level la-

beled with five different colors: red, green, cyan, yellow, and blue, respectively. The images are obtained from Gaofen-2 satellite. GID-5 dataset is widely distributed over the geographic areas covering more than 50,000 km^2. Due to the extensive geographical distribution, GID-5 represents the distribution information of ground objects in different areas. Figure **??** and **??** shows an example of an image and its corresponding mask from Deep Globe dataset.

The Deep Globe Land Cover dataset (?, ?) contains 1146 satellite images of size 2448× 2448 pixels. The dataset is split into training, validation and test sets, each with 803/171/172 images. All images contain RGB channel, with a pixel resolution of 50 cm. The dataset is collected from the DigitalGlobe Vivid+ dataset which is an earlier dataset containing satellite images. The annotations are pixel-wise segmentation masks created by professional annotators. There are 7 total classes:

1. Urban land: Man-made, built up areas with human artifacts.
2. Agriculture land: Farms, plantation, cropland, orchards, vineyards, nurseries, and ornamental horticultural areas; confined feeding infrastructure.
3. Rangeland: Any non-forest, non-farm, green land, grass.
4. Forest land: Any land with at least 20% tree crown density plus clear cuts.
5. Water: Rivers, oceans, lakes, wetland, ponds.

6. Barren land: Mountain, rock, dessert, beach, land with no vegetation.

7. Unknown: Clouds and others.

Figure **??** and **??** shows an example of an image and its corresponding mask from Deep Globe dataset.

The LandCover.ai dataset (?, ?) is a simple RGB-only dataset with spatial resolution of 25 or 50 cm per pixel. There are 33 images with resolution 25 cm(9000 × 9500 pixels) and 8 images with resolution 50 cm (4200 × 4700 pixels). Pixel-wise annotations are made manually with VGG Image Annotator (VIA) by a group of people using polygon shape and polylines. LandCover.ai dataset has 5 classes: buildings, woodlands, water, roads and background. Figure **??** and **??** show a pair of its image and mask.

3.2 Semantic Segmentation Before Deep Learning

This section would elaborate on traditional methods of semantic segmentation, methods that do not apply any neural networks for semantic segmentation. Before deep learning, semantic segmentation relied heavily on feature extraction and traditional machine learning (ML) algorithm as shown in Figure **??**. These two methods are rendered obsolete by deep learning.

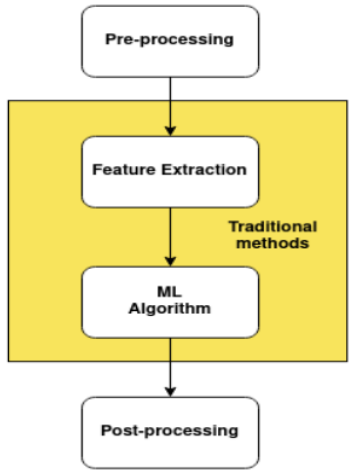

Figure 3.3: Semantic Segmentation Before Deep Learning.

3.2.1 Feature Extraction

Before we apply any of the traditional ML algorithm that will be discussed in the proceeding sections we must extract the features from an image. The accuracy of traditional semantic segmentation methods heavily depends on the selected features. The features may be the numerical value of each pixel or the feature map containing the gradient of each pixel. There are three feature extraction methods discussed in this section, namely they are Histogram of Oriented Gradients, Scale-Invariant Feature Transform and Bag of Visual Words .

1. ***Histogram of Oriented Gradients (HOG).*** HOG features interpret any given image as a discrete function $I : \mathbb{N}^2 \rightarrow \{0...255\}$ that maps a pair value (x, y) which is the coordinate of the pixel to an RGB value. Then, the partial derivative in of

x and y are calculated for every pixel. The input image is now transformed into a feature map with the gradient of each pixel. Lastly, the constructed feature map is divided into smaller patches and the direction and magnitude of the histogram is calculated for each patch. (?, ?).

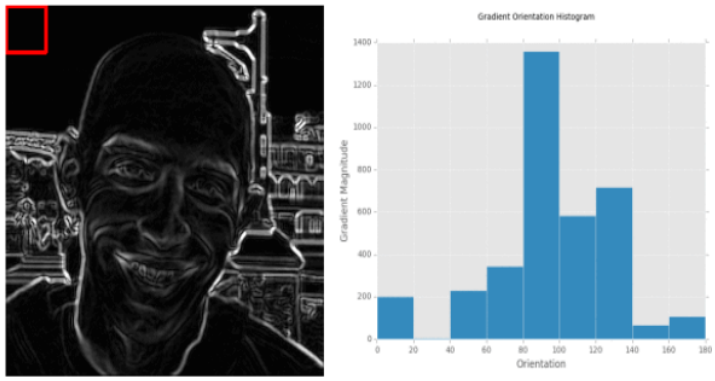

Figure 3.4: Computing a histogram of oriented gradients for the first patch of an input image.

2. ***Scale-Invariant Feature Transform (SIFT).*** SIFT is a feature extraction algorithm that was introduced in 2004. Unlike HOG, SIFT is not affected by the orientation or scale of the input image. (?, ?). An image will be divided into smaller patches and the difference-of-Gaussian (DoG)is calculated. DoG is obtained as the difference of Gaussian blurring of an image with two different σ. Next, local extrema is searched to be be assigned as potential key points. Lastly, after the key points are discovered, an 8-bin orientation histogram is created for each patch to math the key points. The final output will be a feature map containing accepted key points. A more thorough explanation is available in the

original paper (?, ?).

Figure 3.5: Each circle represents the location and orientation of SIFT keypoints

3. ***Bag of Visual Words (BOV).*** BOV construct sparse histograms that contain the frequency of features in an image. Those features are usually extracted using SIFT (?, ?). BOV is often used alongside other feature extractors such as SIFT by assigning each SIFT descriptor to the closest entry in a visual dictionary.

3.2.2 Traditional Machine Learning Algorithm

3.2.2 (a) Random Decision Forest for Semantic Segmentation

A decision tree is a tree where each leaf represents a class and each non-leaf nodes uses the feature inputs to decide which branch to descend to (?, ?). A random decision tree is as decision tree that is injected with some randomness during the training phase to reduce over-fitting and increase accuracy. Random Decision Forest is an unsupervised

ensemble learning method that are made up of multiple independently constructed random decision trees. An in-depth explanation to semantic segmentation using Random Decision Forest is given by (?, ?).

3.2.2 (b) Support Vector Machines (SVM) for Semantic Segmentation

In SVM the training data is represented as (x_i, y_i) where x_i is the feature vector, $y_i \in \{-1, 1\}$ is the class label and $i \in \{1...m\}$ where m is the number of inputs.

Assuming that the data is linearly separable, SVM is a task of solving the optimal margin classifier:

$$min_{w,b} \quad \frac{1}{2}||w||^2 \tag{3.1}$$

$$s.t. \quad y^i(w^T x^i + b) \geqslant 1, i \in 1...m \tag{3.2}$$

Where w is the weight, x the input vector, y is the output vector and b is the bias. $w$can be defined as:

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{3.3}$$

Where α is the Lagrange multiplier.

Not every dataset is linearly separable thus this problem can be solved by transforming the feature vectors x into a higher dimension using a non-linear mapping $/psi$. Thus instead of learning using x, we may learn using a higher-dimensional features generated by $\psi(x)$, this method is called the *kernel trick*. Specifically, given a feature mapping ψ, we define the corresponding kernel to be:

$$K(x,z) = \psi(x)^T \psi(z) \tag{3.4}$$

The SVM described above can only distinguish between binary classes. The one-vs-all strategy and the one-vs-one strategy are methods used to expand it to be a multi-class classifier.. In the one-vs-all strategy n classifiers have to be trained which can distinguish one of the n classes against all other classes. In the one-vs-one strategy $\frac{n^2-n}{2}$ classifiers are trained; one classifier for each pair of classes.

3.2.2 (c) *Markov Random Field (MRF)*

MRF maps an image onto an undirected graph where each node is a pair of random variable (x, y) assigned to each pixel and the edges connect adjacent pixels (?, ?). x represents the class label of a pixel and y represents the RGB value of a pixel. Which means x has a range of $0...n$ and y has a range of $0...255$ with n being the number of classes. Every edge is assigned conditional dependencies of its connecting nodes as weight. The probability of x, y can be expressed as:

$$P(x,y) = \frac{1}{Z} e^{-E(x,y)} \tag{3.5}$$

Where $Z = \sum_{x,y} e^{-E(x,y)}$ and it is called the partition function and E is the energy function. A commonly used energy function is $E(x,y) = \sum_{c \in C} \psi_c(x,y)$, where ψ is called the clique potential (?, ?). A thorough presentation of MRF can be found in (?, ?).

3.2.2 (d) *Conditional Random Field (CRF) for Semantic Segmentation*

CRF is an extension of MRF. Instead of learning the distribution $P(x,y)$, it chooses to learn $P(x|y)$ (?, ?). There are two advantages that CRF has over MRF. The first one being it does not to estimate the distribution of x. The second advantage is the consequence of the first one, as the distribution of x is not being estimated, less computation

is required hence making CRF faster than MRF (?, ?). CRF has the partition function $Z(x)$:

$$Z(x) = \sum_{x} P(x, y) \tag{3.6}$$

and the joint probability distribution is given as follows:

$$P(y|x) = \frac{1}{Z(x)} \prod_{c \in C} \psi_c(y_c|x) \tag{3.7}$$

CRF is often used in conjunction with neural networks as a post-processing method for semantic segmentation task. It is is used to smoothen the output mask (?, ?).

3.3 Limitations of Traditional Methods

1. The traditional method is simply less accurate. As an example the best traditional method back in 2015 which utilized SIFT features extraction method and Fisher Vectors, had a performance of about 25.7% error rate on semantic segmentation task on ILSVRC-2010 dataset. While AlexNet proposed by (?, ?) had an error rate of 17.0% (?, ?).

2. Feature extraction method such as SIFT and Random Decision Forest require researchers to come up with a good hand-crafted feature to achieve high accuracy

while good features are very hard to produce. Compared this with the automatically learned features provided by deep learning, traditional methods would require a lot more time and effort.

3.4 *Semantic Segmentation of Satellite Images Using Convolutional Neural Networks*

Semantic segmentation task has been long dominated by Convolutional Neural network (CNN). The Fully Convolutional Network (FCN) (?, ?) is the first network proven to be an effective method to extract features automatically and serves as an effective end-to-end CNN structure for semantic segmentation task. The outcome of FCN, although encouraging, appears to be coarse due to the over-simplified design of the decoder.

To tackle this problem, better CNNs were proposed such as U-Net (?, ?) which introduced the encoder-decoder framework. Since its introduction, the encoder-decoder framework has become the standard structure of satellite images segmentation network (?, ?). U-Net introduced two symmetric paths: a contracting path, which is also known as the encoder to extract local features, and an expanding path which is also called the decoder, for extracting position. The encoder gradually apply convolutions and max pooling to reduce the resolution of the feature map, while the decoder extracts contextual information by progressively restoring the spatial resolution. At every level of the decoder skip connections are used to concatenate the output of the decoder with

the feature maps from the encoder. Figure **??** show the U-Net architecture. Benefiting from its translation equivariance and locality, U-Net enhances the semantic segmentation performance significantly and the encode-decoder framework has been a major influence towards semantic segmentation task.

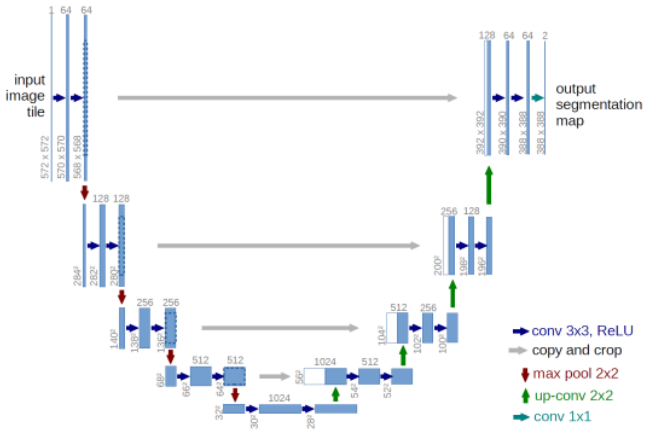

Figure 3.6: U-Net Architecture (?, ?)

Even though the results are promising, the long-range dependency of U-Net is limited by the locality property of the convolutional mechanism, which is critical for semantic segmentation. There are two types of approaches to address this issue, either modifying the convolution operation or utilizing the attention mechanism. Such examples of the first approach is to enlarge the receptive fields using large kernel sizes (?, ?), or utilising feature pyramids (?, ?). On the other hand the second approach focuses on integrating attention mechanisms with the encoder-decoder architecture to capture long-range dependencies of the feature maps, examples can be found in Attentive Bilateral Contextual Network (?, ?) and Attention U-Net (?, ?). Although the second

approach showed better performance, both approaches failed to decouple the network from the dependence of the encoder-decoder structure. In the context of semantic segmentation, per-pixel classification is often inaccurate if only local information is taken into account (?, ?).

Attention U-Net (?, ?) is a model that aims to improve U-Net by introducing attention gates in the skip connections between the encoder and decoder as shown in **??**. The results showed that Attention U-Net performs better than traditional U-Net for segmentation of CT scans images.

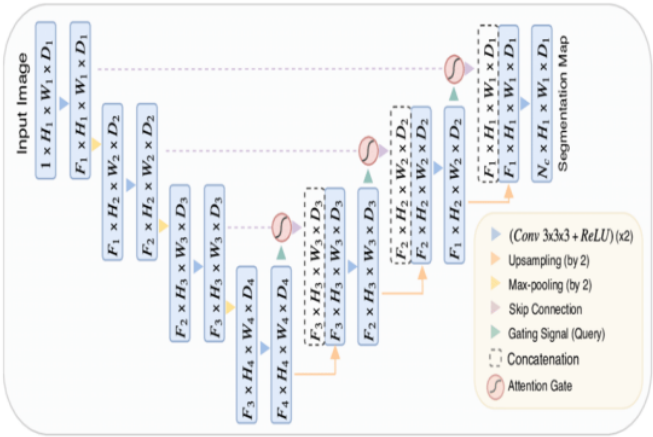

Figure 3.7: Attention U-Net Architecture (?, ?)

U-Net with added attention mechanism is the approach favored by most literature focusing on semantic segmentation of satellite images. Multi Attention U-Net(MA-Unet) (?, ?) uses residual structure and simple attention modules. According to figure **??** the decoder in MA-Unet uses transposed convolution and attention modules

at every feature fusion stage. MAUnet performed semantic segmentation task on the WHDLD datasets and DLRSD datasets. WDLD dataset have 8 classes while DLRSD dataset have 17 classes covering common classes such as farms, grasslands and lakes. MA-Unet was compared with other U-Net based models such as U-Net, U-Net++ and Attention U-Net on the same task. MA-Unet performs better at segmenting classes with fine details such aeroplanes. MA-Unet has the highest mIoU on both dataset with 63.94% on WHDLD and 61.90% on DLRSD.

Spatial Attention U-Net (SA-Unet) (?, ?) uses atrous spatial pyramid pooling as its encoder and attention modules at every skip connection like we have seen in Attention U-Net. (?, ?) employs U-Net with attention to extract roads from satellite images. Instead of using simple attention modules at every skip connections, they uses attention module only once at the lowest skip connection.

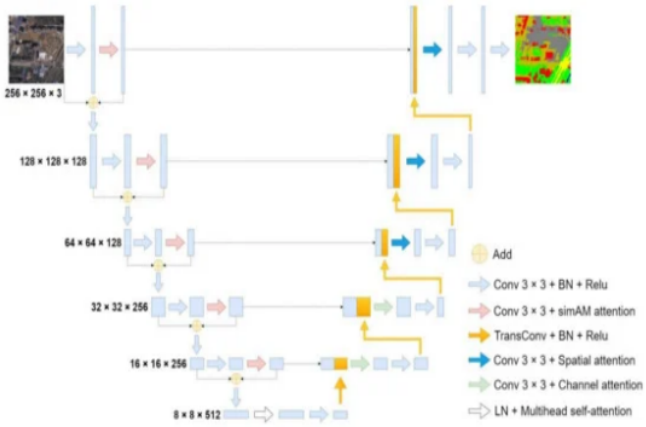

Figure 3.8: Multi Attention U-Net Architecture (?, ?)

3.5 *Semantic Segmentation of Satellite Images Using Vision Transformers*

Transformers were introduced by (?, ?) and since its inception it has been the de-facto model for Natural Language Processing (NLP). Transformers that are used for computer vision are called Vision Transformers to differentiate it from its NLP counterpart. Vision Transformers essentially translates 2D image-based tasks into 1D sequence-based tasks.

Since its inception, there are lot of researches that aim to utilize Transformers for semantic segmentation of satellite images. (?, ?) did a comparison of ViT, ResNet, MLPMixer and VGG trained using the BigEarthNet dataset for semantic segmentation and they concluded that ViT with a patch size of 6 delivered a slightly better performance than the rest while consuming less time to train. Interestingly, ViT with smaller patch size has lower F scores while requiring more time to train.

3.5.1 *Vision Transformer (ViT)*

Vision Transformer from (?, ?) is the first group of researchers that experimented with Vision Transformer by applying a standard Transformer directly to images, with the fewest possible modifications. Their model is known as Vision Transformer (ViT) as it is the first Vision Transformer. They split an image into patches, apply linear embeddings to transform the patches into vectors and provide the sequence of those

linear embedding as an input to a Transformer. The image patches were treated the same way as word tokens do in an NLP application. This methods fails to capture the local context provided CNNs hence it is unsuitable for semantic segmentation task.

Another issue with Vision Transformer is the attention mechanism itself is $O(n^2)$ because it is a dot product thus requiring it to consume significant computational time and memory to capture the global context, which in turn, reducing its efficiency, scaling potential and its potential for real-world applications. Figure **??** shows the ViT architecture.

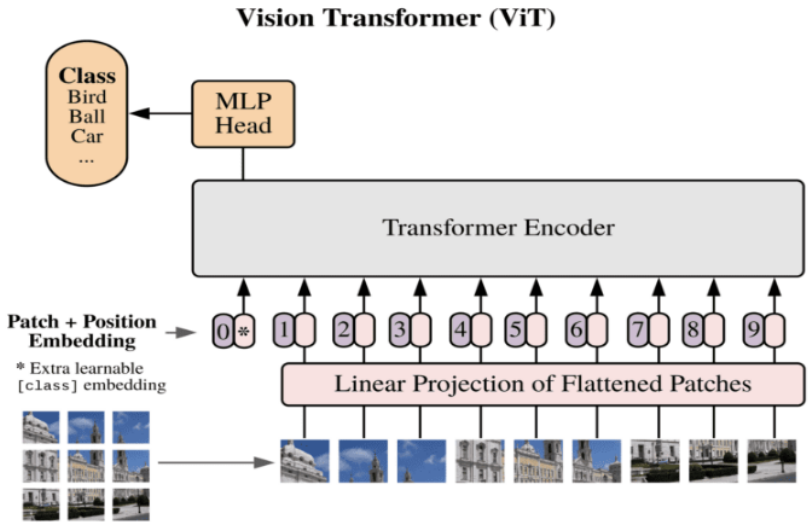

Figure 3.9: ViT Architecture (?, ?)

