Degree Project in Machine Learning

First cycle, 30 credits

Exploring Deep Semantic Segmentation Models for Cloud Cover Estimation Using Cloud Images

DAVID MITROKHIN TOUMA

Exploring Deep Semantic Segmentation Models for Cloud Cover Estimation Using Cloud Images

DAVID MITROKHIN TOUMA

Master's Programme, Machine Learning, 120 credits
Date: November 7, 2024

Supervisors: Saikat Chatterjee, Isabel Ghourchian
Examiner: Pawel Herman
School of Electrical Engineering and Computer Science
Host company: Zenon AB
Swedish title: Utforskning av Djupa Semantiska Modeler för Molnighetskattning med hjälp av Molnbilder

Abstract

Predicting cloud cover from ground-based observations is crucial for weather forecasting and meteorological analyses. Current methodologies face challenges in terms of cost, time consumption, and accuracy. This study evaluates the effectiveness of three state of the art semantic segmentation models in accurately segmenting cloud images for cloud coverage computation.

The objectives include evaluating performance of each respective model, comparing their results with human observer predictions, and examining the impact of dataset volume on model performance. Methodologically, the study involves fine-tuning models on a custom dataset and conducting experiments to assess their capabilities.

An evaluation with various metrics showed that all models were able to segment cloud images well, with DeepLabV3 exhibiting superior performance in all evaluation metrics. Comparison with human predictions for cloud cover in cloud images suggests practical alignment, showing the viability of deep learning models in predicting cloud cover in cloud images. Moreover, the study revealed that dataset modifications, including data augmentation, expansion, and reduction of the dataset did not lead to significant improvements in model performance. For this reason, further exploration is encouraged, considering the homogeneity of the custom dataset.

In conclusion, this study advances cloud observation methodologies, providing insight into the applicability of deep learning models. Future research should focus on refining model generalization, exploring diverse datasets, and enhancing real-world applicability.

Keywords

Cloud Observation, Cloud Cover Prediction, Deep Learning, Semantic Segmentation

Sammanfattning

Att förutsäga molnighet baserat på markobservationer är avgörande för väderprognoser och metereologiska analyser. De nuvarande metoder som används står inför utmananingar avseende kostnads, tid och noggranhet. Denna studie utvärderar prestandan hos tre semantiska segmenteringsmodeller för segmentering av molnbilder för att kunna förutsäga molnigheten i bilderna.

Målen i studient innefattar utvärdering av de tidigarenämnda modellerna, jämföra molnigheten berkänad från segmentering med en metereologs gissning av molnighet på bilderna, samt undersöka datasetets påverkan på modellpresetandan. Modellerna i denna studie tränades på ett eget byggt dataset och användes som bas vid fine-tuning av modellerna.

En utvärdering med olika mätetal visade att alla modeller var kapabla till att segmentera molnbilderna väl, där DeepLabV3 uppvisade överlägsen prestanda på alla utvärderingsmått (IoU, F1, och pixelnoggranhet). Resultaten av jämförelse för molnighet på molnbilder mellan modell och metereolog visade kapaciteten hos djupinlärningsmodeller för att användas vid förutsägelser av molnighet i molnbilder. Slutligen visade resultaten i denna studie ingen signifikant förbättring vid modifikation av datat. Av denna anledning uppmuntras ytterligare utforskning att studera detta, med tanke på datasetes begränsningar.

Sammanfattningsvis främjar denna studie molnobservationsmetodik och ger insikter om användbarheten hos djupinlärningsmodeller. Framtida forskning bör fokusera på att förbättra modellgeneralisering, utforska olika dataset och förstärka användningen i verkliga situationer.

Nyckelord

Molnobservationer, Molnskattningar, Djupinlärning, Semantisk Segmentering

Acknowledgments

I would like to give huge thanks to Isabel Ghourchian and Per Hagström from Zenon AB for introducing me to the field of cloud observation and helping me define the scope of evaluating deep learning models for cloud cover analysis. My gratitude is also extended to my supervisor at KTH Royal Institute of Technology, Saikat Chatterjee, whom supervised this project. I would also like to thank Pawel Herman, who took his time to act as examiner for the thesis. I would also like to extend my gratitude to Youssef Taoudi for providing substantial feedback during the writing process. Finally, I want to extend my greatest gratitude to Jinya Sakurai, who contributed substantially to the thesis by aiding in defining the scope, design process, data collection, and being part of the experimental conduction. Without the contribution of these people, the thesis work could not have been finished.

Stockholm, November 2024
David Mitrokhin Touma

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Clouds are one of the most common weather phenomena, covering around 67% of the global surface [1], and have been widely studied as clouds play a pivotal role in the Earth's atmospheric movement, surface temperature regulation, and hydrological cycle [2]. Therefore, the development of cloud observation methods is important.

Cloud observations are made mainly in two ways, satellite-based observations and ground-based observations, where satellite-based observations are used to analyze a greater surface area, while ground-based observations are used to analyze a local area [2].

Ground-based cloud observation is highly flexible and accessible, and is good at monitoring the bottom characteristics of clouds, such as cloud height, cloud type and cloud cover [2]. Currently, cloud observations are mainly conducted by trained human observants, such as metereologist. However, it leads to a high human resource burden and uncertain subject bias toward the observation results [2] [3]. To alleviate the need for human resources, efforts have been put into developing tools that can accurately analyze clouds. Many ground-based cloud measurement devices, such as radar and lidar, are used to detect and determine cloud cover and cloud height. However, a much cheaper and more accessible tool has been developed in the form of all-sky-view imaging cameras (ASI), thus, effort has been put into using the images from the ASIs, mainly image segmentation, to be able to determine cloud characteristics, such as cloud type and cloud cover. In this thesis, we looked at how we can predict cloud cover based on cloud images

Precise cloud segmentation plays a crucial role in the analysis of ground-based ASI equipment, particularly in providing accurate information on cloud cover. Enhancing the precision of cloud cover information can enable meteorologists to obtain a better comprehension of the prevailing climatic conditions. Therefore, accurate cloud segmentation has emerged as a significant research area, with numerous algorithms proposed to address this issue. [3]

The emergence of image acquisition devices has led to several robust algorithms. Long et al. described in his article how clouds more evenly emits red and blue light whereas sky mostly only emits blue light, making it possible to propose a thresholding algorithm based on the red and blue channels [4]. Heinle et al. [5] used the Red-Blue color channels to set a threshold in their algorithm to classify clouds from whole sky images. Shi et al.[6] proposed a different approach to cloud image segmentation; using superpixels and graph models, they were able to segment cloud images effectively and more accurately than previous methods. Although progress has been made, achieving satisfactory cloud image segmentation based on ground-based imaging has shown to be a challenging task due to the blurry edged and varied shapes of the clouds. For this reason, the results have remained unsatisfactory and methods that could provide more accurate and robust cloud segmentation are still being explored.

The powerful representation ability of deep learning has made it the mainstream approach for numerous computer vision tasks, with convolutional neural networks (CNN) being a prominent technique [2]. For example, Shi et al. have proposed CloudU-Net and CloudU-Netv2 [7] [8], while CloudSegNet [9] and SegCloud [10] are similar models that employ an encoder-decoder architecture. In this architecture, the encoder consists of a CNN, which learns high-level and low-resolution features, while the decoder generates a segmentation mask of the input image.

Despite extensive research that has demonstrated the effectiveness of deep learning for cloud segmentation, the lack of labeled cloud image datasets in practical applications remains a challenge. Consequently, these models often exhibit the disadvantage of limited generalization ability [11]. The goal of transfer learning is to enhance the performance of target networks in target domains by leveraging the knowledge acquired from other source domains. This approach can help reduce the need for extensive target-domain data during the construction of the target network. One commonly adopted TL

strategy is to use pre-trained weights on a large dataset instead of random initialization [11].

1.1 Problem Statement

Ground-based cloud observations traditionally relies on manual observations by meteorologists or ceilometers, both of which have limitations. Manual observations are resource-intensive and prone to human error, while ceilometers have accuracy concerns [2]. This thesis explores the use of deep semantic segmentation models to automate cloud observations, aiming to improve accuracy and efficiency.

1.1.1 Scientific and engineering issues

Accurate cloud image segmentation using deep learning is challenging due to the limited availability of labeled datasets. This thesis investigates whether transfer learning can enhance model performance on cloud images and explored the dataset sizes required for effective segmentation. By addressing these issues, we aim to develop a robust approach for cloud cover prediction that can complement existing methods and reduce the need for extensive manual observation.

1.1.2 Research Questions

This thesis will explore the application of deep learning-based semantic segmentation methods for cloud cover prediction, focusing on their performance, efficiency, and scalability. The following research questions will guide our investigation:

1. *What are the advantages and disadvantages of deep learning-based semantic segmentation methods for cloud cover prediction in terms of predictive performance, efficiency, and implementation complexity?*

2. *How does the accuracy of cloud cover prediction differ between deep learning based methods and manual observations made by meteorologists in the Swedish cloud observation process?*

3. *What is the impact of dataset volume on the performance of state-of-the-art deep semantic segmentation models? Specifically, what is the minimum volume of data sets required to achieve accurate segmentation,*

and how does increasing the volume of data sets further enhance model performance?

1.2 Purpose

The purpose of this thesis is two-fold. Firstly, it aims to contribute to the advancement of the Swedish cloud observation process by incorporating machine learning techniques. Specifically, the thesis seeks to identify a feasible and efficient method that can deliver results comparable to those obtained via manual observations. The thesis seeks to address concerns about resource-intensiveness associated with manual cloud observations by developing a commercially viable solution that can improve the accuracy of cloud cover prediction. Additionally, machine learning methods have the potential to complement the information obtained from ceilometers, which have been associated with concerns about the accuracy of their predictions. Second, this thesis seeks to gain a deeper scientific understanding of the impact of a limited data set on deep semantic segmentation models. By evaluating model performance across varying dataset sizes, we aim to unravel the intricate relationship between dataset volume and the learning efficacy of these models. This knowledge will provide valuable insights into optimizing the learning process

Moreover, this thesis aims to improve upon current methods of cloud cover prediction, which have implications for achieving Sustainable Development Goals (SDGs). Notably, accurate cloud cover predictions align with SDG 11 (Sustainable Cities and Communities) by informing urban planning strategies, promoting energy efficiency, and contributing to the development of resilient cities in the face of climate-related challenges. Furthermore, advancements in cloud cover prediction contribute to SDG 7 (Affordable and Clean Energy) by optimizing the efficiency of renewable energy resources through improved forecasting techniques.

1.3 Delimitations

This study focuses mainly on the initial stage of the cloud observation process, namely cloud cover prediction, as defined by the World Meteorological Organization (WMO). Consequently, this thesis does not explore the remaining stages of the cloud observation process which includes, predicting

cloud height, cloud types, or other cloud features.

Due to the limitations of the camera setup, in this study only images captured during the day were selected.

Given the scarcity of data, transfer learning is the preferred method in exploring deep learning-based approaches. Thus, this study does not aim to propose a novel architecture, but rather scrutinizes the currently available state-of-the-art methods and pre-trained models.

1.4 Structure of the thesis

Chapter 2 of this thesis presents the background information relevant to this study as well as introducing the models used in this study. Chapter 3 describes the methodology in this thesis. The results of this study are presented in chapter 4 and discussed in chapter 5. Finally, the conclusions are stated in chapter 6.

Chapter 2

Background

This chapter introduces the key domains of this project, starting with an overview of image processing, image and semantic segmentation, which involves the methods and techniques employed for analyzing images. Subsequently, we delve into the field of machine learning, highlighting its applications in image processing and its evolutionary development over time. Moreover, a detailed description of the models utilized in this thesis is provided, elucidating their architecture and functionalities.

Furthermore, a section on related work is included, which offers a concise summary of recent advancements in the field of sky cloud image analysis. This section provides valuable context and outlines notable contributions made by researchers in this area.

2.1 Cloud Observation Analysis

Cloud observation can generally be categorized into two areas, satellite-based observations and ground-based observations [2]. Currently, ground-based observations are generally carried out by a trained metereologist [3]. Cloud observations are important for many reasons, such as weather forecasting, environmental studies, and more [3] [12]. To conduct a ground-based cloud observation correctly, the world metereological organization (WMO) has released a step-by-step guide on how to conduct such observations. The steps are listed in the following order:

1. Estimate or measure total cloud amount
2. Identify all clouds in the sky by genus, and where possible, species,

varieties, supplementary features, accessory clouds, mother-cloud and any other meteors associated with the cloud

3. Estimate or measure cloud amounts of the individual cloud genera and cloud layers
4. Estimate or measure cloud height
5. Estimate direction of movement

In many cases, only some parts of a cloud observation is of interest. For example, estimation of cloud cover are generally used for flight planning and aviation [12]. In this thesis, we will explore this area of cloud observation and how we can automate the process of estimating cloud cover in a local area.

2.2 Digital Image Processing

Digital image processing is a field that is comprised of the task of manipulating digital images using digital computers [13]. Image processing focuses on the analysis, manipulation, and interpretation of digital images and involves applying various techniques, algorithms, and methodologies to enhance, transform, and extract meaningful information from images. The goal of image processing is to improve the visual quality of images, extract important features, and enable automated understanding and interpretation of visual data for a wide range of applications in fields such as computer vision, medical imaging, remote sensing, and more.

