SEMANTIC SEGMENTATION OF SATELLITE IMAGES USING TRANSFORMERS

MUHAMMAD HAZIQ FAIZ BIN MOHD RIPIN

SESSION 2021/2022

FACULTY OF COMPUTING AND INFORMATICS
MULTIMEDIA UNIVERSITY

JANUARY 2022

SEMANTIC SEGMENTATION OF SATELLITE IMAGES USING TRANSFORMERS

BY

MUHAMMAD HAZIQ FAIZ BIN MOHD RIPIN

SESSION 2021/2022

THIS PROJECT REPORT IS PREPARED FOR

FACULTY OF COMPUTING AND INFORMATICS
MULTIMEDIA UNIVERSITY
IN PARTIAL FULFILLMENT

FOR

BACHELOR OF COMPUTER SCIENCE

B.C.S (HONS) DATA SCIENCE

FACULTY OF COMPUTING AND INFORMATICS
MULTIMEDIA UNIVERSITY

January 2022

DECLARATION

I hereby declare that the work has been done by myself and no portion of the work contained in this Thesis has been submitted in support of any application for any other degree or qualification on this or any other university or institution of learning.

Muhammad Haziq Faiz Bin Mohd Ripin

Faculty of Computing and Informatics

Multimedia University

Date: 30:12:2022

ACKNOWLEDGEMENTS

Thanks guys. I owe you many.

To my parents, my husband, and my daughter.

ABSTRACT

This can be your **Management Summary** or **Abstract**. An abstract or management summary should be not more than one page in length. The abstract should allow the reader or moderator who is unfamiliar with the work to gain a swift and accurate impression of what the project is about, how it arose and what has been achieved.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

The preface in a report is something that comes before the report. This section will typically set up the stage for whatever your report is going to discuss. It may give some background information on the subject.

Normally a preface it will be a three paragraph length answer. The first paragraph should be explaining what you are investigating and why. the second should be the scope of your investigation. the third should be the conclusion that your investigation brought you to.

If your report does not have any preface, you may remove it from your latex.

CHAPTER 1

INTRODUCTION

1.1 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 (Yu et al., 2018). At that time, this terminology was equivalent to non-semantic image segmentation but emphasized that the segmented regions must contain a "semantic meaning". Semantic segmentation algorithms learn information about the classes

that each pixel belongs to before segmenting an image.they On the other hand, non-semantic segmentation algorithms try to detect consistent regions or region boundaries. Non-semantic segmentation can be solved using many unsupervised algorithms.

In the 1990s, “object segmentation and recognition” distinguished semantic objects of all classes from background and can be viewed as a two-class image segmentation problem. As the complete partition of foreground objects from the background is very 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.

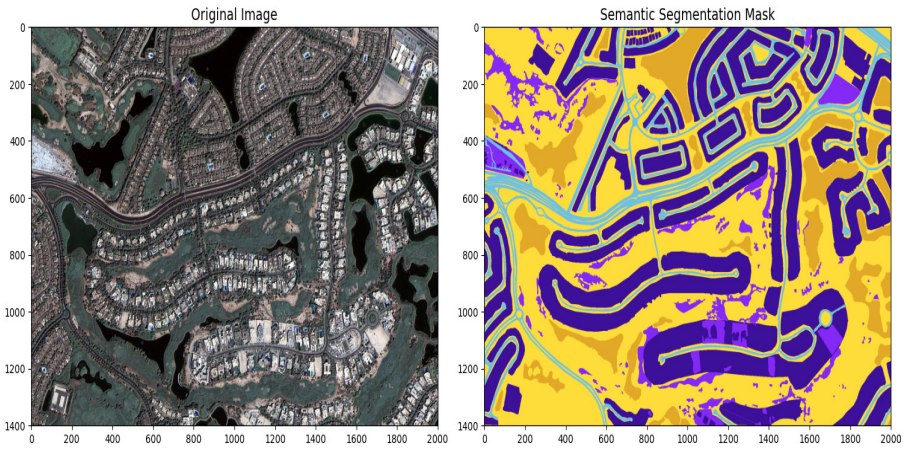

Figure 1.1: Semantic Segmentation of a Satellite Image

Figure 1.2: Object Detection of a Satellite Image

1.2 Introduction to Satellite Images

Satellite images are images of the Earth that are collected by either drones or 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 1.1: 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.

1.3 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 (Thoma, 2016). 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(Yuan et al., 2020).

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% (Talal et al., n.d.).

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 (Pandey, Kumar, & Chakraborty, 2021).

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 (Leach, Popien, Goodman, & Tellman, n.d.).

1.4 Problem Statement

1. All of the datasets studied suffer from class imbalanced meaning that the number of classes in the samples are not equally distributed. This is a very common problem in semantic segmentation task.

2. In the majority of the papers reviewed, semantic segmentation models are not trained using High Spatial Resolution (HSR) satellite images that are balanced.

1.5 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.

2. Design and train a transformer model to perform semantic segmentation using chosen dataset.

3. Identify appropriate evaluation metrics and use that metrics to evaluate the performance of our model so that we can provide benchmark for future experiments.

1.6 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 (px). 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.

1.7 Chapter Organization

CHAPTER 2

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.

2.1 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 2.1: Datasets for Semantic Segmentation Task

Table 2.1 shows the list of dataset that were evaluated and considered for this project. Each of the dataset is made for semantic segmentation task and is provided by the TorchGeo library. The dataset in bold are the ones that I think deserve more in depth exploration and discussion.

The Potsdam and Vaihingen datasets (Rottensteiner et al., 2012) 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 (L. Wang, Fang, Zhang, Li, & Duan,

2021), (L. Wang et al., 2022), (Li et al., 2022) and (Li, Zheng, & Duan, 2021) 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:

1. Impervious Surfaces - roads, concrete surfaces
2. Buildings
3. Low Vegetation
4. Trees
5. Cars
6. Clutter - representing uncategorizable land covers

