FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Detection of cloud coverage using advanced deep learning methods
Student:	Jaroslav Hradil
Supervisor:	Mgr. Petr Šimánek
Study program:	Informatics
Branch / specialization:	Knowledge Engineering
Department:	Department of Applied Mathematics
Validity:	until the end of summer semester 2024/2025

Instructions

Cloud detection is crucial in the preprocessing of optical satellite imagery because it directly impacts the quality and accuracy of subsequent remote sensing applications, including weather forecasting, climate monitoring, and environmental analysis. Accurate cloud detection enables the effective differentiation between clouds and other high-reflectance ground features, ensuring that the data used for analysis is reliable and precise. This process is essential for enhancing the accuracy of earth observation data interpretations, which supports a wide range of scientific research and practical applications. By improving cloud detection methods, researchers and practitioners can significantly reduce misclassification errors and enhance the utility of satellite imagery for studying and monitoring the Earth's surface and atmosphere, contributing to better-informed decision-making in areas like agriculture, disaster management, and climate change mitigation.

The tasks for the student:
1/ Review the literature. Review the existing cloud detection methods, focusing on deep learning approaches. Mention their advantages and disadvantages, especially in terms of handling spatial information and cloud boundary refinement.
2/ Choose and study in detail two methods, one standard (e.g. UNET) and one advanced (e.g. CDUNet). Understand how the advanced components work together to improve cloud detection accuracy.
3/ Choose, acquire, understand, and preprocess a cloud dataset: Collect dataset for

---

Electronically approved by Ing. Magda Friedjungová, Ph.D. on 8 February 2024 in Prague.

training and testing the model, such as cloud datasets and SPARCS datasets. Prepare the data by performing any necessary preprocessing steps to ensure it is suitable for input into the deep learning model. The student will choose publicly available dataset.
4/ Implement and train the model: Implement the chosen models using the pytorch deep learning framework. Train the model on the prepared datasets, carefully adjusting parameters and settings to optimize performance. Utilize techniques like cross-validation to ensure the model's generalizability.
5/ Evaluate the results: Evaluate the model's performance using relevant metrics, such as segmentation accuracy. Compare the results between the methods. Analyze some instances of misclassification or errors to understand the model's limitations.
6/ Document the process: Document the research process, model architecture, training procedure, evaluation results, and comparison with existing methods in detail. Analyze the implications of the findings for cloud detection and remote sensing.

Bachelor's thesis

DETECTION OF CLOUD COVERAGE USING ADVANCED DEEP LEARNING METHODS

Jaroslav Hradil

Faculty of Information Technology
Department of Applied Mathematics
Supervisor: Mgr. Petr Šimánek
May 16, 2024

Czech Technical University in Prague
Faculty of Information Technology

Citation of this thesis: Hradil Jaroslav. *Detection of cloud coverage using advanced deep learning methods.* Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2024.

Contents

List of Figures

List of Tables

I would like express my sincere gratitude to my supervisor, Mgr. Petr Šimánek, for the provided expertise, feedback and patience, as well as the Data Science Laboratory at Faculty of Information Technology for providing me with the computational resources necessary to carry out this research. I would also like to thank my family, my girlfriend and my classmates for their unwavering support.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis.

I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Czech Technical University in Prague has the right to conclude a licence agreement on the utilization of this thesis as a school work pursuant of Section 60 (1) of the Act.

In Prague on May 16, 2024

Abstract

Accurate cloud detection is a key component of the preprocessing needed to prepare optical satellite imagery for various remote sensing applications. In recent years, numerous deep learning methods for cloud detection, such as convolutional neural networks or visual transformer, have been introduced, and this thesis aims to compare them.

In the thesis, current literature on cloud detection methods is reviewed, and the deep learning architectures U-Net and SegFormer are investigated in detail. They are implemented in the PyTorch framework, trained on the 38-Cloud dataset, which consists of Landsat-8 satellite imagery. Their performance is evaluated using accuracy, intersection over union (IoU) and F1 score.

The results show that SegFormer slightly outperforms the U-Net based architecture in terms of accuracy, while U-Net slightly outperforms SegFormer in the IoU and F1 score. Moreover, the U-Net based architecture has much shorter training time. This proves that even though being relatively simple and old, U-Net is still relevant and efficient architecture today.

The findings of this study can help other professionals choose the most suitable deep learning architecture for their use-case and provide a solid basis for further research.

Keywords cloud detection, optical satellite imagery, remote sensing, deep learning, U-Net, SegFormer

Abstrakt

Přesná detekce mraků je klíčovou součástí předzpracování satelitních snímků pro různé aplikace dálkového průzkumu Země. V posledních letech byla navržena řada metod hlubokého učení pro detekci mraků, jako jsou například konvoluční neuronové sítě nebo vision transformery, a cílem této práce je tyto metody porovnat.

V této práci je provedena rešerše aktuální literatury zabývající se detekcí mraků a jsou podrobně zkoumány architektury U-Net a SegFormer. Ty byly implementovány ve frameworku PyTorch a trénovany na datasetu 38-Cloud, který obsahuje snímky ze satelitu Landsat-8. Výkon těchto dvou metod je pak hodnocen pomocí metrik přesnosti, Intersection over Union (IoU) a F1 skóre.

Výsledky ukazují, že SegFormer mírně překonává architekturu založenou na U-Netu v přesnosti, zatímco U-Net mírně překonává SegForer v metrikách IoU a F1. U-Net navíc potřebuje výrazně méně času k natrénování. To dokazuje, že i přesto, že je U-Net poměrně jednoduchý a starý, stále zůstává relevantní a efektivní architekturou hlubokého učení.

Zjištění této práce mohou pomoci ostatním odborníkům zvolit vhodný model hlubokého učení pro jejich aplikaci a poskytují dobrý základ pro další výzkum.

Klíčová slova detekce mraků, optické satelitní snímky, dálkový průzkum Země, hluboké učení, U-Net, SegFormer

List of Abbreviations

CNN	Convolutional Neural Network
IoU	Intersection over Union
MLP	Multilayer Perceptron
NLP	Natural Language Processing
ReLU	Rectified Linear Unit
ViT	Vision Transformer

Introduction

Every day, almost every person on our planet relies on weather forecasts. From making simple decisions about going for a walk, to farmers planning their spraying schedules, to managing extreme weather and disaster management. The quality of weather forecasts, as well as the quality of many other remote sensing applications such as climate monitoring, depends on the quality of preprocessing of optical satellite imagery, where cloud detection is a crucial component.

The task of cloud detection in optical satellite imagery is non-trivial for two main reasons. Firstly, clouds have variable characteristics and can vary widely in terms of their shape, size, texture, or opacity. Secondly, they might be highly similar to other bright features in satellite images, such as snow, ice, or reflective surfaces like rooftops or water surfaces.

To tackle these challenges, various novel deep learning-based methods, such as the U-Net architecture-based RS-Net [1] or transformer architecture-based Cloudformer [2], were introduced in recent years. However, the last paper reimplementing and comparing the performance of cloud detection methods is from 2020 [3] and compares only traditional algorithms based mainly on thresholding.

The aim of this thesis is therefore to conduct a review of modern methods based on deep learning, to reimplement the two most relevant deep learning-based architectures, to select suitable datasets, and to test and compare their performance on these datasets. The objectives of the work are described in more detail in the following subsection.

Objectives

This bachelor's thesis aims to review the recently published cloud detection methods, focusing on deep learning approaches, and to reimplement the most relevant ones and compare their performance using the most relevant datasets.

The main goal of the research part of this bachelor's thesis is to review literature and research the existing cloud detection methods. The focus will be on deep learning approaches and their respective advantages and disadvantages, especially in terms of handling spatial information and cloud boundary refinement, and to study in detail at least two of them, one based on the U-Net and one based on the transformer architecture. These will be reimplemented in the practical part. The subsequent goal of the research part is to review the existing cloud detection datasets as well as suitable evaluation metrics and choose the most relevant ones for the experiments.

The key objective of the practical part of this bachelor's thesis is to reimplement the chosen cloud detection methods and evaluate their performance on relevant datasets. That consists of setting up the data loading pipeline, reimplementing the methods, setting up the model training and hyperparameter tuning pipeline and evaluating the models using the chosen metrics. Finally, the obtained results will be carefully examined and we will attempt to explain the limitations of the chosen methods.

Structure

This thesis is divided into seven chapters. The first chapter provides an introduction to the domain of cloud detection, emphasizing its importance and the challenges it presents. Chapter 2 gives a brief overview of satellites, their key characteristics, and the most commonly used satellite imagery datasets for cloud detection. Chapter 3 explains selected concepts of deep learning that the reader needs to understand the subsequent chapters. Chapter 4 comprehensively reviews the related work on cloud detection, and Chapter 5 describes the chosen methods for reimplementation and experiments in detail. Chapter 6 explains the implementation, from the selected datasets, through the hardware and software used, to the training procedure. Finally, Chapter 7 discusses the results of the performed experiments.

Chapter 1
Cloud Detection

First, we will define the task of cloud detection in satellite imagery. The task of object detection usually refers to identifying objects within an image and defining a bounding box around each object of interest. However, in the domain of satellite imagery, it usually refers to delineating the exact boundaries of clouds at the pixel level and providing a precise mask that highlights the shape of the cloud—a task that is otherwise known as segmentation. Given that this designation is more commonly used in scientific papers, we will adhere to it as well.

1.1 Importance

Various remote sensing applications across different domains rely on cloud-free satellite imagery. To obtain cloud-free satellite imagery, the cloud detection is an essential part of the satellite imagery preprocessing. Examples of such domains include, but are not limited to:

- agriculture: crop health monitoring [4] or yield estimation [5, 6]
- disaster response and management: floods monitoring [7, 8], hurricane damages mapping [9], earthquakes and landslides detection [10] or forest fires monitoring [11, 12]
- environmental monitoring: deforestation [13, 14], desertification [15] or size changes of water bodies [16]
- urban development monitoring [17].

However, according to [18], 67-68 percent of satellite imagery is on average obstructed by clouds. Accurate cloud detection is therefore an essential part of the satellite imagery preprocessing.

1.2 Challenges

As mentioned in the Introduction, the task of cloud detection in satellite imagery is challenging for various reasons:

- Variable cloud characteristics: Clouds can vary widely in terms of shape, size, texture, and opacity, making it difficult to design an algorithm that reliably detects all the possible variations of clouds across different types of satellite imagery.

- Similarity to other features: Clouds can often look similar to other bright or white features in satellite imagery, such as snow, ice, or reflective surfaces like metal rooftops or water surfaces. This similarity often leads to misclassification, where clouds are either not detected or non-cloud elements are incorrectly labeled as clouds.
- Sensor limitations and variations: Different satellites have different sensors with varying resolutions, spectral bands, and sensitivities. These differences make it challenging to design an algorithm capable of detecting clouds consistently across satellite imagery from different sources.
- Complexity of dataset preparation: Researchers must first manually select a sufficiently large and representative dataset of satellite images and then manually label it using annotation software, such as eCognition and commercial MODIS, which is a process both time-consuming and susceptible to human error. According to [19], the inter-annotator agreement was only 96 % when independent interpreter annotated one of the most used datasets, proving the inherent susceptibility to human subjectivity.
- Computational complexity: The datasets for training deep learning-based methods for cloud detection in satellite imagery are large (see next section) which results in long training time even when being run on high end GPUs [20].

Chapter 2

Satellite Imagery

Satellite imagery refers to images of Earth collected by imaging satellites orbiting the Earth, operated by various government agencies and businesses. This chapter aims to briefly introduce the characteristics of satellite imagery and the most commonly used datasets for cloud detection.

2.1 Key Characteristics

Satellite images are primarily defined by the following three main characteristics, which may vary among satellites with different sensors:

- Spectral resolution: Satellites capture data at various wavelengths across the electromagnetic spectrum. Different spectral bands are sensitive to different cloud properties, such as water vapor content and cloud thickness, which can significantly enhance the model's ability to segment clouds accurately.
- Spatial resolution: Refers to the size of one pixel on the ground. High spatial resolution helps in distinguishing fine details within cloud formations, which is essential for accurate segmentation by deep learning models. Sensors for various spectral bands might have different spatial resolution.
- Temporal resolution: Also called the revisit time, it refers to the frequency at which satellites capture imagery of the same area. Higher temporal resolution allows for better tracking of cloud movement and development over time, and some of the proposed methods utilize it for cloud segmentation. [21, 22, 23]

2.2 Datasets

According to the UCS Satellite Database [24], there were 7,560 satellites orbiting Earth as of 2023. Of these, approximately 1,260 are designated for Earth observation and have some capability of satellite imaging.

Based on the analysis of [25], the satellite imagery most commonly used for cloud detection datasets comes from the Landsat Program by National Aeronautics and Space Administration (NASA), the Sentinel Program by European Space Agency (ESA), and the Gaofen Program by China National Space Administration (CNSA).

	Spectral Band	Wavelength range (µm)	Resolution (m)
1	Blue	0.45–0.52	30
2	Green	0.52–0.60	30
3	Red	0.63–0.69	30
4	Near-infrared	0.77–0.90	30
5	Short-wave Infrared	1.55–1.75	30
6	Thermal	10.40–12.50	60
7	Mid-Infrared	2.08–2.35	30
8	Panchromatic	0.52–0.90	15

Table 2.1 Spectral and spatial resolutions of Landsat-7 [21]

2.2.1 Landsat

The Landsat program, launched in 1972 by NASA, is the longest-running program for Earth's satellite imagery acquisition. Since its inception, a total of nine satellites, named Landsat 1 through Landsat 9, have been launched within this program, with the most recent being launched in 2021 [26]. The datasets for cloud detection most commonly utilize the satellite imagery from the Landsat-7 and Landsat-8 satellites.

2.2.1.1 Landsat-7

Launched in 1999, Landsat-7 captures high-resolution imagery while orbiting at an altitude of 705 kilometers [21]. The revisit time, which is the period in which it captures the same land under the same viewing angle, is 16 days. The spectral and spatial resolutions are summed up in Table 2.1.

The most used dataset based on its imagery is the L7-Irish dataset [27, 28]. It contains 206 Landsat-7 scenes that are evenly divided among 9 different biomes around the Earth, with manually generated cloud masks, which distinguish three classes: cloud, thin cloud, and clear. Additionally, 45 scenes are labeled for cloud shadows too. The scenes were acquired between June 2000 and December 2001.

2.2.1.2 Landsat-8

Launched in 2013 as the successor to Landsat-7, Landsat-8 is equipped with two advanced sensors, the Operational Land Imager (OLI, bands 1-9) and the Thermal Infrared Sensor (TIRS, bands 10-11), improving both the spectral and spatial resolution of the imagery and among others allowing researchers to detect clouds more accurately [22]. The revisit time of Landsat-8 is 16 days and the spectral and spatial resolutions are summarized in Table 2.2.

Numerous popular datasets for cloud detection utilize the Landsat-8 imagery:

- L8-Biome [29, 28]: Consists of 96 Landsat-8 scenes evenly divided between eight different biomes around the Earth with manually generated cloud masks, which distinguish three classes: cloud, thin cloud, and clear. Additionally, 32 scenes are labeled for cloud shadows as well. The scenes were acquired between April 2013 and October 2014 with the resolution of 8000×8000 pixels.
- L8-SPARCS [30]: Contains 80 Landsat-8 scenes with manually generated masks, that distinguish six different classes: cloud, cloud shadow, snow/ice, water, flooded and clear. The size of each scene is 1000×1000 pixels, making this dataset much smaller than the L8-Biome dataset.

	Spectral Band	Wavelength range (µm)	Resolution
1	Coastal Aerosol	0.43–0.45	30
2	Blue	0.45–0.51	30
3	Green	0.53–0.59	30
4	Red	0.64–0.67	30
5	Near-infrared	0.85–0.88	30
6	Short-wave Infrared 1	1.57–1.65	30
7	Short-wave Infrared 2	2.11–2.29	30
8	Panchromatic	0.50–0.68	15
9	Cirrus	1.36–1.38	30
10	Thermal Infrared 1	10.6–11.19	100
11	Thermal Infrared 2	11.5–12.51	100

Table 2.2 Spectral and spatial resolutions of Landsat-8 [22]

- 38-Cloud [31, 32]: Was created by dividing 38 Landsat-8 scenes into 384 × 384 patches and constitutes of 8,400 such patches for training and 9,201 patches for testing. The number of bands is reduced to four, keeping only red, green, blue and near-infrared. According to authors, using all the bands might slightly improve the accuracy, but these bands are provided by most satellites, making the algorithm trained using this dataset more universal.
- 95-Cloud [31, 32, 33]: Is an extension of the 38-Cloud dataset. The number of training scenes is increased to 95, resulting in 34,701 patches, while the test set remains the same. It is worth mentioning that most scenes are from North America.

2.2.2 Sentinel

The Sentinel missions are a key part of the Copernicus program, managed by the European Commission and aims to provide comprehensive datasets for Earth observation. As of 2024, there have been six missions named Sentinel-1 through Sentinel-6, each focusing on collecting different observations and each based on a constellation of two satellites to provide redundancy and increased coverage [34].

