FEDERAL UNIVERSITY OF SANTA CATARINA
COMPUTER SCIENCE GRADUATE PROGRAM

Bruno Juncklaus Martins

Segmentation and Classification of Individual Clouds in Images captured with Horizon-Aimed Cameras for Nowcasting of Solar Irradiance Absorption

Florianópolis
2024

Bruno Juncklaus Martins

Segmentation and Classification of Individual Clouds in Images captured with Horizon-Aimed Cameras for Nowcasting of Solar Irradiance Absorption

Master thesis submitted to the Computer Science Graduate Program of the Federal University of Santa Catarina in fulfillment of the requirements for the degree of Master of Science.
Advisor: Aldo von Wangenheim.

Florianópolis
2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Martins, Bruno Juncklaus
Segmentation and Classification of Individual Clouds in Images captured with Horizon-Aimed Cameras for Nowcasting of Solar Irradiance Absorption / Bruno Juncklaus Martins ; orientador, Aldo von Wangenheim, 2024.
147 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Ciência da Computação, Florianópolis, 2024.

Inclui referências.

1. Ciência da Computação. 2. Classificação de Nuvens. 3. Aprendizado Profundo. 4. Visão Computacional. 5. Segmentação Semântica. I. von Wangenheim, Aldo. II. Universidade Federal de Santa Catarina. Programa de Pós Graduação em Ciência da Computação. III. Título.

Bruno Juncklaus Martins

Segmentation and Classification of Individual Clouds in Images captured with Horizon-Aimed Cameras for Nowcasting of Solar Irradiance Absorption

This master thesis is recommended in partial fulfillment of the requirements for the degree of Master of Science which has been approved in its present form by the Graduate Program in Computer Science:

Sylvio Mantelli
INPE

Antonio Carlos Sobieranski
INE/UFSC

Mauro Roisenberg
INE/UFSC

This is the **original and final work** properly evaluated to acquire the tittle of Master's degree in the Computer Science Graduate Program.

ICP-Edu
Documento assinado digitalmente
Marcio Bastos Castro
Data: 03/05/2024 14:50:44-0300
CPF: ***.707.720-**
Verifique as assinaturas em https://v.ufsc.br

Márcio Bastos Castro
Graduate Program Coordinator

ICP-Edu
Documento assinado digitalmente
Aldo Von Wangenheim
Data: 04/05/2024 05:51:23-0300
CPF: ***.566.679-**
Verifique as assinaturas em https://v.ufsc.br

Aldo von Wangenheim
Advisor

Florianópolis, February 21, 2024

I dedicate this work to my dear friend Karthik Garimella.

ACKNOWLEDGEMENTS

I want to thank professor Ricardo Rüther which allowed the development of the current research in the Fotovoltaic Laboratory facilities, and helped our research team at Lapix with exceptional suggestions. Thank you professor.

I'd also like to thank professor and currently climatologist at Aqueris Engenharia e Soluções Ambientais, Maurici Amantino Monteiro. With several years of experience in synoptic observation, he helped our team to understand the true nature of clouds and associate them with our research.

Also thanks to Karthik Garimella, currently a Computer Science PhD student at the New York University, for all the help provided throughout my graduation.

Last but not least, I want to thank the PPGCC program for fostering this collaborative and scientific environment and help me develop my research skills.

RESUMO

Um aspecto crucial da geração de energia solar, particularmente em locais intertropicais, é a variabilidade local das nuvens. Imagens de satélite não possuem resolução temporal necessária para a previsão imediata dos impactos em usinas solares, o que torna o monitoramento por câmeras terrestres essencial. A detecção e monitoramento de nuvens apresentam desafios significativos devido à natureza dinâmica das formas das nuvens, às limitações dos dispositivos de câmeras lineares e autoajustáveis e às distorções introduzidas por lentes olho-de-peixe. Este trabalho é dedicado ao avanço das técnicas de segmentação de nuvens com o objetivo final de prever os impactos das nuvens na geração de energia solar, que variam de acordo com a climatologia e geografia únicas de cada local. Utilizando câmeras baseadas em Raspberry-Pi apontadas para o horizonte, na região de Florianópolis, Brasil, este estudo supera as limitações das lentes olho-de-peixe, possibilitando a observação da distribuição vertical das nuvens. Uma análise extensiva de imagens de nuvens levou à adoção de métodos de aprendizado profundo como U-net, HRNet e Detectron, com aprendizado por transferência aplicado a partir de pesos treinados no conjunto de dados ”2012 ILSVRC ImageNet” e configurações arquitetônicas como ResNet, EfficientNet e RCNN. O Experimento 28, utilizando uma arquitetura U-net com ResNet 18, alcançou um IoU médio de 0,564, demonstrando resultados promissores na classificação de nuvens em uma resolução mais baixa. O Experimento 44 melhorou ainda mais o desempenho, superando outros modelos com um IoU médio de 0,594, destacando a eficácia de uma abordagem arquitetônica mais simples, porém robusta. Apesar dos desafios impostos pela variabilidade na frequência das nuvens e condições atmosféricas, esses modelos mostraram potencial significativo na classificação automatizada de nuvens, levando a previsões mais precisas para os impactos na energia solar. Os resultados sublinham a necessidade de desenvolvimento contínuo em métodos de segmentação e ajuste de modelos para lidar com a complexidade dos padrões das nuvens. Embora a identificação de nuvens permaneça uma tarefa complexa, esta pesquisa demonstrou que modelos de aprendizado profundo mais simples muitas vezes superam os mais complexos, e a augmentação de dados desempenha um papel crítico no aumento da robustez e generalização do modelo. No entanto, a variabilidade nas frequências dos tipos de nuvens, condições atmosféricas e época do ano apresenta desafios significativos para a comparação direta com a literatura existente. As experiências deste estudo revelaram que, enquanto os modelos CNN convencionais oferecem desempenho confiável, há uma necessidade premente de avanços para lidar de forma eficaz com classes de nuvens mais intrincadas.

Os achados reforçam a necessidade de experimentação contínua no campo da segmentação de imagens e o desenvolvimento de modelos sofisticados e contextualmente conscientes para enfrentar a natureza multifacetada das tarefas de segmentação de nuvens.

Palavras-chave: Segmentação de Nuvens; Energia Solar; Aprendizado Profundo; Imagens do Horizonte.

RESUMO EXPANDIDO

Introdução

A presença de nuvens tem um efeito significativo nas usinas fotovoltaicas, causando variabilidade na energia solar que atinge a superfície, e consequentemente, na geração de energia elétrica. A detecção de nuvens e a estimativa de seus impactos nas usinas solares são tarefas desafiadoras, dada a metamorfose contínua das nuvens, a escala logarítmica de sua luminosidade e a variedade e dinâmica de suas formas, sempre associadas à geografia local e às condições climáticas atuais. Os diferentes tipos de nuvens e altitudes também têm efeitos distintos na dispersão, reflexão e absorção da energia solar, influenciando a produção de energia. As variações na espessura, forma e volume das nuvens podem causar mudanças repentinas na cobertura do céu, resultando em alterações significativas na radiação ao longo do dia.

A Organização Meteorológica Mundial (OMM) classifica as nuvens pela forma, agrupamento e altura da base. A estimativa do tipo e cobertura das nuvens é feita por um operador sinótico. O desenvolvimento de sistemas e métodos de observação automatizados ainda é um assunto em aberto, especialmente em termos de substituição da percepção altamente desenvolvida de um classificador humano.

Sistemas comerciais automatizados, como o Whole Sky Imager (WSI), estão disponíveis para identificação de nuvens, avaliando seu impacto na geração de energia. No entanto, esses sistemas apresentam problemas ao lidar com imagens usadas para previsões imediatas. As imagens do WSI mostram em detalhes apenas as nuvens que estão na posição do zênite, mas perto do horizonte, as nuvens parecem comprimidas e a imagem degrada nos detalhes.

Este estudo visa fornecer informações fundamentais sobre as categorias amplas de nuvens, que podem ser refinadas posteriormente para abordar tipos específicos de nuvens em detalhes. Devido à rara presença de nuvens do tipo *Cumulonimbusform* na região, esta categoria foi removida do conjunto de dados criado.

A identificação e classificação de nuvens perto do horizonte e a previsão de seu caminho em direção a uma instalação fotovoltaica ainda é um campo de pesquisa aberto. Outras configurações de métodos combinados com a câmera, bem como dados reais orientados por aprendizado de máquina, também podem ser explorados. O aprendizado de máquina tem avançado nos últimos anos quando se trata de previsão de irradiação solar.

Diante dos desafios e lacunas nas metodologias atuais para classificação de nuvens e previsão de irradiação, este trabalho visa explorar e avaliar metodologias existentes para segmentação de nuvens. O objetivo é identificar uma abordagem confiável para a classificação de tipos de nuvens e avaliar a viabilidade de automatizar esse processo usando técnicas de aprendizado de máquina. A pesquisa começa com uma revisão de estudos anteriores sobre o assunto, conforme delineado na seção de revisão da literatura.

O estudo avança ao criticar o uso de lentes olho-de-peixe para capturar imagens de nuvens, que se mostraram subótimas para discernir a distribuição vertical das camadas de nuvens. Como alternativa, foram desenvolvidos dois sistemas baseados no modelo Raspberry PI 2, oferecendo qualidade de imagem comparável aos Whole Sky Imagers (WSIs), orientados na direção predominante do movimento das nuvens, estabelecida com a experiência de meteorologistas locais.

Objetivos

Os objetivos dessa dissertação de mestrado são:

- Coletar imagens terrestres de nuvens por meio de sistemas de computador de placa única com câmeras apontando para o horizonte.
- Criar um conjunto de dados rotulado das imagens capturadas.
- Desenvolver e comparar modelos de aprendizado de máquina para classificação de nuvens usando imagens terrestres.
- Analisar e realizar experimentos com modelos atuais de aprendizado de máquina para segmentação de nuvem.
- Criar um conjunto de dados rotulado de imagens de nuvens capturadas no solo com câmeras apontadas para o horizonte.
- Desenvolver e comparar modelos de aprendizado de máquina para classificação de nuvens usando imagens terrestres.
- Analisar e experimentar modelos atuais de aprendizado de máquina para segmentação.

Metodologia

Imagens de nuvens foram capturadas através do sistema Nimbus Gazer. Três versões do conjunto de dados foram criados para os experimentos realizados, com 450, 1000 e 1500 imagens, respectivamente. As principais métricas utilizadas para validação foram a Intersection over Union (IoU), precision e recall. Inicialmente apenas 25 experimentos foram realizados para dar direcionamento à pesquisa e ajudar a levantar pontos de melhoria para os próximos experimentos. A segunda bateria de experimentos foi composta por um total de 9 experimentos utilizando técnicas mais aprimoradas que potencialmente levariam a um resultado melhor que a versão anterior de experimentos. A última bateria de experimentos foi composta por 29 experimentos, combinando todo o aprendizado adquirido nas experimentações anteriores e exaustando configurações de cada experimento realizado.

Resultados e discussão

Esta pesquisa ajudou a entender melhor quais técnicas funcionam melhor para a segmentação de nuvens. Também é evidente que o desbalanceamento de dados está afetando o desempenho de todos os modelos desenvolvidos. No geral, enquanto modelos CNN convencionais, particularmente quando combinados com U-net, ofereceram um desempenho mais confiável em vários tipos de nuvens, eles também exigem aprimoramentos adicionais para as classes mais complexas de nuvens Cirriformes e Cumuliformes. O modelo HRNet parece mais promissor, pois trabalha com diferentes resoluções, levando a uma segmentação mais refinada, no nível do pixel. Mesmo assim, alguns resultados parecem indicar que um modelo tão complexo não é necessário para detectar as nuvens mais predominantes no céu. Um modelo mais simples, usando U-net com Resnet 18, conseguiu alcançar resultados satisfatórios, usando uma resolução muito mais baixa. Isso pode ser útil no futuro, já que o objetivo principal é usar tais modelos para prever o movimento das nuvens e prever o impacto que terão na geração de energia solar. A lista abaixo sumariza os principais aprendizados com essa pesquisa.

- Arquiteturas de modelo mais simples frequentemente superam as mais complexas para tarefas de segmentação de nuvens.
- Modelos de maior resolução podem levar a uma super-segmentação devido às limitações de precisão das anotações ground-truth.
- A classe Árvore consistentemente apresenta altas métricas de desempenho, indicando um potencial viés na avaliação do modelo.

- Augmentação de dados é fundamental para a robustez do modelo, mas precisa ser cuidadosamente adaptado para evitar generalização subótima.

- Existe um compromisso entre a sensibilidade a classes específicas de nuvens e o desempenho geral do modelo em vários tipos.

- Modelos Transformers, apesar de suas fortes capacidades contextuais globais, não superam significativamente os modelos baseados em CNN para padrões complexos de nuvens.

- A super-representação de certas classes de nuvens em conjuntos de dados apresenta desafios para o treinamento e generalização do modelo.

- Validação detalhada é necessária para garantir que os resultados de segmentação representem as nuvens com mais precisão do que as anotações ground-truth.

- Pesquisas futuras devem considerar modelos híbridos, augmentação de dados avançada e manejo eficaz em relação ao desbalanceamento de classe.

- Augmentação de dados, embora benéfico, não pode mitigar completamente os efeitos da super-representação de classe.

- O alto desempenho de classes facilmente segmentáveis no conjunto de dados pode não se traduzir em uma melhoria da segmentação para tipos complexos de nuvens.

Em comparação com os experimentos iniciais realizados, pode-se observar uma melhoria nas métricas gerais e na qualidade da segmentação. Os resultados obtidos levantaram questões sobre por que um modelo mais simples supera um mais complexo, o que leva à necessidade de investigações futuras. Seis causas potenciais foram identificadas para exploração futura: 1) Overfitting, pois modelos complexos com mais parâmetros são propensos a overfitting, enquanto modelos mais simples podem generalizar melhor; 2) Complexidade apropriada, onde a tarefa de segmentação de nuvens pode não ser tão complexa para um modelo de aprendizado de máquina quanto inicialmente se pensava; 3) Disponibilidade de dados, já que modelos complexos requerem mais dados para aprender efetivamente, enquanto modelos mais simples podem ter melhor desempenho com dados limitados; 4) Ajuste de hiperparâmetros, uma vez que modelos complexos possuem mais hiperparâmetros que precisam de ajuste ótimo para um desempenho ideal; 5) Técnicas de regularização como dropout, weight decay ou early stop, que podem prevenir overfitting em modelos complexos; e 6) Qualidade dos dados, onde um modelo mais simples pode ser mais robusto contra dados ruidosos. Esses fatores serão abordados em trabalhos futuros para obter mais informações.

Considerações finais

É importante ressaltar que não foi realizada uma análise detalhada para validar se os resultados representavam de fato as nuvens melhor do que a verdade básica, mesmo que o conjunto de dados tenha sido criado com a ajuda de especialistas, sempre há a probabilidade de erro humano ao determinar um tipo de classe durante a anotação de imagens. Para estudos futuros, recomenda-se um modelo de classificação de nuvens melhor e mais abrangente com base nos resultados apresentados nesta pesquisa.

Palavras-chave: Segmentação de Nuvens; Energia Solar; Aprendizado Profundo; Imagens do Horizonte.

ABSTRACT

One crucial aspect of solar energy generation, particularly in inter-tropical sites, is the local variability of clouds. Satellite imagery lacks the temporal resolution necessary for nowcasting the impacts on solar plants, thus necessitating monitoring by ground-based cameras. Cloud detection and monitoring pose significant challenges due to the dynamic nature of cloud shapes, the limitations of linear and self-adjusting camera devices, and distortions introduced by fish-eye lenses. This work is dedicated to advancing cloud segmentation techniques with the ultimate goal of predicting cloud impacts on solar energy generation, which vary according to each site's unique climatology and geography. Utilizing Raspberry-Pi-based cameras pointed at the horizon, in the region of Florianópolis, Brazil, this study overcomes the limitations of fish-eye lenses, enabling the observation of clouds' vertical distribution. An extensive analysis of cloud images has led to the adoption of deep learning methods such as U-net, HRNet, and Detectron, with transfer learning applied from weights trained on the ”2012 ILSVRC ImageNet” dataset and architectural configurations like ResNet, EfficientNet, and RCNN. Experiment 28, utilizing a U-net with ResNet 18 architecture, achieved an average IoU of 0.564, demonstrating promising results in cloud classification at a lower resolution. Experiment 44 further improved the performance, surpassing other models with an average IoU of 0.594, highlighting the effectiveness of a simpler, yet robust, architectural approach. Despite the challenges posed by variability in cloud frequency and atmospheric conditions, these models have shown significant potential in automated cloud classification, leading to more accurate nowcasting for solar energy impacts. The findings underscore the need for continuous development in segmentation methods and model tuning to address the complexities of cloud patterns. Although cloud identification remains a complex task, this research has demonstrated that simpler deep learning models often outperform more complex ones, and data augmentation plays a critical role in enhancing model robustness and generalization. Nevertheless, the variability in cloud type frequencies, atmospheric conditions, and the time of year presents significant challenges for direct comparison with existing literature. This study's experiments have revealed that while conventional CNN models offer reliable performance, there is a pressing need for further advancements to handle more intricate cloud classes effectively. The findings underscore the necessity for continuous experimentation in the field of image segmentation and the development of sophisticated, contextually aware models to address the multifaceted nature of cloud segmentation tasks.

Keywords: Cloud Segmentation; Solar Energy; Deep Learning; Horizon Images.

List of Figures

List of Tables

LIST OF ABBREVIATIONS AND ACRONYMS

WMO	Meteorological Organization
WSI	Whole Sky Imager
ANN	Artificial Neural Networks
CNN	Convolutional Neural Networks
GLCM	Gray-Level Co-Occurrence Matrix
SVC	Support Vector Classification
IoU	Intersection Over Union
CSL	Clear-Sky Library
FCN	Fully-Convolutional Network
OS	Operational System
COCO	Common Objects in Context
Resnet	Residual Network
HRNet	High-Resolution Network

Contents

1 Introduction

The presence of clouds has a major effect on the photovoltaic power plants, causing significant variability in solar energy that reaches the surface, and as a consequence, in the power energy generation HU and STAMNES [2000]. The detection of clouds and the estimation of their impacts on solar plants is a challenging task. Clouds are always in continuous metamorphosis. The logarithmic scale of its luminosity Mantelli et al. [(2020], the variety and dynamic of their shapes, along with their forming and extinction processes are always associated with local geography and current weather conditions. Different types of clouds and altitudes, also have distinct effects on the scattering, reflection, and absorption of solar energy, influencing energy power production. Different thicknesses, shapes, and volumes of clouds, could cause sudden changes in sky coverage trapping and releasing long and short waves resulting in significant changes in radiation throughout the day.

The World Meteorological Organization (WMO) classify clouds by their shape, clustering, and height of their base. According to WMO[1] clouds can also be divided by groups into specific categories such as species, variety, and additional supplementary features, as described in WMO Cloud Atlas[2]. The estimation of cloud type and coverage is made by a synoptic operator. The development of automated systems and methods of observation is still an open subject. Especially in terms of the replacement of the highly developed perception of a human observer classification.

There are commercial automated solutions available for cloud identification, in order to assess their impact on energy generation like Whole Sky Imager (WSI), Juncklaus Martins et al. [(2021,2]. This system can be configured to use single or double fish-eye surface cameras. They use pixel value analysis, and stereo techniques to evaluate the clouds. The single system poses a problem when dealing with images used for nowcasting. WSI images show in detail only clouds that lie on the zenith position. Near the horizon and close to the lens border, clouds seem to be compressed and the image degrades in the details. Double fish-eye images are coupled with additional geometric and stereo technology to determine cloud-based. But the embedded software and additional cameras are expensive and they have to be placed kilometers apart. One important feature of cloud classification is its vertical distribution in different layers. The pixel value analysis used is still far from achieving the classification proposed by WMO

[1] https://public.wmo.int/en

[2] https://cloudatlas.wmo.int/en/cloud-classification-summary.html

Mantelli et al. [(2020].

It is desirable to estimate cloud shade casting in detail, especially when it causes a partial coverage of large power plants. Scattered cloud's condition throughout the day has intermittent effects on the generation and does not cause only attenuation in energy. But also a surplus is known as over-irradiation by multiple reflections that result in levels of irradiance above the top of atmosphere values Martins et al. [(2022]. This excess could result in some operational problems with inverters, unbalanced energy generation among module strings, overloads, and even safety shutdowns do Nascimento et al. [(2019, (2020]. Therefore, it is important to have tools to model and predict the energy generated by photovoltaic technologies Tarrojam et al. [(2012], especially when associated with storage systems. Many energy grids combine power from multiple sources. Predicting solar power output, using accurate cloud forecasting, helps grid managers decide when to tap into alternative energy sources like wind or hydropower, ensuring a steady power supply to consumers. Additionally, precise prediction of cloud patterns allows power plants to anticipate and adjust for these variations, ensuring more consistent power output. Consistent and predictable power generation can lead to stable financial returns, since power plants can face penalties or reduced rates if they fail to deliver the promised power output to the grid. Accurate forecasting through cloud segmentation can help in avoiding such scenarios.

From the computer vision point of view, clouds could be segmented and their pathways monitored by tracking. Their impact on energy generation is measured by determining the present solar position combined with the geometric estimation of clouds shading over the power plant. There are several segmentation methods used in the classification of clouds. Mostly based on their shapes and inner features like texture, color similarity, brightness, and contour continuity in an image Long et al. [2006], Mantelli et al. [2010], Mejia et al. [2016], Piccardi [2004], Souza-Echer et al. [2006]. The albedo of a cloud has inherent characteristics that are distinguished from common objects and outdoor scene features. Its reflectivity in the visible spectrum is higher than the other wavelengths and its luminance values are usually cropped due to camera scale limitations Mantelli et al. [(2020]. In general, objects only reflect the local surrounding radiation and this approach does not comprehensively describe albedo features and scenery under the sun. Therefore, the use of the brightness parameter is not accurate enough to distinguish a cloud. Cloud textures are random and their diffuse edges contain gray level jumps which are more similar to a phase step in large areas. To a certain degree, smaller parts of clouds are similar to the whole, and the cloud cluster has a certain fractal similarity Li et al. [(2015]. The shape, size, formation, extinction, and changing level are variable along the cloud's pathway which made it difficult to monitor their surface shades.

Some computer vision-based methods rely on cross-classification and divide clouds into broader physical forms. These classifications are based on the shared properties of clouds, such as opacity, structure, and formation processes. Specifically, following the classification proposed by Barrett and Grant [1976], clouds can be categorized as follows:

1. *Stratiform*, grouping Cirrostratus, Altostratus, Stratus, and Nimbostratus.

2. *Cirriform*, which only includes Cirrus.

3. *Stratocumuliform*, encompassing Cirrocumulus, Altocumulus, and Stratocumulus.

4. *Cumuliform*, containing only Cumulus.

5. *Cumulonimbusform*, exclusive to Cumulonimbus.

These groupings were chosen to explore the broader categories, understanding that there may be variations within each group. This study aims to provide foundational insights into these groupings, which can later be refined to address specific cloud types in detail. However, due to the rare presence of *Cumulonimbusform* clouds in the region, this category was removed from the created dataset.

As mentioned before, the identification and classification of clouds near the horizon and the prediction of their path toward a photovoltaic installation is still an open research field. Other configurations of methods combined with the camera as well as real data-oriented by machine learning could also be explored. Machine learning has gained some ground in recent years when it comes to solar irradiation prediction Juncklaus Martins et al. [(2021,2], Kumari and Toshniwal [(2021]. This is due to the popularization and easy access to artificial intelligence frameworks, which have several ready-to-use models for image segmentation and detection. There are several recent reviews on this subject made by Juncklaus Martins et al. [(2021], Kumari and Toshniwal [(2021], Mellit and Kalogirou [(2008], Pelland et al. [(2013], Voyant et al. [(2017] describing recent methods recently used, but they're no comparative evaluation of performance among them.

The research question of this study is ”Can deep learning techniques improve the segmentation and classification of individual clouds in horizon-aimed camera images for nowcasting of solar irradiance absorption usage?".

Considering the identified challenges and gaps in the current methodologies for cloud classification and irradiation prediction, this work aims to explore and evaluate existing methodologies for cloud segmentation. The objective is to identify a reliable approach for cloud type classification and assess the feasibility of automating this process using machine learning techniques. The research commences with a review of prior studies on the subject matter, as outlined in the literature review section.

The specific objectives are:

- Collect terrestrial images of clouds via single-board computer systems with cameras aiming toward the horizon.
- Create a labeled dataset of cloud images captured on the ground with cameras pointed toward the horizon.
- Develop and compare machine learning models for cloud classification using terrestrial images.
- Analyze and experiment with current machine learning models for cloud segmentation.

The independent variables are:

- Types of Deep Learning Techniques: Different models and algorithms such as U-net, HRNet, Detectron, etc.
- Parameters of the Models: Learning rates, number of layers, types of layers, etc.
- Image Quality and Type: Different resolutions, angles, and formats of horizon-aimed camera images.

In contrast, the dependent variables of the study are:

- Accuracy of Cloud Segmentation and Classification: How accurately the clouds are identified and classified in the images by the deep learning models, quantified by the suite of metrics compiled throughout the experimental trials.
- Effectiveness in Nowcasting Solar Irradiance Absorption: Measured in terms of the precision of predictions about solar irradiance absorption based on cloud types.

The study advances by critiquing the use of fish-eye lenses for capturing cloud imagery, which was found to be suboptimal for discerning the vertical distribution of cloud layers. As an alternative, two systems based on Raspberry PI model 2, offering comparable imaging quality to WSIs, were developed. These systems were oriented towards the predominant cloud movement directions, established with the expertise of local meteorologists Monteiro [(2001]. Real image datasets were then utilized to evaluate the performance of various frameworks in cloud classification tasks.

In the Methodology section, the process of dataset production is elaborated, along with a comprehensive account of the experimental procedures. The results are systematically presented in the corresponding experiments section, providing an initial analysis of the performance of the models developed. The Discussion section delves into an in-depth analysis of all pertinent findings, outcomes, and considerations.

Lastly, the Conclusion section synthesizes the insights gained from this research, articulating the implications for future technological advancements in cloud classification and irradiation prediction. It encapsulates the contributions of the study to the field and outlines prospective avenues for continued exploration and innovation.

2 Related Works

Several machine learning techniques have been used to forecast solar irradiance in the past years Kumari and Toshniwal [(2021], Martins et al. [(2022], Voyant et al. [(2017]. Some perform cloud identification by doing a binary image segmentation on either a patch of the sky or using WSI. Others use the physical properties of clouds and the interaction with light and atmosphere while others use current meteorological data or exogenous data Voyant et al. [(2017] from side stations.

Machine learning techniques can be classified as Support Vectors, K-means, Artificial Neural Networks (ANN), and Convolutional Neural Networks (CNN). For example, in Paletta and Lasenby [(2020], the dataset used in this study originated from the SIRTA laboratory Haeffelin (2005), France. The RGB images were collected over a period of seven months from March 2018 to September 2018, with a resolution of 768 x 1024 pixels. The work is composed of two distinct networks merged into one which outputs the irradiance estimate. On one side, a ResNet CNN is used to extract features from sky images and on the other side, an ANN treats available auxiliary data (past irradiance measurements, the angular position of the sun, etc). Both outputs are fed into another ANN, which integrates them to give its prediction.

In Anagnostos et al. [(2019], sky images are retrieved every 10s from sunrise to sunset with a camera equipped with a fisheye lens and 1920 x 1920 pixels resolution. Specific image features are computed for each image, then provided as inputs for the machine learning applications. The authors quantify characteristics such as image texture, color values and other metrics. The extracted image features are used for irradiance modeling (k-neighbors neural network model), cloud classification (support vector classification model) and the energy yield prediction with neural networks. First, image Contrast is retrieved from the gray-level co-occurrence matrix (GLCM) which is used as input for the neural network. Additionally, image features are derived from RGB channel-based color statistics and the solar position cloud coverage ratio are used as additional inputs for the neural network. The sky imaging software determines for each image the predominant sky or cloud type as one of seven categories: Cumulus (Cu); Cirrostratus (Cs), Cirrus (Ci); Cirrocumulus (Cc), Altocumulus (Ac); Clear sky (Clear); Stratocumulus (Sc); Stratus (St), Altostratus (As); Nimbostratus (Ns), Cumulonimbus (Cb). The Support Vector Classification (SVC) has been chosen with best classification results, achieving an accuracy of more than 99% of correct classifications.

The authors in Fabel et al. [(2022] focus on the semantic segmentation of ground-based all-sky images (ASIs) to provide high-resolution cloud coverage information of distinct cloud types. The authors acknowledge the challenges in classifying clouds due to their variable shape and appearance, and the high similarity between cloud types. Therefore, most state-of-the-art methods focus on distinguishing between cloudy and cloud-free pixels without considering the cloud type. To address these challenges, the authors propose a self-supervised learning approach that leverages a large amount of data for training, thereby increasing the model's performance. They use about 300,000 ASIs in two different pretext tasks for pretraining. One task focuses on image reconstruction, while the other is based on the DeepCluster model, an iterative procedure of clustering and classifying the neural network output. The model is then fine-tuned on a small labeled dataset of 770 ASIs. The results of the study show that their self-supervised model outperforms conventional approaches of random and pretrained ImageNet initialization. The model achieved 85.75% pixel accuracy on average, compared to 78.34% for random initialization and 82.05% for pretrained ImageNet initialization. The improvement was even more significant when considering precision, recall, and Intersection over Union (IoU) of the respective cloud classes, where the improvement ranged between 5 and 20 percentage points, depending on the class. Furthermore, when compared to a clear-sky library (CSL) from the literature for binary segmentation, their model outperformed the CSL by over 7 percentage points, reaching a pixel accuracy of 95.15%.