3.5.2 Swin Transformer

The first version of Swin Transformer (?, ?) presents a hierarchical feature representation scheme that demonstrates impressive performances with linear computational complexity, which makes it suitable for semantic segmentation. Just like in ViT, Swin Transformer will use patch embedding to the input image but the difference now is the patch size is 4x4x3 instead of 16x16x3. The biggest difference between ViT and Swin Transformer is ViT attention layers work on the entire image while Swin Transformer will first compute attention on local windows. Take an example in figure **??**, the input image is 64x64 and Swin Transformer will first use 16 non-overlapping local windows to divide the image into 16 patches. As self-attention is computed at local window, they only handle the relationships between 16 image patches. At the next level, Swin Transformer merges neighboring patches to form 4 new local windows. Again, self-attention layers is computed within each local window. Therefore, those layers only deal with 16 feature patches hence maintaining the number of patches at every level. Eventually, one local window covers the entire image with 16 feature patches.

Swin Transformer is made up of 4 stages as shown in **??**. The input image has a dimension of H x W x 3. Swin Transformer uses a patch size of 4x4 which will produce H/4 x W/4 number of patches, and each patch's feature dimension is 48 (4 x 4 x 3). In stage 1, the linear layer projects the raw-valued features to an arbitrary dimension C.

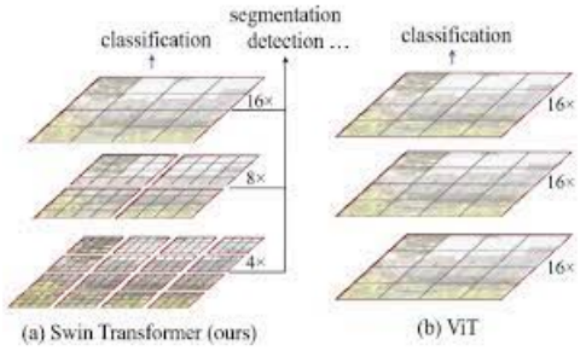

Figure 3.10: The difference between Swin Transformer and ViT (?, ?)

In stage 2, the patch merging layer concatenates the features of each group of 2 x 2 neighboring patches. Each patch's feature dimension becomes 4C, on which it applies a linear layer, reducing the output dimension to 2C. So, we have H/8 x W/8 patches, each of which has 2C-dimensional features. The same processes are repeated in stages 3 and 4, reducing the size by 4 and increasing the feature dimension by 2.

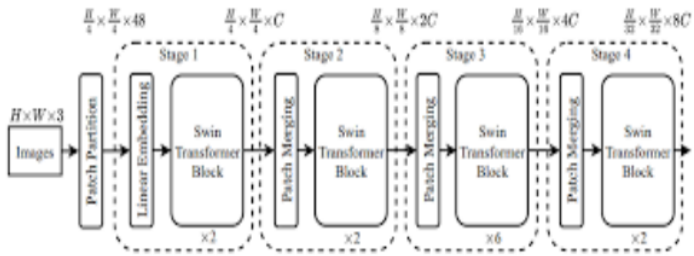

Figure 3.11: Swin Transformer V1 Architecture (?, ?)

The self-attention module in ViT has $O(n^2)$computational complexity to the number of tokens which makes it unsuitable for semantic segmentation which require dense high-resolution classifications. Swin Transformer introduced a Shifted Window Multi-

headed Self-Attention (W-MSA) to make the self-attention linear to the number of tokens. The computational complexity of the original Multi-Headed Self-Attention is:

$$\Omega(\mathbf{MSA}) = 4hwC^2 + 2(hw)^2C \tag{3.8}$$

Where h is the height of the patch, w is the width of the patch and C is the feature size. While the improved computational complexity of W-MSA is linear to the number of patches as shown in the equation below:

$$\Omega(\mathbf{W-MSA}) = 4hwC^2 + 2M^2hwC \tag{3.9}$$

Swin Transformer alternates between two partitioning configurations in consecutive Swin Transformer blocks. As shown figure **??**, layer 1 uses regular partitioning while layer 2 uses a windowing configuration shifted by half of the window size. They introduces connections between neighboring windows while keeping the local computation within each non-overlapping window.

The main issue with this approach is the number of local windows increased from 4 to 9. For this reason, Swin Transformer uses another mechanism called cyclic shifting. Cyclic shifting shift the image patches toward the top-left direction and then applies a masking mechanism to limit self-attention within adjacent patches. It also utilizes

reverse cyclic shifting to cover the other areas. Cyclic shifting is shown in figure **??**

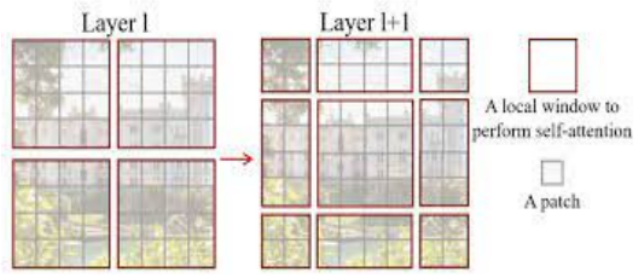

Figure 3.12: Shifted Window Multi-head Self-Attention (?, ?)

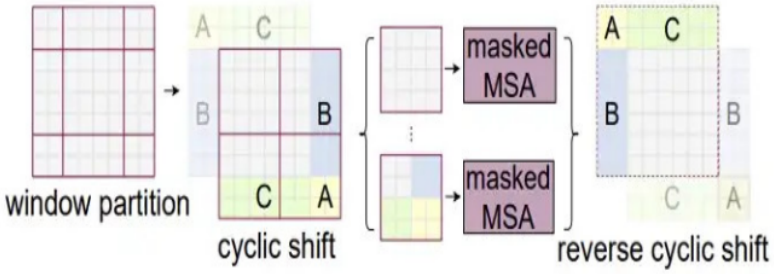

Figure 3.13: Cyclic Shifted Windows in Swin Transformer (?, ?)

SW-MSA successfully introduced connections between windows while masking prevents the attention computation from processing across the image boundary. Figure **??** show two consecutive Swin Transformer blocks with each one having a different partitioning scheme.

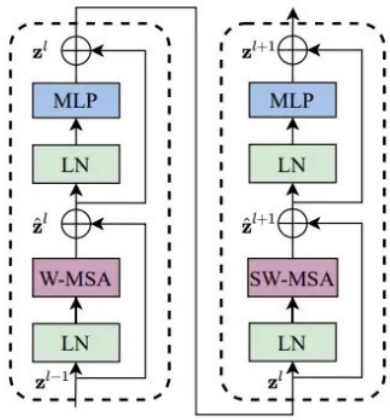

Figure 3.14: Two Successive Swin Transformer Blocks (?, ?)

The self-attention score using SW-MSA can be calculated as follows:

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d}+B})V \tag{3.10}$$

Swin Transformer V2 (?, ?) was introduced in 2021 and it is an improved version of Swin Transformer to increase scalability, accuracy and stability during training. As shown in figure **??**, the Layer Normalization block is moved from front to back of the residual unit and a new scaled cosine self-attention unit is used in place of the old matrix multiplication based self-attention. The new self-attention score can be calculated as:

$$Sim(\mathbf{q}_i,\mathbf{k}_j) = \frac{cos(\mathbf{q}_i,\mathbf{k}_j)}{\tau} + B_{ij} \tag{3.11}$$

Where B_{ij} is the relative position bias between pixel i and j; τ is a learnable scalar, non-shared across heads and layers.τ is set larger than 0.01. These approaches make the model insensitive to the magnitude (largeness) of activations.

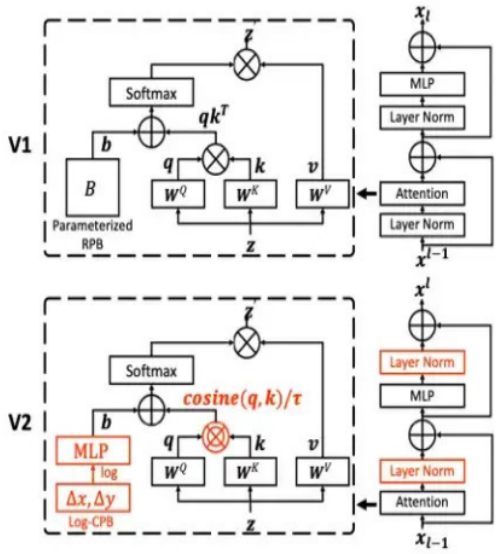

Figure 3.15: Difference Between Swin V1 and Swin V2 (?, ?)

3.5.3 UNetFormer

UNetFormer (?, ?) proposes a UNet-like Transformer for semantic segmentation task trained using the UAVid, Vaihingen, Potsdam and LoveDA datasets.UNetFormer has the same encoder-decoder framework as the other U-Net variants. Referring to figure **??** UNetFormer uses either a ResNet18 endcoder and a Transformer with an added global-local Transformer block (GLTB) that are used as its decoder.

ResNet18 consists of four stages and each stage will down-sample the feature map with a scale factor of 2. The feature maps generated by each stage are fused with the corresponding feature maps of the decoder using a 1x1 convolution with the channel dimension of 64. The semantic features produced by the ResNet18 are aggregated with the features generated by the GLTB of the decoder using a weighted sum operation.

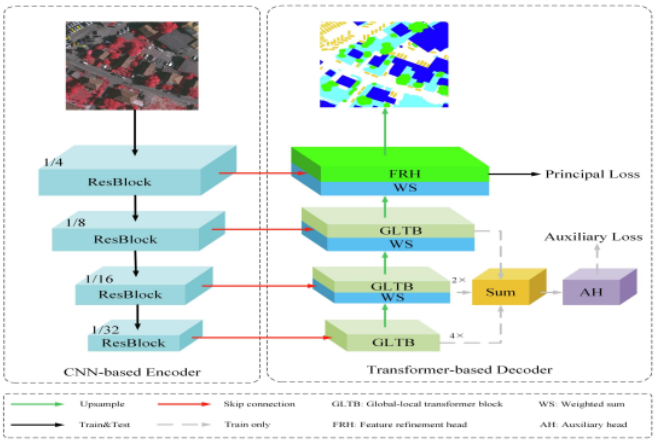

Figure 3.16: UNetFormer Architecture (?, ?)

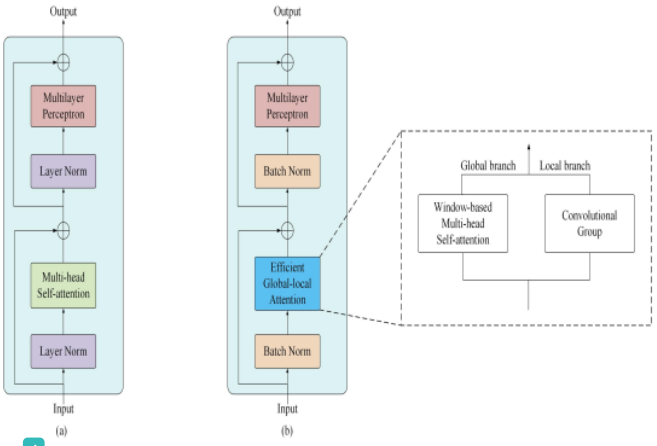

Figure 3.17: Illustration of (a) the standard Transformer block and (b) the Transformer block with GLTB (?, ?)

The GLTB is made up of the global–local attention, multi layer perceptron, two batch normalization layers and two additional operations. The global–local attention has two branches, the local branch and the global branch as shown in figure **??**. The local branch uses two parallel convolutional layers with kernel sizes of 3 and 1 to extract the local context. Two batch normalization operations are employed before the final sum operation.As illustrated in figure **??** the global branch uses the same window-based multi-head self-attention in Swin Transformer to capture the global context.

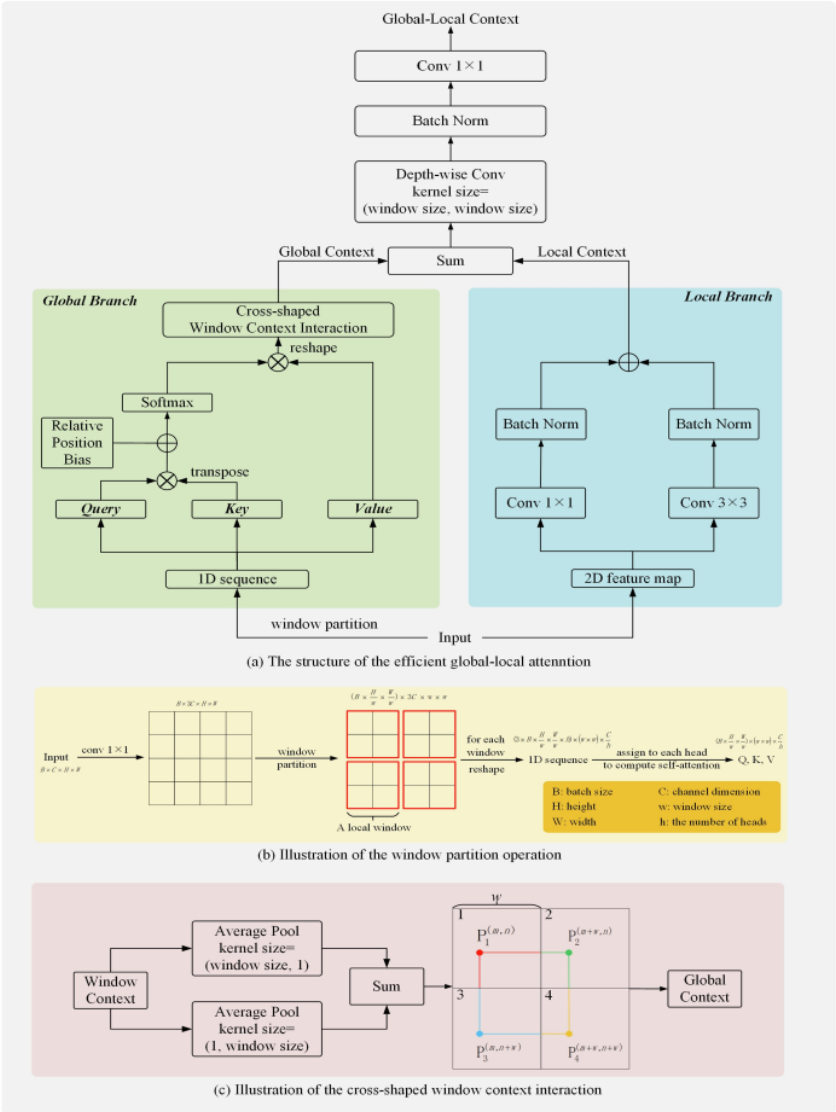

Figure 3.18: Cross-shaped Window Context Interaction in UNetFormer (?, ?)

The Swin Transformer uses a shifting windows mechanism to generate the relationship between the windows but this significantly increases the computation thus the author proposed a cross-shaped window context interaction to reduce the computation. The cross-shaped window context interaction captures the global context by fusing the two feature maps produced by a horizontal average pooling layer and a vertical average pooling layer. The horizontal average pool layer generates the horizontal relationship between windows, such as $Win_1 = H(Win_2)$. For any point $P_1^{m,n}$ win Win_1 its dependency with $P_2^{m+w,n}$ can be expressed as:

$$P_1^{m,n} = \frac{\sum_{i=0}^{w-m-1} P_1^{m+i,n} + \sum_{j=0}^{m} P_2^{m+w-j,n}}{w} \tag{3.12}$$

$$P_1^{m+i,n} = D_i(P_1^{m,n}) \tag{3.13}$$

$$P_2^{m+w-j,n} = D_j(P_2^{m+w,n}) \tag{3.14}$$

Where w is the window size and D denotes the self-attention computation. Similarly, the vertical relationship between Win_1 and Win_3 can be established in a similar way, $Win_1 = V(Win_3)$ and for Win_4 the relationship is $Win_1 = V(H(Win_4)) + H(V(Win_4))$.

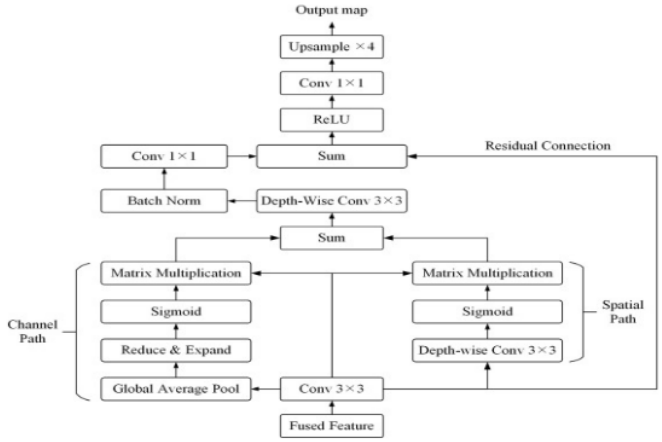

Figure 3.19: Feature refinement Head of UNetFormer (?, ?)

Finally, UNetFormer uses a Feature Refinement Head (FRH) just before the output is produced. The reason why FRH is needed is because while the ResNet encoder provides rich spatial details of satellite images, it lacks semantic content, on the other hand the deep GLTB provides precise semantic information, but with a coarse spatial resolution. FRH shrinks the semantic gap between the two features for improved accuracy. FRH received the fused feature as shown in **??** as its input. Then, two path are constructed to strengthen the channel-wise and spatial-wise feature representation as shown in figure **??**. The first path is called the channel path while the second path is called the spatial path. The channel path employs a global average pooling layer to generate a channel-wise attention map $\mathbf{C} \in \mathbb{R}^{1\times 1\times c}$, where c is the channel dimension. The reduce & expand operation contains two 1 x 1 convolutional layers, which first reduces the channel dimension c by a factor of 4 and then expands it to the original dimension. The spatial path utilizes a depth-wise convolution to produce a spatial-wise

attention map $\mathbf{S} \in \mathbb{R}^{h \times w \times 1}$, where h and w are the height and width of the feature map. The features generated by the two paths are fused before a 1x1 convolutional layer and an up sampling operation are applied as post processing.

? (?) also experimented with the choice of the encoder by substituting the ResNet18 with a list of lightweight Vision Transformers: ViT-Tiny (?, ?), Swin-Tiny (?, ?) and CoaT-Mini (?, ?). The results showed that by using lightweight Vision Transformers as the encoder provided very litle incrase of accuracy but greatly reduces the inference speed as shown in Table **??**.

Encoder	**Complexity (G)**	**Parameters (M)**	**Speed**	**mIoU**
ViT-Tiny	35.31	8.6	30.2	69.1
Swin-Tiny	104.4	28.0	28.8	70.6
CoaT-Mini	159.7	10.6	10.6	70.5
ResNet18	46.9	11.7	115.6	70.0

Table 3.2: Experiment of Different Encoders on UNetFormer

3.5.4 *LANet*

Local Attention Network (LANet) (?, ?)introduced two additional modules to improve semantic segmentation of satellite images. The first module is the Patch Attention Module (PAM) to enhance the embedding of local context information. The second module is the Attention Embedding Module (AEM) to improve the use of spatial information. Referring to figure **??** The high-level features produced by late layers of CNNs will pass through a PAM to enhance its feature, while the low-level features produced by early layers of a CNN are first enhanced by a PAM, before being embedded with semantic information from high-level features through AEM. The final output is the fusion of the outputs from the top and bottom channels.