2.2.1 Image Segmentation

Image segmentation, a specific field within digital image processing, involves partitioning an image into distinct regions or objects, serving as a crucial initial step for subsequent image analysis [14]. The primary objective of image segmentation is to delineate meaningful areas within an image that are pertinent to specific tasks or objects. For example, in medical imaging, the detection and isolation of organs of interest can be facilitated by segmentation techniques. Similarly, in applications such as autonomous driving and object location, computer systems interpret images by assigning labels to objects, allowing a comprehensive understanding of the environment [15].

Numerous techniques have been proposed for image segmentation, including thresholding methods, edge-based methods, and clustering methods [16]. This thesis aims to delve into neural network-based approaches, with a particular focus on exploring semantic segmentation, a specialized type of image segmentation that assigns semantic labels to individual pixels.

2.2.2 Semantic Segmentation

Semantic segmentation constitutes one of the prominent methodologies in image segmentation, aiming to assign precise semantic class labels to individual pixels within an image. As a supervised learning problem, semantic segmentation necessitates the training of classifiers from pixel-level labeled data [17]. As semantic segmentation is able to provide information at the pixel level, many real-world applications can benefit from this task, including self-driving vehicles, pedestrian detection, defect detection, and more [18]. By providing pixel-level information, this task aids systems to make informed decisions and accurate judgements.

In recent years, semantic segmentation has seen promising results with the introduction of deep learning and deep neural networks. Leveraging sufficient images and their corresponding pixel-wise labeling maps as training data, deep neural networks learn to establish a robust mapping between semantic labels and diverse visual representations. This learning process progressively reconciles the disparity between high-level semantics and low-level visual features, thereby enhancing the network's awareness of various semantic concepts [18].

In section 2.3.3 some of the state-of-the-art models used in this thesis are introduced and described.

2.3 Machine Learning

Machine learning (ML) is a subfield of artificial intelligence (AI) and focuses on the development of methodologies and algorithms that empower computational systems to learn from data, improve their performance over time, and make informed decisions without explicit programming [19]. ML can be defined as the process of learning and understanding patterns in data by adapting and tuning model parameters to the underlying probability distribution of the dataset. In turn, the model is capable of generating an output

"y" (predicted labels or values), representing predicted labels or values, based on the input *"x"* (input features or data points), which represents input features or data points [19]. The learning process that involves understanding the underlying distribution of the dataset is referred to as the *training phase*. Once the model completes its training, it gains the ability to identify the identity of new inputs, often referred to as the *test set*, which share characteristics with the data used during training. The model's capacity to accurately categorize novel examples, distinct from those encountered during the training phase, is recognized as *generalization*. Achieving a generalized model, capable of effectively handling diverse and previously unseen data, is an important objective in machine learning endeavors [19]. ML consists of different learning types, in this thesis, *supervised learning* [19] is the method used during model training.

- Supervised learning, also known as inductive learning, imitates human learning by gaining knowledge based on previous experiences. ML systems in supervised learning are trained on labeled datasets, where input and validation data have known labels. During training, the model adjusts its parameters to minimize the difference between its predictions and the true labels, learning from experience. This process enables the model to generalize patterns and make accurate predictions on new, unseen data. The success of supervised learning hinges on the quality of labeled training data, as it enables the model to adapt its knowledge effectively. [20]

Machine learning has gained widespread application in various fields, such as robotics, finance, medical sciences, and computer vision, owing to its capacity to discern patterns in extensive and multidimensional datasets. Despite its immense potential, ML is not without challenges, particularly those associated with data. The availability and quality of data play a pivotal role in model performance. In instances where data is insufficient, training a robust and generalizable model becomes challenging, leading to suboptimal outcomes. Data quality also influences model performance significantly, necessitating the development of sophisticated engineering techniques to mitigate such issues. Moreover, interpretability poses a common challenge, particularly in deep learning, rendering the decision-making process of ML models more intricate and less transparent [21]. In this thesis, we delve into these pertinent challenges and propose effective approaches to address them in Chapter 3.

2.3.1 Artificial Neural Networks and Deep Learning

Artifical neural networks (ANN), is a popular ML method that tries to simulate the mechanism of learning of biological organisms [22]. ANNs are at the core of deep learning, a subfield of machine learning that has revolutionized various domains with its ability to handle complex tasks and large datasets [22].

The architecture of artificial neural networks (ANNs) aims to replicate the communication observed in the brain, where neurons process and transmit information through interconnected pathways. In ANNs, nodes, also known as neurons, are organized into layers, and each node receives input and computes outputs based on its associated weights and activation function [22]. By combining perceptrons and arranging them into multiple layers, known as multi-layer perceptrons (MLP), ANNs have demonstrated the ability to solve nonlinear problems [22]. This integration of depth through the addition of layers, enabling the learning of hierarchical representations, together with advancements in activation functions, optimizations algorithms, initialization techniques, and the availability of large-scale datasets and powerful computational resources such as graphics processing unit (GPUs), has been the main reason for the advancements in deep learning [22].

Recent years has seen great progress for different deep learning techniques in several fields. In natural language processing (NLP), transformers has shown extraordinary results in tasks such as text comprehension [23]. Deep reinforcement learning integrates neural networks and reinforcement learning to enable agents to make decisions in different environments through trial and error [24]. Computer vision (CV) has made significant strides using convolutional neural networks (CNNs), which excel at encoding image-specific features and solving complex image analysis tasks[25][22].

2.3.2 Convoloutional Neural Network

One of the largest limitations of classic ANN is that they tend to struggle with the computational complexity involved with image data [25]. Similar to traditional ANN, convolutional neural networks has a similar structure in the way that it is a feed-forward network, meaning that an input will be fed to the input layer in the form of raw image vectors and eventually an output score is given, which then with a loss function, similar techniques are used to adjust the weights within the network [25].

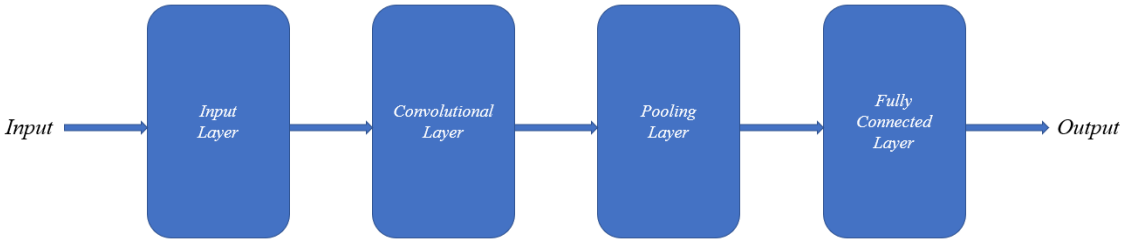


Figure 2.1: A general structure of a CNN with four layers

The architecture of a CNN is mainly comprised of these four components, the *input layer* - which holds the pixel values of the image, the *convolutional layer* - where convolution filters are applied to the input, the *clustering layers* - which downsamples the input, and a *fully connected layer* - which is used to produce class scores used for prediction [25]. Figure 2.1 demonstrates a standard CNN architecture. The layers are described in the following sections.

Input Layer

The input layer is the first and initial layer. The input layer is where the data points serving as input is introduced to the model. In this study, the input data is images, and will be seen as an array of pixels by the computer. [25] [26]

Convolutional Layer

The convolutional layer consists of a set of learnable kernels that perform the convolutions on the image [25]. The kernels are small in spatial dimensionality, but they spread along the entirety of the depth of the input [25]. When the input hits a convolutional layer, the kernel slides across the image, producing a 2-dimensional activation map [25]. This way, the kernel will know when to "fire" when they see a specific feature at a given spatial position; these are commonly known as **activations**. Each kernel will have the corresponding activation map, which will be stacked along the depth dimension to form the output volume of the convolutional layer [25]. The benefit of this, in comparison to traditional neural networks, is that we are able to drastically reduce the number of parameters. For example, if we have an input image of size $64x64x3$ and we set the kernel size to $6x6$, we would have a total of 108 weights in each neuron in the convolutional layer. Compared to a standard ANN, each neuron would contain 12,288 weights each [25].

To optimize the output of the convolutional layer, the programmer can set three

different hyperparameters, **stride**, **depth**, and **zero-padding**.

- Depth in the convolution layer refers to the number of channels/filters in the layer [25]. For example, setting depth to 16 would mean that the convolutional layer has 16 different kernel that are associated with each own kernel, that in turn are responsible of recognizing features in the image.
- Stride determines the sliding size of the kernel. If we set stride to 1, the kernel will move one pixel at a time, leading to a highly overlapping receptive field and a larger output. This is more computationally expensive, however, this enables the layer to capture more details and features. In contrast, a higher stride means that the step the kernel takes each iteration is larger, which means less overlap and a smaller output. This makes the computation faster and less expensive; however, this also means that the details captured by the kernels are less. [25]
- Zero-padding is the process of padding the borders of the input [25]. This is done so that the kernel does not go out-of-bounds when sliding through the input.

By using these hyperparameters, the spatial dimensionality of the output will change. To compute the output size of the convolutional layer, one can use the following equation:

$$\frac{(V - R) + 2Z}{S + 1}$$

Where V represents the input size ($height * width * depth$), R represents the kernel size, Z is the amount of zero padding, and S is the stride size [25]. The output will then become the new input for a pooling layer or sent to another convolutional layer for further processing.

Pooling layers

Pooling layers are used to gradually reduce the spatial dimensionality of the representation, thus further reducing the number of parameters and the computational complexity of the model [25]. This is usually done with max-pooling layers, which includes a small kernel (often the size of $2x2$ or $3x3$) that like kernels in the convolutional layers, slides through the feature input. Then it uses a "MAX" function, where the maximum value of each region is saved [25]. This enables the layer to reduce the spatial dimensionality while

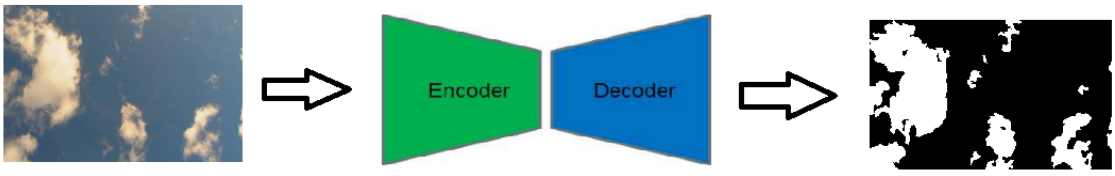


Figure 2.2: Typical encoder-decoder architecture of CNN based semantic segmentation

preserving important features [25].

If we have an activation map of size W x W x D, a pooling kernel of spatial size F, and stride S, then the size of output volume can be determined by the following formula:

$$\frac{W - F}{S} + 1$$

Fully-connected layer

Neurons in the fully-connected layer have full connectivity with neurons in the preceeding and and succeeding layer as in traditional fully-connected neural networks [25]. The output from the fully-connected layer is then sent in as input to the output layer which is responsible for prediction.

2.3.3 Deep Semantic Segmentation Models

As for other computer vision tasks, CNNs have been heavily used for semantic segmentation. Semantic segmentation architectures typically include a encoder-decoder, where the encoder extracts features from the image which are then decoded to produce a semantic segmentation output, like in figure 2.2 [27]. Although the proposed models generally have a common architecture, many different optimization techniques have been tested to increase current benchmarks [27] [28]. In this thesis, we introduce three of these models, the Fully Convolutional Networks (FCN) [29] that laid the foundation for the most modern segmentation architecture [28], the U-net model that built on FCN by introducing skip-connections [28], and finally, DeepLabv3 that introduced atrous convolutions and atrous spatial pyramid grouping [30]. In the following sections, these models are thouroughly explained.

2.3.3.1 Fully Convolutional Networks

The model proposed by Long, Shelhamer, and Darrel was the basis for how state-of-the-art architectures for semantic segmentation have been built [28]. The main challenge with semantic segmentation tasks in deep learning is the nature of losing local spatial information due to downsampling, which are crucial for accurate pixel-wise segmentation [29]. The authors addresses this problem by mainly incorporating two modifications, replacing the fully-connected layers with convolutional layers and incorporating skip connections throughout the network [29].

The authors use a encoder-decoder architecture where the encoder is responsible for downsampling the input image to capture high-level features. During this process, the spatial dimensions of the feature maps are reduced, which is the main cause of losing the spatial information. However, by incorporating skip connections between the encoder and decoder, it enables the transfer of feature maps from the encoder to the decoder, which effectively fuses low-level and high-level features [29].

As mentioned previously, the algorithms replaced fully connected layers with convolutional layers, as fully connected layers produce a fixed-size output vector, leading to a loss of spatial information. The output from fully connected layers are usually used for classification tasks, however, as this prediction made from the information from these layers do not contain spatial information, this is not suitable for pixel wise predictions. By replacing them with convolutional layers, the model is able to produce spatially dense predictions [29].

As ground truth is available at every output cell, end-to-end learning is possible with straightforward forward and backward passes, without the need of processing the raw image input [29]. The decoder network consists of upsampling layers that are responsible for increasing the spatial dimensions of the feature maps. In FCNs, two upsampling methods are used, backward convolution (sometimes called deconvolution), and bilinear interpolation [29]. Convolutional layers are also used in the decoder to further process upsampled feature maps and refine segmentation predictions [29].

