Cloud Segmentation Using Machine Learning

Submitted Aug 24 2023, in partial fulfillment of
the conditions for the award of the degree **MSc Scientific and Data Intensive Computing.**

Tianyi Zhang

22031979

Email: ucaptz0@ucl.ac.uk

Supervised by Dr. Eloise Marais

School of Physics and Astronomy of University College London

I, Tianyi Zhang confirm that the work presented in this dissertation is my own, where information has been derived from other sources. I confirm that this has been indicated in this dissertation. The GitHub code link for this project is `https://github.com/ZChenAn/Cloud-Segmentation-ResUNet`.

Abstract

Atmospheric pollution detection, a crucial component of environmental monitoring, often faces challenges such as the difficulty in detecting and retrieving information about the amount of cloud cover. This study addresses these challenges through the development of a machine learning-based approach for accurate cloud detection and its coverage estimation. A dataset obtained from web cameras mounted on an instrument located on the rooftop of the 11-story Torrington Place building on the UCL campus is utilized for the experiment.

A UNet[1] neural network model, powered by a ResNet[2] encoder, is designed and implemented as the core solution to distinguish and quantify cloud cover from sky images by using cloud segmentation theory. The model is trained and validated using the SWINYSEG[3] and SWIMSEG[4] datasets from the National University of Singapore.

The proposed model demonstrates encouraging results when tested on the dataset obtained from web cameras on the UCL campus, effectively determining the presence and extent of cloud cover under diverse conditions. These findings underline the potential effectiveness of the machine learning model in mitigating the impact of cloud cover on the accuracy of remote sensing instruments used for atmospheric pollution detection.

This research demonstrates the applicability of machine learning and computer vision techniques to cloud segmentation techniques for completing cloud detection and its coverage estimation. The study also contributes to the domain of atmospheric analysis and remote sensing, and the results provide a stepping stone for future research aimed at improving the reliability and precision of environmental sensing devices.

Acknowledgements

First, I would like to express my deepest and sincere gratitude to my supervisor Dr.Eloise Marais for his help and guidance throughout the project and his concern for my life in London in general. I have had a great year on the project and have gained a deeper understanding of machine learning and computer vision, and it has been a great pleasure to discuss these areas with my supervisor.

I would also like to express my sincere gratitude to all the faculty and staff of the School of Physics and Astronomy for helping me to learn the relevant background knowledge.

Finally, I am also very grateful to my parents for their support and help during the period of my final year project, and to all my friends for their companionship and concern.

Contents

Chapter 1

Introduction

This chapter presents the background of atmospheric pollution detection with the influence of cloud cover and cloud detection methods, the motivation, objectives, and contribution of the project. The dissertation outline will also be included.

1.1 Background

Atmospheric pollution detection is a vital process in monitoring and mitigating environmental hazards. This process often relies on remote sensing instruments, which use sunlight measurements to retrieve information about pollutants present in the atmosphere. However, this methodology is often influenced by natural elements such as cloud cover, which can drastically alter or obscure the sunlight that these instruments measure, making it difficult to detect and retrieve information about the amount of cloud cover.

Clouds, by nature, have a high albedo, meaning they reflect a significant proportion of incident sunlight into space. This reflection changes the amount of sunlight that reaches the ground, thus altering the measurements taken by remote sensing instruments. Furthermore, clouds can entirely block sunlight, leading to shadowed regions where the detection of atmospheric pollutants becomes extremely difficult. Consequently, the ability to accurately detect and quantify cloud cover is crucial to improving the reliability of data retrieved by these instruments. In remote sensing, consideration of cloud parameters is generally necessary to ensure accurate atmospheric pollution detection[5].

To accurately detect and quantify cloud layers, the concept of cloud detection was then introduced. Cloud detection is a pivotal facet of meteorological and climate studies, underpinning various scientific and practical applications. Over the years, numerous methods have been proposed and employed for cloud detection. The following are some of the more widely used techniques:

1. Visual Observation: This is the traditional approach, where trained observers identify and categorize clouds based on their appearance and position in the sky. While this method is simple and intuitive, it is highly subjective and lacks scalability.

2. Ground-Based Detectors: These instruments, such as the Total Sky Imager (TSI) and the Whole Sky Imager (WSI)[6], capture sky images and analyze them to detect cloud presence and coverage. However, their application is limited by their field of view, operational difficulties in poor weather conditions, and the requirement for regular maintenance.

3. Satellite Imagery: Remote sensing via weather satellites has proven instrumental in cloud detection on a large scale. Satellites, such as those in the GOES[7] (Geostationary Operational Environmental Satellite) and NOAA[8] (National Oceanic and Atmospheric Administration) series, offer extensive spatial coverage and spectral channels useful for cloud detection. Yet, such data may be affected by the satellite's orbital position, the atmosphere's optical properties, and temporal resolution.

4. Infrared and Radiometric Techniques: These methods are based on the measurement of radiative properties of clouds, particularly in the infrared spectrum, where clouds exhibit distinct radiative signatures. However, this technique is sensitive to the variation in cloud types and atmospheric conditions.

In light of these methods, the growing necessity for accurate, efficient, and high-resolution cloud detection has led to the advent of image segmentation techniques. In particular, cloud segmentation has shown promising results. This approach involves partitioning a digital image into multiple segments, or pixels, representing cloud and non-cloud regions. The effectiveness of cloud segmentation heavily depends on the image quality and the employed algorithm's sophistication.

The advent of machine learning has provided a substantial boost to cloud segmentation. With the ability to learn from large data volumes, machine learning algorithms offer accurate distinctions between cloud and non-cloud regions in sky images. Despite requiring substantial computational resources, these techniques are increasingly favored due to their superior performance and reliability.

1.2 Motivation

The motivation for undertaking this project arises from the recognized need for more accurate and reliable cloud detection in the context of atmospheric pollution measurement. Traditional cloud detection methodologies, while foundational, are constrained by a range of limitations, from the subjectivity of visual observations to the scale, cost, and interpretative challenges presented by satellite imagery and ground-based detectors. Emerging technological advancements offer promising avenues to address these challenges. Cloud segmentation and machine learning have the potential to substantially enhance the precision and efficiency of cloud detection, thereby mitigating the effects of cloud interference on atmospheric pollution detection. That means I must design a neural network to implement cloud segmentation.

1.3 Aims and Objectives

The primary aim of this project is to design and train an efficient and high-accuracy neural network for cloud segmentation. The code of this research project will also be released. The objectives to achieve this aim are as follows:

1. Development of a Neural Network Model: Utilize the principles of machine learning and computer vision to develop a neural network model tailored to the task of cloud segmentation. The model will be structured around the U-Net architecture, recognized for its effectiveness in

various image segmentation tasks.

2. Implementation of ResNet-50 as the Encoder: Incorporate ResNet-50 as the encoder within the U-Net model. ResNet-50, renowned for its depth and ability to learn complex patterns without significant performance degradation, is anticipated to enhance the model's learning capacity and, therefore, its segmentation accuracy.

3. Training and Validation: Train and validate the model using the SWINYSEG and SWIMSEG datasets, a robust dataset provided by the National University of Singapore. This will involve fine-tuning the model's parameters to minimize the loss function and maximize its predictive accuracy.

4. Testing with UCL Webcam Dataset: To evaluate the performance of the trained model, I conducted tests using the UCL Webcam Dataset. This proprietary dataset comprises 10,000 sky images captured by two webcams mounted on the rooftop of University College London. The UCL Webcam Dataset provides a diverse range of sky images, presenting a realistic and challenging testing environment for assessing the effectiveness of the model.

5. Cloud Coverage Estimation: Use the model's cloud segmentation results to calculate the cloud coverage of the sky in each image. This will provide an estimate of the proportion of the sky covered by clouds, further assisting in discerning whether the weather at the time of image capture was cloudy or clear.

By fulfilling these objectives, I hope to contribute a powerful tool for cloud detection to the atmospheric research community and demonstrate the utility of machine learning and computer vision in this important scientific domain.

1.4 Contributions

Over the last year, all research objectives have been achieved. The main contribution of this project was highlighted as follows:

1. Innovative Cloud Segmentation Model: I will provide an innovative neural network model for cloud segmentation, combining the power of ResNet-50 and U-Net. This model is expected to outperform traditional methods in terms of accuracy and efficiency, thereby advancing the state-of-art in cloud detection.

2. Validation with Real-world Dataset: The model will be validated against the UCL Webcam dataset, a collection of 10,000 real-world sky images captured in various weather conditions. This rigorous testing will ensure that the model is robust and capable of functioning effectively in real-world scenarios.

3. Cloud Coverage Estimation Tool: The project will contribute a tool for estimating cloud coverage based on the outputs of the segmentation model. This tool will provide an accurate measure of cloud cover, thereby facilitating the interpretation of atmospheric conditions.

4. Contribution to Open Science: The research makes a significant contribution to open science by sharing the codebase, sample results, and methodologies used on the UCL Webcam dataset.

Although the dataset itself remains unpublicized, providing access to the code and sample results empowers fellow researchers to replicate and build upon the work in cloud detection and atmospheric research. This open sharing of resources fosters collaboration and enables the broader scientific community to benefit from the findings and advance their investigations in related fields.

5. Enhanced Understanding of Machine Learning in Cloud Detection: By applying and thoroughly testing machine learning algorithms in cloud detection, the work will deepen the understanding of how these advanced technologies can be harnessed in this domain.

1.5 Dissertation Outline

The dissertation describes all the relevant details of the cloud segmentation project I worked on this year and consists of 7 chapters. **Chapters 1 and 2** are the introduction to the project and related work. **Chapter 3** mainly describes the model design and implementation, including related algorithms, which is one of the main contributions of this project. Another important contribution is presented in **Chapter 4**, which describes the usage of the UCL Webcam dataset. **Chapter 5** describes all relevant experiments. **Chapters 6 and 7** are the retrospective and summary of the project.

Chapter 2

Related Work

This chapter mainly describes the research work of cloud segmentation technology in recent years, including traditional image processing methods and machine learning computer vision methods.

2.1 Image processing for cloud segmentation

Traditional image processing algorithms play a crucial role in the field of cloud segmentation, usually expressed in the processing of image pixels, and thresholds...

2.1.1 Fixed Thresholding Method

This method, also known as binary thresholding, is one of the simplest forms of image segmentation. The core principle is to set a specific threshold value, T, such that pixel intensities above this threshold are classified as one class (e.g., cloud). In contrast, intensities below the threshold are classified as another (e.g., sky). This approach assumes that the histogram of the image pixel intensities will show a bimodal distribution, making it a straightforward and effective method for segmenting images with high contrast.

$$I(i,j) = \begin{cases} 1, & \text{if } I(i,j) \geq T \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

Where I(i, j) is the intensity of the pixel at location (i,j), and T is the chosen threshold.