2.2.2.1 Sentinel 2

The datasets for cloud detection utilize observations from the Sentinel-2, which focuses on high-resolution multi-spectral imaging for land monitoring [23]. It has 13 high-resolution spectral bands allowing to improve the precision of remote sensing applications, which are summed up in the Table 2.3. The revisit time under the same viewing angle is 10 days when using only one satellite, and 5 days when using imagery from both Sentinel-2 satellites.

Numerous commonly used datasets for cloud detection utilize the Sentinel-2 imagery:

- S2-Hollstein [35]: Is a pixel-wise dataset that consists of 5,647,725 labeled pixels from various regions around the Earth. The labels distinguish six classes: cloud, cirrus, snow/ice, shadow, water and clear sky.
- S2-BaetensHagolle [36, 37]: Contains 38 Sentinel-2 scenes from ten different locations around Earth to ensure diversity that were manually labeled for six categories: land, water, snow, high clouds, low clouds, and cloud shadows. The scenes were acquired during 2017 and 2018.
- WHUS2-CD+ [38, 39]: Consists of 36 Sentinel-2 scenes from various areas distributed around mainland China. The scenes are labeled at 10-meter resolution, and the labels distinguish only classes: clear and cloud.

	Spectral Band	Central wavelength (µm)	Bandwith (nm)	Resolution
1	Coastal aerosol	0.443	21	60
2	Blue	0.490	66	10
3	Green	0.560	36	10
4	Red	0.665	31	10
5	Vegetation Red Edge 1	0.705	15	20
6	Vegetation Red Edge 2	0.740	15	20
7	Vegetation Red Edge 3	0.783	20	20
8	Near-infrared	0.842	106	10
9	Narrow Near-infrared	0.865	21	20
10	Water vapor	0.945	20	60
11	Short-wave Infrared 1	1.375	31	60
12	Short-wave Infrared 2	1.610	91	20
13	Short-wave Infrared 3	2.190	175	20

Table 2.3 Spectral and spatial resolutions of Sentinel-2 [23]

2.2.3 Others

- Gaofen: The Gaofan series is a fleet of high-resolution satellites developed and operated by China National Space Administration (CNSA) for both civilian and military purposes. There are various datasets based on its imagery, such as GF1-WHU [40], GF1MS-WHU and GF2MS-WHU [41], or AIR-CD [42], used almost solely by Chinese research teams. They differ in the number of scenes, covered regions, distinguished classes, and resolutions.
- HRC-WHU [43]: This dataset comprises 150 high-resolution scenes (0.5 m to 15 m spatial resolution, 3 RGB channels) from Google Earth that were labeled for clouds. The scenes are globally distributed across five main land-cover types: water, vegetation, urban, snow/ice and barren.

Chapter 3
Deep Learning Background

This bachelor thesis covers the application of deep learning methods for the cloud detection task. This chapter describes selected concepts in neural networks and deep learning necessary to understand this thesis.

It is assumed that the reader is familiar with basic neural networks concepts, such as perceptron, activation functions, loss function, backpropagation, gradient descent and optimizers like Adam, as well as with basic model validation techniques, particularly the train validation test split and cross-validation.

3.1 Convolutional Neural Networks

A convolutional neural network (CNN) is a type of neural network that is designed to learn spatial features within grids of data [44]. It is widely used for tasks involving pattern recognition in images, such as object detection, image classification, or image segmentation.

In general, CNNs consist of several building blocks, which are the convolutional layers, pooling layers, and fully connected layers. A block of one or multiple convolutional layers, each immediately followed by an activation function to introduce nonlinearity, with one pooling layer at the end, is repeated multiple times to perform the feature extraction. The obtained feature maps are then passed to the fully-connected layers, with the last fully-connected layer computing the output, such as predicted label, as shown in Figure 3.1.

Thanks to the pooling layers, the feature maps that are passed to the fully-connected layers have much smaller dimensions than the original input data. That greatly reduces the number of parameters of the fully-connected layers, makes the computations more efficient, and allows the CNNs to process high-resolution data.

3.1.1 Convolution

The mathematical operation of convolution is an essential component of each convolutional layer. The convolution is performed using a kernel, which is a two-dimensional array of weights of dimensions $k \times k$, that is slided across a two-dimensional array of dimensions $H \times W$ representing an image to calculate the dot product of the kernel and the part of the image currently overlapping with the kernel, where k is the kernel size and H and W the height and width of the input image.

This produces a two-dimensional output, usually called feature or activation map, and the network iteratively learns what kernels are activated when sliding over a particular visual feature, such as specifically oriented edge or a specific color [46]. Hence the kernels are often called

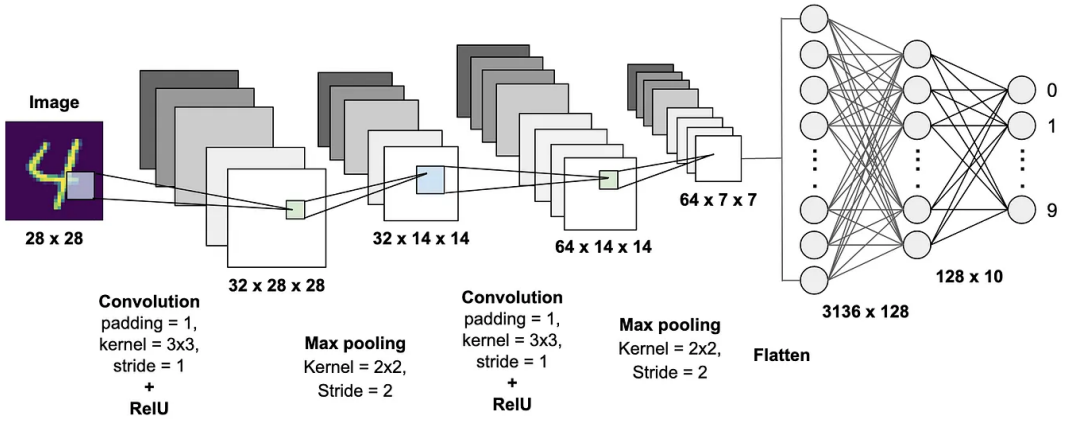


Figure 3.1 An example of CNN architecture. Image from [45]

learnable, as their weights are adjusted during the training process. The calculation of feature map is illustrated in Figure 3.2.

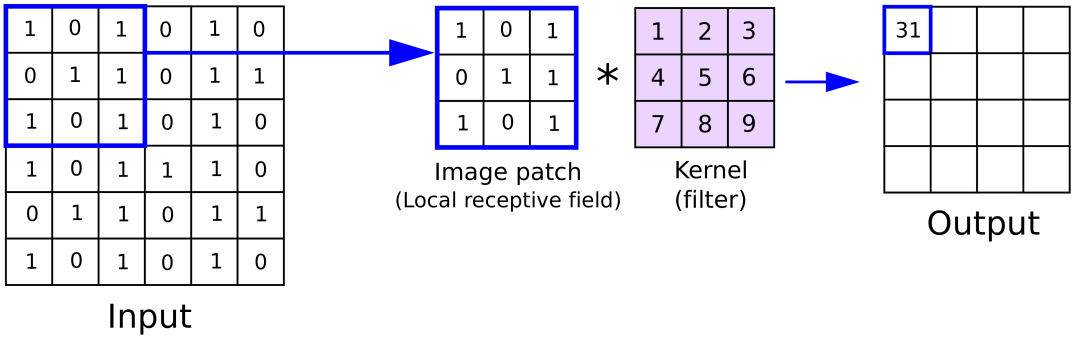


Figure 3.2 Example of convolution with 3×3 kernel, stride of 2 and no padding. Image from [47]

Usually, there is a need to detect more features or to extract features from images with more than one color channel. In such cases, a collection of multiple kernels is used, which is called a filter, or a learnable filter. From now on, kernel and filter will be used interchangeably.

Besides the kernel size k, convolution has two additional parameters, stride and padding. Stride represents the size of the step that the kernel makes when moving to the next position. Padding refers to adding extra pixels around the borders of the input to preserve information near the edges. The final dimensions of the output feature map can then be computed as

$$(\frac{H + 2p + k}{s} + 1, \frac{W + 2p + k}{s} + 1, n_k), \tag{3.1}$$

where H and W are the dimensions of the input image or feature map, p is the size of the padding, k is the size of the kernel, s is the stride and n_k is the number of kernels in the filter.

3.1.2 Pooling

Pooling layers are used in CNNs to progressively reduce the spatial dimensions of images and feature maps, while retaining essential information to represent the input data efficiently and to reduce the computational complexity of the following layers. Similarly to the convolution, it can be imagined as a kernel of size $k \times k$ sliding across the input image or feature map and applying an aggregation function to the values within the kernel. The most commonly used aggregation functions are maximum and average, which have edge extracting and smoothing effect, respectively. Also similarly to the convolution, their configurable hyperparameters are the kernel size and stride [46]. On the other hand, unlike convolution, they have no trainable weights. This process is also known as downsampling and is illustrated in Figure 3.3.

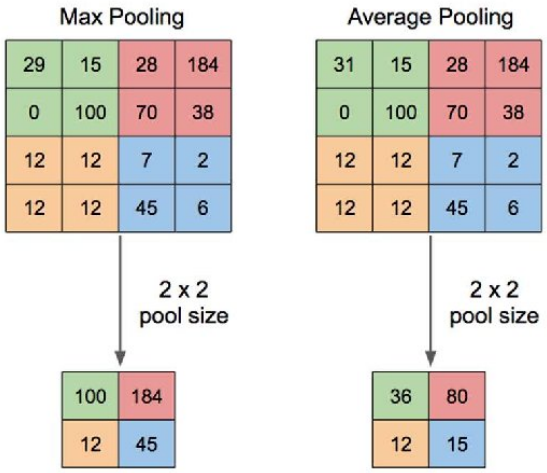


Figure 3.3 Max-pooling and avergage-pooling examples. Image from [48]

3.1.3 Transposed Convolution

This operation is not used in traditional CNNs, but it is necessary to explain as it is used in more advanced methods introduced later in the thesis.

The transposed convolution is similar to a regular convolution, except that it performs the convolution operation in the opposite direction. While convolution reduces the input image or feature map dimensions by sliding a kernel across it, transposed convolution increases the spatial dimensions of the input feature map by applying a kernel that effectively projects the input feature map onto a larger grid, as demonstrated in Figure 3.4. This is often called upsampling, and the size of the output is again controlled by the kernel size, stride and padding. Similarly to convolution, the weights in the kernel are learnable.

It is worth mentioning that transposed convolution should not be confused with deconvolution. If the output of convolution is deconvoluted, the result would be the same as the original input for the convolution. If the output of convolution is passed through a transposed convolution, the result is not the same as the original input for the convolution, it only has the same dimensions as it has been upsampled.

3.2 Transformers

Transformer is a deep learning architecture, first introduced for natural language processing (NLP) [50], but later successfully extended to image processing tasks [51]. The architecture

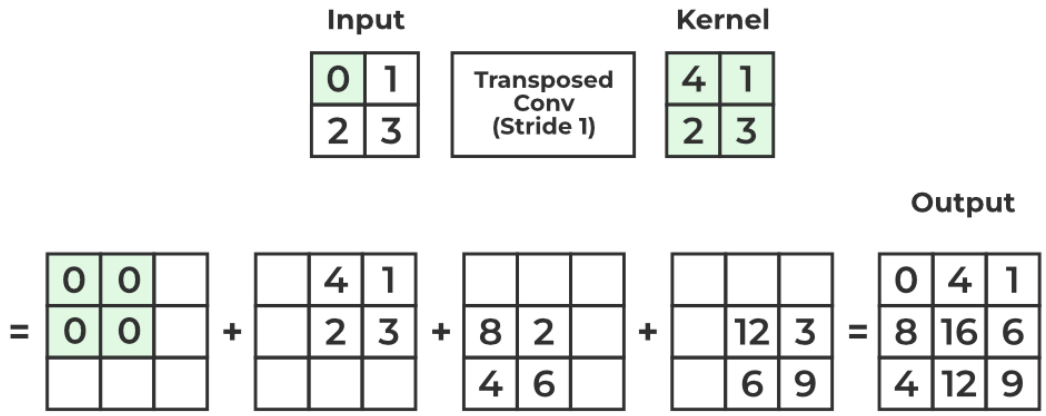


Figure 3.4 Example of transposed convolution with 2×2 kernel, stride of 1 and no padding. Image from [49]

consists of an encoder that encodes the input sequence into a more efficient representation that captures the relationships between different parts of the input, and a decoder that decodes it and generates the output sequence, as shown in the Figure 3.5. Both components rely on a mechanism known as self-attention to process the sequential data, such as natural language or imagery. However, unlike other architectures that process the data sequentially, transformers can be parallelizable.

3.2.1 Attention

The attention mechanism, originally introduced in [50], simulates how human attention works by assigning different levels of importance to different parts of the sequential input, be it words in natural language or patches of an image.

It allows the model to focus on specific parts of the input at each decoding step by enabling the decoder to periodically revisit the input sequence. That is done by the encoder producing a representation of the same length as the input sequence, while the decoder receives a weighted vector of these representations where the weights decide how much attention should be given to each input token. This way, the model focus only to the parts of the input that are likely to be relevant for the current prediction.

The attention usually used in transformers is known as scaled dot-product attention and is defined as

$$\text{Attention}(Q, K, V) = \text{Softmax}(\frac{QK^{\text{T}}}{\sqrt{d_k}})V. \tag{3.2}$$

The inputs to the attention function are the queries, keys, and values, which are all vectors derived from the same input sequence through linear transformations. For each position in the input sequence, a query vector is computed and then used to score its relationship with every key vector from all positions in the sequence. The scoring is done through a dot product, which is then scaled down by the square root of the dimension of the key vector. The scaling helps to stabilize the gradients during training, as it prevents the softmax function from having extremely small gradients in cases when the dot product is large. The softmax function then converts them

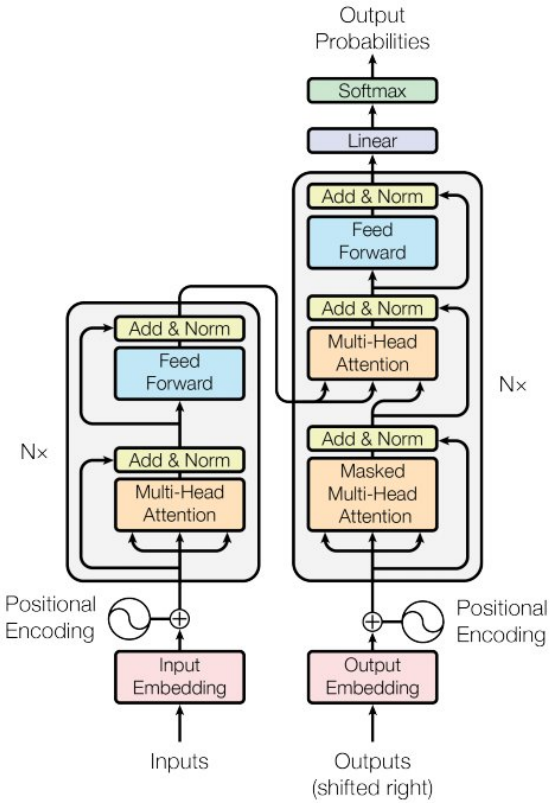


Figure 3.5 The architecture of transformer. Image from [50]

into probabilities, which determine how much each part of the input sequence contributes to the output at each position. The probabilities add up to 1. Finally, the output of softmax function is multiplied by the value vectors, which aggregates the information from different parts of the input sequence, weighted by their relevance to the query, into a single output vector for each position.

To sum it up, each output vector of the attention mechanism is a weighted sum of the value vectors, where the weights are the softmax probabilities. The output then serves as an input for the next layer in the transformer.

3.2.2 Multi-Head Attention

This mechanism further improves the ability of transformers to focus on different parts of the input sequence. It consists of multiple attention heads, each performing the scaled dot-product attention independently with different linear transformations of the input vectors. The multiple outputs are then concatenated and linearly transformed into the expected dimension. That allows the transformer to capture more complex relationships within the input sequence.

3.3 Evaluation Metrics

Measuring the performance of models and the quality of their predictions is a crucial part of machine learning as the evaluation metrics allow us to compare the results of various architectures. This section introduces metrics relevant for evaluating the task of image segmentation, which is essentially pixel-wise classification.

First, we will introduce the confusion matrix, which is a tabular summarization of the performance of a classification model. The confusion matrix consists of four values:

- True Positives (TP): number of positive samples correctly classified as positive
- False Positives (FP): number of negative samples incorrectly classified as positive
- True Negatives (TN): number of negative samples correctly classified as negative
- False Negatives (FN): number of positive samples incorrectly classified as negative

It compares the actual labels with the predicted labels, provides insight into the types of errors made by the model, and its values are also necessary for calculating other metrics. The confusion matrix is visualized in Figure 3.6.

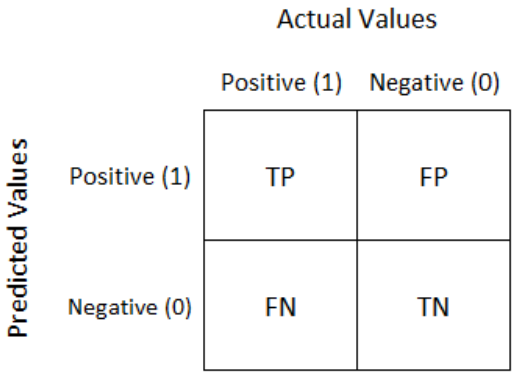


Figure 3.6 Confusion matrix. Image from [52]

3.3.1 Accuracy

Pixel accuracy is the proportion of correctly classified pixels to the total number of pixels. It is easy to understand metric, however, it might be misleading when handling imbalanced classes. For example, in a scenario where one of the classes is much more frequent than the other, high accuracy can be achieved by predicting the dominant class. It is defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{3.3}$$

Accuracy can take a value between zero and one, where zero indicates that no correct predictions were made, and 1 indicates perfect prediction.