By incorporating these modifications, the authors could achieve results that transcended previous state-of-the-art in pixel-wise predictions [29].

2.3.3.2 U-Net

The U-net [31] model incorporates the same idea as the FCN with included modifications to skip connections. The U-Net model was developed for biomedical image segmentation tasks [31]. Similarly to the architecture of the FCN model proposed by [29], the U-net utilizes an encoder-decoder structure that, in comparison with the previous model, is symmetrical and attains a U-shape, thus the name U-Net [31]. The major difference between the U-Net and FCN is the way the skip connections work. Previously in the FCN model, skip connections are implemented as "upsampling and sum" connections. During the upsampling process, the encoder feature maps are upsampled to match the size of the corresponding decoder feature maps. The upsampled feature maps are then element-wise added to the decoder feature maps to combine low-level and high-level information. U-net's skip connections are direct concatenation connections [31], where the feature maps from the encoder are not upsampled as in FCNs, but instead directly concatenated with the corresponding decoder feature maps. The authors showed that their architecture outperformed previous models in biomedical image segmentation challenges, and proposed that the model be used in other fields for semantic segmentation [31].

2.3.3.3 DeepLabv3

In more recent years, the DeepLabV3 [30] model was proposed by Chen, Zhu, Papandreou, Schroff, and Adam. It was an upgrade to its previous DeepLab predecessors [30]. To address the challenges of deep semantic segmentation models, the authors mainly utilized atrous convolution, also known as dilated convolution [30]. Atrous convolutions allow for control in how densely to compute feature responses, as atrous convolutions increase the receptive field of the kernel by adding holes in the kernel [30]. Using dilated kernels, the network is able to extract denser feature responses without having to learn any additional parameters, enabling maintenance of computational efficiency and performance [30]. Examples of the atrous convolution filter can be seen in figure 2.3.

To effectively capture multiscale information, the authors used spatial pyramid pooling (ASPP) [30]. ASPP is a module used at the end of the network that consists of parallel branches and uses multiple dilated convolutions at different rates, which means that it has different kernels with different sizes of the receptive field [30]. The ASPP module then uses global average pooling,

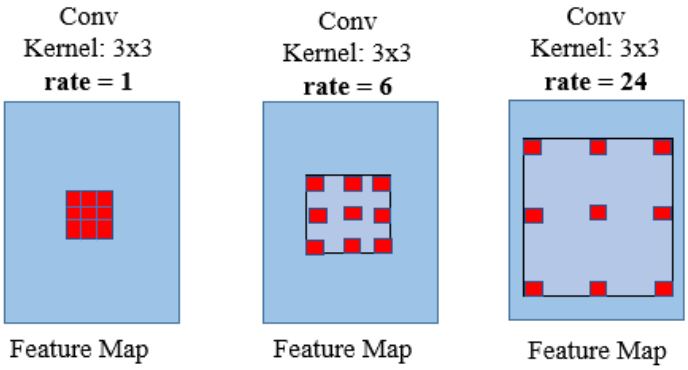


Figure 2.3: Atrous convolution with kernel size $3x3$ and different rates.

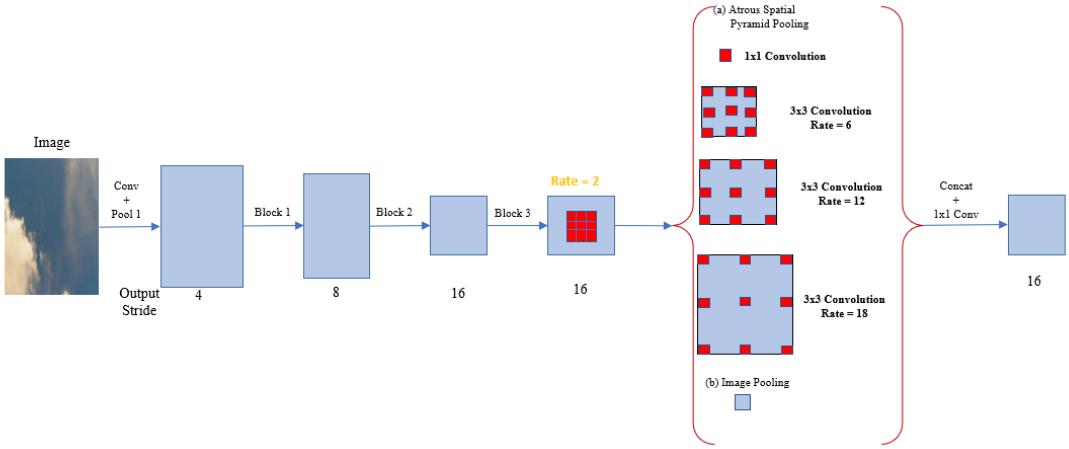


Figure 2.4: DeepLabV3 architecture.

as when the dilation rates become larger the number of valid filter weights decreases [30]. The DeepLabV3 architecture scheme, in comparison with previously mentioned models, does not adapt an encoder-decoder architecture in the same way. Instead, the DeepLabV3 model uses a ResNet model as the backbone, followed by dilated convolutional layers. Then an ASPP module is placed to capture multi-scale context in the image. Finally, the output of the ASPP module is passed through a $1x1$ convolution to get the actual size of the image, which then will be the final segmented mask for the image created by the model [30]. The architecture can be seen in figure 2.4.

The authors showed that DeepLabV3 outperformed previous DeepLab models, as well as achieving results comparable to state-of-the-art models on

the PASCAL VOC 2012 semantic image segmentation benchmark [30].

2.4 Related work

Effort to adapt semantic segmentation networks to sky cloud images has been made during recent years, as effective cloud image analysis is sought after. In this section, recent work on sky cloud semantic segmentation is presented and described.

2.4.1 CloudU-Netv2: A Cloud Segmentation Method for Ground-Based Cloud Images Based on Deep Learning

In [8], Shi, Zhou, and Qiu proposed the CloudU-Netv2 model, an upgrade of their previous CloudU-Net[8] model that was based on the U-net model. The authors proposed three different changes to their model, which significantly increased performance [8]. The CloudU-Netv2 model consists of the encoder-decoder architecture, as well as a dual-attention module (DAM) between the encoder and decoder. The DAM consists of position attention modules (PAM) and channel attention modules (CAM) [8]. The Position Attention Module (PAM) selectively combines features from different locations in the feature maps through weighted summation, emphasizing long-range dependencies and context at each position. On the other hand, the Channel Attention Module (CAM) emphasizes relevant interdependent channel mappings within the feature maps. Combining the outputs of both attention modules through concatenation enhances the overall feature representation, allowing the model to capture richer and more discriminative information [8].Other modifications include changing the upsampling method to bilinear upsampling, using rectified Adam as optimizer. The architecture of the proposed model can be found in figure 2.5.

Bilinear Upsampling

Bilinear upsampling is a method used in the decoder to upsample the image to its original spatial dimensions. The idea is to improve the spatial resolution of the feature maps from the previous model version [8]. Imagine you have a small image with a grid of pixels. Bilinear upsampling looks at the colors of neighboring pixels and creates new pixels in between them, filling in the gaps to make the image bigger. The idea is to perform linear interpolation

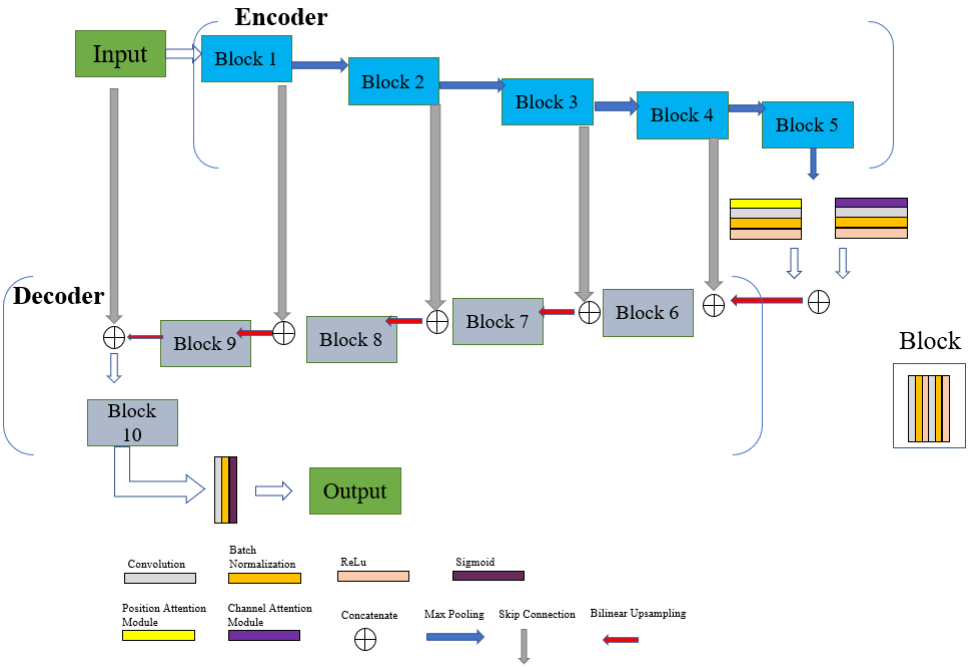


Figure 2.5: CloudU-Netv2 Architecture.

successively along the two directions of the coordinate axis [8]. Imagine that four points are known, $A = (x_1, y_2)$, $B = (x_2, y_2)$, $C = (x_1, y_1)$, $D = (x_2, y_1)$, and $f(x_3, y_3)$ at G is unknown. To compute the position of $f(x_3, y_3)$ at G, we first perform linear interpolation in the direction of the x-axis:

$$f(x_3, y_1) \approx \frac{x_2 - x_3}{x_2 - x_1} f(C) + \frac{x_3 - x_1}{x_2 - x_1} f(D)$$

$$f(x_3, y_2) \approx \frac{x_2 - x_3}{x_2 - x_1} f(A) + \frac{x_3 - x_1}{x_2 - x_1} f(B)$$

Then we can compute the position of $f(x_3, y_3)$ by linear interpolation in the direction of the y coordinate axis:

$$f(x_3, y_3) \approx \frac{y_2 - y_3}{y_2 - y_1} f(x_3, y_1) + \frac{y_3 - y_1}{y_2 - y_1} f(x_3, y_2)$$

Dual Attention Modules

The DAM was introduced to this model to capture local feature maps and global dependencies in spatial and channel dimensions [8]. The module is placed between the encoder and the decoder and consists of two different modules, the position attention module and the channel attention module [8].

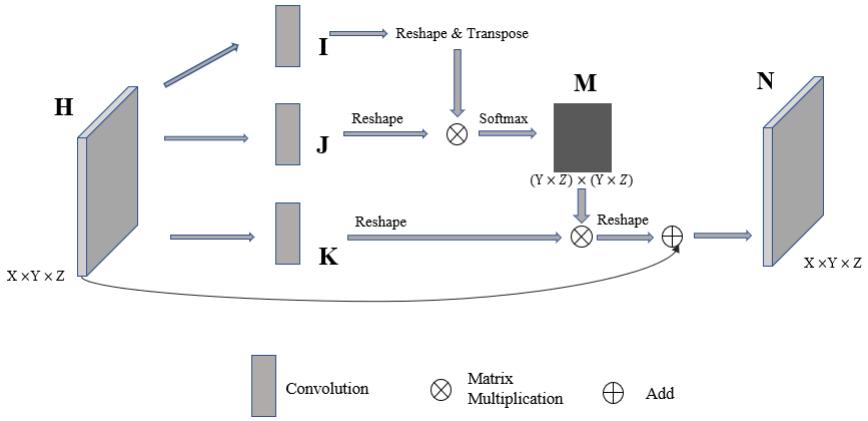


Figure 2.6: The structure detail of PAM.

Position Attention Module

As illustrated in figure 2.6, the encoder generates the local feature maps **H**. Firstly, H performs convolution operation to generate three feature maps **I**, **J** and **K**, where $\{I, J, K\} \in \mathrm{R}^{XxYxZ}$. The feature maps are then reshaped to R^{XxW}, where $W = YxZ$ represents the number of pixels. Second, the matrices **I** and the transpose of **J** are multiplied, and obtain the position attention features maps **M** by using a softmax activation function where m_{ji} represents the influence of the i-thj pixel on the j-th pixel. The closer the two pixels are, the greater their m_{ji} value. After that, the matrices **M** and the reshaped **K** are multiplied. Finally, the result is multiplied by a parameter α and an add operation is performed on the local feature maps H to obtain the output N of the PAM.[8]

Channel Attention Module

The channel attention module emphasizes the interdependence of features between different channels [8]. The structure detail of the CAM can be seen in figure 2.7. Compared to PAM, CAM computes attention feature maps **O** directly from **H**. The calculations are consistent with PAM. Finally, convolution operations on the output of the two modules are performed and fused to improve the feature representation. [8]

RAdam optimizer

During the initial phase of model training, insufficient training samples can lead to heightened influence of the initial few samples on the model parameters. Consequently, this can result in notable fluctuations in the

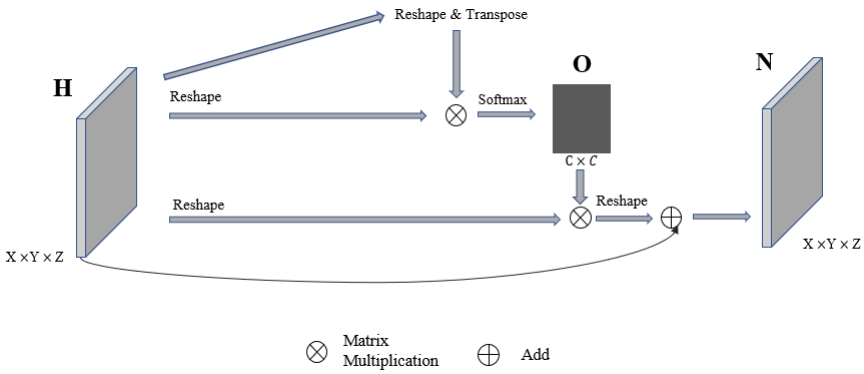


Figure 2.7: Structure detail of CAM.

adaptive learning rate, contributing to the model getting trapped in a local optimal solution. To address this issue, the present paper presents the RAdam optimizer, a modified version of the Adam optimizer. RAdam enhances convergence speed and effectively optimizes convergence, demonstrating robustness in achieving favorable outcomes. [8]

2.4.1.1 CloudU-Netv2 against other state-of-the-art

To thoroughly evaluate their model, the researchers compared its results with other advanced models using both quantitative and visual assessments. The outcomes revealed that their model outperformed the competition across all measures. Interestingly, their model's segmentation maps closely resembled the original ground truth data in most cases, highlighting its effectiveness and strength [8].