The study of Ye et al. [(2019] discusses the challenges of fine-grained cloud detection in different regions with varying air qualities. For instance, the authors collected WSIs from Hangzhou, a densely populated city with low air quality, and Lijiang, a sparsely populated plateau area with high air quality. The differences in these regions add complexity to the cloud detection problem. The authors also discuss the limitations of existing methods for cloud detection, such as threshold segmentation, graph-based methods, and superpixel-based methods. They argue that these methods often ignore cloud-type classification or treat it as a separate task from cloud detection. The authors tested their proposed method for fine-grained cloud detection and recognition against a well-known semantic segmentation model, fully-convolutional network (FCN). They fine-tuned a pre-trained FCN model with 400 images from their dataset, which included images from Lijiang and Hangzhou and used 8 cloud types and the sky as ground truth label classes. The results showed that their approach outperformed the FCN model. They noted that due to the superpixel segmentation, their method was able to maintain edges better than FCN, despite causing fragmentary classification errors in a very small number of

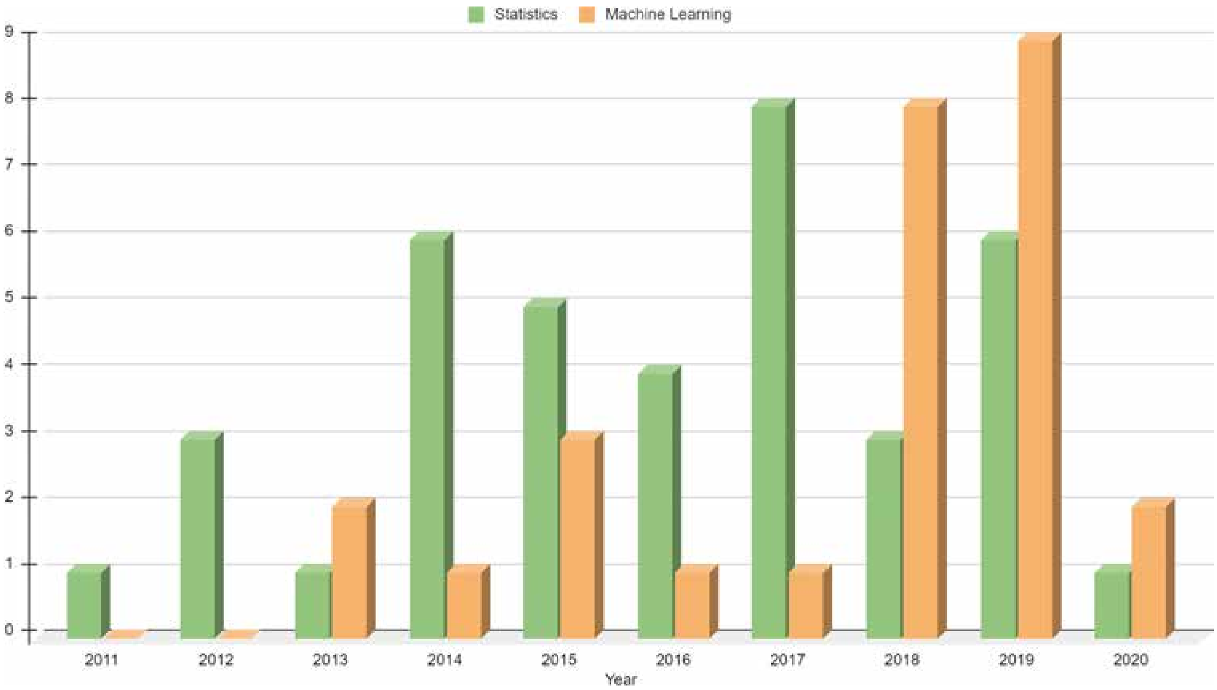


Figure 1: Adoption of statistical methods and machine learning approaches for solar energy generation forecasting comparison

superpixels. The computed evaluations were presented as the commonly used in semantic segmentation tasks, such as precision, recall, IoU for each class, and accuracy for each image. The authors achieved an average precision of 42.75%, average recall of 44.78%, average IoU 34.06% and an accuracy of 71.28%.

Overall, we can see a trend shift towards using machine learning approaches, from 2018 and further, as presented by Juncklaus Martins et al. [(2022]. Figure 1 shows that during the last decade there was approximately 2.7 articles being published every year using a machine learning approach, with a standard deviation of 3.19. Meanwhile, approximately 3.8 articles were published using classical statistics approaches for the same subject, with a standard deviation of 2.44. Even with the increase of machine learning methods, statistics methods are predominant overall and are still being used as a reliable way to predict solar energy.

The paper also presents a brief analysis of all related works, with an in-depth analysis of several metrics, including cloud identification and tracking, and evaluation metrics comparison.

3 Methodology

This section of the dissertation is organized to systematically introduce the comprehensive research approach taken in this study. It begins with a detailed account of the data acquisition process, describing the specific location and weather conditions under which the cloud imagery was captured, followed by the technical aspects of the Nimbus Gazer used in the dataset creation. The section then transitions into describing the structure of the dataset compiled for the experiments.

Subsequent subsections provide an overview of the various sets of experiments conducted, starting with the initial experiments, classified as Clouds-450 experiments, to set the stage for the study's broader aims. This is followed by a thorough exploration of the Clouds-1000 and Clouds-1500 experiments, each detailed in their respective subsections. These explore different segmentation techniques and deep learning models ranging from EfficientNet to advanced Transformer models.

The section culminates with a comparison of results derived from these experiments against existing literature, leading to a detailed discussion on the findings and their implications for the field of cloud classification. The methods conclude with a transition to the concluding remarks that synthesize the research insights and their relevance to future work in the domain.

In order to ease the understanding of this study, the items below summarize the overall methodology applied:

- Cloud images were captured using the Nimbus Gazer system.
- Three versions of the dataset were created for the experiments performed, with 450, 1000 and 1500 images, respectively.
- The main metrics used for validation were Intersection over Union (IoU), F1-score, Precision and Recall.
- Initially, 25 experiments were carried out to give direction to the research and help raise points for improvement for future experiments.
- The second battery of experiments was composed of a total of 8 experiments using more improved semantic segmentation techniques - which would potentially lead to a better result than the previous version of experiments; and 1 instance segmentation experiment to attempt to solve an identified problem in semantic segmentation models.

- The last battery of experiments consisted of 29 experiments, combining all the learning acquired in previous experiments and exhausting configurations of each experiment carried out.

Figure 2 showcases the overall workflow adopted. First sky images are captured with cameras angled slightly above the horizon, ensuring a frame rich in sky and devoid of terrestrial obstructions like trees or buildings. A subset of these images was then manually labeled. After accumulating sufficient labeled data, specialists reviewed and validated the annotations. Subsequent to this, came the training of the cloud detection models with this vetted data. The ensuing step involved evaluating the model's performance and concurrently using its output to further validate manual annotations. With a range of models trained, a comparative study of the results is performed.

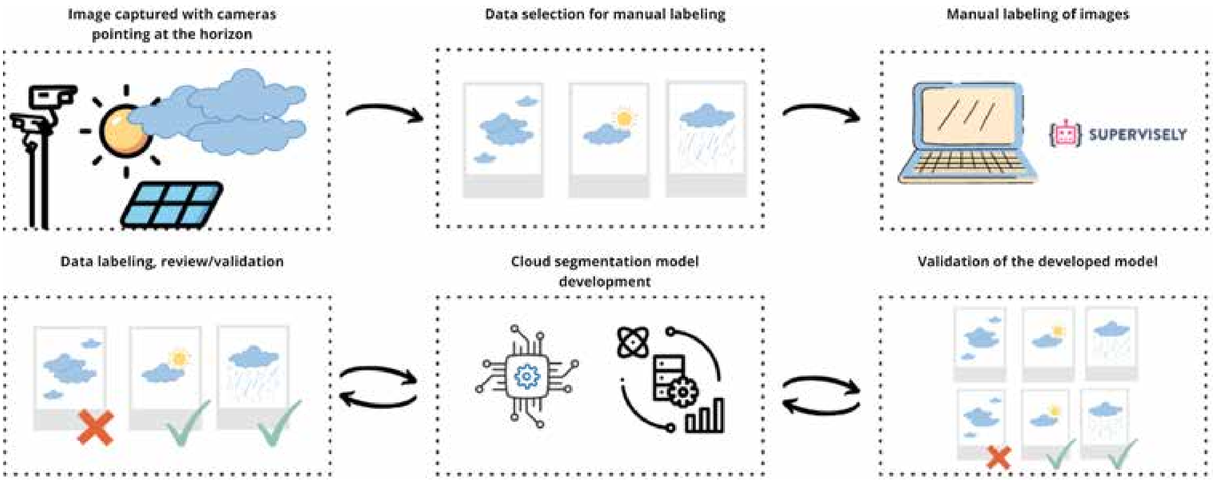


Image captured with cameras pointing at the horizon
Data selection for manual labeling
Manual labeling of images
SUPERVISELY
Data labeling, review/validation
Cloud segmentation model development
Validation of the developed model

Figure 2: Overall flow of work describing the steps used for all experiments

All experiments described in the subsequent sections were conducted using a variety of hardware setups. This diversity in hardware selection was primarily driven by the availability of resources at the time of each experiment. As a result, different models were trained on different hardware configurations.

It is important to note that the choice of hardware can have implications on the training efficiency and performance of the models.

3.1 Data

To construct the dataset, images were captured with cameras directed towards the horizon in the north and south directions, in city of Florianópolis, Brazil. In order to capture images from the sky, the Lapix research group developed a low-cost equipment using Raspberry Pi with a custom Operational System (OS) called Nimbus Gazer. Two equipment with cameras aiming just above the horizon line in order to capture the sky and incoming clouds are implemented. Both equipment are encased in a aluminum shell for protection against the weather, with a clear acrylic at the front where the camera is located. The cameras used are the Raspberry Pi camera model V1.3, with a field of view of 65°. Since there are two cameras aiming at opposite directions, there's the equivalent of a 130°view of the sky at all times. The equipment is connected via ethernet cable to the lab facility.

3.1.1 Location and Weather Conditions

The place chosen to capture the cloud images was a hill nearby Federal University of Santa Catarina's campus: an area with good view of the sky. The area is located in Santa Catarina Island, the insular part of Florianópolis, a city in the state Santa Catarina, located in South Region of Brazil, as seen in Figure 3. The location's geographic coordinates are 27°36'28.1"S 48°30'48.5"W.

According to de Meteorologia [1984], the climate of Santa Catarina island belongs to the fundamental type Cf and the specific variety Cfa in Köppen's classification Köppen and Geiger [1928], with humid mesothermal climate and well-distributed rainfall throughout the year. The average annual temperature is 20.8°C, with February being the hottest month, with a monthly average of 24.9°C and July being the coldest month, with an average of 16.4°C. The most frequent winds in the region, according to the description in Gaplan [1986], are from the northeast and north, but the southern winds has more repercussions, being the culprit of the sudden changes in temperature, also affecting clouds movement direction and speed. So, the strong southerly winds of the island come to be strong gusts, accompanied or not by rain, which can usually last for three days, bringing with it a variety of cloud types. During the autumn, these bursts become more common, therefore most photos present in this dataset were taken during this period (from 03/21 to 06/21). Based on this analysis, the research team decided that the best locations to point the cameras was north and south, where the most predominant winds are, so that clouds approaching the area could be seen more clearly.

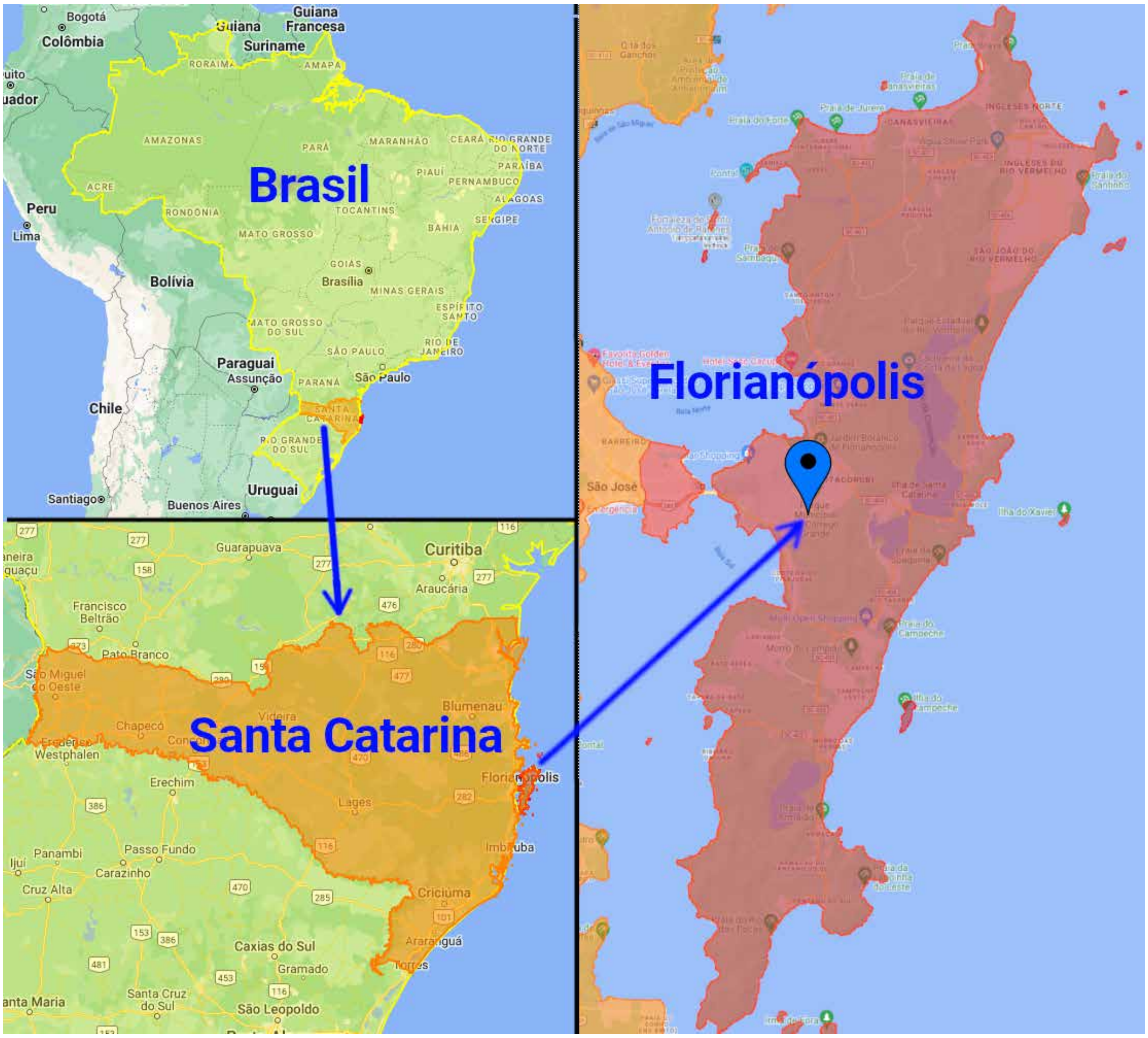


Figure 3: Location in the city of Florianópolis, Brazil were the images were captured

3.1.2 Nimbus Gazer

The Nimbus Gazer is an adaptation of the motionEyeOS[3], which is a Linux distribution that turns a single-board computer into a video surveillance system. The motionEyeOS OS is based on BuildRoot[4] and uses Motion[5] as a backend and motionEye[6] for the frontend. Figure 4 shows the equipment.

Nimbus Gazer uses motionEye version 0.41 and Motion version 4.2.2. The system is set to GMT time zone and prevents any camera LEDs from blinking, by disabling all LEDs in the boot configuration file. This configuration is set by default to prevent any LED reflection

[3] https://github.com/motioneye-project/motioneyeos/wiki

[4] http://buildroot.uclibc.org/

[5] https://motion-project.github.io/

[6] https://github.com/ccrisan/motioneye/

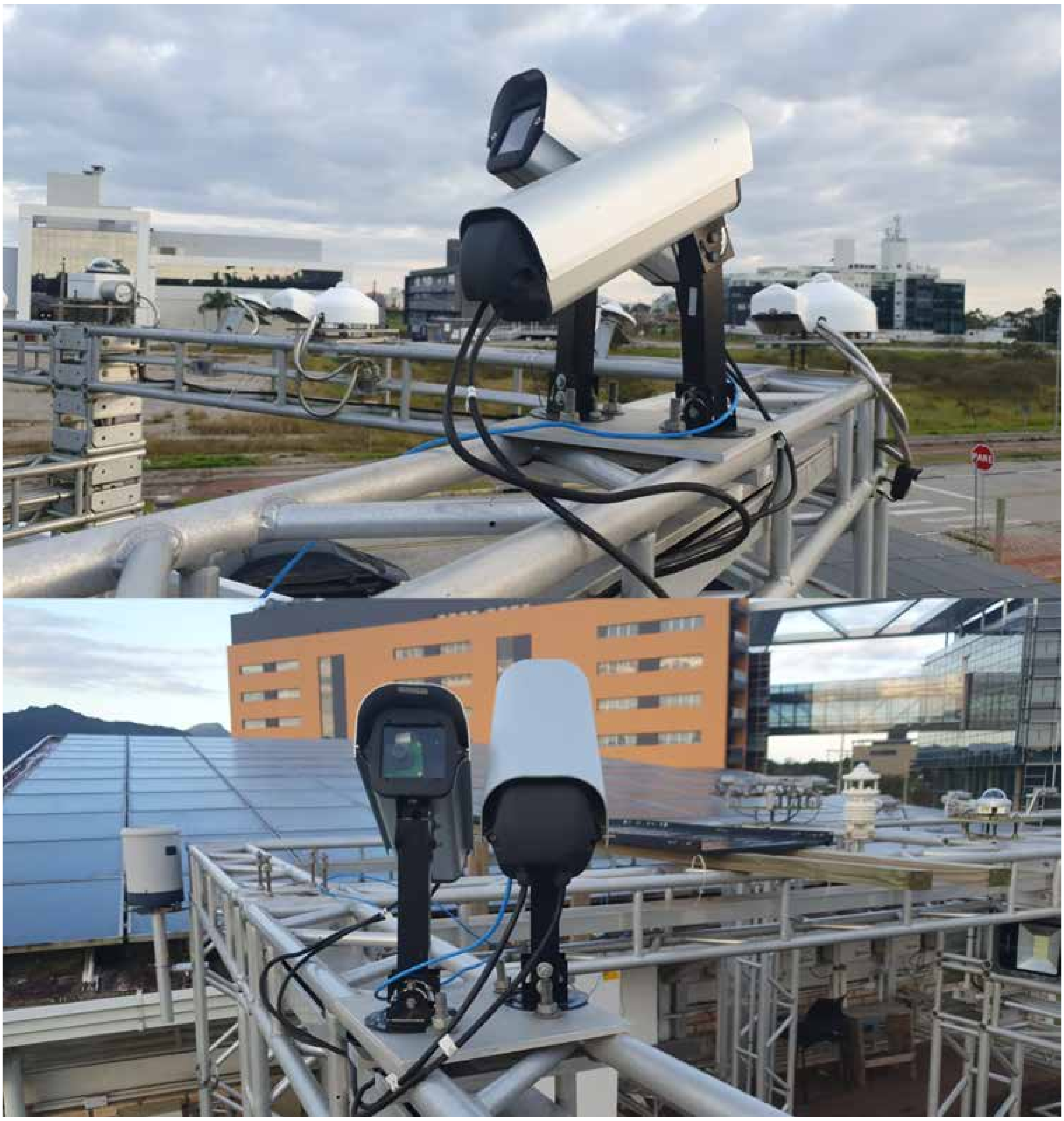


Figure 4: Low-cost equipment developed by the research team. In this picutre, the equipment is installed at the UFSC Photovoltaic Laboratory at the Federal University of Santa Catarina, in the city of Florianópolis, Brazil.

onto the camera lens. The system is set to start capturing images at 08:00 GMT and stop at 22:00 GMT. The chosen time interval was defined to capture only images with at least some level of sunlight. The time zone of the research lab is at GMT-3. To install the OS, it is necessary to have at least 32GB of free memory.

The configuration of the motion system is set at the lowest available frame rate of 1 frame per minute to match the time resolution of sensory data from the lab. That means that every minute, an image is captured.

Captured images are configured at 2592 x 1944 resolution and are stored in a local directory before being uploaded to the cloud. A built-in option was used to upload to a Google Drive directory to upload the images. All of the custom configurations can be seen in Appendix A.

For monitoring, the system sends a health-check e-mail each time the system either: boots-up, start or stop recording.

The monitoring scripts are located in the /root directory and can be customized by editing the the corresponding .sh files:

- boot-email.sh - sends an e-mail when the system is booted.
- start-recording.sh - sends an e-mail when the system starts recording.
- stop-recording.sh - sends an e-mail when the system stops recording.

For a step-by-step configuration of the system please you can access this repository[7].

Two examples of north and south images can be seen in Figure 5 below.

3.1.3 Datasets

Initially, clouds were classified into eight classes, according to the standards established by WMO. These classes include: Altocumulus, Cirrus, Stratocumulus, Cumulus, Stratus, Nimbostratus, Altostratus and Cirrocumulus. In addition to the cloud classes, an additional class was created to represent land features present in the images, such as trees and buildings, called ”Tree”.

However, after carrying out some experiments, details of which will be presented in the following sections, a decision was made to employ the clustering method recommended by Barrett and Grant [1976]. The process resulted in four distinct classes: Cirriform, Cumuliform, Stratiform and Stratocumuliform (Table 1).

It is worth mentioning that Barrett and Grant [1976] also suggests an additional classification called *Cumulonimbiform* which encompasses clouds. During the image acquisition period for the dataset, few cumulonimbus clouds were detected, and to keep the dataset more balanced, these clouds were grouped in the Cumuliform category.

In total we developed 3 versions of the dataset used for different experiments:

[7] https://github.com/bjuncklaus/Clouds-1000

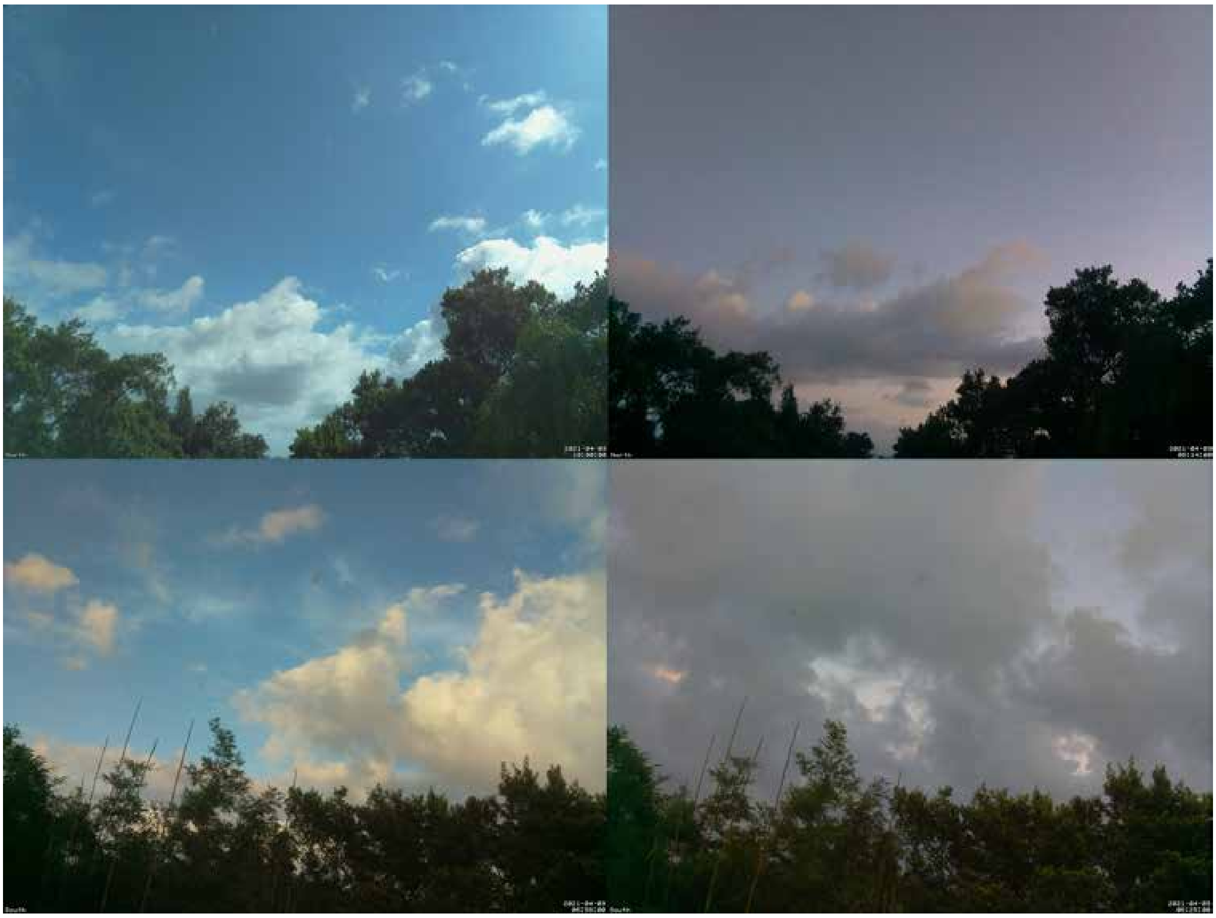


Figure 5: Example of north (top) and south (bottom) images captured by cameras pointing towards the horizon

- Version 1: 450 images with 8 cloud types and Tree class. March-April, 2021.
- Version 2: 1000 images with 5 cloud types and Tree class. March-June, 2021.
- Version 3: 1500 images with 5 cloud types and Tree class. March 2021-January 2022.

Thus, our research group decided to name each dataset version based on the amount of images present in it. Hence, we have the: Clouds-450, Clouds-1000 and Clouds-1500 dataset.

The annotations were handmade using the Supervisely tool. The tool was created

Table 1: Cloud grouping according to Barrett and Grant [1976]

Classes	**Cloud Type**
Stratiform	Cirrostratus, Altostratus, Stratus, Nimbostratus
Cirriform	Cirrus
Stratocumulus	Cirrocumulus, Altocumulus, Stratocumulus
Cumuliform	Cumulus, Cumulonimbus

for image annotation and data management in which it's possible to create the annotations via interface available, similar to other image editors. Each image was annotated with the polygon tool and classified using 4 cloud types: Cirriform, Cumuliform, Stratiform, Stratocumuliform and 1 class representing trees and buildings. This classification is based on solar radiation absorption characteristics. Due to the humid climate of the region, the Cumulonimbus (Cb) cloud seldom forms. This type of cloud usually form in dryer regions, thus there isn't any occurrence of this cloud in the dataset. The cloud type distribution is shown in Table 2.

Table 2: Distribution of version 2 of the dataset by cloud type

Cloud Type	**Amount in Dataset**	**% in Dataset**
Tree	989	99.30%
Stratocumuliform	812	81.53%
Stratiform	271	27.21%
Cirriform	285	28.61%
Cumuliform	90	9.04%

The manual annotation process with Supervisely is time consuming and error prone, mainly because of clouds shapes where, even to specialists, classification takes a long time to be precise and features like cloud height and density may not be clear, leading to doubts about its type. This problem with the clouds shape also slows down the specialists work of reviewing annotations, forcing the Data Revision process to be done by sampling, that is, not all images in the dataset were validated. Another problem is the mechanical process of annotating itself, which requires the annotator to manually wrap clouds in Supervisely using the mouse cursor, which can take a long time, especially when the image is complex, such as when clouds intersect each other, when they have holes or complex shapes.

3.2 Clouds-450 Experiments

For the first experiments a version of the dataset that consists of 450 images and uses all 8 cloud types as classes alongside the Tree class was used. This dataset was used only for the initial experiments that set the course correction and led to the development of this research. It is important to notice that 7 images had to be removed from this version of the dataset due to the lack of precise annoation. Figure 6 shows examples of annotated images of this dataset.

The distribution of this version of the dataset can be seen in Table 3. This table already excludes the 7 faulty images.

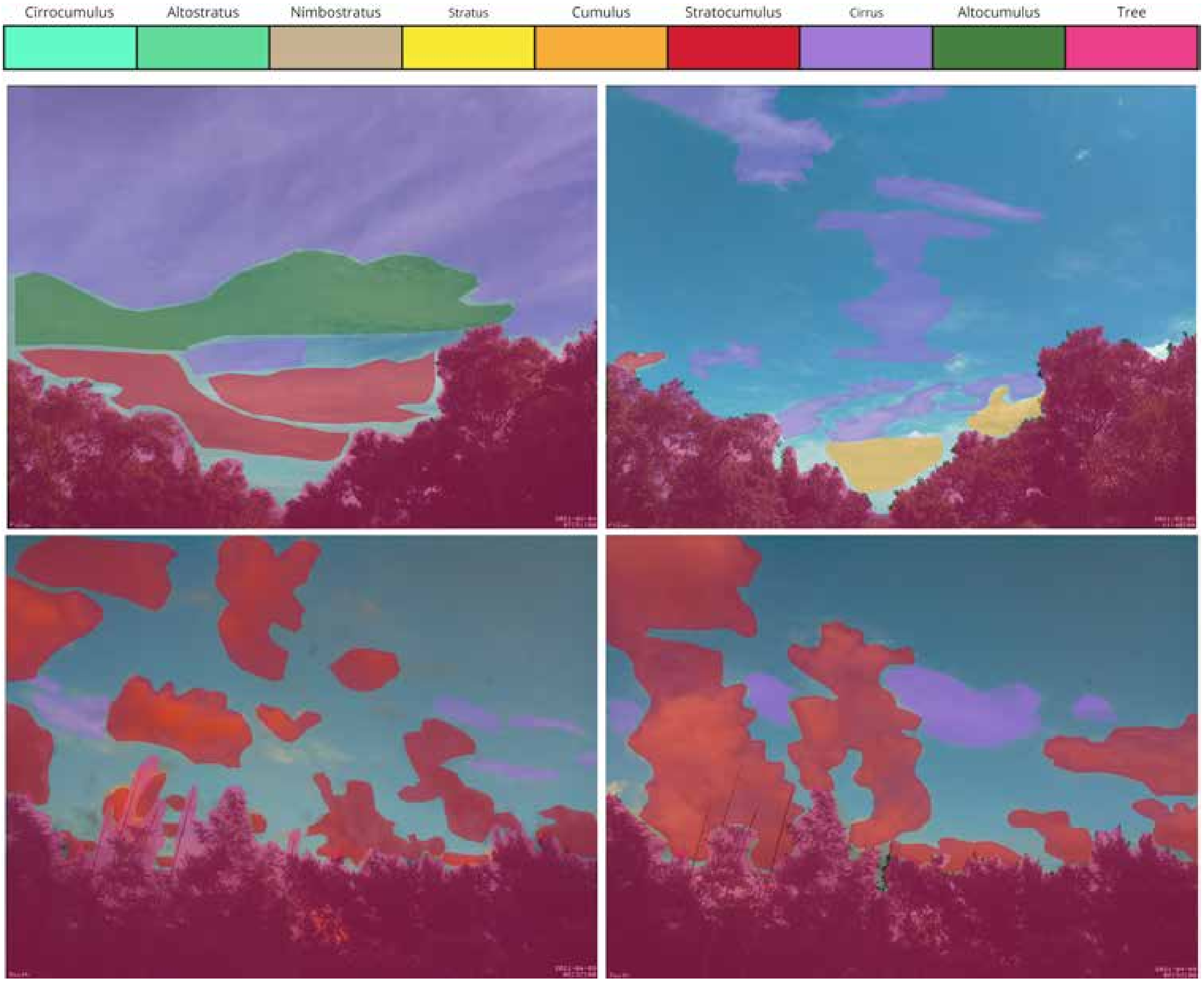


Figure 6: Example of annotated images of the first version created of the dataset used for early experiments

Table 3: Distribution of version 1 of the dataset by cloud type

Cloud Type	**Amount in Dataset**	**% in Dataset**
Tree	443	100%
Stratocumulus	311	71.82%
Cirrus	205	47.34%
Altocumulus	79	18.24%
Cumulus	66	15.24%
Stratus	59	13.63%
Nimbostratus	8	1.855%
Altostratus	2	0.46%
Cirrocumulus	0	0%

In order to determine the type of cloud present the Semantic Segmentation technique was used. This technique involves a neural network identifying individual pixels in an image according to an object class to which each pixel belongs, dividing the image into sections that each represent an object.