3.3.2 Intersection over Union (IoU)

Intersection over union, also known as the Jaccard index, measures the overlap between the predicted mask and the ground truth. As the name suggests, it is the ratio of the intersection of the mask with the ground truth to the union of the mask with the ground truth. Compared to accuracy, it gives better measurement of how much the predicted mask overlaps with the ground truth. However, it can penalize predictions with small mismatches quite heavily, especially when the segmentation mask is relatively small. It can be defined as

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{3.4}$$

or alternatively as

$$\text{IoU} = \frac{\|A \cap B\|}{\|A \cup B\|}, \tag{3.5}$$

where A is the predicted mask and B is the ground truth. IoU can take a value between one and zero, where zero means no overlap between the predicted and actual mask, while 1 means perfect overlap.

3.3.3 F1 score

The F1 score, also known as Dice score or Dice coefficient, is similar to IoU, but emphasizes the overlap between the predicted mask and the ground truth more. The sensitivity to the overlap is crucial in fields such as medical image segmentation, where accurate boundary detection is vital. Although less penalizing than IoU, the F1 score can still be influenced by small mismatches. It is defined as

$$\text{F1} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3.6}$$

or alternatively as

$$\text{F1} = \frac{2\|A \cap B\|}{\|A \cup B\|}, \tag{3.7}$$

where A is the predicted mask and B is the ground truth. The F1 score can take a value between one and zero, where zero indicates completely incorrect prediction with either precision or recall being zero, while one indicates completely correct prediction with both precision and recall perfect.

Chapter 4
Related Work

The majority of the research conducted to date can be divided into two categories: traditional methods, such as physical rule-based thresholding algorithms, and deep learning-based methods, such as convolutional neural networks or transformers, which are the focus of this bachelor's thesis.

4.1 Traditional Methods

The traditional cloud detection methods can be divided into three main categories: threshold-based methods, temporal change-based methods, and machine learning (excluding deep learning) methods.

4.1.1 Threshold-Based Methods

These methods are usually constructed based on the physical features of clouds that can be extracted by thresholding the spectral bands of the satellite imagery, such as high brightness, low temperature, and high elevation [53]. The thresholds can be fixed or dynamic and are usually determined through initial manual judgment and then automatically and systematically optimized [54]. These methods are widely used in cloud detection due to their simplicity and relatively fast inference.

For example, the Automatic Cloud Cover Assessment (ACCA) algorithm [55] contains 26 thresholding conditions based on which it segments the clouds in the images and has been used as the official method to evaluate the quality of the Landsat-7 satellite imagery. Another relevant examples of this approach are FMask [56, 54] or Sen2Cor [57], that are used to generate cloud masks that are distributed together with Landsat 4-8 and Sentinel-2 imagery, respectively.

However, this approach has some limitations. Most importantly, this approach rely heavily on the contrast between clouds and the earth surface and cannot cope with complex surfaces well due to the phenomenon of homospectral foreign matter [58]. That usually results in misclassification of other bright and highly reflective features such as snow, ice, water surfaces or large metal rooftops as clouds.

4.1.2 Temporal Change-Based Methods

This approach treats the cloud detection task as a change detection task and achieves it by detecting sudden changes in image time series. Pixels with abrupt increase in reflectance are more likely to be clouds, while on the other hand pixels with abrupt decrease in reflectance

are more likely to be land cover. The cloud-free land is assumed to change relatively smoothly compared to the sudden changes in reflectance caused by clouds.

For example, [59] designed a simple method called multi-temporal cloud detection (MTCD) that detects clouds by detecting steep increase of reflectance of a pixel in the blue band and verifying that the increase correlates with the neighbouring pixels.

[60] proposed Tmask, which combines the threshold-based Fmask method and multi-temporal imagery for cloud detection. The method first applies the Fmask algorithm to obtain the cloud masks and then models mask changes based on multi-temporal imagery to detect final cloud masks, significantly improving performance compared to the mono-temporal Fmask method.

Another relevant method based on temporal changes and used mostly for the European Sentinel-2 imagery is the MACCS-ATCOR Joint Algorithm (MAJA) [61], which is a joint project of French and German research teams and combines temporal change-based and threshold-based approach: it detects low clouds using their spectral differences relative to the most recent cloud-free reference image, but high clouds are detected using a single-date imagery.

In general, temporal change-based cloud detection methods benefit from the use of multi-temporal information and usually outperform the mono-temporal threshold-based methods. However, this approach requires either selection of a cloud-free reference image or construction of a cloud-free reference image by combining multi-temporal images over a short period of time, but both is difficult to impossible in areas with rapid land cover changes.

4.1.3 Machine Learning Methods

These methods treat the cloud detection task as an image classification task by constructing and optimising a machine learning model that classifies each pixel for either clear or clouds. For this approach, it is essential to select sufficiently representative training data and use appropriate strategies to optimise the model parameters so that the model can generalise for unseen satellite imagery.

Traditional machine learning algorithms, such as classical Bayesian [62], random forests [62, 63, 64, 65], XGBoost [66, 67], support vector machines (SVM) [68, 69, 70], and simple neural networks [71, 72], have been widely used for cloud detection in satellite imagery. Furthermore, some papers proposed ensembles combining these algorithms or domain-specific improvements of these methods [73, 74].

Summarily, with sufficient amount of sufficiently diverse training data, the machine learning methods can achieve better cloud detection results than traditional methods, but they still rely almost solely on spectral features and have only very limited capability to exploit the spatial features.

4.2 Modern Methods

With the acceleration of GPU, deep learning approaches have witnessed significant breakthroughs in recent years and have been successfully applied in various domains, including remote sensing [75, 76], as they have powerful feature extraction capabilities and are able of learning from both spatial and spectral features.

4.2.1 Convolutional Neural Networks

Inspired by the great performance of CNNs such as FCNs [77], SegNet [78], U-Net [79] or DeepLabV3+ [80] for semantic segmentation tasks in other fields, numerous papers adopted these methods for cloud detection.

For example, [81, 82] proposed methods for cloud detection based on the FCNs called Cloud-FCN and Cloud-Net respectively, [83] utilised the SegNet architecture for cloud detection and

[1] introduced cloud detection methods based on the U-Net architecture. [84, 85] extended the U-Net based architecture with spatial attention and [86] investigated thoroughly the effect of U-Net parameters on its receptive field.

[87, 88] proposed deep learning-based cloud detection algorithms exploiting multi-scale features. In the context of deep learning for cloud detection in satellite imagery, multi-scale features refer to the extraction and usage of features at multiple resolutions to effectively detect clouds. According to the results and analysis of [87, 88], this approach is beneficial for two main reasons. Firstly clouds can appear in various sizes and shapes, and some types of cloud coverage can have distinct textures and boundaries that might be more or less visible at different scales. Secondly, different satellites are orbiting at different heights, meaning that the same type of clouds look differently in satellite imagery from different satellite, and using multi-scale features may allow the detection algorithm to be used for imagery from different satellites.

[89] introduced weakly supervised deep learning-based cloud detection algorithm (WDCD) that reduces the labour needed to annotate the training data by needing less annotated data to train the model and [90] proposed a deformable contextual and boundary weighted network (DABNet) that improves the detection of clouds at their boundaries.

Keeping in view the high computational complexity and a large number of parameters of deep convolutional neural networks, attention was also paid to lightweight models for devices with limited computational resources, such as micro-satellites. For example [91] developed a lightweight U-Net based model and [92] designed a lightweight model based on DeeplabV3+ architecture with attention for Landsat-9 imagery, while [93] proposed a lightweight network fusing multi-scale spectral and spatial features tailored for Sentinel-2 imagery.

4.2.2 Transformers

The CNN-based methods mentioned in the previous chapter achieve very good results, however, their receptive field is inherently local due to convolution operations, which makes it challenging to capture global semantic features and long-range dependencies.

In other domains, this limitation of CNNs for semantic segmentation tasks was successfully tackled by using Vision Transformer (ViT) models recently, such as SETR [94], SegFormer [95], MaskFormer [96], Mask2Former [97], Segmenter [98] and lightweight MobileViT [99]. They use self-attention mechanism which allows them to model both local and global features of an image. For large inputs such as satellite imagery data, transformers can be more efficient since they can be parallelized. They also proved efficient in other remote sensing applications [100].

Compared to CNNs, much less research has been done on utilisation of transformers for cloud detection tasks. [101] introduced a transformer-based model for Sentinel-2 imagery over India. [2] proposed Cloudformer, a model with architecture that combines CNNs with transformer, and CloudformerV2 [102] and CloudformerV3 [103] further improves the accuracy and reduces the training time. [104] designed CloudViT based on the lightweight MobileViT in order to control the large number of parameters.

They achieved better performance than CNN-based models, as the transformer-based models attend to every part of the image and are able to capture global semantic features exploiting both spectral and spatial features. On the other hand, it makes the transformer-based models even more computationally expensive to train [105].

The structure of related work is shown in Figure 4.1.

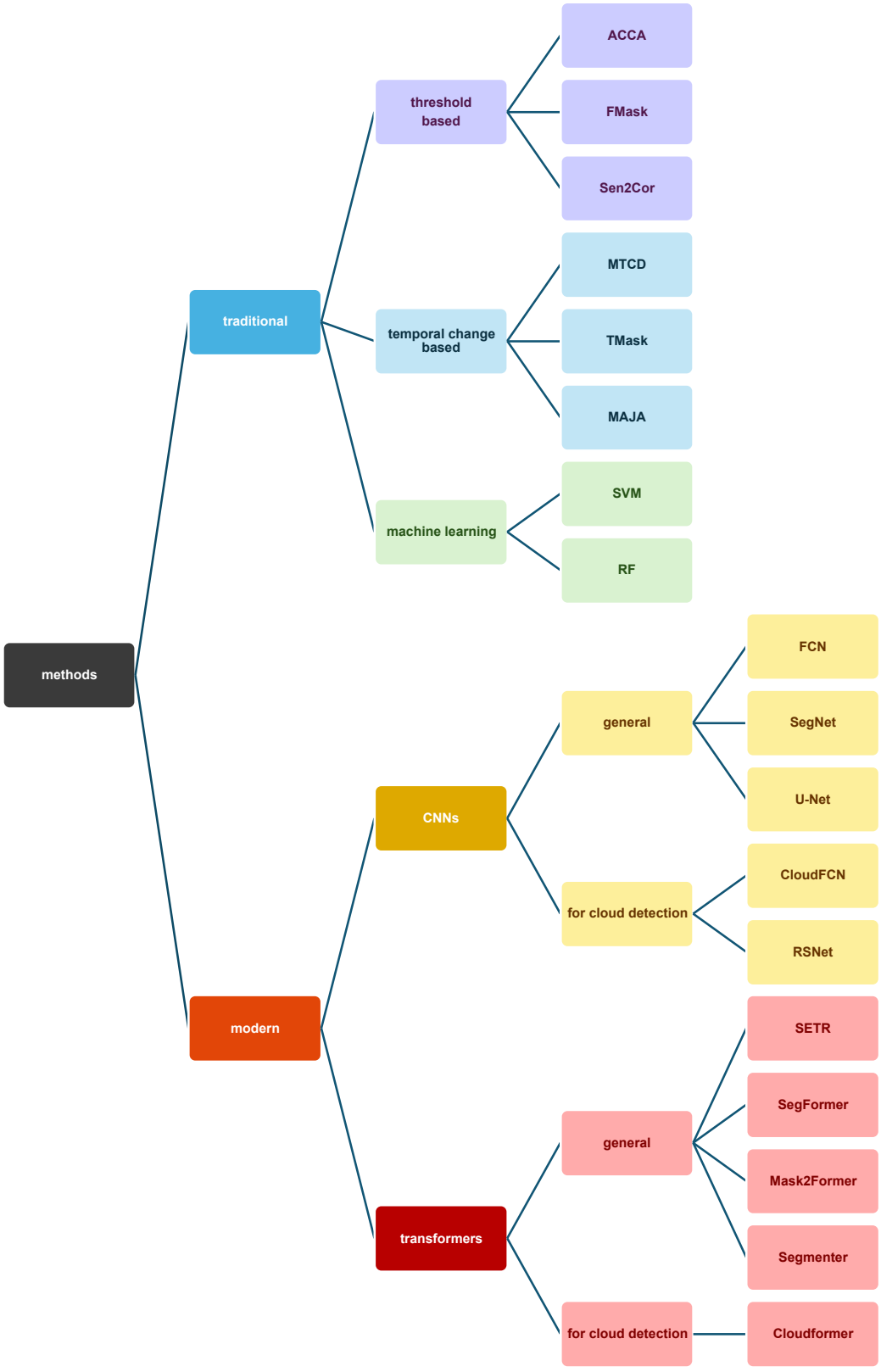


Figure 4.1 The structure of related work

4.3 Other Related Work

4.3.1 Cloud Shadow Detection

Numerous papers aim to detect not only clouds, but cloud shadows as well [54, 106, 107], as detection of them is also important for many remote sensing applications in various domains such as agriculture. The methods employed for this task are similar to the methods for cloud detection described above, the only difference is the number of classes detected.

4.3.2 Cloud Removal

Some related papers go even one step further and aim to remove the detected clouds and recover the pixels obstructed by clouds using various approaches such as generative adversarial networks (GANs) [108, 109].

Chapter 5
Examined Methods

This chapter provides a detailed description of selected methods that are implemented in the following chapter, namely U-Net and SegFormer. By choosing SegFormer, we slightly deviate from the assignment, which mentions CDUnet as an example of an advanced model. CDUnet is based on U-Net enhanced with a spatial attention module, which was introduced to establish the cloud spatial position information [85]. However, a transformer-based model inherently captures global spatial features more effectively [95], and therefore we consider it a more suitable choice for examination. From the transformer-based models mentioned in the previous chapter, SegFormer was chosen, particularly due to its state-of-the-art performance, but also because, unlike other models, its architecture is clearly and transparently described in [95].

5.1 U-Net

The U-Net architecture has been first proposed for biomedical image segmentation in [79]. It consists of three main parts: the contracting path consisting of encoder layers, the bottleneck, and the expansive path consisting of decoder layers. These parts are typically visualized in the shape of the letter U, as can be seen in the Figure 5.1, hence the name.

In the contracting path, each encoder layer consists of repeated convolutions, each followed by a nonlinear function such as ReLU, and a pooling operation for downsampling. The nonlinear function is crucial as it allows the neural network to learn and model complex patterns and relationships that are not possible to capture using only linear operations. The convolution and nonlinear function can be repeated multiple times in one encoder layer, extracting and learning the image features. The pooling operation then reduces the resolution of the input image in order to reduce the computational complexity.

The bottleneck is similar to the encoder layers, as it consists of convolution and nonlinear function. The only difference is that it lacks the pooling operation.

In the expansive path, each decoder layer starts with upsampling the output of the previous decoder layer or the bottleneck. That is done by using a transposed convolution, which increases the resolution while decreasing the number of features. At the final layer a 1x1 convolution is used to map each feature map to the desired number of classes.

In each encoder layer, fine details and higher spatial frequency are lost, as the convolution and pooling operations act as a low-pass filter. To address this, [original paper] introduced skip-connections, which concatenate feature maps before pooling from the contracting path with the respective upsampled feature maps from the expanding path. This is illustrated be the grey arrows in the Figure 5.1.