Figure 2.1: An Image From Potsdam Dataset

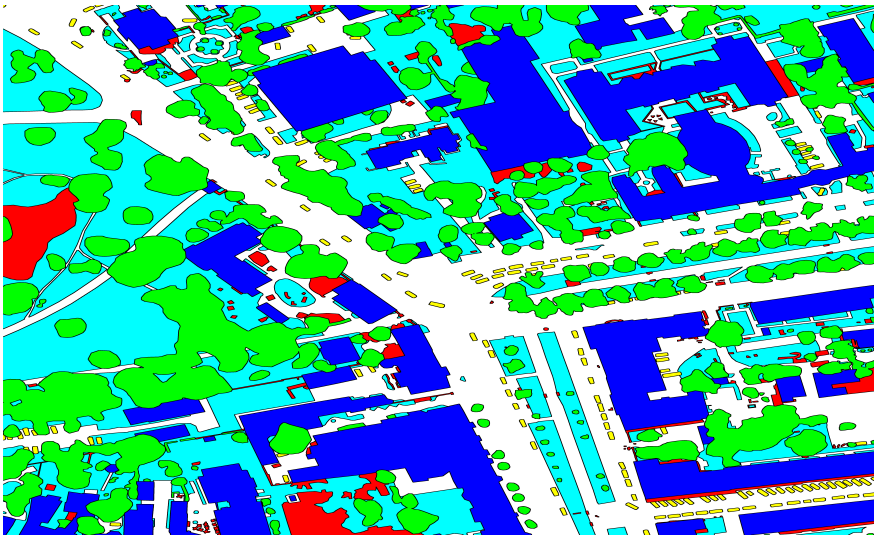

Figure 2.2: An mask From Potsdam Dataset

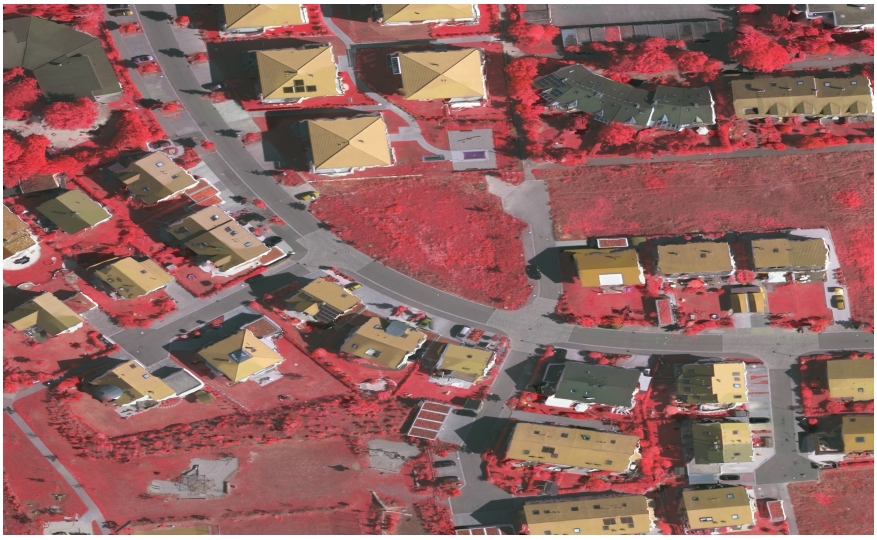

Figure 2.3: An Image From Vaihingen Dataset

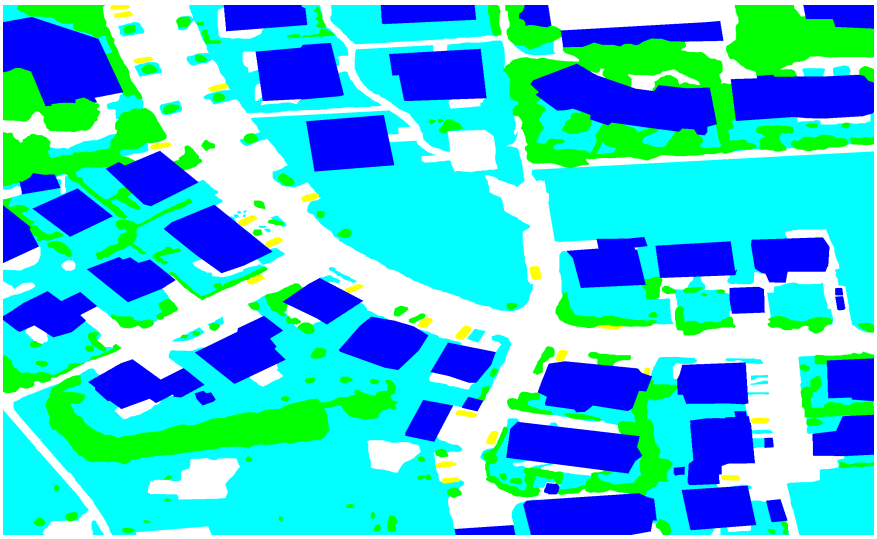

Figure 2.4: An mask From Vaihingen Dataset

The LoveDA dataset (J. Wang, Zheng, Ma, Lu, & Zhong, 2021) 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 2.5: An Image From LoveDa Dataset

masukka mask

The GID-5 dataset (Tong et al., 2020) contains 120 6800 x 7200 HSR images with

5 classes. The classes are built-up, farmland, meadow, farmland and water which are pixel-level labeled 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 2.6 and 2.7 shows an example of an image and its corresponding mask from Deep Globe dataset.

Figure 2.6: An Image From GID-5 Dataset

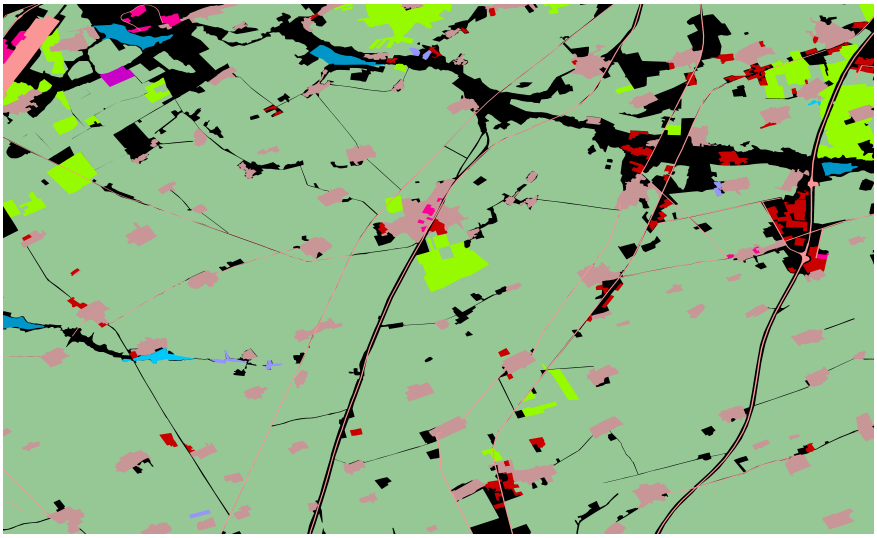

Figure 2.7: A Mask From GID-5 Dataset

The Deep Globe Land Cover dataset (Demir et al., 2018) 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 2.8 and 2.9 shows an example of an image and its corresponding mask from Deep Globe dataset.

Figure 2.8: An Image From Deep Globe Dataset

Figure 2.9: A 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 px) and 8 images with resolution 50 cm (4200 × 4700 px). 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.

masukkan landcover.ai

2.2 Semantic Segmentation Before Deep Learning

This section would elaborate on traditional methods of semantic segmentation, methods that do not apply any neural networks but make heavy use of domain knowledge and feature extraction methods.

2.2.1 Feature Extraction

Before we apply any of the classification method that will be discussed in the proceeding sections (SVM, Random Decision Forest, MRF, CRF) 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. (Thoma, 2016).

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. (Thoma, 2016). An image will be divided into smaller patches and the difference-of-Gaussian (DoG)is calculated.

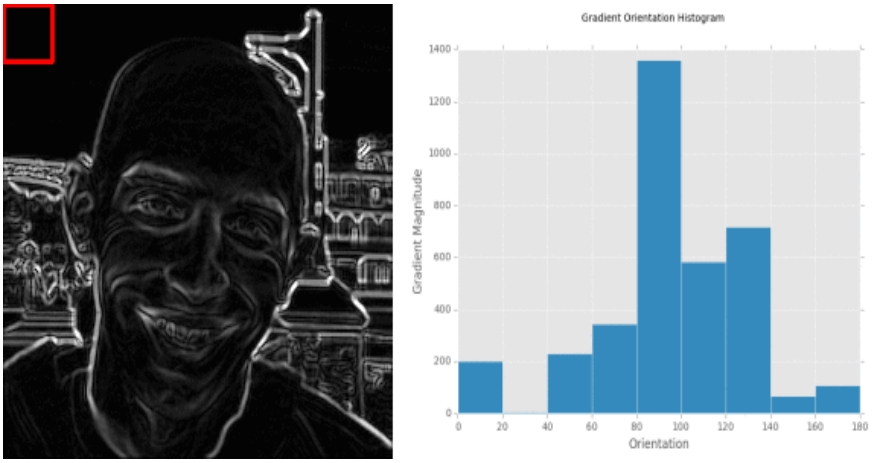

Figure 2.10: Computing a histogram of oriented gradients for the first patch of an input image.

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 (Lowe, 2004).

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 (Thoma, 2016). BOV is often used alongside other feature extractors such as SIFT by assigning each SIFT descriptor to the closest entry in a visual dictionary.

Figure 2.11: Each circle represents the location and orientation of SIFT keypoints

2.2.2 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 (Thoma, 2016). 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 (Schroff, Criminisi, & Zisserman, 2008).

2.2.3 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:

$$min_{w,b} \quad \frac{1}{2}||w||^2 \tag{2.1}$$

$$s.t. \quad y^i(w^T x^i + b) \geqslant 1, i \in 1...m \tag{2.2}$$

w is the linear combination of the training data x:

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{2.3}$$

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 $\psi(x)$, this method is called the *kernel trick*. Specifically, given a feature mapping ψ, we define the corresponding kernel to be:

$$K(x,z) = \psi(x)^T \psi(z) \tag{2.4}$$

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.

2.2.4 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 (Yu et al., 2018). 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:

$$P(x,y) = \frac{1}{Z} e^{-E(x,y)} \tag{2.5}$$

where $Z = \sum_{x,y} e^{-E(x,y)}$ and it is called the partition function whereas E is called 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 (Thoma, 2016). A thorough presentation of MRF can be found in (Blake, Kohli, & Rother, 2011).

2.2.5 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 $P(x|y)$ (Thoma, 2016). 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 (Yu et al., 2018). CRF has the partition function $Z(x)$:

$$Z(x) = \sum_{x} P(x,y) \tag{2.6}$$

and the joint probability distribution is given as follows:

$$P(y|x) = \frac{1}{Z(x)} \prod_{c \in C} \psi_c(y_c|x) \tag{2.7}$$

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 (Teichmann & Cipolla, 2018).

2.3 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 (Krizhevsky, Sutskever, & Hinton, 2012) had an error rate of 17.0% (Thoma, 2016).

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.

2.4 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) (Long, Shelhamer, & Darrell, 2015) is the first network proven to be an effective method to extract features automat-

ically 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 (Ronneberger, Fischer, & Brox, 2015) which introduced the encoder-decoder framework. Since its introduction, the encoder-decoder framework has become the standard structure of satellite images segmentation network (L. Wang, Fang, et al., 2021). 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 by concatenate the output of the decoder with the feature maps from the encoder. Figure 2.12 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.

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 modi-

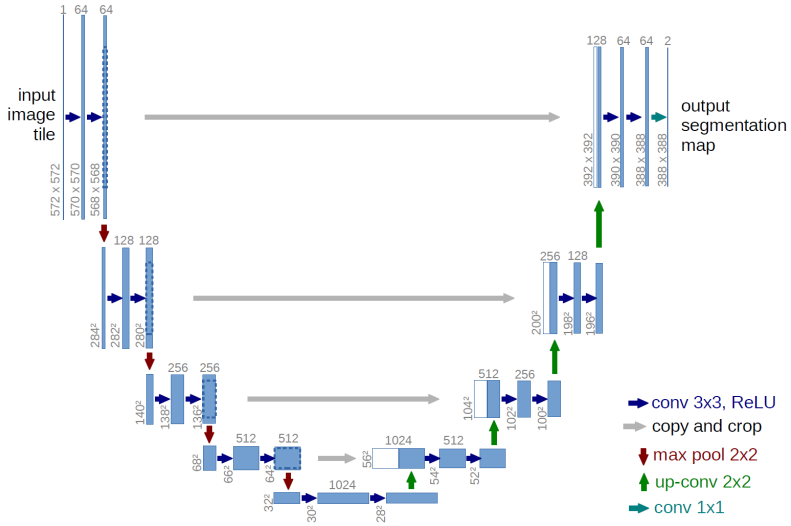

Figure 2.12: U-Net Architecture

fying the convolution operation or utilizing the attention mechanism. Such examples of the first approach is to enlarge the receptive fields using large kernel sizes (W. Liu, Zhang, Lin, & Liu, 2022), or utilising feature pyramids (Zhao, Shi, Qi, Wang, & Jia, 2016). 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 (Li, Zheng, Zhang, et al., 2021) and Attention U-Net (Oktay et al., 2018). 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 (Z. Liu, Lin, et al., 2021).