The models were constructed using the Segmentation Models library[8] version 1.0.1 with the TensorFlow 2.5.0 and Keras 2.4.3 ecosystem. TensorFlow is an end-to-end open-source platform for machine learning, developed by the Google Brain team, that provides a comprehensive and flexible ecosystem of tools, libraries, and community resources Abadi et al. [2016]. It allows for easy Model 28uilding and deployment. The library is a high level API with 4 models architectures for binary and multi-class image segmentation available out-of-the-box, 25 available backbones for each architecture in which all backbones have pre-trained weights for faster and better convergence, ready-to-use segmentation losses and metrics. TensorFlow offers multiple levels of abstraction, so users can choose the right one for their needs, from high-level Keras API, which simplifies model development and testing, to lower-level APIs for expert users requiring more control[9].

Keras, on the other hand, is an open-source software library providing a Python interface for artificial neural networks, acting as an interface for the TensorFlow library Chollet et al. [2015]. Initially supporting multiple backends, Keras now exclusively supports TensorFlow backend from version 2.4 onwards. It is designed to enable fast experimentation and prototyping through user-friendliness, modularity, and extensibility, contributing to its popularity in the deep learning community[10].

For all the experiments described in this section, three model architectures were used: the U-Net, Linknet and Feature Pyramid Network (FPN) model architecture.

The U-net is renowned for its efficacy in semantic segmentation tasks. U-net and CNN are somewhat similar. U-net networks are Deep Convolutional Neural Networks that were originally designed for segmentation of electron microscopy images Ronneberger et al. [(2015].

Linknet Chaurasia and Culurciello [2017] is modeled unlike others neural networks architectures, due to the fact that it links each encoder with the decoder, by doing this it prevents a lost of spacial information, which can be used for up-sampling operations. Since the decoder is sharing information by the decoder each level, there is no need for some previously used parameters, causing an overall more efficient network and real-time operations.

[8] https://github.com/qubvel/segmentation_models

[9] https://www.tensorflow.org

[10] https://keras.io/api/

Feature Pyramid Network Lin et al. [(2017] is a top-down architecture with lateral connections, created with the intent of building high-level semantic feature maps to any scale, achieving state-of-the-art performances. This method efficiently generates object segmentation proposals using an image-centric training strategy and many ideas from DeepMask/SharkMask to create their own FPN mask generation. Each level of the feature pyramid is used for predicting masks at different scales, respectively scales of 32, 64, 128, 256, 512 and other measures are mapped to the nearest scale.

3.2.1 Description

As initial experiments, 25 models were created. For the development and evaluation of these models, the initial version of the dataset, comprising 433 images, was strategically partitioned into separate sets for training, validation, and testing. The allocation was determined as follows: 70% of the images were used for training the model, allowing it to learn and adapt to the characteristics of the cloud classes. A subset of 10% was reserved for validation purposes, serving to fine-tune the model parameters and prevent overfitting during the training phase. The remaining 20% constituted the test set, which was utilized to assess the model's performance and generalization capabilities on unseen data.

As a training strategy, the Adam optimizer was used in conjunction with a learning rate reduction technique. The learning rate is reduced when the validation loss stops improving for 2 consecutive epochs and the reduction is done by Equation 3.2.1:

$$new\_lr = current\_lr * 0.2 \tag{3.2.1}$$

It was decided that all experiments would require resizing of the input images. This decision was made empirically, due to the time and memory requirements for training the models. Therefore, the ”Input Size” column in Table 3.2.1 represents the size of the resized input images (eg. 128 means 128x128 pixel images). This table contains all the performed experiments, in which the ”Batch Size” represents how many images were used in each batch of data for training. ”Backbone” is the transfer learning model used in the experiment, whereas ”Model” is the actual model architecture being trained. ”Epochs” represents how many epochs the model was trained on.

We used the Keras Adam optimizer and the sum of the Categorical Focal loss and

Dice loss function. The choice of loss function is due to the nature of the problem, which is a multi-label segmentation problem. Therefore, using these two functions together can better assess the model's performance instead of just one, thus leading to better convergence. Equation 3.2.2 shows the Categorical Focal loss calculation:

$$L(gt, pr) = -gt \cdot \alpha \cdot (1 - pr)^{\gamma} \cdot \log(pr) \tag{3.2.2}$$

Where α is a weighting factor (same as in balanced cross entropy) and γ is a focusing parameter for modulating factor (1 - p). With gt and pr being the ground-truth and predicted values of the image pixel.

The Dice loss function can be calculated by Equation 3.2.3:

$$L(precision, recall) = 1 - (1 + \beta^2) \frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{3.2.3}$$

Where β is a coefficient to balance precision and recall.

The composite Categorical Focal Dice loss function combines equations 3.2.2 and 3.2.3 to form a robust loss function that is sensitive to the class imbalance and the need for precise segmentation.

During training, specific callback functions provided by Keras were employed to enhance the training process. The ModelCheckpoint and ReduceLROnPlateau play a pivotal role in preserving the best model and optimizing the learning rate, respectively.

The ModelCheckpoint callback function is designed to save the model at specific intervals. For this experiment, the callback is configured to only save the model's weights when there is an improvement in the validation loss, which is a common practice to avoid overfitting and to ensure that the model can be restored to its most effective state post-training. The settings *save_weights_only=True* and *save_best_only=True* ensure that only the model's weights that yield the lowest validation loss are stored.

Table 3.2.1 summarizes all the experiments performed with this version of the dataset.

3.2.2 Results

The validation metrics used are the mean IoU and the average Dice metric (F-score or F1-score) for multi-class targets in segmentation. These metrics were selected as they are commonly used for semantic segmentation as they best represent both the successes and errors

Table 4: Summary of experiments configurations

Id	Input Size	Batch Size	Backbone	Epochs	Model
1	128	32	resnet18	250	Linknet
2	128	32	resnet34	250	Linknet
3	128	32	densenet169	250	FPN
4	128	32	resnet50	250	FPN
5	128	32	resnet18	250	FPN
6	128	32	densenet121	250	FPN
7	256	16	resnet18	250	Linknet
8	128	32	senet154	250	Unet
9	128	32	seresnext50	250	FPN
10	512	8	resnet34	200	Unet
11	512	8	seresnext50	200	Unet
12	512	8	efficientnetb0	200	Unet
13	512	8	efficientnetb1	200	Unet
14	512	8	efficientnetb4	200	Linknet
15	512	8	efficientnetb1	200	Linknet
16	1024	1	seresnext50	100	Linknet
17	1024	1	seresnext50	200	Unet
18	256	16	vgg16	250	FPN
19	256	16	vgg19	250	FPN
20	512	8	efficientnetb0	200	Linknet
21	512	8	resnet34	200	FPN
22	512	8	efficientnetb0	200	FPN
23	512	8	resnet18	200	Linknet
24	1024	1	resnet18	200	FPN
25	1024	4	resnet18	200	Unet

of models.

The IoU, also known as the Jaccard index or Jaccard similarity coefficient (originally coined coefficient de communauté by Paul Jaccard), is a statistic used for comparing the similarity and diversity of sample sets. The Jaccard coefficient measures the similarity between finite sample sets, and is defined as the size of the intersection divided by the size of the union of the sample sets, as seen in Equation 3.2.4.

$$J(A,B) = \frac{A \cap B}{A \cup B} \tag{3.2.4}$$

The F-score, also known as the Dice coefficient, is similar to the dice loss and can be interpreted as a weighted average of precision and recall, where an F-score reaches its best value at 1 and worst score at 0. The relative contributions of precision and recall to the F1-score are equal. Equation 3.2.5 shows the F-score formula:

$$F_\beta(precision, recall) = (1+\beta^2)\frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{3.2.5}$$

Where β is a coefficient to balance precision and recall.

Table 3.2.2 presents the results obtained with this batch of initial experiments with the developed models.

It's clear from the table that models with a 128x128 pixel input size did better than those with bigger sizes. For example, the first two experiments with the Linknet model and a 128x128 input size got higher Mean IoU and Mean F1 Scores compared to others. The FPN model with the same input size also showed good results in experiments 3 to 6.

In Figure 7, we can see the Linknet best model's (experiment 1) ability to segment and classify clouds. While this model maintains the overall structure and distribution of cloud types, subtle inaccuracies can be observed. The model appears to struggle with differentiating between closely related cloud types, as evident in the areas where Altocumulus and Stratocumulus clouds are present. This is possibly due to their textural and color gradient similarities. This issue is less pronounced than any other model as will be presented next, suggesting that while the Linknet model has improved performance, it still faces challenges with complex cloud patterns and edge definitions.

The segmentation and classification results from the best FPN model (experiment 3) are depicted in Figure 8. The image presents a sky scene with varied cloud formations and

Table 5: Summary of results for the Clouds-450 experiments

Id	Input Size	Model	Loss	Mean IoU	Mean F1
1	128	Linknet	0.81292	0.73839	0.76193
2	128	Linknet	0.8095	0.73142	0.75492
3	128	FPN	0.80124	0.72725	0.75473
4	128	FPN	0.80623	0.72452	0.75161
5	128	FPN	0.81297	0.71854	0.74528
6	128	FPN	0.8049	0.71444	0.74161
7	256	Linknet	0.80289	0.70637	0.73162
8	128	Unet	0.78772	0.69827	0.72518
9	128	FPN	0.7882	0.68258	0.71105
10	512	Unet	0.7896	0.67929	0.70693
11	512	Unet	0.77779	0.66695	0.69519
12	512	Unet	0.78552	0.66512	0.69398
13	512	Unet	0.77683	0.65735	0.68708
14	512	Linknet	0.77741	0.61545	0.64549
15	512	Linknet	0.78173	0.61246	0.64179
16	1024	Linknet	0.81084	0.60858	0.63365
17	1024	Unet	0.80807	0.60798	0.63557
18	256	FPN	0.77944	0.60423	0.63368
19	256	FPN	0.78139	0.59913	0.62852
20	512	Linknet	0.78488	0.59905	0.62845
21	512	FPN	0.78563	0.59752	0.62706
22	512	FPN	0.78009	0.58874	0.6187
23	512	Linknet	0.78878	0.58128	0.60976
24	1024	FPN	0.71126	0.54989	0.57785
25	1024	Unet	0.7941	0.54038	0.57024

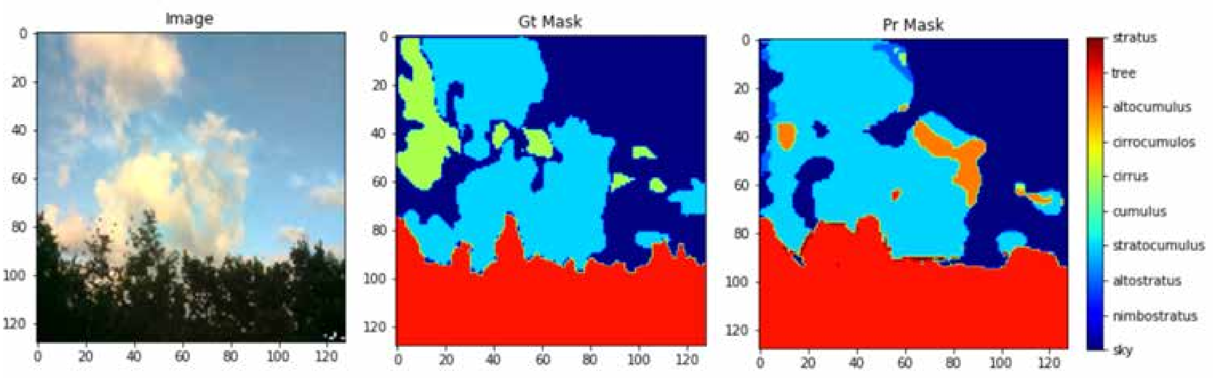


Figure 7: Segmentation performance of the best Linknet model from the Clouds-450 experiments

the treeline at the bottom, a typical setup for testing cloud segmentation models. In this instance, the model correctly identifies the majority of the cloud classes, including the cumulus and altocumulus clouds which are central in the image. However, we can discern a noticeable challenge in differentiating between Stratocumulus and Altocumulus clouds. This suggests that while the FPN model has achieved a high level of understanding of the cloud forms, there remains an area of ambiguity in capturing the transitional zones between certain cloud types. This could indicate a need for further model refinement or additional training data that adequately represents the variability and subtlety of cloud transitions.

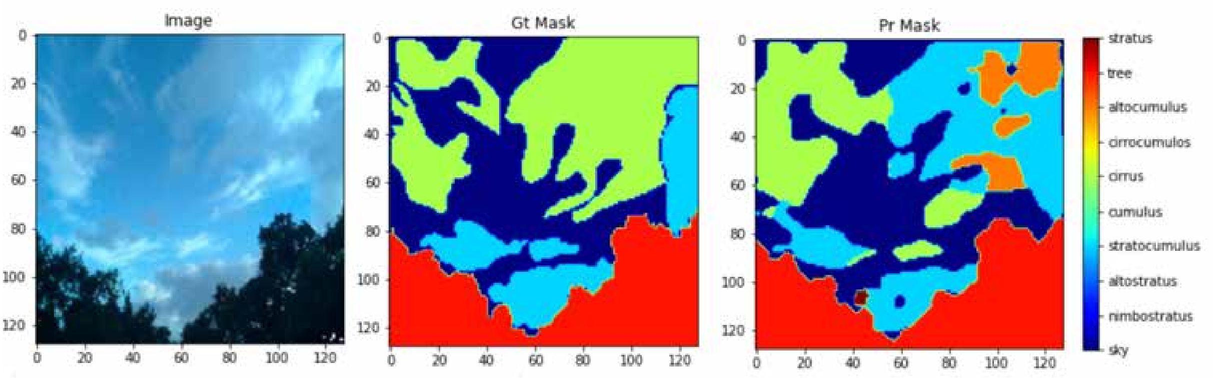


Figure 8: Segmentation results of the best FPN model from the Clouds-450 experiments

We can gather some insights into the U-net best model's (experiment 8) performance by looking at Figure 9. The original sky image showcases a cumulus cloud, which is typically characterized by its dense, fluffy appearance and distinct edges. The ground truth (Gt) mask accurately delineates the cloud's boundaries and classifies it alongside other cloud types and terrestrial features. However, when we examine the predicted (Pr) mask from the best U-net model, we observe certain discrepancies.

The U-net model, while adept at general segmentation tasks, exhibits some challenges in precisely capturing the complex boundaries and subtle gradations of cloud density. The model's making more errors between altocumulus and stratocumulus cloud types. Such misclassification could stem from the model's training on this limited version of the dataset, where the distinguishing features between these cloud types may not have been adequately captured. Additionally, the model's interpretation of texture and color gradients within the cloud formations might have contributed to this error.

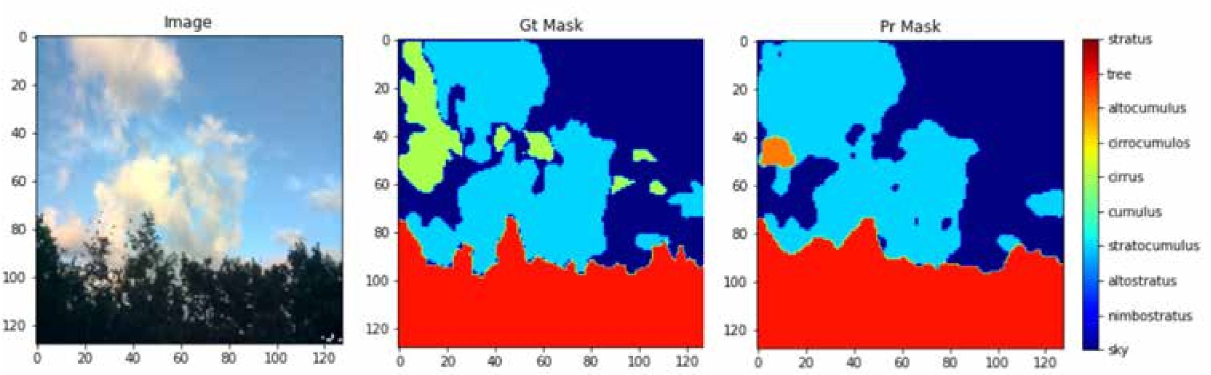


Figure 9: Comparative visualization of cloud segmentation using the best U-net model from the Clouds-450 experiments

When we increased the input size to 512x512 and 1024x1024 pixels, the performance went down. This is especially true for experiments 14 to 25, where both the Mean IoU and Mean F1 scores are lower. The lowest scores were in experiments with a 1024x1024 input size, like experiment 16 with the Linknet model and experiment 17 with the Unet model.

3.2.3 Discussion

Looking at our results, it's clear that models with 128x128 input sizes do better than those with larger sizes. Smaller images have fewer pixels, which makes them simpler and less likely to cause mistakes in identifying different parts of the image. This is an issue and it needs addressing on the next batches of experiments. Hence, the resizing of images will only be applied onto the input images.

On the other hand, models with 1024x1024 input sizes didn't do as well. The FPN model had the lowest loss rate at 0.71126, but there's an issue with the ”Tree” class affecting the overall results. Because this class is in every image and doesn't change much, it makes the models seem like they are doing better than they really are. This is a problem, especially for accurately identifying clouds.

We also found problems with all of our models are predicting more than one class for the same part of a cloud. This, combined with an unbalanced dataset and misleading results from the Tree class, shows how complex and challenging it is to segment images accurately.

3.2.4 Next Steps

Given these challenges, we planned several steps to improve our image segmentation models for the next experiments. We used more advanced model designs that are better at handling detailed images and also tried new ways to make our training data more varied to help the models learn to handle different scenarios with different cloud types.

Another important step is to fix the unbalanced dataset. We need to check how well each class is being predicted and use that information to make our models better. This also involves simplifying the task by reducing the number of classes our models have to predict.

These steps are aimed at making our models more accurate and reliable, balancing the need to fit our current data while also being able to work well with new, unseen data.

3.3 Clouds-1000 Experiments

For these experiments, two categories of segmentation models were utilized: Semantic and Instance Segmentation. The inclusion of the latter was made after performing a qualitative evaluation over the results obtained from the semantic segmentation experiments. We identified a problem where distinct regions of the same cloud are erroneously classified as different classes, we called this the ”localization” problems. This problem can’t be easily distinguished through the validation metrics and in order to address it, this additional technique of Instance Segmentation was included in the experiments suite. Our hypothesis is that this technique, alongside the library used to train and evaluate the model, would help mitigate the localization problem.

Following the overall flow described at the beginning of the section in Figure 2, an initial evaluation was carried and led to the creation of the Clouds-1000 dataset Juncklaus Martins et al. [(2022]. This dataset consists of of 1000 sky following the same strategy described in the 3.1 section. The images were collected every minute over the period of March–June of 2021, and the cloud annotation task was divided between 3 Data Analysts, responsible for analyzing and labeling the images in the Data Annotation step, and 2 meteorologists, responsible for supervision and validation in the Data Revision step. For this version the clustering method recommended by Barrett and Grant [1976], described in the 3.1.3, was followed.

This version of the dataset had faced several validations and during an inspection 4 images were either partially annotated or missing annotation entirely. Therefore, the experiments using this dataset actually use 996 fully hand-annotated images. Examples of annotated images can be seen in Figure 10.

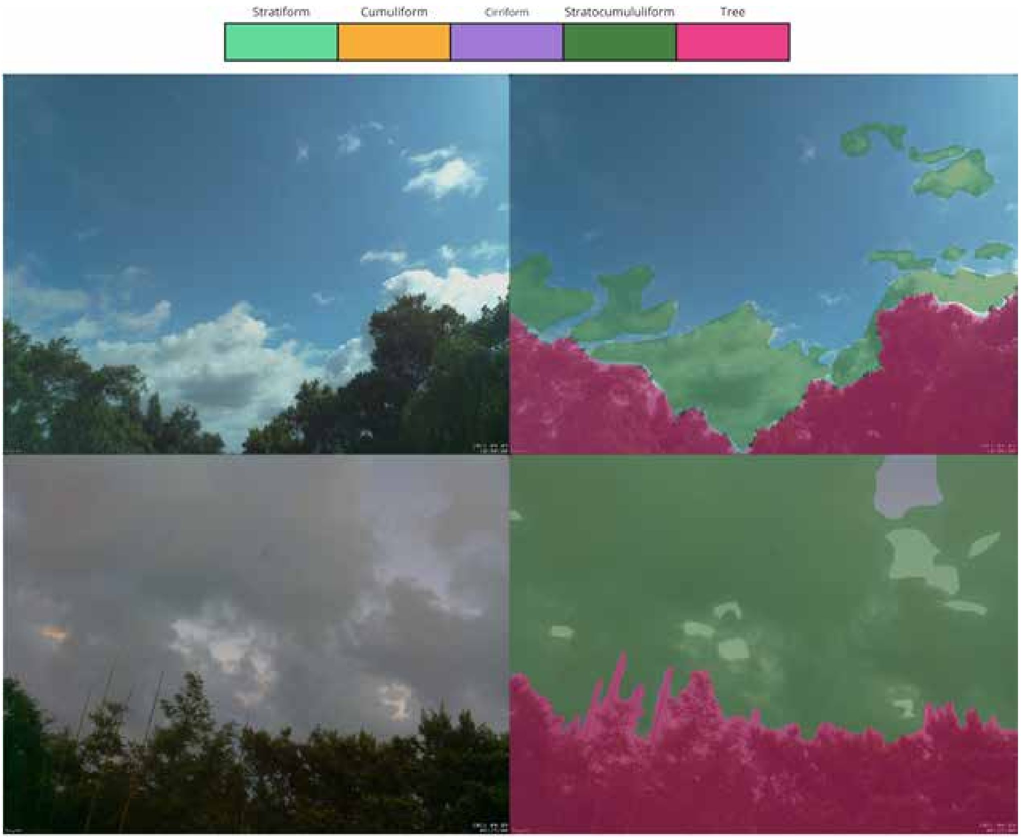


Figure 10: Example of Clouds-1000 dataset images (left) compared with its annotated version (right)

The annotation tool Supervisely, uses a proprietary format called Json-based Supervisely Annotation Format to represent labels on images, a format that makes it hard to use that labels elsewhere, except on the Supervisely platform itself. Therefore, it was necessary to convert the annotations to COCO[11] (Common Objects in Context) format in order to use it. COCO is a large-scale object detection and segmentation dataset including evaluation techniques for instance segmentation models. Annotations examples for the same image in Supervisely JSON format and in COCO format can be seen in Appendix B and C, respectively.

The High-Resolution Network (HRNet) Wang et al. [2020] is particularly noteworthy within the semantic segmentation models for its unique architecture. Unlike traditional segmen-

[11] https://cocodataset.org/

tation networks that downsample the image to a low resolution and then gradually recover the spatial details, HRNet maintains high-resolution representations through the entire process. It starts with a high-resolution subnetwork, then progressively adds lower-resolution subnetworks in parallel while exchanging the cross-resolution information through repeated multi-scale fusions.

The mathematical formulation of HRNet's fusion strategy can be expressed as follows:

$$\mathbf{X}_{t+1}^{i} = \mathscr{F}(\mathbf{X}_{t}^{i}, \mathbf{X}_{t}^{i+1}, \ldots, \mathbf{X}_{t}^{N}) \tag{3.3.1}$$

where $\mathbf{X}_{t}^{i}$ represents the feature maps at the i-th resolution at stage t, and $\mathscr{F}$ denotes the series of fusions across the resolutions. This enables HRNet to simultaneously capture rich contextual information and precise spatial details, which is paramount for segmentation tasks.

Figure 11 showcases an example of the the HRNet architecture. The image illustrates the interconnected high-to-low resolution convolutions that enable the network to preserve high-resolution feature maps throughout the depth of the network, a stark contrast to the typical encoder-decoder structures.

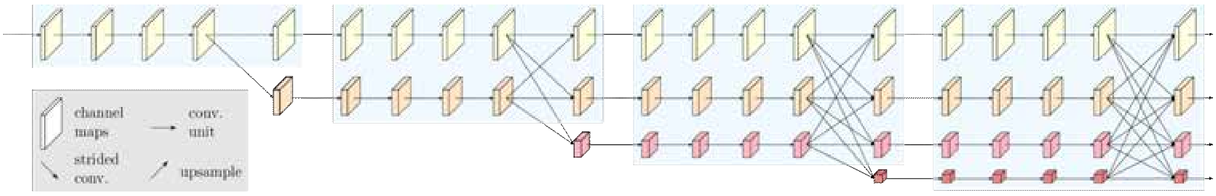


Figure 11: Original HRNet architecture example as demonstrated by Wang et al. [2020]. The example illustrates the simultaneous multi-resolution convolutions and the fusion of feature maps across different resolutions.

The other two semantic segmentation architectures used were the previously explained U-net with Resnet and the Efficientnet. EfficientNet Tan and Le [(2019] is distinguished by its balance of efficiency in computation and model size. This network utilize a systematic approach called Compound Scaling, which involves scaling the depth, width, and resolution of the network in a principled way. The Compound Scaling method is governed by a compound coefficient, ϕ, which is used to proportionally scale the network width (w), depth (d), and resolution (r) following the equations:

$$\text{depth: } d = \alpha^{\phi} \tag{3.3.2}$$

$$\text{width: } w = \beta^{\phi} \tag{3.3.3}$$

$$\text{resolution: } r = \gamma^{\phi} \tag{3.3.4}$$

where α, β, and γ are constants that determine how each of these dimensions is increased for a given value of ϕ. This method allows EfficientNets to achieve higher accuracy with fewer parameters compared to models scaled by traditional methods.

Figure 12 presents the EfficientNet architecture, including an example of the baseline network (a) and the effects of scaling network width (b), depth (c), and resolution (d) independently. Panel (e) illustrates the compound scaling method, which scales all three dimensions uniformly with a fixed ratio, as proposed in the original EfficientNet design. EfficientNets' architecture is trained over the ImageNet dataset Tan and Le [(2019] and is adept at various tasks including image classification, object detection, and, crucially for the purposes of this work, semantic segmentation.

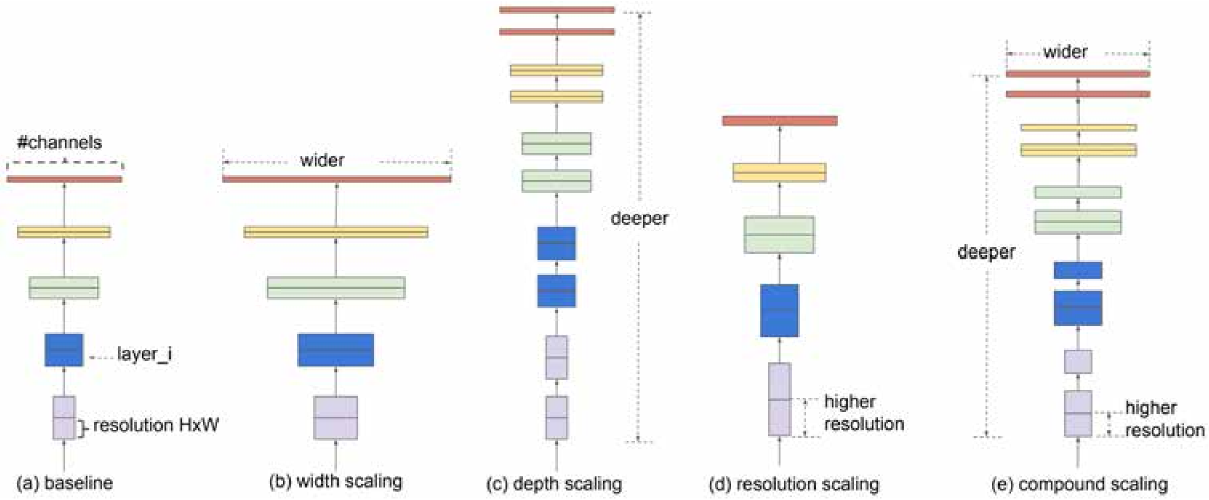


Figure 12: Original EfficientNet architecture example as demonstrated by Tan and Le [(2019]. The example illustrates the model scaling capabilities of the proposed architecture where (a) is a baseline network example; (b)-(d) are conventional scaling methods that only increase one dimension of network width, depth, or resolution. (e) is the proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

For the instance segmentation experiment the Detectron2 library[12], an open-source machine learning library developed by Facebook AI Research, which offers cutting-edge algorithms for detection and segmentation tasks. Detectron2, the successor to Detectron and maskrcnn-benchmarkWu et al. [(2019], was chosen for two main reasons. Firstly, it excels in robust object detection, making it well-suited for scenarios where objects, such as different types of clouds, overlap or are closely situated. Its advanced object detection capabilities enable

[12] https://github.com/facebookresearch/detectron2

more accurate identification and classification of objects within images, thereby reducing the occurrence of a cloud being assigned multiple classes. Secondly, Detectron2 boasts impressive segmentation capabilities, including state-of-the-art algorithms like Mask R-CNN.