The U-Net architecture only uses the part of the convolution for which the full context is

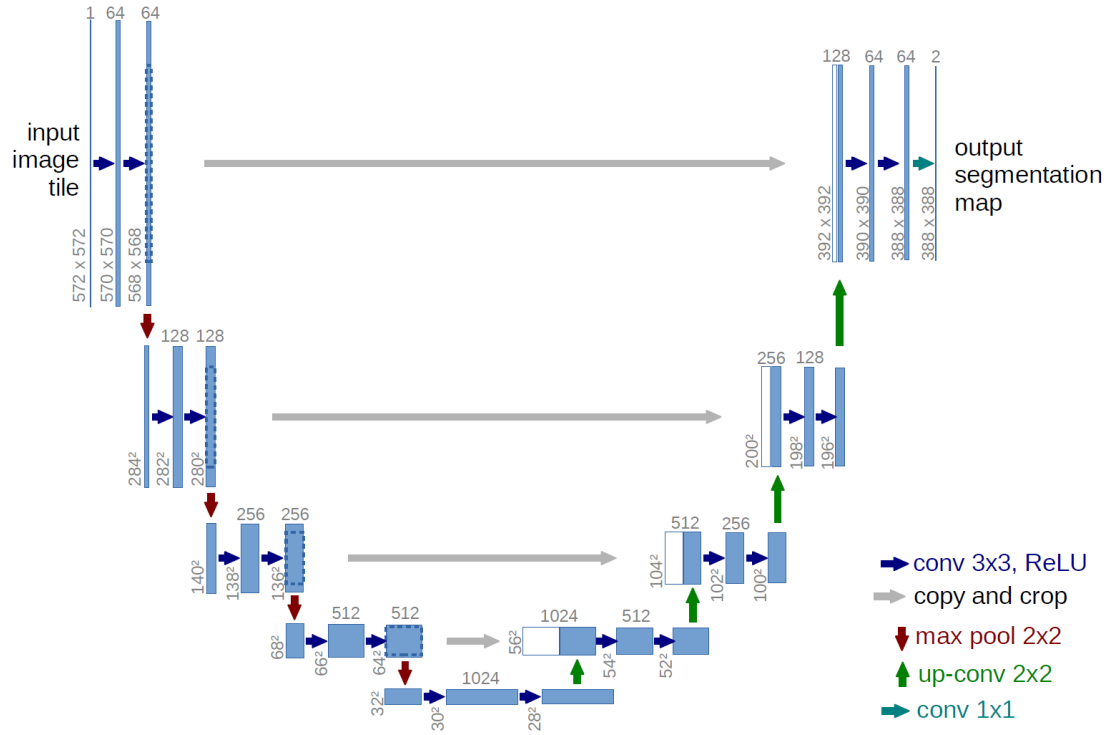


Figure 5.1 U-Net architecture as proposed in [79]. The blue boxes represent the multi-channel feature maps, while the number of channels is denoted on top of each of them. The resolution of each feature map is provided at the lower left edge of each box. The white boxes correspond to the feature maps copied by the skip connections. The arrows denote the different operations. Image from [79]

available in the input image. This allows for segmentation of arbitrarily large images using an overlap-tile strategy as illustrated in Figure 5.2. The missing context around the edges of the input image is extrapolated by mirroring the input image. This approach is crucial for making predictions for high resolution imagery, as the resolution would otherwise be limited by the GPU memory.

In conclusion, U-Net is simple, yet powerful architecture, that has been applied in various domains, including the cloud detection task [1, 84].

5.2 SegFormer

This section introduces SegFormer [95], a transformer-based model for semantic segmentation. As depicted in Figure 5.3, SegFormer consists of two main parts: a hierarchical transformer encoder to generate multi-scale features, and a lightweight MLP decoder to fuse these multi-scale features, and thus combining both local and global attention, to obtain the final semantic segmentation masks.

Given an input image of size $H \times W \times 3$, it is first divided into a grid of patches of size 4×4, where each patch is treated somewhat like a token in NLP transformers: flattened, and linearly projected to a higher-dimensional space. These patches are then used as input to the hierarchical transformer encoder to obtain multi-scale feature maps at $\frac{1}{4}, \frac{1}{8}, \frac{1}{16}$ and $\frac{1}{32}$ of the original input image resolution. The obtained multi-scale feature maps are then passed to the MLP decoder to predict segmentation mask at a $\frac{H}{4} \times \frac{W}{4} \times N_{cls}$ resolution, where N_{cls} is the number of categories.

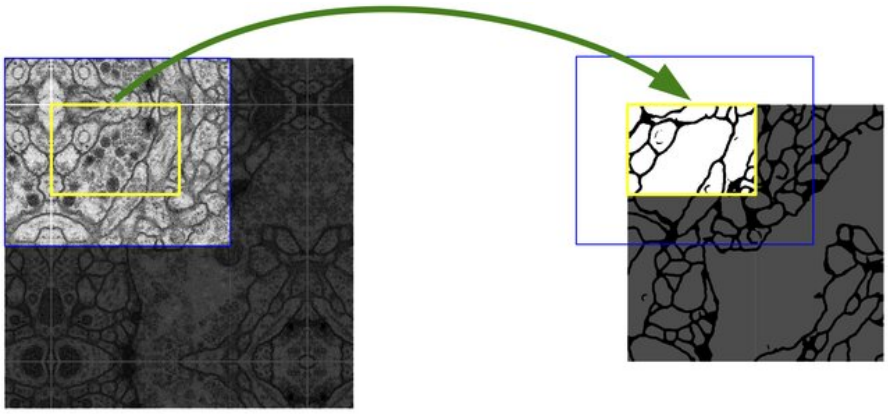


Figure 5.2 The overlap-tile strategy allows for segmentation of arbitrarily large images. Segmentation of the yellow area requires input image information from the blue area, which is extrapolated by mirroring. Image from [79]

5.2.1 Hierarchical Transformer Encoder

The goal of the hierarchical transformer is to generate multi-scale features to improve the semantic segmentation performance by capturing both local and global features. The hierarchical feature map is, given an input image with a resolution of $H \times W \times 3$, constructed as $\frac{H}{2^{i}+1} \times \frac{W}{2^{i}+1} \times C_i$, where $i \in \{1, 2, 3, 4\}$, and C_{i+1} is larger than C_i, using patch merging.

The patch merging process, as proposed in ViT [51], unifies a $N \times N \times 3$ patch into a $1 \times 1 \times C$ vector, which can be easily extended to unify a $2 \times 2 \times C_i$ feature map into a $1 \times 1 \times C_{i+1}$ vector. This is utilized for the construction of hierarchical feature maps, as for example to obtain $\frac{H}{8} \times \frac{W}{8} \times C_2$ from $\frac{H}{4} \times \frac{W}{4} \times C_1$. To allow for continuity around the borders of patches, overlapped patch merging is used with the following parameters: $K = 7, S = 4 \text{and} P = 3$ where K is the patch size, S is the stride between two adjacent patches and P is the padding size.

Self-attention is computationally the most expensive part of the encoder. In the paper that introduced self-attention [50], each of the heads Q, K and V have same dimensions $N \times C$, where $N = H \times W$ is the length of the sequence, and the computational complexity for self-attention as defined in Equation 3.2 is $O(N^2)$ [50], which is limiting for high resolution imagery. Therefore, the SegFormer architecture uses the following sequence reduction that was introduced in [110]

$$\begin{aligned} \hat{K} &= \text{Reshape}(\frac{N}{R}, C \cdot R)(K) \\ K &= \text{Linear}(C \cdot R, C)(\hat{K}), \end{aligned} \tag{5.1}$$

where K is the sequence to be reduced, R is the reduction rate, $\text{Reshape}(\frac{N}{R}, C \cdot R)(K)$ denote reshaping K to the shape of $\frac{N}{R} \times (C \cdot R)$, and $\text{Linear}(C_{in}, C_{out})(\cdot)$ denotes a linear layer that takes C_{in}-dimensional tensor as input and outputs a C_{out}-dimensional tensor. After the reduction, the new K has dimensions $\frac{N}{R} \times C$, resulting in the complexity of the self-attention to $O(\frac{N^2}{R})$. In the default configuration of SegFormer, the R is set to $64, 16, 4 \text{and} 1$ from first transformer block to the last one, respectively [95].

The ViT [51] and similar models use positional encoding to retain the positional information of each patch, since the self-attention treats each input independently of its position in the sequence. Without positional encoding, the transformer-based models would lose the spatial information. The SegFormer authors however argue, that positional encoding is not necessary for retaining the positional information, and use the findings of [111] that the positional information is implicitly encoded in CNNs. Based on that, Mix Feed Forward Network (Mix-FFN) is designed, which

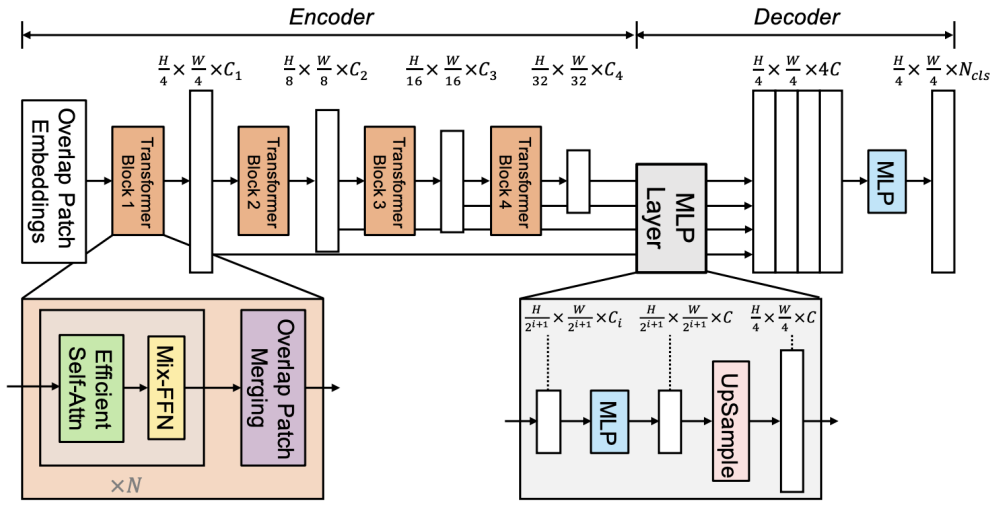


Figure 5.3 The SegFormer architecture consists of two main parts: a hierarchical transformer encoder to extract multi-scale features, and a lightweight MLP decoder to fuse these multi-scale features, and thus obtain the final semantic segmentation mask by combining both local and global features. Image from [95]

incorporates convolution operation and an MLP directly into the feed-forward network instead. That can be formulated as

$$x_{out} = \text{MLP}(\text{GELU}(\text{Conv}_{3\times3}(\text{MLP}(x_{in})))) + x_{in}, \tag{5.2}$$

where x_{in} is the output from the self-attention and GELU is the Gaussian Error Linear Unit.

5.2.2 MLP Decoder

A lightweight decoder consisting only of MLP layers is used, which is enabled by the hierarchical transformer encoder that has larger receptive field than CNNs.

The decoder consists of four parts. Firstly, multi-scale features from the encoder are passed through an MLP layer to unify the channel dimension. Secondly, the features are upsampled to $\frac{1}{4}$th of original resolution and concatenated together. Thirdly, the concatenated features are fused using another MLP layer. Finally, the last MLP layer predicts a segmentation mask M with resolution of $\frac{H}{4} \times \frac{W}{4} \times N_{cls}$ based on the fused features. This can be formulated as

$$\begin{aligned} \hat{F}_i &= \text{Linear}(C_i, C)(F_i), \forall i \\ \hat{F}_i &= \text{Upsample}(\frac{H}{4} \times \frac{W}{4})(\hat{F}_i), \forall i \\ F &= \text{Linear}(4C, C)(\text{Concat}(\hat{F}_i)), \forall i \\ M &= \text{Linear}(C, N_{cls})(F), \end{aligned} \tag{5.3}$$

where N_{cls} denotes the number of categories, M denotes the predicted segmentation mask, and $\text{Linear}(C_{in}, C_{out})(\cdot)$ denotes a linear layer that takes C_{in}-dimensional tensor as input and outputs a C_{out}-dimensional tensor.

In conclusion, SegFormer is simple, yet efficient transformer-based architecture for semantic segmentation. Thanks to the hierarchical transformer encoder and MLP decoder, the SegFormer does not need any too complex hand-crafted modules common in similar methods, leading to both lower computational efficiency and better performance.

Chapter 6
Implementation

This chapter introduces the technologies used for implementation and training of the models, describes the used dataset and how it has been preprocessed as well as the specific model architectures and the training pipeline.

6.1 Technologies

The project was implemented in Python 3.11[1] using deep learning framework PyTorch 2.2 [112] with CUDA 12.1[2].

The deep learning models were trained using two NVIDIA A100 Tensor Core GPUs with 40 GB of high-bandwidth memory, which is a high-performance GPU designed specifically for training large deep learning models, and which was provided by the Data Science Laboratory at FIT CTU.

Moreover, the following machine learning libraries were used:

- NumPy [113] for array operations,
- Python Imaging Library[3] (PIL) for imagery preprocessing,
- Python NVIDIA Management Library[4] (pynvml) for choosing the currently less busy GPU,
- scikit-learn [114] for cross-validation,
- HuggingFace[5] PyTorch-based datasets, transformers [115] and evaluate for implementation of SegFormer,
- Matplotlib [116] for visualizations.

The implemented code is attached to this bachelor thesis, mostly in the form of commented Jupyter notebooks.

[1] https://www.python.org/downloads/release/python-3110/
[2] https://pytorch.org/get-started/locally/
[3] https://pillow.readthedocs.io/en/stable/
[4] https://pypi.org/project/pynvml/
[5] https://huggingface.co/docs

6.2 Dataset

The dataset chosen for the experiments is the 38-Cloud dataset mentioned in Section 2.2.1.2. It is sufficiently large and representative, and also used in papers such as [2] and [84], allowing us to verify the performance of our models.

It consists of 38 scenes from various biomes acquired by the Landsat-8 satellite that are divided into 8,400 patches with the resolution of 384×384 and labeled for clouds. The number of bands is reduced from the 9 that Landsat-8 captures to four, keeping red, green, blue and near-infrared, as these are usually provided by other satellites such as Sentinel-2 or Gaofen-2 as well [31, 32].

The patches are separated into bands and each band is stored as a one channel TIF file. The preprocessing therefore consists of combining the separate bands into a multi-channel image, using the NumPy and PIL libraries, and then normalizing it. We do not apply any augmentation, as the dataset is already sufficiently large and representative. The manually labeled masks are also stored as TIF files and are normalized into the range $[0, 1]$. Figure 6.1 shows an example of a patch with its corresponding cloud mask.

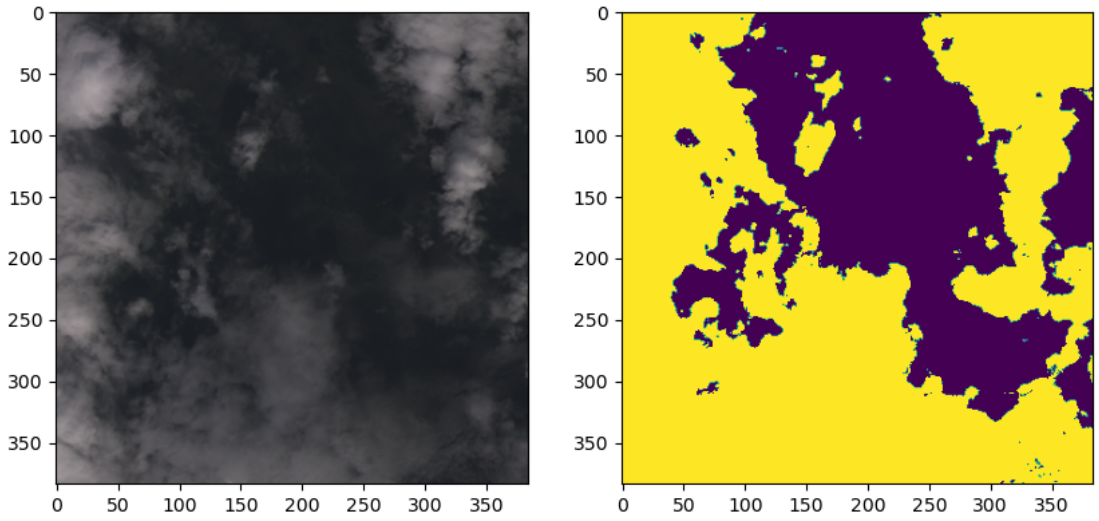
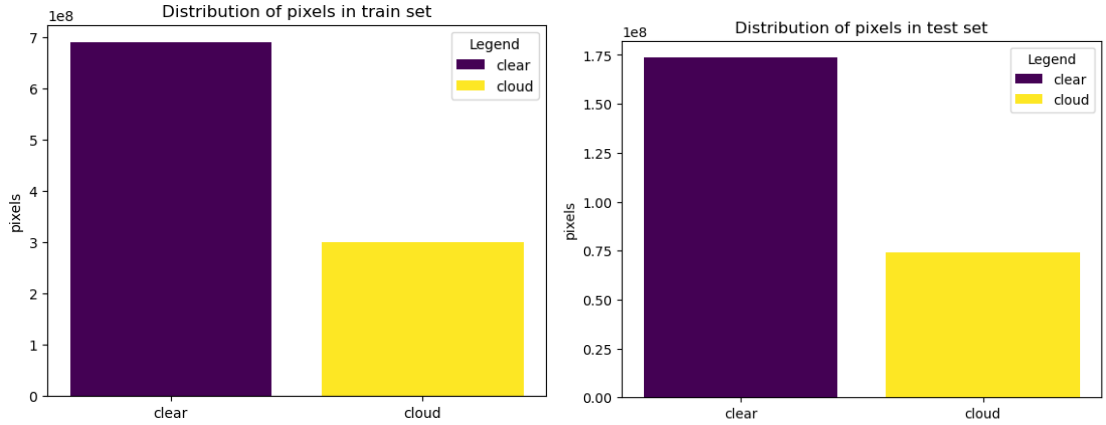


Figure 6.1 Example of a patch and its corresponding cloud mask from the 38-Clouds dataset.

6.3 Models

For the U-Net, a similar architecture to the one in the original paper [79] has been chosen, as it was successfully utilized in [1] and [84] for cloud detection. In the original paper, the U-Net consists of five blocks of convolution and downsampling in the contracting path and of five blocks of transposed convolution and convolution in the expansive path. In Section 6.5, we have tested removing some blocks and our final PyTorch implementation only employs four such blocks in the contracting and expansive path as they have comparable performance while reducing the computational complexity.