Attention U-Net (Oktay et al., 2018) is a model that aims to improve U-Net by in-

troducing attention gates in the skip connections between the encoder and decoder as shown in 2.13. The results showed that Attention U-Net performs better than traditional U-Net for segmentation of CT scans images.

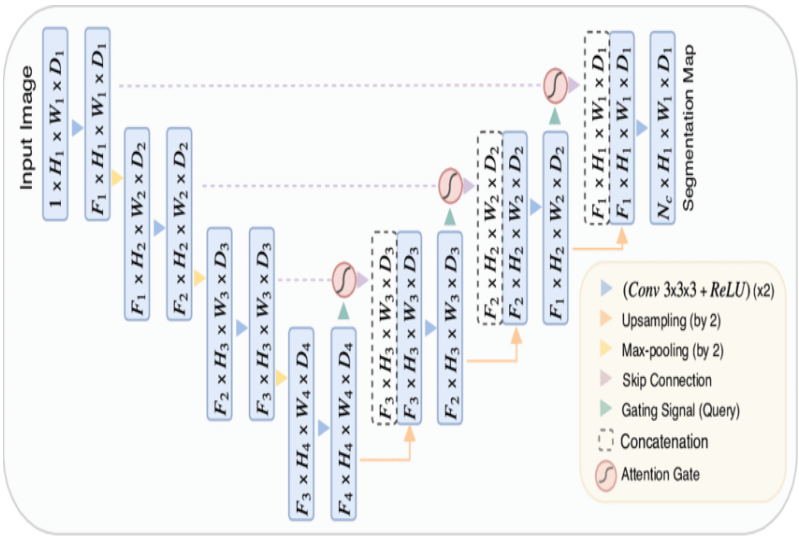

Figure 2.13: Attention U-Net Architecture

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) (Sun, Bi, Gao, Chen, & Feng, 2022) uses residual structure and simple attention modules. According to figure 2.14 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 compareed with other U-Net based models susch as U-Net, U-Net++, Attention U-Net and MagNet 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) (Fan, Yan, Fan, & Wang, 2022) uses atrous spatial pyramid pooling as its encoder and attention modules at every skip connection like we have seen in Attention U-Net. (Tao, Ding, & Cao, 2021) 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.

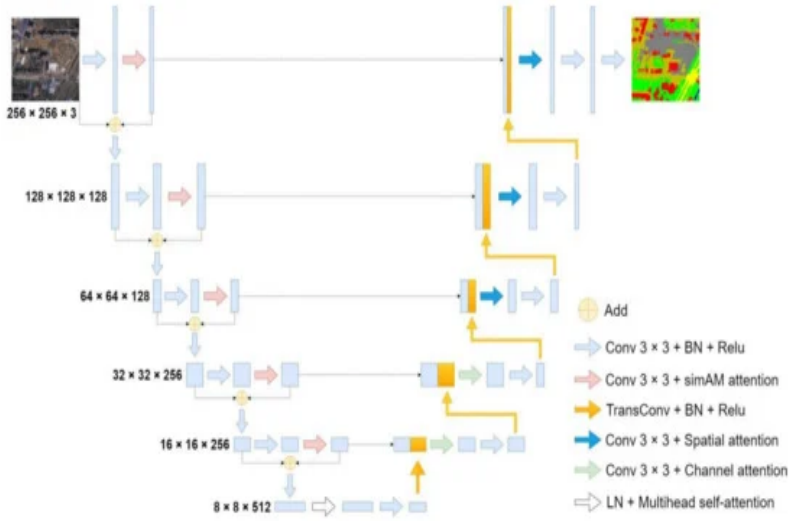

Figure 2.14: Multi Attention U-Net Architecture

2.5 Semantic Segmentation of Satellite Images Using Vision Transformers

Transformers were introduced by (Vaswani et al., 2017) and since its inception it has been the de facto model for Natural Language Processing (NLP). Transformers that are used for image processing 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. (Papoutsis, Bountos, Zavras, Michail, & Tryfonopoulos, 2021) 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.

2.5.1 Vision Transformer (ViT)

Vision Transformer from (Dosovitskiy et al., 2020) 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 translation equivariance and locality provide by the 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 2.15 shows the ViT architecture.

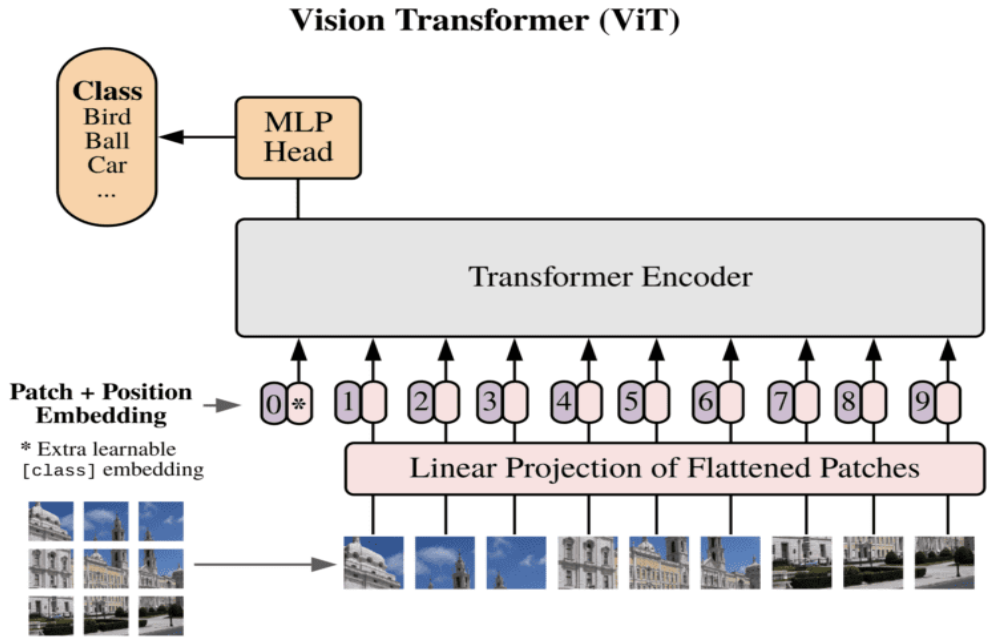

Figure 2.15: ViT Architecture

2.5.2 Swin Transformer

The first version of Swin Transformer (Z. Liu, Lin, et al., 2021) 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 Tansformer will use patch embedding to the input image but the difference now is the patch size is 4x4x3 instead of 16x16x3. The biggest differ-

ence between ViT and Swin Transformer is ViT attention layers work on the entire image while Swin Transformer will firsst compute attention on local windows. Take an example in figure 2.16, the input image is 16x16 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.

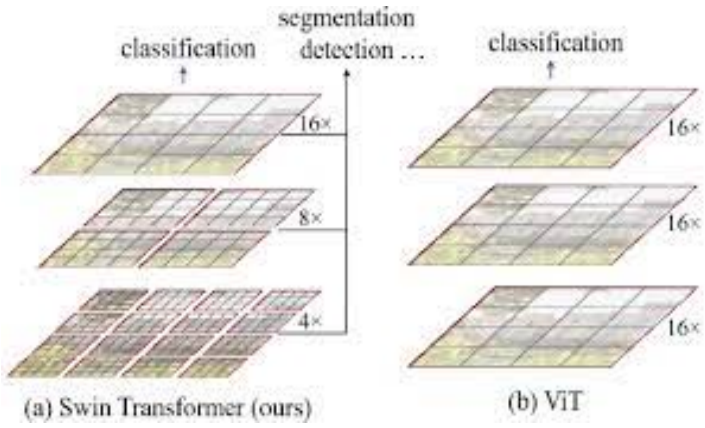

Figure 2.16: The difference between Swin Transformer and ViT

Swin Transformer is made up of 4 stages as shown in 2.17. 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.

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.