A concern was raised during the planning of these experiments regarding using different frameworks, might not result in a fair comparison. However, all hyperparameters were tried to approach the same values accross all frameworks in order to reduce any bias towards a specific framework. To better visualize the processes used in the experiments, from labeling to validation, refer to Figure 13 which provides a detailed flow.

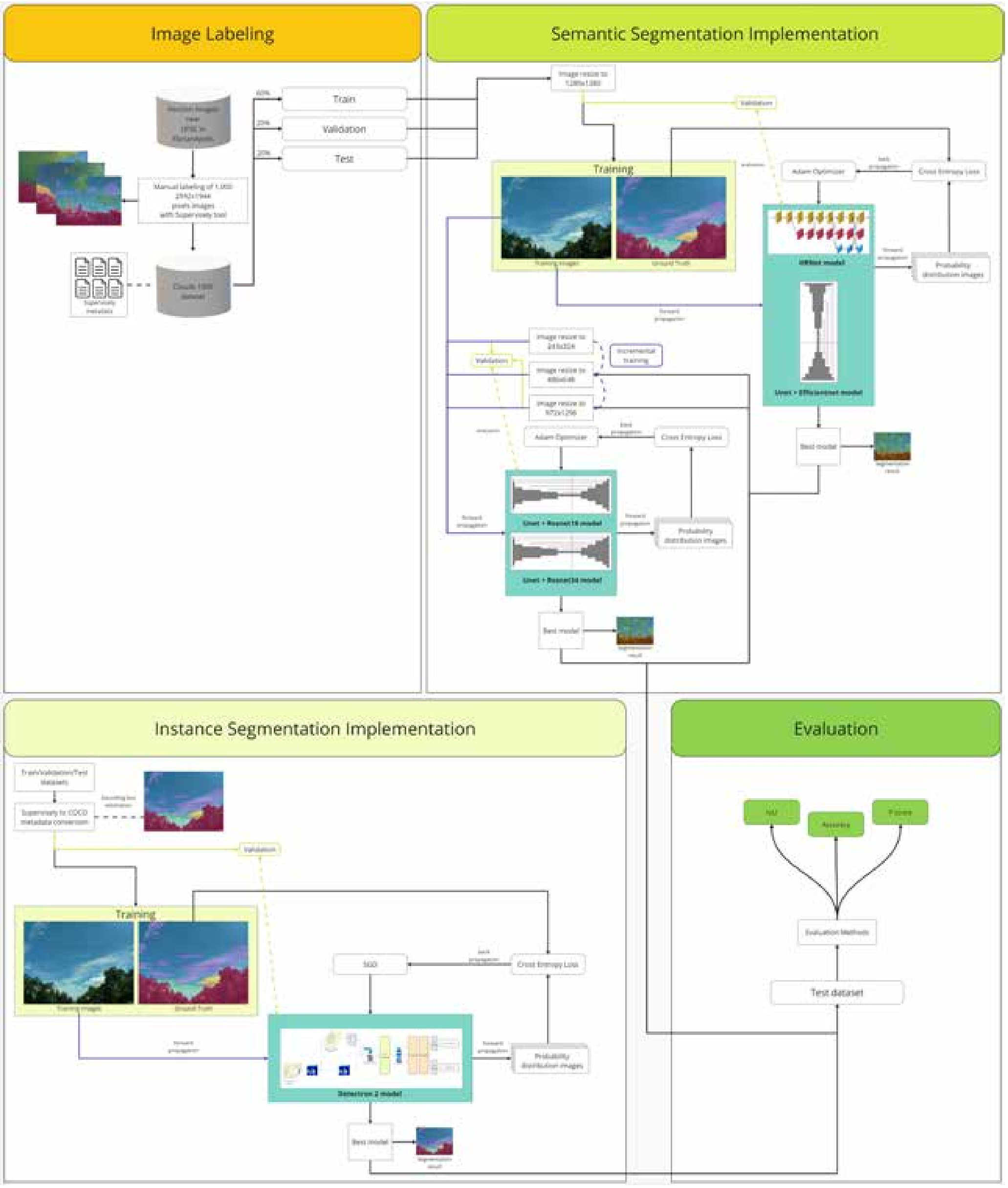


Figure 13: Step-by-step process from data labeling up to results and validation

The ground-truth labels used in the experiments follow the segmentation based on solar radiation absorption characteristics. The images were divided into three datasets: training, validation and testing. Where 60% of the data was used for training, 20% for validation and testing, respectively. The selection strategy was done randomly, without substitution and the sampling of validation data was done only over the training dataset. In order to validate each experiment the mean IoU, accuracy and F-score were used.

In total we performed 8 semantic segmentation and 1 instance segmentation experiment. All experiments were performed on a Tesla p100-pcie-16gb gpu and followed the same division and sampling criteria described above. The data used in the experiments are all from the Clouds-1000 dataset. The division process was executed only once, therefore the training, validation, and test sets are equal across all experiments.

3.3.1 Description

For the HRNet models, the PaddleSeg[13] framework was used, which is an end-to-end highly-efficient development toolkit for image segmentation based on PaddlePaddle, and helps both developers and researchers in the whole process of designing segmentation models, training models, optimizing performance and inference speed, and deploying models. This framework was chosen because it could train the HRNet architecture faster with less vRAM requirements than the original code.

Both the Adam optimizer and the Polynomial Decay training policy were used and the models were trained using the standard transfer-learning/fine-tuning workflow. The network was also fed with images with 1280x1280 resolution and trained for 80,000 iterations with a batch size of 2 images in order to compare its results with previously trained models. For this experiment a A100-SXM4-40GB video card was used.

The U-net with Resnet models were trained using the FastAI v2 framework[14]. Two models were trained with different Resnet architectures using transfer learning, with 18 and 34 residual layers, and employed the same incremental resolution training strategy described in section 3.2 with different parameters, in which the model is trained with a specific Resnet architecture with different resolutions for a number of epochs. The number of epochs was determined empirically, based on the described previous experiments and the available hardware for training. For every resolution, the learning rate finder technique was used, which consists of plotting the learning rate vs loss relationship for a model. The idea is to reduce the amount

[13] https://github.com/PaddlePaddle/PaddleSeg

[14] https://www.fast.ai/

of guesswork in picking a good starting learning rate. The F-score metric was monitored for validation during the training step. This method was applied for both experiments with Resnet18 and Resnet34.

The U-net with EfficientNet models used the mobile-size baseline network, named EfficientNet-B0. This pre-trained model was used for transfer learning due to hardware limitations and to prevent overfitting, since more complex models need more data. The only transformation applied to the input images was a change in the original 2592x1944 resolution to 1280x1280 due to vram limitations. The model was trained using the Pytorch framework over 13 epochs with a learning rate of $1x10^{-4}$ and a batch size of 2 images while monitoring the Cross Entropy loss function.

For better visualization and understanding, Table 6 encapsulates all the hyperparameters tailored for each of the semantic segmentation model of the performed experiments, where *LR* stands for Learning Rate and the value *LR Finder* means that used the learning rate finder technique describe in Smith [2017].

Table 6: Hyperparameters for all semantic segmentation models developed

Id	Model	Resolution	Epochs	LR	Batch	Optimizer	Loss
26	HRNet	1280x1280	80,000	1×10^{-4}	2	Adam	Cross Entropy
27	Unet + Resnet 18	243x324	45	LR Finder	12	Adam	Cross Entropy
28	Unet + Resnet 18	486x648	45	LR Finder	4	Adam	Cross Entropy
29	Unet + Resnet 18	972x1296	55	LR Finder	1	Adam	Cross Entropy
30	Unet + Resnet 34	243x324	45	LR Finder	12	Adam	Cross Entropy
31	Unet + Resnet 34	486x648	45	LR Finder	4	Adam	Cross Entropy
32	Unet + Resnet 34	972x1296	55	LR Finder	1	Adam	Cross Entropy
33	Unet + Efficientnet	1280x1280	13	1×10^{-4}	2	Adam	Cross Entropy
34	Detectron2	2592x1944	3,000	25×10^{-5}	8	SGD	Cross Entropy

In order to use the Detectron2 library, the dataset was converted from the Supervisely json format to the COCO format. The conversion process involves extracting image-level and object-level information from Supervisely annotations and reformatting it into the COCO standard. The conversion is performed through the following steps in Algorithm 1.

The algorithm converts annotations from the Supervisely format to the COCO format. For each annotation file, image-level and object-level information is extracted. These details are transformed and collected into a new COCO object, which is added to a set of COCO annotations. The process repeats for each object in all annotation files. Finally, the complete set

Algorithm 1 Supervisely to COCO Conversion

```
**Input:** $F$ {AnnotationFiles}
**Output:** $C$ {COCOAnnotations}
for each $f$ in $F$ do
    $I \leftarrow GetImgInfo(f)$
    $O \leftarrow GetObjects(f)$
    for each $o$ in $O$ do
        $c \leftarrow \emptyset$
        $c.bbox \leftarrow ToCOCOBbox(o.bbox)$
        $c.segmentation \leftarrow ToCOCOSeg(o.segmentation)$
        $c.category \leftarrow CatToID(o.category)$
        $c.is\_crowd \leftarrow 0$
        $C \leftarrow C \cup \{c\}$
    end for
end for
$SaveToCOCO(C)$
```

of COCO annotations is saved for future use. The conversion process ensures that all relevant image-level and object-level information is accurately preserved in the resulting COCO annotations.

After the dataset preparation, the model to predict the bounding boxes and segmentation pixels for the objects is trained. Firstly, a baseline model previously trained with Detectron2 called Mask RCNN R 50 FPN model is initiated in order to have better tradeoffs between speed and accuracy Wu et al. [(2019]. The model's training parameters have a batch size of 8, a learning rate of 25×10^{-5}, and a stochastic gradient descent optimizer. The original resolution of the input images was kept and the model was trained for 3,000 iterations on the available Google Colab[15] GPU, taking approximately three and half hours to train.

3.3.2 Results

Table 7 summarizes the overall performance of the best models tested. Each model is identified with an unique *Id* for later reference. Model 28, which is the combination of U-net and Resnet18, achieved the highest mIoU of 0.6, an accuracy of 0.8564 and F-score of 0.7234, indicating its overall strong performance and generalization across the entire dataset.

The HRNet model achieved a mIoU of 0.3889, an accuracy of 0.7316, and an F-score of 0.4869 over the test dataset. These results were obtained after training the network for 63,300 epochs.

[15] https://colab.research.google.com/

Table 7: Average results of the best models over the test dataset

Id	Model	Input Size	mIoU	Accuracy	F-score
26	HRNet	1280x1280	0.3889	0.7316	0.4869
28	**Unet + Resnet18**	**486x648**	**0.6**	**0.8564**	**0.7234**
32	Unet + Resnet34	972x1296	0.4796	0.7967	0.59
33	Unet + EfficientNet	1280x1280	0.4187	0.8141	0.4871

The U-net with Resnet experiments had different results with different resolutions, as expected. However, is possible to verify in Table 8 that the model that achieved the best quantitative metrics is the second simplest model is Model 28, composed of a Resnet18 with 486x648 resolution. This model achieved an average IoU of 0.6 across the entire test dataset. In contrast, the Resnet model with 18 residual layers and 972x1296 resolution presented only a slight improvement over the model using 243x324 resolution.

Table 8: Quantitative results of the best U-net models for each architecture and resolution

Resnet Size	Input Size	Accuracy	F-Score	mIoU
18	243x324	0.27	0.12	0.07
18	**486x648**	**0.85**	**0.72**	**0.6**
18	972x1296	0.48	0.23	0.17
34	243x324	0.33	0.13	0.09
34	486x648	0.17	0.09	0.05
34	972x1296	0.79	0.59	0.47

A comparison of quantitative results by semantic segmentation model is presented in Table 9. This table shows the results of the best models over the test dataset for each class. Model 28 outperformed the other models once again, in most of the classes, achieving the highest mIoU and precision for the Tree and Background classes, as well as the highest precision for the Stratocumuliform and Cirriform classes. Model 28 also achieved the highest recall for the Tree and Background classes, and the highest recall for the Stratiform and Stratocumuliform classes.

Model 3, which consisted of an U-net architecture combined with an EfficientNet backbone, achieved a mIoU of 0.4187, an accuracy of 0.8141, and F-score of 0.4871 over the

Table 9: Results of the best semantic segmentation models over the test dataset, by class

Metric	**mIoU**				**Precision**				**Recall**			
Target Class / Id	26	28	32	33	26	28	32	33	26	28	32	33
Background	0.59	**0.76**	0.69	0.72	0.75	**0.85**	0.81	0.79	0.73	**0.88**	0.82	**0.88**
Tree	0.9	**0.94**	0.9	0.92	0.91	**0.97**	0.92	0.94	**0.98**	0.96	0.97	0.97
Stratocumuliform	0.55	**0.75**	0.65	0.69	0.67	**0.86**	0.78	0.75	0.76	0.86	0.80	**0.89**
Stratiform	0.07	**0.38**	0.18	0.03	0.18	**0.56**	0.35	0.71	0.10	**0.5**	0.28	0.03
Cirriform	0.1	**0.46**	0.35	0.13	0.45	**0.69**	0.6	0.49	0.12	**0.58**	0.46	0.15
Cumuliform	0.09	**0.29**	0.07	0	0.13	**0.39**	0.13	0	0.25	**0.53**	0.12	0

test dataset at the last epoch which was trained on. However, this model also presents problems with segmenting certain classes.

Figure 14 presents the predicted segmentation of each model on the same input image. The Stratiform class is predominant overall with a small area of Cirriform clouds on another layer, behind the main clouds. No model was able to identify the latter, with only model 32 inferring classes Cumuliform and Stratiformes, however none are present in the input image. It's also possible to observe that models a, b and d make very similar predictions, however, looking closely is possible to see that model 26 makes a more refined prediction, at the pixel level. Model 28 and d are more similar to the ground truth mask, this can be one of the main reasons that Model 28 outperforms the other models.

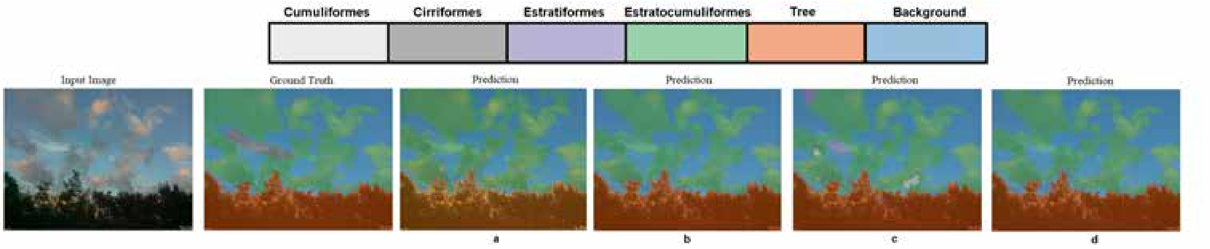


Figure 14: Predicted cloud segmentation of different models on the same input image, showing the predominance of Stratiform class and the differences in segmentation performance between models

However, that's not always the case. In Figure 15 the model 26 makes wrong predictions, resulting in a much rougher inference, especially over the clouds of class Cumuliform. Models b and c have small patches of this class inside the predicted Stratocumuliform class, which is not correct. This shows a ”localization” problem, in which the model classifies different regions of the same object (cloud) as multiple classes, where the ground truth is actually only one object. The model is not able to discern that there are two main cloud objects of the same class. The models are probably being influenced more by texture and shape than other characteristics. The Figure shows that this problem occurs with small clouds as well, the Stratiform clouds below the main clouds are classified as Stratocumuliform, Stratiform and Cirriform, all in the same small region. Model 28 is able to classify more parts of the Stratiform clouds correctly, however, only model 26 is able to detect the faint areas of these clouds at the lower level, even though it classified it incorrectly.

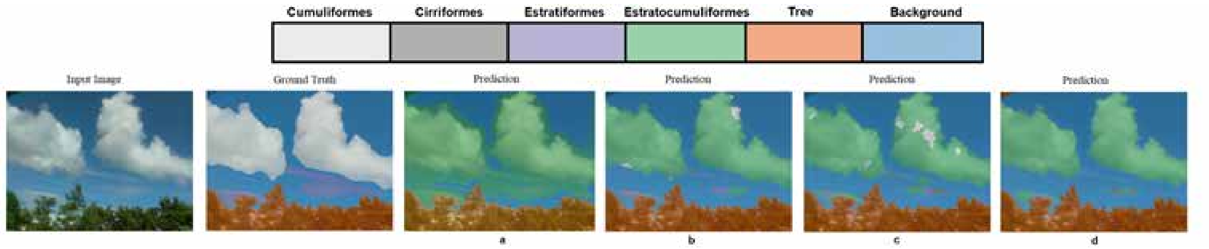


Figure 15: Example of model's inference with incorrect predictions for Cumuliform clouds and localization problem

Figure 16 shows two examples of segmentation inference of the best overall model (b). In contrast, the Resnet model with 18 residual layers and 972x1296 resolution presented only a slight improvement over the model using 243x324 resolution. With this model, it barely segment the most predominant class in the dataset, the Tree class.

The results achieved using the Resnet with 34 residual layers architecture (model 32) were not so positive. Figure 17 shows an example of inference using the best models with 486x648 (top) and 972x1296 (bottom) resolution. All resulting inferences presented the same problem with poor segmentation, with the Model 28arely able to identify the Tree class.

The Efficientnet model achieved an average mIoU of 0.3622. A good segmentation is presented in Figure 18 (top). The model performs well when inferring the class with more training samples, Stratocumuliform. Even though the resulting segmentation is not very fine where patches of the sky appear in the middle of the thin clouds atop the image, the model is able to make fine segmentation with the Tree class at the bottom. Some very distant clouds on the horizon were not segmented as well. However, the struggle to segment well the less represented classes is clearly visible (bottom). This result shows that the model is capturing some information about the Stratiform class, but still making incorrect inferences over the same cloud, giving preference to the more predominant class. The same occurs at the top of the image, only this time the model was able to identify only a very small patch of the correct

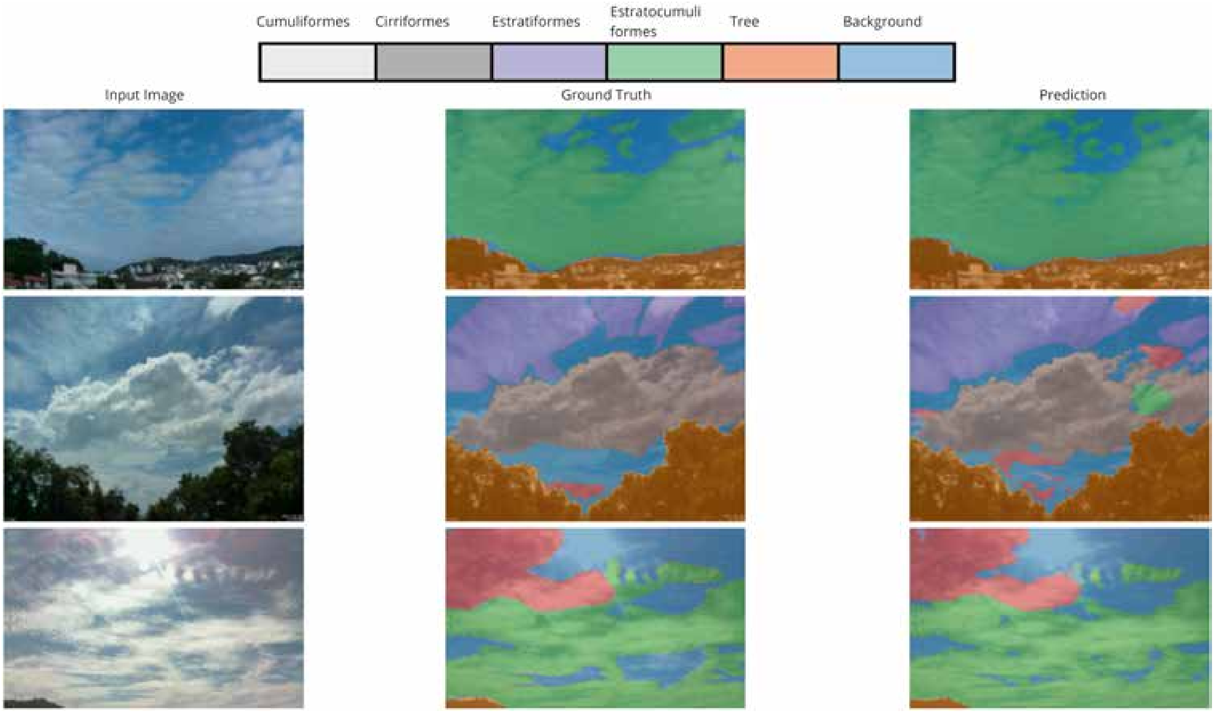


Figure 16: Example of resulting segmentation inference using the Resnet18 with 486x648 resolution model (b)

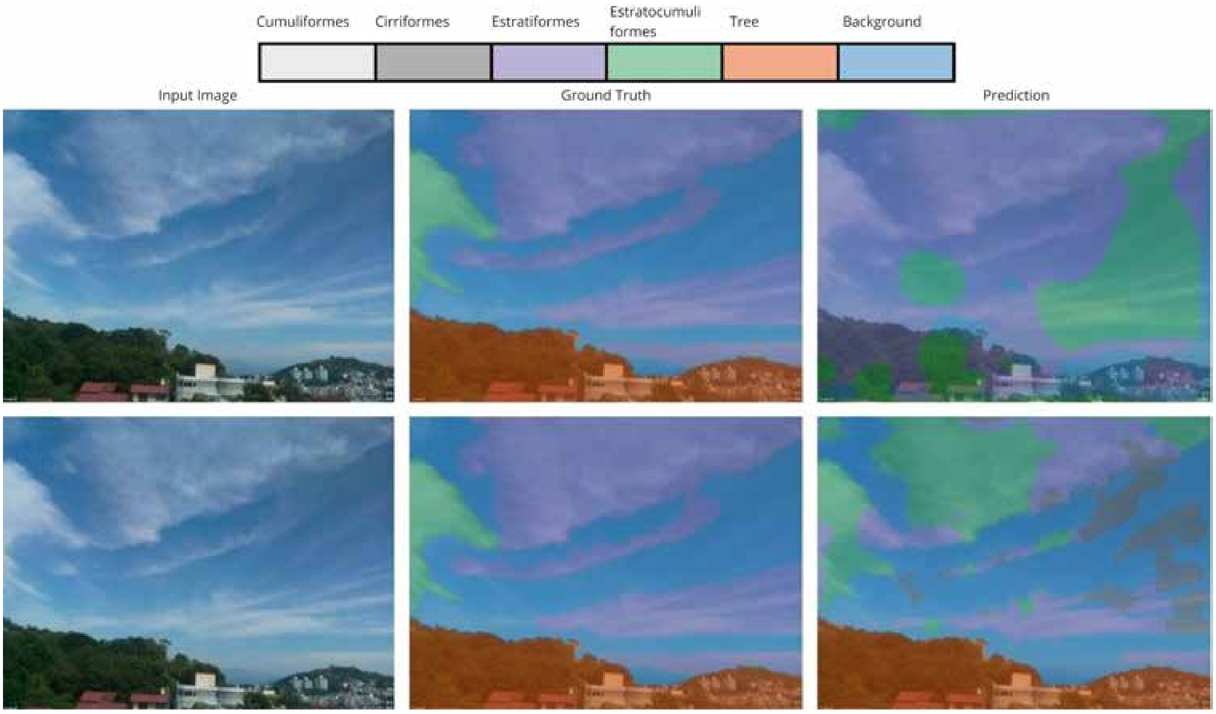


Figure 17: Example of inference results with 486x648 resolution (top) and 972x1296 resolution (bottom), using the Resnet with 34 residual layers architecture. The latter corresponds to model 32 discussed previously.

Cirriform class and wrongly segmented the cloud, similar to the bottom cloud. The model captures information about the Cirriform class and segments the same cloud into two different classes. This is most likely due to the thin texture of these clouds.

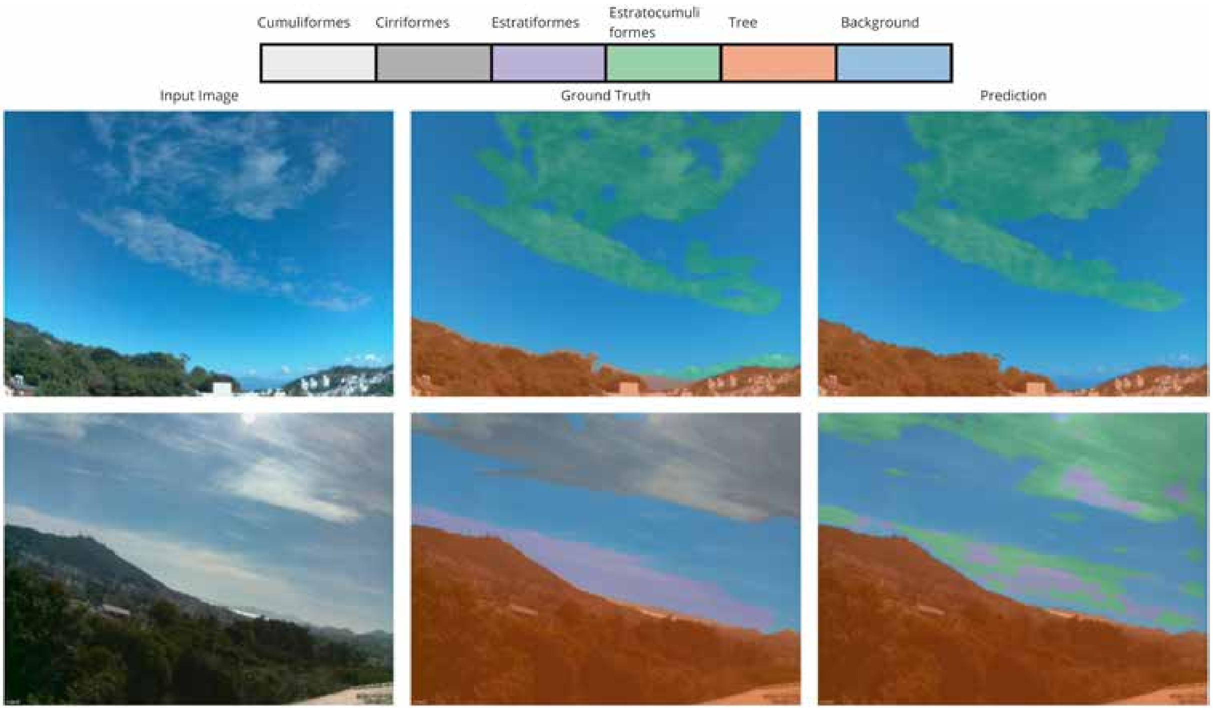


Figure 18: Example of good and bad resulting segmentation inference with U-net and Efficientnet model

The instance segmentation model was assessed for Average Precision (AP) after training using the COCOEvaluator class in Detectron2 Lin et al. [(2014]. The results can be seen in Table 10, where Type represents the type of result, which can be: bounding boxes results or segmentation pixels. The category represents one of the 5 classes, in which the ”Tree” category represents trees and buildings and the remaining 4 classes are cloud types. A threshold of 80% confidence was used for inference, which is a common practice.

It’s possible to see that the Tree class has the highest score, which is expected since the trees and buildings are virtually static, are present in basically all images, and can be easily distinguishable from clouds. Following that, the Stratocumuliform class is the cloud class with the highest score likely due to the abundance of images with this type of cloud in the dataset. This class is present in 81.53% of the entire dataset. The classes Stratiform and Cirriform are present in 27.21% and 28.61% of the images in the dataset, respectively. However one can see that the model can distinguish better Cirriform clouds in both types of results. The Cumuliform class is only present in 9.04% of the images, leading to believe that the results are a reflection of that as well. Overall results are shown in Figure 19.

A qualitative evaluation was also performed to observe if the ”localization” problem

Table 10: Detectron2 results separated by Bounding box and Segmentation pixels. Results are given by Average Precision (AP) per image category.