However, its primary limitation lies in its sensitivity to changes in lighting conditions and its inability to adapt to varying contrasts across different regions of an image.

2.1.2 Visibility Condition-based Segmentation

This method is designed to address the common occurrence of poor visibility conditions in satellite imagery due to clouds. It involves applying the Fast Fourier Transform[9] to each color channel of an image, thus shifting the spatial domain representation to the frequency domain.

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot e^{-2\pi i\left(\frac{ux}{M}+\frac{vy}{N}\right)} \tag{2.2}$$

Where F(u,v) is the image in the frequency domain, f(x,y) is the image in the spatial domain, and (M, N) is the size of the image.

By calculating the magnitude spectrum and analyzing the power distribution, this method can distinguish areas of low visibility (clouds) from those with clear visibility[10]. It improves the cloud detection process by employing a more dynamic approach to considering spatial and temporal changes. However, the computational complexity of this approach is relatively high, limiting its applicability in real-time cloud detection scenarios.

2.1.3 Mean and Hybrid Detection Algorithms

These methods involve converting the original RGB image into grayscale and then calculating the mean intensity value to use as the segmentation threshold[11]. By utilizing the mean value, these methods take into account the overall image lighting conditions, making them robust against variations in illumination. The primary difference between the two lies in how they handle the thresholding: the Mean Algorithm uses a global mean, while the Hybrid Algorithm adjusts the threshold locally according to the image characteristics, leading to more accurate segmentation.

$$T = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I(i,j)}{MN} \tag{2.3}$$

Where I(i,j) is the intensity of the pixel at location (i,j) and MN is the total number of pixels in the image.

While this approach effectively maintains the integrity of the image, it demonstrates limitations in detecting certain cloud types, such as thin clouds. The development of a more comprehensive algorithm capable of identifying all types of clouds remains a formidable challenge within the research community.

2.1.4 Hybrid Threshold Algorithm

This approach (HYTA) combines elements of both fixed and adaptive thresholding methods, providing a more flexible solution. The algorithm first calculates the Blue to Red ratio, determining the type of the image histogram (unimodal or bimodal) based on this ratio[12].

$$B/R = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} B(i,j)}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} R(i,j)} \tag{2.4}$$

Where B(i,j) and R(i,j) are the intensities of the blue and red channels respectively at location (i,j).

Depending on the histogram type, the algorithm then selects the most appropriate thresholding method. This adaptability enables the HYTA to effectively handle a wider range of cloud scenarios compared to the traditional fixed thresholding methods. However, this method was not well suited for real-world applications.

2.1.5 Fuzzy Clustering Algorithm

This method leverages the concept of fuzzy logic to address the inherent uncertainty and variability in cloud structures. Unlike other methods that classify pixels as either cloud or not cloud, the Fuzzy Clustering Algorithm[13] assigns each pixel a membership value between 0 and 1 based on its similarity to the cloud and sky classes. To calculate the membership of a pixel to a certain class:

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{d_{ik}}{d_{ij}}\right)^{\frac{2}{m-1}}} \tag{2.5}$$

Where u_{ik} is the membership of pixel i to class k, d_{ik} is the distance from pixel i to the center of class k, d_{ij} is the distance from pixel i to the center of class j, m is a fuzziness parameter, and c is the number of classes.

The algorithm iteratively adjusts these membership values until a satisfactory classification is achieved, making it highly effective for complex cloud structures but computationally demanding.

2.1.6 Watershed and ROI-Based Segmentation

The Watershed[14] algorithm is based on visualizing an image in three dimensions: x, y (spatial coordinates), and intensity values. The image can be seen as a topographic relief considering intensity values as height. The watershed lines represent the boundaries of these relief zones and, in image processing, delineate the regions of the image (the catchment basins or ROIs). The core concept is visualized mathematically as follows:

Given image I, a point x is a regional minimum if all points in the neighborhood of x have greater intensity. The set of all points y in the image can be connected to x by a path in the image such that the intensity of any point on the path is not less than the intensity of y in a catchment basin or region of interest[15] (ROI).

Despite its wide usage, the Watershed and ROI-Based Segmentation algorithm has a notable downside: it is prone to over-segmentation, especially for images with complex textures or noise, which could lead to inaccurate cloud segmentation results.

2.1.7 Superpixel Segmentation Algorithm

The Superpixel Segmentation Algorithm[16] operates by grouping pixels into perceptually meaningful atomic regions, which can be used to replace the rigid structure of the pixel grid. One common method is the Simple Linear Iterative Clustering (SLIC) algorithm[17].

$$E = \sum_{k=1}^{K} \sum_{x_i \in S_k} [||x_i - m_k||^2 + ||y_i - l_k||^2] \tag{2.6}$$

Where K is the total number of superpixels, x_i and y_i denote the color and spatial coordinates respectively, m_k and l_k are the corresponding cluster centers in the 5D space (color + spatial). The algorithm iteratively minimizes this energy function to achieve coherent superpixels.

Despite its advantages, the Superpixel Segmentation algorithm tends to fail when dealing with small or thin structures, such as thin clouds, because they could be ignored or incorporated into larger superpixels. Moreover, the selection of the number of superpixels is crucial and could affect the segmentation results.

2.2 Machine learning for cloud segmentation

With the advent of machine learning, many neural networks have been used for cloud segmentation, such as fully connected networks, CNNs, and U-Net. In the following, three designed neural networks that have been used with good results in cloud segmentation will be introduced.

2.2.1 Enhancement fully convolutional network

The Enhancement Fully Convolutional Network[18] (EFCN) is designed for the segmentation of cloud pixels from both daytime and nighttime All-Sky Imager (ASI) images, offering an improvement over existing algorithms due to its ability to handle images under varied lighting conditions. The main innovation lies in the EFCN's ability to convert RGB images into Hue Saturation Intensity (HSI) space, enhance the 'Intensity' channel via histogram equalization, and convert them back into RGB space before the segmentation process. This step significantly improves the visual clarity of the images, especially those captured at night under complex weather conditions. The architecture of the EFCN model includes several layers such as RGB-HSI layer, histogram equalization layer, HSI-RGB layer, convolutional layers, pooling layers, deconvolutional layers, skip architecture and activation functions. It uses a deconvolution operation to up-sample the heat maps, allowing the generated predictions to retain the spatial information from the original input image. To address the issue of detail loss in up-sampled projections, the authors propose a skip connection structure that better restores the features in the input image.

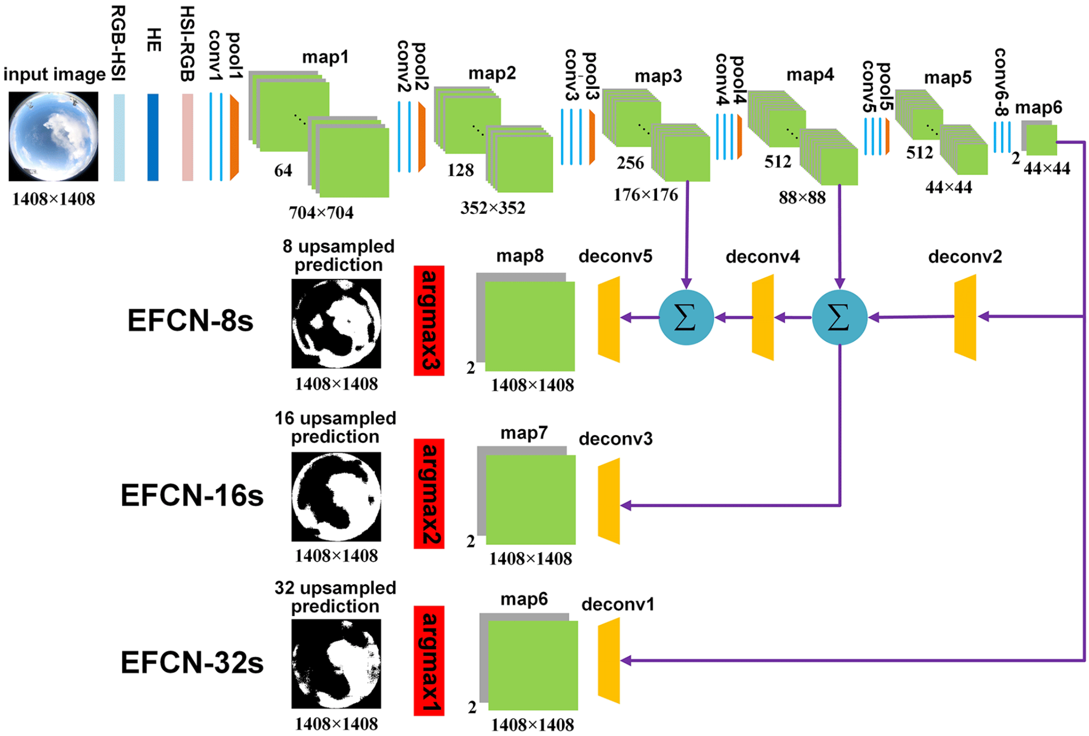


Figure 2.1: Detailed architecture of EFCN model including RGB–HSI layer, histogram equalization layer, HSI–RGB layer, convolutional layers, pooling layers, deconvolutional layers, skip connection, and activation functions.[18]

One key advantage of the EFCN is its capability to accept and process images of any size and produce predictions for each pixel, effectively performing pixel-level classification. It improves upon previous methods' inability to distinguish cloud pixels in nighttime ASI images, where pixel values are low and thus harder to differentiate. On the other hand, there are some drawbacks associated with the EFCN. While the use of histogram equalization improves image clarity, it introduces an extra image processing step, potentially increasing computational time. Furthermore, although the skip connection structure provides more accurate segmentation, it increases the complexity of the network and requires careful tuning of hyperparameters.

2.2.2 CloudSegNet network

CloudSegNet[3], a deep-learning architecture, is proposed for efficient cloud segmentation. The model utilizes an encoder-decoder structure, built around convolution, deconvolution, and pooling layers. It offers pixel-based semantic segmentation of sky cloud images, labeling each point as sky or cloud. The model can operate both during the day and night, making it highly versatile for nychthemeron (24-hr period including day and night) cloud image segmentation.