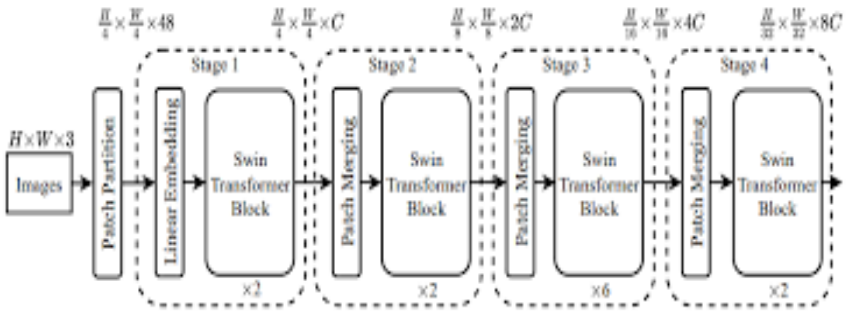

Figure 2.17: Swin Transformer V1 Architecture

The self-attention modeule in ViT has $O(n^2)$computational complexity to the number of tokens which makes it unsuitable forsemantic segmentation which require dense high-resolution classifications. Swin Transformer introduced a Shifted Window Multi-headed Self-Attention (W-MSA) to make the self-attetion linear to the number of tokens. The computational complexity of the original Multi-Headed Self-Attention is:

$$\Omega(\mathbf{MSA}) = 4hwC^2 + 2(hw)^2C \tag{2.8}$$

While the improved computational complexityof W-MSA is linear to the number of

patches as shown in the equation below:

$$\Omega(\mathbf{W}-\mathbf{MSA}) = 4hwC^2 + 2M^2hwC \tag{2.9}$$

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 2.19

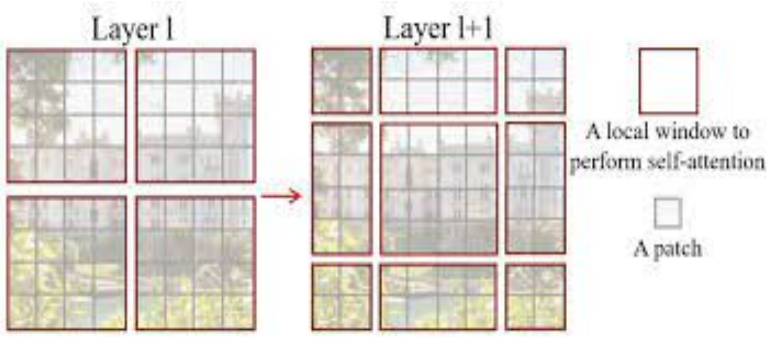

Figure 2.18: Shifted Window Multi-head Self-Attention

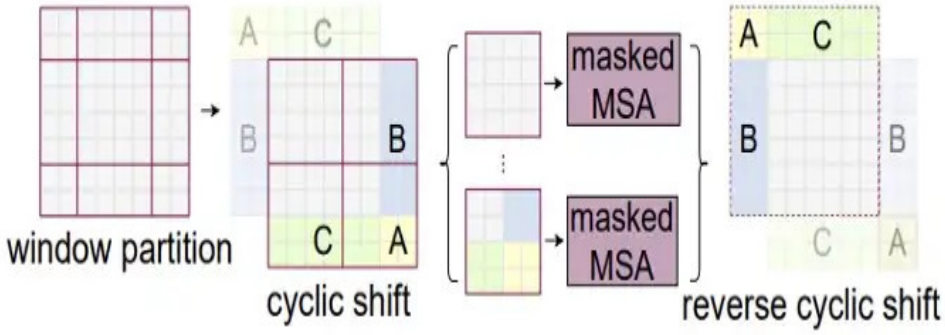

Figure 2.19: Cyclic Shifted Windows in Swin Transformer

SW-MSA successfully introduced connections between windows while masking prevents the attention computation from processing across the image boundary. Figure 2.20 show two consecutiveSwin Transformer blocks with each one having a different partitioning scheme.

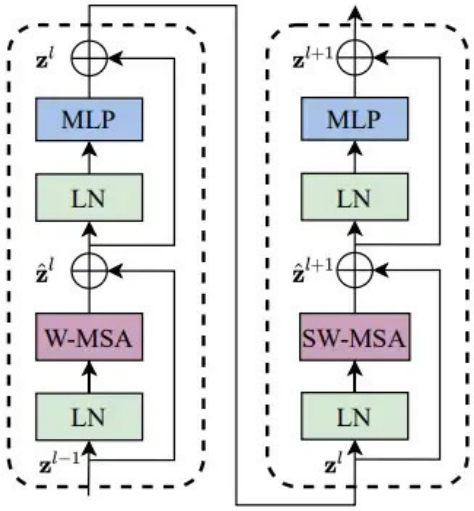

Figure 2.20: Two Successive Swin Transformer Blocks

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

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d}+B})V \tag{2.10}$$

Swin Transformer V2 (Z. Liu, Hu, et al., 2021) was introsuced in 2021 and it is an improved version of Swin Transformer to increase scability, accuracy and stablity during training. As shown in figure 2.21, the Layer Normalization block is moved from front to back of the residual unit and a new scaled cosine self-attention unit is used inplace of the old matrix multiplication based self-attention. The new self-attention score can be calculated as:

$$Sim(\mathbf{q}_i,\mathbf{k}_j) = \frac{cos(\mathbf{q}_i,\mathbf{k}_j)}{\tau} + B_{ij} \tag{2.11}$$

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.

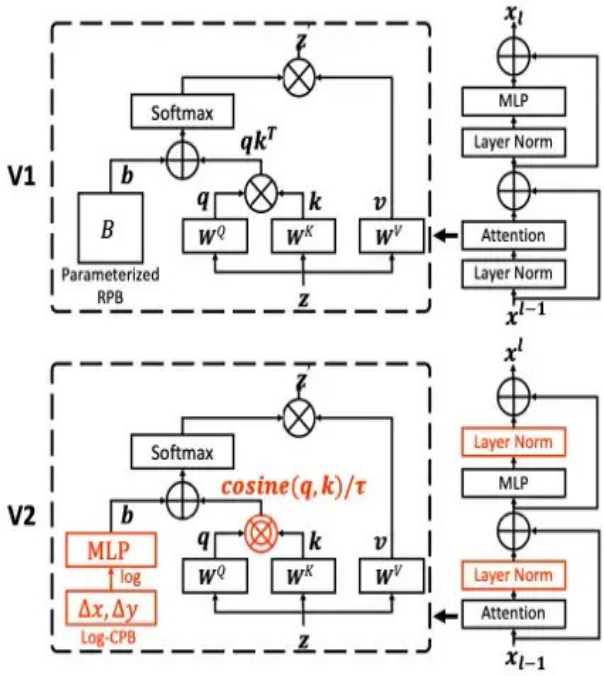

Figure 2.21: Difference Between Swin V1 and Swin V2

2.5.3 UNetFormer

UNetFormer (L. Wang, Fang, et al., 2021) 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 2.22 UNetFormer uses pretrained ResNet18 as its encoder 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.

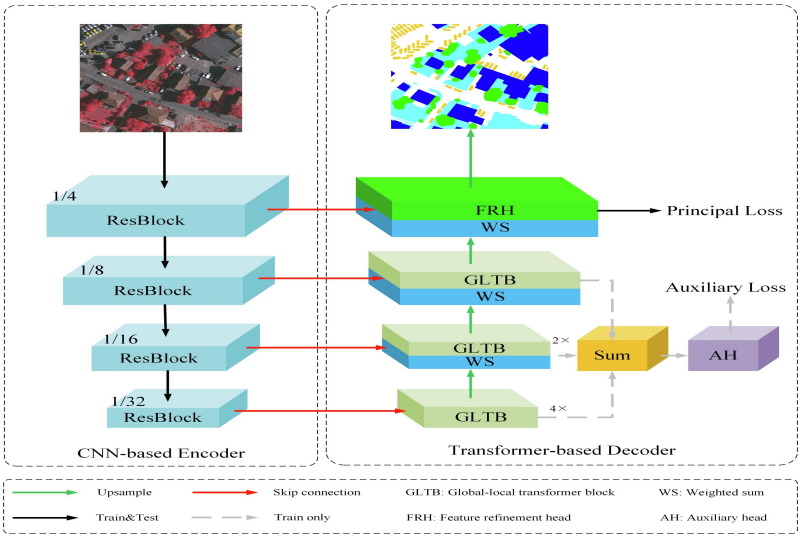

Figure 2.22: UNetFormer Architecture

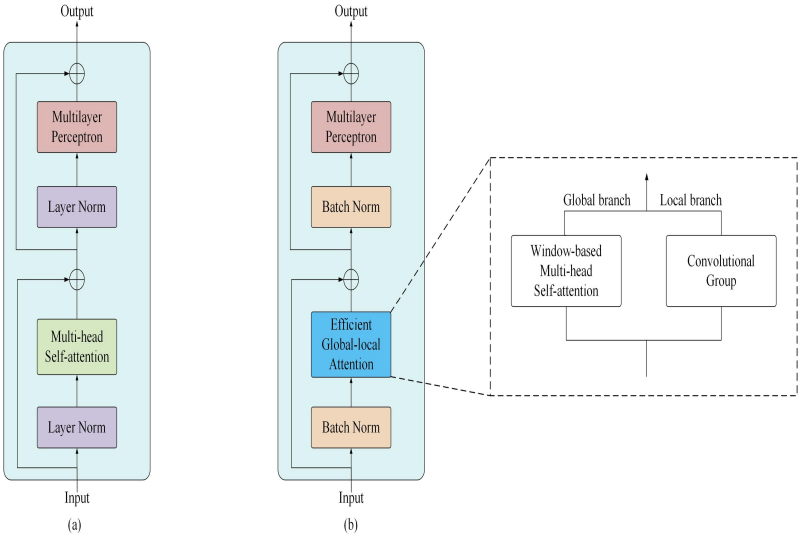

Figure 2.23: Illustration of (a) the standard Transformer block and (b) the Transformer block with GLTB

The GLTB is made up of the global–local attention, multilayer 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 2.24. 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 2.24 the global btanch uses the same window-based multi-head self-attention in Swin Transformer to capture the global context.