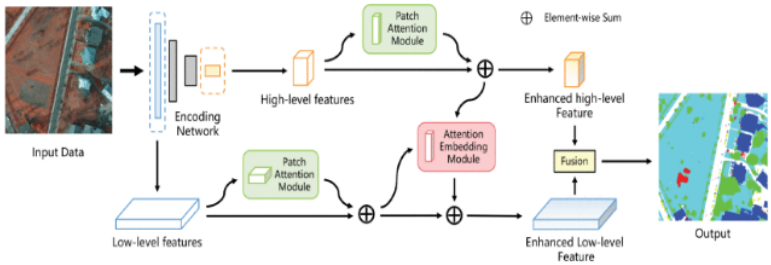

Figure 3.20: LANet Architecture (?, ?)

Referring to figure **??**, PAM will first generate a local descriptor for each channel of every patch. The descriptor z_c for the c th channel of a patch is calculated as

$$z_c = \frac{1}{h_p w_p} \sum_{i=1}^{h_p} \sum_{j=1}^{w_p} x_c(i, j) \tag{3.15}$$

where h_p and w_p is the horizontal and vertical size of the patch and x_c is the pixel value at the cth channel. In this way, a c-channel vector z_p will be generated, which contains the statistics describing the patch p. Then, they apply convolutional layer to learn an attention vector $a_p \in \mathbb{R}^{c \times h_p \times w_p}$. The entire gating operation to generate attention maps includes a 1 x 1 dimension-reduction convolution, 1×1 dimension-increasing convolution that recovers the feature dimension back to c and an up sampling operation.

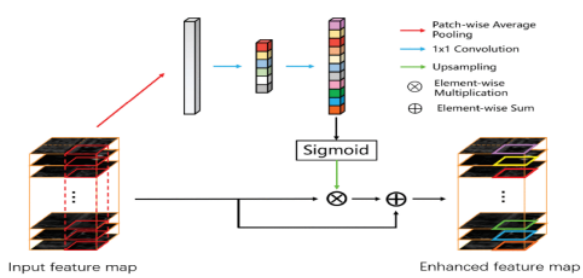

Figure 3.21: Patch Attention Module in LANet (?, ?)

In most networks, the most frequently used way of employing low-level features is to

concatenate them with high-level features, which brings only slight improvement in performance. The authors proposed AEM to bridge the gap between high-level and low-level features without sacrificing the spatial details of the latter. Figure **??** shows the implementation of AEM. First, they generate a high-level feature descriptor map, z_c and low-level descriptor map , x_c using the same formula in PAM. Then, an attention map, a_c is generated using average pooling, 1 x 1 convolution and up sampling. Finally a_c will be added to x_c to generate an improved low-level descriptor.

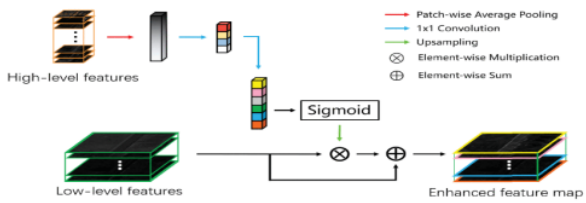

Figure 3.22: Attention Embedding Module in LANet (?, ?)

3.5.5 DC-Swin

A novel semantic segmentation scheme of densely connected (DC-Swin) (?, ?) proposed by combining Swin Transformer and a densely connected feature aggregation module (DCFAM). Swin Transformer is used as the encoder to extract the context information while DFCAM act as the decoder to restore the resolution and produce the segmentation map.

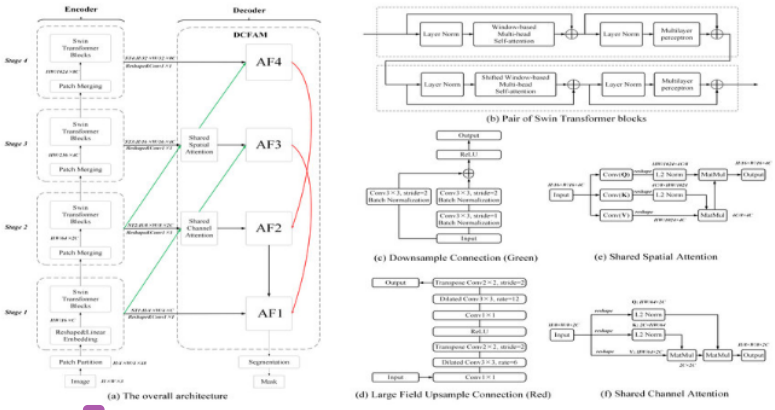

Figure 3.23: (a) Overall architecture of DC-Swin. (b) Pair of Swin Transformer blocks. (c) Downsample Connection. (d) Large Field Upsample Connection. (e) SSA. (f) SCA (?, ?)

Referring to figure **??**(a), the Swin Transformer split the input RGB image into non-overlapping patches as "tokens" using a patch partition module. This tokens are then fed into the multistage feature transformation. In stage 1, a linear embedding layer is deployed to project features to an arbitrary dimension C. Next, a pair of Swin Transformer as shown in figure **??**(b) are utilized to extract semantic features. In stages 2 to 4, the number of tokens is gradually reduced by patch merging layers along with the increasing depth of the network to produce a hierarchical representation. The outputs of the four stages are processed by a standard 1x1 convolution to generate four hierarchical Swin Transformer features. By adjusting the hyperparameters of tge Swin Transformer, they can construct backbones with different complexities.

DFCAM has two important modules namely the Shared Spatial Attention (SSA) and

a Shared Channel Attention (SCA) to enhance the spatial-wise and channel-wise relationship. Then, multi-level features are further integrated using the Downsample Connection and the Largefield Upsample Connection for improving multi-scale. According to figure **??** DFCAM connects the four hierarchical features with cross-scale connections and attention blocks to generate four aggregation features: AF1, AF2, AF3, and AF4.

The Downsample connection which is the green line in figure **??** connects the low-level and high-level transformer features for fusion and it can be defined as follow:

$$D_j^i(\mathbf{X}) = f_\sigma(f_\delta(\mathbf{X}) + f_\mu(f_\theta(\mathbf{X}))) \tag{3.16}$$

Where $\mathbf{X}$ is the input vector; f_σ is a ReLU activation function; f_δ and $f_m u$ are a 3x3 convolution layer with a stride of 2; f_θ is a 3x3 convolution layer with astride of 1 with each convolution layer involves a batch normalization operation; i and j are the number of the input channels and output channels.

The red line is the Large field Upsample Connection. It is used to capture multi-scale context effectively and can be mathematically expressed as:

$$LU_m^n(\mathbf{X}) = f_\varphi^{12}(f_\sigma(f_\varphi^6(\mathbf{X}))) \tag{3.17}$$

Where f_{φ}^{12} is a composite function that contains a standard 1x1 convolution, a dilated convolution with a dilated rate of 12, and a standard transpose convolution; f_{φ}^{6} is similar but with a dilated rate of 6; $m$and n are the number of the input channel and output channel.

SSA is used to model the long range dependencies and it is defined as:

$$SSA(\mathbf{X}) = \frac{\sum_n V(\mathbf{X}_{c,n}) + \frac{Q(\mathbf{X})}{||Q(\mathbf{X})||_2}((\frac{K(\mathbf{X})}{||K(\mathbf{X}))||_2})^T V(\mathbf{X})}{N + \frac{Q(\mathbf{X})}{||Q(\mathbf{X})||_2}\sum_n(\frac{K(\mathbf{X})}{||K(\mathbf{X}))||_2})^T} \tag{3.18}$$

Where $Q(\mathbf{X}), K(\mathbf{X}))andV(\mathbf{X}))$ are the convolutional operation to generate the Q, K and V; N is the number of pixels in the input feature maps; c is the channel dimension and n is the flattened spatial dimension.

SCA is used to extract the long-range dependencies between the channel dimensions and can be defined as:

$$SCA(\mathbf{X}) = \frac{R(\mathbf{X}_{c,n}) + (R(\mathbf{X}_{c,n}))\frac{R(\mathbf{X})}{||R(\mathbf{X}))||_2})^T \frac{R(\mathbf{X})}{||R(\mathbf{X}))||_2}}{N + \frac{R(\mathbf{X})}{||R(\mathbf{X})||_2}\sum_c(\frac{R(\mathbf{X})}{||R(\mathbf{X}))||_2})^T} \tag{3.19}$$

Where $R((X))$ is the reshape operation to flatten the spatial dimension.

Each of the aggregation features can be mathematically expressed as follows:

$$\mathbf{AF_4} = \mathbf{ST_4} + D_{384}^{768}(SSA(D_{192}^{384}(\mathbf{ST_2})) \tag{3.20}$$

$$\mathbf{AF_3} = SSA(\mathbf{ST_3}) + D_{192}^{384}(SSA(D_{96}^{192}(\mathbf{ST_1})) \tag{3.21}$$

$$\mathbf{AF_2} = SCA(\mathbf{ST_2}) + LU_{768}^{192}(\mathbf{AF_4}) \tag{3.22}$$

$$\mathbf{AF_1} = \mathbf{ST_1} + U(\mathbf{AF_2}) + LU_{384}^{96}(\mathbf{AF_3}) \tag{3.23}$$

where U is a bi-linear interpolation up sample operation with a scale factor of 2; $\mathbf{ST_1}, \mathbf{ST_2}, \mathbf{ST_3} and \mathbf{ST_4}$ are the Swin Transformer features from its respective block.

3.5.6 BANet

The architecture of Bilateral Awareness Network (BANet) is shown in figure **??** and it is made up of two paths: the dependency path and the texture path. The texture path is designed to extract textural feature and the dependency path is for capturing long range dependencies.

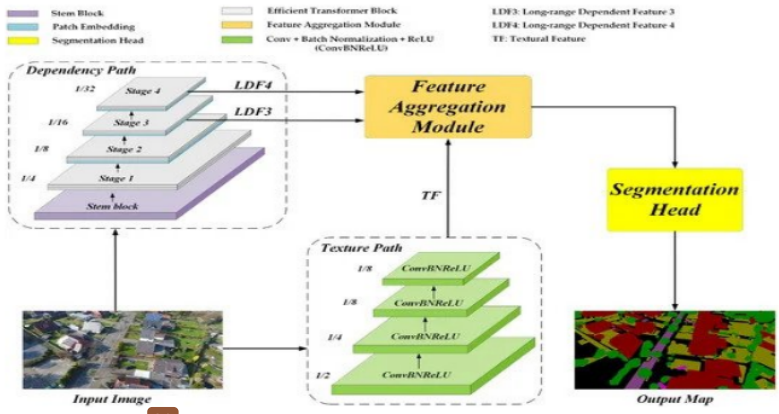

Figure 3.24: Architecture of Bilateral Awareness Network (BANet) (?, ?)

The dependency path is made up of a stem block and four Transformer stages to extract where each stage consists of two efficient transformer blocks (ETB). Stage 2,3 and 4 has a patch embedding (PE) operation. The dependency path will generate two long range dependent features namely the LDF3 and LDF4. The ETB is made up of ResT-Lite (?, ?) pre-trained on ImageNet. ETB as illustrated in figure **??** uses efficient multi-head self-attention (EMSA) instead of the usual multi-headed self attention.

The self-attention score from EMSA can be calculated as follows:

$$EMSA(\mathbf{Q},\mathbf{K},\mathbf{V}) = LP(IN(softmax(conv(\frac{\mathbf{QK}^T}{\sqrt{m}}))).\mathbf{V}) \tag{3.24}$$

where IN is the instance normalization function and LP stands for linear projection.

The texture path has four convolution layers and each convolutional layer is equipped

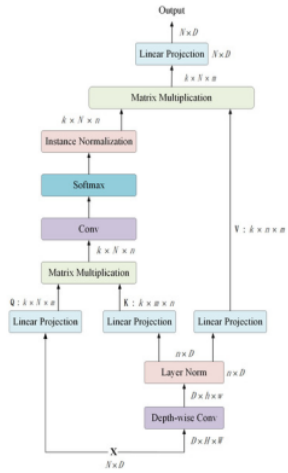

Figure 3.25: Efficient Transformer Block (?, ?)

with batch normalization and ReLU activation function. The down sampling factor is set to 8. The output for the texture path can be expressed as:

$$TF(\mathbf{X}) = T_4(T_3(T_2(T_1(\mathbf{X})))) \tag{3.25}$$

where each T represents a combined function consisting of a convolutional layer, a batch normalization operation, and a ReLU activation function. The convolutional layer in T_1 has a kernel size of 7 and a stride of 2, which expands the channel dimension from 3 to 64. For T_2 and T_3, the kernel size and stride are 3 and 2, respectively. The channel dimension is kept as 64. Lastly for T_4, the convolutional layer is a standard 1x1 convolution with a stride of 1, expanding the channel dimension from 64 to 128. Thus, the output textural feature is down-scaled 8 times and has a channel dimension of 128.

The last module in BANet is the Feature Aggregation Module (FAM) and it is designed to leverage the benefits of the dependent features and texture features for powerful feature representation. FAM is illustrated in figure **??**. The input features for the FAM include the LDF3, LDF4 and TF. To fuse those features, they first employ an attentional embedding module (AEM) to merge LDF3 and LDF4. Then, the merged feature is upsampled to concatenate with the TF to produce the aggregated feature. Finally, the linear attention module is deployed to reduce the fitting residual of the aggregated feature.

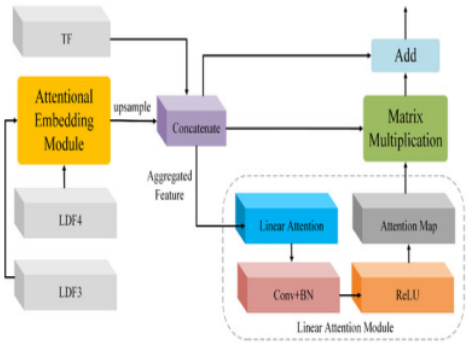

Figure 3.26: Feature Aggregation Module (?, ?)

3.5.7 AESwin-UNet

Adaptive Enhanced Swin Transformer with U-Net (AESwin-UNet) (?, ?) uses a combination of Vision Transformer and U-Net for semantic segmentation of satellite images. As shown in figure **??** instead of using CNN for the encoder, they opted to use Enhanced Swin Transformer instead. For the decoder, each step involves an upsampling of the feature map followed by a 2×2 deconvolution, a concatenation with

a feature map from the encoder, and two 3×3 convolutions.

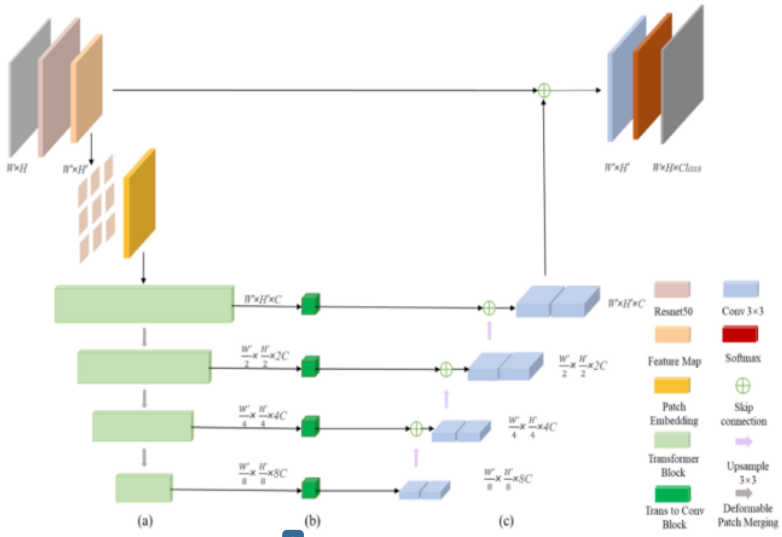

Figure 3.27: Architecture of Adaptive Enhanced Swin Transformer with U-Net (AESwin-UNet) (?, ?)

Enhanced Swin Transformer is a Swin Transformer with enhanced multi-head self-attention (EMHSA) and deformable adaptive patch merging layer (DeforAPM). The EMHSA calculate the attention score by fusing the low-level features and the global channel context. The enhanced attention can be formally defined as:

$$Attention(Q,K,V) = Softmax(\frac{att(QK^T)}{\sqrt{d_k}} + B)V \tag{3.26}$$

Model \ Dataset	Potsdam	Vaihingen	LoveDA	UAVid
UNetFormer (?, ?)	-	-	-	-
LANet (?, ?)	91.95	88.09	-	-
DC-Swin (?, ?)	93.25	90.71	-	-
BANet (?, ?)	92.50	89.58	-	-
AESwin-Unet (?, ?)	-	-	-	-

Table 3.3: F1-score achieved by each model.

Model \ Dataset	Potsdam	Vaihingen	LoveDA	UAVid
UNetFormer (?, ?)	85.5	81.6	-	70
LANet (?, ?)	-	-	-	-
DC-Swin (?, ?)	87.56	83.22	-	-
BANet (?, ?)	86.25	81.35	-	64.6
AESwin-Unet (?, ?)	-	-	66.1	-

Table 3.4: mIoU score achieved by each model.

Model \ Dataset	Potsdam	Vaihingen	LoveDA	UAVid
UNetFormer (?, ?)	-	-	-	-
LANet (?, ?)	90.84	89.83	-	-
DC-Swin (?, ?)	92.00	91.63	-	-
BANet (?, ?)	91.06	90.48	-	-
AESwin-Unet (?, ?)	-	-	53.96	-

Table 3.5: Accuracy score achieved by each model.

3.5.8 Summary of Semantic Segmentation Using Transformer

The three tables below contain the F1-score, mIou score and accuracy score of each semantic segmentation model reviewed.

3.6 *Advantages of Vision Transformer for Semantic Segmentation of Satellite Images*

1. ***General Modelling Capability***

   There are two aspects that gives a vision transformer general modelling capabilities. The first one being performing a task using a transformer can be interpreted as working on a fully connected graph. Any concept, can be represented by the nodes in a graph, and the relationship between concepts are represented by the graph edges.

   Every task in computer vision deals with processing two basic granular elements: pixels and objects. Thus, there are three type of relationship that can be found: pixel-to-pixel, object-to-object and pixel-to-object. The transformer's attention mechanism allows researchers to include all 3 types of relationships in one network. For examples, networks such as DETR (?, ?), LearnRegionFeat (?, ?) and RelationNet++ (?, ?) model the relationship between object and pixel to achieve state of the art performance in semantic segmentation task.

2. ***Attention Mechanism Complements Convolution***

   Unlike convolution which is a local operation, the attention mechanism is a global one which means it can model the relationship between all the pixels in an image. CNN and Transformers complement each other very well and works such as DETR (?, ?) and UNetFormer (?, ?) are evidence of this claim.