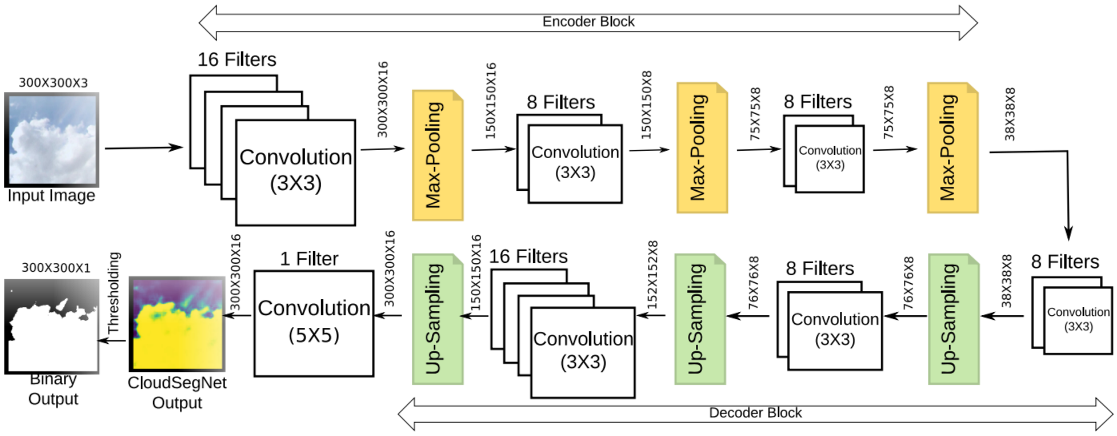


Figure 2.2: Encoder–decoder architecture of the CloudSegNet model. The tensor dimensions at the output of each block are specified. The CloudSegNet output is a probability mask, which is binarized via thresholding in the final step.[3]

The novelty of the method lies in its light architecture and universal applicability to both day and night sky cloud images. Unlike many other deep neural network architectures, it doesn't use any fully connected layers or a skip architecture. Despite this, it performs well in segregating discriminatory image features from different levels and generating a semantic segmentation mask at the same resolution as the input image. However, input images must be resized to a fixed size (300 × 300 pixels), which might be a limitation for diverse image datasets.

2.2.3 CCAD-Net network

CCAD-Net[19] is a cascade cloud attribute discrimination network that aims at improving cloud genera segmentation. The network is designed with an improved encoding-decoding model and adds a binary segmentation branch for cloud detection and an attribute discrimination branch for cloud attribute feature learning in the decoding stage. The network capitalizes on visual attributes defined according to professional knowledge to design the attribute discrimination constraint, thus improving the segmentation performance. The structure of the network includes multiple tasks operating in a cascade, with each additional branch extracting task-specific features successively. The network's design aims to fuse raw features, binary segmentation features, attribute discrimination features, and cloud genera features to improve the performance of cloud genera segmentation.

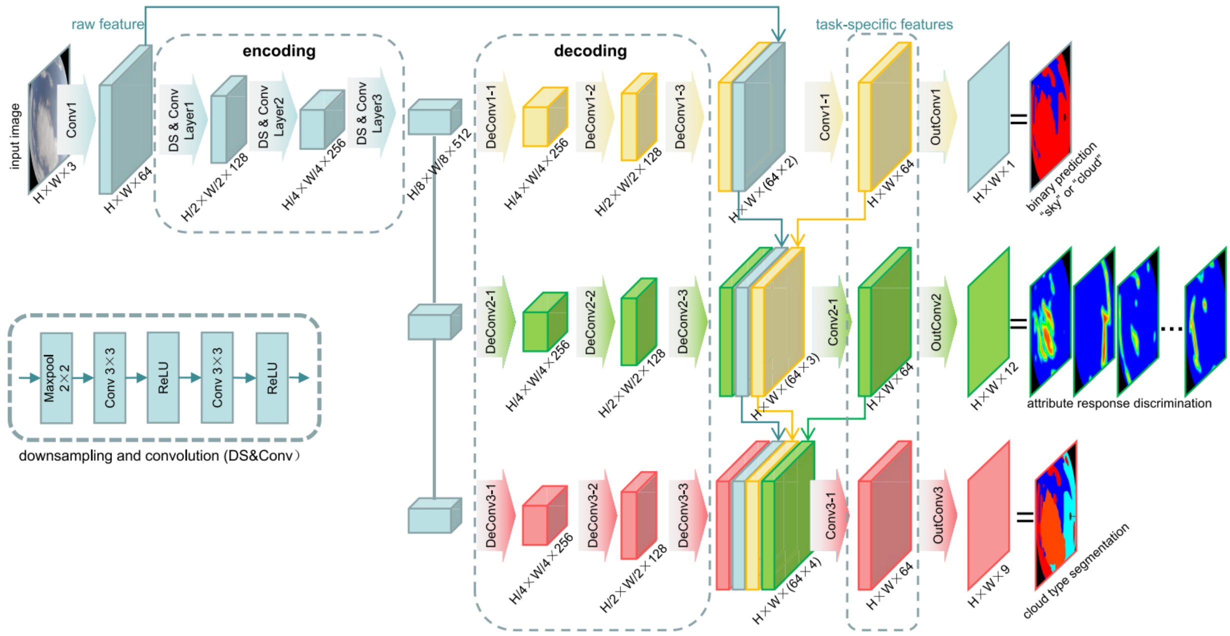


Figure 2.3: Architecture of the CCAD-Net model.[19]

The major innovation of CCAD-Net lies in defining attribute discrimination as a subtask for cloud genera segmentation and integrating it into the network. The model outperforms many state-of-the-art methods in cloud genera segmentation. However, the network might become complex due to the additional branches, potentially increasing the computational cost.

Chapter 3

Model Design and Implementation

This chapter focuses on the design and implementation of the proposed model, a ResNet50-based U-Net neural network, for sky/cloud image segmentation. It comprises two main sections: methodology and code implementation, and the specific details for network structure design, the training validation process, and the rectification strategy.

3.1 Methodology

The project method begins with the preprocessing of the input images, where they are cut into patches of the appropriate size for the network. These patches can be used for testing later. During training and validating, the model is trained and validated on a separate set of images to monitor its performance and adjust the learning rate accordingly. Once the model is trained, it is fine-tuned using a rectification strategy that involves the use of manually labeled patches from the testing dataset. After rectification, the model is ready for testing. The testing phase involves feeding the model with patches from the testing images and combining the segmented patches to form the final segmented image. The details of each step will be described in each of the following subsections.

3.1.1 Network Structure Design

The proposed model for sky/cloud image segmentation is a U-Net neural network, which utilizes ResNet50 as the backbone for the encoder part. The choice of U-Net stems from its proven success in various image segmentation tasks, owing to its ability to capture both local features and global context through its encoder-decoder structure. The inclusion of ResNet50 in the encoder part enhances this feature extraction capability by allowing the network to extract complex features through its deep architecture.

The unique aspect of this model is the integration of the ResNet50 architecture into the U-Net, which is not commonly seen in traditional U-Net models. This combination allows the model to leverage the advantages of both architectures, i.e., the context-capturing ability of U-Net and the complex feature extraction ability of ResNet50.

Moreover, the decoder part of the U-Net uses transposed convolutions for up-sampling, which is a common approach in such architectures. However, to retain high-resolution features lost during the encoding process, each block in the decoder part receives the output from the corresponding encoder block through a skip connection. This strategy helps to retain the high-resolution features, which are crucial for accurate segmentation.

The final layer of the U-Net is a 1x1 convolution. This layer maps each 64-component feature vector to the desired number of classes, making it suitable for multi-class segmentation tasks.

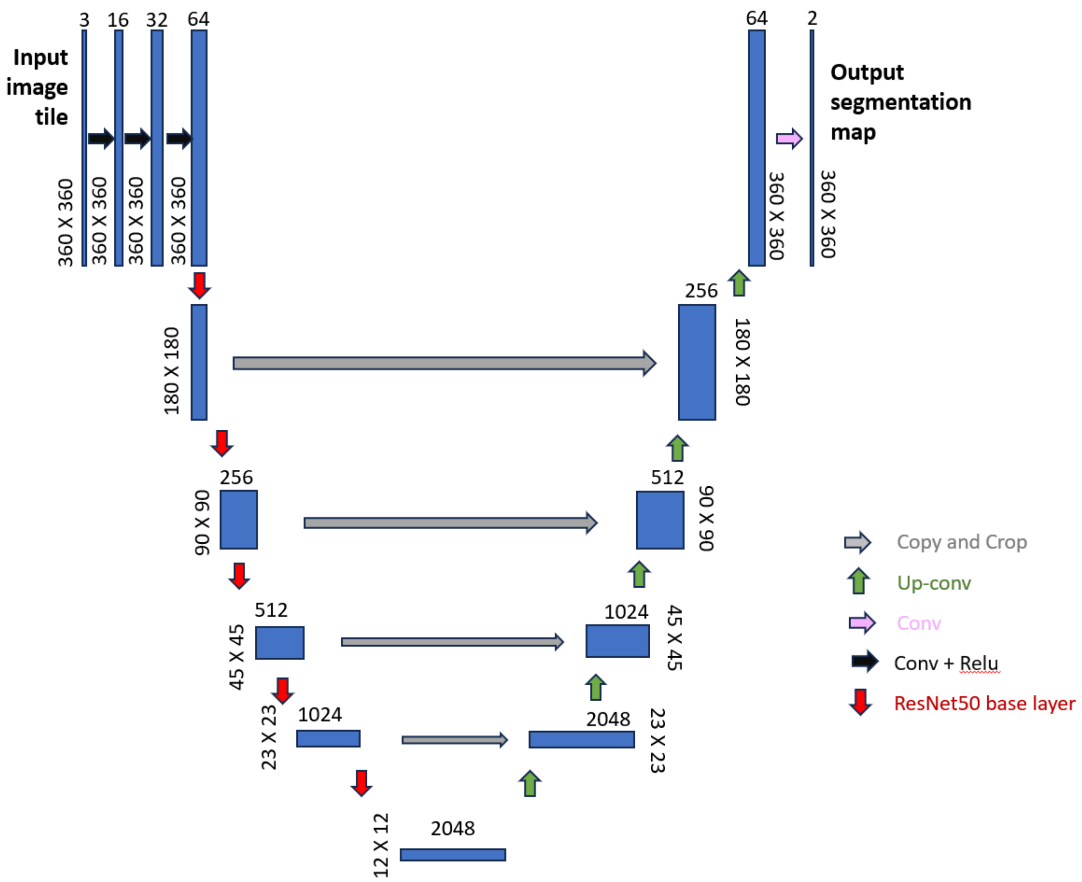


Figure 3.1: Architecture of the ResUNet model which is a combination of ResNet50 as the encoder and a partial UNet as the decoder. The input is a 3-color channel image of size 360x360 pixels. Starting with original-size convolutions, gradually increasing channels to 64, the ResNet50 encoder performs feature extraction generating maps at different stages. Subsequently, upsampling with skip connections fuses encoder outputs with feature maps in the decoder for high-resolution feature restoration. Final convolutional layers produce desired segmentation predictions. The intricate details captured by the ResUNet model make it promising for accurate image segmentation tasks, outputting a 360x360x2 tensor representing binary classification of cloud and non-cloud regions.