2.4.2 Cloud Image Segmentation Using Deep Transfer Learning

Due to the limited amount of labeled data, the authors in [11], explored how deep transfer learning can be used in the context of cloud image segmentation. Twelve state-of-the-art semantic segmentation models were selected, FCN-8, FCN-16, FCN-32, U-Net, SegNet, PSPNet [24], RefineNet, PAN, DeepLabV3, DeepLabV3+, DenseASPP, and BiSeNet [11]. The models were trained on the GBCS (Ground-Based Cloud Segmentation) dataset, which was manually created by the authors by collecting images from the web. When comparing the models quantitatively, they found that the DeepLabV3+ model had the best performance and was selected for further

training and evaluation [11]. Strong generalizability and robustness are important attributes for deep learning models. Therefore, the authors tested two different cloud image datasets with their trained DeepLabV3+ model. The qualitative analysis showed that their model was able to recognize different shapes of cloud quite well, however, it performed slightly worse on night-time images since they have less illumination [11].

The authors were able to show that transfer learning methods have a positive effect and efficiently improved the predictive power of cloud segmentation tasks. This can significantly reduce time consumption and workload in the annotation process. [11].

Chapter 3

Method

In this chapter, the main methods employed in the study are presented. Section 3.1 outlines the data-related methods, including data collection, datasets used, and data augmentation techniques. Section 3.2 discusses the evaluation methods utilized, along with the statistical techniques employed to validate the experimental results. Lastly, in Section 3.4, the experimental environment is detailed, covering software, tools, model implementation, parameters, training procedures, and the overall experimental setup.

3.1 Data

In this section, the datasets used for the thesis are explained, as well as how they were collected and processed.

3.1.1 Datasets

For this thesis, three distinct datasets have been used to facilitate comprehensive research and analysis. These datasets include our custom-built dataset, the Singapore Whole Sky IMagin SEGmentation Database (SWIMSEG) [32], and the Singapore Whole Sky Nighttime Image SEGmentation Database (SWINSEG) [33]. The SWIMSEG dataset, contains 1013 images of sky/cloud patches, and the SWINSEG dataset, contains 115 nighttime images of sky/cloud patches.

Our custom dataset serves as the primary resource for training and evaluating the different models investigated throughout the research. Concurrently, the SWIMSEG and SWINSEG datasets have been utilised in assessing the broader

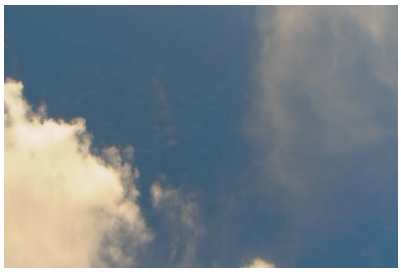


Figure 3.1: Original Image

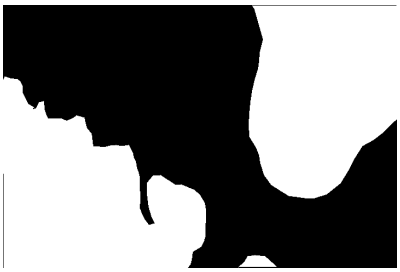


Figure 3.2: Binary Image

applicability and generalizability of the developed models. These publicly available datasets offer a diverse range of cloud and nighttime sky imagery, enabling a more comprehensive evaluation of the model's performance beyond our specific dataset.

3.1.2 Data Collection

Building our custom dataset was done in two steps, first collecting and filtering images that came from our image provider, and secondly we had to manually label our data using the MIT LabelMe tool. Our images were provided by the Swedish Meteorological and Hydrological Institute (SMHI) using a camera located at their station in Norrköping. Their camera was installed to take an image once every five minutes and then uploaded for us to download. Images older than ten days were deleted in their database, for this reason, we downloaded the images available to our local machine once every ten days. All the nightime images were removed as the camera was not able to take nightime images. Any images with noise were also removed. In the end, 264 images were used for training, and 60 separate images were used for testing. The labels for the images were manually made by using the LabelMe tool from MIT. An example of how an image looks like and its corresponding label can be found in 3.1 and 3.2. The images had a resolution of 720x480.

3.1.3 Data Augmentation

To observe whether the performance difference from the models could be increased, we tried to expand our datasets with different data augmentation techniques and append the transformed images to our existing dataset. As deep learning models run the risk of overfitting the data, data augmentation techniques can be used to reduce this risk [34]. Common techniques such as random rotation, random contrast changes, random brightness changes were

made to our existing dataset.

In the end, we combined the augmented images together with the original images to construct four different datasets;

- **Dataset 1:** Represents the original dataset with no modifications applied to the images, serving as the baseline for performance comparison. Contains 264 images.
- **Dataset 2:** An upsampled version of Dataset 1, augmented with random rotations to introduce variability in image orientation. Contains 528 images.
- **Dataset 3:** Similar to Dataset 2, this is an upsampled version of Dataset 1; however, the images are modified by applying random changes in brightness and contrast, testing model robustness against lighting variations. Contains 528 images.
- **Dataset 4:** An upsampled composite of Dataset 1 that incorporates both types of modifications from Datasets 2 and 3 (rotations and brightness/contrast adjustments), designed to challenge the models with a combination of augmentations. Contains 792 images.
- **Dataset 5:** A downsampled version of Dataset 1, containing half the volume of images but maintaining an equivalent distribution of different cloud coverages, to assess model performance with reduced data availability. Contains 132 images.

3.2 Evaluation

To quantitatively estimate the capacity and effectiveness of different models in extracting and decoding cloud features, four evaluation metrics were used. These were pixel accuracy, F1-score, and Intersection-over-Union (IoU).

3.2.1 F1 Score

The F1 score is based on the true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). From these parameters, precision, recall, and further, the F1 score can be computed [35]. The definition of precision and recall is the following;

$$Precision = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{(TP + FN)}$$

Then the F1 score can be calculated as following;

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The F1 score ranges for values between 0 and 1, and a value as close to 1 is desirable.

3.2.2 Intersection over Union

Intersection over Union (IoU), also known as the Jaccard index, is a measure of the overlap between two sets. It is a commonly used evaluation metric in object detection and semantic segmentation, as it is used to evaluate the quality of the model predictions compared to the true boundaries and masks of objects [36].

The intersection refers to the number of pixels that are correctly classified by both the model and the ground truth. It represents the number of pixels where the model prediction matches the actual object present in the image.

The union represents the number of pixels that are classified as an object by either the model or the ground truth. It encompasses all the pixels that are part of the object in either the prediction or the ground truth.

The IoU is computed using the following formula:

$$IoU = \frac{\text{Area of intersection}}{\text{Area of union}}$$

The formula quantifies the degree of overlap between the model prediction and the ground truth. A higher IoU score indicates a better segmentation result, as it would mean that there is a higher level of overlap. The IoU score ranges in values between 0 and 1, where 0 means no overlap between prediction and ground truth, and 1 indicates a perfect match. [36]

3.2.3 Pixel Accuracy

Pixel accuracy is a straight-forward evaluation metric for image segmentation task, that serves as an easy-to-understand metric that provides a quick assessment of the segmentation performance. However, its simplicity also comes with a drawback, as the pixel accuracy does not account for the spatial alignment between the ground truth and the predicted segmentation. This can be clearly seen when you have a class imbalance in the data with a lot of background. The pixel accuracy represents the ratio between correctly instances and all the instances in the dataset [35]:

$$\textit{Pixel Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

This can be translated into:

$$\textit{Pixel Accuracy} = \frac{\text{Number of correctly classified pixels}}{\text{Total number of pixels}}$$

3.3 Statistical testing

To validate our results, two different statistical tests were performed. A one-way analysis of variance (ANOVA) test to determine whether there was any statistical significance between the models, and a paired T test to explore where the significance lies between the models.

A one-way analysis of variance (ANOVA) is used to determine if there is a difference in the means of three or more groups and is one of the most commonly used statistical methods [37]. In other words, the null hypothesis in a comparison of three groups in an ANOVA test would be "the population means of three groups are all the same", and thus the alternative hypothesis would be "at least one of the population means of three groups is different". For this reason, the ANOVA test cannot be conducted by itself to draw any conclusion about the difference between the group; rather it tells us that there is at least one difference, but not where the difference exists. Therefore, a post-hoc test has to be conducted to observe where the significance lies [37].

To summarize, the following hypothesis can be described as follows;

$$H_0 : \mu_1 = \mu_2 = \mu_3$$

$$H_1 : \mu_1 \neq \mu_2 or \mu_1 \neq \mu_3 or \mu_2 \neq \mu_3$$

As the one-way ANOVA test only tells us whether there is a significant difference between the mean groups but not where, the ANOVA test must be followed by an ad hoc test. In this paper, a pairwise t-test with a level of α of 0.05 is used.

3.4 Software and tools

In this section, the software and tools used for this thesis are mentioned. As well as the implementation of the models and the experimental setup.

3.4.1 Data Labeling

To create the ground-truth mapping of our cloud images, the MIT LabelMe [38] tool was used. The tool makes it possible to draw and encircle relevant objects in the images that we want to label. The output from the tool is a JSON-file that we can later use with a script to create the binary images where the cloud pixels were assigned the value 255 (white) and the background were assigned the value 0 (black)

3.4.2 Environment

The majority of the experimental work was conducted on the Google Colab platform, leveraging the PRO edition for enhanced memory capacity and access to the A100 GPU, optimizing the training of deep learning models. Additionally, the data labeling process involved the use of the LabelMe tool, where ground-truth labels were generated as JSON-files. Some custom scripts were developed independently to handle specific tasks, such as converting these JSON files into binary images.

3.4.3 Cloud Cover Definition

The definition of cloud cover is defined by the World Meteorological Organization (WMO) as *"Total cloud cover is the fraction of the sky covered by all the visible clouds"*. To be able to use this definition for our purpose and use it for cloud images, we will say that the cloud cover in an image is the

fraction of clouds in an image. As cloud cover is measured in octaves, we take the floor of the mean of the cloud pixels and multiply it by 8.

$$\text{Cloud Cover} = \lfloor \frac{\text{\# Cloud Pixels}}{\text{\# Total Amount of Pixels}} * 8 \rfloor$$

3.4.4 Evaluation and Validation tools

This study aimed to evaluate the performance of the models using various metrics, including IoU, F1 score and pixel accuracy. To implement and use these evaluation metrics, functionalities of the Python library Torchmetrics were used. The results were transferred over to an Excel file, where the validation methods were used for statistical testing.

3.4.5 Implementation of models

In this study, three models based on deep learning were implemented and evaluated. The models chosen for this study were FCN, U-Net, DeepLabV3. Due to their frequent use in image segmentation as well as their popularity in image segmentation tasks, these models were chosen. A pretrained version of these models was utilized through the Python library PyTorch, a widely used open-source deep learning framework.

3.4.6 Fine-Tuning

Fine-tuning is a concept of transfer learning, a machine learning technique that involves using knowledge gained previously learned during training in one type of problem and then transferred to improve performance in another task [39]. In deep learning, the first layers are trained to identify the features of the task [39]. Therefore, during transfer learning, you can freeze layers and "re-train" the last few layers and adjust output layers to the task at hand. Fine-tuning is particularly valuable in scenarios where data is scarce, thus training a model from scratch is impractical. Additionally, training models using fine-tuning are much faster than training from scratch, and can also be more accurate [39]. As the availability of data is limited in this study, fine-tuning of the models will be crucial.

3.4.7 Experimental Setup & Hyperparameters

In the experimental setup, a standard 80-20 split of the data set was used for training and validation. Each model shared similar hyperparameter configurations to maintain consistency. The batch size was set to 8, the learning rate was set at 0.0001, and the training process spanned 10 epochs. Cross-entropy loss served as the chosen loss function for all models, and the Adam optimizer was consistently applied. It is important to note that this study did not dive into hyperparameter tuning or explore model training optimization techniques. The selected hyperparameter values were used uniformly across all models for the sake of simplicity and comparability as well as time constraints and resource constraints.