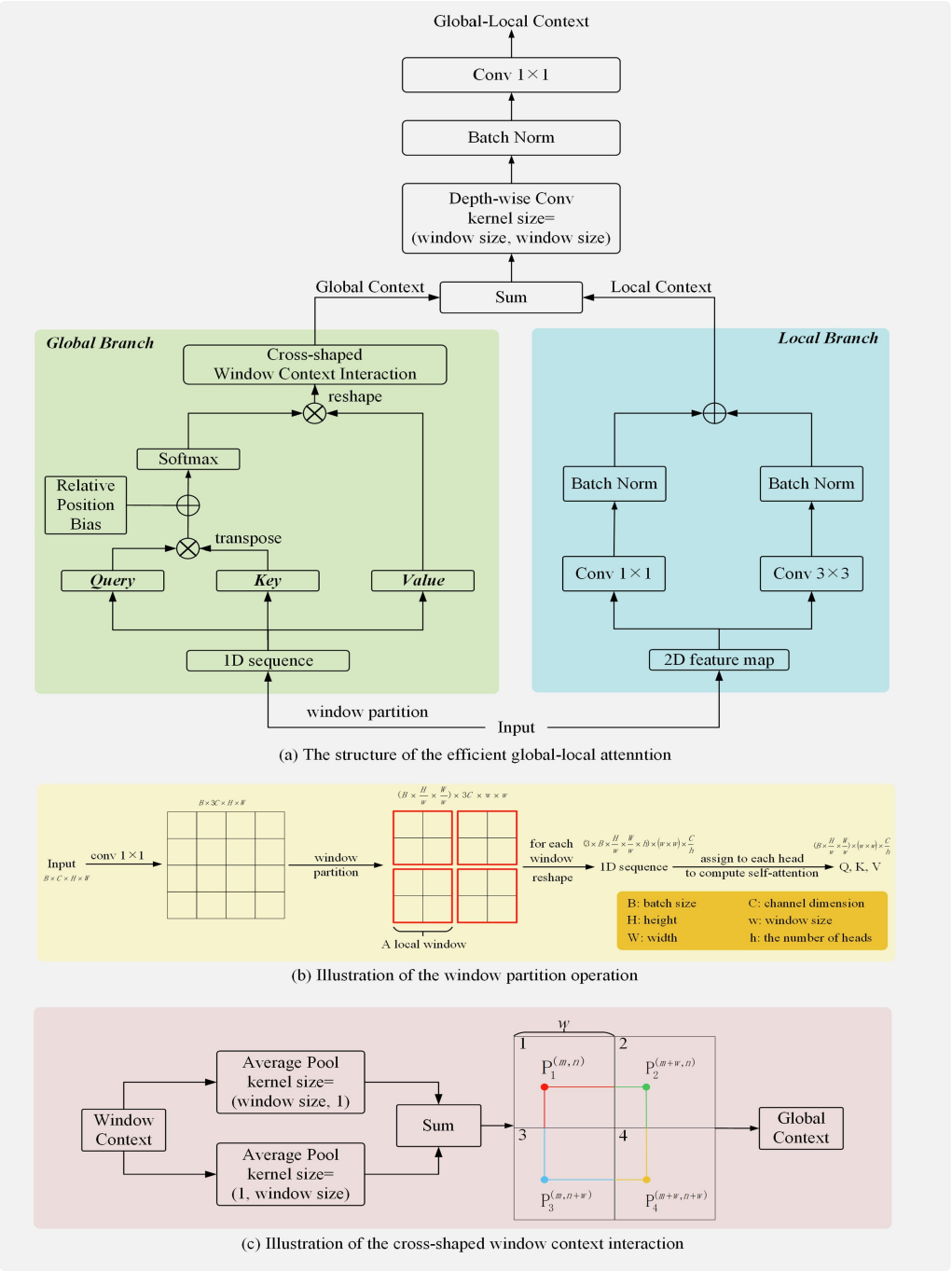

Figure 2.24: Cross-shaped Window Context Interaction in UNetFormer

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:

$$P_1^{m,n} = \frac{\sum_{i=0}^{w-m-1} P_1^{m+i,n} + \sum_{j=0}^{m} P_2^{m+w-j,n}}{w} \tag{2.12}$$

$$P_1^{m+i,n} = D_i(P_1^{m,n}) \tag{2.13}$$

$$P_2^{m+w-j,n} = D_j(P_2^{m+w,n}) \tag{2.14}$$

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 realtionship is $Win_1 = V(H(Win_4)) + H(V(Win_4))$.

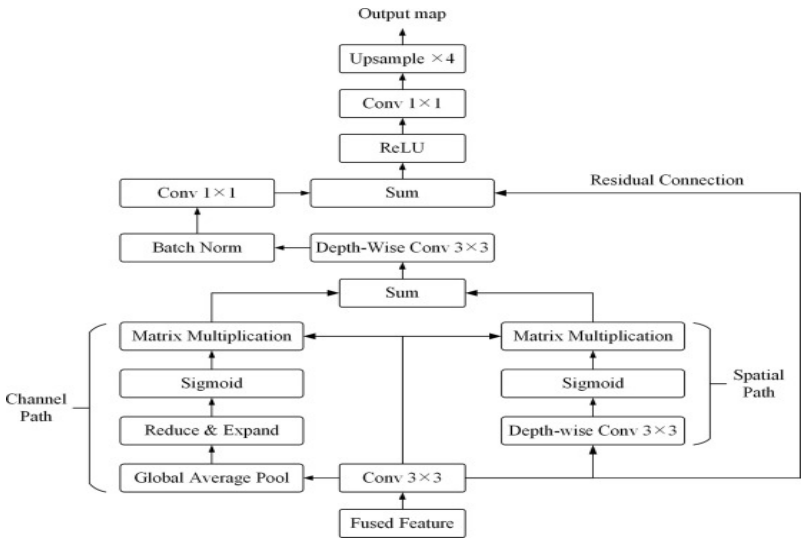

Figure 2.25: Feature refinement Head of UNetFormer

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 2.22 as its input. Then, two path are constructed to strengthen the channel-wise and spatial-wise feature representation as shown in figure 2.25. 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 attentional map $\mathbf{C} \in \mathbb{R}^{1\times1\times c}$, 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 attentional 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 upsampling operation are applied as post processing.

2.5.4 LANet

Local Attention Network (LANet) (Ding, Tang, & Bruzzone, 2021)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 2.26 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 ouput is the fusion of the outputs from the top and bottom channels.

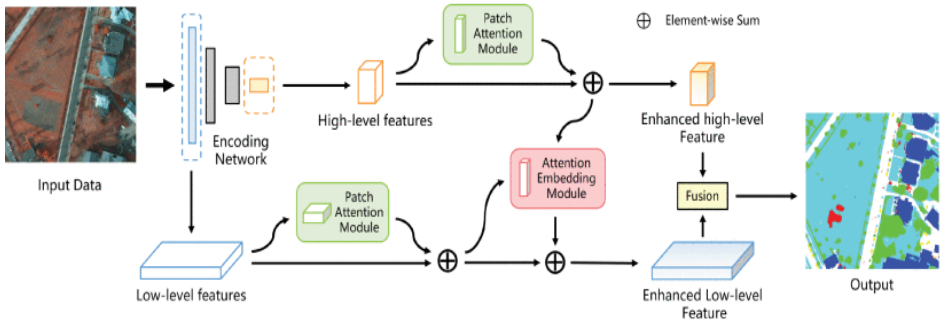

Figure 2.26: LANet Architecture

Referring to figure 2.27, 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

$$z_c = \frac{1}{h_p w_p} \sum_{i=1}^{h_p} \sum_{j=1}^{w_p} x_c(i,j) \tag{2.15}$$

where h_p and w_p is the horizontal and vertical size of the patch and x_c is the pixel value at the c th 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 upsampling operation.

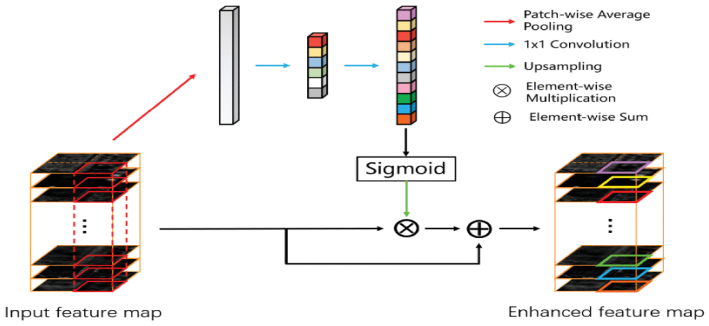

Figure 2.27: Patch Attention Module in LANet

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 2.28 shows the implementation of AEM. First, they generate a high-level feature descriptor map, z_c and low-level descriptor map , x_c usinging the same formula in PAM. Then, an attention map, a_c is generated using average pooling, 1 x 1 convolution and upsampling. Finally a_c will be added to x_c to generate an improved low-level descriptor.