3. ***Transformers Make It Easier for Parallel Processing***

   The sequential nature of Transformers makes it easier for researchers to utilizes parallel processing with TPUs compared when they are training CNN models. There are less steps to do parallel processing when training a Transformer model (?, ?). This would save a lot of time and effort.

4. ***Transformers are Scalable***

   Transformers has shown excellent scalability in Natural Language Processing. However, when transformers were initially used for computer vision, a lot of researchers doubted it ability to scale because all of the networks are dense as it has to process every pixel as input. Fortunately, there are recent works that shows we can improve the scalability of transformer by increasing its efficiency and reducing its computational load. Vision MoE from Google managed to match the performance of SOTA networks, while slashing the compute time into half by using a sparse network. It managed to train a 15 billions parameter model with 90.35% accuracy on ImageNet dataset (?, ?). Figure **??** shows the recorded number of parameters in Vision Transformer models from 2018 until 2022

3.7 Pruning Vision Transformer

Pruning a deep learning model is a technique to reduce the number of its parameters by removing redundant values in order to reduce storage, memory, battery, and hardware consumption without sacrificing its performance. This would allow us to deploy the

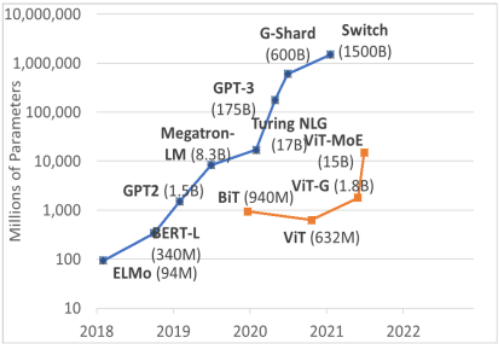

Figure 3.28: The size records of Vision Transformer in recent years. Retrieved from paperswithcode.com

model and guarantee that it would run smoothly on every device. This section would discuss several techniques to prune a Vision Transformer.

Pruning can be divided into two schemes, structured pruning and unstructured pruning. Structural pruning alters the structure of the network by physically removing grouped parameters, while unstructured pruning remove the redundant weights after the training is finished without any modification to the network (?, ?). Unstructured pruning are easier to deploy as it does not relies on specific techniques, thereby making it more popular.

An example of unstructured pruning is to set a minimum threshold of weight at the end of a layer and any values that is smaller than that threshold will be change to zero. This method would be easily done using torch.nn.utils.prune package provided by Pytorch however it greatly decrease the performance of the model. We will need a

more comprehensive and systematic approach to prune our Vision Transformer.

Structured pruning is a challenging task deep neural networks are made up of a large number of basic modules like convolution, normalization, or attention, that are intrinsically connected. As an example, if we seek to remove only one layer from a Transformer, we have to inevitably take care of its inter- dependencies to all layers simultaneously, otherwise we will end up with a broken network.

The first comprehensive method to prune a Vision Transformer is to learn its associated importance score before pruning its parameters (?, ?). For the features $x \in \mathbb{R}^{n \times d}$, where n is the number of features that we want to prune and d is the dimension of each feature, the proposed method will preserve the generated important features and remove the useless ones. Suppose the optimal importance score is $a^* \in \mathbb{R}^d$. This method is applied on DeiT-base (?, ?) trained on ImageNet-1k and ImageNet 100. The result of pruning Vision Transformer using the methods proposed by ? (?) is shown in Table **??**.

Dataset	**Model**	**Pruning Rate**	**Params (M)**	**Flops (B)**	**Accuracy**
ImageNet-100	Deit	0	86.4	17.6	98.94
ImageNet-100	Deit	0.2	66.1	13.7	98.36
ImageNet-100	Deit	0.4	47.3	10.0	98.04
ImageNet-1k	Deit	0	86.4	17.6	81.8
ImageNet-1k	Deit	0.2	67.3	13.8	81.3
ImageNet-1k	Deit	0.4	48.0	10.0	80.7

Table 3.6: The Result of Pruning Vision Transformer.

? (?) identified that there are three ways to compress a model. The first method is distillation which utilize targeted distillation to produce smaller models with a dense structure that is fast on standard hardware. Distillation has been proven successful in language models such as DistilBERT (?, ?) and TinyBERT (?, ?) which are all compressed versions of BERT (?, ?). The second method is pruning and third method is a combination of pruning and distillation. Distillation requires a very tedious engineering process and it is very hard to achieve.

? (?) also categorized pruning into three methods. The first one being the score-based pruning methods (?, ?) which modify the model by introducing score parameters **S** for each parameter **i** and replace the original parameter matrices with a masked version. The simplest one would be to zero-out the parameters with the lowest value as discussed earlier. The second pruning method is movement pruning which encourages the model to fine-tune the parameters while lowering the scores of unimportant parameters and thus encouraging more sparsity.

The third pruning method is block movement pruning (?, ?) which is a hybrid of movement and block pruning. They proposed to partition each matrix in the Transformer into multiple fixed-sized blocks. Every block acts as an individual group in the regularization with a shared score parameter and a different mask would be applied to each block. This method has not been used alongside a Vision Transformer but they did use it to train BERT (?, ?), an encoder-only Transformer language model with 110M

parameters for question answering and sentence classification. Overall, they managed to reduce the number of parameters by a factor of 5.91 and trained it 2.24 times faster with only a 1.99% drop in F1 score. The full result of using movement block pruning can be seen in table **??**.

Model	**Size**	**Compression**	**Speed**	**F1**
BERT	85M	1.00	1.00	88.5
TinyBert	42M	2.00	1.88	87.5
DistilBERT	42M	2.00	2.06	86.9
Block Movement	20M	4.09	1.97	88.07
Block Movement	15M	5.44	2.26	87.15
Block Movement	14M	5.91	2.24	86.51

Table 3.7: Parameters Count, Compression Rate, Speed gain relative to BERT-base and F1.

CHAPTER 4

THEORETICAL FRAMEWORK

In this chapter we would start with a brief history of artificial intelligence research that are directly or indirectly related to the innovation of transformers. Then, the second section would covers the foundations of deep learning which includes feed-forward neural networks, activation functions, loss functions and evaluation metric. The section would cover the basics of transformers. The final section would include the potential challenges and limitations of this project.

4.1 A Brief History of Deep Learning

Modern deep learning as we know it today can be traced back to when Frank Rosenblatt introduced the perceptron in 1959, referring to it as the "Mark I Perceptron" shown in figure **??**. Given an input, the perceptron will generate an output based on a linear thresholding logic. The weights in the perceptron were updated and learned by iteratively reducing the difference between the generated output and the desired output and passing in a new input.

The perceptron never took off in popularity because Marvin Minsky and Seymour Papert showed the limitations of perceptrons in learning the simple XOR function. In

1986, David Rumelhart, Geoff Hinton, and Ronald Williams showed how by incorporating "hidden" layers, a multi-layer perceptron can be used to overcome the weakness of perceptrons in learning complex patterns. Multi-layer perceptrons are also known as neural networks.

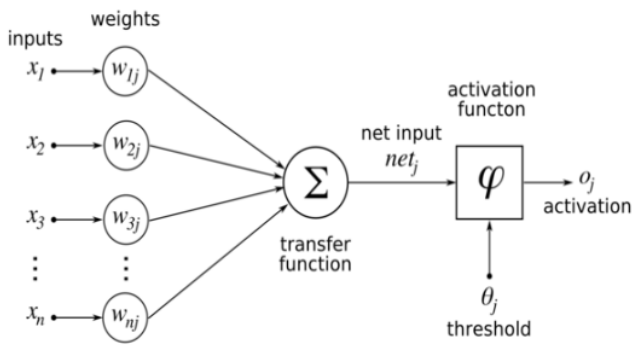

Figure 4.1: Rosenblatt Perceptron (?, ?)

LeCun et al., published a method to recognized hand-written digits and it was utilized by the U.S. Postal Service (?, ?), making it the first neural network model that received widesread adoption . This is a huge milestone for deep learning, proving the usefulness of convolution operations and weight learning the features in computer vision.

However, there are still a lot of flaws with neural networks. Take for example the backpropagation algorithm, has a number of issues such as vanishing gradients, exploding gradients, and the inability to learn long-term information. Hochreiter and Schmidhuber showed how Long short-term memory (LSTM) architecture could overcome shortcomings of backpropagation over time.

LeCun et al. also showed the advantages of deep learning through more complex neural networks architectures such as convolutional neural networks (CNNs), restricted Boltzmann machines (RBMs), and deep belief networks (DBNs). They also showed the importance of techniques such as unsupervised pre-training with fine-tuning, thus inspiring the next wave of deep learning. Li et al., launched ImageNet, which was the most extensive collection of labelled images and highlighted the importance of robust dataset to train a deep learning model for computer vision task.

Mikolov et al. and Graves proposed language models using Recurrent Neural Networks (RNN) and LSTM, which later became the building blocks for many natural language processing (NLP) architectures. Sequence-to-sequence framework became the core architecture for a wide range of NLP tasks. Bahdanau et al. proposed the attention mechanism to overcome the bottleneck issue with sequence-tosequence model. Attention mechanismplays a crucial role in subsequent evolution of Transformers.

In 2017, Transformers were formally introduced in the "Attention is All You Need" paper (?, ?) and became the most popular architecture in NLP. In 2020, Vision Transformers (?, ?) introduced a method to adapt Transformers for computer vision by splitting the input image into patches and represent them as vectors.

4.2 *Convolutional Neural Networks and Its Application in Semantic Segmentation of Satellite Images*

4.3 *Introduction to Transformers*

This section would lays out the various building blocks of transformers such as attention, multi-head attention, positional encodings, residual connections, and encoder-decoder architecture. Subsections **??** and **??** would give a brief introduction to Recurrent Neural Networks (RNN) and sequence-to-sequence model (seq2seq) and the subsequent subsections would explain why attention unit and Transformers are born from RNN and seq2seq.

4.3.1 *Encoder-Decoder Architecture*

Because Transformers has its origin in NLP that relies on sequential input, the use of encoder-decoder architecture as shown in **??** such as RNN is very common. The encoder module takes a variable-length sequence and converts it into a fixed-length output-state while the decoder module takes a fixed-length state and converts it back into a variable-length output (?, ?).

4.3.2 *Sequence-To-Sequence*

Sequence-to-sequence (seq2seq) is a type of deep learning approach where the input is in a sequence like a sentence and the context for each item is the output from the previous step. Seq2seq is the most famous model in Natural Language Processing.

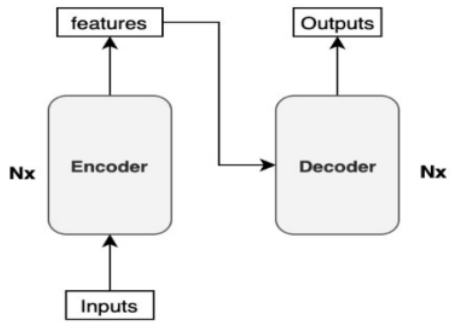

Figure 4.2: Encoder-Decoder Architecture (?, ?)

Suppose that we have an input sequence $x_1 ... x_t$, embedding mapping will transform the input sequence into vectors $\mathbf{x}_1 ... \mathbf{x}_t$. A unidirectional RNN at any time t with a previous hidden state $\mathbf{h}_{t-1}$ and input $\mathbf{x}_t$ will generate a new hidden state:

$$\mathbf{h}_t = tanh(\mathbf{W}\mathbf{h}_{t-1} + \mathbf{W}\mathbf{x}_t) \tag{4.1}$$

The decoder has the output of the encoder and will generate the decoded output at each step. RNNs have issues with vanishing and explosive gradients. Also RNNs dependence on previous time steps makes it very difficult to parallelize.

4.3.3 Attention Mechanism

Attention mechanism was first introduced by (?, ?) and it is the most fundamental part of a Transformer which was introduced by Google in 2017 (?, ?). The attention

unit will take input vectors $\mathbf{x}_1 ... \mathbf{x}_t$ to produce output vectors $\mathbf{y}_1 ... \mathbf{y}_t$. $\mathbf{y}_i$ is the weighted average of all the input vectors

$$\mathbf{y}_i = \sum w_{ij} \mathbf{x}_j \tag{4.2}$$

Where w_{ij} is the dot product of x_i and x_j. The dot product gives us a real numbered value so we apply a softmax function to map the values to [0,1] and to ensure that they sum to 1 over the entire sequence:

$$w_{ij} = \frac{expw_{ij}}{\sum_j expw_{ij}} \tag{4.3}$$

4.3.4 Queries, Keys and Values

Every input vector $\mathbf{x}_i$ is used in three different ways in the self attention operation:

1. It is compared to every other vector to establish the weights for its own output $\mathbf{y}_i$
2. It is compared to every other vector to establish the weights for the output of the j-th vector $\mathbf{y}_j$
3. It is used as part of the weighted sum to compute each output vector once the weights have been established.

These roles are called the **query**, **key** and **value**. Three new weight matrices are de-

veloped for each role namely: $\mathbf{W_q}, \mathbf{W_k}, \mathbf{W_v}$. These are the weights that are going to be learned by backpropagation which is going to be discussed in the next section. Since the average value of the dot product grows with the embedding dimension d_k, it helps to scale the dot product back a little to stop the inputs to the softmax function from growing too large. The attention mechanism can be formally described as:

$$\mathbf{Q} = \mathbf{W}_q \mathbf{x}_i \tag{4.4}$$

$$\mathbf{K} = \mathbf{W}_k \mathbf{x}_i \tag{4.5}$$

$$\mathbf{V} = \mathbf{W}_v \mathbf{x}_i \tag{4.6}$$

$$ATTENTION(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{QK}^T}{\sqrt{d_k}})\mathbf{V} \tag{4.7}$$

$$\mathbf{y} = \sum ATTENTION(\mathbf{Q}, \mathbf{K}, \mathbf{V})\mathbf{x} \tag{4.8}$$

Where $\mathbf{Q}$ is the query, $\mathbf{K}$ is the key, $\mathbf{V}$ is the value, $\mathbf{d}_k$ is the number of heads and $\mathbf{y}$ is the output vector. The attention mechanism is shown in figure **??**

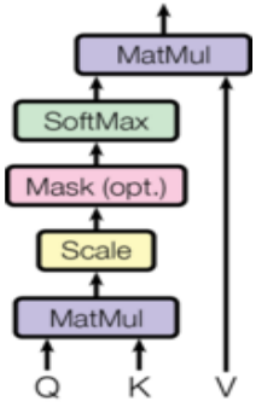

Figure 4.3: Illustration of Attention Mechanism Unit (?, ?)

4.3.5 Multi Head Attention

With the introduction of the softmax function in the previous section, we limit the attention value to either 0 or 1 when in reality attention can be anywhere in between. This is a negative consequence of using the softmax function. The easiest solution is to have multiple attention heads running at once. The multi-attention head is shown in figure **??**.

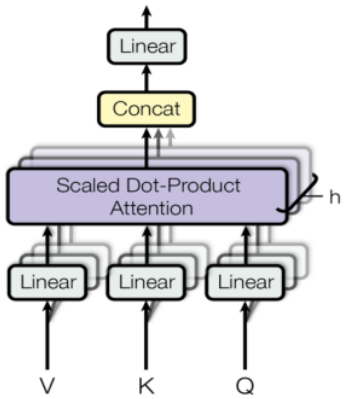

Figure 4.4: Illustration of Multi-head Attention (?, ?)

Multi-attention head can be thought of as multiple copies of the self-attention mechanism running in parallel, each with their own key, value and query values. However, generating **K**, **Q** and **V** for each head would be computationally expensive. We can avoid this problem by rescaling the weight matrix by dividing it with the square root of the number of heads and concatenating the **K**, **Q** and **V** matrix as shown in ? (?). The entire Transformer architecture with multi-head attention head is shown in figure **??**

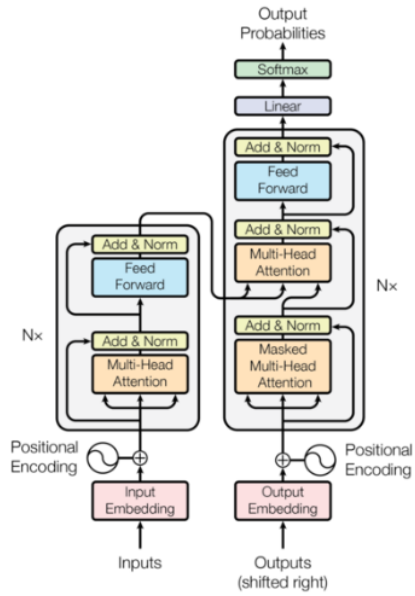

Figure 4.5: Illustration of Transformer Architecture (?, ?)

4.4 Activation Functions

This section contains the three most commonly used asctivation functions in deep learning, namely the sigmoid function, the ReLU function and the softmax function.

4.4.1 Sigmoid Function

The sigmoid function is a special form of the logistic function and is given by:

$$\sigma(x) = \frac{1}{1 + exp(-x)} \tag{4.9}$$

The sigmoid function would map any real numbered input to eaither 0 or 1. Figure **??** shows a plot of the sigmoid function.

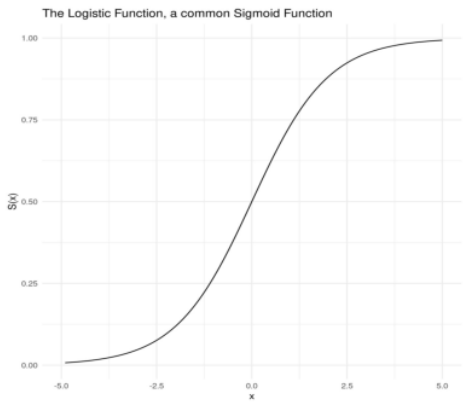

Figure 4.6: Sigmoid Function

4.4.2 Rectified Linear Unit Function

There are several issues when we use sigmoid function for deep learning. The first issue is the function is only really sensitive to changes around its mid-point of its input. The second issue is that sigmoid function is computationally intensive. In order to use optimization technique such as SGD or ADAM with backpropagation to train deep neural networks, we need a non-linear activation function that acts like a linear function. The function must also be more sensitive to changes in the input.

Rectified Linear Unit (ReLU) function returns the input value directly, or the value 0 if the input is 0 or less. Figure **??** shows a plot of the ReLU function. ReLu function

can be written as:

$$f(x) = max(0, x) \tag{4.10}$$

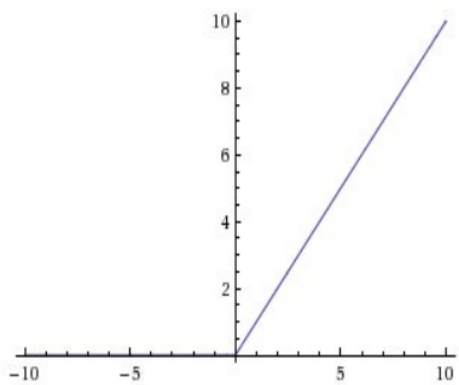

Figure 4.7: ReLU Function

4.4.3 Softmax Function

The softmax function is a function that takes a vector of K real values and ouput a vector of K real values that sum to 1. The input values can be any real numbers and softmax would map them into values between 0 and 1, so that they can be interpreted as probabilities. The softmax function is also known as the multi-class logistic regression and can be written as follows:

$$\sigma(z) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{4.11}$$

where z_i is an element of the input vector and the denominator is the sum of the input

vector.