3.1.2 Training and Validation Process

The model was trained on the SWINYSEG dataset[3], which consists of 6768 images of 300x300 pixels. The model was validated on the SWIMSEG dataset, which contains 1013 images of 600x600 pixels. The training input to the network was patches of 300x300 pixels, the entire images in the SWINYSEG dataset.

In terms of loss function, the model uses a combination of Binary Cross-Entropy (BCE) loss and

Dice loss. The BCE loss is a commonly used loss function for binary classification problems, and it measures the error between the model's predictions and the ground truth. On the other hand, the Dice loss measures the overlap between the predicted segmentation and the ground truth segmentation, making it a suitable choice for segmentation tasks. The combination of BCE loss and Dice loss allows the model to balance between pixel-wise classification accuracy (BCE loss) and segmentation quality (Dice loss). This combination makes the model robust to different types of segmentation errors and improves the overall segmentation performance.

For optimization, the Adam optimizer was used with a learning rate of 1e-4. The learning rate was reduced by a factor of 0.1 if the validation loss did not decrease for 10 epochs. The training was stopped if the validation loss did not improve for 25 epochs. This strategy of adjusting the learning rate and early stopping helps prevent overfitting and ensures the model generalizes well to unseen data. To get results quickly, the default epoch is set to 5 only.

During the whole training process, various data augmentation techniques were applied to the training images, such as random horizontal and vertical flips, random rotation, and brightness and contrast adjustments. These augmentations help the model to learn more generalized features and prevent overfitting.

3.1.3 Rectification Strategy

In the testing phase, the model was applied to the UCL Webcam dataset, which consists of sky/cloud images captured at UCL without ground truth masks. To adapt the model trained on the SWINYSEG dataset to the UCL Webcam dataset, a rectification strategy was employed. Eleven patches were manually labeled from the UCL Webcam dataset and used to fine-tune the model. To overcome the limitation of a small sample size, a semi-supervised learning approach was applied. The unlabeled patches from the UCL Webcam dataset were enhanced using weak and strong augmentations. The weakly augmented patches were fed into the trained network to generate pseudo-labels, which were then binarized and used as the ground truth to calculate the loss with the strongly augmented patches. The final loss was a combination of the loss from the manually labeled data and the loss from the pseudo-labeled data. This semi-supervised learning strategy effectively exploited the unlabeled data and helped improve the model's performance on the UCL Webcam dataset.

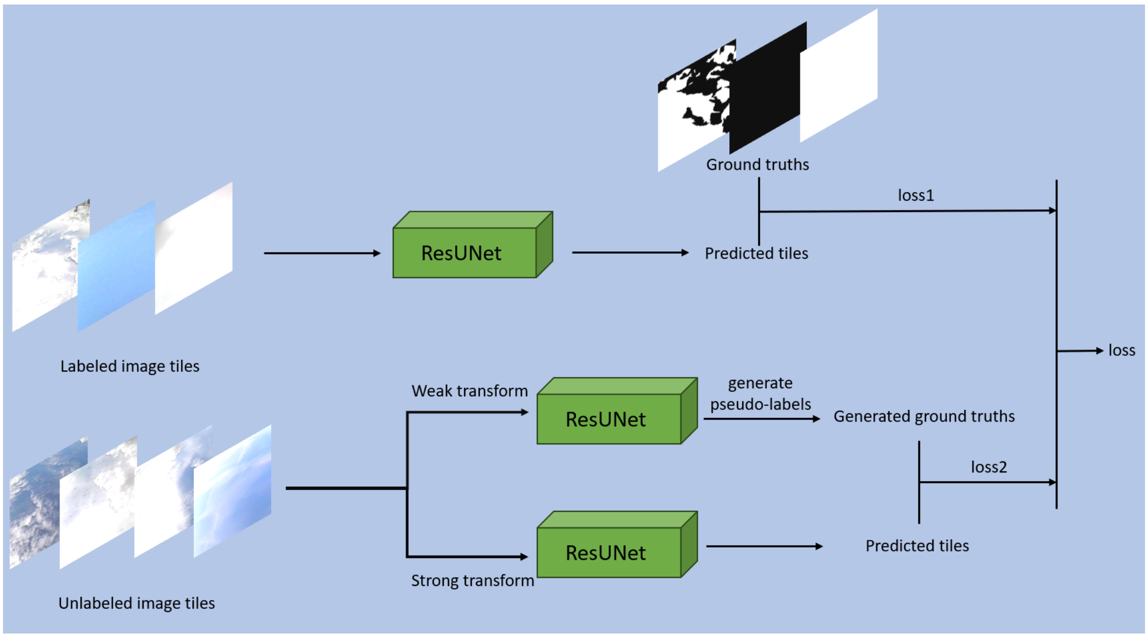


Figure 3.2: The progress of rectification training using semi-supervised learning. The entire training process is bifurcated into two components: labeled tiles and unlabeled tiles. The training of labeled tiles follows the conventional approach. For the training of unlabeled tiles, two distinct strategies are employed: weak transform and strong transform. Weak transform encompasses image transform such as rotation and symmetry, while strong transform comprises a series of operations that include weak transform alongside contrast enhancement. The weakly transformed image tiles are subsequently fed into the neural network, enabling the generation of pseudo-ground truth based on the approximate trends predicted by the neural network. This pseudo-ground truth, also called a semi-supervised signal, facilitates regular training, assisting in achieving improved performance.

3.2 Implementation

3.2.1 Environment, Languages, and Libraries

The main code of the project is run on my computer on a 3060 Ti, dual-core GPU computer, Nvidia Cuda version 11.8, and cuDNN version 8.9. The only programming language I use is Python. Python, a widely used, straightforward, advanced, and general-purpose programming language, supports a variety of programming paradigms, including functional, imperative, reflective, structured, and object-oriented programming. It has a dynamic typing system and garbage collection features, automatic memory usage management, and an extensive standard library of its own. The main libraries I use are the Pytorch[20] machine learning library, an open-source Python machine learning library based on Torch with an underlying C++ implementation for artificial intelligence applications such as natural language processing. The PyTorch library provides functionalities for building and training neural networks, including various types of layers, loss functions, optimization algorithms, and data-loading utilities. Its become the most widely used library for academic computer research. In addition, numpy, opencv python, scikit-image, matplotlib, and Pillow will also be used.

3.2.2 Code implementation

The code implementation was organized into separate Python scripts for modularity and easier debugging. The main script (main.py) coordinates the flow of the project, calling functions from the other scripts when needed. The logic of the code follows what was mentioned in the previous methodology, the cut_images part code is responsible for cutting the original images into patches of a specific size. These patches are used as the input for the network during the testing phases. The train part code is where the main training loop of the model is implemented. During each epoch, the function goes through the entire training dataset, feeding the model with input patches and updating the model's weights based on the calculated loss. The loss is a combination of Binary Cross-Entropy (BCE) loss and Dice loss, balancing between pixel-wise classification accuracy and segmentation quality. The validate part code is similar to the train function, but it does not update the model's weights. Instead, it computes the loss and other metrics on the validation dataset to monitor the model's performance and adjust the learning rate accordingly. The rectified part code is called after the training and validating phase. It fine-tunes the model using manually labeled patches from the UCL Webcam dataset. The test part code feeds the model with patches from the testing images and combines the segmented patches to form the final segmented image.

For the other scripts, the ResUNet.py script contains the implementation of the ResNet50-based U-Net model. It defines a Python class for the model, with different methods for constructing the model's architecture, forwarding an input through the model, and initializing the model's weights. The rectify.py script is responsible for the rectification strategy. It defines a Python class Rectify that creates, trains, and validates a model on a small manually labeled dataset from the UCL Webcam dataset. The handle_images.py contains several utility functions for handling images, such as loading and saving images, converting color spaces, equalizing histograms, and applying transformations. These functions are used throughout the other scripts for preprocessing and post-processing the images. The test.py script handles the UCL Webcam dataset for testing. The way these scripts are organized and linked together makes the project modular and easier to debug and maintain. Each script can be tested and modified independently, and changes in one script do not affect the others. This structure also makes it easier to understand the flow of the project, as each script is responsible for a specific part of the process.

In addition to the aforementioned Python scripts, a Jupyter notebook, named demo.ipynb, is also provided for a more streamlined execution of the project. This notebook offers an accessible interface for users to conveniently test the sky images and quickly visualize the results. Within demo.ipynb, results are demonstrated for four sample images that have been pre-included. The model utilized in the demonstration is pre-trained and rectified, saved as ty_rectify.pth. This model has undergone training for 5 epochs, providing a preliminary indication of the model's performance. Users can use this pre-trained model for their experiments or as a temporary point for further training. By providing this Jupyter notebook, the usability, and accessibility of the project are significantly enhanced, enabling users to quickly familiarize themselves with the workflow and engage in experimentation.

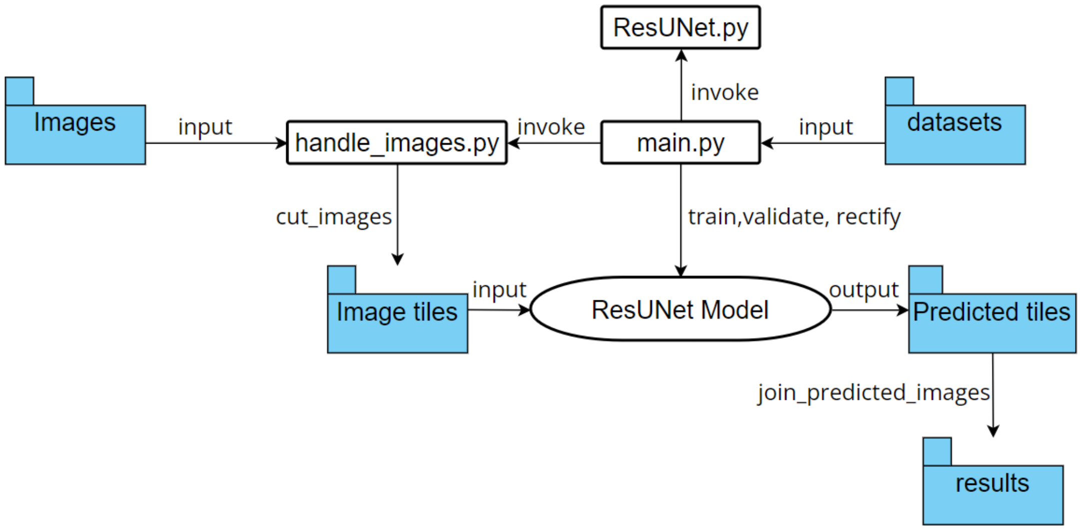


Figure 3.3: Code implementation procedure. The blue folder represents the folder of images used in the project, the rectangular box represents the python code file, and the oval box represents the model. Please refer to the previous text description for the detailed code implementation flow.

Chapter 4

Usage of UCL Webcam Dataset

This chapter describes the shortcomings of the existing sky cloud segmentation datasets, the introduction to the UCL Webcam dataset, and its evaluation protocol.