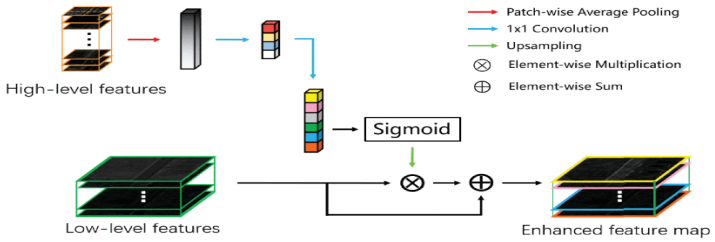

Figure 2.28: Attention Embedding Module in LANet

2.5.5 DC-Swin

A novel semantic segmentation scheme of densely connected (DC-Swin) (L. Wang et al., 2022) 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.

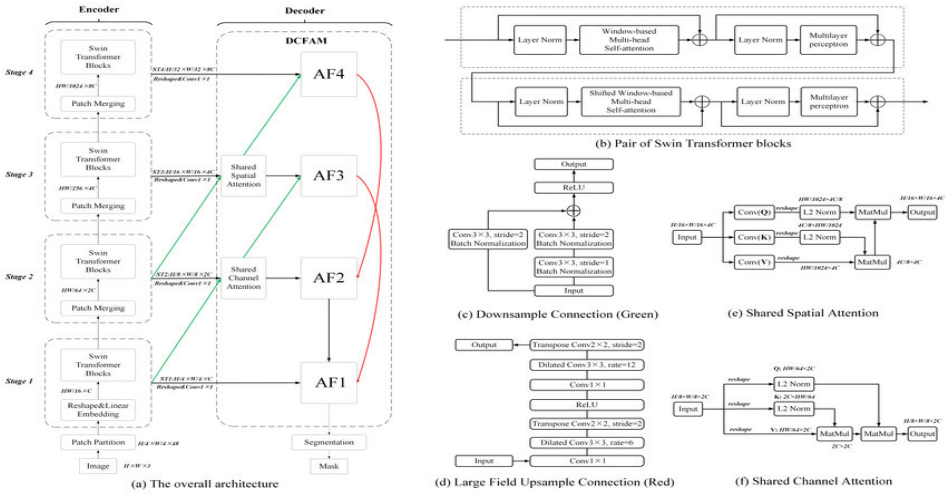

Figure 2.29: (a) Overall architecture of DC-Swin. (b) Pair of Swin Transformer blocks. (c) Downsample Connection. (d) Large Field Upsample Connection. (e) SSA. (f) SCA

Referring to figure 2.29(a), the Swin Transformer split the input RGB image into nonoverlapping 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 2.29(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 1 1 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 m lti-scale. According to figure 2.29 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 2.29 connects the low-level and high-level transformer features for fusion and it can be defined as follow:

$$D_j^i(\mathbf{X}) = f_\sigma(f_\delta(\mathbf{X}) + f_\mu(f_\theta(\mathbf{X}))) \tag{2.16}$$

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:

$$LU_m^n(\mathbf{X}) = f_\varphi^{12}(f_\sigma(f_\varphi^6(\mathbf{X}))) \tag{2.17}$$

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^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: jangan fail

$$SSA(\mathbf{X}) = \frac{\sum_n V(\mathbf{X}_{c,n}) + \frac{Q(\mathbf{X})}{|||Q(\mathbf{X})||_2}((\frac{K(\mathbf{X})}{||K(\mathbf{X}))||_2})^T V(\mathbf{X})}{N + \frac{Q(\mathbf{X})}{||Q(\mathbf{X})||_2}\sum_n(\frac{K(\mathbf{X})}{||K(\mathbf{X}))||_2})^T} \tag{2.18}$$

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:

$$SCA(\mathbf{X}) = \frac{R(\mathbf{X}_{c,n}) + (R(\mathbf{X}_{c,n}))\frac{R(\mathbf{X})}{||R(\mathbf{X}))||_2})^T \frac{R(\mathbf{X})}{||R(\mathbf{X}))||_2}}{N + \frac{R(\mathbf{X})}{||R(\mathbf{X})||_2}\sum_c(\frac{R(\mathbf{X})}{||R(\mathbf{X}))||_2})^T} \tag{2.19}$$

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

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

$$\mathbf{AF_4} = \mathbf{ST_4} + D_{384}^{768}(SSA(D_{192}^{384}(\mathbf{ST_2})) \tag{2.20}$$

$$\mathbf{AF_3} = SSA(\mathbf{ST_3}) + D_{192}^{384}(SSA(D_{96}^{192}(\mathbf{ST_1})) \tag{2.21}$$

$$\mathbf{AF_2} = SCA(\mathbf{ST_2}) + LU_{768}^{192}(\mathbf{AF_4}) \tag{2.22}$$

$$\mathbf{AF_1} = \mathbf{ST_1} + U(\mathbf{AF_2}) + LU_{384}^{96}(\mathbf{AF_3}) \tag{2.23}$$

where U is a bilinear interpolation upsample 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.

2.5.6 BANet

The architecture of Bilateral Awareness Network (BANet) is shown in figure **??** and it is madeup 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.

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 pathwill generate two long

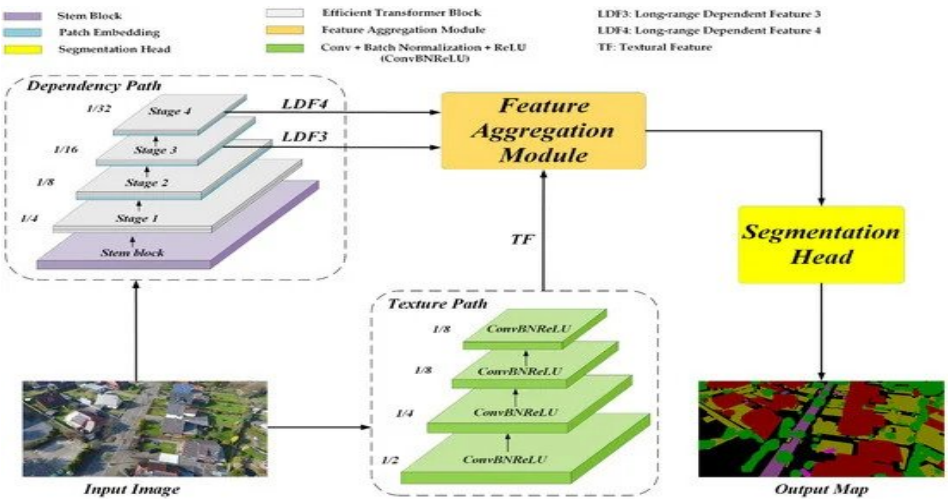

Figure 2.30: Architecture of Bilateral Awareness Network (BANet)

range dependent features namely the LDF3 and LDF4. The ETB is made up of ResT-Lite (Zhang & bin Yang, 2021) pretrained on ImageNet. ETB as illustrated in figure 2.31 uses efficient multihead self-attention (EMSA) instead of the usual multi-headed self attention.

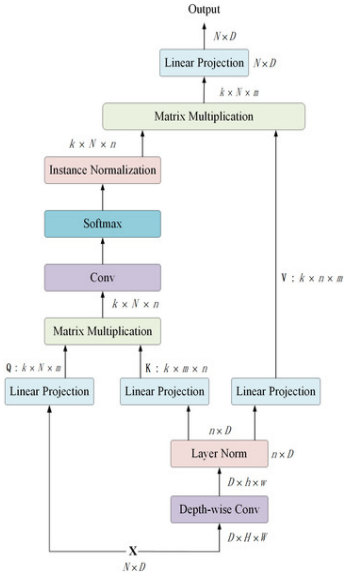

Figure 2.31: Efficient Transformer Block

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

$$EMSA(\mathbf{Q},\mathbf{K},\mathbf{V}) = LP(IN(softmax(conv(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{m}}))).\mathbf{V}) \tag{2.24}$$

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

$$TF(\mathbf{X}) = T_4(T_3(T_2(T_1(\mathbf{X})))) \tag{2.25}$$

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 downscaled 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 2.32. 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.

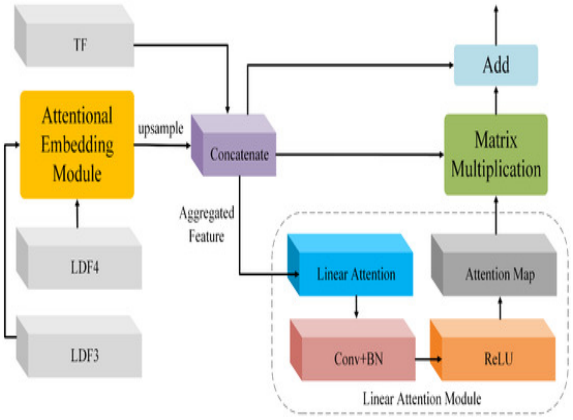

Figure 2.32: Feature Aggregation Module

2.5.7 AESwin-UNet

Adaptive Enhanced Swin Transformer with U-Net (AESwin-UNet) (X. Gu et al., 2022) uses a combination of Vision Transformer and U-Net for semantic segmentation of satellite images. As shown in figure 2.33 instead of using CNN for the encoder, they opted to use Enhanced Swin Transformer instead. For the decoder, each step involves an up-sampling of the feature map followed by a 2×2 deconvolution, a concatenation with a feature map from the encoder, and two 3×3 convolutions.