4.5 Backpropagation

The goal of backpropagation is to compute the partial derivative $\frac{\partial C}{\partial w}$ which is the partial derivative of C with respect to weights and $\frac{\partial C}{\partial b}$ which is the partial derivative of C with respect to bias. The cost function C can be written as:

$$C(X, \theta) = \frac{1}{2n} \sum_{x} ||\hat{y}(x) - y(x)||^2 \tag{4.12}$$

where n is the total number of training examples; the sum is over individual training examples, x; $\hat{y}(x)$ is the corresponding desired output; and y(x) is the vector of activations output from the network when x is input.

In this example C is the Euclidean distance between $y(x)$ and $a^L(x)$ which is a very common choice when training a neural network.Backpropagation attempts to minimize C with respect to the neural network's weights by calculating, for each weight w_{ij}^k, the value of $\frac{\partial C}{\partial w_{ij}^k}$. At each iteration, the weights and biases (collectively denoted as θ) is updated according to the learning rate, α as follows:

$$\theta^{t+1} = \theta^t - \alpha \frac{\partial C(X, \theta^t)}{\partial \theta} \tag{4.13}$$

where θ^t is the weights and biases of the neural network at iteration t in gradient descent.

Since C can be decomposed into a sum over individual error terms for each individual input-output pair, the derivative can be calculated with respect to each input-output pair individually and then combined at the end (since the derivative of a sum of functions is the sum of the derivatives of each function):

$$\frac{\partial C(X,\theta)}{\partial w_{ij}^k} = \frac{1}{N}\sum_x \frac{\partial}{\partial w_{ij}^k}(\frac{1}{2}(\hat{y}_d - y_d)^2) = \frac{1}{N}\sum_x \frac{\partial C_d}{\partial w_{ij}^k} \tag{4.14}$$

The derivation of the backpropagation algorithm begins by applying the chain rule to the partial derivative of C. Lastly, the weights are updated as follows:

$$\Delta w_{ij}^k = -\alpha \frac{\partial C(X,\theta)}{\partial w_{ij}^k} \tag{4.15}$$

4.6 Potential Challenges and Limitations

1. **Weak Supervision**

   (?, ?) identified that a lot of semantic segmentation models doesn't work well under weak supervision. In this paper weak supervision is defined as using dataset that fulfil at least one of this criteria:

a) *Incomplete Supervision* - The dataset is small and insufficient to train a good model.

b) *Inexact supervision* - The labelling is not as exact as necessary which usually occurs in land cover labels with low resolution.

c) *Inaccurate supervision* - The labels are wrong.

2. **Inadequate Computing Resource**

   Training a Vision Transformer model requires a lot of computation resource. Several papers took multiple days to train their model even though they are using multiple GPUs in parallel.

3. **Large Model Size**

   The trained model would be quite large because we used a Transformer instead of CNN. A large model would take a long time to train and evaluate. Furthermore, it would also be harder to deploy it in the real world due to its size.

CHAPTER 5

METHODOLOGY

The implementation of this project is split into five different steps namely Data Collection, Data Preprocessing, Model Design and Implementation, Model Evaluation and Model Refinement. The general flow of the research methodology can be seen in figure ??.

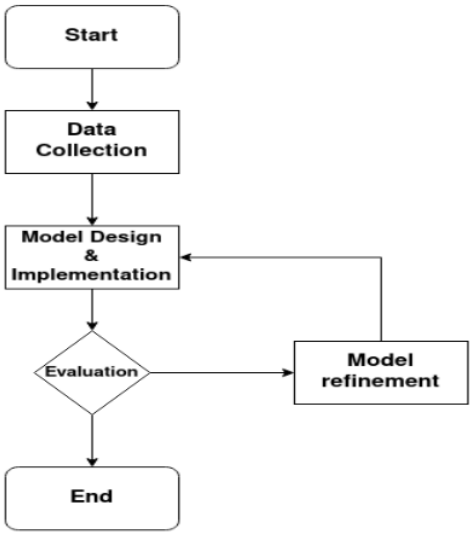

Figure 5.1: Flowchart of Proposed Research Methodology

5.1 Data Collection

There were 12 different datasets considered at the beginning of this project and their details can be found in Chapter 2. In the end, we decided to choose LoveDA dataset

(?, ?) because of several reasons. The first reason is it has one of the lowest spatial resolution with 0.3m. The second reason is it has a total of 5,987 samples which we considered as a good number of samples. Te third reason is the images in LoveDA dataset is actually collected using satellites unlike several other dataset that is a mixed of images collected through satellites and drones. The final reason is the images on LoveDA comes in PNG format which is much easier to work with compared to some other dataset that comes in GeoTIFF format. Although GeoTIFF carry more information, we concluded that the time and effort required to process and train models using GeoTIFF format is too much.

LoveDA dataset was introduced in Chapter 2 but here we are going to give a more detailed statistical analysis on the dataset. Table **??** explains the description of the classes in the dataset.

Classes	***Description***
Background	Any objects that don't really belong in the rest of the class.
Building	Include objects such as houses, schools, farmhouses
Road	Include tar roads and dirt roads.
Water	Include rivers, lakes, sea, pools and drains.
Barren	Unused land that is not used for any residential, commercial or agriculture activities.
Forest	Forest.
Agriculture	Farms.

Table 5.1: Classes Description for LoveDA Dataset.

Each pixel in the dataset is labelled as it is made specifically for semantic segmentation task. The images in LoveDA dataset are separated into two groups; the rural images

and the urban images. The sample images and maasks are shown in Figure **??** and **??**.

The corresponding colour to each class in the mask is shown in Table **??**.

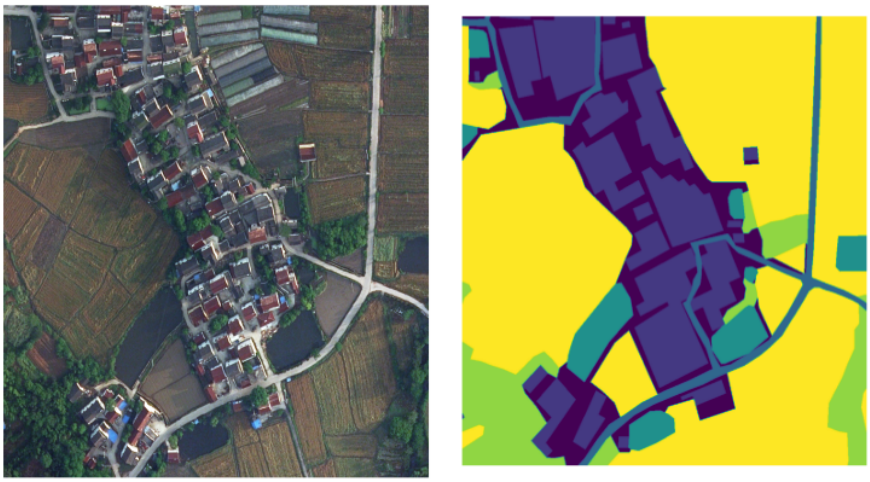

Figure 5.2: An Image and Mask From The Rural Set (?, ?)

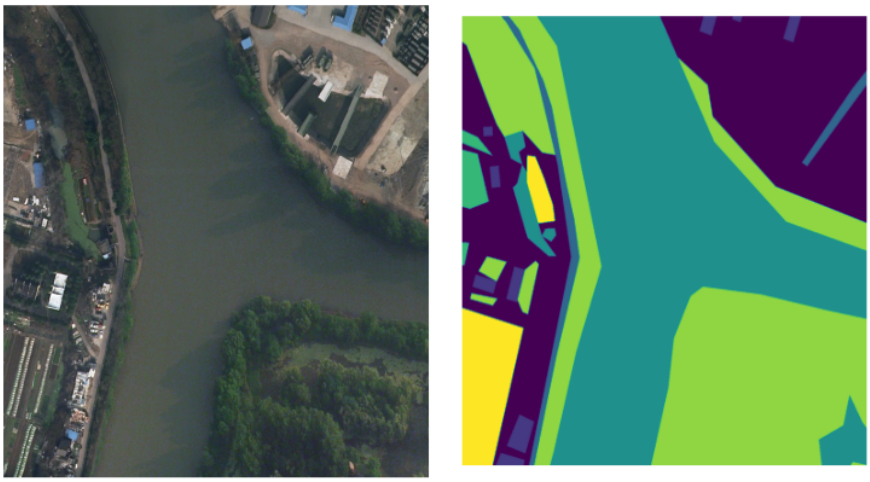

Figure 5.3: An Image and Mask From The Urban Set (?, ?)

Class	Colour
Agriculture	
Forest	
Road	
Water	
Building	
Background	
Barren	

Table 5.2: Classes Colour for LoveDA Mask.

Each of the group is already split into a training set, a validation set and a testing set. To be specific, the training set is composed of 2522 images , the validation set is composed of 1669 images and the 1796 images is used for testing. The structure of the dataset is shown in figure **??** .

```
LoveDA
├── Train
│   ├── Urban
│   │   ├── images_png (original images)
│   │   ├── masks_png (original masks)
│   │   ├── masks_png_convert (converted masks used for training)
│   │   ├── masks_png_convert_rgb (original rgb format masks)
│   ├── Rural
│   │   ├── images_png
│   │   ├── masks_png
│   │   ├── masks_png_convert
│   │   ├── masks_png_convert_rgb
├── Val (the same with Train)
├── Test
├── train_val (Merge Train and Val)
```

Figure 5.4: The Structure of LoveDA Dataset

Figure **??**, **??**, **??** and **??** show the box plots of the classes in the training and test set while figure **??**, **??**, **??** and **??** show the distribution of the same datasets. We observed that there is definitely class imbalanced in all four sets of the data. Rural areas have more agriculture and forest as expected. Urban area have more roads and buildings as the buildings in urban area are bigger and situated closely together while the roads in urban area are generally wider than the ones in rural area. Water are often

exist in the form of large-scale rivers, drains or lakes in the urban scenes, while small ponds and ditches are more common in the rural scenes. In urban area, the agricultural part is often found in the gaps between the buildings while the rural area have a more continuous and large agricultural lands.

There are two concerning observations from the graphs. The first one being the amount of agriculture in the urban test dataset is larger than expected. the second concern being both rural and urban areas have a very large amount of background area and the background have very different meaning in these two areas. The background in urban area are usually the sidewalks, parking lots and parks while the background in rural area are usually the unused gaps between agricultural area.