Type	**Category**	**AP Val**	**AP Test**
Bounding box	Tree	89.948	89.064
Segmentation pixels	Tree	85.603	84.029
Bounding box	Stratocumuliform	22.394	21.021
Segmentation pixels	Stratocumuliform	19.306	17.524
Bounding box	Stratiform	2.305	5.128
Segmentation pixels	Stratiform	2.063	4.939
Bounding box	Cirriform	9.676	9.419
Segmentation pixels	Cirriform	7.079	5.678
Bounding box	Cumuliform	0	5.941
Segmentation pixels	Cumuliform	0.594	6.733

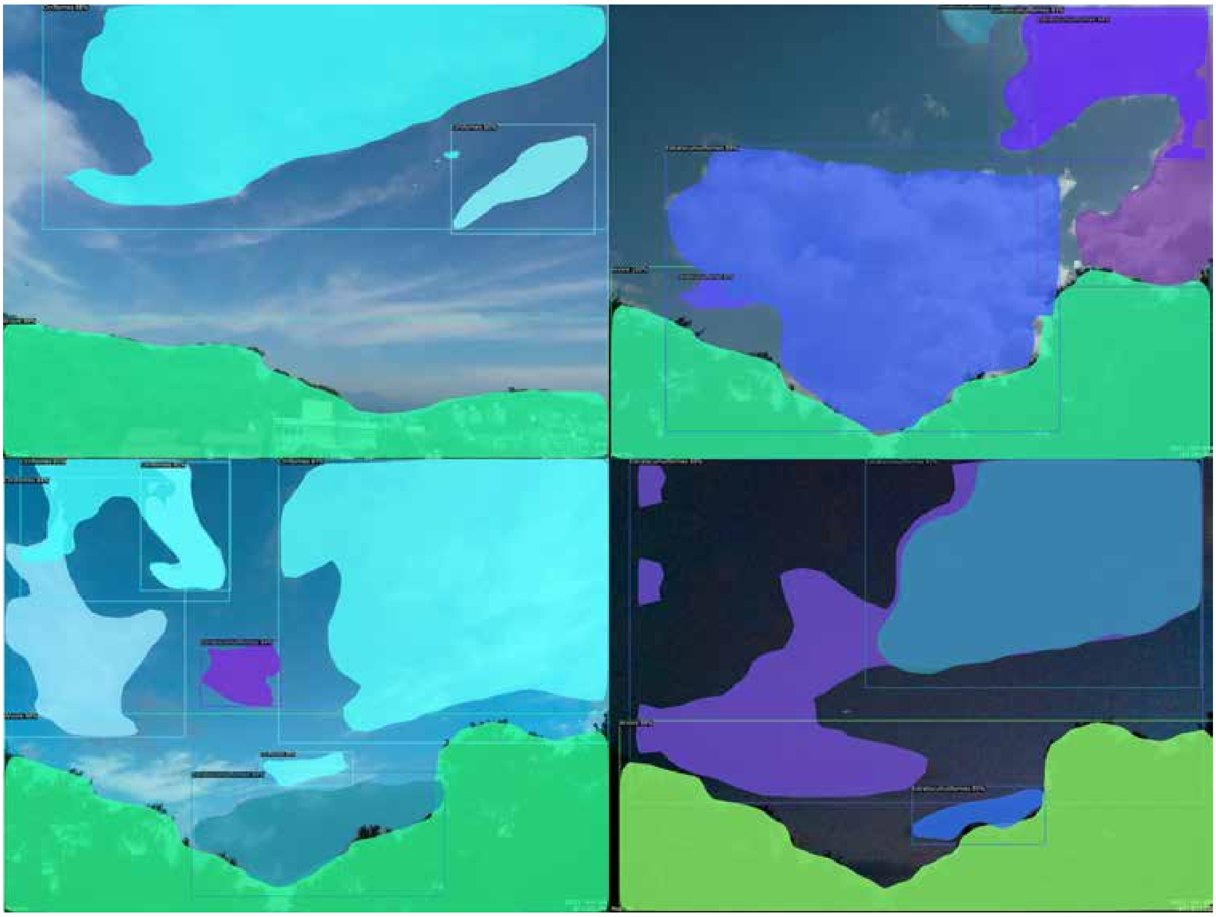


Figure 19: Resulting Detectron2 segmentation examples

ocurred with this type of model as well. Result examples of this can be seen in Figure 20. This problem is less common with this type of model, however, it's still present with different characteristics. On the right side of the image, is possible to observe a detection of multiple Stratocumuliform clouds where in fact there's only one predominant large cloud present with a few scattered on top. On the left side one Cirriform cloud is being classified as two objects.

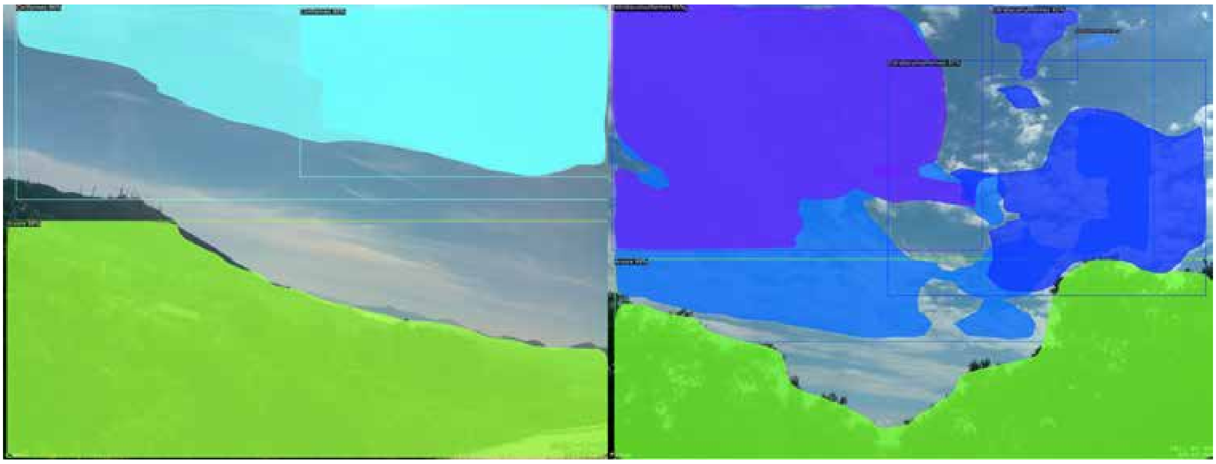


Figure 20: Example of localization problem where one big cloud is classified as two or more clouds of the same type

A problem with the detected region was also identified, where sometimes the model tends to crop out some parts of the object. Examples of this situation are shown in Figure 21. This is most likely due to the imposed 80% threshold for plotting the bounding boxes. During inference, the threshold is utilized to filter out low-scored bounding boxes predicted by the model's Fast R-CNN component. Predictions with a confidence score lower than the threshold are discarded, therefore it is possible to have resulting inference with no cloud classification whatsoever.

3.3.3 Discussion

This batch of experiments offers significant insights into cloud classification using deep learning models. A variety of architectures, including U-net with ResNet and HRNet, were employed, demonstrating nuanced advantages in handling different cloud categories. Despite inherent challenges posed by atmospheric conditions and cloud variability, the models showcased promising results in automated cloud classification, which is crucial for nowcasting solar irradiance absorption.

One notable finding was the effectiveness of simpler architectural models in certain

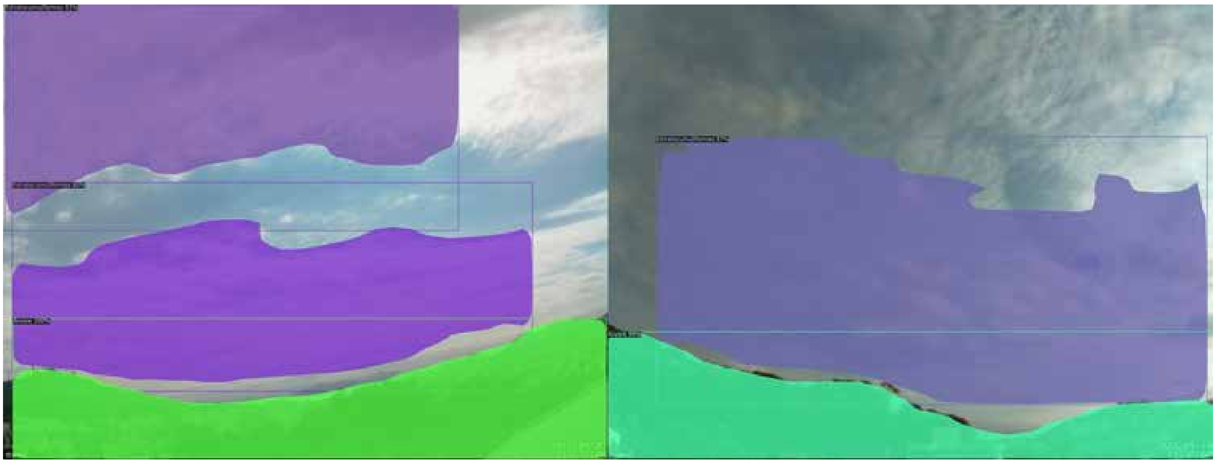


Figure 21: Example of threshold problem where clouds are not classified due to the confidence being lower than the imposed threshold

scenarios. For instance, a U-net with ResNet 18 architecture achieved an average IoU of 0.564, indicating that less complex models can sometimes yield sufficiently accurate results, especially in less demanding resolution settings. This finding is particularly relevant for practical applications where computational efficiency is as important as accuracy.

Moreover, the study revealed the necessity of balancing the representation of various cloud classes in the dataset. The over-representation or under-representation of certain cloud types can skew the model's learning, leading to biases in classification. This highlights the need for a meticulously curated and balanced dataset for training robust and generalizable cloud classification models. This is especially apparent in the distribution of the Tree class, which appears in 99.3% of the dataset images. This over-representation leads to skewed model learning, heavily favoring the predominant class over others. Addressing this imbalance is crucial for achieving a more accurate and unbiased cloud classification.

The significant presence of the Tree class in the dataset has led to inflated results. The models' high accuracy in identifying this class has obscured their performance in accurately classifying cloud types. A class-specific evaluation is necessary to understand the models' true performance in cloud segmentation.

Data augmentation emerged as a crucial strategy in enhancing model robustness and dealing with dataset imbalances. However, its application must be carefully calibrated to prevent suboptimal generalization.

In summary, the Clouds-1000 experiments underscore the dynamic interplay between model complexity, dataset characteristics, and the inherent variability of cloud formations. They advocate for ongoing experimentation and refinement of deep learning approaches in cloud segmentation tasks, acknowledging both the strides made and the challenges that remain.

3.3.4 Next Steps

Building on the insights gained from this round of experiments, the next phase of the research will focus on enhancing our understanding of the impact of various architectural choices, data preprocessing techniques, and class considerations on the performance of neural networks in image segmentation. The planned experiments are designed to explore the following key areas:

- **Assessment of Existing Models**: We plan to evaluate U-net with different backbones like Resnet, HRNet, and EfficientNet, using a class-balanced dataset. This will help us understand how established architectures perform with varied class distributions.
- **Data Augmentation Exploration**: We aim to investigate the effects of diverse data augmentation techniques on model robustness and generalization. This will help in understanding how different augmentation strategies can enhance model performance.
- **Tree Class Analysis**: By considering the inclusion or exclusion of the Tree class, we seek to gain insights into the impact of specific class characteristics on the overall performance of the model.
- **Comparison with Vision Transformers**: We will extend our evaluation to include Vision Transformers (ViTs). Comparing these with convolutional models will provide valuable insights into their respective strengths and weaknesses in the context of image segmentation.

3.4 Clouds-1500 Experiments

After another set of experiments and validation of the proposed framework, including the dataset quantity and quality, a new version of the dataset was created. This version follows the same strategy as the Clouds-1000 version of the dataset, with the addition of 500 images.

Despite efforts to maintain a balanced dataset, the team was unable to achieve an ideal balance, even with virtually one year of captured images. This imbalance is largely due to the peculiarity of the climatology of Florianópolis, which presents a significant predominance of

Table 11: Distributions of image groups used in the most recent version of the dataset, Clouds-1500

Cloud Type	Amount in Dataset	% in Dataset
Tree	1446	96.40%
Stratocumuliform	1122	74.80%
Stratiform	435	29.00%
Cirriform	383	25.53%
Cumuliform	246	16.40%

Stratocumuliform clouds compared to clouds of the Cumulus family. The table 11 shows the number of images that use their respective class.

As shown in Table 12, the Tree class had an increase of 387 instances (39.13%) the Stratocumuliform class grew by 310 instances (38.18%), and the Stratiform class expanded by 164 instances (60.52%). Additionally, Cirriform and Cumuliform classes increased by 98 (34.39%) and 156 (173.33%) instances, respectively, indicating significant augmentation of the dataset to facilitate more comprehensive studies.

Table 12: Comparison of dataset expansion from version Clouds-1000 to Clouds-1500

Class Type	Amount Added	Percentage Increase (%)
Tree	457	46.20%
Stratocumuliform	310	38.18%
Stratiform	164	60.52%
Cirriform	98	34.39%
Cumuliform	156	173.33%

The experiments were carefully structured, taking into account both the qualitative and quantitative aspects of model performance. The use of two versions of the Clouds-1500 dataset, one including the Tree class and one without, maintained consistency across all tests, ensuring that the insights derived were valid and comparable. This comprehensive examination aims to contribute valuable insights to the field of image segmentation and offers practical guidance for researchers and practitioners working on similar challenges.

For this batch of experiments, 29 models were created and evaluated in total. In order to perform a data selection process for creating train, validation, and test subsets from a dataset of images and their corresponding masks the technique demonstrated in 2 was performed.

This version of the dataset consists of a total of 1500 images, however some images were missing class annotations and had to be removed from the dataset for the following experiments. This exclusion lead to a total of 1478 images used. Now, let $\mathscr{D}$ represent the set of available directories containing images and masks, and let $\mathscr{P} = \{5,4,3,2,1\}$ be the set of priority values assigned to the pixel values in the masks. The algorithm follows the following steps:

Algorithm 2 Image Subset Selection Algorithm

```
Let $\mathscr{D}$ be the set of available directories
Let $\mathscr{P} = \{5,4,3,2,1\}$ be the priority values
Initialize counters: $n_{\text{train}} \leftarrow 0$, $n_{\text{validation}} \leftarrow 0$, $n_{\text{test}} \leftarrow 0$
for subset $\in \{\text{train}, \text{validation}, \text{test}\}$ do
    Create output directories for subset
    for all files in subset directory do
        if file is an image (e.g., ".jpg") then
            mask_file $\leftarrow$ Generate mask filename
            selected $\leftarrow$ False
            for all directory $\mathscr{D}_i$ in shuffled $\mathscr{D}$ do
                if image and mask files exist in $\mathscr{D}_i$ then
                    Copy image and mask to subset and corresponding mask subset
                    Update subset image counter
                    selected $\leftarrow$ True
                    Break loop
                end if
            end for
            if selected is False then
                Continue to next file
            end if
        end if
    end for
end for
```

The algorithm ensures a representative distribution of images with varying pixel values across the three subsets, while also considering a shuffling mechanism to avoid any bias in the selection order. The output is the creation of three subsets: train, validation, and test, containing a total of 1227, 28, and 223 images, respectively.

3.4.1 Description

Table 13 summarizes all the performed experiments, consolidating parameters across all different approaches implemented. This synthesis provides a coherent view of the setup, serving as a reference for details and groundwork for discussing results.

Efficientnet is grounded in an established architecture, complemented with an efficient

Table 13: Summary of all the performed experiments

#	Experiment	Backbone	Tree	Augmentation	Input Size	Batch Size	Device
35	EfficientNet	U-net with b0	Yes	No	1280 x 1280	2	RTX 3090
36	EfficientNet	U-net with b0	No	No	1280 x 1280	2	RTX 3090
37	EfficientNet	U-net with b0	No	Yes	1280 x 1280	2	RTX 3090
38	EfficientNet	U-net with b1	No	Yes	1280 x 1280	2	RTX 3090
39	EfficientNet	U-net with b1	No	No	1280 x 1280	2	RTX 3090
40	HRNet	OCRNet	Yes	No	1280 x 1280	2	Tesla V100
41	HRNet	OCRNet	No	No	1280 x 1280	2	RTX 3090
42	HRNet	OCRNet	No	Yes	1280 x 1280	2	RTX 4090
43	HRNet	OCRNet	No	Yes	1280 x 1280	2	RTX 3090
44	U-net + Resnet	Resnet 18	Yes	No	243x324	12	Tesla V100
45	U-net + Resnet	Resnet 18	Yes	No	486x648	4	Tesla V100
46	U-net + Resnet	Resnet 18	Yes	No	972x1296	1	Tesla V100
47	U-net + Resnet	Resnet 18	No	No	243x324	60	a100 40gb
48	U-net + Resnet	Resnet 18	No	No	486x648	16	a100 40gb
49	U-net + Resnet	Resnet 18	No	No	972x1296	4	a100 40gb
50	U-net + Resnet	Resnet 18	No	Yes	243x324	60	a100 40gb
51	U-net + Resnet	Resnet 18	No	Yes	486x648	16	a100 40gb
52	U-net + Resnet	Resnet 18	No	Yes	972x1296	4	a100 40gb
53	U-net + Resnet	Resnet 34	Yes	No	243x324	12	Tesla V100
54	U-net + Resnet	Resnet 34	Yes	No	486x648	4	Tesla V100
55	U-net + Resnet	Resnet 34	Yes	No	972x1296	1	Tesla V100
56	U-net + Resnet	Resnet 34	No	No	243x324	12	Tesla T4
57	U-net + Resnet	Resnet 34	No	No	486x648	4	Tesla T4
58	U-net + Resnet	Resnet 34	No	No	972x1296	1	Tesla T4
59	U-net + Resnet	Resnet 34	No	Yes	243x324	60	a100 40gb
60	U-net + Resnet	Resnet 34	No	Yes	486x648	16	a100 40gb
61	U-net + Resnet	Resnet 34	No	Yes	972x1296	4	a100 40gb
62	Transformer	PP-LiteSeg	No	Yes	1280 x 1280	2	a100 40gb
63	Transformer	Segformer	No	Yes	1280 x 1280	2	a100 40gb

backbone. For image processing, dimensions were changed to 1280×1280 based on hardware processing capabilities constraints; empirical observations from prior testing indicated optimal memory utilization on 16GB RAM GPUs.

For calculating the loss we used the *CrossEntropyLoss*. For it allows for heavier penalization of incorrect predictions, providing a nuanced error gradient which aids learning. Optimization was achieved using an adaptive optimizer. This optimizer is a combination of two other optimizer, which computes adaptive learning rates for parameters. The update rule adapts weights using moment estimates, speeding up convergence and lessening the need for learning rate adjustments.

A consistent rate was maintained across experiments to ensure uniformity and comparability. Fluctuating rates could introduce variances, complicating the attribution of performance differences to architectures or other adjustments.

Training duration was capped due to time constraints associated with computational resources. Specifically, processing time for a single epoch was approximately 28 hours with one concurrent worker.

1. A baseline model developed through architecture search to identify optimal architectures at different resolutions.
2. A scaled-up version of the previous model, with more parameters, potentially more accurate but computationally expensive. Scaled up using a method which increases network aspects.

Techniques to artificially expand training data by applying various transformations were mentioned. Here are the specific techniques:

- An augmentation that mirrors images on a horizontal axis. Useful since the vertical orientation is crucial and constant.
  - Probability of mirroring during training.
- An augmentation combining three transformations:
  - Moving the image by pixels.
  - Zooming in or out, aiding in recognizing varying sizes.
  - Rotating by an angle, aiding in orientation invariance.

With the foundation of an efficient architecture and key augmentations chosen, a series of tests were conducted. Each aimed at understanding the impact of variants and transformations on task performance. The following detail each setup:

- A combination with an efficient architecture:

  - Variation 1: Included a class with a certain number of workers.
  - Variation 2: Excluded a class with a different number of workers.
  - Variation 3: Used transformations, excluded a class, and adjusted the number of workers.

- Another combination with a scaled-up architecture:
  - Configuration 1: Excluded a class, used transformations similar to the previous combination, and set a certain number of workers.
  - Configuration 2: Excluded a class, did not use transformations, and set the same number of workers.

Adjusting the number of workers across configurations was strategic, responding to the environment. Given concurrent processes, adjusting helps in managing resources efficiently. By doing so, data loading is optimized without overburdening, allowing for smooth sessions and consistent testing.

For the HRNet experiments we performed a resizing to 1280×1280 matched memory specifications, ensuring batch processing accommodation and used a backbone and variant optimized for long-range contextual capture in segmentation.

For loss calculation we also used *CrossEntropyLoss* optimized with *AdamW*, a variant including weight decay, parameters set to penalize weight magnitude, aiding overfit prevention and model generalization.

Training spanned 80,000 iterations, a figure based on complexity and dataset size to ensure convergence.

Exploration of training conditions was aimed at performance optimization and understanding impacts of various configurations:

1. Inclusion of the tree class allowed direct comparison with prior experiments incorporating this class. Trained using infrastructure with a balance between computational power and accessibility.

2. Noting possible detractions from overall performance by the tree class, training of a configuration specifically excluding it aimed to discern its influence on segmentation and ascertain any efficacy improvement. Trained using an advanced computational ecosystem.

3. With potential augmentation benefits recognized, an exploration combined class exclusion with augmentation techniques to measure enhancement impacts. The implemented augmentations introduced mirrored image versions, combined with color and geometric perturbations, and pixel value scaling.

For the U-net + Resnet experiments, the same incremental resizing technique described in section 3.2 was adopted, allowing training at multiple resolutions. Beginning with 243×324, then 486×648, and 972×1296. This method's rationale is that smaller images speed up initial epochs, while larger images, introduced later, provide finer details. These experiments employ a series of incremental input sizes and variations in the Resnet backbone (Resnet 18 and Resnet 34) to evaluate the model's segmentation performance across different cloud classes.

Each size increase was preceded by a fine-tuning phase of 15 epochs to adapt to new dimensions. The *lr_find()* method from FastAI, plotting loss versus rates, guided optimal rate selection. Post-fine-tuning training lasted 30 epochs for the first sizes and 40 for the largest, necessary for capturing intricate details at higher resolutions.

CrossEntropyLoss was, once again, consistently used, with a steady weight decay of $1e^{-3}$ to mitigate overfitting. A custom metric, DiceMulti, monitored performance, aggregating scores across classes, offering a comprehensive efficacy view.

Transformers are a class of neural network architectures that have revolutionized sequence-to-sequence tasks. Introduced by Vaswani et al. [2023], their fundamental premise is built around the self-attention mechanism.

Given a sequence of input tokens, $x_1, x_2, \ldots, x_n$, each token is transformed into corresponding query Q, key K, and value V representations using learned weight matrices W_Q, W_K, and W_V respectively:

$$Q = x \times W_Q$$

$$K = x \times W_K$$

$$V = x \times W_V$$

The attention scores between a query from one token and the keys from all tokens are computed using a dot product:

$$\text{Score}(Q, K) = Q \cdot K^T$$

These scores determine the weight of each token's value for the current token, after being normalized with a softmax function:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{\text{Score}(Q,K)}{\sqrt{d_k}}\right) \times V$$

Where d_k is the dimension of the key vectors, and the division by $\sqrt{d_k}$ is for scaling purposes.

The output of the self-attention mechanism for each token is a weighted sum of all value vectors, where the weights are determined by the attention scores.

In practice, multiple self-attention mechanisms (or "heads") are run in parallel, each using different weight matrices. The outputs of these heads are concatenated and linearly transformed to produce the final output.

When applied to cloud segmentation, the transformer can capture long-range dependencies between different regions of an image. Traditionally, convolutions in CNNs have a localized view (defined by their kernel size), but the self-attention mechanism in transformers can weigh the importance of all other pixels when considering a particular pixel.

In the context of cloud segmentation:

1. Cloud structures can be vast and intricately connected, making it vital to understand the global context and dependencies between different parts of an image.

2. Transformers can provide a more holistic view of an image, allowing the model to recognize patterns and structures that span large distances.

3. By attending to distant but related parts of an image, transformers can potentially improve the granularity and accuracy of cloud segmentation.

While transformers were originally designed for sequence-to-sequence tasks, their inherent capability to capture long-range dependencies makes them promising candidates for tasks like cloud segmentation, where understanding the global context of an image is crucial.

Within the scope of these experiments, the transformative capabilities of Transformer architectures were explored, notably through the lenses of the *PP-LiteSeg* and *Segformer* models. Both models were trained for 80.000 iterations in order to address the intricate challenges of cloud segmentation, especially the demarcation between nebulous cloud structures and clear skies, necessitated a particularly adept model. It was in this context that the *PP-LiteSeg*, with its integration of transformer mechanisms, emerged as a pivotal asset.

The *PP-LiteSeg* is introduced as a lightweight model tailored for real-time semantic segmentation tasks. A cornerstone of its design is the *Flexible and Lightweight Decoder (FLD)*, which has been conceptualized to mitigate the computational overhead traditionally associated with decoders in segmentation tasks. Pushing the envelope further, the Model 28oasts a *Unified Attention Fusion Module (UAFM)*. This module is a harmonious blend of spatial and channel attention mechanisms, culminating in the generation of a weight which subsequently fuses with the input features to augment the quality of feature representation.

Additionally, in a bid to combine global context without being penalized by high computational costs, the *Simple Pyramid Pooling Module (SPPM)* was introduced into the model. The synergy of these modules and mechanisms enables *PP-LiteSeg* to carve out a unique niche for itself, where it champions a commendable trade-off between speed and accuracy, setting it leagues apart from contemporaneous methodologies.

Recognizing the intrinsic challenges of cloud segmentation, where interpreting vast and complex cloud configurations is paramount, the decision to integrate *PP-LiteSeg* was strategic. The model's understanding of spatial relationships and patterns rendered it particularly adept at navigating the intricate mazes of cloud structures. The transformer functionalities within *PP-LiteSeg* furnished the comprehensive and holistic perspective of the task warranted, making it an invaluable component of the experimental arsenal.

Both experiments have the input images resized to 1280×1280 dimensions, with a batch size for training. The *RandomHorizontalFlip* data augmentation technique was applied to enhance the model's generalization capabilities.

To further tackle the localization problem described in section 3.3, a tailored approach within the Transformer architectures was employed, notably by leveraging a *MixedLoss* function. This function combines CrossEntropyLoss and SemanticConnectivityLoss in a calculated ratio to refine the segmentation output, specifically addressing the challenge of correctly delineating cloud structures without fragmenting them into multiple classes erroneously.

The *CrossEntropyLoss* serves as the primary component of the loss function, driving the model to align its predictions with the ground-truth class labels. It fundamentally encourages accurate classification on a per-pixel basis, which is critical for the segmentation task. However, when used alone, it might not fully capture the relational nuances between neighboring pixels, potentially leading to the previously observed localization problem where different regions of the same cloud are incorrectly classified as separate entities.

To counteract this, the *SemanticConnectivityLoss* was integrated, a novel loss component designed to preserve semantic continuity across the segmented regions. By emphasizing the connectivity of semantically similar regions, this loss component helps the model to recognize and maintain the integrity of individual cloud structures within the segmentation map. It effectively penalizes the model for producing fragmented classifications within a single cloud, encouraging it to produce a coherent segmentation that reflects the unified nature of each cloud entity.

By combining these losses through the *MixedLoss* method, with coefficients $[0.8, 0.2]$, individual pixel accuracy and semantic coherence across the entire image were aimed. This mixed approach fosters a balance, allowing the model to learn from the global structural information provided by the transformer's attention mechanisms while being steered away from over-segmentation.

The Transformer-based *PP-LiteSeg* and *Segformer* models inherently capture long-range dependencies within the image, which is beneficial for understanding the global context of cloud configurations. The integration of *SemanticConnectivityLoss* within the *MixedLoss* function complements these architectural strengths. It ensures that while the model benefits from the global perspective afforded by the Transformers, it does not lose sight of the local continuity that is crucial for accurate cloud segmentation.

The learning rate schedule follows a *PolynomialDecay* policy, starting from 0.0001 and decaying according to the equation:

$$\text{learning rate} = \text{initial learning rate} \times \left(1 - \frac{\text{current iteration}}{\text{total iterations}}\right)^{0.9}$$

This method ensures a smooth and adaptive learning rate adjustment, conducive to stable and effective optimization. The AdamW optimizer, with its β_1 and β_2 values of 0.9 and 0.999, respectively, along with a weight decay of 0.01, completes the optimization strategy, combining momentum and regularization for efficient convergence.

In the first Transformer experiment, the PP-LiteSeg model was employed, focusing on specific architectural choices. The STDC2 architecture backbone was used. This backbone choice involves a Spatio-Temporal Deep Convolutional design (STDC2), aimed at capturing both spatial and temporal information through a series of convolutional layers. It helps in leveraging hierarchical features from different stages for more effective segmentation.

The architecture is further characterized by the *arm_out_chs* and *seg_head_inter_chs* parameters. The *arm_out_chs* vector $[32,64,128]$ represents the number of output channels in different arm stages, which is essential in controlling the capacity and complexity of the model. The *seg_head_inter_chs* vector $[32,64,64]$ defines the number of channels in the intermediate layers of the segmentation head, offering flexibility in design and efficiency in computation.

The other experiment was conducted using the SegFormer model, which integrates the Vision Transformer (ViT) architecture with a novel segmenting strategy. The SegFormer leverages a backbone known as the MixVisionTransformer_B0, combining convolutional neural networks with transformers.

Vision Transformers (ViTs) represent an adaptation of the transformer architecture, initially designed for natural language processing tasks, to computer vision challenges. Unlike regular transformers, which operate primarily on sequential data, ViTs apply the transformer's self-attention mechanism directly to a sequence of image patches. This allows the model to capture relationships and dependencies across different regions of the image, which is a departure from the localized feature extraction commonly found in convolutional neural networks (CNNs).

In a traditional transformer, the input is usually a sequence of tokens, often words or subwords, embedded in a continuous vector space. Attention mechanisms within the transformer then enable the model to weigh the significance of each token in relation to others, allowing for the modeling of complex dependencies in the input sequence.

In contrast, the Vision Transformer begins by dividing an input image into non-overlapping patches, treating them as analogous to the tokens in a text sequence. These patches are linearly embedded into a vector space and processed through a series of transformer layers. The self-attention mechanism allows the model to consider the entire image at once, thereby capturing global patterns and dependencies that might be missed by localized convolutions.

While the transformer's self-attention mechanism offers a powerful means of modeling dependencies in sequential data, the Vision Transformer's adaptation of this architecture to visual data opens up new possibilities for understanding and representing images. By operating on patches of the image rather than on individual pixels or localized features, the Vision Transformer can abstract and generalize visual patterns across the image, potentially offering superior performance in tasks like image classification, object detection, and semantic segmentation.

The departure from localized convolutions to global attention represents a fundamental shift in perspective and offers a complementary view of visual data. This has led to the emergence of hybrid models, such as SegFormer, that combine the strengths of both transformers and traditional CNNs. Such integrations aim to balance the local feature extraction capabilities of CNNs with the global context awareness provided by transformers, creating models that are both interpretative and effective in complex visual tasks.

The SegFormer's innovative integration of transformers with traditional CNNs, coupled with the specific configuration for the cloud segmentation task, offers a promising direction for achieving both efficiency and effectiveness in segmenting cloud regions. Extensive evaluations are expected to elucidate the comparative performance between this model and the previously described PP-LiteSeg.

3.4.2 Results

Table 14 shows the results of all models created for this suite of experiments, highlighting the best results of each metric column in bold.

Results from experiments 35 through 39, applying EfficientNet with U-net structures (with b0 and b1 structures), show varied performance across different cloud categories. These outcomes reveal nuanced capabilities under varying conditions of category inclusion and data transformation.