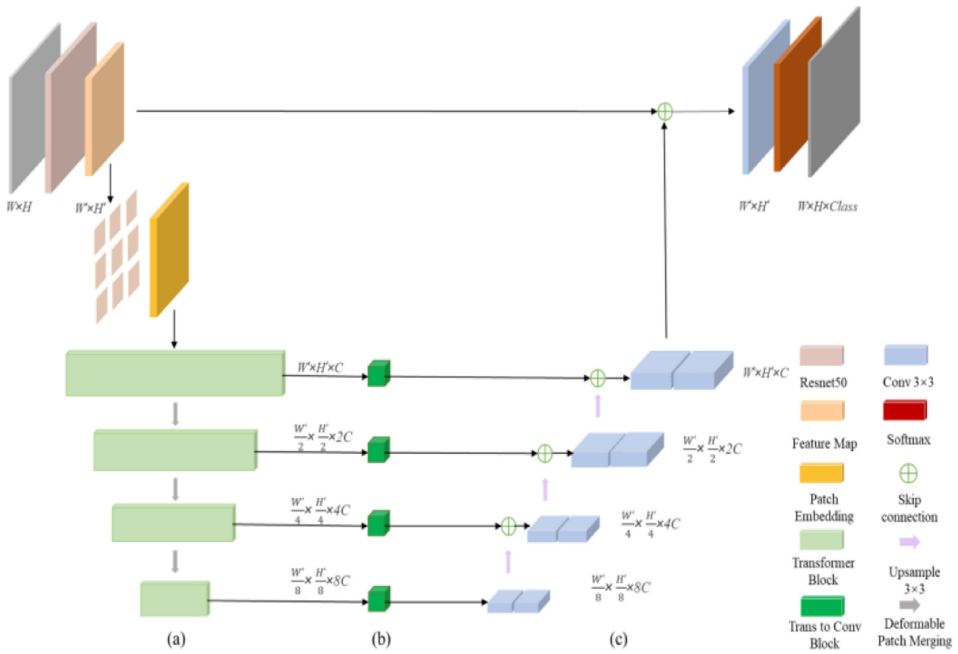

Figure 2.33: Architecture of Adaptive Enhanced Swin Transformer with U-Net (AESwin-UNet)

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:

$$Attention(Q,K,V) = Softmax(att(QK^T)\sqrt{d_k} + B)V \tag{2.26}$$

2.5.8 Summary of Literature Review

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

Model \ Dataset	Potsdam	Vaihingen	LoveDA	UAVid
UNetFormer (L. Wang, Fang, et al., 2021)	-	-	-	-
LANet (Ding et al., 2021)	91.95	88.09	-	-
DC-Swin (L. Wang et al., 2022)	93.25	90.71	-	-
BANet (L. Wang, Li, et al., 2021)	92.50	89.58	-	-
AESwin-Unet (X. Gu et al., 2022)	-	-	-	-

Table 2.2: F1-score achieved by each model.

Model \ Dataset	Potsdam	Vaihingen	LoveDA	UAVid
UNetFormer (L. Wang, Fang, et al., 2021)	85.5	81.6	-	70
LANet (Ding et al., 2021)	-	-	-	-
DC-Swin (L. Wang et al., 2022)	87.56	83.22	-	-
BANet (L. Wang, Li, et al., 2021)	86.25	81.35	-	64.6
AESwin-Unet (X. Gu et al., 2022)	-	-	66.1	-

Table 2.3: mIoU score achieved by each model.

Model \ Dataset	Potsdam	Vaihingen	LoveDA	UAVid
UNetFormer (L. Wang, Fang, et al., 2021)	-	-	-	-
LANet (Ding et al., 2021)	90.84	89.83	-	-
DC-Swin (L. Wang et al., 2022)	92.00	91.63	-	-
BANet (L. Wang, Li, et al., 2021)	91.06	90.48	-	-
AESwin-Unet (X. Gu et al., 2022)	-	-	53.96	-

Table 2.4: Accuracy score achieved by each model.

2.6 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 (Carion et al., 2020), LearnRegionFeat (J. Gu, Hu, Wang, Wei, & Dai, 2018) and RelationNet++ (Chi, Wei, & Hu, 2020) model the relationship between object and pixel to achieve SOTA 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. This two layers complement each other very well and works such

as DETR (Carion et al., 2020) and Swin Transformer V2 (Z. Liu, Hu, et al., 2021) 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 whewn they are training CNN models. There are less steps to do parallel processing when training a Transformer model (Z. Liu, Lin, et al., 2021). 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 (Riquelme et al., 2021). Figure 2.34 shows the recorded number of parameters in Vision Transformer models from 2018 until 2022

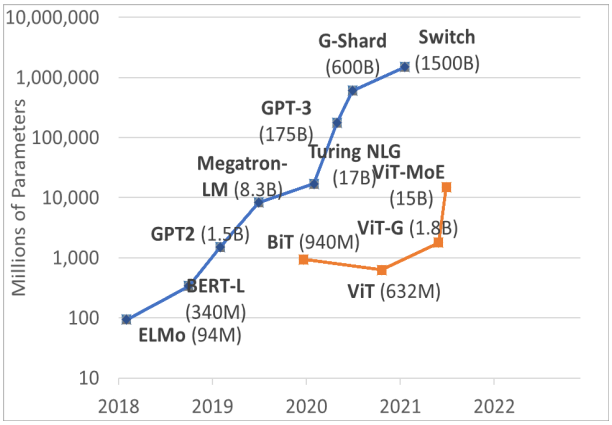

Figure 2.34: The size records of Vision Transformer in recent years

CHAPTER 3

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.

3.1 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 3.1. 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.

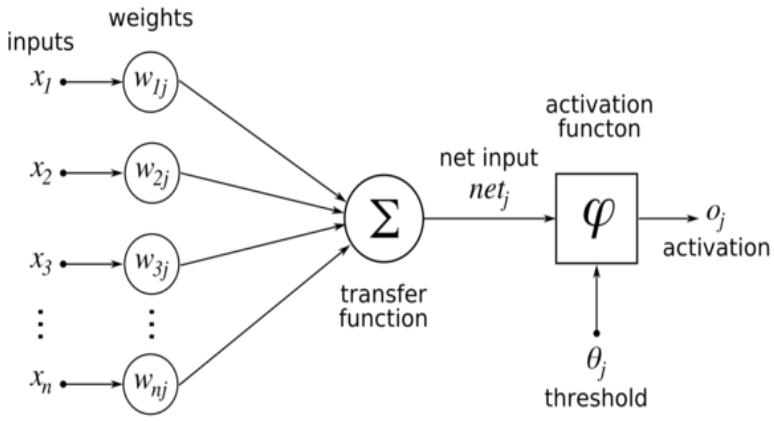

Figure 3.1: Rosenblatt Perceptron

LeCun et al., published a method to recognized hand-written digits and it was utilized bythe U.S. Postal Service (LeCun et al., 1989), 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 back-propagation, the weight learning algorithm in neural networks, 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 (Vaswani et al., 2017) and became the most popular architecture in NLP. In 2020, Vision Transformers (Dosovitskiy et al., 2020) introduced a method to adapt Transformers for computer vision by splitting the input image into patches and represent them as vectors.

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

3.3 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 3.3.1 and 3.3.2 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.

3.3.1 Encoder-Decoder Architecture

Because Transformers has its origin in NLP that relies on sequential input, the use of encoder-decoder architecture as shown in 3.2 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.

3.3.2 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.

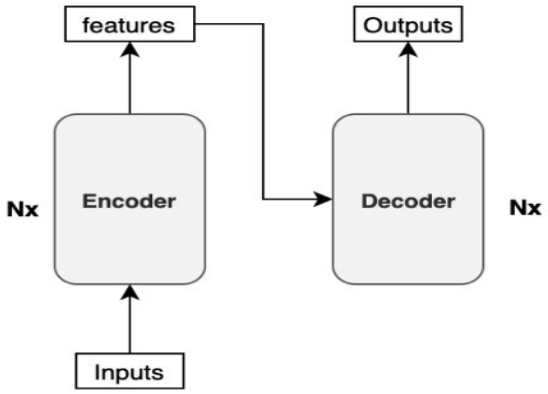

Figure 3.2: Encoder-Decoder Architecture

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:

$$\mathbf{h}_t = tanh(\mathbf{W}\mathbf{h}_{t-1} + \mathbf{W}\mathbf{x}_t) \tag{3.1}$$

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.