Chapter 4

Results and Analysis

In this chapter, the results of the experiments are mentioned. In the first section, the model performances are shown, both quantitatively and qualitatively. In the next section, we show the performance after the data-oversampling and -augmentation methods have been applied. In the third section, the validation of the relevant results are shown. In the fourth section, the difference in cloud cover estimation between a meteorologist and DeepLabV3 is shown. In the fifth section, we see if the knowledge learned from our dataset can be transferred to other datasets. Finally, in the last section, the findings of the key results are summarized.

4.1 Results of Different Models

This section addresses the first research question, found in 1.1.2, by analyzing the performance of various models on the cloud segmentation task. An assessment of the models' performance using the key evaluation metrics is given, to provide a comprehensive overview of each model's ability to accurately segment cloud cover in the provided test dataset.

4.1.1 Baseline Performance of Each Model Before Fine-Tuning

Before applying any fine-tuning specific to our cloud image datasets, it is essential to establish the baseline performance of the selected deep learning models: FCN, U-Net, and DeepLabV3. This baseline serves as a reference point, allowing us to measure the impact of subsequent fine-tuning and adaption to our specific task.

Baseline Performance Before Fine-tuning Models			
Model	IoU	F1	Pixel Accuracy
FCN	0.400	0.486	0.506
U-Net	0.173	0.268	0.183
DeepLabV3	0.000	0.000	0.24743

Table 4.1: Baseline performance metrics of pre-trained models, FCN, U-net, and DeepLabV3, on cloud segmentation tasks in terms of IoU, F1 Score, and Pixel Accuracy before fine-tuning.

Table 4.1 presents the baseline performance metrics - IoU, F1 score and Pixel Accuracy - achieved by each model in our test set, which consists of 73 images. These metrics provide an initial assessment of the models' ability to segment cloud cover accurately before any fine-tuning was applied.

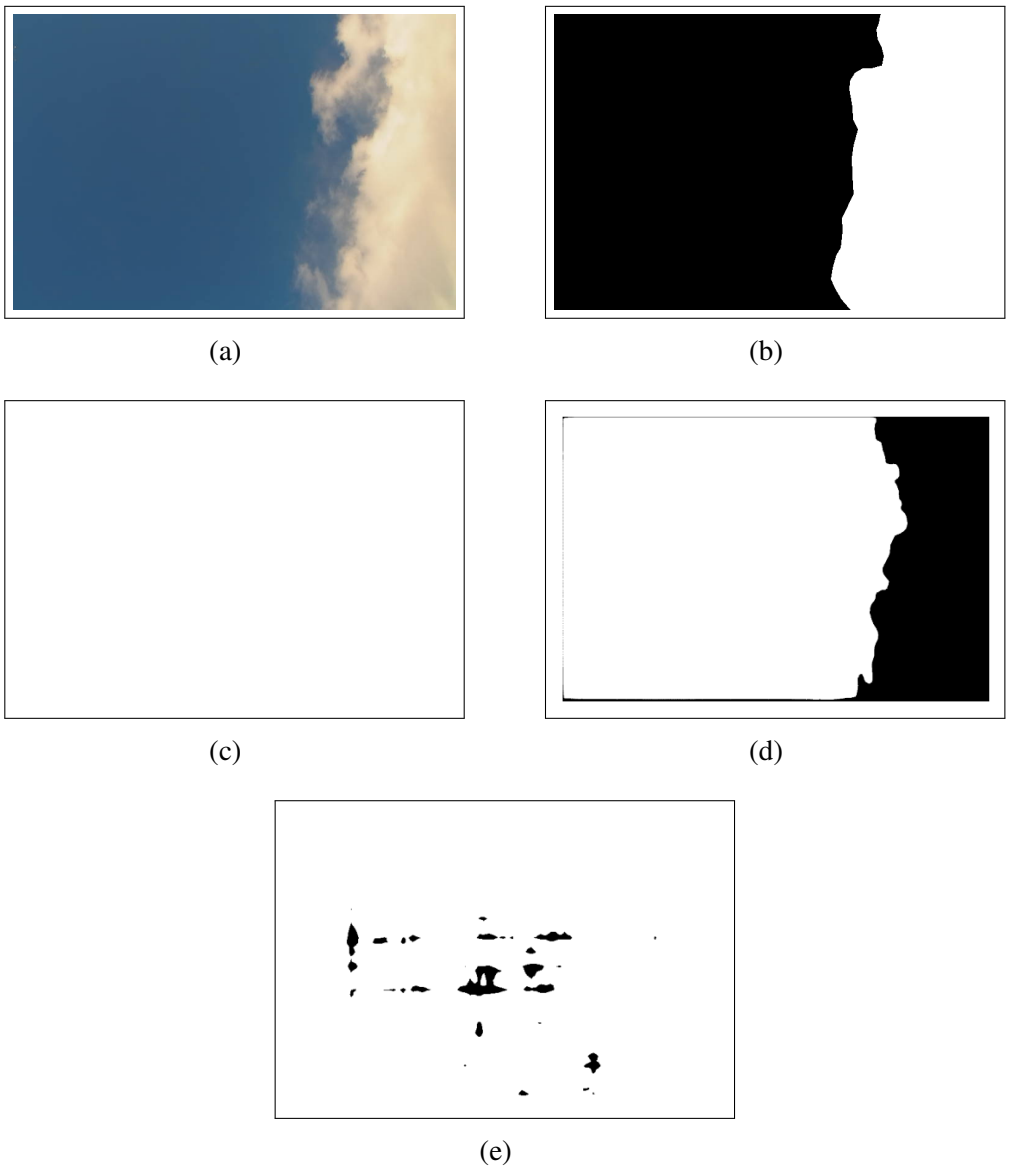
Figure 4.1 showcases an example image from our test set alongside its ground truth mapping and the segmentation outputs produced by each model prior to fine-tuning. The visual comparison highlights the initial capabilities and limitations of the models in handling cloud images.

4.1.2 Comparative Results for Cloud Segmentation Models

In this section, the results from the applied fine-tuning process to our deep learning models - FCN, U-Net, and DeepLabV3 - on the newly constructed datasets mentioned in Section 3.1.3 are presented. This analysis aims to highlight the improvements and capabilities of each model in performing cloud segmentation tasks.

Table 4.2 summarizes the performance metrics—Intersection over Union (IoU), F1 score, and Pixel Accuracy—for each model across our datasets, illustrating the enhancements achieved through fine-tuning. The datasets, labeled from 1 to 5, are configured as follows for comparative analysis:

- **Dataset 1:** Represents the original dataset with no modifications applied to the images, serving as the baseline for performance comparison.
- **Dataset 2:** An upsampled version of Dataset 1, augmented with random



(a) (b) (c) (d) (e)

Figure 4.1: Baseline segmentation outputs from pre-trained models: a) Original Image, b) Ground-Truth, c) FCN, d) U-Net, and d) DeepLabV3.

rotations to introduce variability in image orientation.

- **Dataset 3:** Similar to Dataset 2, this is an upsampled version of Dataset 1; however, the images are modified by applying random changes in brightness and contrast, testing model robustness against lighting variations.

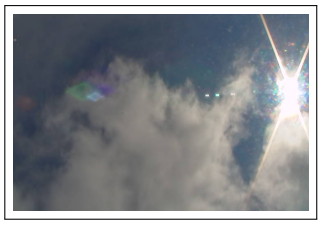
Performance on All Deep Learning Models						
Model	Dataset	Num Images	Run Time (Min)	IoU	F1	Pixel Accuracy
DeepLabV3	1	264	18	0.937	0.965	0.962
DeepLabV3	2	528	38	0.933	0.960	0.956
DeepLabV3	3	528	37	0.933	0.963	0.958
DeepLabV3	4	792	54	0.935	0.964	0.959
DeepLabV3	5	151	4	0.929	0.961	0.956
FCN	1	264	17	0.913	0.950	0.940
FCN	2	528	28	0.927	0.959	0.951
FCN	3	528	29	0.929	0.959	0.952
FCN	4	792	46	0.914	0.952	0.931
FCN	5	151	8	0.909	0.949	0.937
U-Net	1	264	5	0.874	0.925	0.918
U-Net	2	528	9	0.887	0.935	0.931
U-Net	3	528	8	0.889	0.935	0.933
U-Net	4	792	11	0.897	0.929	0.929
U-Net	5	151	3	0.863	0.925	0.920

Table 4.2: Performance Metrics for Deep Learning Models Across Multiple Datasets. This table displays the average results from five training runs for each model (DeepLabV3, FCN, U-Net) on five different datasets. Metrics include Run Time (minutes), IoU, F1 Score, and Pixel Accuracy. Green-highlighted values indicate the top four performance metrics in each category, while red highlights denote the four lowest scores in each category.

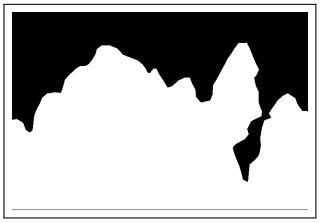
- **Dataset 4:** An upsampled composite of Dataset 1 that incorporates both types of modifications from Datasets 2 and 3 (rotations and brightness/contrast adjustments), designed to challenge the models with a combination of augmentations.
- **Dataset 5:** A downsampled version of Dataset 1, containing half the volume of images but maintaining an equivalent distribution of different cloud coverages, to assess model performance with reduced data availability.

To highlight some of the key differences found between the models from the output achieved when testing the models on our test set, visual examples are presented in figures 4.2, 4.3, and 4.4. These figures highlight key differences in the performance of the FCN, U-Net, and DeepLabV3 models:

- **Figures 4.2 and 4.3 - Images with Sun Rays (Noise):**

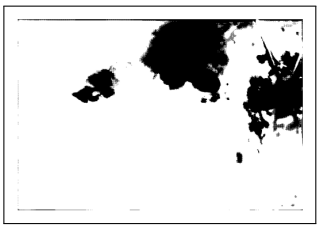


(a) Original Image



(b) Ground Truth Segmentation

(c) FCN Output



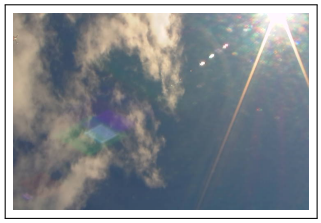
(d) U-Net Output

(e) DeepLabV3 Output

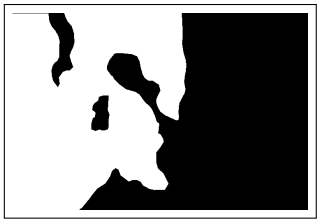
Figure 4.2: Comparative results for an example cloud image across different models. The original image (a) and its corresponding ground truth segmentation (b) are shown on the top row. The outputs of three deep learning models: FCN (c), U-Net (d), and DeepLabV3 (e) are presented on the bottom row.

  - **DeepLabV3 and FCN:** Both models handle the noise introduced by sun rays relatively well, maintaining a clear segmentation between cloud and sky.
  - **U-Net:** The U-Net model struggles significantly with images containing noise, such as sun rays. This results in misclassification, where parts of the cloud are incorrectly segmented as background, leading to a less accurate representation.

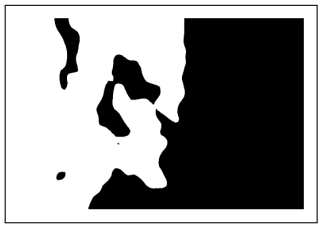
- **Figure 4.4 - Fully Covered Sky:**
  - **DeepLabV3 and FCN:** These models correctly identify the entire image as clouds, matching the expected segmentation.
  - **U-Net:** In this scenario, the U-Net model classifies certain pixels as background, even when the entire sky is covered with clouds.



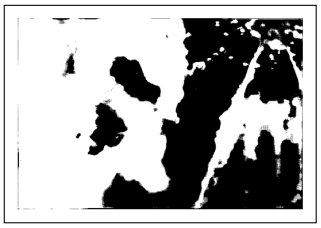
(a) Original Image



(b) Ground Truth Segmentation



(c) FCN Output

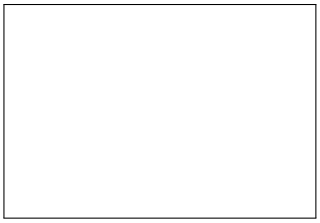


(d) U-Net Output

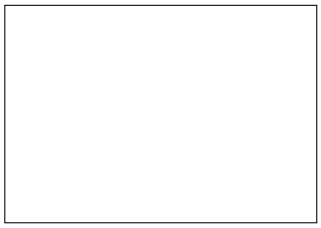
(e) DeepLabV3 Output

Figure 4.3: Comparative results for an example cloud image across different models. The original image (a) and its corresponding ground truth segmentation (b) are shown on the top row. The outputs of three deep learning models: FCN (c), U-Net (d), and DeepLabV3 (e) are presented on the bottom row.

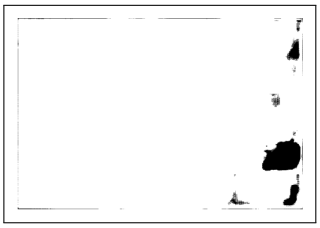
(a) Original Image



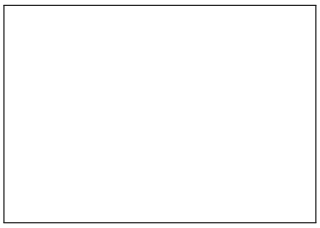
(b) Ground Truth Segmentation



(c) FCN Output



(d) U-Net Output



(e) DeepLabV3 Output

Figure 4.4: Comparative results for an example cloud image across different models. The original image (a) and its corresponding ground truth segmentation (b) are shown on the top row. The outputs of three deep learning models: FCN (c), U-Net (d), and DeepLabV3 (e) are presented on the bottom row.