For SegFormer, its PyTorch-based implementation from the HuggingFace library was utilized, which has the exact architecture as in the original paper [95]. The original paper introduces six variants called SegFormer-B0 to Segformer-B6, where the B0 version is the most compact and lightweight, and B5 is the largest with 22 times more parameters. Due to limited available computational resources, we have chosen the smallest SegFormer-B0 pretrained model.



(a) Distribution of the train set

(b) Distribution of the test set

Figure 6.2 Comparison of the class distributions in the train and test subsets.

6.4 Training

In order to achieve reproducibility of all results, the seed for all random number generators (RNG) was fixed. This was accomplished by generating three random numbers, running the training and evaluation for one of them, and then verifying the results for the other two.

Furthermore, the dataset was divided into training and testing subsets, in a ratio of 80 % and 20 % respectively. Figure 6.2 shows that the distribution of cloud pixels in the subsets is balanced. Both train-validation-test split approach and cross-validation approach were explored. In the first case, a validation subset of 25 % was separated from the training set, while in the other case, 4-fold cross-validation was utilized.

Similarly to other papers on cloud detection using deep learning methods, we chose the cross-entropy as the loss function. Hyperparameters such as optimizer, learning rate, batch size, number of contracting and expansive block in the U-Net and size of the pretrained SegFormer model were subject to tuning.

6.5 Hyperparameter Tuning

Initially, we planned to use systematic methods such as GridSearch or Optuna for finding values of hyperparameters that lead to the best performance. However, the performance of U-Net was tuned quite quickly and the training time of SegFormer combined with small improvements was too long to justify running larger tuning. Therefore, we decided to only use manual tuning, as even that is beyond the scope of the assignment of this thesis.

The combinations of hyperparameters tested for U-Net and SegFormer are summed up in Tables 6.1 and 6.2, respectively. Their performance is to be found in the Section 7.1.

U-Net	epochs	optimizer	blocks	batch size	learning rate
1	25	Adam	4	16	0.001
2	25	AdamW	4	16	0.001
3	25	AdamW	3	16	0.001
4	25	AdamW	5	16	0.001
5	50	AdamW	4	16	0.001
6	50	AdamW	4	16	0.0001

Table 6.1 Configurations of hyperparameters tuned for U-Net

SegFormer	version	optimizer	epochs	batch size	learning rate
1	b0	AdamW	25	16	0.00005
2	b0	AdamW	25	16	0.0001

Table 6.2 Configurations of hyperparameters tuned for SegFormer

Chapter 7
Evaluation

In this chapter, the conducted experiments are presented and evaluated. These include the comparison of the performance of U-Net and SegFormer models on an unseen test set and both qualitative and quantitative analysis of their predictions. Special attention is given to how they handle spatial information and cloud boundary refinement. At the end of this chapter, we discuss how our results compare to the available papers and suggest the directions of future research.

7.1 Performance Comparison

The performance of the configurations of U-Net and SegFormer presented in Tables 6.1 and 6.2 was evaluated on an unseen test set using accuracy (Acc), Intersection over Union (IoU), F1 score, training time and the number of parameters, with accuracy and IoU being the key metrics optimized. The results for U-Net and SegFormer are summarized in Tables 7.1 and 7.3, respectively.

U-Net	**Acc ↑**	**IoU ↑**	**F1 ↑**	**training time ↓)**	**parameters ↓)**
1	0.962	0.875	0.932	45:43	2.2M
2	0.965	0.883	0.937	**35:32**	2.2M
3	0.953	0.846	0.915	46:12	**0.6M**
4	0.962	0.872	0.930	37:36	8.7M
5	0.961	0.872	0.930	1:59:38	2.2M
6	**0.967**	**0.894**	**0.943**	1:20:04	2.2M

Table 7.1 Performance comparison of the configurations of U-Net from the Table 6.1

SegFormer	**Acc ↑**	**IoU ↑**	**F1 ↑**	**training time ↓**	**parameters ↓**
1	0.980	0.889	0.916	6:05:23	3.8M
2	**0.981**	**0.892**	**0.919**	**5:50:48**	3.8M

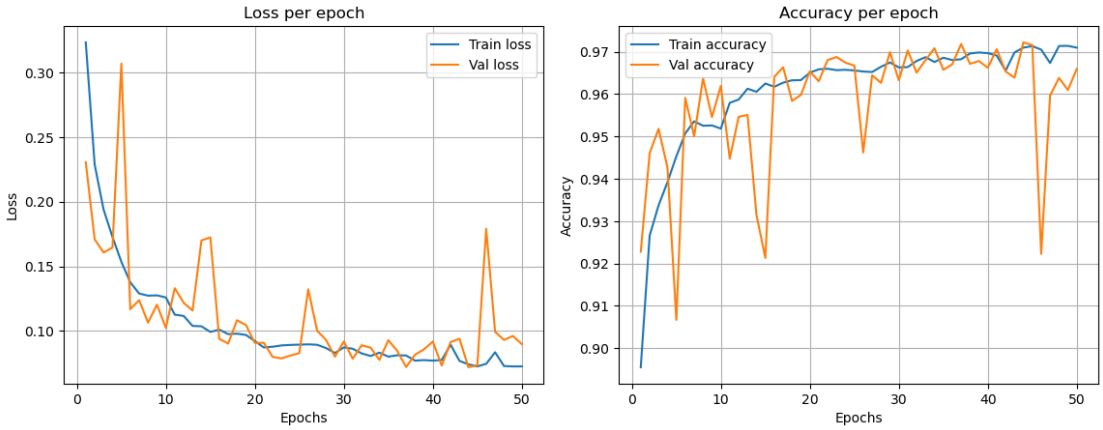
Table 7.2 Performance comparison of the configurations of SegFormer from the Table 6.2

In Table 7.1, it can be observed that:

- The model using AdamW as an optimizer expectably performs better than model employing Adam, as AdamW decouples weight decay from the gradient updates [117], leading to more effective regularization and therefore improved generalization performance.
- The model with only three contracting and expansive blocks predictably performs worse as it can capture less information. The model with four blocks performs the best, and a model with five blocks might capture slightly more information if trained for more epochs, but that would be at the cost of longer training time.
- The tested learning rates do not make much difference.
- The best-performing U-Net is therefore the version 7 with the following parameters:

$$\begin{aligned} \text{epochs} &= 5 \\ \text{blocks} &= 4 \\ \text{batch\_size} &= 16 \\ \text{learning\_rate} &= 0.001. \end{aligned} \tag{7.1}$$

Figure 7.1 then captures the development of the loss function and accuracy over epochs for the configuration six of the U-Net model, from which can be deduced from the shape of the graph that 50 epochs are sufficient, and that further training would not bring any significant improvement.



(a) Loss per epoch **(b)** Accuracy per epoch

Figure 7.1 Developement of loss and accuracy over epochs of the best U-Net

In the Table 7.3, it can be observed that:

- Using a larger learning rate than the Hugging Face library default of 0.00005 improves the performance.
- The best-performing SegFormer is therefore version 3 with the following parameters:

$$\begin{aligned} \text{version} &= b0 \\ \text{epochs} &= 10 \\ \text{batch\_size} &= 16 \\ \text{learning\_rate} &= 0.0001. \end{aligned} \tag{7.2}$$

7.2 Cross-validation

We have employed cross-validation to obtain a more accurate estimate of the performance of the best U-Net. The validation accuracy for each fold has been calculated, and then they were averaged to obtain the estimate, as shown in Table 7.4. The variance in accuracy is small, and the average 0.971 is even slightly higher than the accuracy of 0.967 from Table 7.1 measured on a test set, indicating that both are good estimates of how the model would generalize on unseen data, and that the result in Table 7.1 is not the product of a fortunate data distribution in the test set.

Table 7.3 Performance comparison of the configurations of SegFormer from the Table 6.2

Fold	1	2	3	4	avg
Accuracy	0.970	0.961	0.976	0.975	0.971

Table 7.4 Validation accuracy for each fold of the 4-fold cross-validation

7.3 Spatial Information

We carefully examined how both models handle spatial information. Intuitively, we expect that the transformer-based SegFormer, due to its ability to better capture global features, will perform better compared to U-Net, whose receptive field is local and limited by the size of the convolution kernel. Examination of the predictions confirms our hypothesis, as shown in Figure 7.2, where U-Net struggles to detect the cloud over a bright background. More examples of predictions are to be found in Appendix A.

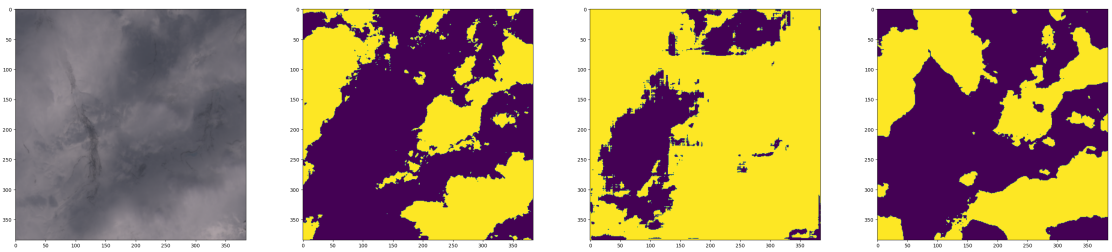


Figure 7.2 Samples of predicted cloud masks. The first column is the **image**, the second column is the **ground truth**, the third column is the prediction of the best **U-Net** model and the last column is the prediction of the best **SegFormer** model. Notice how U-Net struggles to detect the cloud over a bright background.

7.4 Cloud Boundary Detection

We also carefully investigated how both models handle cloud boundaries. Intuitively, we expect the U-Net to perform slightly better, due to its relatively small receptive field, that is capable of capturing fine details. Examination of the predictions confirmed our hypothesis, as shown in Figure 7.3. More examples of predictions are to be found in Appendix A.

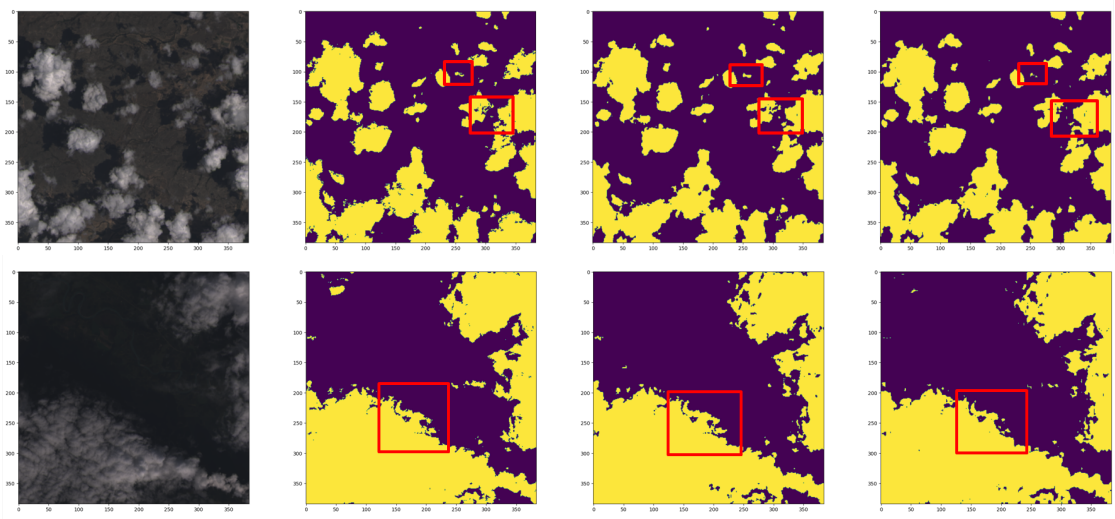


Figure 7.3 Samples of predicted cloud masks. The first column is the **image**, the second column is the **ground truth**, the third column is the prediction of the best **U-Net** model and the last column is the prediction of the best **SegFormer** model. Notice the difference in cloud boundary detection between the U-Net and Segformer prediction.

7.5 Discussion

The obtained results are fundamentally in agreement with the papers that focused on U-Net [84] and transformer-based [2] methods with this dataset, as summarized in Table 7.5. However, it does not make sense to make any further direct comparisons since none of these papers provide a transparent description of methodology used for training and evaluation.

U-Net based	**Acc** ↑	**IoU** ↑	**F1** ↑
U-Net [84]	0.958	0.861	0.923
Cloud-AttU [84]	0.971	0.887	0.941
U-Net (ours best)	**0.975**	**0.919**	**0.957**
Transformer based	**Acc** ↑	**IoU** ↑	**F1** ↑
CloudFormer [2]	0.979	**0.907**	-
SegFormer-B0 (ours best)	**0.982**	0.894	0.920

Table 7.5 Performance comparison of our best U-Net and SegFormer models with the results from papers [84] and [2]

Generally, the results show that SegFormer slightly outperforms the U-Net based architecture in terms of accuracy, while U-Net slightly outperforms SegFormer in the IoU and F1 score. Moreover, the U-Net based architecture has much shorter training time. This proves that even though being relatively simple and old, U-Net is still relevant and efficient architecture today.

It would be interesting to conduct more extensive hyperparameter tuning or to use a larger dataset with more channels; however, it is questionable how much room for improvement actually exists. During data exploration, we encountered a number of controversially labeled masks, such as shown in Figure 7.4. According to the study in [19], the inter-annotator agreement was 96 % when a cloud dataset was annotated by an independent interpreter, implying that around 4 % of the pixels in the dataset are ambiguous, and that the performance of cloud detection models will always be to an extent limited by the accuracy of human annotators.

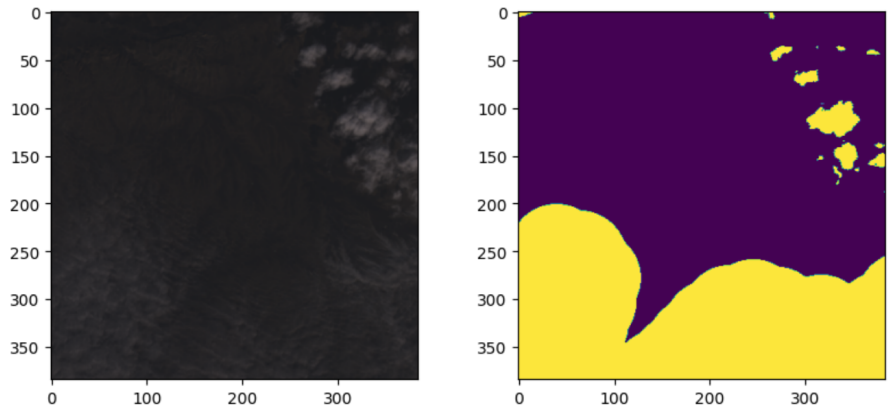


Figure 7.4 Example of a patch from the 38-Cloud dataset with its controversial mask. The lower part is suspiciously smooth and the upper right corner lacks a few small clouds, while clouds of this size are perfectly labelled in other patches.

It is also necessary to note that, as shown in Figure 6.2, the number of clear and obstructed pixels in the entire dataset is not balanced, meaning that all the models in Table 7.5 might be slightly biased towards predicting clear pixels.

7.6 Future Work

Even though the room for improvements in terms of accuracy is limited for this dataset, there are other directions for further research:

- Reducing the computational complexity: As shown in Table 7.3, the training time is still major drawback of deep learning-based methods. Techniques designed to efficiently fine-tune large-scale pre-trained models, such as Low-Rank Adaptation (LoRA) [118] might be examined and utilized.
- Integrating multi-source data from various satellites: Practically all current algorithms are trained and evaluated using datasets consisting of imagery from a single satellite. Designing a model capable of processing datasets fro various satellites would broaden its applicability.
- Utilizing all available bands and temporal history: Various traditional algorithms utilize all available bands and exploit the temporal history, but only a few deep learning-based methods do. This approach has the potential to increase accuracy, especially for very small and thin clouds.
- Improving the performance in more complex tasks: Such tasks include cloud and cloud shadow detection, or even cloud classification based on its thickness.
- Exploring the possibilities to remove the detected clouds: Some methods, such as generative adversarial networks, have been researched [108], but their performance is not yet convincing.

In general, these directions could improve the usability of satellite imagery in numerous remote sensing applications. However, in general, most of them require more computational resources and are far beyond the scope of bachelor's thesis.

Conclusion

This bachelor thesis aimed to review the recently published cloud detection methods, focusing on deep learning approaches, and to reimplement the most relevant ones and compare their performance using the most relevant datasets.

In the first part of this thesis (Chapters 1, 2 and 3), the introduction to cloud detection and satellite imagery has been made, and the deep learning background necessary to understand the thesis was defined.

In the second part of this thesis (Chapters 4 and 5), the current literature has been reviewed with focus on deep learning approaches. Based on the findings, the U-Net based architecture and transformer-based Segformer architecture have been chosen for further examinations, and have been explained in detail.

Finally, in the last part of the thesis (Chapters 6 and 7), our PyTorch implementation of the U-Net and SegFormer models is described, as well as all experiments performed. Generally, the results show that SegFormer slightly outperforms the U-Net based architecture in terms of accuracy, while U-Net slightly outperforms SegFormer in the IoU and F1 score. Moreover, the U-Net based architecture has much shorter training time. This proves that even though being relatively simple and old, U-Net is still relevant and efficient architecture today.