3.3.3 Attention Mechanism

Attention mechanism was first introduced by (Bahdanau, Cho, & Bengio, 2014) and it is the most fundamental part of a Transformer. 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

$$\mathbf{y}_i = \sum w_{ij} \mathbf{x}_j \tag{3.2}$$

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 soft,ax function to map the values to [0,1] and to ensure that they sum to 1 over the entire sequence:

$$w_{ij} = \frac{expw_{ij}}{\sum_j expw_{ij}} \tag{3.3}$$

3.3.4 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 matices are developed for each role namely they are $\mathbf{W_q}, \mathbf{W_k}, \mathbf{W_v}$ and 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:

$$\mathbf{q}_i = \mathbf{W}_q \mathbf{x}_i \tag{3.4}$$

$$\mathbf{k}_i = \mathbf{W}_k \mathbf{x}_i \tag{3.5}$$

$$\mathbf{v}_i = \mathbf{W}_v \mathbf{x}_i \tag{3.6}$$

$$ATTENTION(\mathbf{q}, \mathbf{k}, \mathbf{v}) = softmax(\frac{\mathbf{q}_i \mathbf{k}_j^T}{\sqrt{d_k}})\mathbf{v}_i \tag{3.7}$$

$$\mathbf{y}_i = \sum ATTENTION(\mathbf{q}, \mathbf{k}, \mathbf{v})\mathbf{x}_j \tag{3.8}$$

The attention mechanism is shown in figure 3.3

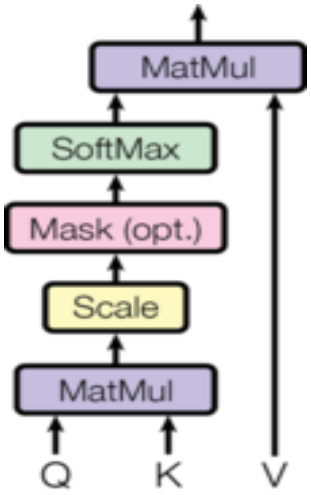

Figure 3.3: Attention Mechanism Unit

3.3.5 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 3.4.

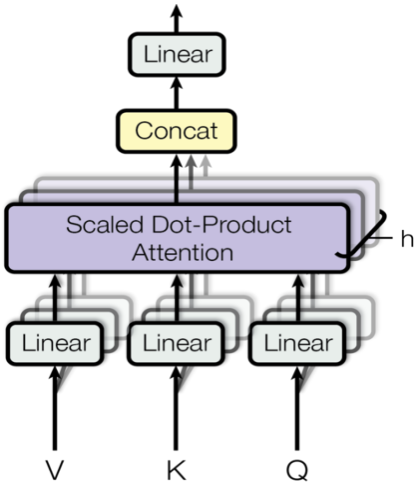

Figure 3.4: Multi-head Attention

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**, $/mathbfQ$ 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**, $/mathbfQ$ and **V** matrix as shown in **??**. The entire Transformer architecture with multi-head attention head is shown in figure 3.5

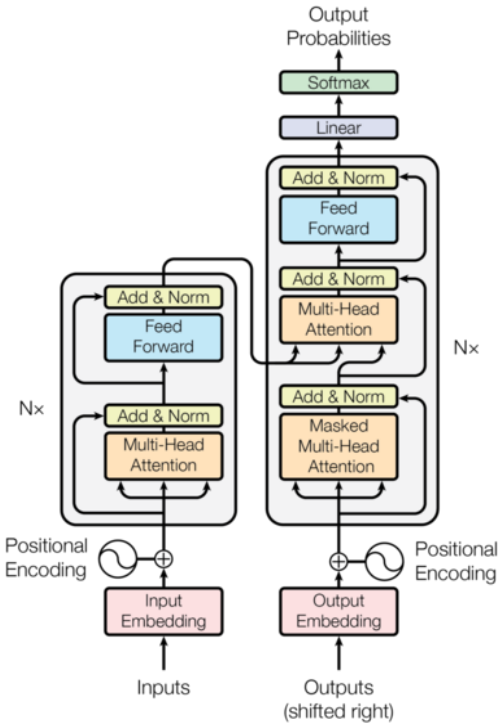

Figure 3.5: Transformer Architecture

3.4 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.

3.4.1 Sigmoid Function

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

$$\sigma(x) = \frac{1}{1+exp(-x)} \tag{3.9}$$

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

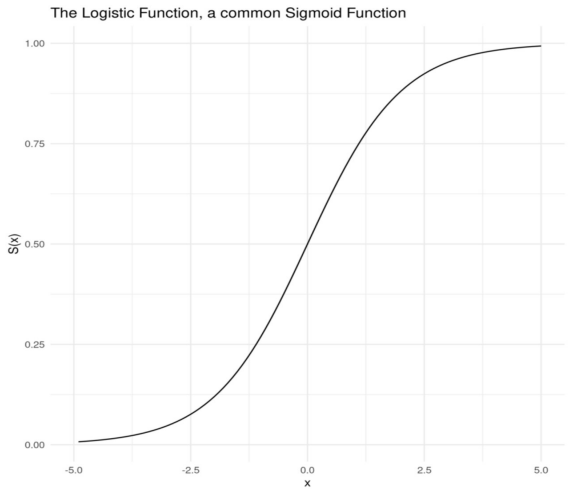

Figure 3.6: Sigmoid Function

3.4.2 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 3.7 shows a plot of the ReLU function. ReLu function

can be written as:

$$f(x) = max(0,x) \tag{3.10}$$

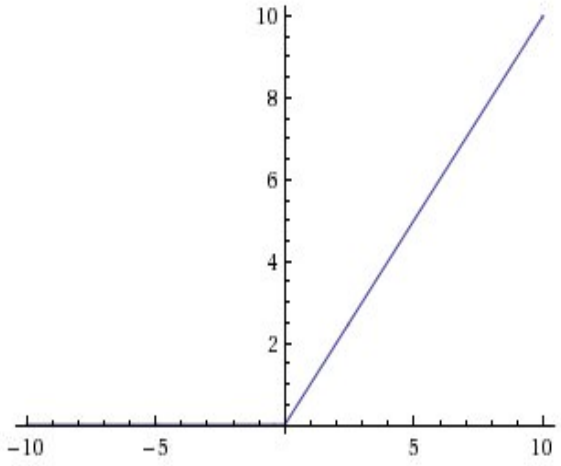

Figure 3.7: ReLU Function

3.4.3 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:

$$\sigma(z) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{3.11}$$

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

vector.

3.5 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:

$$C(X,\theta) = \frac{1}{2n}\sum_{x} ||\hat{y}(x) - y(x)||^2 \tag{3.12}$$

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 betwween $y(x)$ 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:

$$\theta^{t+1} = \theta^t - \alpha\frac{\partial C(X,\theta^t)}{\partial \theta} \tag{3.13}$$

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):

$$\frac{\partial C(X,\theta)}{\partial w_{ij}^k} = \frac{1}{N}\sum_x \frac{\partial}{\partial w_{ij}^k}(\frac{1}{2}(\hat{y}_d - y_d)^2) = \frac{1}{N}\sum_x \frac{\partial C_d}{\partial w_{ij}^k} \tag{3.14}$$

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:

$$\Delta w_{ij}^k = -\alpha \frac{\partial C(X,\theta)}{\partial w_{ij}^k} \tag{3.15}$$

3.6 Evaluation Metrics

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 coefficeint. The IoU for a single class is defined as:

$$IoU = \frac{TruePositives}{TruePositives + FalsePositives + FalseNegatives} \tag{3.16}$$

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:

$$Accuracy = \frac{TruePositives + TrueNegatives}{TruePositives + TrueNegatives + FalsePositives + FalseNegatives} \tag{3.17}$$

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:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3.18}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{3.19}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.20}$$

3.7 Potential Challenges and Limitations

1. **Weak Supervision**

   (Schmitt, Prexl, Ebel, Liebel, & Zhu, 2020) 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 occurss 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 eventhough they are using multiple GPUs in parallel.

CHAPTER 4

RESEARCH METHODOLOGY

CHAPTER 5

IMPLEMENTATION PLAN AND INITIAL RESULTS

5.1 blablabla

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio

metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor.

Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

TASK	OCT	NOV	DEC	JAN	FEB
Model Searching					
Data Collection					
Data Pre-processing					
Baseline Model Implementation and Evaluation					
Model Design					
Preliminary Result Analysis					
Report Writing					
Presentation					

Table 5.1: Gantt Chart for FYP 1

TASK	MARCH	APRIL	MAY	JUNE
Model Design				
Model Refinement				
Model Evaluation				
Prototype Implementation				
Report Writing				
Presentation				

Table 5.2: Gantt Chart for FYP 2

CHAPTER 6

CONCLUSION

APPENDIX A

MANUALS, TECHNICAL SPECIFICATIONS, DOCUMENTATIONS, EXAMPLE SCENARIOS

You may want to include appendix in your report. Appendix such as manuals, technical specification, or documentations. You should **NOT** include all your source codes as appendix. Generally source code should be included in CD/DVD and **NOT** in your report.

APPENDIX B

APPENDIX 2: WHAT IS APPENDIX

Appendix is included in your report as it is information that is not essential to explain your findings, but that supports your analysis (especially repetitive or lengthy information), validates your conclusions or pursues a related point should be placed in an appendix (plural appendices). Sometimes excerpts from this supporting information (i.e. part of the data set) will be placed in the body of the report but the complete set of information (i.e. all of the data set) will be included in the appendix. Examples of information that could be included in an appendix include figures/tables/charts/graphs of results, statistics, questionnaires, transcripts of interviews, pictures, lengthy derivations of equations, maps, drawings, letters, specification or data sheets, computer program information.

There is no limit to what can be placed in the appendix providing it is relevant and reference is made to it in the report. The appendix is not a catch net for all the semi-interesting or related information you have gathered through your research for your report: the information included in the appendix must bear directly relate to the research problem or the report's purpose. It must be a useful tool for the reader

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.

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

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

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

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

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

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

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.

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. (2021). 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

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

NOTES

PUBLICATION LIST