4.1.3 Statistical Testing

To evaluate the effect of data volume on model performance, we first conducted a one-way ANOVA test on the performance metrics—IoU, F1 score, and Pixel Accuracy—across different dataset sizes for each model. Each model was independently tested and no comparisons were made between the models (for example, DeepLabV3 trained on dataset 1 was not compared to FCN trained on dataset 2). The null hypothesis was that the volume of data did not have a significant impact on the performance of the model. The tests were carried out with 4 degrees of freedom between groups and 20 degrees of freedom within groups. The significance level for the test was set at = 0.05

The results indicated that, for all metrics except the IoU score for the U-Net model, data volume did not significantly affect performance ($p > 0.05$). As a result, we were unable to reject the null hypothesis that data volume had no significant impact on performance for most tests. The results of this can be found in table 4.3.

Next, we performed an ANOVA test to compare the performance of the models on the original dataset, since data volume did not significantly affect model performance. The ANOVA tests were conducted with 2 degrees of freedom between groups and 12 degrees of freedom within groups with a significance threshold set at $\alpha = 0.05$. This test revealed statistically significant differences in performance between models in all metrics. The resulting p-values were **4.88e-07** for IoU, **1.62e-06** for F1 Score, and **9.15e-06** for Pixel Accuracy. This test revealed statistically significant differences in performance between models in all metrics.

Following these results, we conducted a post-hoc pairwise t-tests to compare the models' performance on the original dataset. These t-tests helped identify which specific model pairs exhibited statistically significant performance differences. The results can be found in table 4.4.

Model	IoU	F1 Score	Pixel Accuracy
DeepLabV3	0.861	0.948	0.882
U-Net	0.015	0.753	0.558
FCN	0.702	0.802	0.405

Table 4.3: P-values from one-way ANOVA tests conducted separately for each model across datasets of varying sizes. The analysis was performed to determine whether data volume had a significant impact on each model's performance.

Metric	Comparison	DeepLabV3	FCN	U-Net
IoU	DeepLabV3	-	0.006	$2.58e^{-6}$
	FCN	0.006	-	0.0005
	U-Net	$2.58e^{-6}$	0.0005	-
F1 Score	DeepLabV3	-	0.015	$1.04e^{-6}$
	FCN	0.015	-	0.0014
	U-Net	$1.04e^{-6}$	0.0014	-
Pixel Accuracy	DeepLabV3	-	0.008	$4.96e^{-6}$
	FCN	0.008	-	0.006
	U-Net	$4.96e^{-6}$	0.006	-

Table 4.4: P-values from post-hoc paired t-tests comparing the performance of DeepLabV3, FCN, and U-Net models across three metrics: IoU, F1 Score, and Pixel Accuracy. Each comparison reflects the significance of the differences between the models.

4.2 Best performing model vs Meteorologist cloud cover estimation

To see if our definition of cloud cover aligns with the definition covered in Section 3.4.3, we used the DeepLabV3 model to segment images and calculate cloud cover based on segmentation. We then compared our result with the estimation made of the cloud cover on those images by a meteorologist from the Swedish Meteorological and Hydrological Institute. The results are shown in Table 4.5. Cloud cover is measured in octaves, where 0 means that there are no clouds and a score of 8 means that the sky is fully covered by clouds.

Based on these figures in table 4.5, of 20 images, nine images were correctly classified, eight images were classified with an error of one octave, two images were classified with an error of two octaves, and one image was classified with an error of more than two octaves. To note, is that on some images, the meteorologist made some assumptions about the cloud which were taken into account when making the estimation. These were for images 7, 8, 16, and 17. Out of these four images, one image was classified correctly, image 8, two images were classified with an error of 1, images 16 and 17, and one image was classified with an error of two, image 7.

Cloud Cover Estimation			
Image Number	Model Estimate	Meteorologist Estimate	Error in Octaves
1	4	4	0
2	4	3	1
3	3	2	1
4	6	6	0
5	4	7	3
6	3	2	1
7	4	6	2
8	6	6	0
9	8	8	0
10	7	6	1
11	5	5	0
12	6	6	0
13	5	4	1
14	3	2	1
15	7	7	0
16	1	2	1
17	3	2	1
18	4	4	0
19	5	5	0
20	4	2	2

Table 4.5: Cloud Cover Estimations between DeepLabV3 Segmentation and Meteorologist

4.3 Generalization - Performance on other dataset

To assess the generalization capability of the models trained during our study, we evaluated their performance on two publicly available cloud image datasets: SWIMSEG and SWINSEG. Specifically, we tested DeepLabV3, FCN, and U-Net, which were all trained on dataset 1 from our primary experiments.

Given that the previous statistical analysis revealed no significant differences between models trained on different datasets, only the models trained on dataset 1 were included in this experiment. This approach ensures that we are testing the most representative version of each model.

The images from the SWIMSEG and SWINSEG datasets were fed into the trained models, and the respective performance metrics—IoU, F1 Score, and Pixel Accuracy—were computed. The results for each model are shown in Table 4.6.

Model	**Dataset**	**IoU**	**F1**	**Pixel Accuracy**
DeepLabV3	SWIMSEG	0.518	0.646	0.716
FCN	SWIMSEG	0.279	0.372	0.660
U-Net	SWIMSEG	0.082	0.118	0.554
DeepLabV3	SWINSEG	0.504	0.648	0.554
FCN	SWINSEG	0.506	0.653	0.567
U-Net	SWINSEG	0.555	0.601	0.452

Table 4.6: Performance results of models trained on Dataset 1 across SWIMSEG and SWINSEG datasets for three metrics: IoU, F1 Score, and Pixel Accuracy.

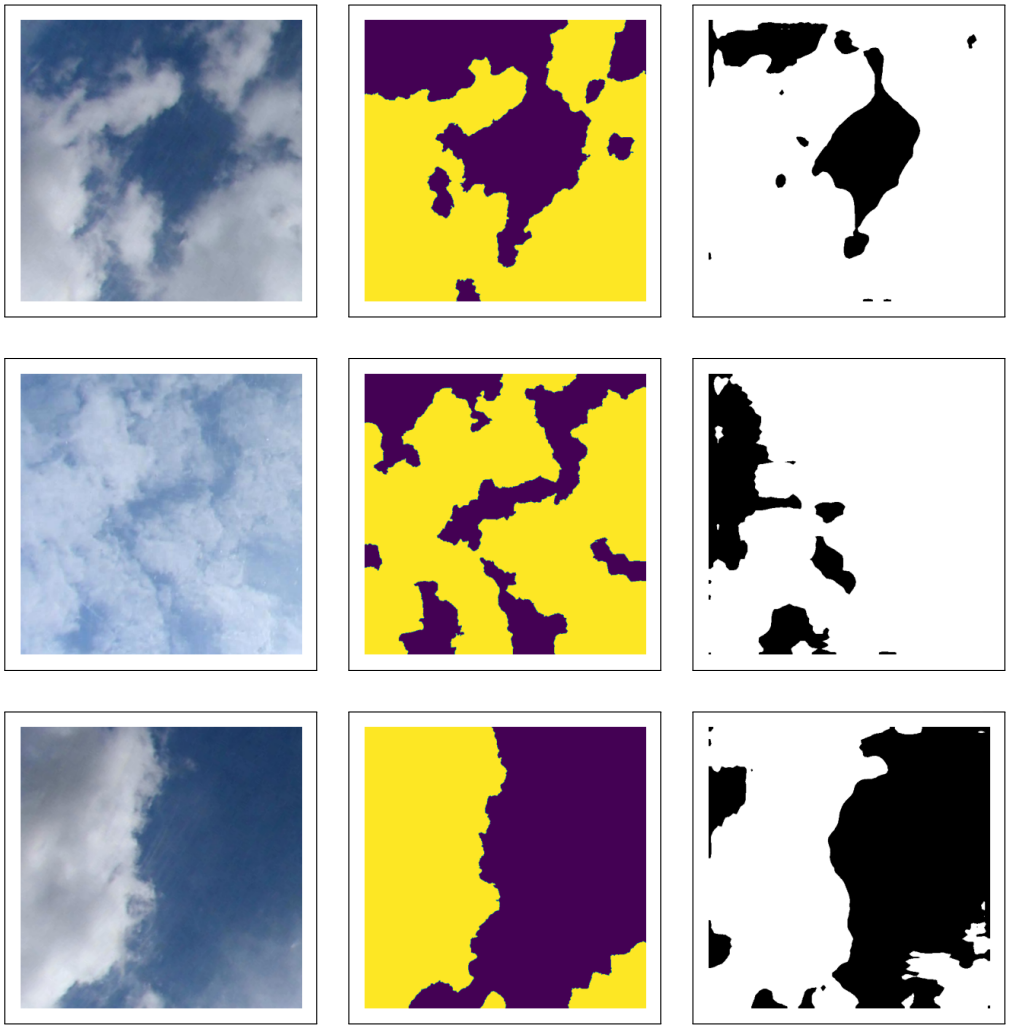


Figure 4.5: Segmentation output from DeepLabV3 on example images from the SWIMSEG dataset. Each row shows the original image (left), the ground truth mapping (middle), and the segmentation output from DeepLabV3 (right).

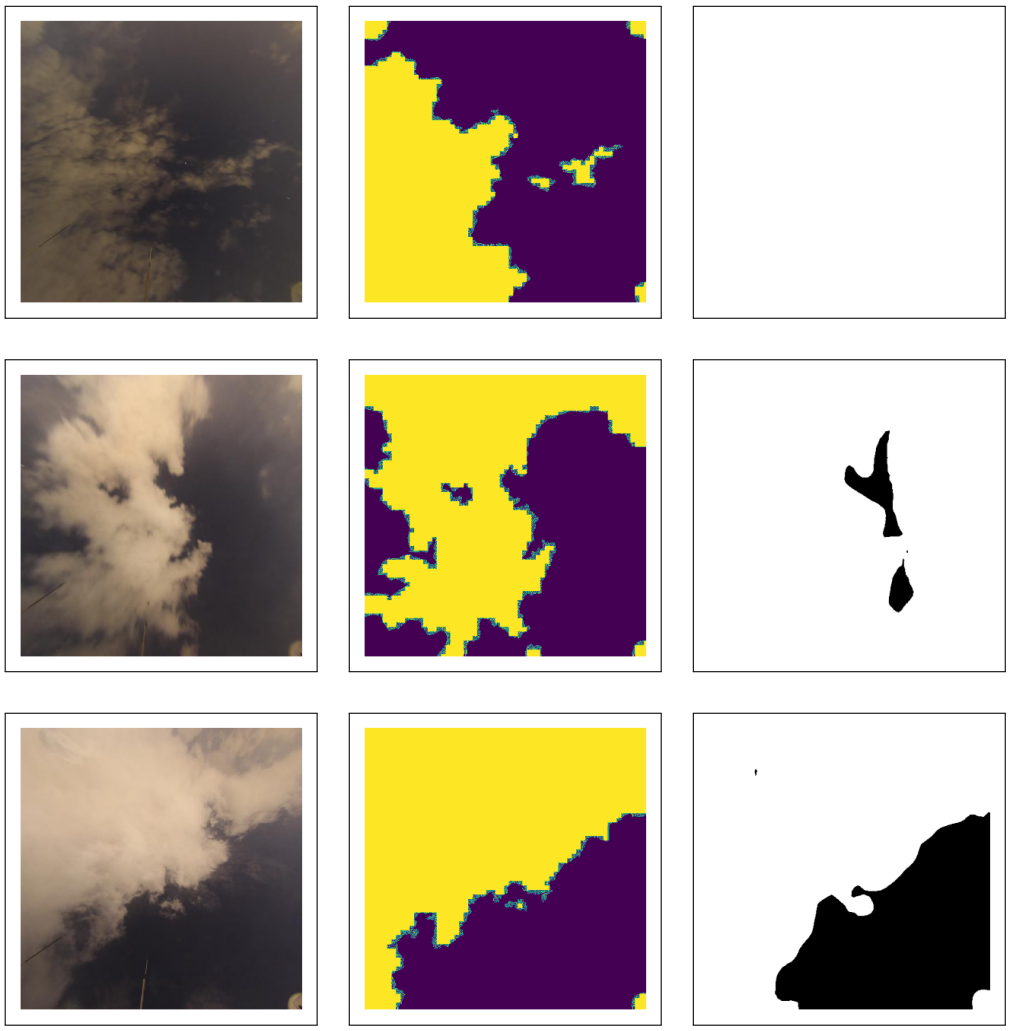


Figure 4.6: Segmentation on SWINSEG images from DeepLabV3. Each row shows the original image (left), the ground truth segmentation (middle), and the model's segmentation output (right).

4.4 Summary of Key Findings

This section provides an overview of the key findings of the experiments conducted. The following summary highlights the performance of the models, the impact of data augmentation and data volume, their effectiveness in predicting cloud cover compared to meteorologists' predictions, and the ability of the models to generalize to external datasets.