4.1 Deficiencies of existing datasets

The existing sky/cloud segmentation datasets have different degrees of deficiencies. I mainly discuss three existing well-known sky/cloud segmentation datasets which are HYTA dataset[21], SWIMSEG dataset[4], and SWINYSEG dataset[3].

4.1.1 HYTA dataset

The HYbrid Thresholding Algorithm (HYTA) dataset, originating from the University of Alberta, Canada, is a collection of ground-based sky/cloud images taken with a Whole Sky Imager. With images of diverse cloud types and sky conditions, it serves as a valuable resource for training and testing cloud detection algorithms. However, the HYTA dataset exhibits a couple of significant limitations.

Firstly, the HYTA dataset is relatively small, containing only 32 images. The small number of images may not sufficiently represent the variability of sky conditions, limiting the generalization ability of models trained on this dataset.

Secondly, the images in the HYTA dataset have varying sizes, ranging from 328 x 308 pixels to 745 x 512 pixels. The inconsistency in image size can pose challenges in model training and evaluation, as it necessitates additional preprocessing steps such as resizing or cropping, which may inadvertently alter the image content.

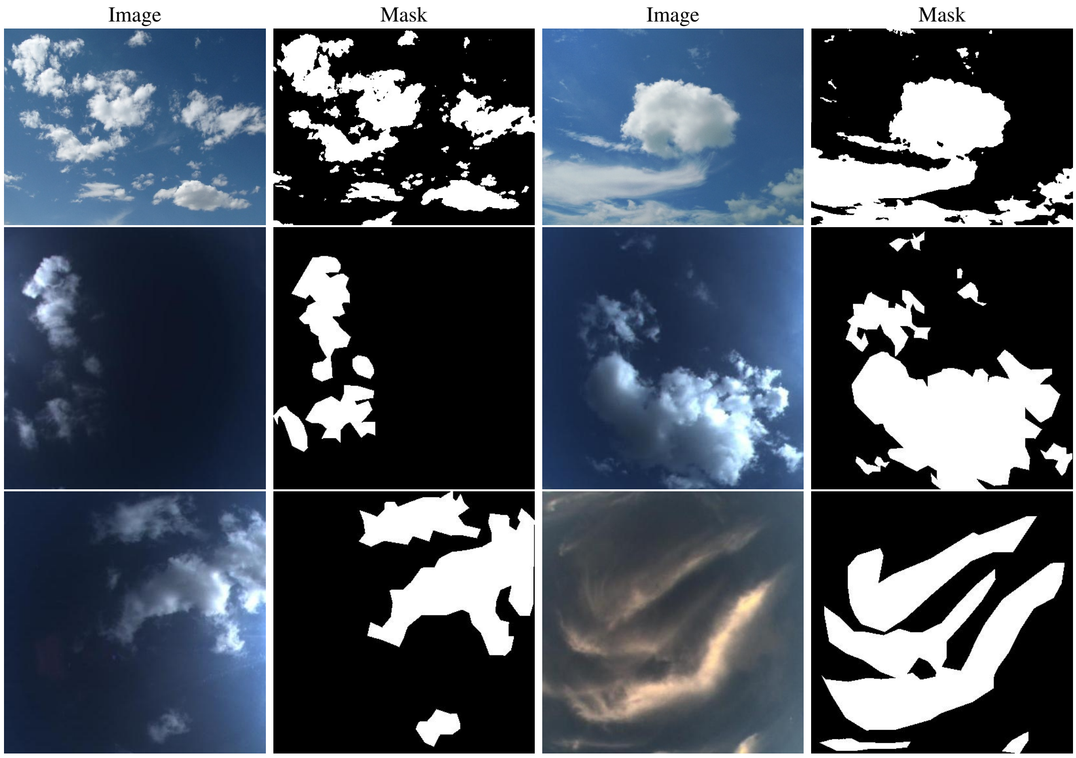


Figure 4.1: Samples of HYTA dataset. Resolution: 495x371, 400x400

4.1.2 SWIMSEG dataset

The Singapore Whole Sky IMaging SEGmentation (SWIMSEG) dataset comprises 1013 sky/-cloud patches of approximately 60-70 degrees of the sky, with a resolution of 600 x 600 pixels. Captured using a calibrated ground-based whole sky imager in Singapore, the dataset spans 22 months from October 2013 to July 2015.

While the SWIMSEG dataset provides a larger number of images compared to the HYTA dataset, it also has its deficiencies. The primary drawback is that it only contains patches of the sky rather than the whole sky images, which may limit its application in tasks that require a comprehensive view of the sky.

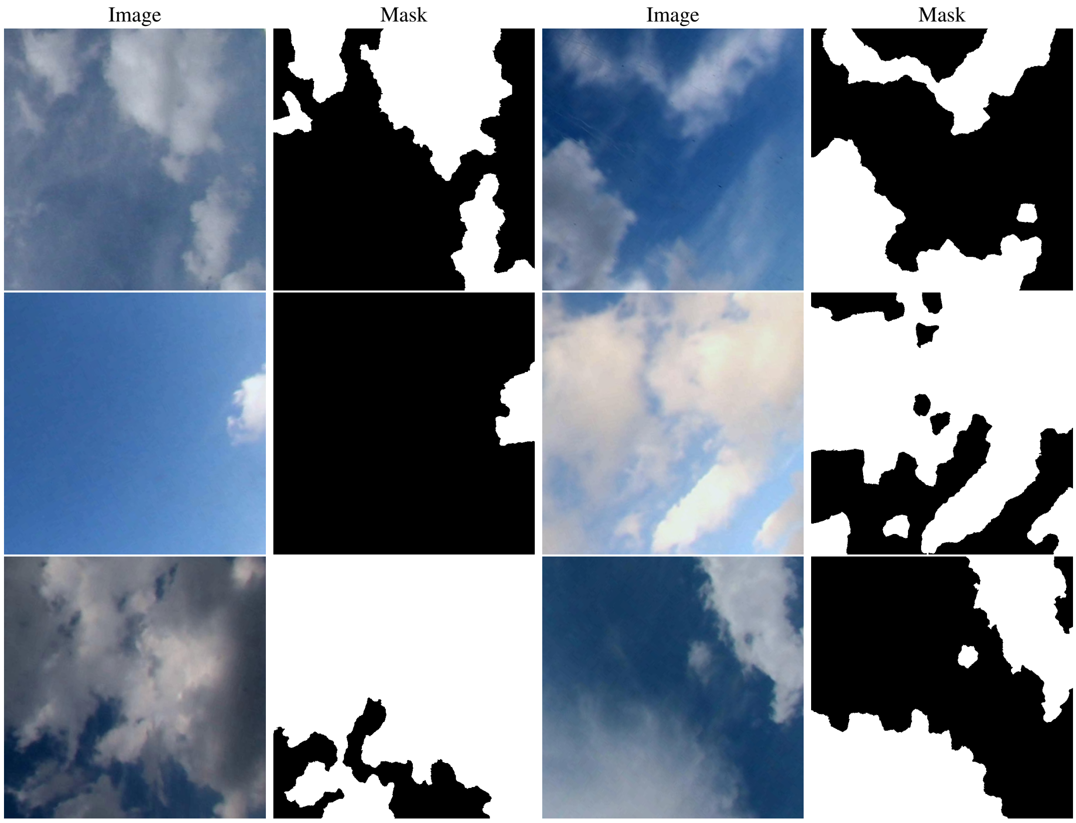


Figure 4.2: Samples of SWIMSEG dataset. Resolution: 600 x 600

4.1.3 SWINYSEG dataset

The Singapore Whole Sky Nychthemeron Image Segmentation (SWINYSEG) dataset consists of 6768 daytime and nighttime sky/cloud patches, captured over 12 months in 2016. Similar to the SWIMSEG dataset, the images in SWINYSEG are patches of the sky instead of the whole sky.

Although the SWINYSEG dataset includes both daytime and nighttime images, which enhance the diversity of lighting conditions, it shares the same limitation as the SWIMSEG dataset. The patch-based nature of the images may not fully represent the complexity and variability of whole sky images, potentially constraining the effectiveness of models trained on this dataset in real-world scenarios.

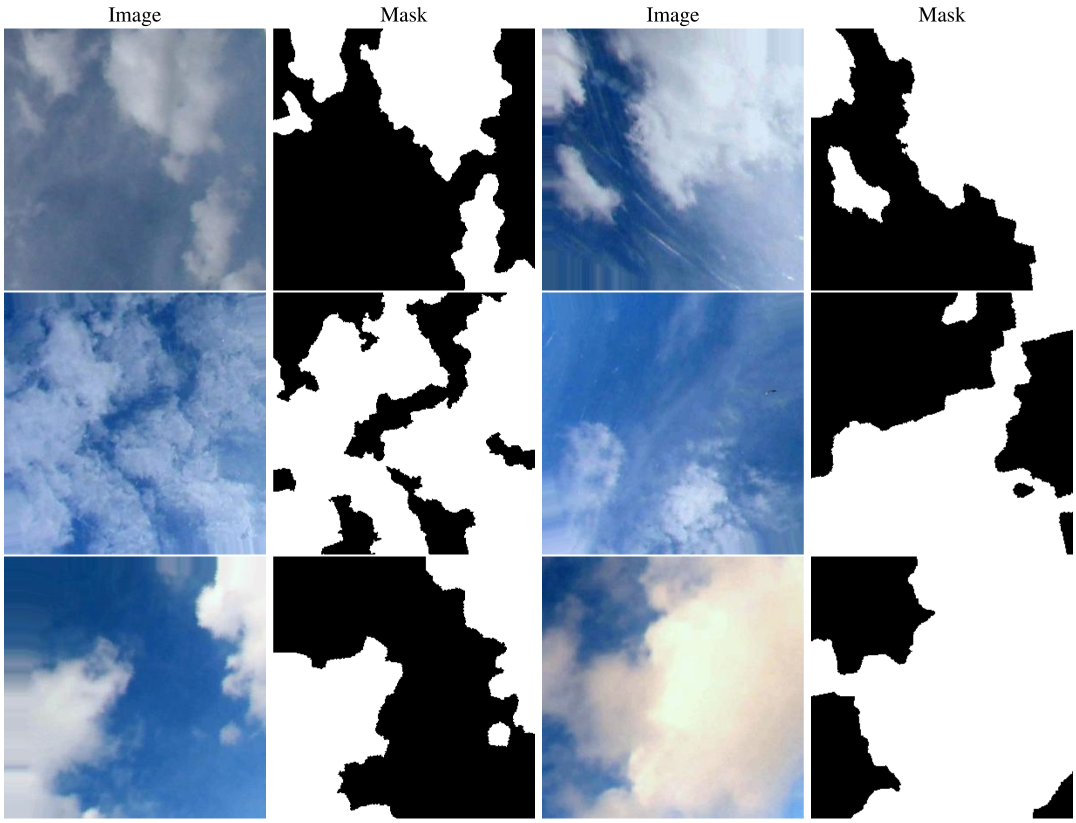


Figure 4.3: Samples of SWINYSEG dataset. Resolution: 300 x 300

4.2 Description of UCL Webcam dataset

My research introduces the UCL Webcam dataset, comprising 10,000 whole sky images with dimensions of 720 x 576 pixels. These images are obtained from web cameras mounted on an instrument located on the rooftop of the 11-story Torrington Place building on the UCL campus is utilized for the experiment. It captures the entire sky overhead every 10 minutes.