However, the analysis of the results also showed that each model has its strengths. U-Net, thanks to its convolutional layers, has a strong local receptive field, which enables it to effectively detect cloud boundaries. On the other hand, SegFormer, although it does not perform as well in cloud boundaries detection, can thanks to its transformer-based architecture capture complex and global spatial features, which enables it to detect clouds in challenging conditions, such as a cloud over a bright background, better.

Overall, the findings of this study can help other professionals choose the most suitable deep learning architecture for their use-case and provide a solid basis for the future research directions mentioned in Chapter 7.

Appendix A

Samples of Predicted Cloud Masks for Unseen Patches

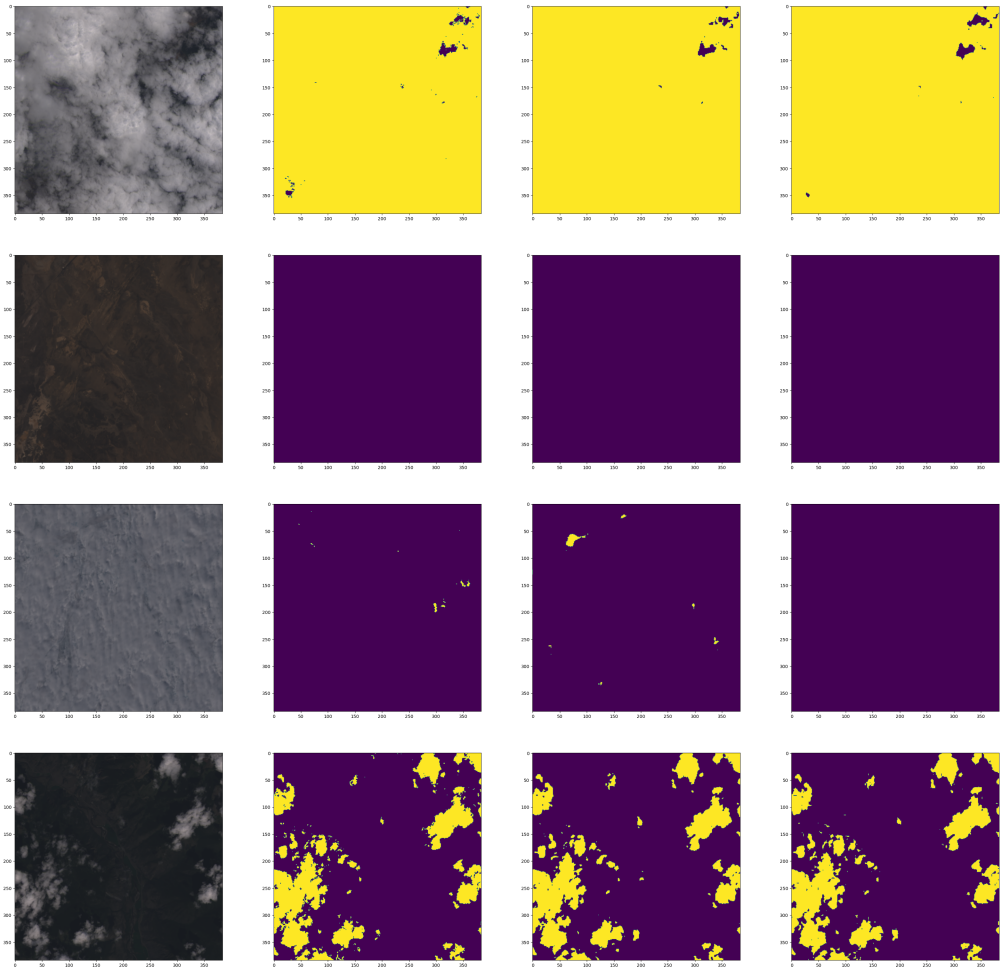


Figure A.1 Samples of predicted cloud masks. The first column is the **image**, the second column is the **ground truth**, the third column is the prediction of the best **U-Net** model and the last column is the prediction of the best **SegFormer** model.

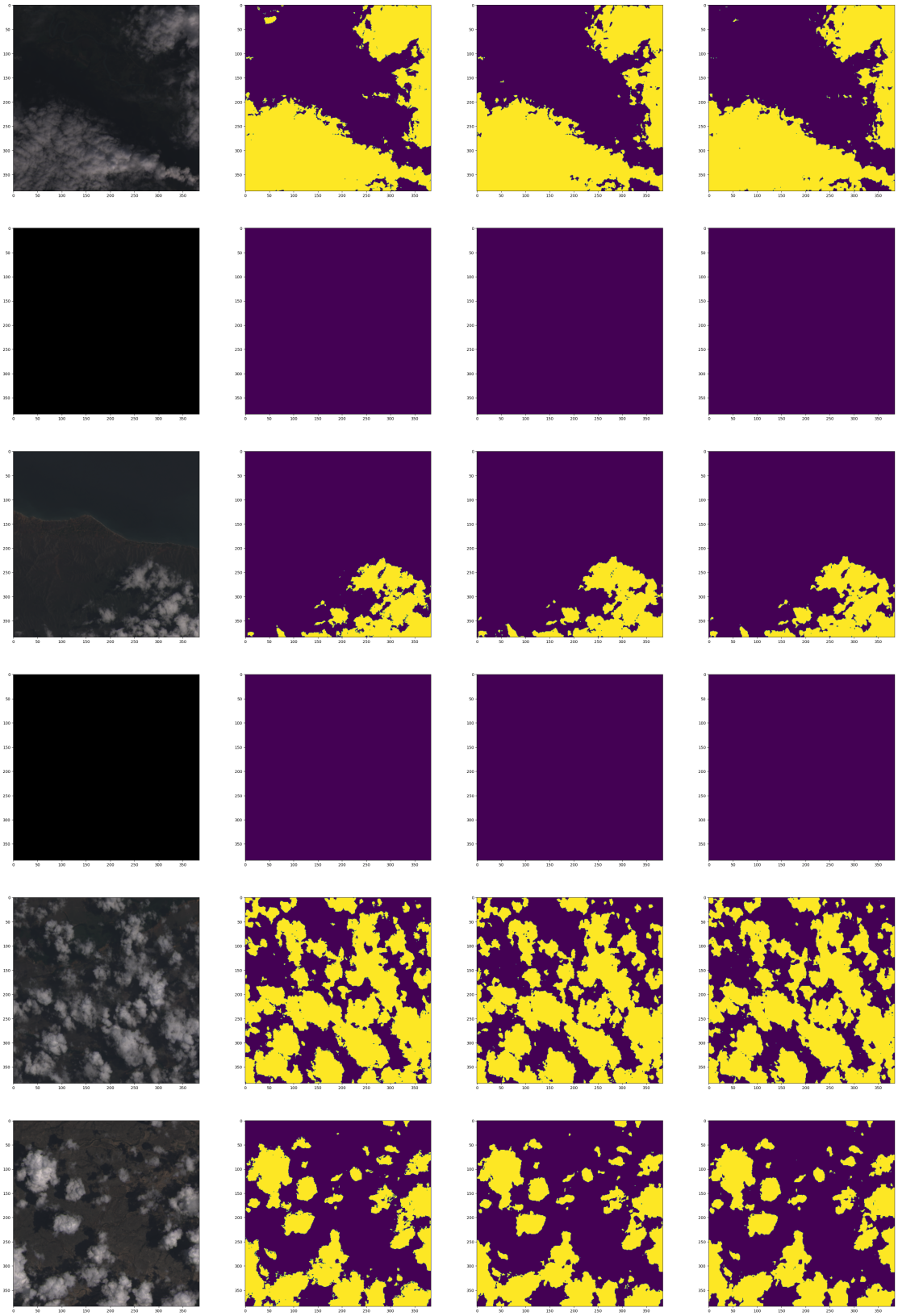


Figure A.2 Samples of predicted cloud masks. The first column is the **image**, the second column is the **ground truth**, the third column is the prediction of the best **U-Net** model and the last column is the prediction of the best **SegFormer** model.

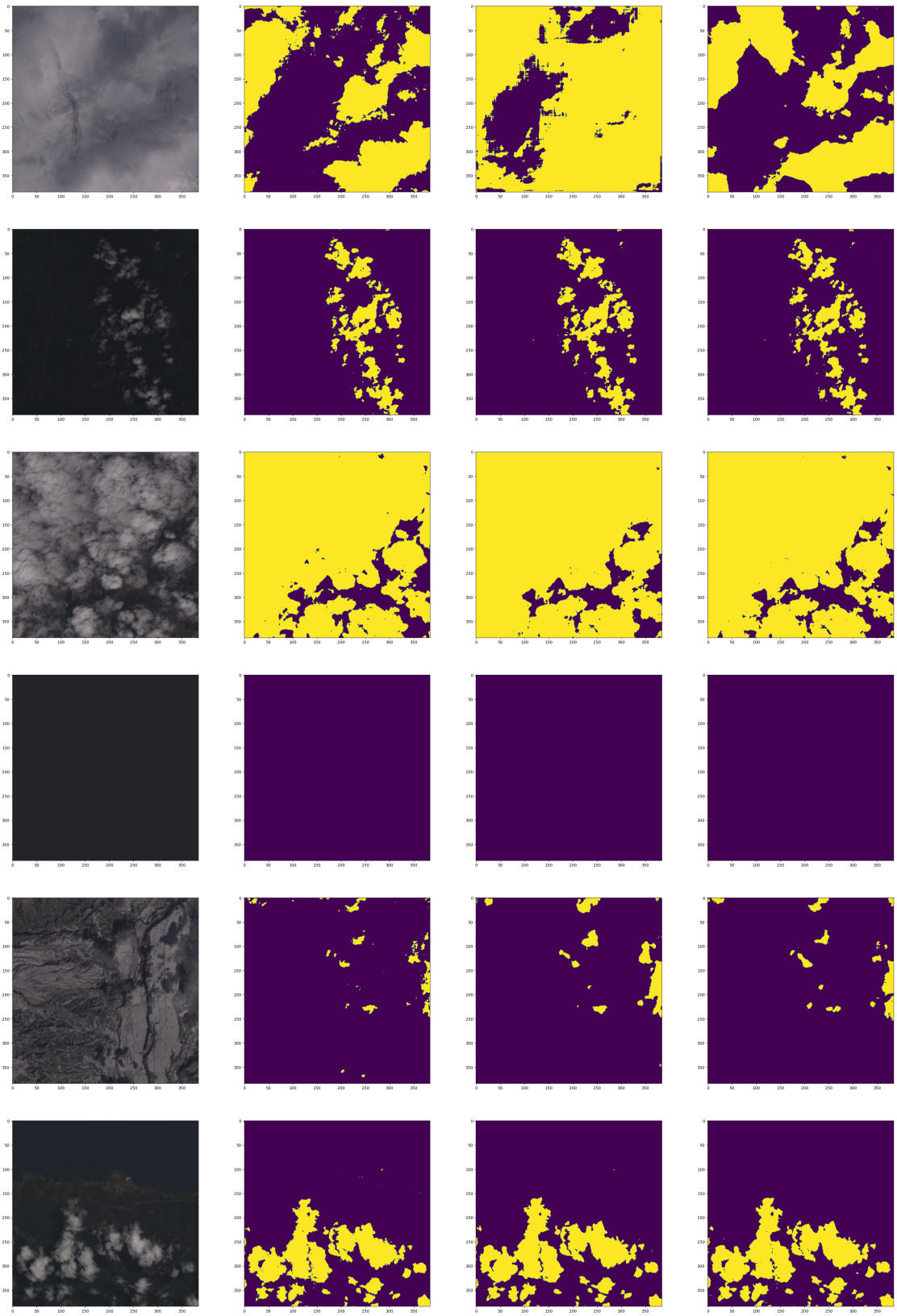


Figure A.3 Samples of predicted cloud masks. The first column is the **image**, the second column is the **ground truth**, the third column is the prediction of the best **U-Net** model and the last column is the prediction of the best **SegFormer** model.

Bibliography

1. JEPPESEN, Jacob Høxbroe; JACOBSEN, Rune Hylsberg; INCEOGLU, Fadil; TOFTEGAARD, Thomas Skjødeberg. A cloud detection algorithm for satellite imagery based on deep learning. *Remote Sensing of Environment*. 2019, vol. 229, pp. 247–259. ISSN 0034-4257. Available from DOI: https://doi.org/10.1016/j.rse.2019.03.039.
2. ZHANG, Zheng; XU, Zhiwei; LIU, Chang'an; TIAN, Qing; WANG, Yanping. Cloudformer: Supplementary Aggregation Feature and Mask-Classification Network for Cloud Detection. *Applied Sciences*. 2022, vol. 12, no. 7. ISSN 2076-3417. Available from DOI: 10.3390/app12073221.
3. TARRIO, Katelyn; TANG, Xiaojing; MASEK, Jeffrey; CLAVERIE, Martin; JU, Junchang; QIU, Shi; ZHU, Zhe; WOODCOCK, Curtis. Comparison of cloud detection algorithms for Sentinel-2 imagery. *Science of Remote Sensing*. 2020, vol. 2. Available from DOI: 10.1016/j.srs.2020.100010.
4. LI, Qianjing; TIAN, Jia; TIAN, Qingjiu. Deep Learning Application for Crop Classification via Multi-Temporal Remote Sensing Images. *Agriculture*. 2023, vol. 13, no. 4. ISSN 2077-0472. Available from DOI: 10.3390/agriculture13040906.
5. FUENTE, David; RIVILLA, Elena; TENA, Ana; VITORINO, João; NAVASCUÉS, Eva; TABASCO, Antonio. Yield estimation using machine learning from satellite imagery. *BIO Web of Conferences*. 2023, vol. 68. Available from DOI: 10.1051/bioconf/20236801013.
6. YU, Feng; WANG, Ming; XIAO, Jun; ZHANG, Qian; ZHANG, Jinmeng; LIU, Xin; PING, Yang; LUAN, Rupeng. Advancements in Utilizing Image-Analysis Technology for Crop-Yield Estimation. *Remote Sensing*. 2024, vol. 16, no. 6. ISSN 2072-4292. Available from DOI: 10.3390/rs16061003.
7. BENTIVOGLIO, R.; ISUFI, E.; JONKMAN, S. N.; TAORMINA, R. Deep learning methods for flood mapping: a review of existing applications and future research directions. *Hydrology and Earth System Sciences*. 2022, vol. 26, no. 16, pp. 4345–4378. Available from DOI: 10.5194/hess-26-4345-2022.
8. TANIM, Ahad Hasan; MCRAE, Callum Blake; TAVAKOL-DAVANI, Hassan; GOHARIAN, Erfan. Flood Detection in Urban Areas Using Satellite Imagery and Machine Learning. *Water*. 2022, vol. 14, no. 7. ISSN 2073-4441. Available from DOI: 10.3390/w14071140.
9. MCCARTHY, Matthew J.; JESSEN, Brita; BARRY, Michael J.; FIGUEROA, Marissa; MCINTOSH, Jessica; MURRAY, Tylar; SCHMID, Jill; MULLER-KARGER, Frank E. Mapping hurricane damage: A comparative analysis of satellite monitoring methods. *International Journal of Applied Earth Observation and Geoinformation*. 2020, vol. 91, p. 102134. ISSN 1569-8432. Available from DOI: https://doi.org/10.1016/j.jag.2020.102134.

10. YI, Yaning; XU, Xiwei; XU, Guangyu; GAO, Huiran. Landslide Detection Using Time-Series InSAR Method along the Kangding-Batang Section of Shanghai-Nyalam Road. *Remote Sensing*. 2023, vol. 15, no. 5. ISSN 2072-4292. Available from DOI: 10.3390/rs15051452.

11. SEYDI, Seyd Teymoor; SAEIDI, Vahideh; KALANTAR, Bahareh; UEDA, Naonori; HALIN, Alfian Abdul. Fire-Net: A Deep Learning Framework for Active Forest Fire Detection. *J. Sensors*. 2022, vol. 2022, pp. 1–14. Available also from: https://api.semanticscholar.org/CorpusID:247073229.

12. HU, Xikun; BAN, Yifang; NASCETTI, Andrea. Sentinel-2 MSI data for active fire detection in major fire-prone biomes: A multi-criteria approach. *International Journal of Applied Earth Observation and Geoinformation*. 2021, vol. 101, p. 102347. ISSN 1569-8432. Available from DOI: https://doi.org/10.1016/j.jag.2021.102347.

13. CANDRA, Danang. Deforestation detection using multitemporal satellite images. *IOP Conference Series: Earth and Environmental Science*. 2020, vol. 500, p. 012037. Available from DOI: 10.1088/1755-1315/500/1/012037.

14. HUANG, Chengquan; THOMAS, Nancy; GOWARD, Samuel; MASEK, Jeffrey; ZHU, Zhiliang; TOWNSHEND, J.; VOGELMANN, Jim. Automated masking of cloud and cloud shadow for forest change analysis using Landsat images. *International Journal of Remote Sensing - INT J REMOTE SENS*. 2010, vol. 31, pp. 5449–5464. Available from DOI: 10.1080/01431160903369642.

15. HARROU, Fouzi; BOUYEDDOU, Benamar; ZERROUKI, Nabil; DAIRI, Abdelkader; SUN, Ying; ZERROUKI, Yacine. Detecting the signs of desertification with Landsat imagery: A semi-supervised anomaly detection approach. *Results in Engineering*. 2024, vol. 22, p. 102037. ISSN 2590-1230. Available from DOI: https://doi.org/10.1016/j.rineng.2024.102037.