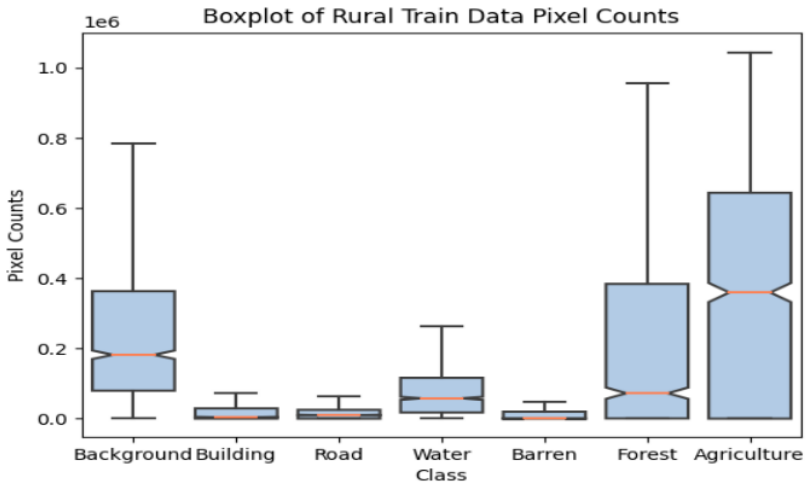

Figure 5.5: Box plot of the Pixel Counts of the Rural Train Dataset

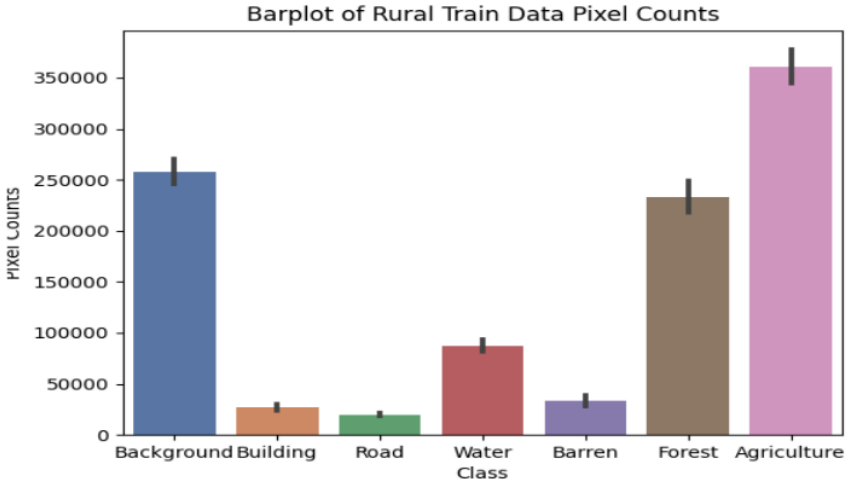

Figure 5.6: Bar plot of the Pixel Counts of the Rural Train Dataset

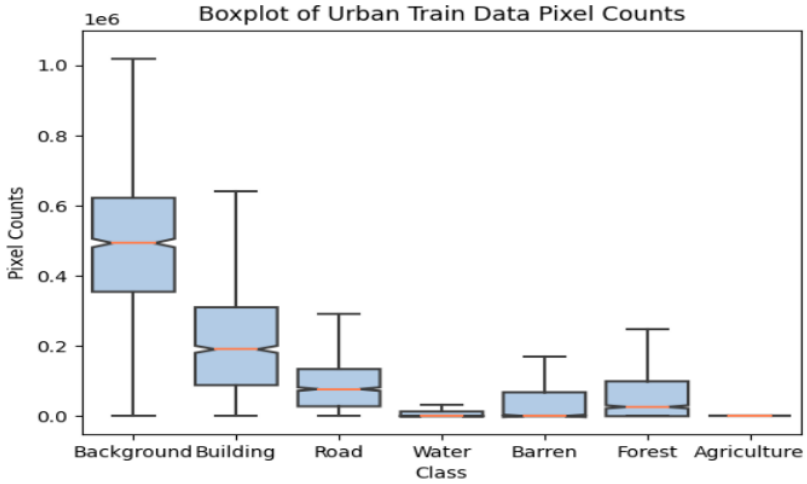

Figure 5.7: Box plot of the Pixel Counts of the Urban Train Dataset

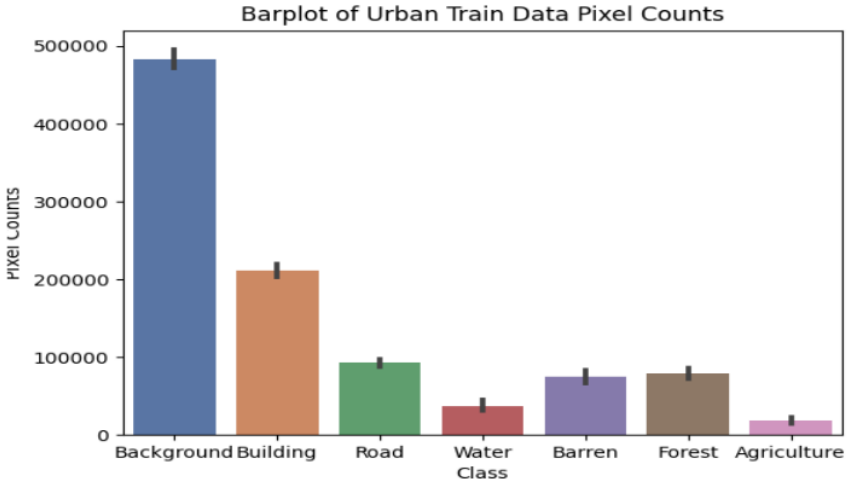

Figure 5.8: Bar plot of the Pixel Counts of the Rural Train Dataset

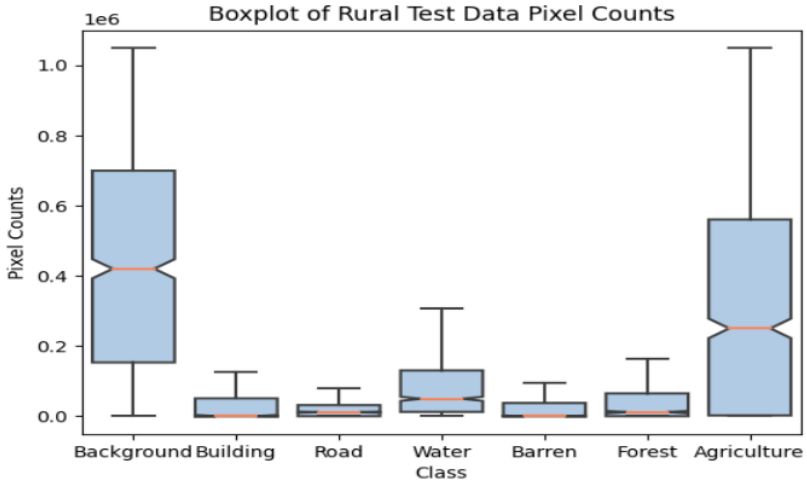

Figure 5.9: Box plot of the Pixel Counts of the Rural Test Dataset

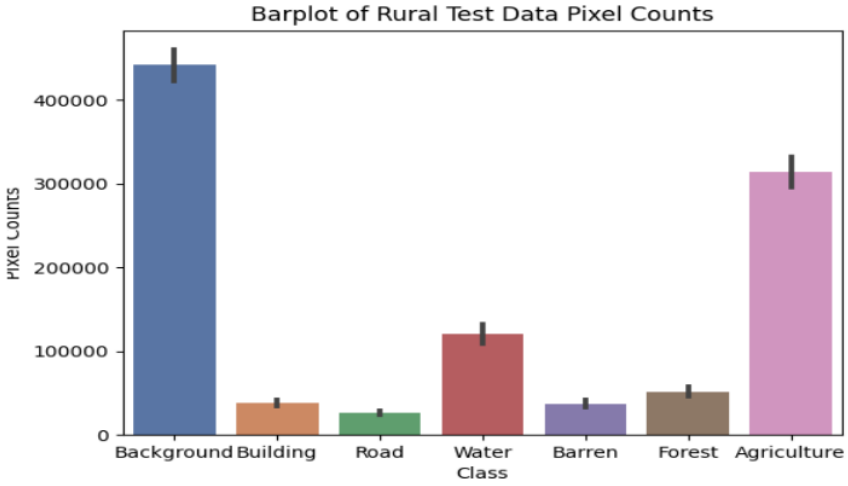

Figure 5.10: Bar plot of the Pixel Counts of the Rural Test Dataset

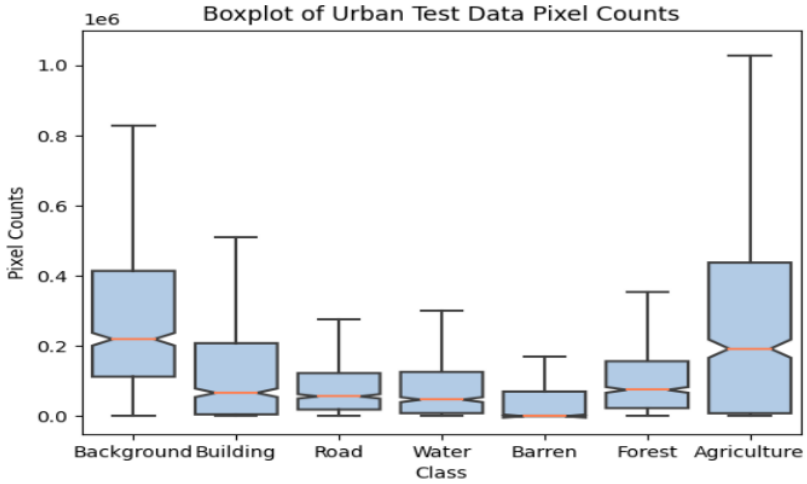

Figure 5.11: Box plot of the Pixel Counts of the Urban Test Dataset

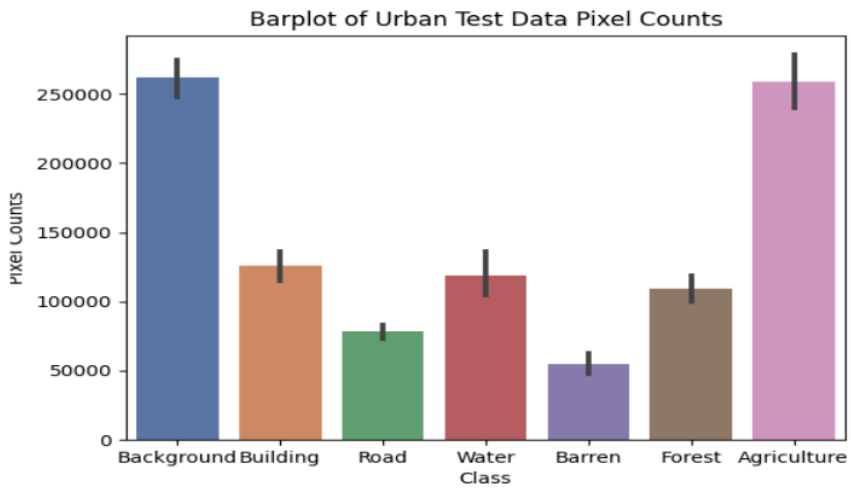

Figure 5.12: Bar plot of the Pixel Counts of the Urban Test Dataset

5.2 *Data Preprocessing*

As shown in figure **??**, the dataset is split into urban and rural area. Those two categories are further split into training, testing and validation dataset. The mask are only avaialble for the training and validation set. When the actual training is being done, we combined the training and validation set into one folder as shown in figure **??**. The model will be trained using both training sets.

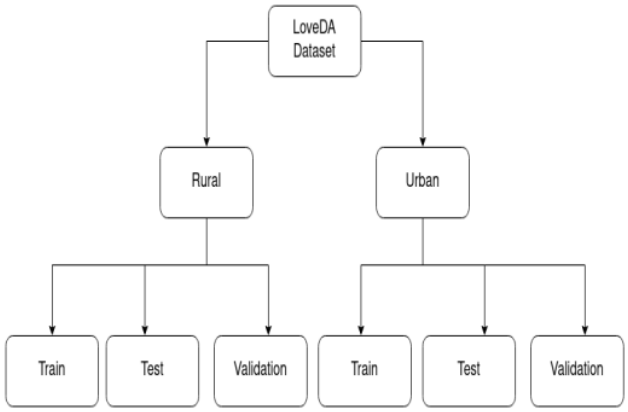

Figure 5.13: Diagram of Data Utilization

Before training starts, the training images and mask would undergo several transformation. First, they would be cropped into smaller images of 512x512 pixels as required by UNetFormer. Then, those images would be randomly flipped, horizontally and vertically before randomly increasing its brightness.

5.3 Model Design and Implementation

The initial model will be written using PyTorch, a deep learning library written in Python. Table **??** shows the list of important libraries used to build and test the model. This model would give us a baseline performance and gives us insight on how to proceed for this project.

43
The model is trained on a computer with Nvidia GTX 2080 GPU. The semantic segmentation model chosen is UNetFormer (?, ?) and it was discussed in detail in Chapter 2.

Library	*Version*
Pytorch	1.10
Torchvision	0.11.0
Cudatoolkit	11.3
Timm	-
Catalyst	20.09
Albumentations	1.1.0
Numpy	-
Opencv-python	-

Table 5.3: List of Important Libraries Used

masukkan transfer learning dengan pre-trained weights.

On the other hand, we get the baseline result by training the same a U-Net model with ResNet18 backbone using the same dataset. The code for the model is written in PyTorch and is publicly available at the author's Github page . The model can be summarised in Figure **??**.

Based on Figure **??**, the training image are augmented as described in the previous section. Then, both the training images and masks would be cropped to reduce its size. All the components inside the red line belongs to UNetFormer. Patch embedding would be applied to the cropped images before it is fed into the encoder that is made up of resNet18. The encoder is connected to the decoder through a series of skip connections. The decoder is made up of four Tiny-Swin Transformer. The output from the decoder will be used to calculate the losses and to update the weights. This process is repeated as many times as the desired epoch value. Lastly, after it finishes

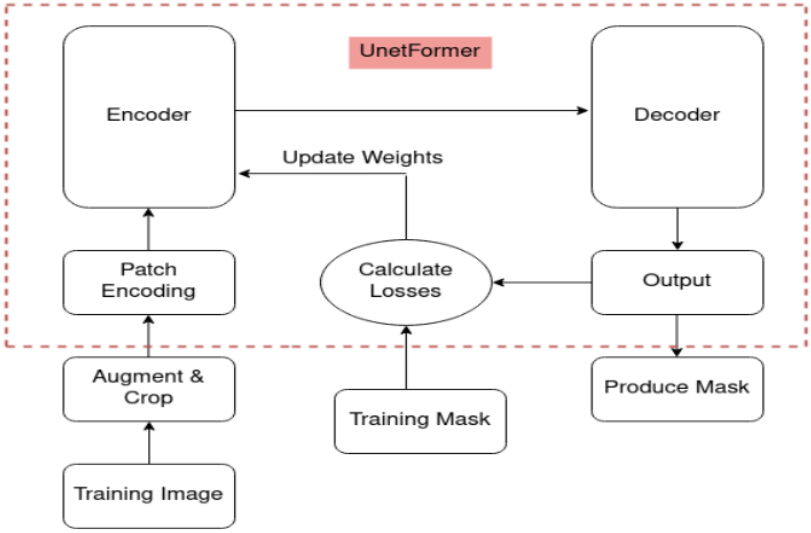

Figure 5.14: Proposed Initial Model

all epoch, the output will be a collection of cropped black and white images. The mask produced will have a pixel values of 0-7 and this makes it impossible to assess its quality through visual inspection. We converted the 0-7 values to RGB values to make it easier to assess. Table **??** shows the values of hyperparameters used in the model.

Hyperparameters	***Description***
Learning Rate	
Epoch	
Batch Size	
Optimizer	

Table 5.4: List and Value of Hyperparameters

5.4 Model Evaluation

The most common evaluation metric used by semantic segmentation is the mean Intersection over Union (mIoU). The mIoU is also known as the Jaccard Index or the

Jaccard similarity coefficient. The IoU for a single class is defined as:

$$IoU = \frac{TruePositives}{TruePositives + FalsePositives + FalseNegatives} \tag{5.1}$$

We get the mIoU by finding the mean of the IoU for all of the classes.

Accuracy is the ratio of correct predictions and can be calculated as:

$$Accuracy = \frac{TruePositives + TrueNegatives}{TruePositives + TrueNegatives + FalsePositives + FalseNegatives} \tag{5.2}$$

F1-score gives the accuracy of the model by combining precision and recall and it is widely preferred for imbalanced dataset. Precision, recall and F1-score can be calculated as:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{5.3}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{5.4}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{5.5}$$

We plan to use all five of the evaluation metrics to evaluate our model.

5.5 *Model Refinement*

This section would explain the methods that can be taken to refine the model. During FYP 1, we proposed two methods to refine the model. The first method is hyperparameter tuning and the second one is pruning the parameters.

5.5.1 *Hyperparameter Tuning*

Hyperparameters are the variables that determines how a network updates its weight. Changing the value of the hyperparameters could result in a vastly different level of performance for the network. Hyperparameter tuning is the process of finding the optimal values of hyperparameters. Table **??** shows the list and definition of the hyperparameters available to the model.

Hyperparameters	***Description***
Learning Rate	Learning rate determines how fast the network updates the learnt weight. Learning rate that is too small would result in a very long training time that may get stuck. While if the learning rate is too large may produced a sub-optimal learnt weights or an unstable learning process.
Epoch	The number of times the model will go through the training data set.
Batch Size	The number of samples to work through before updating the model's parameters.

Table 5.5: List and Definition of Hyperparameters

5.5.2 Pruning Vision Transformer

Pruning was discussed in Chapter 2.7 and it is a method to reduce the number of parameter while maintaining the same level of performance. Pruning will reduce training the training and inference time and compress the size of the model. We did not use any pruning technique in FYP 1 but this is a technique that we wish to investigate in depth in FYP 2.

CHAPTER 6

IMPLEMENTATION PLAN AND INITIAL RESULTS

6.1 Planning

An implementation strategy was created with milestones achieve at key project check-points in order to complete the project in a timely manner. In figures **??** and **??** , two Gantt charts were created to make it easier to keep track our progress in FYP 1 and FYP 2. Table 5.1 and Table 5.2, respectively, contain the detailed description of the milestones displayed in the respective Gantt charts.

TASK	OCT	NOV	DEC	JAN	FEB
Knowledge Gaining					
Literature Review					
Data Collection					
Data Pre-processing					
Model Searching					
Model Testing					
Model Design					
Model Implementation					
Preliminary Result Analysis					
Report Writing					
FYP 1 Presentation					

Table 6.1: Gantt Chart for FYP 1

TASK	MARCH	APRIL	MAY	JUNE
Knowledge Gaining				
Model Design				
Model Implementation				
Model Evaluation				
Model Refinement				
Experimental Result Analysis				
Report Writing				
FYP 2 Presentation				

Table 6.2: Gantt Chart for FYP 2

6.2 Initial Results

The summary of the result of our experiments is shown in Figure **??**. For our U-Net model that was used to provide the baseline performance achieved an mIoU of 33.6%. It took 4.5 hours to train the U-Net model and it has 2.4 million parameters.

Our UNetFormer model gave an mIoU of 72.5% which is a very significant increase compared to U-Net. However it took 14.2 hours to train the UNetFormer model and the model has 11.2 million parameters. The model size and training time is a huge concern and it would be our main objective in FYP 2 to reduce its size and training time.

Model	mIoU	Parameters	Training Time
U-Net	33.6	2.4 M	4.5 hours
UNetFormer	72.5	11.2 M	14.2 hours

Table 6.3: Baseline and Initial Result

Figure **??** is a sample image from the testing dataset and **??** displays the segmentation result from U-Net and UNetFormer. We observed that U-Net struggles to classify that

are within the boundary of pixels from another class. It also struggles to classify large objects that span across the image. On top of that, the edges between the classes are not very sharp. Compared its result to UNetFormer, the edges produced are very sharp and it has little problems of classifying small objects that are surrounded by other large objects.

Figure 6.1: An Image From The Test Set

Figure 6.2: (a) Segmentation Result From U-Net (b) Segmentation Result from UNet-Former

CHAPTER 7

CONCLUSION

This chapter contain the problems we encountered, our plans for FYP 2 and the conclusion of our project.

7.1 Problems Encountered

1. The model that we found used timm, a framework that provides pre-trained encoders and decoders for semantic segmentation models. We found that the documentation is not up to date and it was very difficult to set up our model.

2. The classes in LoveDa dataset is imbalanced which is an extremely common scenario in semantic segmentation task. None of the works we reviewed did any kind of resampling to balance it out.

7.2 Plans For FYP 2

1. Based on our literature review, the most common evaluation metrics when training a neural network are the amount of memory used (MB), the number of frames processed by the models in one second (FPS), the complexity of the

model (G) and the number of parameters. We believe that these parameters would be very useful in evaluating our model. We tried to get these parameters using PyTorch but were unsuccessful. These parameters will be included in our FYP 2.

2. We listed out five models in our literature review but we have only managed to train one model of them for FYP 1 as most of our effort went to learning and literature review. We wish to train the other models in FYP 2.

3. Our most important plan for FYP 2 is to prune UNetFormer to reduce its size and training time. We are going to conduct more research on how to prune a semantic segmentation model.

7.3 Conclusion

In FYP1, we studied the fundamentals of semantic segmentation, satellite images and cutting edge deep learning methods in depth and formulated useful and interesting objectives. Literature reviews were conducted to identify the works done in this field. On top of that, we spent months to present an in depth theoretical background necessary for this project. Furthermore, we crafted a research methodology with multiple stages to make sure that our efforts are not sidetracked and meet the project deadline. Other than that, we successfully trained two semantic segmentation of satellite images model . The initial results are quite promising but our training method require a more detailed analysis in FYP 2.

REFERENCES

Bahdanau, D., Cho, K., & Bengio, Y. (2014, 09). Neural machine translation by jointly learning to align and translate. *ArXiv*, *1409*.

Blake, A., Kohli, P., & Rother, C. (2011). *Markov random fields for vision and image processing*. Cambridge, Massachusetts: MIT Press.

Boguszewski, A., Batorski, D., Ziemba-Jankowska, N., Dziedzic, T., & Zambrzycka, A. (2021, June). Landcover.ai: Dataset for automatic mapping of buildings, woodlands, water and roads from aerial imagery. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (cvpr) workshops* (p. 1102-1110).

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. *CoRR*, *abs/2005.12872*. Retrieved from `https://arxiv.org/abs/2005.12872`

Chi, C., Wei, F., & Hu, H. (2020). Relationnet++: Bridging visual representations for object detection via transformer decoder. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (Vol. 33, pp. 13564–13574). Curran Associates, Inc. Retrieved from `https://proceedings.neurips.cc/paper/2020/file/9d684c589d67031a627ad33d59db65e5-Paper.pdf`

Demir, I., Koperski, K., Lindenbaum, D., Pang, G., Huang, J., Basu, S., ... Raskar, R. (2018). Deepglobe 2018: A challenge to parse the earth through satellite images. *CoRR*, *abs/1805.06561*. Retrieved from `http://arxiv.org/abs/1805.06561`

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: pre-training of deep bidirectional

transformers for language understanding. *CoRR*, *abs/1810.04805*. Retrieved from http://arxiv.org/abs/1810.04805

Ding, L., Tang, H., & Bruzzone, L. (2021). Lanet: Local attention embedding to improve the semantic segmentation of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *59*(1), 426-435. doi: 10.1109/TGRS.2020.2994150

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, *abs/2010.11929*. Retrieved from https://arxiv.org/abs/2010.11929

Fan, X., Yan, C., Fan, J., & Wang, N. (2022). Improved u-net remote sensing classification algorithm fusing attention and multiscale features. *Remote Sensing*, *14*(15). Retrieved from https://www.mdpi.com/2072-4292/14/15/3591 doi: 10.3390/rs14153591

Fang, G., Ma, X., Song, M., Mi, M. B., & Wang, X. (2023). *Depgraph: Towards any structural pruning*. arXiv. Retrieved from https://arxiv.org/abs/2301.12900 doi: 10.48550/ARXIV.2301.12900

Gu, J., Hu, H., Wang, L., Wei, Y., & Dai, J. (2018). Learning region features for object detection. *CoRR*, *abs/1803.07066*. Retrieved from http://arxiv.org/abs/1803.07066

Gu, X., Li, S., Ren, S., Zheng, H., Fan, C., & Xu, H. (2022). Adaptive enhanced swin transformer with u-net for remote sensing image segmentation. *Computers and Electrical Engineering*, *102*, 108223. Retrieved from https://www.sciencedirect.com/science/article/pii/S004579062200461X doi: https://doi.org/10.1016/j.compeleceng.2022.108223

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., ... Liu, Q. (2019). Tinybert: Distilling BERT for natural language understanding. *CoRR*, *abs/1909.10351*. Retrieved from http://arxiv.org/

abs/1909.10351

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, & K. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 25). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Lagunas, F., Charlaix, E., Sanh, V., & Rush, A. M. (2021). Block pruning for faster transformers. *CoRR*, *abs/2109.04838*. Retrieved from https://arxiv.org/abs/2109.04838

Leach, N. R., Popien, P., Goodman, M. C., & Tellman, B. (n.d.). Leveraging convolutional neural networks for semantic segmentation of global floods with planetscope imagery. *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium, Geoscience and Remote Sensing Symposium, IGARSS 2022 - 2022 IEEE International*, 314 - 317. Retrieved from https://go.openathens.net/redirector/mmu.edu.my?url={UrlEncode(https://search.ebscohost.com/login.aspx?direct=true&AuthType=shib&db=edseee&AN=edseee.9884272&site=eds-live)},Year={2022},

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*, 541-551.