Experiment 35 showed exceptional detection in one category is noted with the highest measures of 0.94 and 0.96, and an outstanding value of **0.98**, as seen in predictions for another category. Despite this, there is a notable decline in metrics for other categories across all tests, with measures reaching zero. This suggests limitations in segmentation abilities for these categories, potentially due to factors like category imbalance or unrepresentative data for training.

As demonstrated in Figure 22, comparisons between predictions from experiment 35 and the ground truth emphasize segmentation effectiveness for certain cloud categories. Yet, they also reveal difficulties in capturing subtle features of other categories, with zero recall for these. The consistent metrics in background elements, contrasted with over-segmentation in some clouds, reveal nuanced strengths and areas for refinement, such as addressing category imbalances and enhancing feature capture for fine-structured clouds.

Experiments with a more complex backbone (38 and 39) do not show a marked improvement over the simpler backbone (35 to 37), suggesting that increasing complexity does not translate to proportional gains in segmentation for the cloud categories and data used.

Table 14: Results of all models over the test dataset, by class

	Tree			**Stratocumuliform**			**Stratiform**			**Cirriform**			**Cumuliform**		
#	IoU	Pr	Recall	IoU	Pr	Recall	IoU	Pr	Recall	IoU	Pr	Recall	IoU	Pr	Recall
35	0.94	0.96	**0.98**	0.54	0.57	0.91	0.54	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	-	-	-	0.53	0.55	**0.94**	0.00	0.14	0.00	0.00	0.11	0.00	0.00	0.09	0.00
37	-	-	-	0.53	0.55	**0.94**	0.00	0.03	0.00	0.00	0.02	0.00	0.00	0.07	0.00
38	-	-	-	0.52	0.55	0.91	0.00	0.09	0.00	0.00	0.04	0.00	0.00	0.08	0.00
39	-	-	-	0.54	0.58	0.89	0.00	0.13	0.00	0.00	0.02	0.00	0.00	0.20	0.00
40	**0.97**	**0.99**	**0.98**	0.48	0.74	0.57	0.44	0.52	0.76	0.26	0.44	0.39	**0.64**	**0.77**	0.79
41	-	-	-	0.64	0.70	0.87	0.46	0.67	0.59	0.16	**0.73**	0.17	0.39	0.69	0.48
42	-	-	-	0.58	0.69	0.78	0.34	0.65	0.42	0.34	0.52	0.50	0.61	0.65	**0.92**
43*	-	-	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
44	0.89	0.97	0.91	**0.89**	**0.98**	0.91	**0.59**	**0.71**	0.77	0.26	0.49	0.36	0.34	0.56	0.46
45	0.89	0.97	0.91	0.59	0.69	0.79	0.23	0.47	0.31	**0.39**	0.63	0.51	0.27	0.62	0.33
46	0.91	0.97	0.93	0.59	0.69	0.79	0.23	0.47	**0.88**	**0.39**	0.63	0.51	0.27	0.62	0.33
47	-	-	-	0.51	0.75	0.62	0.20	0.40	0.30	0.34	0.46	0.57	0.28	0.49	0.39
48	-	-	-	0.58	0.67	0.82	0.14	0.48	0.16	0.37	0.57	0.52	0.28	0.51	0.38
49	-	-	-	0.57	0.63	0.85	0.09	0.41	0.11	0.38	0.50	**0.63**	0.28	0.51	0.38
50	-	-	-	0.58	0.68	0.80	0.23	0.48	0.32	0.36	0.57	0.50	0.25	0.51	0.33
51	-	-	-	0.59	0.67	0.82	0.17	0.50	0.20	**0.39**	0.61	0.52	0.27	0.46	0.40
52	-	-	-	0.54	0.67	0.73	0.24	0.38	0.39	0.37	0.60	0.49	0.23	0.54	0.29
53	0.88	0.98	0.90	0.55	0.73	0.69	0.26	0.43	0.39	**0.39**	0.57	0.56	0.29	0.50	0.40
54	0.92	0.97	0.94	0.55	0.71	0.71	0.22	0.45	0.29	0.37	0.52	0.57	0.30	0.37	0.32
155	0.94	0.97	0.96	0.52	0.62	0.77	0.11	0.32	0.14	0.31	0.50	0.45	0.05	0.48	0.06
56	-	-	-	0.59	0.71	0.67	0.27	0.48	0.38	**0.39**	0.61	0.51	0.27	0.55	0.34
57	-	-	-	0.56	0.68	0.75	0.22	0.46	0.29	0.37	0.50	0.58	0.23	0.50	0.30
58	-	-	-	0.50	0.60	0.75	0.12	0.26	0.18	0.17	0.54	0.20	0.10	0.34	0.12
59	-	-	-	0.59	0.71	0.77	0.24	0.52	0.32	0.37	0.57	0.52	0.32	0.50	0.47
60	-	-	-	0.58	0.67	0.81	0.17	0.47	0.22	0.38	0.58	0.52	0.25	0.51	0.33
61	-	-	-	0.59	0.67	0.83	0.19	0.49	0.24	0.38	0.62	0.50	0.27	0.57	0.35
62	-	-	-	0.56	0.66	0.77	0.17	0.34	0.25	**0.39**	0.68	0.47	0.28	0.43	0.43
63	-	-	-	0.59	0.72	0.76	0.25	0.45	0.35	0.37	0.55	0.54	0.35	0.50	0.54

Figures 29, 30, 31, and 32 in the Appendix D section, illustrate outcomes of experiments 36 - 39. All structures have mislabeled most cloud formations, except for one category in Experiment 35. This misclassification is quantitatively reflected in the precision scores for Stratocumuliform clouds, which are relatively low compared to their recall rates that exceed 89% across the experiments. Such high recall rates indicate that the models are likely overfitting to

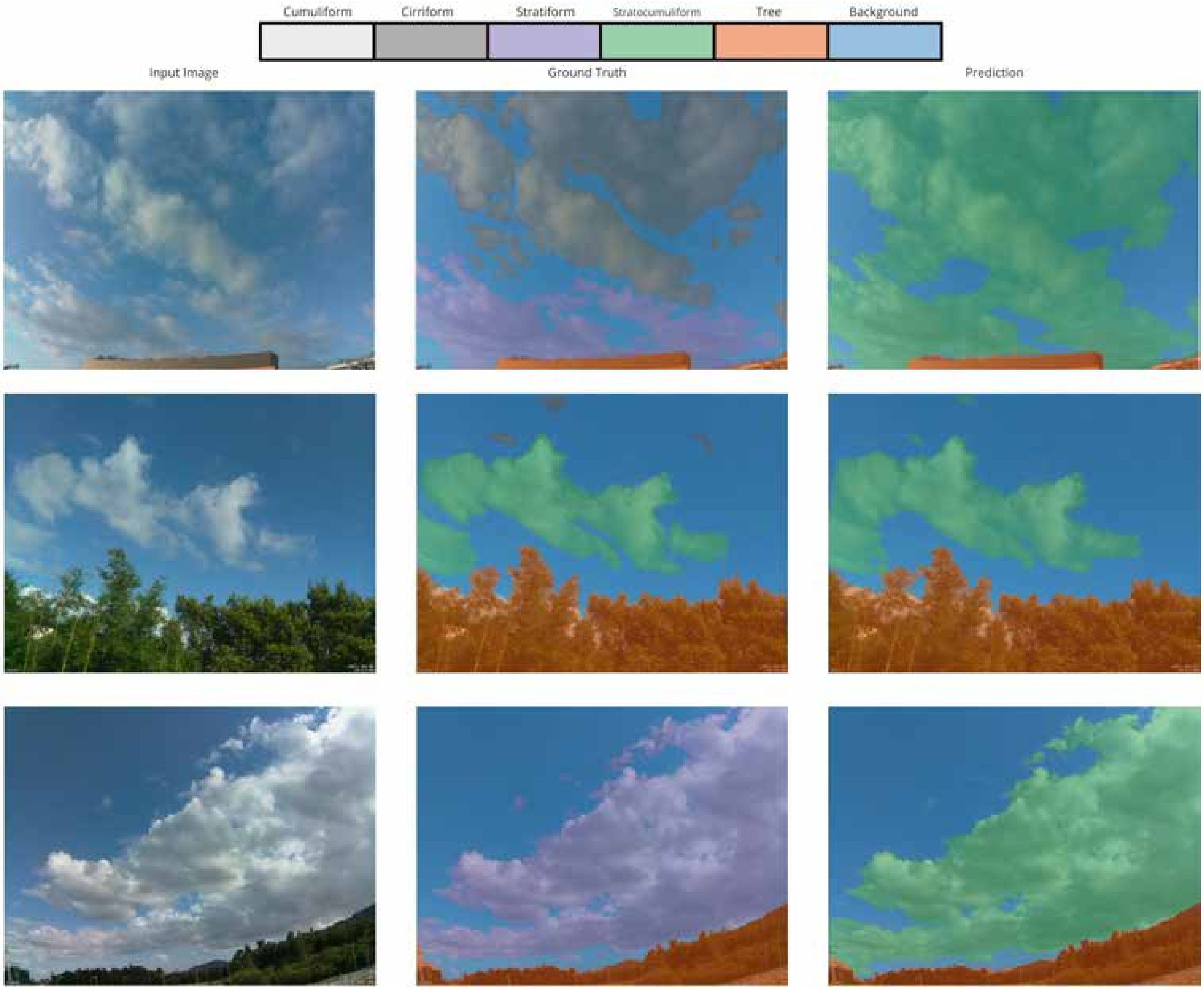


Figure 22: Experiment 35 sample results of inference over the test dataset

the Stratocumuliform class, leading to a high number of false positives. This overgeneralization severely undermines the models' ability to accurately identify and classify other cloud types, as seen by the negligible IoU scores for the Stratiform, Cirriform, and Cumuliform classes.

The results from experiments 40 through 43, utilizing HRNet combined with OCR-Net, illustrate a varied performance that is particularly informative regarding the network's ability to generalize across different cloud classes, resulting in some very good overall models.

Experiment 40 presents exemplary performance for the Background class with an IoU of 0.97, precision of 0.99, and recall of 0.98, signifying exceptional ability in distinguishing clear sky from clouds. This trend continues with high metrics for the Stratocumuliform and Cumuliform classes, although a significant dip in IoU for the Stratiform class is observed. Notably, the Cirriform class shows a robust precision of 0.77, despite a moderate IoU, which suggests confidence in the model's predictions for this complex cloud type. This model is the best model overall from experiments 40 to 43. It shows a strong ability to distinguish the background from cloud types and maintains a robust performance across various cloud classes. Experiment 40 exhibits a good balance between precision and recall, suggesting it is capable of generalizing well across the different cloud structures represented in the dataset.

Experiment 41 demonstrates an improved IoU of 0.64 for the Stratocumuliform class over experiment 40, indicating better segmentation capabilities. However, there is a noticeable decline in precision and recall for the Cumuliform class, as shown by the precision of 0.69 and recall of 0.48. This could suggest that while experiment 41 improves upon certain cloud types, it may not be as effective in distinguishing Cumuliform clouds as experiment 40.

Experiment 42 shows a strong recall of 0.92 for the Cumuliform class, which is a significant improvement, indicating the model's sensitivity to this type of cloud has been enhanced. Nevertheless, the lower IoUs for Stratocumuliform and Stratiform classes, when compared to experiment 40, imply that there could be a trade-off in the model's performance across different cloud types.

The difference in performance of experiment 43, where all metrics collapse to zero except for the Background class, which has perfect recall but extremely low precision and IoU, suggests a major issue with the model's ability to generalize. The high recall alongside a very low precision indicates that while the model is labeling almost all pixels as background, it is doing so incorrectly, leading to a failure in segmenting any cloud classes. The utilization of ”RandomHorizontalFlip, RandomDistort, Normalize” data augmentation techniques might have contributed to this generalization issue. It's possible that the combination of these augmentations introduced too much variability or altered the images in a way that the model could no longer effectively learn the relevant features for cloud segmentation.

The qualitative results from experiment 40 can be visualized in Figure 23, providing an illustrative representation of the model's performance. The strong capacity for cloud classification is evident in the first sample, where the model demonstrates a pronounced ability to distinguish between Cumuliform and Stratocumuliform clouds, reflecting the high Intersection over Union (IoU) scores. This aligns with the quantitative measures, which recorded an IoU of 0.97 for Cumuliform and 0.99 for Stratocumuliform classes. Despite some minor misclassifications, the predicted cloud contours are largely accurate, denoting a robust feature extraction capability. The other three experiments can be seen on Figures 33, 34 and 35 in the Appendix D section.

Challenges in classifying Stratiform and Cirriform clouds are observable in the first

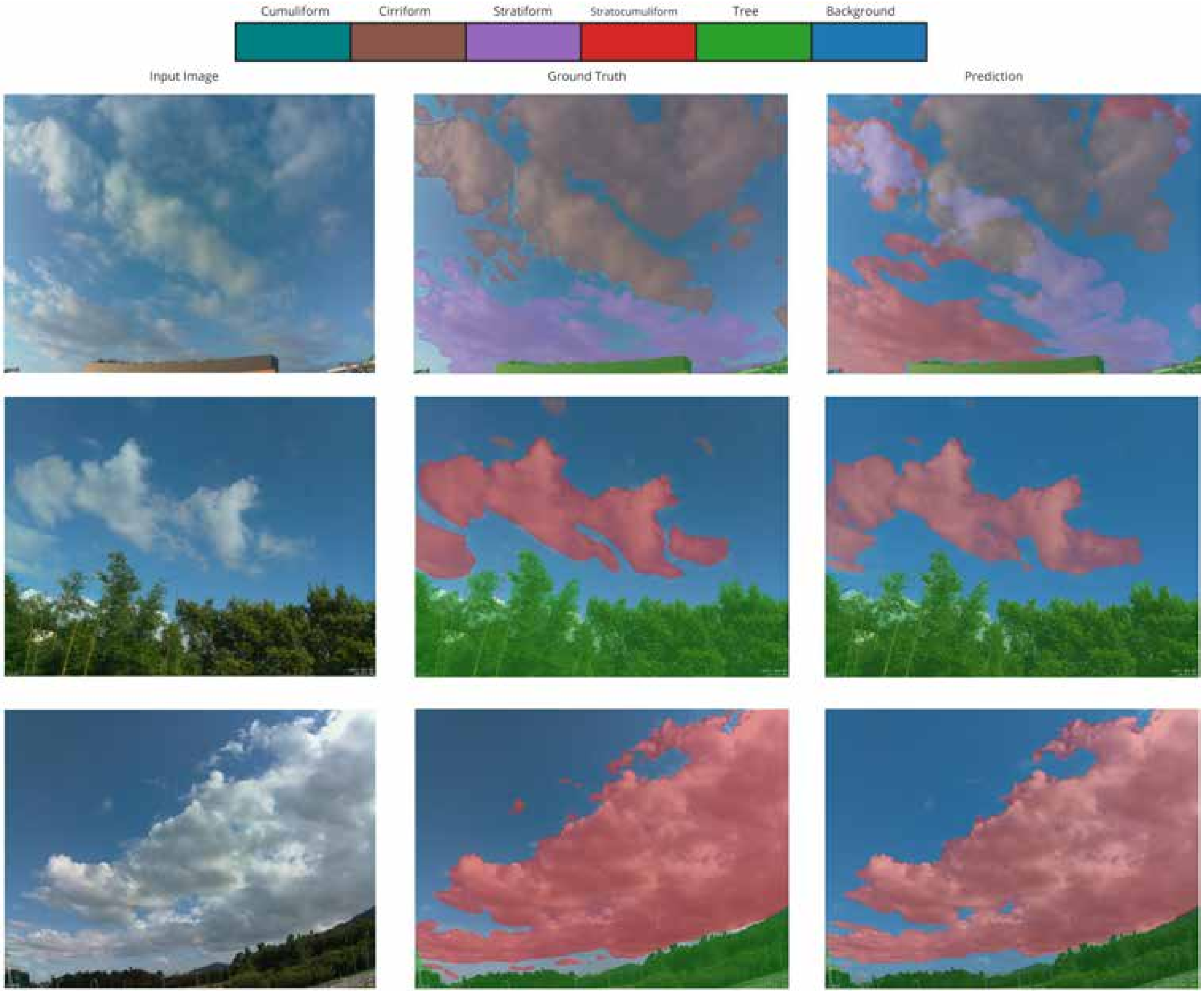


Figure 23: Experiment 40 sample results of inference over the test dataset

sample, which is consistent with the comparatively lower IoU scores of 0.48 and 0.57 for these classes, respectively. The model's propensity to overpredict certain cloud types indicates a potential area for improvement, specifically in differentiating cloud classes with subtler characteristics.

Overall, the visual samples from experiment 40, as depicted in Figure 23, underscore the model's proficiency in segmenting multiple cloud types with substantial accuracy, particularly for Cumuliform and Stratocumuliform clouds. The fidelity of the spatial distribution in the model's predictions points to a high degree of accuracy in cloud segmentation tasks. Nonetheless, the performance on Stratiform and Cirriform clouds, alongside the segmentation of trees and background, indicates a scope for further optimization. The demonstrated capabilities establish experiment 40's HRNet combined with OCRNet as the most adept model across the experiments, especially when considering its ability to generalize and its detailed adherence to the intricate structures of cloud forms.

The Unet + Resnet experiments generally show good performance for the Background and Stratocumuliform classes across most experiments, as indicated by relatively high IoU, precision, and recall values. The performance on the Stratiform class is moderate but significantly lower than for the Background and Stratocumuliform classes, which suggests that the models are less adept at capturing the features of this cloud type. However, experiment 44 demonstrates the highest IoU for Tree and Stratocumuliform, and very high precision and recall for these classes as well, establishing it as the strongest model overall for these classes. It also presents decent scores for Cirriform and Cumuliform clouds, indicating a more balanced performance compared to other models in the experiment.

The Cirriform and Cumuliform classes consistently present challenges for segmentation, as evidenced by lower performance metrics across all experiments. However, models 45 and 46 achieve the highest IoU values for Cirriform clouds, suggesting certain configurations within these experiments were more adept at segmenting this cloud type. Model 46 also shows notably high recall for Stratiform clouds, indicating its effectiveness at detecting most instances of this class despite the moderate IoU.

The incremental input sizes do not demonstrate a straightforward relationship between input size and segmentation performance. Some larger input sizes lead to better performance for certain classes (e.g., the Stratiform class in Experiment 46), but this is not a consistent trend across all classes and experiments. The varying performance across the input sizes suggests that while larger input sizes may provide more detailed information, they do not necessarily translate into better segmentation performance. This could be due to the fact that the increased resolution might introduce noise or overfitting issues.

Comparing the Resnet 18 (experiments 44 - 52) backbone variations, there is no clear indication that one backbone outperforms the other significantly across the different cloud classes up to Experiment 52. This indicates that the increase in resolution does not have a major impact on segmentation performance for this task.

For the Stratiform class, the highest IoU is observed in Experiment 44, suggesting that this particular configuration might be capturing the relevant features of this class more effectively. While the performance on Cirriform and Cumuliform classes is low across the board, models 45 and 46 for Cirriform and 10c for Cumuliform classes show that there are individual experiments where these classes have slightly higher metrics. This indicates that there might be specific configurations or conditions under which the models can capture these cloud types better, and these warrant further investigation.

In conclusion, while the experiments from 44 to 52 provide valuable insights, model 44 stands out for its overall balanced performance across various cloud classes. It appears to be the best model of the series, suggesting a potential direction for optimizing cloud segmentation tasks. Further investigation into the configurations of model 44 could yield important insights into improving segmentation accuracy for more complex cloud types.

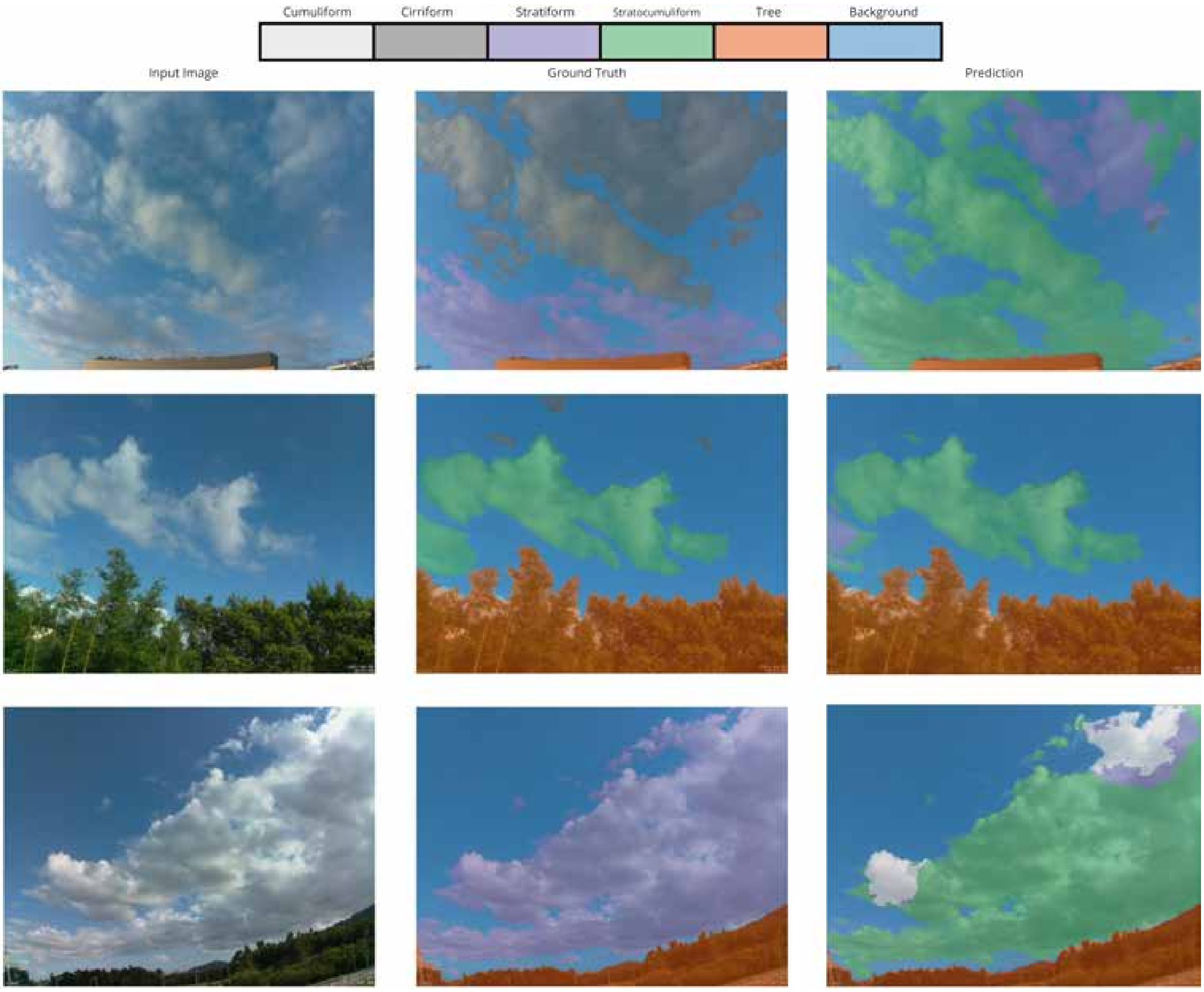


Figure 24: Experiment 44 sample results of inference over the test dataset

The inference samples in Figure 24 demonstrate the Resnet18 model's performance in classifying various categories such as Cumuliform, Cirriform, Stratiform, Stratocumuliform, Tree, and Background. The predictions made by the model largely conform to the ground truth with some noticeable exceptions, which is indicative of the model's high precision and recall in most categories, as reflected by the quantitative metrics of experiment 10a, showing particularly strong performance in the Stratocumuliform class with IoU 0.89, precision 0.98 and recall 0.91.

However, some challenges are visible in the differentiation between very similar classes, such as between Stratiform and Stratocumuliform clouds, as well as between cloud types and the background, where the contrasts are subtle. This is quantitatively corroborated by the relatively lower precision score in the latter. The results imply that while the model is generally robust in classifying distinct features, it may struggle with classes that have less pronounced distinguishing characteristics or where the segmentation boundaries are less defined. This observation could lead to further model refinement, potentially by augmenting the training data in underperforming categories or by tuning the model to better capture the nuances between similar classes.

The remaining results of the Resnet18 experiments are presented in Figures 36, 37, 38, 39, 40, 41, 42, 43, in the Appendix D section.

Moving forward to experiments 53 to 59, the modifications in the U-net + Resnet structure and input sizes continue to impact the model's performance on cloud segmentation. These experiments further the understanding of how backbone variations and input parameters affect segmentation accuracy across different cloud types.

In this series, models 53, 54, and 55, which utilize the Resnet 34 backbone, show an intriguing pattern. Notably, 53 and 54 both exhibit high IoU scores, with 53 demonstrating an impressive balance across IoU, precision, and recall metrics, particularly in Background, Stratocumuliform, and Stratiform classes. Model 54, while it has slightly lower IoU in certain classes compared to 53, it has a better recall across almost all classes, which may be preferred in applications where detecting the presence of a cloud type is more critical than precisely delineating its boundaries.

On the other hand, model 55, despite achieving the highest IoU and precision for Background, Stratocumuliform, and Stratiform classes, experiences a significant drop in performance in the Cirriform and Cumuliform classes. This hints at a potential overfitting to the more easily segmented classes at the expense of the model's generalization capabilities.

Experiments 56 to 61 do not consistently surpass the segmentation accuracy of the model from experiments 53 - 55. While there are improvements in certain areas, such as the Cumuliform class in model 61, they do not present a clear overall advancement in segmentation performance.

Figure 25 presents a visual assessment of the Resnet34 model's performance in discerning distinct classes in sky images. The figure suggests that the Resnet34 model has a commendable degree of accuracy in identifying and segmenting the main cloud formations. The

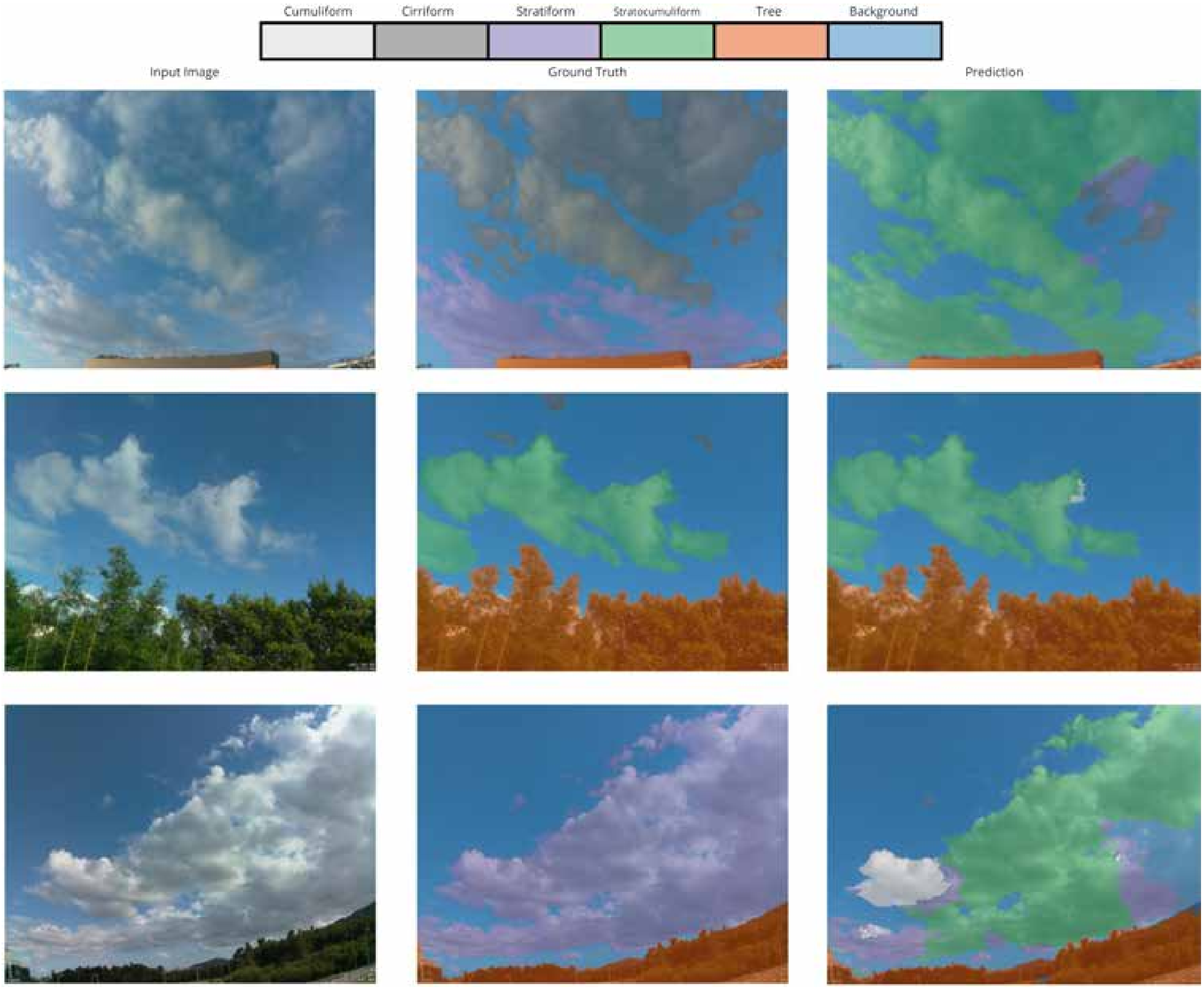


Figure 25: Experiment 53 sample results of inference over the test dataset

model's prediction of Cirriform clouds are not precise, which are thinner and more dispersed clouds, posing a challenge for accurate segmentation. The predictions for Stratiform and Stratocumuliform classes reveal some confusion, possibly due to their similar visual textures, which could result in misclassification. The model's interpretation of the 'Tree' class and its separation from the background are performatic. Although the differentiation is quite clear in the input images, the prediction panels show some inconsistencies, especially in the boundary areas where the tree line meets the sky.

The remaining results of the Resnet34 experiments are presented in Figures 44, 45, 46, 47, 48, 49, 50, 51, in the Appendix D section.