16. PERIN, Vinicius; ROY, Samapriya; KINGTON, Joe; HARRIS, Thomas; TULBURE, Mirela G.; STONE, Noah; BARSBALLE, Torben; REBA, Michele; YAEGER, Mary A. Monitoring Small Water Bodies Using High Spatial and Temporal Resolution Analysis Ready Datasets. *Remote Sensing*. 2021, vol. 13, no. 24. ISSN 2072-4292. Available from DOI: 10.3390/rs13245176.

17. ZHANG, Pengbin; KE, Yinghai; ZHANG, Zhenxin; WANG, Mingli; LI, Peng; ZHANG, Shuangyue. Urban Land Use and Land Cover Classification Using Novel Deep Learning Models Based on High Spatial Resolution Satellite Imagery. *Sensors*. 2018, vol. 18, no. 11. ISSN 1424-8220. Available from DOI: 10.3390/s18113717.

18. STUBENRAUCH, C. J.; ROSSOW, W. B.; KINNE, S.; ACKERMAN, S.; CESANA, G.; CHEPFER, H.; GIROLAMO, L. Di; GETZEWICH, B.; GUIGNARD, A.; HEIDINGER, A.; MADDUX, B. C.; MENZEL, W. P.; MINNIS, P.; PEARL, C.; PLATNICK, S.; POULSEN, C.; RIEDI, J.; SUN-MACK, S.; WALTHER, A.; WINKER, D.; ZENG, S.; ZHAO, G. Assessment of Global Cloud Datasets from Satellites: Project and Database Initiated by the GEWEX Radiation Panel. *Bulletin of the American Meteorological Society*. 2013, vol. 94, no. 7, pp. 1031–1049. Available from DOI: 10.1175/BAMS-D-12-00117.1.

19. FOGA, Steve; SCARAMUZZA, Pat L.; GUO, Song; ZHU, Zhe; DILLEY, Ronald D.; BECKMANN, Tim; SCHMIDT, Gail L.; DWYER, John L.; JOSEPH HUGHES, M.; LAUE, Brady. Cloud detection algorithm comparison and validation for operational Landsat data products. *Remote Sensing of Environment*. 2017, vol. 194, pp. 379–390. ISSN 0034-4257. Available from DOI: https://doi.org/10.1016/j.rse.2017.03.026.

20. LI, Liyuan; LI, Xiaoyan; JIANG, Linyi; SU, Xiaofeng; CHEN, Fansheng. A review on deep learning techniques for cloud detection methodologies and challenges. *Signal, Image and Video Processing*. 2021, vol. 15. Available from DOI: 10.1007/s11760-021-01885-7.

21. UNITED STATES GEOLOGICAL SURVEY (USGS). *Landsat 7* [online]. [N.d.]. Available from https://www.usgs.gov/landsat-missions/landsat-7, [Accessed 28-04-2024].
22. UNITED STATES GEOLOGICAL SURVEY (USGS). *Landsat 8* [online]. [N.d.]. Available from https://www.usgs.gov/landsat-missions/landsat-8, [Accessed 28-04-2024].
23. EUROPEAN SPACE AGENCY (ESA). *Sentinel-2* [online]. [N.d.]. Available from https://www.esa.int/Applications/Observing_the_Earth/Copernicus/Sentinel-2, [Accessed 28-04-2024].
24. UNION OF CONCERNED SCIENTISTS. *UCS Satellite Database* [online database]. 2023. Available from https://www.ucsusa.org/resources/satellite-database, [Accessed 20-04-2024].
25. LI, Zhiwei; SHEN, Huanfeng; WENG, Qihao; ZHANG, Yuzhuo; DOU, Peng; ZHANG, Liangpei. Cloud and cloud shadow detection for optical satellite imagery: Features, algorithms, validation, and prospects. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2022, vol. 188, pp. 89–108. Available from DOI: 10.1016/j.isprsjprs.2022.03.020.
26. NATIONAL AERONAUTICS AND SPACE ADMINISTRATION (NASA). *Landsat Science* [online]. [N.d.]. Available from https://landsat.gsfc.nasa.gov/, [Accessed 20-04-2024].
27. SCARAMUZZA, Pat; DWYER, John. *L7 Irish Cloud Validation Masks*. U.S. Geological Survey, 2016. Available from DOI: 10.5066/F7XD0ZWC.
28. FOGA, Steve; SCARAMUZZA, Pat L.; GUO, Song; ZHU, Zhe; DILLEY, Ronald D.; BECKMANN, Tim; SCHMIDT, Gail L.; DWYER, John L.; JOSEPH HUGHES, M.; LAUE, Brady. Cloud detection algorithm comparison and validation for operational Landsat data products. *Remote Sensing of Environment*. 2017, vol. 194, pp. 379–390. ISSN 0034-4257. Available from DOI: 10.1016/j.rse.2017.03.026.
29. SCARAMUZZA, Pat; FOGA, Steven; DWYER, John. *L8 Biome Cloud Validation Masks*. U.S. Geological Survey, 2016. Available from DOI: 10.5066/F7251GDH.
30. M. JOSEPH HUGHES. *L8 SPARCS Cloud Validation Masks*. U.S. Geological Survey, 2016. Available from DOI: 10.5066/F7FB5146.
31. MOHAJERANI, S.; SAEEDI, P. Cloud-Net: An End-To-End Cloud Detection Algorithm for Landsat 8 Imagery. In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*. 2019, pp. 1029–1032. ISSN 2153-6996. Available from DOI: 10.1109/IGARSS.2019.8898776.
32. MOHAJERANI, S.; KRAMMER, T. A.; SAEEDI, P. "A Cloud Detection Algorithm for Remote Sensing Images Using Fully Convolutional Neural Networks". In: *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*. 2018, pp. 1–5. ISSN 2473-3628. Available from DOI: 10.1109/MMSP.2018.8547095.
33. MOHAJERANI, S.; SAEEDI, P. Cloud-Net+: A Cloud Segmentation CNN for Landsat 8 Remote Sensing Imagery Optimized with Filtered Jaccard Loss Function. In: 2020, vol. 2001.08768.
34. EUROPEAN SPACE AGENCY (ESA). *The Sentinel missions* [online]. [N.d.]. Available from https://www.esa.int/Applications/Observing_the_Earth/Copernicus/The_Sentinel_missions, [Accessed 20-04-2024].
35. HOLLSTEIN, A.; SEGL, K.; GUANTER, L.; BRELL, M.; ENESCO, M. *Database File of Manually classified Sentinel-2A Data* [online repository]. 2017. Available from https://git.gfz-potsdam.de/EnMAP/sentinel2_manual_classification_clouds, [Accessed 22-04-2024].

36. LI, Jianfeng; WANG, Luyao; LIU, Siqi; PENG, Biao; YE, Huping. An automatic cloud detection model for Sentinel-2 imagery based on Google Earth Engine. *Remote Sensing Letters*. 2021, vol. 13, no. 2, pp. 196–206. ISSN 2150-7058. Available from DOI: 10.1080/2150704x.2021.1988753.

37. LI, Zhiwei; SHEN, Huanfeng; WENG, Qihao; ZHANG, Yuzhuo; DOU, Peng; ZHANG, Liangpei. Cloud and cloud shadow detection for optical satellite imagery: Features, algorithms, validation, and prospects. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2022, vol. 188, pp. 89–108. ISSN 0924-2716. Available from DOI: 10.1016/j.isprsjprs.2022.03.020.

38. LI, Jun; WU, Zhaocong; HU, Zhongwen; JIAN, Canliang; LUO, Shaojie; MOU, Lichao; ZHU, Xiao Xiang; MOLINIER, Matthieu. A Lightweight Deep Learning-Based Cloud Detection Method for Sentinel-2A Imagery Fusing Multiscale Spectral and Spatial Features. *IEEE Transactions on Geoscience and Remote Sensing*. 2022, vol. 60, pp. 1–19. ISSN 1558-0644. Available from DOI: 10.1109/tgrs.2021.3069641.

39. WU, Zhaocong; LI, Jun; WANG, Yisong; HU, Zhongwen; MOLINIER, Matthieu. Self-Attentive Generative Adversarial Network for Cloud Detection in High Resolution Remote Sensing Images. *IEEE Geoscience and Remote Sensing Letters*. 2020, vol. 17, no. 10, pp. 1792–1796. ISSN 1558-0571. Available from DOI: 10.1109/lgrs.2019.2955071.

40. LI, Zhiwei; SHEN, Huanfeng; LI, Huifang; XIA, Guisong; GAMBA, Paolo; ZHANG, Liangpei. Multi-feature combined cloud and cloud shadow detection in GaoFen-1 wide field of view imagery. *Remote Sensing of Environment*. 2017, vol. 191, pp. 342–358. ISSN 0034-4257. Available from DOI: 10.1016/j.rse.2017.01.026.

41. ZHU, Shaocong; LI, Zhiwei; SHEN, Huanfeng. Transferring Deep Models for Cloud Detection in Multisensor Images via Weakly Supervised Learning. *IEEE Transactions on Geoscience and Remote Sensing*. 2024, vol. 62, pp. 1–18. ISSN 1558-0644. Available from DOI: 10.1109/tgrs.2024.3358824.

42. HE, Qibin; SUN, Xian; YAN, Zhiyuan; FU, Kun. DABNet: Deformable Contextual and Boundary-Weighted Network for Cloud Detection in Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*. 2022, vol. 60, pp. 1–16. ISSN 1558-0644. Available from DOI: 10.1109/tgrs.2020.3045474.

43. LI, Zhiwei; SHEN, Huanfeng; CHENG, Qing; LIU, Yuhao; YOU, Shucheng; HE, Zongyi. Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2019, vol. 150, pp. 197–212. ISSN 0924-2716. Available from DOI: 10.1016/j.isprsjprs.2019.02.017.

44. ALBAWI, Saad; MOHAMMED, Tareq Abed; AL-ZAWI, Saad. Understanding of a convolutional neural network. In: *2017 International Conference on Engineering and Technology (ICET)*. 2017, pp. 1–6. Available from DOI: 10.1109/ICEngTechnol.2017.8308186.

45. ASTHANA, Meghna. *Introduction to Convolutional Neural Networks* [online]. [N.d.]. Available from https://medium.com/analytics-vidhya/introduction-to-convolutional-neural-networks-c50f41e3bc66, [Accessed 08-05-2024].

46. GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

47. REYNOLDS, Anh. *Convolutional Neural Networks (CNNs)* [online]. [N.d.]. Available from https://anhreynolds.com/blogs/cnn.html, [Accessed 08-05-2024].

48. YANI, Muhamad; IRAWAN, S; SETIANINGSIH, Casi. Application of Transfer Learning Using Convolutional Neural Network Method for Early Detection of Terry's Nail. *Journal of Physics: Conference Series*. 2019, vol. 1201, p. 012052. Available from DOI: 10.1088/1742-6596/1201/1/012052.

49. GEEKSFORGEEKS. *What is Transposed Convolutional Layer?* [Online]. [N.d.]. Available from https://www.geeksforgeeks.org/what-is-transposed-convolutional-layer/, [Accessed 09-05-2024].

50. VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N.; KAISER, Lukasz; POLOSUKHIN, Illia. *Attention Is All You Need*. 2023. Available from arXiv: 1706.03762 [cs.CL].

51. DOSOVITSKIY, Alexey; BEYER, Lucas; KOLESNIKOV, Alexander; WEISSENBORN, Dirk; ZHAI, Xiaohua; UNTERTHINER, Thomas; DEHGHANI, Mostafa; MINDERER, Matthias; HEIGOLD, Georg; GELLY, Sylvain; USZKOREIT, Jakob; HOULSBY, Neil. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *CoRR*. 2020, vol. abs/2010.11929. Available from arXiv: 2010.11929.

52. NARKHEDE, Sarang. *Understanding Confusion Matrix* [online]. [N.d.]. Available from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62, [Accessed 10-05-2024].

53. LI, Zhiwei; SHEN, Huanfeng; WENG, Qihao; ZHANG, Yuzhuo; DOU, Peng; ZHANG, Liangpei. Cloud and cloud shadow detection for optical satellite imagery: Features, algorithms, validation, and prospects. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2022, vol. 188, pp. 89–108. ISSN 0924-2716. Available from DOI: 10.1016/j.isprsjprs.2022.03.020.

54. ZHU, Zhe; WANG, Shixiong; WOODCOCK, Curtis E. Improvement and expansion of the Fmask algorithm: cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and Sentinel 2 images. *Remote Sensing of Environment*. 2015, vol. 159, pp. 269–277. ISSN 0034-4257. Available from DOI: 10.1016/j.rse.2014.12.014.

55. IRISH, Richard; BARKER, John; GOWARD, Samuel; ARVIDSON, Terry. Characterization of the Lfandsat-7 ETM+ Automated Cloud-Cover Assessment (ACCA) algorithm. *Photogrammetric Engineering & Remote Sensing*. 2006, vol. 72, pp. 1179–1188. Available from DOI: 10.14358/PERS.72.10.1179.

56. ZHU, Zhe; WOODCOCK, Curtis E. Object-based cloud and cloud shadow detection in Landsat imagery. *Remote Sensing of Environment*. 2012, vol. 118, pp. 83–94. ISSN 0034-4257. Available from DOI: https://doi.org/10.1016/j.rse.2011.10.028.

57. MAIN-KNORN, Magdalena; PFLUG, Bringfried; LOUIS, Jerome; DEBAECKER, Vincent; MÜLLER-WILM, Uwe; GASCON, Ferran. Sen2Cor for Sentinel-2. In: 2017, p. 3. Available from DOI: 10.1117/12.2278218.

58. PURKIS, Sam; KLEMAS, Victor. Remote Sensing and Global Environmental Change. In: 2013, pp. 91–121. ISBN 9781444339352. Available from DOI: 10.1002/9781118687659.ch6.

59. HAGOLLE, O.; HUC, M.; PASCUAL, D. Villa; DEDIEU, G. A multi-temporal method for cloud detection, applied to FORMOSAT-2, VENµS, LANDSAT and SENTINEL-2 images. *Remote Sensing of Environment*. 2010, vol. 114, no. 8, pp. 1747–1755. ISSN 0034-4257. Available from DOI: https://doi.org/10.1016/j.rse.2010.03.002.

60. ZHU, Zhe; WOODCOCK, Curtis E. Automated cloud, cloud shadow, and snow detection in multitemporal Landsat data: An algorithm designed specifically for monitoring land cover change. *Remote Sensing of Environment*. 2014, vol. 152, pp. 217–234. ISSN 0034-4257. Available from DOI: https://doi.org/10.1016/j.rse.2014.06.012.

61. LONJOU, Vincent; DESJARDINS, Camille; HAGOLLE, Olivier; PETRUCCI, Beatrice; TREMAS, Thierry; DEJUS, Michel; MAKARAU, Aliaksei; AUER, Stefan. MACCS-ATCOR joint algorithm (MAJA). In: 2016, p. 1000107. Available from DOI: 10.1117/12.2240935.

62. HOLLSTEIN, André; SEGL, Karl; GUANTER, Luis; BRELL, Maximilian; ENESCO, Marta. Ready-to-Use Methods for the Detection of Clouds, Cirrus, Snow, Shadow, Water and Clear Sky Pixels in Sentinel-2 MSI Images. *Remote Sensing*. 2016, vol. 8, no. 8. ISSN 2072-4292. Available from DOI: 10.3390/rs8080666.

63. GHASEMIAN, Nafiseh; AKHOONDZADEH, Mehdi. Introducing two Random Forest based methods for cloud detection in remote sensing images. *Advances in Space Research*. 2018, vol. 62, no. 2, pp. 288–303. ISSN 0273-1177. Available from DOI: 10.1016/j.asr.2018.04.030.

64. FU, Hualian; SHEN, Yuan; LIU, Jun; HE, Guangjun; CHEN, Jinsong; LIU, Ping; QIAN, Jing; LI, Jun. Cloud Detection for FY Meteorology Satellite Based on Ensemble Thresholds and Random Forests Approach. *Remote Sensing*. 2018, vol. 11, no. 1, p. 44. ISSN 2072-4292. Available from DOI: 10.3390/rs11010044.

65. CHEN, Xidong; LIU, Liangyun; GAO, Yuan; ZHANG, Xiao; XIE, Shuai. A Novel Classification Extension-Based Cloud Detection Method for Medium-Resolution Optical Images. *Remote Sensing*. 2020, vol. 12, no. 15, p. 2365. ISSN 2072-4292. Available from DOI: 10.3390/rs12152365.

66. BHAGWAT, Rohit; SHANKAR, B. Uma. A novel multilabel classification of remote sensing images using XGBoost. In: 2019, pp. 1–5. Available from DOI: 10.1109/I2CT45611.2019.9033768.

67. ZAMANI JOHARESTANI, Mehdi; CAO, Chunxiang; NI, Xiliang; BASHIR, Barjeece; TALEBIESFANDARANI, Somayeh. PM2.5 Prediction Based on Random Forest, XGBoost, and Deep Learning Using Multisource Remote Sensing Data. *Atmosphere*. 2019, vol. 10, no. 7. ISSN 2073-4433. Available from DOI: 10.3390/atmos10070373.