Figure 4.4: The instrument is located on the rooftop of the 11-story Torrington Place and captures the entire sky overhead every 10 minutes.

The UCL Webcam dataset offers several distinct advantages over existing datasets. Firstly, it provides comprehensive whole-sky images, offering a holistic view of the sky rather than partial sky patches. This broader perspective better represents the complexity and variability of sky conditions, potentially enhancing the realism and practicality of models trained on this dataset.

Moreover, the UCL Webcam dataset significantly surpasses existing datasets in terms of size, containing nearly ten times more images than the SWINYSEG dataset and over three hundred times more images than the HYTA dataset. This extensive collection of images better captures the variability of sky conditions, resulting in improved generalization capabilities for models trained on this dataset.

However, I acknowledge that the current version of the UCL Webcam dataset lacks ground truth labels, which poses a challenge for traditional supervised learning approaches. To address this limitation, I am committed to exploring innovative techniques, such as semi-supervised learning and manual labeling of a selected subset of images, to train my models effectively on this dataset.

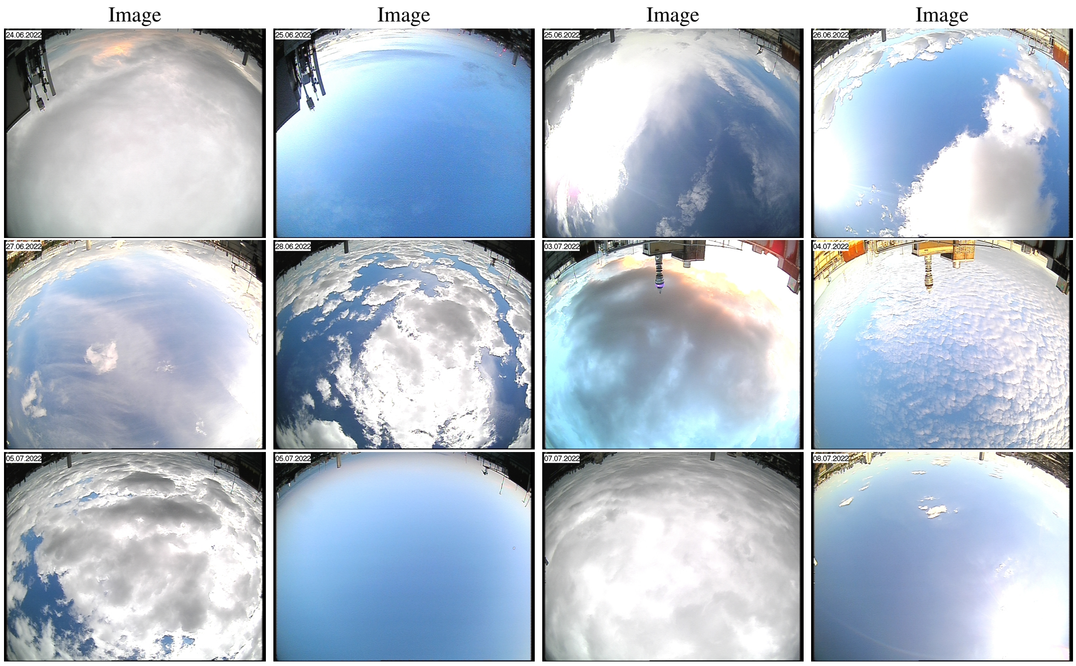


Figure 4.5: Samples of UCL Webcam dataset. Resolution: 720 x 576

4.3 Dataset Evaluation Protocol

For model training, I utilize the SWINYSEG dataset and SWIMSEG dataset, which offer comprehensive sky/cloud patches and expert-annotated ground truth labels, respectively. These datasets serve as reliable resources for training and validation purposes.

The novel UCL Webcam dataset is utilized for testing, even without ground truth labels. To mitigate this challenge, I adopt a semi-supervised learning approach that leverages both labeled data from the training set and unlabeled data from the test set. Additionally, I manually label a small subset of the UCL Webcam dataset, employing a rectification strategy to fine-tune my model to adapt specifically to this dataset.

In conclusion, the novel UCL Webcam dataset, despite its current limitations, represents a significant contribution to the field of sky/cloud image segmentation. I anticipate that it will facilitate the development of more realistic and applicable models, and I am committed to enhancing its utility in future endeavors.

Chapter 5

Experimental Results

This chapter describes the experiments on the SWIMSEG dataset (validation) and the UCL Webcam dataset (test). All the experiment analyses will also be included.

5.1 Experimental datasets

The datasets used for the experiments include the previously introduced SWIMSEG dataset and the UCL Webcam dataset. These two datasets have been described in detail in previous sections and can be found in the previous section. All images from both datasets will be used in the experiment. Because the UCL Webcam dataset has no real groud-truth, my experimental results on that dataset will only put out the prediction images of my network, which need to be used by your eyes to judge the goodness of my results.

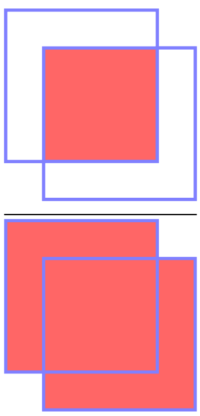
5.2 Test indicators of experiment

The test indicators I use in the experiment are IOU and F-score which are commonly used indicators for cloud segmentation.

IoU: Intersection over Union

Intersection over Union[22] (IoU) is a standard that measures the accuracy of detecting corresponding objects in a specific dataset. IoU is a simple measurement standard. As long as it is a task that obtains a bounding box in the output, IoU can be used for measurement.

$$IoU = \frac{\text{area of overlap}}{\text{area of union}} =$$



two squares represent predict box and ground truth

(5.1)

F-score: F1-score or F-measure

The F-score[23], also known as the F1-score or F-measure, is a commonly used metric for binary classification problems, including binary image segmentation tasks. It is defined as the harmonic mean of precision and recall:

$$F_{score} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5.2}$$

where:

- Precision, also known as the positive predictive value, is the proportion of true positive predictions (i.e., correctly identified cloud pixels) out of all positive predictions (i.e., all pixels predicted to be cloud).

$$precision = \frac{TP}{TP + FP} \tag{5.3}$$

- Recall, also known as sensitivity or true positive rate, is the proportion of true positive predictions out of all actual positive instances (i.e., all actual cloud pixels).

$$recall = \frac{TP}{TP + FN} \tag{5.4}$$

Here, TP stands for True Positives, FP for False Positives, and FN for False Negatives.

5.3 Experimental results on the SWIMSEG dataset

1. Experimental purpose

The purpose of this experiment was two-fold. First, it served as a validation of my trained model's performance. By testing on the well-known SWIMSEG dataset, I could assess the model's ability to generalize to new data. Second, it allowed us to compare my model's performance with that of other state-of-the-art methods that have also been evaluated on the SWIMSEG dataset. This would give us an indication of how well my model stands in the current research landscape. 2. Test indicators: IoU, F-score

3. Test images: 1013 sky/cloud segmentation images of 600 x 600 pix

4. Evaluation protocol

The model's performance on the SWIMSEG dataset was evaluated using the IoU and F-score I mentioned before.

IoU compares the overlap between the model's predicted segmentation mask and the actual ground truth mask. For each image in the SWIMSEG dataset, the model outputted a predicted binary segmentation mask. The IoU was then calculated against its corresponding ground truth mask for each predicted mask. The IoU scores were averaged over all images in the dataset to give an overall performance metric. This process was repeated for each epoch of training. As such, I could track the model's learning progress and see how its performance improved with each epoch.

In the context of cloud segmentation, the F-score is calculated for each image in the dataset by comparing the predicted binary segmentation mask against the ground truth mask. Specifically, True Positives (TP) are the cloud pixels correctly identified as cloud, False Positives (FP) are the non-cloud pixels incorrectly identified as cloud, and False Negatives (FN) are the cloud pixels incorrectly identified as non-cloud. it provides a balanced measure of the model's performance in terms of both precision and recall. This is particularly important in tasks like cloud segmentation, where both false positives (misidentifying non-cloud as the cloud) and false negatives (missing actual cloud pixels) are undesirable. By comparing the F-scores achieved by the model against those reported for other cloud segmentation methods in the literature, I can objectively evaluate the effectiveness of my method.

5. Result in analysis and case study

Table 5.1: The validation IoU corresponding to each epoch.

Epoch	IoU
1	0.859
2	0.872
3	0.886
4	0.889
5	**0.898**

The results show that my model achieved high IoU scores, indicating that it was able to accurately segment clouds in the SWIMSEG dataset. The IoU scores increased with each epoch, suggesting that the model was effectively learning to segment the clouds with more training better. However, a comparative analysis with other methods is required to determine the relative standing of my model in the research landscape. As of the time of writing, I do not have this comparative data.

I evaluate the performance of the proposed ResUNet model against current state-of-the-art cloud detection algorithms, using the F-score as the comparison metric. Li et al.[12] employ a combination of fixed and adaptive thresholds applied to the ratio of red and blue color channels to generate binary cloud maps. Souza et al.[24] leverage the saturation component from the Intensity-Hue-Saturation (IHS) color model for cloud detection in images. Dev et al.[13] implement a clustering technique on the ratio channel of the red and blue channels. Mantelli-Neto et al.[25] utilize the RGB color model for identifying clouds. Long et al.[26] model atmospheric scattering and establish a threshold for efficient cloud detection. Finally, I also compare my approach with three neural networks for cloud detection: FCN[27], PSPNet[28], and CloudSegNet[3]. By benchmarking my ResUNet against these well-established methods, I aim to demonstrate the effectiveness of my model in the task of cloud segmentation.

Table 5.2: Comparison of ResUNet with other cloud detection methods on SWIMSEG dataset using F-score metric.