4.4.1 Model Performance Across Metrics

DeepLabV3 consistently outperformed both FCN and U-Net across all three peformance metrics; IoU, F1 Score, and Pixel Accuracy. For example, on the test dataset, DeepLabV3 achieved an IoU of 0.937, an F1 score of 0.965 and a Pixel precision of 0.962, which was significantly higher than FCN (IoU; 0.913, F1 Score; 0.950, Pixel Accuracy 0.940) and U-Net (IoU; 0.874, F1 Score; 0.925, Pixel Accuracy 0.918). The statistical tests confirmed the statistical significance in both the ANOVA test, where the p value was lower than the significance level in all metrics as shown in the table **??** and was further supported by the paired t tests, where the p value for IoU between DeepLabV3 and U-Net was $2.58e^{-6}$, and between DeepLabV3 and FCN was 0.006 (see Table 4.5). Similarly, for Pixel Accuracy, DeepLabV3 significantly outperformed U-Net ($p = 4.96e^{-6}$) and FCN ($p = 0.008$), further validating the superiority of DeepLabV3.

4.4.2 Impact of Data Augmentation and Data Volume

The results indicate that the augmentation of the data contributed to marginal improvements in the performance of the model. However, the ANOVA test performed on models trained with varying dataset sizes showed no statistically significant difference in performance ($p > 0.05$), suggesting that the volume of training data had a lesser effect on model precision compared to model architecture.

4.4.3 Cloud Cover Prediction Based on DeepLabV3 segmentation

To observe whether our definition of cloud cover is in line with the definition of cloud cover in the meteorology world, we used the segmentation of the DeepLabV3 model to compute the cloud coverage in an image and

then compared that to the estimate made of the cloud coverage from a meteorologist. Out of 20 images, we found that in 17 cases, our model had a maximum error of 1 octave, indicating that our definition is well aligned with the real world.

4.4.4 Generalization on External Datasets

When evaluating the models trained on Dataset 1 on external datasets (SWIMSEG and SWINSEG), the results indicated that the knowledge gained during training did not transfer well to these datasets. The performance scores, particularly for U-Net and FCN, were significantly lower than those observed on the original dataset. For example, U-Net achieved an IoU of only 0.082 on the SWIMSEG dataset and 0.555 on SWINSEG, which are unsatisfactory for practical cloud cover segmentation tasks. Similarly, FCN's IoU was 0.279 on SWIMSEG and 0.506 on SWINSEG, also showing poor generalization. Even DeepLabV3, the best performing model on our own dataset, showed a significant drop in performance, achieving an IoU of 0.518 on SWIMSEG and 0.504 on SWINSEG. The F1 and Pixel Accuracy scores also followed similar patterns.

Chapter 5

Discussion

The aim of this study was to evaluate deep semantic segmentation models on cloud images, to later utilise them for predicting cloud coverage. This chapter begins with an examination of the results, followed by revisiting and answering the research questions.

5.1 Restating the research problem

Ground-based cloud observations face challenges such as inaccuracy, time consumption, resource intensiveness, and high costs. Previous research has highlighted the potential of leveraging deep learning methods to address these issues, offering accurate and efficient cloud observations, including cloud cover prediction and various applications. This study aims to evaluate three state-of-the-art deep semantic segmentation models, demonstrating their viability in the realm of cloud observation. The goal is to serve as a catalyst for transforming current practices in ground-based cloud observations, paving the way for a more effective and resource-efficient approach.

5.2 Revisiting Research Questions and Hypotheses

In this section, we will be revisiting the research questions and hypotheses stated in Section 1.1.2 this paper and discuss how the results relate to each.

Research question 1: *'What are the advantages and disadvantages of deep learning-based semantic segmentation methods for cloud cover prediction in terms of predictive performance, efficiency, and implementation complexity?'*

Upon evaluating three prominent semantic segmentation models: FCN, U-Net, and DeepLabV3—after fine-tuning in our training set, all models demonstrated remarkable precision in segmenting cloud images. Evaluation metrics consistently scored above 0.87 for all models, confirming their proficiency in correctly segmenting the test images. Segmentation output further substantiated these findings, providing visual evidence of the effectiveness of the models.

Despite the commendable performance of all three models, distinctions emerged during the qualitative analysis. FCN and DeepLabV3 exhibited superiority over U-Net. The qualitative assessment revealed that U-Net displayed a higher sensitivity to noise, rendering images with noise more susceptible to poor generalization. This observation underscores a potential limitation of U-Net in handling noisy input.

The superior performance of FCN and DeepLabV3 suggests their suitability in terms of efficiency and implementation complexity. Notably, these models exhibited robustness against noise without necessitating major modifications. In contrast, U-Net's sensitivity to noise indicates a potential need for additional adaptations to enhance its noise resilience.

The choice between these models becomes crucial when considering real-world implementation and operational efficiency. The resilience of FCN and DeepLabV3 to noise positions them as favorable options, requiring minimal modifications to handle challenging image conditions. On the other hand, U-Net's heightened sensitivity, rooted in its original development for medical image segmentation, raises considerations about its adaptability to diverse environmental conditions.

It's important to note that while U-Net may excel in certain contexts, such as medical imaging, the specific requirements for cloud observation demand models capable of handling various atmospheric conditions and potential image distortions. This study provides valuable information on the comparative strengths and weaknesses of these models, helping inform decision making for practical applications in cloud cover prediction.

Research question 2: *'How does the accuracy of cloud cover prediction differ between deep learning based methods and manual observations made by meteorologists in the Swedish cloud observation process?'*

In pursuit of demonstrating the utility of semantic segmentation models for cloud cover prediction, we conducted a comparative analysis between cloud cover predictions generated from segmented images and those derived from manual observations by meteorologists at the Swedish Meteorological and Hydrological Institute. Since the models do not provide a direct cloud cover score, we employed our defined metric – *'the amount of sky covered by clouds'* – computed by evaluating the fraction of cloud pixels in the images. The meteorologist's predictions served as the ground truth for our assessment.

Out of the twenty images evaluated, the models accurately predicted the cloud cover in nine instances. Eight images were predicted with an error of one octave, two images with an error of two octaves, and one image with an error exceeding two octaves. Most of the predictions were correct or exhibited a minimal error of one octave, suggesting that semantic segmentation models provide accurate cloud cover predictions.

However, it is crucial to recognize that these results, while promising, cannot serve as a definitive conclusion. Further experiments are warranted to refine our understanding of the models' performance.

Research question 3: *What is the impact of dataset volume on the performance of state-of-the-art deep semantic segmentation models? Specifically, what is the minimum volume of data sets required to achieve accurate segmentation, and how does increasing the volume of data sets further enhance model performance?*

To assess the influence of dataset volume on semantic segmentation models, we curated a dataset comprising 264 cloud images as our baseline. To investigate the effects of data volume on model performance, we employed data enhancement, up-sampling, and down-sampling techniques, as detailed in Section 3.1.3. Following independent experiments with the three models, the results indicated that modifications to our dataset did not exert significant effects on model performance.

Considering these findings, one might infer that the original dataset volume of 264 images suffices when employing fine-tuning techniques. Several factors contribute to this conclusion. Firstly, in semantic segmentation, each pixel serves as a data point, resulting in a substantial number of data points even with a limited number of images. Given our image dimensions of 720x480 pixels, each image yields 345.600 pixels, translating to an equivalent number of labeled data points. With our baseline dataset volume, training on our dataset provides approximately 91.5 million data points.

Secondly, the homogeneity of our dataset should be acknowledged. Constructed solely from images captured by a single camera at a fixed location, our dataset exhibits uniformity in terms of contrast, brightness, and colors. This uniformity proved beneficial in training models to effectively discern clouds from the background within our set-up. However, when testing the models on diverse public datasets, we observed challenges in consistently separating clouds in those images.

These observations underscore the nuanced impact of the volume of the dataset. Although our baseline dataset proves to be effective for fine-tuning and demonstrates the potential of semantic segmentation models, it is essential to recognize its limitations when applied to more diverse data sets. Future work could explore strategies to enhance model generalization, such as incorporating additional diverse images in the training dataset or exploring advanced transfer learning techniques. Additionally, assessing the models' performance across varying atmospheric conditions and camera setups could contribute to a more comprehensive understanding of their capabilities and limitations.

5.3 Potential Impact of Future Cloud Observations

Cloud observation is at the forefront of meteorological research, playing a pivotal role in understanding and predicting weather patterns. As technology continues to advance, there is a growing need for innovative approaches that can enhance the accuracy, efficiency, and automation of cloud observation processes. In this context, the findings of from our study show the potential of reshaping the landscape of cloud observation methodologies.

Our study, while focused on the prediction of cloud coverage, has demonstrated promising applications of state-of-the-art semantic segmentation models. These models showcase the ability to accurately separate clouds from the background in images, laying the groundwork for future cloud cover estimation methodologies.

The introduction of machine learning methodologies into this field presents numerous possibilities. Firstly, it has the potential to reduce the reliance on human resources by streamlining operational processes. Secondly, machine learning can potentially complement existing cloud observation technologies, providing more accurate and nuanced observations. Thirdly, in the estimation of cloud cover, these models could enable real-time predictions, a capability lacking in current technologies like ceilometers. However, these possibilities come with their own set of challenges. The ability to scale up and generalize machine learning solutions is crucial. While our results demonstrated strong performance on our dataset, challenges arose when applied to other available datasets, posing a potential obstacle in commercial settings. Additionally, maintaining the quality of the images is paramount, necessitating a dedicated team to ensure proper camera operation throughout all seasons. Cost considerations, encompassing not just hardware but also platforms for training, deployment, and maintenance, add another layer of complexity.

This field, while promising, remains in its infancy. As we delve into potential future work in the next chapter, it becomes evident that much research and evaluation are required to determine the feasibility and widespread adoption of AI and ML solutions in cloud observation. Acknowledging these challenges and embracing ongoing research endeavors will be crucial to realize the full potential of these innovative approaches.

5.4 Limitations

This section discusses potential limitations that could have affected the study results.

5.4.1 Limitations of Dataset

As our dataset that was used for this study was collected during the course of the project, some limitations could have affected the experimental results

in the end. First, our data set was manually labeled, which introduced the possibility of errors in pixel-wise classification. The inherent limitation arises from the difficulty in the human eye discerning every pixel accurately, leading to instances where pixels may be misclassified as cloud or non-cloud. The precision of the manual labeling process also influences the granularity of the segmentation achieved by the models. This variability in precision could affect the quantitative performance metrics. Furthermore, the subjective nature of manual labeling introduces a noteworthy consideration. Depending on the labeler's interpretation, the ground truth may exhibit variations in precision and detail. Consequently, the segmentation produced by the models, although accurate, may appear to be more refined than the manually labeled ground truth. This discrepancy poses a challenge for quantitative metrics as the model's output may differ slightly from the ground truth. Consequently, this discrepancy could potentially result in lower quantitative performance scores, not reflective of the model's actual segmentation capability. In light of these limitations, it is important to supplement quantitative measures with qualitative assessments. Qualitative evaluations provide a nuanced understanding of the segmentation quality, acknowledging potential differences between manual ground truth and model output. This recognition is essential for a comprehensive evaluation of the performance of the models and an informed interpretation of the results.

The homogeneity of our dataset, coupled with its absence of noisy images, significantly impacted the robustness and generalization capabilities of the trained models. Given the study's primary focus on evaluating how well semantic segmentation models handle cloud images within the constraints of data volume, the selected images shared similarities in terms of colors, contrast, brightness, and various other features. Additionally, due to the limitations of our camera setup, only daytime images made their way into the dataset, excluding nighttime captures.

This design choice became evident in the models' performance. They excelled at segmenting images captured by our specific camera setup, but faced challenges when confronted with images from different sources. A notable limitation stemmed from the absence of intentionally noisy images in our dataset. For example, images captured on rainy days that feature rain droplets that create significant aberrations were not included. These types of image are crucial for exploring the model's resilience in diverse and challenging weather conditions. The omission of such variations becomes particularly significant

when considering the demands of real-world applications. Inclusion of diverse images, with variations in weather conditions and environmental factors, is instrumental for enhancing the robustness and generalization capabilities of machine learning solutions. This limitation underscores the need for future research to explore the impact of diverse, noisy data on the performance of semantic segmentation models in cloud observation scenarios.

5.4.2 Hyperparameter Tuning and Model Training

Hyperparameter tuning, or hyperparameter optimization, is a set of methods to optimize hyperparameters in order to increase the performance of the machine learning model for certain tasks [40]. In this study, default settings were used and no hyperparameter optimization methods were applied. The decision to forgo hyperparameter optimization was grounded in practical considerations. Implementing optimization techniques can be time-intensive, particularly since optimal hyperparameter values often hinge on the specific model architecture.

In this study, the data set was divided into 80-20 splits for training and validation data, with a separate test set reserved for the evaluation of the trained models. The choice of this split ratio is a critical aspect of model training, as it influences the model's ability to generalize to new, unseen data. It is essential to recognize that different splits of the data during the training phase can lead to variations in model performance. One approach to mitigate the influence of a specific split is to employ K-fold cross-validation, a widely used technique in machine learning. K-fold cross-validation involves dividing the dataset into K folds, training the model on K-1 folds, and validating it on the remaining fold. This process is repeated K times, each fold serving as the validation set exactly once. The performance metrics are then averaged over the K iterations [41]. The employment of similar methods could be further investigated in order to achieve desirable performance.