68. SUI, Yanlin; HE, Bin; FU, Tianjiao. Energy-based cloud detection in multispectral images based on the SVM technique. *International Journal of Remote Sensing*. 2019, vol. 40, no. 14, pp. 5530–5543. ISSN 1366-5901. Available from DOI: 10.1080/01431161.2019.1580788.

69. JOSHI, Pratik P.; WYNNE, Randolph H.; THOMAS, Valerie A. Cloud detection algorithm using SVM with SWIR2 and tasseled cap applied to Landsat 8. *International Journal of Applied Earth Observation and Geoinformation*. 2019, vol. 82, p. 101898. ISSN 1569-8432. Available from DOI: 10.1016/j.jag.2019.101898.

70. LI, Pengfei; DONG, Limin; XIAO, Huachao; XU, Mingliang. A cloud image detection method based on SVM vector machine. *Neurocomputing*. 2015, vol. 169, pp. 34–42. ISSN 0925-2312. Available from DOI: 10.1016/j.neucom.2014.09.102.

71. LEE, J.; WEGER, R.C.; SENGUPTA, S.K.; WELCH, R.M. A neural network approach to cloud classification. *IEEE Transactions on Geoscience and Remote Sensing*. 1990, vol. 28, no. 5, pp. 846–855. Available from DOI: 10.1109/36.58972.

72. YHANN, S.R.; SIMPSON, J.J. Application of neural networks to AVHRR cloud segmentation. *IEEE Transactions on Geoscience and Remote Sensing*. 1995, vol. 33, no. 3, pp. 590–604. ISSN 0196-2892. Available from DOI: 10.1109/36.387575.

73. HUGHES, M.; HAYES, Daniel. Automated Detection of Cloud and Cloud Shadow in Single-Date Landsat Imagery Using Neural Networks and Spatial Post-Processing. *Remote Sensing*. 2014, vol. 6, no. 6, pp. 4907–4926. ISSN 2072-4292. Available from DOI: 10.3390/rs6064907.

74. YUAN, Yi; HU, Xiangyun. Bag-of-Words and Object-Based Classification for Cloud Extraction From Satellite Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 2015, vol. 8, no. 8, pp. 4197–4205. Available from DOI: 10.1109/JSTARS.2015.2431676.

75. ZHU, Xiao Xiang; TUIA, Devis; MOU, Lichao; XIA, Gui-Song; ZHANG, Liangpei; XU, Feng; FRAUNDORFER, Friedrich. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geoscience and Remote Sensing Magazine*. 2017, vol. 5, no. 4, pp. 8–36. Available from DOI: 10.1109/MGRS.2017.2762307.

76. YUAN, Qiangqiang; SHEN, Huanfeng; LI, Tongwen; LI, Zhiwei; LI, Shuwen; JIANG, Yun; XU, Hongzhang; TAN, Weiwei; YANG, Qianqian; WANG, Jiwen; GAO, Jianhao; ZHANG, Liangpei. Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment*. 2020, vol. 241, p. 111716. ISSN 0034-4257. Available from DOI: https://doi.org/10.1016/j.rse.2020.111716.

77. LONG, Jonathan; SHELHAMER, Evan; DARRELL, Trevor. Fully convolutional networks for semantic segmentation. In: 2015, pp. 3431–3440. Available from DOI: 10.1109/CVPR.2015.7298965.

78. BADRINARAYANAN, Vijay; HANDA, Ankur; CIPOLLA, Roberto. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling. 2015.

79. RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. Available from arXiv: 1505.04597 [cs.CV].

80. CHEN, Liang-Chieh; ZHU, Yukun; PAPANDREOU, George; SCHROFF, Florian; ADAM, Hartwig. *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. 2018. Available from arXiv: 1802.02611 [cs.CV].

81. FRANCIS, Alistair; SIDIROPOULOS, Panagiotis; MULLER, Jan-Peter. CloudFCN: Accurate and Robust Cloud Detection for Satellite Imagery with Deep Learning. *Remote Sensing*. 2019, vol. 11, no. 19. ISSN 2072-4292. Available from DOI: 10.3390/rs11192312.

82. MOHAJERANI, Sorour; SAEEDI, Parvaneh. *Cloud-Net: An end-to-end Cloud Detection Algorithm for Landsat 8 Imagery*. 2019. Available from arXiv: 1901.10077 [cs.CV].

83. CHAI, Dengfeng; NEWSAM, Shawn; ZHANG, Hankui K.; QIU, Yifan; HUANG, Jingfeng. Cloud and cloud shadow detection in Landsat imagery based on deep convolutional neural networks. *Remote Sensing of Environment*. 2019, vol. 225, pp. 307–316. ISSN 0034-4257. Available from DOI: https://doi.org/10.1016/j.rse.2019.03.007.

84. GUO, Yanan; CAO, Xiaoqun; LIU, Bainian; GAO, Mei. Cloud Detection for Satellite Imagery Using Attention-Based U-Net Convolutional Neural Network. *Symmetry*. 2020, vol. 12, no. 6. ISSN 2073-8994. Available from DOI: 10.3390/sym12061056.

85. HU, Kai; ZHANG, Dongsheng; XIA, Min. CDUNet: Cloud Detection UNet for Remote Sensing Imagery. *Remote Sensing*. 2021, vol. 13, no. 22. ISSN 2072-4292. Available from DOI: 10.3390/rs13224533.

86. PENG, Longkang; CHEN, Xuehong; CHEN, Jin; ZHAO, Wenzhi; CAO, Xin. Understanding the Role of Receptive Field of Convolutional Neural Network for Cloud Detection in Landsat 8 OLI Imagery. *IEEE Transactions on Geoscience and Remote Sensing*. 2022, vol. 60, pp. 1–17. Available from DOI: 10.1109/TGRS.2022.3150083.

87. LI, Zhiwei; SHEN, Huanfeng; CHENG, Qing; LIU, Yuhao; YOU, Shucheng; HE, Zongyi. Deep learning based cloud detection for medium and high resolution remote sensing images of different sensors. *ISPRS Journal of Photogrammetry and Remote Sensing*. 2019, vol. 150, pp. 197–212. ISSN 0924-2716. Available from DOI: https://doi.org/10.1016/j.isprsjprs.2019.02.017.

88. SHAO, Zhenfeng; PAN, Yin; DIAO, Chunyuan; CAI, Jiajun. Cloud Detection in Remote Sensing Images Based on Multiscale Features-Convolutional Neural Network. *IEEE Transactions on Geoscience and Remote Sensing*. 2019, vol. 57, no. 6, pp. 4062–4076. Available from DOI: 10.1109/TGRS.2018.2889677.

89. LI, Yansheng; CHEN, Wei; ZHANG, Yongjun; TAO, Chao; XIAO, Rui; TAN, Yihua. Accurate cloud detection in high-resolution remote sensing imagery by weakly supervised deep learning. *Remote Sensing of Environment*. 2020, vol. 250, p. 112045. ISSN 0034-4257. Available from DOI: https://doi.org/10.1016/j.rse.2020.112045.

90. HE, Qibin; SUN, Xian; YAN, Zhiyuan; FU, Kun. DABNet: Deformable Contextual and Boundary-Weighted Network for Cloud Detection in Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing*. 2022, vol. 60, pp. 1–16. Available from DOI: 10.1109/TGRS.2020.3045474.

91. ZHANG, Jiaqiang; LI, Xiaoyan; LI, Liyuan; SUN, Pengcheng; SU, Xiaofeng; HU, Tingliang; CHEN, Fansheng. Lightweight U-Net for cloud detection of visible and thermal infrared remote sensing images. *Optical and Quantum Electronics*. 2020, vol. 52, p. 397. Available from DOI: 10.1007/s11082-020-02500-8.

92. YAO, Xudong; GUO, Qing; LI, An. Light-Weight Cloud Detection Network for Optical Remote Sensing Images with Attention-Based DeeplabV3+ Architecture. *Remote Sensing*. 2021, vol. 13, no. 18. ISSN 2072-4292. Available from DOI: 10.3390/rs13183617.

93. LI, Jun; WU, Zhaocong; HU, Zhongwen; JIAN, Canliang; LUO, Shaojie; MOU, Lichao; ZHU, Xiao Xiang; MOLINIER, Matthieu. A Lightweight Deep Learning-Based Cloud Detection Method for Sentinel-2A Imagery Fusing Multiscale Spectral and Spatial Features. *IEEE Transactions on Geoscience and Remote Sensing*. 2022, vol. 60, pp. 1–19. ISSN 1558-0644. Available from DOI: 10.1109/tgrs.2021.3069641.

94. ZHENG, Sixiao; LU, Jiachen; ZHAO, Hengshuang; ZHU, Xiatian; LUO, Zekun; WANG, Yabiao; FU, Yanwei; FENG, Jianfeng; XIANG, Tao; TORR, Philip H. S.; ZHANG, Li. Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers. *CoRR*. 2020, vol. abs/2012.15840. Available from arXiv: 2012.15840.

95. XIE, Enze; WANG, Wenhai; YU, Zhiding; ANANDKUMAR, Anima; ÁLVAREZ, José M.; LUO, Ping. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. *CoRR*. 2021, vol. abs/2105.15203. Available from arXiv: 2105.15203.

96. CHENG, Bowen; SCHWING, Alexander G.; KIRILLOV, Alexander. Per-Pixel Classification is Not All You Need for Semantic Segmentation. *CoRR*. 2021, vol. abs/2107.06278. Available from arXiv: 2107.06278.

97. CHENG, Bowen; MISRA, Ishan; SCHWING, Alexander G.; KIRILLOV, Alexander; GIRDHAR, Rohit. Masked-attention Mask Transformer for Universal Image Segmentation. *CoRR*. 2021, vol. abs/2112.01527. Available from arXiv: 2112.01527.

98. STRUDEL, Robin; PINEL, Ricardo Garcia; LAPTEV, Ivan; SCHMID, Cordelia. Segmenter: Transformer for Semantic Segmentation. *CoRR*. 2021, vol. abs/2105.05633. Available from arXiv: 2105.05633.

99. MEHTA, Sachin; RASTEGARI, Mohammad. MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer. *CoRR*. 2021, vol. abs/2110.02178. Available from arXiv: 2110.02178.

100. ALEISSAEE, Abdulaziz Amer; KUMAR, Amandeep; ANWER, Rao Muhammad; KHAN, Salman; CHOLAKKAL, Hisham; XIA, Gui-Song; KHAN, Fahad Shahbaz. Transformers in Remote Sensing: A Survey. *Remote Sensing*. 2023, vol. 15, no. 7. ISSN 2072-4292. Available from DOI: 10.3390/rs15071860.

101. ROHIT SINGH, Mantosh Biswas; PAL, Mahesh. A transformer-based cloud detection approach using Sentinel 2 imageries. *International Journal of Remote Sensing*. 2023, vol. 44, no. 10, pp. 3194–3208. Available from DOI: 10.1080/01431161.2023.2216850.

102. ZHANG, Zheng; XU, Zhiwei; LIU, Chang'an; TIAN, Qing; ZHOU, Yongsheng. Cloudformer V2: Set Prior Prediction and Binary Mask Weighted Network for Cloud Detection. *Mathematics*. 2022, vol. 10, no. 15. ISSN 2227-7390. Available from DOI: 10.3390/math10152710.

103. ZHANG, Zheng; TAN, Shuyang; ZHOU, Yongsheng. CloudformerV3: Multi-Scale Adapter and Multi-Level Large Window Attention for Cloud Detection. *Applied Sciences*. 2023, vol. 13, p. 12857. Available from DOI: 10.3390/app132312857.

104. ZHANG, Bin; ZHANG, Yongjun; LI, Yansheng; WAN, Yi; YAO, Yongxiang. CloudViT: A Lightweight Vision Transformer Network for Remote Sensing Cloud Detection. *IEEE Geoscience and Remote Sensing Letters*. 2023, vol. 20, pp. 1–5. Available from DOI: 10.1109/LGRS.2022.3233122.

105. HAN, Kai; WANG, Yunhe; CHEN, Hanting; CHEN, Xinghao; GUO, Jianyuan; LIU, Zhenhua; TANG, Yehui; XIAO, An; XU, Chunjing; XU, Yixing; YANG, Zhaohui; ZHANG, Yiman; TAO, Dacheng. A Survey on Vision Transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2023, vol. 45, no. 1, pp. 87–110. Available from DOI: 10.1109/TPAMI.2022.3152247.

106. WIELAND, Marc; LI, Yu; MARTINIS, Sandro. Multi-sensor cloud and cloud shadow segmentation with a convolutional neural network. *Remote Sensing of Environment*. 2019, vol. 230, p. 111203. ISSN 0034-4257. Available from DOI: https://doi.org/10.1016/j.rse.2019.05.022.

107. LI, Zhiwei; SHEN, Huanfeng; LI, Huifang; XIA, Guisong; GAMBA, Paolo; ZHANG, Liangpei. Multi-feature combined cloud and cloud shadow detection in GaoFen-1 wide field of view imagery. *Remote Sensing of Environment*. 2017, vol. 191, pp. 342–358. ISSN 0034-4257. Available from DOI: https://doi.org/10.1016/j.rse.2017.01.026.

108. ZAYTAR, Mohamed Akram; EL AMRANI, Chaker. Satellite image inpainting with deep generative adversarial neural networks. *IAES International Journal of Artificial Intelligence (IJ-AI)*. 2021, vol. 10, no. 1, p. 121. ISSN 2089-4872. Available from DOI: 10.11591/ijai.v10.i1.pp121-130.

109. ZHAO, Mingmin; OLSEN, Peder A.; CHANDRA, Ranveer. *Seeing Through Clouds in Satellite Images*. arXiv, 2021. Available from DOI: 10.48550/ARXIV.2106.08408.

110. WANG, Wenhai; XIE, Enze; LI, Xiang; FAN, Deng-Ping; SONG, Kaitao; LIANG, Ding; LU, Tong; LUO, Ping; SHAO, Ling. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. *CoRR*. 2021, vol. abs/2102.12122. Available from arXiv: 2102.12122.

111. ISLAM, Md. Amirul; JIA, Sen; BRUCE, Neil D. B. How Much Position Information Do Convolutional Neural Networks Encode? *CoRR*. 2020, vol. abs/2001.08248. Available from arXiv: 2001.08248.

112. PASZKE, Adam; GROSS, Sam; MASSA, Francisco; LERER, Adam; BRADBURY, James; CHANAN, Gregory; KILLEEN, Trevor; LIN, Zeming; GIMELSHEIN, Natalia; ANTIGA, Luca; DESMAISON, Alban; KÖPF, Andreas; YANG, Edward Z.; DEVITO, Zach; RAISON, Martin; TEJANI, Alykhan; CHILAMKURTHY, Sasank; STEINER, Benoit; FANG, Lu; BAI, Junjie; CHINTALA, Soumith. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *CoRR*. 2019, vol. abs/1912.01703. Available from arXiv: 1912.01703.

113. HARRIS, Charles R.; MILLMAN, K. Jarrod; WALT, Stéfan J. van der; GOMMERS, Ralf; VIRTANEN, Pauli; COURNAPEAU, David; WIESER, Eric; TAYLOR, Julian; BERG, Sebastian; SMITH, Nathaniel J.; KERN, Robert; PICUS, Matti; HOYER, Stephan; KERKWIJK, Marten H. van; BRETT, Matthew; HALDANE, Allan; RÍO, Jaime Fernández del; WIEBE, Mark; PETERSON, Pearu; GÉRARD-MARCHANT, Pierre; SHEPPARD, Kevin; REDDY, Tyler; WECKESSER, Warren; ABBASI, Hameer; GOHLKE, Christoph; OLIPHANT, Travis E. Array programming with NumPy. *Nature*. 2020, vol. 585, no. 7825, pp. 357–362. Available from DOI: 10.1038/s41586-020-2649-2.

114. PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, vol. 12, pp. 2825–2830.

115. WOLF, Thomas; DEBUT, Lysandre; SANH, Victor; CHAUMOND, Julien; DELANGUE, Clement; MOI, Anthony; CISTAC, Pierric; RAULT, Tim; LOUF, Rémi; FUNTOWICZ, Morgan; BREW, Jamie. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *CoRR*. 2019, vol. abs/1910.03771. Available from arXiv: 1910.03771.

116. HUNTER, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. 2007, vol. 9, no. 3, pp. 90–95. Available from DOI: 10.1109/MCSE.2007.55.

117. LOSHCHILOV, Ilya; HUTTER, Frank. Fixing Weight Decay Regularization in Adam. *CoRR*. 2017, vol. abs/1711.05101. Available from arXiv: 1711.05101.

118. HU, Edward J.; SHEN, Yelong; WALLIS, Phillip; ALLEN-ZHU, Zeyuan; LI, Yuanzhi; WANG, Shean; WANG, Lu; CHEN, Weizhu. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. Available from arXiv: 2106.09685 [cs.CL].

Concents of the Attachment

- `readme.txt` ..... more detailed information about the contents of the archive
- `src` ..... source code of models and experiments
  - `dataset`
  - `unet`
  - `segformer`
  - `dataset_exploration.ipynb`
  - `plot_predictions.ipynb`
- `text`
  - `thesis.pdf` ..... the thesis text in PDF format
  - `thesis.zip` ..... LaTeX source codes of the thesis