In conclusion, model 53 emerges as the best performer with its high accuracy and balanced metrics, challenging the earlier dominance of model 44. Model 53's results suggest that while the increased depth of the Resnet 34 backbone does not drastically outshine Resnet 18 in all aspects, it does offer benefits in certain configurations and classes. The comprehensive examination of models 10a to 15c underscores the importance of tailored configurations for each cloud class and highlights the potential for further nuanced improvements in cloud segmentation tasks. The quest for an optimal model must, therefore, consider both the nuanced needs of the specific cloud classes being segmented and the trade-offs between different performance metrics.

Experiment 62, utilizing the PP-LiteSeg architecture, presents a balanced performance with a notable IoU of 0.77 for the Cumuliform class, which is impressive compared to previous models. However, it demonstrates moderate performance in other cloud classes with an IoU of 0.56 for the Background class and a lower IoU for the Stratiform class. Despite its moderate IoU, this model achieves reasonable precision and recall values across most classes, indicating a fair balance between detecting the presence of clouds and delineating their boundaries. This suggests that while the model can generally distinguish between cloud types, there is room for improvement in classification accuracy.

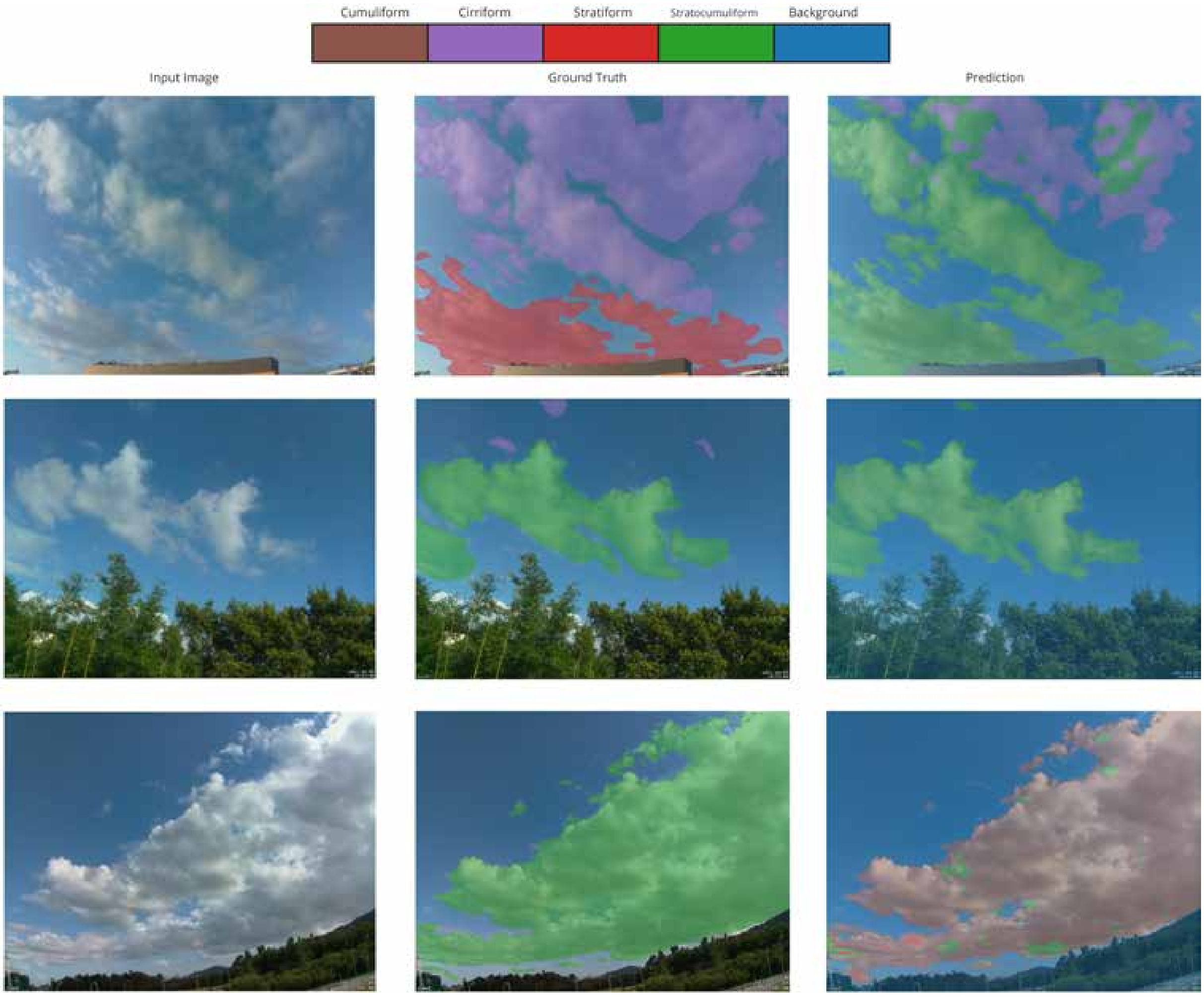


Figure 26: Experiment 62 sample results of inference over the test dataset

As illustrated in Figure 26, allows for the observation of the model's segmentation capabilities. In the predicted images, the model displays some proficiency in recognizing and segmenting Cirriform clouds, as indicated by the relatively accurate representation of their fluffy and dense structures. However, the model's ability to segment well Stratiform, seems to be less precise, as seen in the misclassified regions in the first row of predictions.

The model also appears to struggle with accurately delineating Stratocumuliform clouds, possibly due to their subtle textural differences, leading to instances of mislabeling this class with the Cumuliform class in the third row. This can be a common issue when dealing with classes that have low inter-class variability and high intra-class similarity.

Overall, the visual results presented in Figure 26 imply a solid foundational capability of the model in semantic segmentation tasks, with room for enhancement in class distinction and edge definition. Such qualitative insights, when supplemented with the missing quantitative metrics, could provide a more comprehensive understanding of the model's performance and inform further developmental strategies.

Experiment 63, employing the Segformer architecture, indicates a step forward, especially in terms of Background and Cumuliform classes with IoUs of 0.59 and 0.76, respectively, and impressive recall values across the board. The improvements in precision for the Stratocumuliform class and IoU for the Stratiform class suggest that this architecture is more capable of capturing finer details within these cloud types. Moreover, the improved precision in the Cirriform and Cumuliform classes, coupled with the highest recall of 0.54 in Cumuliform, reflects this model's enhanced ability to segment these complex structures.

The advancements in Experiment 63's Segformer architecture over the PP-LiteSeg architecture of Experiment 62 are evident in the ability to better differentiate between the cloud types, achieving more accurate segmentation, particularly for the Stratocumuliform and Stratiform classes. However, both Transformer-based models show that there are still challenges to overcome, especially when it comes to the complex textures of Cirriform clouds, where CNN-based models had already struggled.

Comparing the performance of the Transformer-based models to the earlier CNN-based models, it's clear that each architecture has its strengths and areas for improvement. While Transformer models provide competitive or superior performance in certain classes, such as the Background and Cumuliform, they do not universally outperform CNN-based models in all aspects of cloud segmentation.

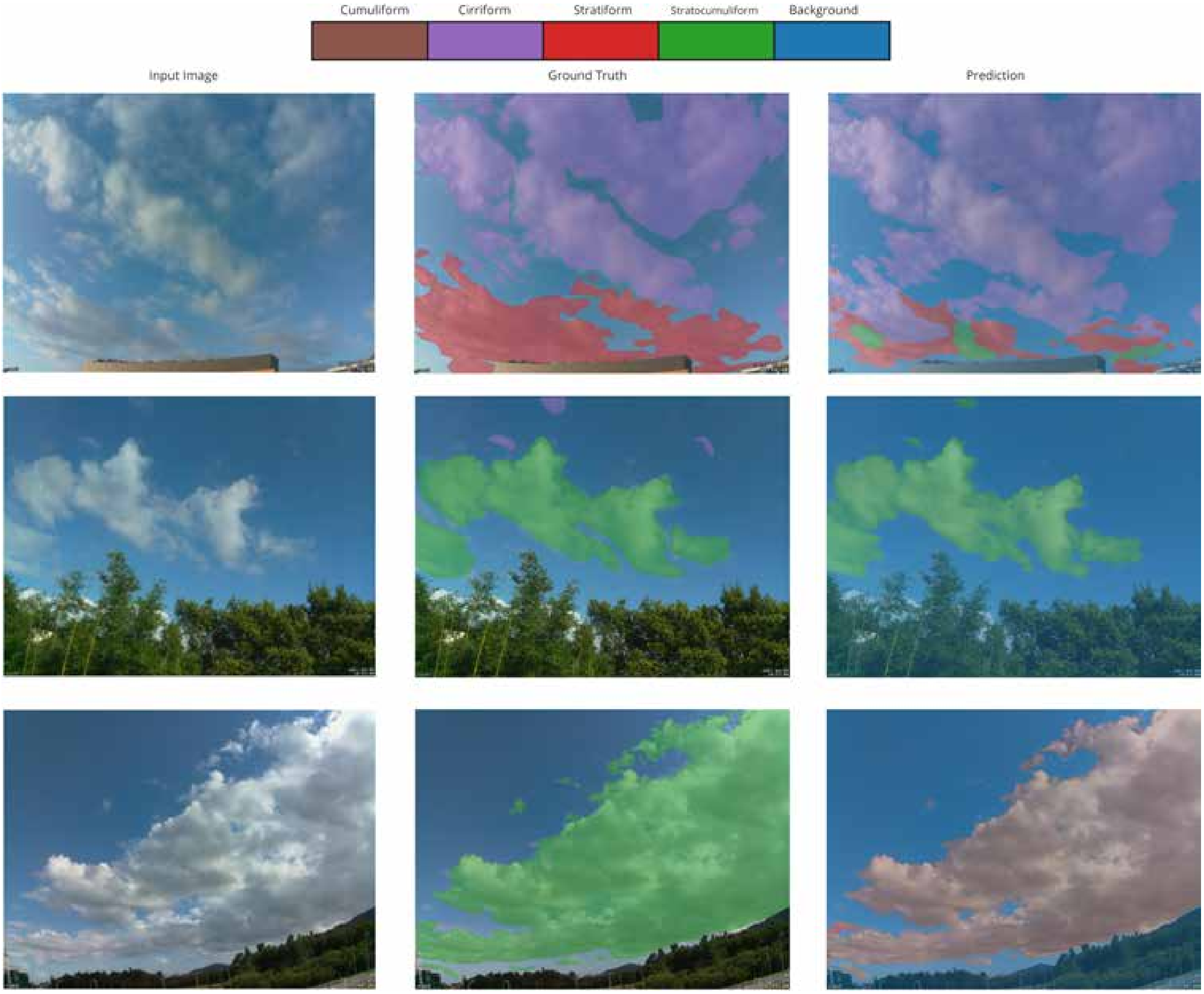


Figure 27: Experiment 63 sample results of inference over the test dataset

As depicted in Figure 27, one can see that the model is also quite adept at identifying Cirriform clouds, as it closely matches the ground truth with similar shapes and extents. This model also presents challenges with Stratocumuliform clouds, however, a slight improvement is indicated in the predicted area compared to experiment 62.

In conclusion, the performance of experiments 62 and 63 highlights the potential of Transformer-based architectures in the field of cloud segmentation. Experiment 63's Segformer model, in particular, with its balanced performance across all metrics and notable improvements in specific cloud classes, suggests a promising direction for further research and development. Nonetheless, it remains evident that the perfect model for cloud segmentation has not yet been realized, and the trade-offs between precision, recall, and IoU must continue to be considered. The complexity of accurately classifying cloud structures requires an ongoing effort to fine-tune these advanced architectures.

3.4.3 Discussion

Comparing study results with those in existing literature presents challenges, primarily due to the dataset's unique nature and cloud classification tasks' particularities. Specific conditions in the dataset significantly impact cloud classification model performance. Results can be influenced by variables like cloud type frequency, atmospheric conditions, and time of year. For example, a dataset with a high percentage of cirrus clouds, which are challenging to categorize, or highly turbid weather, could result in poorer performance metrics. Conversely, a dataset with mostly clear skies and recognizable cloud formations might produce better results. Without using the same dataset for evaluation, this variability complicates direct comparisons between studies.

This study differs from others in the field as horizon-oriented images were chosen for use. Many studies use images of the entire sky or specific sky areas. The chosen method provides more context than patch images while not offering a comprehensive 360-degree view like ASIs, making comparisons with other studies difficult. Additionally, significant differences in methodologies and performance metrics across studies further complicate comparisons.

In Table 15, Fabel et al. [(2022] achieved competitive performance in cloud layer classification using IP-SR* and DC** methods, with average IoU of 0.622 and 0.619, respectively. These results indicate these approaches' effectiveness in distinguishing cloud layers based on height. However, in contrast, Ye et al. [(2019] utilized a fine-grained algorithm and achieved an average IoU of 0.34 for classifying eight different cloud types in a dataset of 500 test images, which demonstrates the difficulties related to this problem.

With the developed Clouds-1000 experiments, the best model used a U-net architecture with ResNet18 (experiment 28) to classify four distinct cloud types, achieving a comparable average accuracy of 0.8564, with promising performance in terms of average precision, recall, and IoU. Now when comparing to the Clouds-1500 experiments and excluding the background class, the proposed methods show promising performance in terms of average precision, recall, and intersection over union. The comparative results shows that the best performing implementation in terms of evaluation metrics is Experiment 10a. This implementation has surpassed metric performance, with exception of AR, indicating an improvement in accurate classification and localization within images. Specifically, Experiment 10a achieved notable advancement, with a slightly simpler architecture, from corresponding scores of the Clouds-1000 best resulting model.

Table 15: Comparative analysis of results between this study and the literature in terms of class type, methodology, number of images used for testing, and performance metrics: Average Precision (AP), Average Recall (AR), and Average Intersection over Union (AIoU)

Study	Class Type	Methodology	Test Size	AP	AR	AIoU
Fabel et al. [(2022]	3 cloud layers	IP-SR*	154	0.779	0.751	0.622
Fabel et al. [(2022]	3 cloud layers	DC**	154	0.766	0.742	0.619
Ye et al. [(2019]	8 cloud types	fine-grained algo.	500	0.427	0.447	0.34
Clouds-1000	4 cloud types + tree	Unet + Resnet18 (#28)	200	0.694	0.686	0.564
Clouds-1500	4 cloud types + tree	U-net + EfficientNet b0 (#35)	223	0.328	0.378	0.404
Clouds-1500	4 cloud types + tree	HRNet (#40)	223	0.692	**0.698**	0.558
Clouds-1500	4 cloud types + tree	U-net + Resnet 18 (#44)	223	**0.742**	0.682	**0.594**
Clouds-1500	4 cloud types + tree	U-net + Resnet 34 (#53)	223	0.642	0.588	0.474
Clouds-1500	4 cloud types	PP-LiteSeg (#62)	223	0.527	0.48	0.345
Clouds-1500	4 cloud types	Segformer (#63)	223	0.555	0.547	0.39

Moreover, experiment 40 of the Clouds-1500 is noted as second-best overall, also surpassing results in terms of AR, with competitive metrics. This illustrates that refinements in methodology yield similar outcomes, although with a more complex implementation. Consistent enhancement in performance from Experiments 35 to 63, especially in 40 and 44, suggests an effective iterative process of experimentation and modification for the problem at hand.

The primary reason for selecting these references for comparison is both presented semantic segmentation results, which went beyond binary classification, and showcased good performance. It is essential to highlight distinctions in methodologies. In Fabel et al. [(2022], the focus was on cloud layer classes, using cloud height for classification. Although providing valuable insights into clouds' vertical distribution, it did not differentiate between cloud types within the same layer. In contrast, Ye et al. [(2019] employed a more comprehensive classification scheme, allowing for a detailed representation of cloud types and their characteristics. Since both Fabel et al. [(2022] and Ye et al. [(2019] compare their results with other studies, it establishes a precedent for comparative analysis. By following this approach, the comparison extends, and results can be evaluated in relation to additional relevant studies in the field.

By comparing the results against the two works, the aim was to evaluate the methodology's effectiveness thoroughly and identify improvement areas. The comparison highlights the importance of considering cloud layer distinctions and a diverse set of cloud classes in semantic segmentation tasks for comprehensive cloud-related phenomena analysis.

The conducted experiments reveal the intricate interactions between model architecture, data augmentation, and class specificity, highlighting the difficulties associated with cloud segmentation. The main theme of these findings is complexity, emphasizing the need for customized solutions to address the wide range of complex patterns that various cloud types present.

Data augmentation has emerged as a pivotal factor influencing model performance. The EfficientNet's variation in response to data augmentation (experiments 35 - 39) underscores its potential in enhancing robustness and generalization. Notably, experiment 35 highlighted the utility of data augmentation in improving results for specific classes. Conversely, aggressive data augmentation in the HRNet (experiment 43) led to suboptimal generalization, particularly for cloud segmentation. This disparity underscores the need for careful calibration of augmentation strategies in relation to the specificities of the task at hand.

EfficientNet and U-net, while promising, demonstrated limitations, especially in segmenting complex cloud types like Cirriform and Cumuliform clouds. Innovative approaches such as nuanced data augmentation, class-centric optimizations, or new architectural innovations may be necessary to advance in these areas.

The combination of HRNet with OCRNet showcased potential yet also indicated that no universal solution exists. The sensitivity of models to specific augmentations and the intricate balance between sensitivity and specificity call for meticulous fine-tuning in training procedures.

The U-net + Resnet configurations revealed that improvements in cloud segmentation do not uniformly arise from increased input size or model depth, particularly for challenging classes like Cirriform and Cumuliform clouds. When comparing results of experiments 44 to 52 with those of 53 to 61, several conclusions can be drawn. Firstly, there appears to be a complex interaction between input sizes and the model's ability to segment various cloud types, as larger input sizes do not uniformly translate to better performance. Secondly, the depth of the Resnet backbone—switching from 18 to 34 layers—does not consistently yield improvements, suggesting that other factors may play a more pivotal role in the segmentation task.

The Transformers models, particularly in experiments 62 and 63, segmented well the Background class due to their global contextual strengths. However, their performance on more complex cloud patterns did not surpass CNN-based models significantly. This outcome was unexpected, given the transformers' inherent capability to capture long-range dependencies which could potentially translate into a nuanced understanding of the spatial coherence necessary for cloud segmentation. Despite employing a *MixedLoss* function designed to enhance localization accuracy by promoting semantic continuity, the Transformer models did not exhibit a notable reduction in the localization problem. The persistence of localization issues in Experiments 16 and 17 suggests that while the Transformer models processed global contextual information effectively, they may lack in capturing the fine-grained textural details that are pivotal in distinguishing complex cloud formations. This points to a potential limitation within the Transformer architectures in their current form when applied to the specific nuances of cloud segmentation tasks. Such findings prompt further investigation into how these models can be adjusted or extended, perhaps through the integration of CNN-like mechanisms or additional specialized layers, to better handle the granular specifics of this domain.

The dataset employed in these experiments showed a notable overrepresentation of the Stratocumuliform class. This imbalance posed a challenge for model training and generalization, as models tend to perform better on classes with more examples. In an attempt to counteract this skew, data augmentation techniques were implemented to artificially enhance the variability and volume of the underrepresented classes. The lack of diversity in the predictions underscores potential issues within the models' training regimen, possibly necessitating a more balanced dataset or an adjustment in the learning algorithm to reduce the current bias towards the Stratocumuliform clouds. It is important to mention that a problem with some inference images for experiment 37 as denoted on Figure 30 was detected. This is the only model that presented this problem of horizontally flipping the inference masks and may suggest that there was data augmentation leakage to the test dataset, indicating that further analysis is necessary to understand the problem.

However, despite these efforts, the improvement in performance metrics for classes other than Stratocumuliform was not as significant as anticipated. Data augmentation, while beneficial in enhancing the robustness of models to variations in input data, did not adequately compensate for the inherent imbalance present in the dataset. This was particularly evident in the Cumuliform and Cirriform classes, which continued to present challenges for all models tested, including the leading ones from Experiments 40 and 44. The results suggest that while data augmentation is a powerful tool, it cannot fully mitigate the effects of class overrepresentation. This has important implications for the field of cloud classification, indicating a need for not only advanced model architectures and training techniques but also well-balanced datasets. Future directions may include further refining data augmentation strategies specifically for underrepresented classes or perhaps synthesizing new training examples through techniques such as Generative Adversarial Networks (GANs). It is also imperative to explore methods for effectively handling class imbalance, such as cost-sensitive learning, where the model penalizes misclassifications of the minority class more heavily during training. While experiments 40 and 42 present significant potential in cloud segmentation, especially for the Cumuliform and Background classes, experiment 41 reveals the difficulties in achieving consistent performance across different types of clouds. This mixed performance emphasizes the complexity of cloud segmentation tasks, where different architectures and configurations can lead to varying levels of success in accurately classifying and segmenting cloud structures. It underscores the necessity for a careful balance of model architecture, training procedures, and data augmentation techniques to achieve the best generalization and predictive performance across all cloud classes.

The inclusion of the Tree class in the dataset warrants a particular discussion. As indicated in Table 14, the Tree class consistently showed high Intersection over Union (IoU), Precision (Pr), and Recall values across multiple experiments, notably in Experiment 6 with an IoU of **0.97**, Pr of **0.99**, and Recall of **0.98**, and in Experiments 44 and 54 where the IoU remained above 0.88. The Tree class's distinct visual characteristics compared to cloud classes may have contributed to this consistency, providing the models with a simpler segmentation task that bolstered overall performance metrics.

However, while the Tree class showed high performance, this did not translate into improved segmentation for cloud classes, which are the primary focus of the study. The overrepresentation of the Stratocumuliform class, despite data augmentation efforts, did not yield significant improvements in the segmentation of other cloud types, particularly for the more complex Cirriform and Cumuliform classes. It raises the question of whether the presence of the Tree class within the dataset might have induced a bias in the models, allowing for high accuracy in an easily identifiable class while not necessarily contributing to the discernment required for more complex cloud segmentation tasks. The high performance in the Tree class suggests that models may be allocating more resources to effectively segment this class, potentially at the cost of neglecting the finer, more subtle features of cloud types. This effect can often lead to a model's overfitting to the majority class and underperforming on minority classes. In this case, the Tree class may be acting as a 'majority' class in terms of segmentation ease, overshadowing the more nuanced cloud classes that require more sophisticated pattern recognition capabilities. The results suggest a need for a careful review of class representation within training datasets, ensuring that the presence of one highly distinguishable class does not detract from the overall objective of balanced and equitable model performance across all classes.

Compared to the initial experiments performed, an improvement in overall metrics and the quality of the segmentation can be seen. The achieved results have raised questions about why a simpler model outperforms a more complex one, leading to the need for future investigations. Six potential causes were identified for further exploration: 1) Overfitting, as complex models with more parameters are prone to overfitting, while simpler models can generalize better; 2) Appropriate complexity, where the task of cloud segmentation may not be as complex for a machine learning model as initially thought; 3) Data availability, as complex models require more data to learn effectively, while simpler models may perform better with limited data; 4) Hyperparameter tuning, since complex models have more hyperparameters that need optimal tuning for optimal performance; 5) Regularization techniques like dropout, weight decay, or early stopping, which can prevent overfitting in complex models; and 6) Data quality, where a simpler model may be more robust against noisy data. These factors will be addressed in future works to gain further insights.

An important disclaimer is that no detailed analysis was performed to validate if the results were actually representing the clouds better than the ground truth, even though the dataset was created with the help of specialists there's always the probability of human error in determining a class type while annotating images.

4 Conclusion

The initial experiments showed that it is possible and feasible to classify clouds using current techniques of machine learning. Flat cameras pointing to the horizon allowed vertical distribution observation for cloud classification and avoided image distortion of fish-eye lenses whilst simplified image processing. When compared to the literature the results are positive overall, even for initial results with the first version of the developed dataset. The collective analysis of the experimental outcomes emphasizes the inherent complexity in cloud segmentation tasks and the consequent necessity for tailored model development and data handling methods. The presented findings show that no single model architecture or technique is uniformly effective across all scenarios. The variable impact of data augmentation and the nuance of class-specific model performance underscore the need for sophisticated, contextually aware approaches in model development for cloud segmentation.

This study has led to several noteworthy conclusions and future research directions. The methodology applied provided valuable insights into the complexities and nuances of cloud segmentation.

The initial hypothesis postulated that higher image resolution would facilitate more accurate cloud classification. However, the empirical evidence gathered through various experiments challenges this assumption. It was observed that simpler models, like the U-net with ResNet 18 architecture, achieved commendable results even with lower-resolution images. This finding suggests that the advantage conferred by high-resolution images may not be as significant as previously believed in the domain of cloud classification.

A critical observation is the relative performance of different model architectures. HRNet, while proficient in handling multi-resolution images and theoretically advantageous for detailed segmentation tasks, did not markedly surpass the simpler models in practice. This outcome highlights a crucial point in model selection: complexity does not necessarily equate to effectiveness, especially in cloud segmentation tasks.

Vision Transformers, offer extensive context-aware capabilities. Surprisingly, they did not demonstrate superiority over CNN-based models in cloud segmentation. This finding can be attributed to the inherent strengths of CNNs in texture and local context detection, which appear to be more aligned with the requirements of cloud segmentation tasks. Thus, we identify several potential avenues for future research. One particularly promising direction is the creation and examination of synthetic data emphasizing specific characteristics like texture and pattern repetition. This could shed light on model behaviors and provide insights into the factors influencing cloud classification performance. Continued experimentation with cloud tracking techniques is also recommended. Exploring simpler computer vision methods could be particularly fruitful in addressing localization issues observed in models like Detectron2. The potential of Vision Transformers in this domain remains to be fully explored, and further experiments are suggested to ascertain their utility in cloud segmentation.

The role of data augmentation emerged as a key theme in the study. Enhancing the dataset's variety, especially in terms of cloud types and atmospheric conditions, could address some of the class imbalance issues encountered and improve model robustness. Investigating the use of very low-resolution images for applications such as solar radiance nowcasting opens up new, efficient pathways for cloud classification methodologies.

One practical outcome of this study is the insight that simpler deep learning models can be effectively employed for real-world applications like solar radiance nowcasting. Developing a baseline model, incorporating the learning from this study, could enable its application in real-world scenarios, enhancing the predictive capabilities for solar energy generation.

Future work may involve the exploration of hybrid models that leverage both the local pattern recognition strengths of CNNs and the global context comprehension of Transformers. Additionally, the creation of more advanced data augmentation pipelines that more closely mirror the dynamic nature of cloud formations could further enhance model performance. Such processes will be instrumental in overcoming the complexities presented by the nuanced and varied classes of clouds. These findings confirm the importance of continuous experimentation in the field of image segmentation, as even small changes in methodology can lead to substantial improvements in model performance.

This research helped better understand which techniques work best for cloud segmentation. It is also evident that data imbalance is affecting the performance of all models developed. Overall, while conventional CNN models, particularly when combined with U-net, offered more reliable performance across various cloud types, they too require further enhancement for the more intricate Cirriform and Cumuliform cloud classes. The HRNet model looks more promising as it works with different resolutions, thus leading to a more refined segmentation, at the pixel level. Even though, some results seem to indicate that such an intricate model is not necessary in order to detect the most predominant clouds in the sky. A simpler model using U-net with Resnet 18 was able to achieve satisfactory results, using a much lower resolution.

This can be useful in the future since the main objective for the future is to use such models to predict cloud motion and forecast the impact it will have on solar power generation. For future studies, a better overall cloud classification model is recommended based on results presented in this research. In conclusion, this research has significantly contributed to the understanding of cloud segmentation and classification. The findings underscore the importance of model selection, the nuanced role of image resolution in cloud classification, and the potential of simpler models in this domain. The future research directions identified from this study not only pave the way for advanced experimentation in cloud segmentation but also hold promise for practical applications in solar energy forecasting.

For ease of reading, below is a summary of the main findings:

- Simpler model architectures often outperform more complex ones for cloud segmentation tasks.
- Higher resolution models may lead to over-segmentation due to the ground-truth annotations' precision limitations.
- The Tree class consistently shows high performance metrics, indicating a potential bias in model evaluation.
- Data augmentation is pivotal for model robustness but needs to be carefully tailored to avoid sub-optimal generalization.
- A trade-off exists between sensitivity to specific cloud classes and overall model performance across various types.
- Transformer models, despite their global contextual strengths, do not significantly surpass CNN-based models for complex cloud patterns.
- Overrepresentation of certain cloud classes in datasets presents challenges to model training and generalization.
- Detailed validation is required to ensure segmentation results represent clouds more accurately than ground truth.
- Future research should consider hybrid models, advanced data augmentation, and handling class imbalances effectively.

- Data augmentation, while beneficial, cannot fully mitigate the effects of class overrepresentation.
- The high performance of easily segmentable classes in the dataset may not translate to improved segmentation for complex cloud types.

References

Abadi, Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems, 2016.

D. Anagnostos, T. Schmidt, S. Cavadias, D. Soudris, J. Poortmans, and F. Catthoor. A method for detailed, short-term energy yield forecasting of photovoltaic installations. *Renewable Energy*, 130:122 – 129, (2019). ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2018.06.058. URL http://www.sciencedirect.com/science/article/pii/S0960148118307109.

E. Barrett and C. K. Grant. The identification of cloud types in landsat mss images. Technical report, 1976.

A. Chaurasia and E. Culurciello. Linknet: Exploiting encoder representations for efficient semantic segmentation. In *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017. doi: 10.1109/vcip.2017.8305148. URL http://dx.doi.org/10.1109/VCIP.2017.8305148.

F. Chollet et al. Keras. https://keras.io, 2015. Accessed: 2023-11-07.

O. D. de Meteorologia. Aspectos do clima de florianópolis., 1984.

L. R. do Nascimento, T. de Souza Viana, R. A. Campos, and R. Rüther. Extreme solar overirradiance events: Occurrence and impacts on utility-scale photovoltaic power plants in brazil. *Solar Energy*, 186:370–381, (2019). ISSN 0038-092X. doi: https://doi.org/10.1016/j.solener.2019.05.008. URL https://www.sciencedirect.com/science/article/pii/S0038092X19304530.

L. R. do Nascimento, M. Braga, R. A. Campos, H. F. Naspolini, and R. Rüther. Performance assessment of solar photovoltaic technologies under different climatic conditions in brazil. *Renewable Energy*, 146:1070–1082, (2020). ISSN 0960-1481. doi: https:

//doi.org/10.1016/j.renene.2019.06.160. URL https://www.sciencedirect.com/science/article/pii/S0960148119310006.