Method	F-score
Li et al.[12]	0.86
Long et al.[26]	0.81
Souza et al.[24]	0.63
Dev et al.[13]	0.89
Mantelli-Neto et al.[25]	0.76
FCN[27]	0.45
PSPNet[28]	0.48
CloudSegNet[3]	0.89
ResUNet	**0.93**

It is evident from the results that my ResUNet model outperforms all the other tested methods, achieving the highest F-score of 0.93. This suggests that my model is highly effective in accurately detecting and segmenting clouds from the sky in the images. Among the other methods, Dev et al.[13] and CloudSegNet[3] achieve competitive results, with F-scores of 0.89 each. However, they are still surpassed by my model, which indicates the superiority of ResUNet in cloud segmentation tasks. This comparison validates the effectiveness of the ResUNet model in accurately segmenting clouds from sky images and demonstrates its potential applicability in real-world cloud detection and analysis tasks.

5.4 Experimental result on the UCL Webcam dataset

1. Experimental purpose

The purpose of this experiment was to test the model's performance on the UCL Webcam dataset. This dataset was captured by a new passive remote sensing instrument on the rooftop of a 10-story building on the UCL campus. It contains 10000 sky/cloud images, and unlike the other datasets, it covers the whole sky. Testing my model on this dataset allows us to assess its performance on a new and different type of data, and also to test its potential for real-world application.

2. Test indicator: Human Eye

3. Test images: 10000 sky/cloud images of 720 x 576 pix

5. Evaluation protocol

Unlike the SWIMSEG dataset, the UCL Webcam dataset does not have ground truth masks. Therefore, the evaluation protocol here is mainly qualitative. The predicted masks are visually compared with the original images to assess the accuracy of the cloud segmentation. In particular, I look for whether the model can accurately distinguish between cloud and non-cloud regions, and whether it can handle challenging conditions such as very bright or very dark images.

6. Result in analysis and case study

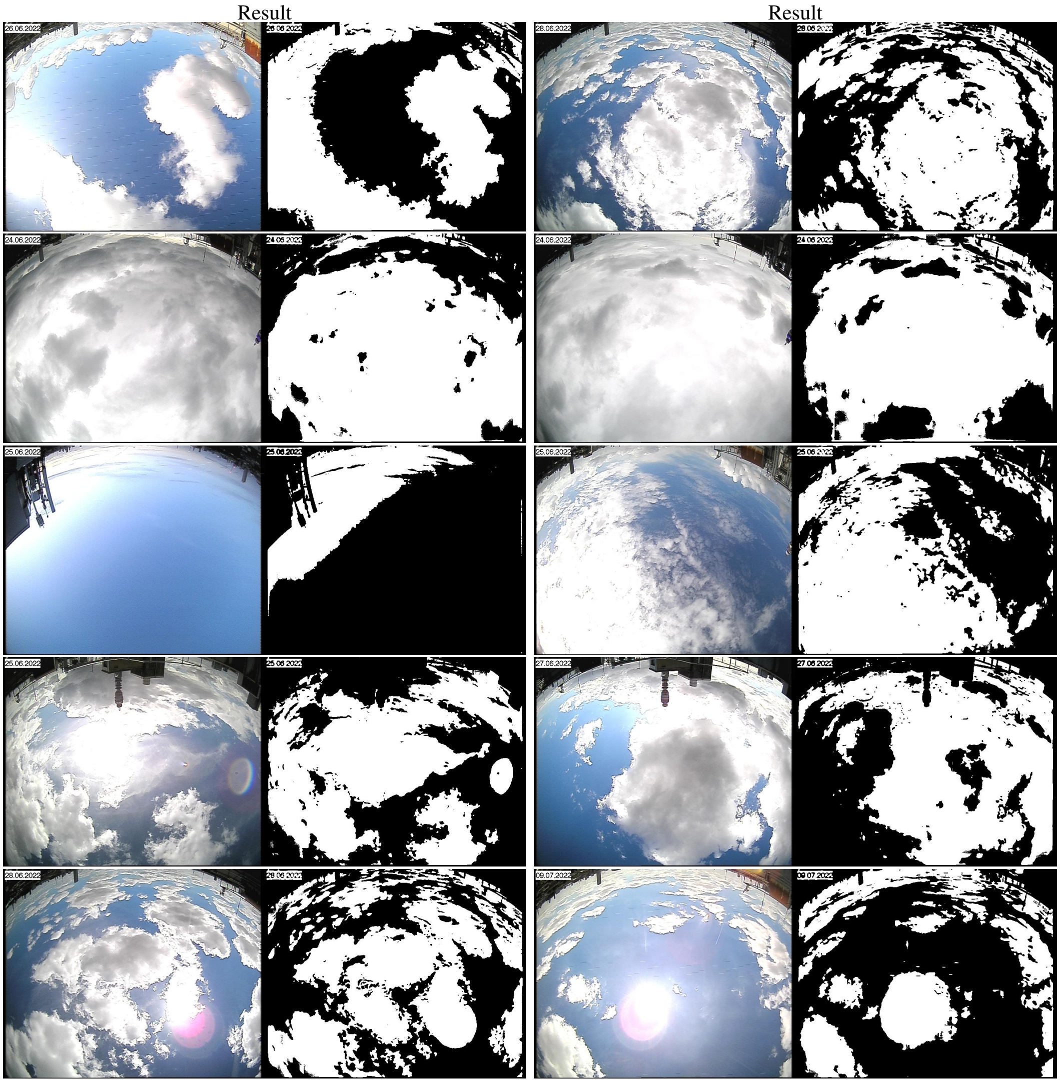


Figure 5.1: The test image and its predicted image towards my ResUNet neural network

The experiment results for this dataset are the predicted segmentation masks produced by the model. Since I do not have ground truth masks for this dataset, I cannot calculate numerical performance metrics such as IoU. However, visual inspection of the predicted masks suggests that the model performed well in segmenting the clouds in the UCL Webcam dataset. It was able to accurately segment clouds in images with a variety of conditions. However, there were some challenges. The below presents a qualitative analysis of the model's performance based on visual inspection.

1. **Blue Sky and White Clouds:** For images representing typical blue sky conditions with white clouds, the model performed exceptionally well. The predicted masks accurately delineated the

boundary between the sky and clouds, reflecting the model's ability to distinguish between these two classes.

2. **Solar Discrimination:** In images where the sun was visible, the model's performance was inconsistent. Since the sun appears as a bright white region in the images, it sometimes confused the model, leading it to misclassify the sun as a cloud. However, there were instances where the model was able to correctly classify the sun as non-cloud, indicating potential for improvement with further training or adjustment of the model's parameters.

3. **Bright Sky:** For images featuring a bright, cloudless sky, the model sometimes misclassified particularly bright regions as clouds. This suggests that the model may associate high intensity in the image with the presence of clouds, a bias possibly induced by the training data.

4. **Overcast Conditions:** In images with overcast skies, where the entire sky is covered with clouds, the model sometimes misclassified slightly brighter regions of the cloud cover as the sky. This is likely because the model has learned to associate brightness with the sky class.

These findings illustrate the strengths and weaknesses of the model. While the model generally performed well, the results also highlight areas for improvement. Addressing these challenges, such as the misclassification of the sun and bright regions, could further enhance the model's performance in sky/cloud segmentation tasks.

Chapter 6

Progress Review

This chapter discusses the progress of the whole project including project management and reflections on final year project work.

6.1 Project Management

The project started in the winter of 2022, and related tasks are being carried out every week, I meet with my mentor when I need to report important progress. The project is constructed using agile development, and the project process follows the basic process of agile development. Since the project is research-based, most of the following tasks depend on the results of the preceding tasks. The main task of the project is to design and train an efficient and high-accuracy neural network for cloud segmentation. Throughout the whole process, the preparation for the exam week started and ended longer than I expected, and the tuning of the network parameters was more complicated than expected. All of these variables gave the schedule some movement. The origin plan and actual plan are all shown below:

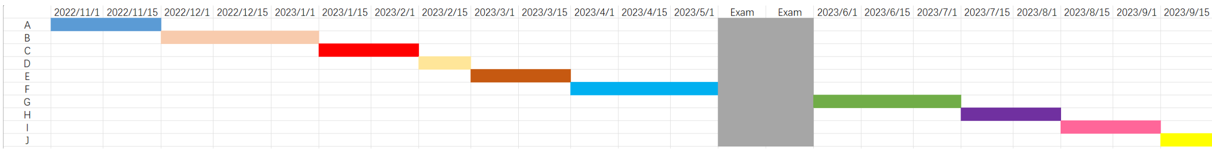


Figure 6.1: Figure2: Original Timeline

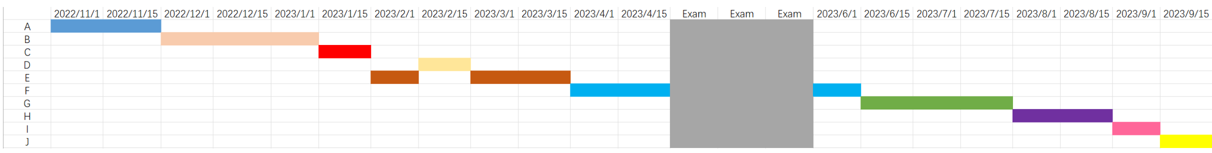


Figure 6.2: Figure2: Actual Timeline

Tasks of this project after updating are:

A. Determine the dissertation project

B. Literature review

C. Learn background technology

D. Submit the project outline

E. Train and validate the model

F. Rectify and test the model

G. Do all needed experiments

H. Writing the final dissertation

I. Modify and submit the final dissertation

J. Oral presentation of project

6.2 Reflections

6.2.1 Positive Reflections

Throughout this year-long project, I've gained invaluable insights into the realms of machine learning and computer vision. I actively engaged with my supervisor, exchanging ideas and taking part in in-depth discussions about the potential and challenges of cloud segmentation using artificial intelligence

In the early stages of this project, I faced the steep learning curve of deep learning with determination and an eagerness to explore this fascinating field. Despite the complexities of cloud detection and the nuances of neural network architectures, I was able to learn and apply theoretical concepts to practical problems, enhancing my understanding and skills in this area. The experience of designing and tuning a neural network model for my specific task was particularly rewarding, as I had the opportunity to delve into the intricate workings of the ResUNet model and fine-tune it for optimal performance.

Moreover, the interaction with my supervisor and the collaborative environment of the team greatly enriched my academic and social skills. Regular meetings and discussions with my supervisor not only guided me through the project but also improved my ability to articulate complex ideas, engage in academic discourse, and receive and implement feedback effectively.

The project also provided a unique opportunity to work with a novel dataset captured using UCL Webcam. Working with real-world, high-resolution sky/cloud images was an enriching experience, as it introduced me to the challenges of handling large-scale, complex datasets and the importance of data quality in machine learning tasks.

6.2.2 Negative Reflections

Despite the overall success of the project, I encountered several difficulties and setbacks along the way. Initially, I had a rather broad understanding of the project's objectives, considering it as a cloud detection problem rather than a more specific cloud segmentation task. This lack of clarity in the early stages led to a wider research scope than necessary, which, while informative, was not the most efficient approach.