5.5 Future work

This study has demonstrated the effectiveness of image processing techniques, specifically semantic segmentation, in predicting cloud cover. While the results are promising, there remain plenty of opportunities for further contributions within this domain.

1. Dataset Extensions for Generalization and Robustness:
To address the limitations of this study, research exploring the effects of enhancing the datasets is of importance. Enhancements could include expanding the dataset to include images that cover larger areas, incorporating nighttime images, and introducing more varied noise levels. Strengthening generalization and robustness is essential, particularly when considering the commercialization of the solution.

2. Hyperparameter Optimization:
As highlighted in the limitations, exploring hyperparameter optimization is a logical next step to improve the study outcomes. This step aims to identify optimal combinations of hyperparameters, potentially leading to refined model performance.

3. Cloud Coverage Prediction Model:
An intriguing improvement would be the development of a model capable of directly predicting cloud coverage. This approach required an investment in creating a specialized dataset, assigning scores to each image for accurate training. Adjustments to the model architecture are necessary for the model to be able to classify the score correctly. Potential enhancements could involve incorporating data from other cloud analysing tools. For instance, in Sweden, the ceilometer is the current tool used for cloud cover prediction; perhaps the data from a ceilometer could be utilized for more accurate cloud cover prediction.

4. Weather and Seasonality Analysis:
Recognizing the correlation between cloud and weather patterns across seasons, future work could focus on analyzing weather and seasonality for improved cloud cover predictions. Techniques that incorporate the temporal aspect and capture seasonal patterns could offer insights for real-time analysis and future cloud projections. The potential of this solution extends beyond predicting cloud coverage; it opens avenues for forecasting various cloud types based on their characteristic patterns and behaviors in different seasons. This broader approach not only enhances our understanding of cloud dynamics but also contributes to a more comprehensive and nuanced cloud observation analysis.

5. Understanding Deep Learning Models:
Addressing the common challenge of deep learning models acting as black

boxes, future investigations should focus on understanding the learning process of image processing models. Drawing inspiration from studies such as [42], which visualizes and understands CNNs, an exploration into how CNN models learn properties could provide deeper insights, minimizing the black box effect and enhancing interpretability.

These potential directions for future work will not only build upon the current study but also pave the way for advancements in cloud observation methodologies. Each suggestion aligns with the overarching goal of automating and enhancing cloud observation methodologies, aiming to make them more accurate and applicable in diverse real-world scenarios.

5.6 Sustainability Aspect

The computational demands of machine learning solutions often translate into significant resource consumption, which can contribute to an increased environmental carbon footprint. Given the commercial focus of this study, it is imperative to consider the environmental impact of different computational solutions. Opting for energy-efficient hardware and leveraging cloud-computing platforms with green computing initiatives are essential strategies to mitigate the ecological effects of machine learning implementations.

In terms of data ethics, this study does not use sensitive data. However, as this field advances, future investigations must carefully navigate the legal landscape concerning image capture, permissible photography areas, and privacy regulations. Especially in commercial deployments, understanding and complying with laws and regulations related to data collection are crucial considerations.

5.7 Ethical Considerations

Ethical concerns extend to the interpretability of machine learning models, particularly the black box nature of deep learning algorithms. For widespread commercial adoption, transparency is crucial. Future studies should examine the workings of these models, seeking to demystify their decision-making processes. Employing solutions that lack explainability may be viewed unfavorably, and efforts to enhance transparency will contribute to the ethical

integrity of the technology.

Lastly, as previously mentioned in the section about impact, it is crucial to delve deeper into the considerations surrounding maintenance and hardware upgrades in the context of environmental sustainability. In the lifecycle of machine learning solutions, ongoing maintenance stands as a linchpin for sustained efficiency and optimal performance. Regular monitoring, periodic system health checks, and responsive software updates not only ensure the reliability of the system, but also contribute significantly to resource efficiency. By emphasizing the importance of maintenance, strategic hardware upgrades, and a comprehensive approach to lifecycle assessment, organizations can effectively contribute to the responsible and sustainable development and deployment of machine learning solutions. These efforts underscore a commitment to environmental stewardship within the evolving landscape of technology.

Chapter 6

Conclusions

The goal of this study was to demonstrate the potential of deep learning in the area of cloud observation analysis - more specifically, to explore the capabilities of deep learning models in predicting cloud cover from images. To align with the goal, three research questions were initially stated in this thesis. To summarize them, the research questions asked about the general performances of the used models, as well as their ability to accurately predict cloud cover by comparing the results with predictions made by a human observer. Lastly, this study explored the effects of data volume, and how that affected model performance.

The experiments conducted with the three models, FCN, U-Net, and DeepLabV3, showed that the models were all capable of learning and able to separate cloud from the background, achieving accurate segmentations of the images. Although all three generated good results, the DeepLabV3 model achieved the best results, while U-Net showed the worst result. Although the model performed well with our test set, when testing the models on other unseen data, we saw a drastic reduction in performance which could possibly be due to the homogeneity of our dataset.

When comparing the model output and the prediction of a human observer, around 85% of the images had the same prediction as the human observer or an error of 1 octave, indicating that the segmentation and the definition used for cloud cover was well aligned with the real world application. These observations suggest that deep learning can indeed be a viable approach for the task of predicting cloud cover from images.

To answer the final research questions, experiments were performed that included changes in the data set. The results showed that the modifications made by our side, including different augmentations as well as over and undersampling, did not have any significant effect on the performances. This indicates that the original size of 264 images was well suited for the approach of this study. However, this cannot be used as conclusive evidence and more exploration in this area is encouraged.

In conclusion, this study has shown that deep learning is indeed a viable method for the prediction of cloud cover in the cloud observation process. However, it is important to note that these results can not be used as conclusive evidence and more research has to be made before AI can be deployed in this field. Further research is therefore highly encouraged, as advances within this field could result in a significant performance increase when it comes to predicting cloud cover, as well as other improvements, such as better real-time predictions in terms of speed and accuracy, as well as having the potential to be used in the other fields of the cloud observation analysis.

References

[1] Z. Zhou, F. Zhang, H. Xiao, F. Wang, X. Hong, K. Wu, and J. Zhang, "A novel ground-based cloud image segmentation method by using deep transfer learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021. [Page 1.]

[2] S. Liu, J. Zhang, Z. Zhang, X. Cao, and T. S. Durrani, "Transcloudseg: Ground-based cloud image segmentation with transformer," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 6121–6132, 2022. [Pages 1, 2, 3, and 7.]

[3] L. Ye, Z. Cao, and Y. Xiao, "Deepcloud: Ground-based cloud image categorization using deep convolutional features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 10, pp. 5729–5740, 2017. [Pages 1, 2, and 7.]

[4] C. N. Long, J. M. Sabburg, J. Calbó, and D. Pagès, "Retrieving cloud characteristics from ground-based daytime color all-sky images," *Journal of Atmospheric and Oceanic Technology*, vol. 23, no. 5, pp. 633–652, 2006. [Page 2.]

[5] A. Heinle, A. Macke, and A. Srivastav, "Automatic cloud classification of whole sky images," *Atmospheric Measurement Techniques*, vol. 3, no. 3, pp. 557–567, 2010. [Page 2.]

[6] C. Shi, Y. Wang, C. Wang, and B. Xiao, "Ground-based cloud detection using graph model built upon superpixels," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 719–723, 2017. [Page 2.]

[7] C. Shi, Y. Zhou, B. Qiu, D. Guo, and M. Li, "Cloudu-net: A deep convolutional neural network architecture for daytime and nighttime cloud images' segmentation," *IEEE Geoscience and Remote Sensing Letters*, vol. 18, no. 10, pp. 1688–1692, 2020. [Page 2.]

[8] C. Shi, Y. Zhou, and B. Qiu, "Cloudu-netv2: A cloud segmentation method for ground-based cloud images based on deep learning," *Neural Processing Letters*, vol. 53, no. 4, pp. 2715–2728, 2021. [Pages 2, 18, 19, 20, and 21.]

[9] S. Dev, A. Nautiyal, Y. H. Lee, and S. Winkler, "Cloudsegnet: A deep network for nychthemeron cloud image segmentation," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 12, pp. 1814–1818, 2019. [Page 2.]

[10] W. Xie, D. Liu, M. Yang, S. Chen, B. Wang, Z. Wang, Y. Xia, Y. Liu, Y. Wang, and C. Zhang, "Segcloud: A novel cloud image segmentation model using a deep convolutional neural network for ground-based all-sky-view camera observation," *Atmospheric Measurement Techniques*, vol. 13, no. 4, pp. 1953–1961, 2020. [Page 2.]

[11] Z. Zhou, F. Zhang, H. Xiao, F. Wang, X. Hong, K. Wu, and J. Zhang, "A novel ground-based cloud image segmentation method by using deep transfer learning," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021. [Pages 2, 3, 21, and 22.]

[12] A. Taravat, F. Del Frate, C. Cornaro, and S. Vergari, "Neural networks and support vector machine algorithms for automatic cloud classification of whole-sky ground-based images," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 3, pp. 666–670, 2015. doi: 10.1109/LGRS.2014.2356616 [Pages 7 and 8.]

[13] W. K. Chen, *The electrical engineering handbook*. Elsevier, 2004. [Page 8.]

[14] B. M. Dawant and A. P. Zijdenbos, "Image segmentation," *Handbook of medical imaging*, vol. 2, pp. 71–127, 2000. [Page 8.]

[15] Ç. Kaymak and A. Uçar, "A brief survey and an application of semantic image segmentation for autonomous driving," *Handbook of Deep Learning Applications*, pp. 161–200, 2019. [Page 8.]

[16] D. Kaur and Y. Kaur, "Various image segmentation techniques: a review," *International Journal of Computer Science and Mobile Computing*, vol. 3, no. 5, pp. 809–814, 2014. [Page 9.]

[17] G. Csurka, D. Larlus, F. Perronnin, and F. Meylan, "What is a good evaluation measure for semantic segmentation?." in *Bmvc*, vol. 27, no. 2013. Bristol, 2013, pp. 10–5244. [Page 9.]

[18] S. Hao, Y. Zhou, and Y. Guo, "A brief survey on semantic segmentation with deep learning," *Neurocomputing*, vol. 406, pp. 302–321, 2020. [Page 9.]

[19] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4. [Pages 9 and 10.]

[20] B. Liu and B. Liu, "Supervised learning," *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, pp. 63–132, 2011. [Page 10.]

[21] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature machine intelligence*, vol. 1, no. 5, pp. 206–215, 2019. [Page 10.]

[22] C. C. Aggarwal *et al.*, "Neural networks and deep learning," *Springer*, vol. 10, no. 978, p. 3, 2018. [Page 11.]

[23] A. M. Braşoveanu and R. Andonie, "Visualizing transformers for nlp: a brief survey," in *2020 24th International Conference Information Visualisation (IV)*. IEEE, 2020, pp. 270–279. [Page 11.]

[24] Y. Li, "Deep reinforcement learning: An overview," *arXiv preprint arXiv:1701.07274*, 2017. [Page 11.]

[25] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *CoRR*, vol. abs/1511.08458, 2015. [Online]. Available: http://arxiv.org/abs/1511.08458 [Pages 11, 12, 13, and 14.]

[26] B. Ahlin and M. Gärdin, "Automated classification of steel samples: An investigation using convolutional neural networks," 2017. [Page 12.]

[27] A. Briot, P. Viswanath, and S. Yogamani, "Analysis of efficient cnn design techniques for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. [Page 14.]

[28] R. P. K. Poudel, S. Liwicki, and R. Cipolla, "Fast-scnn: Fast semantic segmentation network," *CoRR*, vol. abs/1902.04502, 2019. [Online]. Available: http://arxiv.org/abs/1902.04502 [Pages 14 and 15.]

[29] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2016. [Pages 14, 15, and 16.]

[30] L. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017. [Online]. Available: http://arxiv.org/abs/1706.05587 [Pages 14, 16, 17, and 18.]

[31] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. [Page 16.]

[32] S. Dev, Y. H. Lee, and S. Winkler, "Color-based segmentation of sky/cloud images from ground-based cameras," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 1, pp. 231–242, 2016. [Page 23.]

[33] S. Dev, F. M. Savoy, Y. H. Lee, and S. Winkler, "Nighttime sky/cloud image segmentation," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 345–349. [Page 23.]

[34] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE, 2018, pp. 117–122. [Page 24.]

[35] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC genomics*, vol. 21, pp. 1–13, 2020. [Pages 25 and 27.]

[36] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *International symposium on visual computing*. Springer, 2016, pp. 234–244. [Page 26.]

[37] T. K. Kim, "Understanding one-way anova using conceptual figures," *Korean journal of anesthesiology*, vol. 70, no. 1, pp. 22–26, 2017. [Page 27.]

[38] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, pp. 157–173, 2008. [Page 28.]

[39] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019. [Page 29.]

[40] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020. [Page 53.]

[41] B. G. Marcot and A. M. Hanea, "What is an optimal value of k in k-fold cross-validation in discrete bayesian network analysis?" *Computational Statistics*, vol. 36, no. 3, pp. 2009–2031, 2021. [Page 53.]

[42] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer, 2014, pp. 818–833. [Page 55.]

TRITA-EECS-EX-2024:875
Stockholm, Sweden 2024

www.kth.se