Li, R., Zheng, S., & Duan, C. (2021). Feature pyramid network with multi-head attention for semantic segmentation of fine-resolution remotely sensed images. *CoRR*, *abs/2102.07997*. Retrieved from https://arxiv.org/abs/2102.07997

Li, R., Zheng, S., Zhang, C., Duan, C., Su, J., Wang, L., & Atkinson, P. M. (2022). Multiattention network for semantic segmentation of fine-resolution remote sensing images. *IEEE Transactions*

on Geoscience and Remote Sensing, *60*, 1-13. doi: 10.1109/TGRS.2021.3093977

Li, R., Zheng, S., Zhang, C., Duan, C., Wang, L., & Atkinson, P. M. (2021). Abcnet: Attentive bilateral contextual network for efficient semantic segmentation of fine-resolution remotely sensed imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, *181*, 84-98. Retrieved from https://www.sciencedirect.com/science/article/pii/S0924271621002379 doi: https://doi.org/10.1016/j.isprsjprs.2021.09.005

Liu, W., Zhang, C., Lin, G., & Liu, F. (2022, sep). Crcnet: Few-shot segmentation with cross-reference and region–global conditional networks. *Int. J. Comput. Vision*, *130*(12), 3140–3157. Retrieved from https://doi.org/10.1007/s11263-022-01677-7 doi: 10.1007/s11263-022-01677-7

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., ... Guo, B. (2021). Swin transformer V2: scaling up capacity and resolution. *CoRR*, *abs/2111.09883*. Retrieved from https://arxiv.org/abs/2111.09883

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, *abs/2103.14030*. Retrieved from https://arxiv.org/abs/2103.14030

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *2015 ieee conference on computer vision and pattern recognition (cvpr)* (p. 3431-3440). doi: 10.1109/CVPR.2015.7298965

Lowe, D. G. (2004, November). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, *60*(2), 91–110. Retrieved from http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94 doi: 10.1023/B:VISI.0000029664.99615.94

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M. C. H., Heinrich, M. P., Misawa, K., ... Rueckert,

D. (2018). Attention u-net: Learning where to look for the pancreas. *CoRR*, *abs/1804.03999*. Retrieved from http://arxiv.org/abs/1804.03999

Pandey, A., Kumar, D., & Chakraborty, D. B. (2021). Soil type classification from high resolution satellite images with deep cnn. In *2021 ieee international geoscience and remote sensing symposium igarss* (p. 4087-4090). doi: 10.1109/IGARSS47720.2021.9554290

Papoutsis, I., Bountos, N., Zavras, A., Michail, D., & Tryfonopoulos, C. (2021). Efficient deep learning models for land cover image classification. *CoRR*, *abs/2111.09451*. Retrieved from https://arxiv.org/abs/2111.09451

Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A., & Rastegari, M. (2019). What's hidden in a randomly weighted neural network? *CoRR*, *abs/1911.13299*. Retrieved from http://arxiv.org/abs/1911.13299

Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Pinto, A. S., ... Houlsby, N. (2021). Scaling vision with sparse mixture of experts. *CoRR*, *abs/2106.05974*. Retrieved from https://arxiv.org/abs/2106.05974

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR*, *abs/1505.04597*. Retrieved from http://arxiv.org/abs/1505.04597

Rottensteiner, F., Sohn, G., Jung, J., Gerke, M., Baillard, C., Benitez, S., & Breitkopf, U. (2012). The isprs benchmark on urban object classification and 3d building reconstruction. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *I-3*, 293–298. Retrieved from https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/I-3/293/2012/ doi: 10.5194/isprsannals-I-3-293-2012

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of BERT: smaller,

faster, cheaper and lighter. *CoRR*, *abs/1910.01108*. Retrieved from http://arxiv.org/abs/1910.01108

Schmitt, M., Prexl, J., Ebel, P., Liebel, L., & Zhu, X. X. (2020). Weakly supervised semantic segmentation of satellite images for land cover mapping - challenges and opportunities. *CoRR*, *abs/2002.08254*. Retrieved from https://arxiv.org/abs/2002.08254

Schroff, F., Criminisi, A., & Zisserman, A. (2008). Object class segmentation using random forests. In *Bmvc*.

Sun, Y., Bi, F., Gao, Y., Chen, L., & Feng, S. (2022). A multi-attention unet for semantic segmentation in remote sensing images. *Symmetry*, *14*(5). Retrieved from https://www.mdpi.com/2073-8994/14/5/906

Talal, M., Panthakkan, A., Mukhtar, H., Mansoor, W., Almansoori, S., & Ahmad, H. A. (n.d.). Detection of water-bodies using semantic segmentation. *2018 International Conference on Signal Processing and Information Security (ICSPIS), Signal Processing and Information Security (ICSPIS), 2018 International Conference on*, 1 - 4. Retrieved from https://go.openathens.net/redirector/mmu.edu.my?url={UrlEncode(https://search.ebscohost.com/login.aspx?direct=true&AuthType=shib&db=edseee&AN=edseee.8642743&site=eds-live)},Year={2018},

Tao, M., Ding, Z., & Cao, Y. (2021). Attention u-net for road extraction in remote sensing images. In X. Meng, X. Xie, Y. Yue, & Z. Ding (Eds.), *Spatial data and intelligence* (pp. 153–164). Cham: Springer International Publishing.

Teichmann, M. T. T., & Cipolla, R. (2018). Convolutional crfs for semantic segmentation. *CoRR*, *abs/1805.04777*. Retrieved from http://arxiv.org/abs/1805.04777

Thoma, M. (2016). A survey of semantic segmentation. *CoRR*, *abs/1602.06541*. Retrieved from http://arxiv.org/abs/1602.06541

Tong, X.-Y., Xia, G.-S., Lu, Q., Shen, H., Li, S., You, S., & Zhang, L. (2020). Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sensing of Environment*, *237*, 111322.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2020). Training data-efficient image transformers & distillation through attention. *CoRR*, *abs/2012.12877*. Retrieved from https://arxiv.org/abs/2012.12877

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *CoRR*, *abs/1706.03762*. Retrieved from http://arxiv.org/abs/1706.03762

Wang, J., Zheng, Z., Ma, A., Lu, X., & Zhong, Y. (2021, 10). Loveda: A remote sensing land-cover dataset for domain adaptive semantic segmentation..

Wang, L., Fang, S., Zhang, C., Li, R., & Duan, C. (2021). Efficient hybrid transformer: Learning global-local context for urban sence segmentation. *CoRR*, *abs/2109.08937*. Retrieved from https://arxiv.org/abs/2109.08937

Wang, L., Li, R., Duan, C., Zhang, C., Meng, X., & Fang, S. (2022). A novel transformer based semantic segmentation scheme for fine-resolution remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, *19*, 1-5. doi: 10.1109/LGRS.2022.3143368

Wang, L., Li, R., Wang, D., Duan, C., Wang, T., & Meng, X. (2021a). *Transformer meets convolution: A bilateral awareness network for semantic segmentation of very fine resolution urban scene images.* arXiv. Retrieved from https://arxiv.org/abs/2106.12413 doi: 10.48550/

ARXIV.2106.12413

Wang, L., Li, R., Wang, D., Duan, C., Wang, T., & Meng, X. (2021b). Transformer meets convolution: A bilateral awareness net-work for semantic segmentation of very fine resolution ur-ban scene images. *CoRR*, *abs/2106.12413*. Retrieved from https://arxiv.org/abs/2106.12413

Xu, W., Xu, Y., Chang, T. A., & Tu, Z. (2021). Co-scale conv-attentional image transformers. *CoRR*, *abs/2104.06399*. Retrieved from https://arxiv.org/abs/2104.06399

Yu, H., Yang, Z., Tan, L., Wang, Y., Sun, W., Sun, M., & Tang, Y. (2018). Methods and datasets on semantic segmentation: A review. *Neurocomputing*, *304*, 82-103. Retrieved from https://www.sciencedirect.com/science/article/pii/S0925231218304077 doi: https://doi.org/10.1016/j.neucom.2018.03.037

Yuan, Q., Shen, H., Li, T., Li, Z., Li, S., Jiang, Y., ... pei Zhang, L. (2020). Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment*, *241*, 111716.

Zhang, Q., & bin Yang, Y. (2021). Rest: An efficient transformer for visual recognition. In A. Beygelzimer, Y. Dauphin, P. Liang, & J. W. Vaughan (Eds.), *Advances in neural information processing systems*. Retrieved from https://openreview.net/forum?id=6Ab68Ip4Mu

Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2016). Pyramid scene parsing network. *CoRR*, *abs/1612.01105*. Retrieved from http://arxiv.org/abs/1612.01105

Zhu, M., Han, K., Tang, Y., & Wang, Y. (2021). Visual transformer pruning. *CoRR*, *abs/2104.08500*. Retrieved from https://arxiv.org/abs/2104.08500

PUBLICATION LIST

APPENDIX A: FYP 1 MEETING LOGS

FACULTY OF COMPUTING & INFORMATICS

TPT3101 Final Year Project (FYP1) Meeting Log
Trimester 1, 2022/23 (Trimester ID:2210)

Meeting Date: 28/9/2022	**Meeting No.:** 1
Meeting Mode: Physical	
Project ID: 2078	**Project Type:** Research-based
Project Title : Satellite Image Segmentation using Vision Transformer	
Student ID : 1201302740	**Student Name:** Muhammad Haziq Faiz Bin Mohd Ripin
Student Programme and Specialisation: B.C.S (Hons) Data Science	
Supervisor Name: Dr. Loh Yuen Peng	**Co-Supervisor Name:** (if applicable)
Collaborating Company: (if applicable)	**Company Supervisor Name:** (if applicable)

1. WORK DONE

[Please write the details of the work done, after the last meeting]

Tasks: Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion
(Please strike out the tasks, which are not applicable for the work done, after the last meeting)

Details (in point form):

- Read the LoveDA paper.
- Read the Self-supervised Vision Transformers for Land-cover Segmentation and Classification paper.
- Briefed about ensemble model and Feature Fusion during the meeting.
- Briefed about Instance Segmentation and Semantic Segmentation.
- Installed PyTorch with ROCm support that works with AMD GPU. Found out it has a lot of bugs so decided to stick with google colab.
- Watched a few lecture videos on Transformer and Neural Network.
- Research the differences between spatial and spectral images.
- Learned the basics of linear algebra and statistics so that I can get a clearer understanding of the paper.
- Discussed the Introduction and Problem Statement parts of the project.

2. WORK TO BE DONE
[Please write the details of the work to be done, before the next meeting]

Tasks: Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept/ Draft Report Completion
(Please strike out the tasks, which are not applicable for the work to be done, before the next meeting)

Details (in point form):

- Write the Introduction and Problem Statement.
- Find and read more research papers.
- Try to train a model written in PyTorch. Preferably a Transformer based one.
- Research Swin Transformer. try to train a model with it if possible.
- Continue studying about deep learning and improving my math skills.
- Further reading about Ensemble models.

3. PROBLEMS ENCOUNTERED AND SOLUTIONS

[Please write the details of the problems encountered, after the last meeting and provide the solutions / plan for the solutions]

- Didn't have the baseline knowledge about Deep Learning.
  Solution: Refer to lectures by Hung-yi Lee and Yann LeCun. The latter one actually provided notebooks so I can understand it better.
- Didn’t have the prerequisite math knowledge to understand the research papers.
  Solution: Borrowed two books from the library. Gilbert Strang’s Linear Algebra and Bishop’s Pattern Recognition. These are the ones recommended from my research on the Internet. Currently at Chapter 3 of Linear Algebra book. Haven’t started on the Pattern Recognition one.

4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)

Good that you have looked further into other sources to help in understanding the concepts, coding, and related research papers.
Do be mindful of the time taken to study the concepts as it doesn't take too much away from the main research part.

……
Supervisor's Signature

……...
Student's Signature

……..
Co-Supervisor's Signature
(if applicable)

…….
Company Supervisor's Signature
(if applicable)

IMPORTANT NOTES TO STUDENTS:

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.

2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.
   Minimum requirement is SIX Meeting Logs (Period: Week 3 to Week 13). Students can have fortnightly meetings with the supervisor.

3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.

   This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.

4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.

FACULTY OF COMPUTING & INFORMATICS

TPT3101 Final Year Project (FYP1) Meeting Log
Trimester 1, 2022/23 (Trimester ID:2210)

Meeting Date: 28/9/2022	**Meeting No.:** 2
Meeting Mode: Physical	
Project ID: 2078	**Project Type:** Research-based
Project Title : Satellite Image Segmentation using Vision Transformer	
Student ID : 1201302740	**Student Name:** Muhammad Haziq Faiz Bin Mohd Ripin
Student Programme and Specialisation: B.C.S (Hons) Data Science	
Supervisor Name: Dr. Loh Yuen Peng	**Co-Supervisor Name:** (if applicable)
Collaborating Company: (if applicable)	**Company Supervisor Name:** (if applicable)

1. WORK DONE
[Please write the details of the work done, after the last meeting]

Tasks: Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion
(Please strike out the tasks, which are not applicable for the work done, after the last meeting)

Details (in point form):

- Learn more about attention mechanisms.
- Studied Attention is All You Need paper.
- Read Segformer and Swin Transformer papers.
- Run a simple transformer model on Colab, although it has anything to do with semantic segmentation.
- Tried to train the Swin Transformer on Colab but failed.
- Wrote the introduction and the structure of the report.
- Practiced Pytorch.

2. WORK TO BE DONE

[Please write the details of the work to be done, before the next meeting]

Tasks: Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept/ Draft Report Completion

(Please strike out the tasks, which are not applicable for the work to be done, before the next meeting)

Details (in point form):

- Identify 3 objectives.
- List down available datasets including the types, sizes etc.
- Improve understanding of transformers.
- Improve Pytorch skills.

3. PROBLEMS ENCOUNTERED AND SOLUTIONS

[Please write the details of the problems encountered, after the last meeting and provide the solutions / plan for the solutions]

- Can't run Swin Transformer model. Need to improve Python/Pytorch.
- Keep on studying about transformer.

4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)

- Try to match the versions of the libraries used by the code you downloaded as that can be a common issue where the code doesn't work for training.
- Do an outline of the literature review and try to have a first draft before next meeting.

……
Supervisor's Signature

……...
Student's Signature

……..
Co-Supervisor's Signature
(if applicable)

……
Company Supervisor's Signature
(if applicable)

IMPORTANT NOTES TO STUDENTS:

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.

2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.
   Minimum requirement is SIX Meeting Logs (Period: Week 3 to Week 13). Students can have fortnightly meetings with the supervisor.

3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.

   This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.

4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.

FACULTY OF COMPUTING & INFORMATICS

TPT3101 Final Year Project (FYP1) Meeting Log
Trimester 1, 2022/23 (Trimester ID:2210)

Meeting Date: 28/9/2022	**Meeting No.:** 3
Meeting Mode: Physical	
Project ID: 2078	**Project Type:** Research-based
Project Title : Satellite Image Segmentation using Vision Transformer	
Student ID : 1201302740	**Student Name:** Muhammad Haziq Faiz Bin Mohd Ripin
Student Programme and Specialisation: B.C.S (Hons) Data Science	
Supervisor Name: Dr. Loh Yuen Peng	**Co-Supervisor Name:** (if applicable)
Collaborating Company: (if applicable)	**Company Supervisor Name:** (if applicable)

1. WORK DONE

[Please write the details of the work done, after the last meeting]

Tasks: Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion
(Please strike out the tasks, which are not applicable for the work done, after the last meeting)

Details (in point form):

- Finished Chapter 1 and Chapter 2 of the report
- Chapter 3 is about 80% completed.
- Trained a U-Net model with Vaihingen dataset. I used segmentation-models-pytorch to achieve this. Achieved mIOU of 7.5 which is really bad compared to other literatures. Need to identify the mistake in the model.
- Training a SegFormer model using HuggingFace library.
- Training a Swin Transformer V1 model using Timm library.
- Identified class imbalance in Vaihingen, Potsdam, LoveDa, DeepGlobe and GID-15 dataset.

2. WORK TO BE DONE

[Please write the details of the work to be done, before the next meeting]

Tasks: Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept/ Draft Report Completion

(Please strike out the tasks, which are not applicable for the work to be done, before the next meeting)

Details (in point form):

- Trains a SegFormer model using HuggingFace library.
- Trains a Swin Transformer V1 model using Timm library.
- These 2 will serve as the baseline for my project.

3. PROBLEMS ENCOUNTERED AND SOLUTIONS

[Please write the details of the problems encountered, after the last meeting and provide the solutions / plan for the solutions]

- Can't run Swin Transformer model. Need to improve Python/Pytorch.
- Keep on studying about transformer.

4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)

Very good progress. Keep it up.

……
Supervisor's Signature

……...
Student's Signature

……..
Co-Supervisor's Signature
(if applicable)

……
Company Supervisor's Signature
(if applicable)

IMPORTANT NOTES TO STUDENTS:

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.

2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.
   Minimum requirement is SIX Meeting Logs (Period: Week 3 to Week 13). Students can have fortnightly meetings with the supervisor.

3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.

   This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.

4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.

FACULTY OF COMPUTING & INFORMATICS

TPT3101 Final Year Project (FYP1) Meeting Log
Trimester 1, 2022/23 (Trimester ID:2210)

Meeting Date: 22/12/2022	**Meeting No.:** 4
Meeting Mode: Physical	
Project ID: 2078	**Project Type:** Research-based
Project Title : Satellite Image Segmentation using Vision Transformer	
Student ID : 1201302740	**Student Name:** Muhammad Haziq Faiz Bin Mohd Ripin
Student Programme and Specialisation: B.C.S (Hons) Data Science	
Supervisor Name: Dr. Loh Yuen Peng	**Co-Supervisor Name:** (if applicable)
Collaborating Company: (if applicable)	**Company Supervisor Name:** (if applicable)

1. WORK DONE
[Please write the details of the work done, after the last meeting]

Tasks: Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion
(Please strike out the tasks, which are not applicable for the work done, after the last meeting)

Details (in point form):

- Finished the Gantt Chart and flowchart for Chapter 4.
- Trained SegFormer on Vaihingen Dataset using HuggingFace api
- Discovered GeoSeg https://github.com/WangLibo1995/GeoSeg. GeoSeg is an open-source semantic segmentation toolbox based on PyTorch, pytorch lightning and timm. This makes it easier to build and train my model because I don't need to learn frameworks like timm and mmsegmentation.
- Clean and split LoveDA dataset.
- Narrow down the architectures to 3 : UNetFormer, DC-Swin and BANet.

2. WORK TO BE DONE

[Please write the details of the work to be done, before the next meeting]

Tasks: Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept/ Draft Report Completion

(Please strike out the tasks, which are not applicable for the work to be done, before the next meeting)

Details (in point form):

- Run a baseline model of UNetFormer, DC-Swin and BANet using GeoSeg with pre-trained weights on the LoveDA dataset.
- Get the performance on the baseline model.
- Start working on chapter 4 and 5 of the report.
- Figure out how to run the models on google colab or paperspace as my computer is inadequate for the training.

3. PROBLEMS ENCOUNTERED AND SOLUTIONS

[Please write the details of the problems encountered, after the last meeting and provide the solutions / plan for the solutions]

- Had a hard time understanding mmsegmentation framework which is required for most semantic segmentation models. Discovered GeoSeg which abstracted all the hard parts.

4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)

Take note that the evaluation needs to be done using some standard measurements to compare with the groundtruth and get quantitative values, not just observation.

……
Supervisor's Signature

……...
Student's Signature

……..
Co-Supervisor's Signature
(if applicable)

……
Company Supervisor's Signature
(if applicable)

IMPORTANT NOTES TO STUDENTS:

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.

2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.
   Minimum requirement is SIX Meeting Logs (Period: Week 3 to Week 13). Students can have fortnightly meetings with the supervisor.

3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.

   This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.

4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.

FACULTY OF COMPUTING & INFORMATICS

TPT3101 Final Year Project (FYP1) Meeting Log
Trimester 1, 2022/23 (Trimester ID:2210)

Meeting Date: 6/1/2022	**Meeting No.:** 5
Meeting Mode: Physical	
Project ID: 2078	**Project Type:** Research-based
Project Title : Satellite Image Segmentation using Vision Transformer	
Student ID : 1201302740	**Student Name:** Muhammad Haziq Faiz Bin Mohd Ripin
Student Programme and Specialisation: B.C.S (Hons) Data Science	
Supervisor Name: Dr. Loh Yuen Peng	**Co-Supervisor Name:** (if applicable)
Collaborating Company: (if applicable)	**Company Supervisor Name:** (if applicable)

1. WORK DONE
[Please write the details of the work done, after the last meeting]

Tasks: Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept / Draft Report Completion
(Please strike out the tasks, which are not applicable for the work done, after the last meeting)

Details (in point form):

- Did corrections on my chapter 1,2 and 3.
- Ran a baseline on DC-Swin on LoveDA dataset with pretrained weights.
- Start writing chapter 4 and 5

2. WORK TO BE DONE

[Please write the details of the work to be done, before the next meeting]

Tasks: Problem Formulation and Project Planning / Background Study or Literature Review / Requirement Analysis or Theoretical Framework / Design or Research Methodology / Prototype Development or Proof of Concept/ Draft Report Completion

(Please strike out the tasks, which are not applicable for the work to be done, before the next meeting)

Details (in point form):

- Run a baseline model of UNetFormer and DC-Swin.
- Get the performance on the baseline model.
- Figure out how to run the models on google colab or paperspace as my computer is inadequate for the training.