Furthermore, my early attempts at training the neural network model were primarily conducted on the CPU, resulting in a slow and inefficient process. The transition to GPU-based training also presented its own set of challenges, mainly in terms of configuring the Cuda and cuDNN environments for optimal performance. These technical issues, while eventually resolved, did consume a significant amount of time and effort.

Another challenge was the lack of ground truth labels in the UCL Webcam dataset. Although I managed to overcome this issue by adopting semi-supervised learning techniques, it was a complex problem that required a creative solution.

6.2.3 Challenges Encountered and Solutions

As I mentioned in the negative reflection, this project has encountered some challenges in the process. One of the primary problems was the lack of clarity on the project objectives at the outset. This was resolved through in-depth discussions with my supervisor, which helped me gain a more precise understanding of the problem and refine the project's focus to cloud segmentation.

Technical issues related to the code execution environment and the transition from CPU to GPU-based training were other significant challenges. These were addressed by spending time understanding the requirements and dependencies of the tools and libraries, seeking help from peers and online resources, and trial and error.

The absence of ground truth labels in the UCL Webcam dataset was a major obstacle. However, the application of semi-supervised learning techniques provided a viable solution, allowing us to train the model effectively even in the absence of complete annotations.

Reflecting on these challenges, I recognize that there is ample room for improvement in the project, particularly in terms of optimizing the neural network model and improving the efficiency of the training process. Given more time, I believe I could have further enhanced the performance of the system and explored more sophisticated techniques for handling the complexities of the dataset. Nonetheless, the project has been a rewarding journey, providing me with valuable lessons and experience in deep learning, computer vision, and project management.

Chapter 7

Summary and Future Work

This chapter provides a conclusion of the project, summarizing the primary outcomes and achievements. It also proposes potential directions for future enhancements and developments in this field.

7.1 Conclusion

This dissertation presents the development of a deep learning model focused on cloud segmentation, with a specific emphasis on distinguishing between sky and cloud regions in images. A novel ResNet-based U-Net architecture was introduced and thoroughly evaluated on both the SWIMSEG and UCL Webcam datasets, the latter being a proprietary dataset created by the team, consisting of 10,000 high-resolution, whole-sky images.

A significant contribution of this project lies in the usage of the UCL Webcam dataset, which serves as a valuable benchmark for sky/cloud segmentation tasks, particularly for models trained on patch-based images. While the dataset remains private and solely used for testing purposes, it enhances the generalizability and robustness of the model and sets a new standard for future research in this domain.

The project encountered initial challenges in understanding project requirements and configuring the development environment. However, through diligent efforts, the project was completed, yielding high-performance results on both the SWIMSEG and UCL Webcam datasets. The model's performance was evaluated quantitatively using Intersection over Union (IoU) and F-score metrics, while qualitative analysis of segmented images on the UCL Webcam dataset provided valuable insights. Overall, this study contributes to the advancement of cloud segmentation research and deep learning applications in atmospheric analysis.

Through this project, I have gained significant experience in designing and implementing deep learning models, particularly in the field of computer vision. I developed a deeper understanding of the intricacies of machine learning models, the importance of dataset quality, and the impact of various factors on a model's performance. Regular discussions and interactions with my supervisor and peers have enhanced my academic communication skills and my ability to work in a research-oriented environment.

7.2 Future Work

Moving forward, there are several potential areas for improvement and exploration. Firstly, the complexity and sophistication of the U-Net decoder component in the architecture could be increased, possibly incorporating modifications based on variations of the U-Net model found in the literature. This could potentially enhance the model's performance by enabling it to capture more intricate patterns and details in the images.

Secondly, the current model is a binary segmentation model, distinguishing between cloud and non-cloud regions. An interesting extension to this work would be to develop a multi-class segmentation model. For instance, regions containing the sun, which are currently challenging to differentiate from clouds, could be classified as a separate category. This could provide more detailed and nuanced segmentation results.

Lastly, if possible, I intend to release the labeled UCL Webcam dataset, albeit currently restricted to private testing. The inclusion of ground truth labels in this dataset will enable quantitative evaluation of the model's performance, providing a more comprehensive understanding of the model's capabilities and areas that require further improvement.

The images below are my cloud segmentation results assumptions for future work:

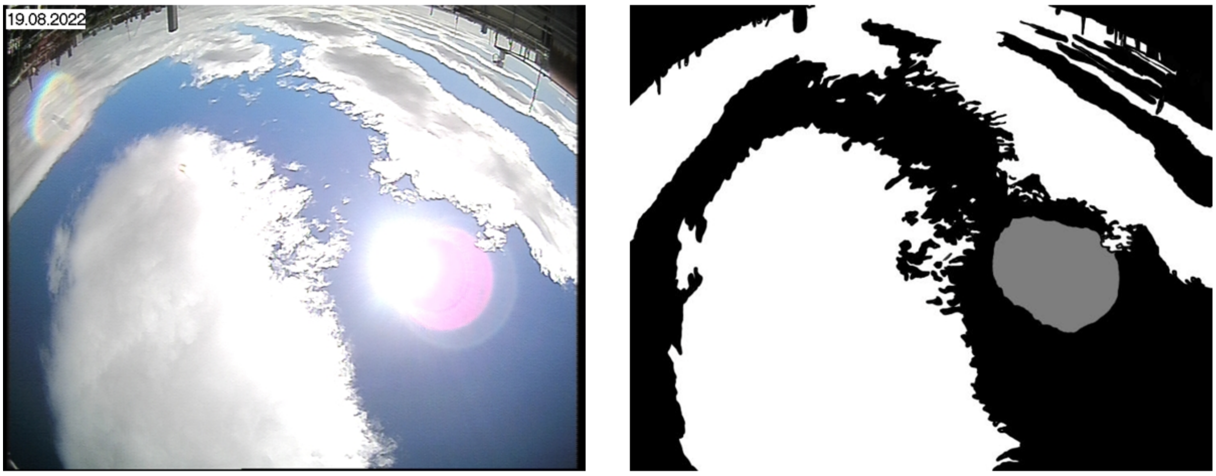


Figure 7.1: The left image represents the original input, while the right image displays the predicted result. The segmentation task aims to classify the image into three distinct regions: cloud regions are depicted in white, the sun is represented in gray, and non-cloud, non-sun regions are illustrated in black. This segmentation process facilitates the precise identification and separation of the various elements within the image, enabling effective characterization of cloud regions and the sun. The described visual depiction serves as a valuable reference for assessing the segmentation performance and showcases the successful outcomes of the segmentation algorithm.

References

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Soumyabrata Dev, Atul Nautiyal, Yee Hui Lee, and Stefan Winkler. Cloudsegnet: A deep network for nychthemeron cloud image segmentation. *IEEE Geoscience and Remote Sensing Letters*, 16(12):1814–1818, 2019.

[4] Soumyabrata Dev, Yee Hui Lee, and Stefan Winkler. Color-based segmentation of sky/-cloud images from ground-based cameras. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(1):231–242, 2016.

[5] Moustafa T Chahine. Remote sensing of cloud parameters. *Journal of Atmospheric Sciences*, 39(1):159–170, 1982.

[6] DW Slater, CN Long, and TP Tooman. Total sky imager/whole sky imager cloud fraction comparison. In *Proc. 11th ARM Science Team Meeting*, pages 1–11, 2001.

[7] W Paul Menzel and James FW Purdom. Introducing goes-i: The first of a new generation of geostationary operational environmental satellites. *Bulletin of the American Meteorological Society*, 75(5):757–782, 1994.

[8] Space Weather Prediction Center. National oceanic and atmospheric administration. *National Weather Service (NOAA/NWS), cited*, pages 1950–2011, 2012.

[9] E Oran Brigham. *The fast Fourier transform and its applications*. Prentice-Hall, Inc., 1988.

[10] Juan Huo and Daren Lu. Cloud determination of all-sky images under low-visibility conditions. *Journal of atmospheric and oceanic technology*, 26(10):2172–2181, 2009.

[11] B Ramesh and J Satheesh Kumar. Cloud detection and removal algorithm based on mean and hybrid methods. *International Journal of Computing Algorithm*, 2(1), 2013.

[12] Qingyong Li, Weitao Lu, and Jun Yang. A hybrid thresholding algorithm for cloud detection on ground-based color images. *Journal of atmospheric and oceanic technology*, 28(10):1286–1296, 2011.

[13] Soumyabrata Dev, Yee Hui Lee, and Stefan Winkler. Systematic study of color spaces and components for the segmentation of sky/cloud images. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5102–5106. IEEE, 2014.

[14] Vijay P Singh and Donald K Frevert. Watershed modeling. In *World water & environmental resources congress 2003*, pages 1–37, 2003.

[15] Alfonso Nieto-Castanon, Satrajit S Ghosh, Jason A Tourville, and Frank H Guenther. Region of interest based analysis of functional imaging data. *Neuroimage*, 19(4):1303–1316, 2003.

[16] Shuang Liu, Linbo Zhang, Zhong Zhang, Chunheng Wang, and Baihua Xiao. Automatic cloud detection for all-sky images using superpixel segmentation. *IEEE Geoscience and Remote Sensing Letters*, 12(2):354–358, 2014.

[17] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[18] Chaojun Shi, Yatong Zhou, Bo Qiu, Jingfei He, Mu Ding, and Shiya Wei. Diurnal and nocturnal cloud segmentation of all-sky imager (asi) images using enhancement fully convolutional networks. *Atmospheric Measurement Techniques*, 12(9):4713–4724, 2019.

[19] Liang Ye, Zhiguo Cao, Zhibiao Yang, and Huasong Min. Ccad-net: A cascade cloud attribute discrimination network for cloud genera segmentation in whole-sky images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[21] Soumyabrata Dev, Yee Hui Lee, and Stefan Winkler. Multi-level semantic labeling of sky/cloud images. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 636–640. IEEE, 2015.

[22] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[23] Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European conference on information retrieval*, pages 345–359. Springer, 2005.

[24] Mariza Pereira Souza-Echer, Enio Bueno Pereira, LS Bins, and MAR Andrade. A simple method for the assessment of the cloud cover state in high-latitude regions by a ground-based digital camera. *Journal of Atmospheric and Oceanic Technology*, 23(3):437–447, 2006.

[25] Sylvio Luiz Mantelli Neto, Aldo Von Wangenheim, Enio Bueno Pereira, and Eros Comunello. The use of euclidean geometric distance on rgb color space for the classification of sky and cloud patterns. *Journal of Atmospheric and Oceanic Technology*, 27(9):1504–1517, 2010.

[26] Charles N Long, Jeff M Sabburg, Josep Calbó, and David Pagès. Retrieving cloud characteristics from ground-based daytime color all-sky images. *Journal of Atmospheric and Oceanic Technology*, 23(5):633–652, 2006.

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[28] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.