3. PROBLEMS ENCOUNTERED AND SOLUTIONS

[Please write the details of the problems encountered, after the last meeting and provide the solutions / plan for the solutions]

- Had a problem with uploading datasets to Google colab.
- Had a problem with installing GeoSeg on Google colab.
- I am still figuring out the solution to the problems above.

4. COMMENTS (Supervisor / Co-Supervisor / Company Supervisor)

Problems with installation in Google Colab might be due to versioning issue or old package that the library was built on.
If this is not a problem using in local machine (only due to processing power limitation), then try to wrap your code and a research PC can be allocated for you to run the main training on. This may be on sharing basis.

……
Supervisor's Signature

……...
Student's Signature

……..
Co-Supervisor's Signature
(if applicable)

……
Company Supervisor's Signature
(if applicable)

IMPORTANT NOTES TO STUDENTS:

1. Items 1 – 3 are to be completed by the students prior to the meeting. Item 4 is to be completed by the supervisor / co-supervisor / company supervisor.

2. Student has to upload the soft copies of the meeting logs in Google Classroom and also attach them along with interim (FYP1) report.
   Minimum requirement is SIX Meeting Logs (Period: Week 3 to Week 13). Students can have fortnightly meetings with the supervisor.

3. Log sheets provide the basis for evaluating the General Effort (Project Management, Attitude, and Technical Competency) of the student, by the supervisor and also for checking the attendance requirement of the student, by the FYP Committee.

   This also provide the student with feedback from the supervisor / co-supervisor / company supervisor on the tasks done and provide the plan for the upcoming tasks. This can provide the motivation for the student to give consistent and efficient effort throughout the period of FYP.

4. Student who fails to meet the minimum requirement (six nos.) of log sheets will not be allowed to submit FYP report.

APPENDIX B: TURNITIN REPORT

PLACEHOLDER APPENDIX

Thesis

ORIGINALITY REPORT

30%	23%	21%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	arxiv.org Internet Source	5%
2	www.arxiv-vanity.com Internet Source	2%
3	Libo Wang, Rui Li, Ce Zhang, Shenghui Fang, Chenxi Duan, Xiaoliang Meng, Peter M. Atkinson. "UNetFormer: A UNet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery", ISPRS Journal of Photogrammetry and Remote Sensing, 2022 Publication	1%
4	www.researchgate.net Internet Source	1%
5	peterbloem.nl Internet Source	1%
6	www.mdpi.com Internet Source	1%
7	torchgeo.readthedocs.io Internet Source	1%

8	Xingjian Gu, Sizhe Li, Shougang Ren, Hengbiao Zheng, Chengcheng Fan, Huanliang Xu. "Adaptive enhanced swin transformer with U-net for remote sensing image segmentation", Computers and Electrical Engineering, 2022 Publication	1 %
9	Libo Wang, Rui Li, Chenxi Duan, Ce Zhang, Xiaoliang Meng, Shenghui Fang. "A Novel Transformer Based Semantic Segmentation Scheme for Fine-Resolution Remote Sensing Images", IEEE Geoscience and Remote Sensing Letters, 2022 Publication	1 %
10	brilliant.org Internet Source	1 %
11	deepai.org Internet Source	1 %
12	Hongshan Yu, Zhengeng Yang, Lei Tan, Yaonan Wang, Wei Sun, Mingui Sun, Yandong Tang. "Methods and Datasets on Semantic Segmentation: A review", Neurocomputing, 2018 Publication	1 %
13	Submitted to Multimedia University Student Paper	1 %
14	ia801805.us.archive.org Internet Source	1 %

15	Lei Ding, Hao Tang, Lorenzo Bruzzone. "LANet: Local Attention Embedding to Improve the Semantic Segmentation of Remote Sensing Images", IEEE Transactions on Geoscience and Remote Sensing, 2021 Publication	<1 %
16	web.archive.org Internet Source	<1 %
17	preview.d2l.ai Internet Source	<1 %
18	Submitted to University of Durham Student Paper	<1 %
19	munin.uit.no Internet Source	<1 %
20	Liang Gao, Hui Liu, Minhang Yang, Long Chen, Yaling Wan, Yurong Qian, Zhengqing Xiao. "STransFuse: Fusing Swin Transformer and Convolutional Neural Network for Remote Sensing Image Semantic Segmentation", Institute of Electrical and Electronics Engineers (IEEE), 2021 Publication	<1 %
21	docplayer.net Internet Source	<1 %
22	www.cs.hhu.de Internet Source	<1 %

23	Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, Baining Guo. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows", 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021 Publication	<1 %
24	dokumen.pub Internet Source	<1 %
25	www.macmillan.co.za Internet Source	<1 %
26	eprints.lancs.ac.uk Internet Source	<1 %
27	Submitted to Letterkenny Institute of Technology Student Paper	<1 %
28	Submitted to University of North Georgia Student Paper	<1 %
29	Submitted to University of Warwick Student Paper	<1 %
30	mafiadoc.com Internet Source	<1 %
31	Libo Wang, Rui Li, Dongzhi Wang, Chenxi Duan, Teng Wang, Xiaoliang Meng. "Transformer Meets Convolution: A Bilateral Awareness Network for Semantic	<1 %

Segmentation of Very Fine Resolution Urban Scene Images", Remote Sensing, 2021
Publication

32	Submitted to Universiti Teknologi Malaysia Student Paper	<1 %
33	d-nb.info Internet Source	<1 %
34	vdoc.pub Internet Source	<1 %
35	Submitted to Curtin University of Technology Student Paper	<1 %
36	Submitted to University of Lincoln Student Paper	<1 %
37	openreview.net Internet Source	<1 %
38	wrap.warwick.ac.uk Internet Source	<1 %
39	Zhuoxuan Li, Junli Yang, Bin Wang, Yaqi Li, Ting Pan. "Maskformer with Improved Encoder-Decoder Module for Semantic Segmentation of Fine-Resolution Remote Sensing Images", 2022 IEEE International Conference on Image Processing (ICIP), 2022 Publication	<1 %
40	Xin-Yi Tong, Gui-Song Xia, Qikai Lu, Huanfeng Shen, Shengyang Li, Shucheng You, Liangpei	<1 %

Zhang. "Land-cover classification with high-resolution remote sensing images using transferable deep models", Remote Sensing of Environment, 2020
Publication

41	www.ncbi.nlm.nih.gov Internet Source	<1 %
42	www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net Internet Source	<1 %
43	"Machine Learning in Medical Imaging", Springer Science and Business Media LLC, 2019 Publication	<1 %
44	Submitted to Aalto Yliopisto Student Paper	<1 %
45	morethingsjapanese.com Internet Source	<1 %
46	Uday Kamath, John Liu, James Whitaker. "Deep Learning for NLP and Speech Recognition", Springer Science and Business Media LLC, 2019 Publication	<1 %
47	github.com Internet Source	<1 %
48	mdpi-res.com Internet Source	<1 %

49	Adrian Boguszewski, Dominik Batorski, Natalia Ziemba-Jankowska, Tomasz Dziedzic, Anna Zambrzycka. "LandCover.ai: Dataset for Automatic Mapping of Buildings, Woodlands, Water and Roads from Aerial Imagery", 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2021 Publication	<1 %
50	Rui Li, Shunyi Zheng, Ce Zhang, Chenxi Duan, Jianlin Su, Libo Wang, Peter M. Atkinson. "Multiattention Network for Semantic Segmentation of Fine-Resolution Remote Sensing Images", IEEE Transactions on Geoscience and Remote Sensing, 2021 Publication	<1 %
51	Submitted to University of Edinburgh Student Paper	<1 %
52	dspace.univ-msila.dz:8080 Internet Source	<1 %
53	Vahid Zamani, Hosein Taghaddos, Yaghob Gholipour, Hamidreza Pourreza. "Deep semantic segmentation for visual scene understanding of soil types", Automation in Construction, 2022 Publication	<1 %
54	repositorio.ufmg.br	

Internet Source

<1 %

55 Michele Volpi, Devis Tuia. "Deep multi-task learning for a geographically-regularized semantic segmentation of aerial images", ISPRS Journal of Photogrammetry and Remote Sensing, 2018
Publication

<1 %

56 Submitted to Associatie K.U.Leuven
Student Paper

<1 %

57 Submitted to Monash University
Student Paper

<1 %

58 Submitted to University of Birmingham
Student Paper

<1 %

59 Submitted to University of Glasgow
Student Paper

<1 %

60 Submitted to University of Wollongong
Student Paper

<1 %

61 "Computer Vision – ACCV 2020", Springer Science and Business Media LLC, 2021
Publication

<1 %

62 "Document Analysis and Recognition – ICDAR 2021", Springer Science and Business Media LLC, 2021
Publication

<1 %

63 Ahmed Shawky Elsharkawy, Mohamed Elhabiby, Naser El-Sheimy. "Second Generation Curvelet Transform for Building Extraction from High Resolution Satellite Imagery", GEOMATICA, 2011
Publication
<1%

64 Anderson, Neal T., Giovanni B. Marchisio, and Paul E. Lewis. "WorldView-2 and the evolution of the DigitalGlobe remote sensing satellite constellation: introductory paper for the special session on WorldView-2", Algorithms and Technologies for Multispectral Hyperspectral and Ultraspectral Imagery XVIII, 2012.
Publication
<1%

65 Bhargav J. Bhatkalkar, Dheeraj R. Reddy, Srikanth Prabhu, Sulatha V. Bhandary. "Improving the Performance of Convolutional Neural Network for the Segmentation of Optic Disc in Fundus Images Using Attention Gates and Conditional Random Fields", IEEE Access, 2020
Publication
<1%

66 Qinghui Liu, Michael Kampffmeyer, Robert Jenssen, Arnt-Børre Salberg. "Multi-modal land cover mapping of remote sensing images using pyramid attention and gated fusion
<1%

networks", International Journal of Remote Sensing, 2022
Publication

67	Zhenglin Xian, Jiaying Wang, Junli Yang, Bin Wang, Zideng Feng, Yifei Huang. "Dual-Path Geometry-Aware Network for Semantic Segmentation of High-Resolution Aerial Images", 2022 26th International Conference on Pattern Recognition (ICPR), 2022 Publication	<1 %
68	"Machine Learning in Medical Imaging", Springer Science and Business Media LLC, 2021 Publication	<1 %
69	"Medical Image Computing and Computer Assisted Intervention – MICCAI 2021", Springer Science and Business Media LLC, 2021 Publication	<1 %
70	Submitted to Nanyang Technological University Student Paper	<1 %
71	Teerapong Panboonyuen, Kulsawasd Jitkajornwanich, Siam Lawawirojwong, Panu Srestasathiern, Peerapon Vateekul. "Transformer-Based Decoder Designs for Semantic Segmentation on Remotely Sensed Images", Remote Sensing, 2021	<1 %

Publication

72	Submitted to The University of Manchester Student Paper	<1 %
73	commons.erau.edu Internet Source	<1 %
74	d2k0ddhflgrk1i.cloudfront.net Internet Source	<1 %
75	www.gsc-europa.eu Internet Source	<1 %
76	"Pattern Recognition", Springer Science and Business Media LLC, 2019 Publication	<1 %
77	Submitted to Technische Universität Clausthal Student Paper	<1 %
78	Yongjie Guo, Feng Wang, Yuming Xiang, Hongjian You. "DGFNet: Dual Gate Fusion Network for Land Cover Classification in Very High-Resolution Images", Remote Sensing, 2021 Publication	<1 %
79	dspace.sunyconnect.suny.edu Internet Source	<1 %
80	Bharati Basu. "Efficiency wages, unemployment and international factor movements", The Journal of International Trade & Economic Development, 1998	<1 %

Publication

81	Koichiro Yasaka, Hiroyuki Akai, Akira Kunimatsu, Shigeru Kiryu, Osamu Abe. "Deep learning with convolutional neural network in radiology", Japanese Journal of Radiology, 2018 Publication	<1 %
82	Lecture Notes in Computer Science, 2014. Publication	<1 %
83	ipfs.io Internet Source	<1 %
84	www.dcbd.nl Internet Source	<1 %
85	www.pharmaceutical-market-research.com Internet Source	<1 %
86	"Computer Vision – ACCV 2016", Springer Science and Business Media LLC, 2017 Publication	<1 %
87	"Medical Image Computing and Computer Assisted Intervention – MICCAI 2022", Springer Science and Business Media LLC, 2022 Publication	<1 %
88	Ali Furkan KAMANLI. "Real-time UAV (Unmanned vehicle) Tracking with	<1 %

Autonomous Drone in Flight", Research Square Platform LLC, 2023
Publication

89	Bo Liu, Jinwu Hu, Xiuli Bi, Weisheng Li, Xinbo Gao. "PGNet: Positioning Guidance Network for Semantic Segmentation of Very-High-Resolution Remote Sensing Images", Remote Sensing, 2022 Publication	<1 %
90	Jianmei Zhang, Hongqing Zhu, Pengyu Wang, Xiaofeng Ling. "ATT Squeeze U-Net: A Lightweight Network for Forest Fire Detection and Recognition", IEEE Access, 2021 Publication	<1 %
91	Jining Yan, Jingwei Liu, Lizhe Wang, Dong Liang, Qingcheng Cao, Wanfeng Zhang, Jianyi Peng. "Land-cover classification with time-series remote sensing images by complete extraction of multi-scale timing dependency", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2022 Publication	<1 %
92	Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, Mubarak Shah. "Transformers in Vision: A Survey", ACM Computing Surveys, 2022 Publication	<1 %

93	Submitted to University of Northumbria at Newcastle Student Paper	<1 %
94	core.ac.uk Internet Source	<1 %
95	gupea.ub.gu.se Internet Source	<1 %
96	link.springer.com Internet Source	<1 %
97	pure.rug.nl Internet Source	<1 %
98	repositorio.unb.br Internet Source	<1 %
99	www.diva-portal.org Internet Source	<1 %
100	www.ijste.org Internet Source	<1 %
101	www2.mdpi.com Internet Source	<1 %
102	"Natural Language Processing and Chinese Computing", Springer Science and Business Media LLC, 2018 Publication	<1 %
103	"Neural Information Processing", Springer Science and Business Media LLC, 2021	<1 %

Publication

104	Afshin Azizi, Yousef Abbaspour-Gilandeh, Edwige Vannier, Richard Dusséaux et al. "Semantic segmentation: A modern approach for identifying soil clods in precision farming", Biosystems Engineering, 2020 Publication	<1 %
105	Submitted to Kingston University Student Paper	<1 %
106	Marjan Stoimchev, Dragi Kocev, Sašo Džeroski. "Deep Network Architectures as Feature Extractors for Multi-Label Classification of Remote Sensing Images", Remote Sensing, 2023 Publication	<1 %
107	Wenxu Shi, Qingyan Meng, Linlin Zhang, Maofan Zhao, Chen Su, Tamás Jancsó. "DSANet: A Deep Supervision-Based Simple Attention Network for Efficient Semantic Segmentation in Remote Sensing Imagery", Remote Sensing, 2022 Publication	<1 %
108	Yao Li, Luo Chen. "Land Cover Classification for High Resolution Remote Sensing Images with Atrous Convolution and BFS", 2019 IEEE 5th International Conference on Computer and Communications (ICCC), 2019 Publication	<1 %

109	d197for5662m48.cloudfront.net Internet Source	<1 %
110	epdf.tips Internet Source	<1 %
111	idoc.pub Internet Source	<1 %
112	repositoriotec.tec.ac.cr Internet Source	<1 %
113	research.gold.ac.uk Internet Source	<1 %
114	www.diva-portal.se Internet Source	<1 %
115	www.igi-global.com Internet Source	<1 %
116	www.matthewtoews.com Internet Source	<1 %
117	www.paperdigest.org Internet Source	<1 %
118	"Artificial Intelligence Oceanography", Springer Science and Business Media LLC, 2023 Publication	<1 %
119	"Biomedical Image Registration", Springer Science and Business Media LLC, 2020 Publication	<1 %

120	Jay S Sevak, Aerika D. Kapadia, Jaiminkumar B. Chavda, Arpita Shah, Mrugendrasinh Rahevar. "Survey on semantic image segmentation techniques", 2017 International Conference on Intelligent Sustainable Systems (ICISS), 2017 Publication	<1 %
121	Shuyang Wang, Xiaodong Mu, Dongfang Yang, Hao He, Peng Zhao. "Road Extraction from Remote Sensing Images Using the Inner Convolution Integrated Encoder-Decoder Network and Directional Conditional Random Fields", Remote Sensing, 2021 Publication	<1 %
122	Yu Sun, Fukun Bi, Yangte Gao, Liang Chen, Suting Feng. "A Multi-Attention UNet for Semantic Segmentation in Remote Sensing Images", Symmetry, 2022 Publication	<1 %
123	Yufen Xu, Shangbo Zhou, Yuhui Huang. "Transformer-Based Model with Dynamic Attention Pyramid Head for Semantic Segmentation of VHR Remote Sensing Imagery", Entropy, 2022 Publication	<1 %
124	"Artificial Neural Networks and Machine Learning – ICANN 2021", Springer Science and Business Media LLC, 2021	<1 %

Publication

125	Lecture Notes in Computer Science, 2013. Publication	<1 %
126	Runrui Liu, Fei Tao, Xintao Liu, Jiaming Na, Hongjun Leng, Junjie Wu, Tong Zhou. "RAANet: A Residual ASPP with Attention Framework for Semantic Segmentation of High-Resolution Remote Sensing Images", Remote Sensing, 2022 Publication	<1 %
127	Wei Liu, Jiawei Liu, Zhipeng Luo, Hongbin Zhang, Kyle Gao, Jonathan Li. "Weakly supervised high spatial resolution land cover mapping based on self-training with weighted pseudo-labels", International Journal of Applied Earth Observation and Geoinformation, 2022 Publication	<1 %
128	Yangyang Mu, Ming Wang, Xuehan Zheng, He Gao. "An improved LSTM-Seq2Seq-based forecasting method for electricity load", Frontiers in Energy Research, 2023 Publication	<1 %
129	Zhen Zhang, Xin Huang, Jiayi Li. "DWin-HRFormer: A High-Resolution Transformer Model with Directional Windows for Semantic Segmentation of Urban Construction Land",	<1 %

IEEE Transactions on Geoscience and Remote Sensing, 2023

Publication

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		