Y. Fabel, B. Nouri, S. Wilbert, N. Blum, R. Triebel, M. Hasenbalg, P. Kuhn, L. F. Zarzalejo, and R. Pitz-Paal. Applying self-supervised learning for semantic cloud segmentation of all-sky images. *Atmospheric Measurement Techniques*, 15(3):797–809, (2022). doi: 10.5194/amt-15-797-2022. URL https://amt.copernicus.org/articles/15/797/2022/.

Gaplan. *Atlas de Santa Catarina.* Gabinete de Planejamento e Coordenação Geral, 1986.

Y. HU and K. STAMNES. Climate sensitivity to cloud optical properties. *Tellus B*, 52(1):81–93, 2000. doi: https://doi.org/10.1034/j.1600-0889.2000.00993.x.

B. Juncklaus Martins, A. Cerentini, S. M. Neto, and A. von Wangenheim. Systematic literature review on forecasting/nowcasting based upon ground-based cloud imaging. 02 (2021). doi: 10.13140/RG.2.2.33598.00323.

B. Juncklaus Martins, A. Cerentini, S. M. Neto, and A. von Wangenheim. Systematic review of nowcasting approaches for solar energy production based upon ground-based cloud imaging. *Solar Energy Advances*, 10 (2022)a.

B. Juncklaus Martins, M. Polli, A. Cerentini, S. Mantelli, T. Chaves, N. Moreira Branco, A. von Wangenheim, and J. Arrais. Clouds-1000, 06 (2022)b. URL https://data.mendeley.com/datasets/4pw8vfsnpx/1.

P. Kumari and D. Toshniwal. Deep learning models for solar irradiance forecasting: A comprehensive review. *Journal of Cleaner Production*, 318:128566, (2021). ISSN 0959-6526. doi: https://doi.org/10.1016/j.jclepro.2021.128566. URL https://www.sciencedirect.com/science/article/pii/S0959652621027736.

W. Köppen and R. Geiger. *Klimate der Erde.* Justus Perthes, 1928.

P. Li, L. Dong, H. Xiao, and M. Xu. A cloud image detection method based on svm vector machine. *Neurocomputing*, 169:34 – 42, (2015). ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2014.09.102. URL http://www.sciencedirect.com/science/article/pii/S0925231215006864. Learning for Visual Semantic Understanding in Big Data ESANN 2014 Industrial Data Processing and Analysis.

T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, (2014).

T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection, (2017).

C. N. Long, J. M. Sabburg, J. Calbo, and J. D. Page. Retrieving cloud characteristics from ground-based daytime color all-sky images. *Journal of Atmospheric and Oceanic Technology*, 23:633–652, 5 2006. doi: http://dx.doi.org/10.1175/JTECH1875.1.

S. L. Mantelli, A. von Wangenheim, E. B. Pereira, and A. C. Sobieranki. Hierarchical color similarity metrics for step-wise application on sky monitoring surface cameras. *Earth and Space Science Open Archive*, page 25, (2020). doi: 10.1002/essoar.10503135.1. URL https://doi.org/10.1002/essoar.10503135.1.

S. L. Mantelli, A. v. Wangenhein, E. B. Pereira, and E. Comunello. The use of euclidean geometric distance on rgb color space for classification of sky and cloud patterns. *Journal of Atmospheric and Oceanic Technology*, 27(9):1504 – 1517, 2010. doi: 10.1175/2010JTECHA1353.1.

G. L. Martins, S. L. Mantelli, and R. Rüther. Evaluating the performance of radiometers for solar overirradiance events. *Solar Energy*, 231:47–56, (2022). ISSN 0038-092X. doi: https://doi.org/10.1016/j.solener.2021.11.050. URL https://www.sciencedirect.com/science/article/pii/S0038092X21010100.

F. A. Mejia, B. Kurtz, K. Murray, L. M. Hinkelman, M. Sengupta, Y. Xie, and J. Kleissl. Coupling sky images with radiative transfer models: a new method to estimate cloud optical depth. *Atmospheric Measurement Techniques*, 9(8):4151–4165, 2016. doi: 10.5194/amt-9-4151-2016. URL https://www.atmos-meas-tech.net/9/4151/2016/.

A. Mellit and S. A. Kalogirou. Artificial intelligence techniques for photovoltaic applications: A review. *Progress in Energy and Combustion Science*, 34(5):574 – 632, (2008). ISSN 0360-1285. doi: https://doi.org/10.1016/j.pecs.2008.01.001. URL http://www.sciencedirect.com/science/article/pii/S0360128508000026.

M. A. Monteiro. Caracterizacao climatica do estado de santa catarina: uma abordagem dos principais sistemas atmosfericos que atuam durante o ano. *Geosul*, 16(31):69–78, (2001). ISSN 0103-3964.

Q. Paletta and J. Lasenby. Convolutional neural networks applied to sky images for short-term solar irradiance forecasting, (2020).

S. Pelland, J. Remund, J. Kleissl, T. Oozeki, and K. De Brabandere. *Photovoltaic and Solar Forecasting: State of the Art*. 10 (2013). ISBN ISBN 978-3-906042-13-8.

M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099–3104 vol.4. IEEE, Oct. 2004. ISBN 0-7803-8566-7. doi: 10.1109/icsmc.2004.1400815. URL http://dx.doi.org/10.1109/icsmc.2004.1400815.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, (2015).

L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472, 2017. doi: 10.1109/WACV.2017.58.

M. P. Souza-Echer, E. B. Pereira, L. Bins, and M. A. R. 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. doi: http://dx.doi.org/10.1175/JTECH1833.1.

M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. (2019). doi: 10.48550/ARXIV.1905.11946. URL https://arxiv.org/abs/1905.11946.

B. Tarrojam, F. Mueller, J. D. Eichman, and S. Samuelsen. Metrics for evaluating the impacts of intermittent renewable generation on utility load-balancing. *Energy*, 42 (1):546 – 562, (2012). ISSN 0360-5442. doi: https://doi.org/10.1016/j.energy.2012.02.040. URL http://www.sciencedirect.com/science/article/pii/S0360544212001351. 8th World Energy System Conference, WESC 2010.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.

C. Voyant, G. Notton, S. Kalogirou, M. Nivet, C. Paoli, F. Motte, and A. Fouilloy. Machine learning methods for solar radiation forecasting: A review. *Renewable Energy*, 105:569–582, (2017). ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2016.12.095. URL https://www.sciencedirect.com/science/article/pii/S0960148116311648.

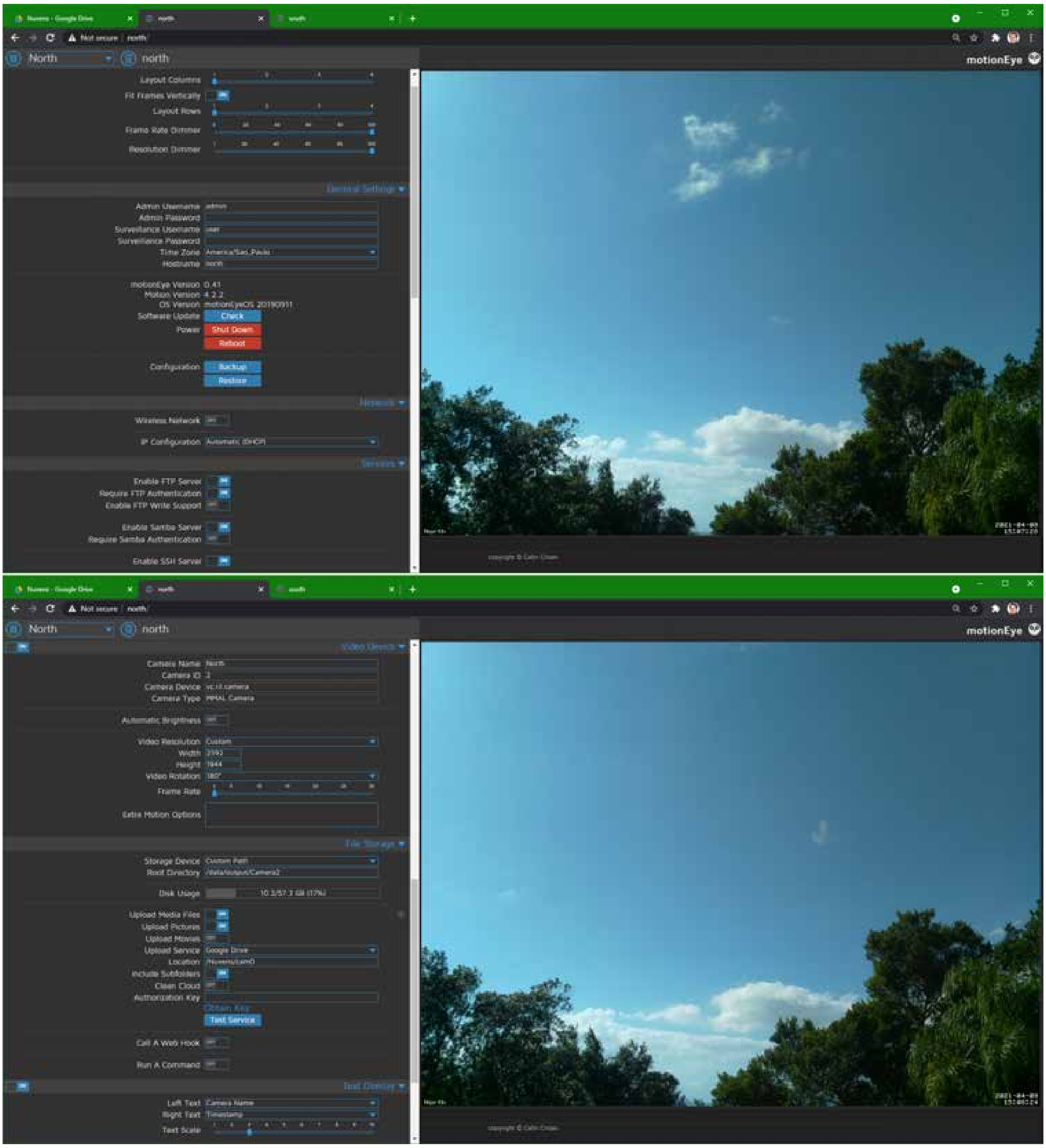
J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao. Deep high-resolution representation learning for visual recognition, 2020.

Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2, (2019). URL https://github.com/facebookresearch/detectron2.

L. Ye, Z. Cao, Y. Xiao, and Z. Yang. Supervised fine-grained cloud detection and recognition in whole-sky images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(10):7972–7985, (2019). doi: 10.1109/TGRS.2019.2917612.

A Appendix A: Nimbus Gazer System Screenshots

The picture below shows a Nimbus Gazer system screenshot, where all custom settings used in image capture can be seen.



Layout Columns 1
Fit Frames Vertically ON
Layout Rows 1
Frame Rate Dimmer 100
Resolution Dimmer 100

General Settings

Admin Username admin
Admin Password
Surveillance Username user
Surveillance Password
Time Zone America/Sao_Paulo
Hostname north

motionEye Version 0.41
Motion Version 4.2.2
OS Version motionEyeOS 20190911
Software Update Check
Power Shut Down Reboot
Configuration Backup Restore

Network

Wireless Network OFF
IP Configuration Automatic (DHCP)

Services

Enable FTP Server ON
Require FTP Authentication ON
Enable FTP Write Support OFF
Enable Samba Server ON
Require Samba Authentication OFF
Enable SSH Server ON

Video Device

Camera Name North
Camera ID 2
Camera Device vc.ril.camera
Camera Type MMAL Camera
Automatic Brightness OFF
Video Resolution Custom
Width 2592
Height 1944
Video Rotation 180°
Frame Rate 1
Extra Motion Options

File Storage

Storage Device Custom Path
Root Directory /data/output/Camera2
Disk Usage 10.2/57.3 GB (17%)
Upload Media Files ON
Upload Pictures ON
Upload Movies OFF
Upload Service Google Drive
Location /Nuvens/cam0
Include Subfolders ON
Clean Cloud OFF
Authorization Key
Obtain Key
Test Service
Call A Web Hook OFF
Run A Command OFF

Text Overlay

Left Text Camera Name
Right Text Timestamp
Text Scale 3

Figure 28: Screenshot of our custom configurations of the Nimbus Gazer system.

B Appendix B: Json-based Supervisely Annotation Format

Below is an annotation example for one of the dataset images, in Json-based supervised annotation format. The image corresponding to this annotation is the same as in the Appendix C.

```
{
    "description": "",
    "tags": [],
    "size": {
        "height": 1944,
        "width": 2592
    },
    "objects": [
        {
            "id": 818029968,
            "classId": 8907989,
            "description": "",
            "geometryType": "polygon",
            "labelerLogin": "unknown",
            "createdAt": "2021-08-03T17:45:49.547Z",
            "updatedAt": "2021-08-03T17:45:49.547Z",
            "tags": [],
            "classTitle": "Arvore",
            "points": {
                "exterior": [
            [1, 1229],[27,1238],[59,1265],[89,1321],[1⌋
             ↪  53,1362],[199,1392],[229,1364],[294,13⌋
             ↪  41],[319,1358],[340,1376],[370,1415],[⌋
             ↪  406,1438],[439,1466],[493,1491],[556,1⌋
             ↪  489],[597,1486],[659,1495],[687,1514],⌋
             ↪  [703,1532],[715,1560],[724,1583],[757,⌋
             ↪  1608],[788,1625],[800,1650],[825,1696]⌋
             ↪  ,[839,1735],[853,1770],[910,1730],[936⌋
             ↪  ,1712],[954,1712],[975,1733],[996,1760⌋
             ↪  ],[1007,1783],[1032,1816],[1062,1836],⌋
             ↪  [1101,1848],[1113,1869],[1129,1896],[1⌋
             ↪  164,1938],[1205,1943],[1260,1869],[126⌋
             ↪  7,1816],[1332,1818],[1428,1742],[1481,⌋
             ↪  1707],[1507,1657],[1523,1590],[1576,15⌋
             ↪  62],[1633,1579],[1679,1618],[1909,1503⌋
             ↪  ],[1882,1431],[1909,1401],[1896,1355],⌋
             ↪  [1877,1284],[1875,1231],[1903,1201],[1⌋
             ↪  946,1213],[1988,1192],[2002,1162],[204⌋
             ↪  3,1137],[2126,1123],[2167,1123],[2193,⌋
             ↪  1187],[2287,1173],[2393,1240],[2430,12⌋
             ↪  49],[2466,1222],[2462,1190],[2510,1208⌋
             ↪  ],[2542,1178],[2533,1146],[2570,1143],⌋
             ↪  [2586,1125],[2588,1097],[2591,1943],[0⌋
             ↪  ,1943]
        ],
        "interior": []
    }
},
{
    "id": 818029967,
    "classId": 8907990,
    "description": "",
            "geometryType": "polygon",
            "labelerLogin": "unknown",
            "createdAt": "2021-08-03T17:45:49.547Z",
            "updatedAt": "2021-08-03T17:45:49.547Z",
            "tags": [],
            "classTitle": "Estratocumuliformes",
            "points": {
                "exterior": [
                    [2154, 490],[2107,532],[2071,567],[2052,589],[2084,620],[2159,599],[2169,620],[2247,643],[2285,596],[2313,596],[2352,615],[2394,632],[2432,605],[2472,567],[2479,527],[2449,504],[2432,497],[2430,466],[2434,420],[2403,383],[2365,387],[2332,404],[2313,426],[2288,437],[2264,442],[2249,463],[2223,483],[2200,490]
                ],
                "interior": []
            }
        },
        {
            "id": 818029966,
            "classId": 8907990,
            "description": "",
            "geometryType": "polygon",
            "labelerLogin": "unknown",
            "createdAt": "2021-08-03T17:45:49.547Z",
            "updatedAt": "2021-08-03T17:45:49.547Z",
            "tags": [],
            "classTitle": "Estratocumuliformes",
            "points": {
                "exterior": [
                    [1725, 297],[1700,311],[1690,357],[1702,40⌋
                     ↪  1],[1742,430],[1835,404],[1851,359],[1⌋
                     ↪  844,328],[1828,311],[1828,295],[1849,2⌋
                     ↪  92],[1868,288],[1886,292],[1915,316],[⌋
                     ↪  1934,340],[1956,369],[1975,383],[2002,⌋
                     ↪  390],[2020,385],[2010,354],[2001,333],⌋
                     ↪  [2027,297],[2038,269],[2024,262],[1993⌋
                     ↪  ,267],[1955,259],[1927,252],[1889,245]⌋
                     ↪  ,[1849,250],[1801,260],[1782,276]
                ],
                "interior": []
            }
        },
        {
            "id": 818029965,
            "classId": 8907990,
            "description": "",
            "geometryType": "polygon",
            "labelerLogin": "unknown",
            "createdAt": "2021-08-03T17:45:49.547Z",
            "updatedAt": "2021-08-03T17:45:49.547Z",
            "tags": [],
            "classTitle": "Estratocumuliformes",
            "points": {
                "exterior": [
                    [2069, 369],[2048,378],[2036,414],[2036,43⌋
                     ↪  2],[2074,452],[2122,452],[2153,468],[2⌋
                     ↪  188,454],[2199,440],[2243,430],[2259,4⌋
                     ↪  25],[2282,423],[2328,401],[2326,368],[⌋
                     ↪  2290,361],[2247,375],[2205,387],[2162,⌋
                     ↪  397],[2119,409],[2096,413]
                ],
                "interior": []
            }
        },
        {
            "id": 818029964,
            "classId": 8907990,
            "description": "",
            "geometryType": "polygon",
            "labelerLogin": "unknown",
            "createdAt": "2021-08-03T17:45:49.547Z",
            "updatedAt": "2021-08-03T17:45:49.547Z",
            "tags": [],
            "classTitle": "Estratocumuliformes",
            "points": {
                "exterior": [
                    [428, 1409],[487,1376],[536,1377],[589,137⌋
                    ↪  6],[707,1365],[807,1343],[875,1336],[9⌋
                    ↪  04,1332],[978,1291],[1029,1263],[1108,⌋
                    ↪  1230],[1170,1198],[1231,1168],[1246,11⌋
                    ↪  91],[1255,1215],[1264,1239],[1290,1253⌋
                    ↪  ],[1317,1265],[1348,1289],[1347,1329],⌋
                    ↪  [1319,1370],[1331,1402],[1362,1407],[1⌋
                    ↪  395,1403],[1418,1388],[1457,1367],[148⌋
                    ↪  2,1365],[1495,1374],[1490,1407],[1506,⌋
                    ↪  1438],[1530,1436],[1528,1400],[1563,13⌋
                    ↪  89],[1582,1402],[1625,1410],[1641,1412⌋
                    ↪  ],[1599,1438],[1579,1457],[1589,1479],⌋
                    ↪  [1618,1486],[1641,1509],[1636,1531],[1⌋
                    ↪  601,1554],[1554,1562],[1530,1583],[151⌋
                    ↪  1,1626],[1501,1673],[1480,1692],[1451,⌋
                    ↪  1723],[1400,1744],[1374,1758],[1350,17⌋
                    ↪  77],[1319,1806],[1298,1808],[1276,1808⌋
                    ↪  ],[1250,1846],[1250,1860],[1241,1882],⌋
                    ↪  [1200,1924],[1164,1908],[1144,1874],[1⌋
                    ↪  127,1851],[1087,1836],[1054,1822],[102⌋
                    ↪  7,1801],[1011,1773],[1008,1723],[975,1⌋
                    ↪  701],[935,1694],[916,1704],[873,1713],⌋
                    ↪  [845,1723],[828,1708],[819,1687],[805,⌋
                    ↪  1647],[795,1621],[773,1595],[750,1580]⌋
                    ↪  ,[736,1562],[719,1524],[691,1486],[657⌋
                    ↪  ,1476],[614,1472],[572,1472],[513,1472⌋
                    ↪  ],[484,1472],[437,1462],[411,1448]
                ],
                "interior": []
            }
        },
        {
            "id": 818029963,
            "classId": 8907990,
            "description": "",
            "geometryType": "polygon",
            "labelerLogin": "unknown",
            "createdAt": "2021-08-03T17:45:49.547Z",
            "updatedAt": "2021-08-03T17:45:49.547Z",
            "tags": [],
            "classTitle": "Estratocumuliformes",
            "points": {
                "exterior": [
            [1686, 1599],[1651,1572],[1638,1540],[1646
             ↪  ,1498],[1667,1456],[1691,1418],[1709,1
             ↪  409],[1739,1390],[1796,1349],[1815,132
             ↪  7],[1828,1291],[1847,1247],[1859,1224]
             ↪  ,[1878,1194],[1904,1158],[1920,1129],[
             ↪  1937,1093],[1964,1060],[2009,1032],[20
             ↪  62,1009],[2140,1000],[2175,1009],[2241
             ↪  ,1026],[2275,1028],[2285,996],[2315,99
             ↪  6],[2352,1009],[2397,1024],[2460,1040]
             ↪  ,[2509,1023],[2553,1015],[2586,1024],[
             ↪  2591,1042],[2591,1068],[2591,1089],[25
             ↪  91,1114],[2557,1139],[2532,1146],[2525
             ↪  ,1169],[2517,1192],[2498,1194],[2458,1
             ↪  194],[2430,1220],[2384,1217],[2352,119
             ↪  8],[2296,1169],[2274,1167],[2226,1169]
             ↪  ,[2201,1163],[2173,1131],[2125,1123],[
             ↪  2102,1125],[2017,1140],[1998,1158],[19
             ↪  84,1180],[1959,1199],[1924,1205],[1906
             ↪  ,1213],[1889,1232],[1878,1255],[1880,1
             ↪  283],[1887,1306],[1887,1331],[1891,135
             ↪  5],[1899,1374],[1910,1399],[1891,1416]
             ↪  ,[1885,1433],[1887,1449],[1897,1468],[
             ↪  1899,1487],[1885,1511],[1857,1527],[18
             ↪  26,1542],[1771,1568],[1749,1578],[1724
             ↪  ,1597]
        ],
        "interior": []
    }
},
{
    "id": 818029962,
    "classId": 8907990,
        "description": "",
        "geometryType": "polygon",
        "labelerLogin": "unknown",
        "createdAt": "2021-08-03T17:45:49.547Z",
        "updatedAt": "2021-08-03T17:45:49.547Z",
        "tags": [],
        "classTitle": "Estratocumuliformes",
        "points": {
            "exterior": [
                [363, 1371],[407,1371],[460,1366],[515,136⌋
                 ↪  3],[545,1345],[581,1325],[630,1333],[6⌋
                 ↪  97,1334],[757,1338],[814,1303],[831,12⌋
                 ↪  61],[852,1218],[839,1155],[823,1154],[⌋
                 ↪  763,1180],[708,1202],[685,1206],[671,1⌋
                 ↪  206],[668,1174],[672,1136],[674,1110],⌋
                 ↪  [716,1073],[749,1028],[716,979],[653,9⌋
                 ↪  84],[561,1026],[498,1034],[452,1025],[⌋
                 ↪  436,1034],[465,1075],[504,1099],[501,1⌋
                 ↪  157],[503,1169],[488,1196],[471,1217],⌋
                 ↪  [451,1237],[433,1234],[396,1251],[358,⌋
                 ↪  1273],[315,1303],[308,1327],[326,1366]
            ],
            "interior": []
        }
    },
    {
        "id": 818029961,
        "classId": 8907990,
        "description": "",
        "geometryType": "polygon",
        "labelerLogin": "unknown",
        "createdAt": "2021-08-03T17:45:49.547Z",
            "updatedAt": "2021-08-03T17:45:49.547Z",
            "tags": [],
            "classTitle": "Estratocumuliformes",
            "points": {
                "exterior": [
                    [164, 1330],[194,1298],[242,1276],[297,125⌋
                     ↪  3],[347,1237],[411,1204],[436,1169],[4⌋
                     ↪  24,1116],[330,1100],[251,1100],[183,10⌋
                     ↪  81],[179,1050],[227,1012],[213,984],[1⌋
                     ↪  85,927],[147,915],[113,927],[69,957],[⌋
                     ↪  32,989],[0,996],[0,1221]
                ],
                "interior": []
            }
        },
        {
            "id": 818029960,
            "classId": 8907990,
            "description": "",
            "geometryType": "polygon",
            "labelerLogin": "unknown",
            "createdAt": "2021-08-03T17:45:49.547Z",
            "updatedAt": "2021-08-03T17:45:49.547Z",
            "tags": [],
            "classTitle": "Estratocumuliformes",
            "points": {
                "exterior": [
            [1839, 976],[1899,956],[1945,956],[1981,97⌋
             ↪  9],[1992,1003],[2070,1003],[2128,999],⌋
             ↪  [2167,982],[2184,942],[2124,945],[2071⌋
             ↪  ,960],[2071,935],[2061,903],[1988,903]⌋
             ↪  ,[1925,902],[1879,899],[1832,939],[182⌋
             ↪  8,950]
        ],
        "interior": []
    }
},
{
    "id": 818029959,
    "classId": 8907990,
    "description": "",
    "geometryType": "polygon",
    "labelerLogin": "unknown",
    "createdAt": "2021-08-03T17:45:49.547Z",
    "updatedAt": "2021-08-03T17:45:49.547Z",
    "tags": [],
    "classTitle": "Estratocumuliformes",
    "points": {
        "exterior": [
            [2212,
             ↪  917],[2231,892],[2254,873],[2267,879],⌋
             ↪  [2315,896],[2335,889],[2345,906],[2314⌋
             ↪  ,935],[2285,962],[2224,947],[2198,947]
        ],
        "interior": []
    }
},
{
    "id": 818029958,
            "classId": 8907990,
            "description": "",
            "geometryType": "polygon",
            "labelerLogin": "unknown",
            "createdAt": "2021-08-03T17:45:49.547Z",
            "updatedAt": "2021-08-03T17:45:49.547Z",
            "tags": [],
            "classTitle": "Estratocumuliformes",
            "points": {
                "exterior": [
                    [2425,
                     ↪   872],[2378,906],[2375,945],[2467,949],⌋
                     ↪   [2463,927],[2524,949],[2535,920],[2548⌋
                     ↪   ,912],[2587,895],[2520,867],[2464,862]
                ],
                "interior": []
            }
        }
    ]
}
```

C Appendix C: COCO Annotation format

Below is an annotation example for one of the dataset images, in COCO format. The image corresponding to this annotation is the same as in the Appendix B.

```
{
    "annotations": [
        {
            "area": 2191986,
            "bbox": [
                0,1097,2591,846
            ],
            "category_id": 1,
            "id": 0,
            "image_id": 0,
            "iscrowd": 0,
            "segmentation": [
                [
                1,1229,27,1238,59,1265,89,1321,153,1362,19⌋
                 ↪  9,1392,229,1364,294,1341,319,1358,340,⌋
                 ↪  1376,370,1415,406,1438,439,1466,493,14⌋
                 ↪  91,556,1489,597,1486,659,1495,687,1514⌋
                 ↪  ,703,1532,715,1560,724,1583,757,1608,7⌋
                 ↪  88,1625,800,1650,825,1696,839,1735,853⌋
                 ↪  ,1770,910,1730,936,1712,954,1712,975,1⌋
                 ↪  733,996,1760,1007,1783,1032,1816,1062,⌋
                 ↪  1836,1101,1848,1113,1869,1129,1896,116⌋
                 ↪  4,1938,1205,1943,1260,1869,1267,1816,1⌋
                 ↪  332,1818,1428,1742,1481,1707,1507,1657⌋
                 ↪  ,1523,1590,1576,1562,1633,1579,1679,16⌋
                 ↪  18,1909,1503,1882,1431,1909,1401,1896,⌋
                 ↪  1355,1877,1284,1875,1231,1903,1201,194⌋
                 ↪  6,1213,1988,1192,2002,1162,2043,1137,2⌋
                 ↪  126,1123,2167,1123,2193,1187,2287,1173⌋
                 ↪  ,2393,1240,2430,1249,2466,1222,2462,11⌋
                 ↪  90,2510,1208,2542,1178,2533,1146,2570,⌋
                 ↪  1143,2586,1125,2588,1097,2591,1943,0,1⌋
                 ↪  943
            ]
        ]
    },
    {
        "area": 111020,
        "bbox": [
            2052,383,427,260
        ],
        "category_id": 2,
        "id": 1,
        "image_id": 0,
        "iscrowd": 0,
            "segmentation": [
                [
                    2154,490,2107,532,2071,567,2052,589,2084,6⌋
                     ↪   20,2159,599,2169,620,2247,643,2285,596⌋
                     ↪   ,2313,596,2352,615,2394,632,2432,605,2⌋
                     ↪   472,567,2479,527,2449,504,2432,497,243⌋
                     ↪   0,466,2434,420,2403,383,2365,387,2332,⌋
                     ↪   404,2313,426,2288,437,2264,442,2249,46⌋
                     ↪   3,2223,483,2200,490
                ]
            ]
        },
        {
            "area": 64380,
            "bbox": [
                1690,245,348,185
            ],
            "category_id": 2,
            "id": 2,
            "image_id": 0,
            "iscrowd": 0,
            "segmentation": [
                [
                    1725,297,1700,311,1690,357,1702,401,1742,4⌋
                     ↪   30,1835,404,1851,359,1844,328,1828,311⌋
                     ↪   ,1828,295,1849,292,1868,288,1886,292,1⌋
                     ↪   915,316,1934,340,1956,369,1975,383,200⌋
                     ↪   2,390,2020,385,2010,354,2001,333,2027,⌋
                     ↪   297,2038,269,2024,262,1993,267,1955,25⌋
                     ↪   9,1927,252,1889,245,1849,250,1801,260,⌋
                     ↪   1782,276
                ]
            ]
        },
        {
            "area": 31244,
            "bbox": [
                2036,361,292,107
            ],
            "category_id": 2,
            "id": 3,
            "image_id": 0,
            "iscrowd": 0,
            "segmentation": [
                [
                    2069,369,2048,378,2036,414,2036,432,2074,4⌋
                     ↪   52,2122,452,2153,468,2188,454,2199,440⌋
                     ↪   ,2243,430,2259,425,2282,423,2328,401,2⌋
                     ↪   326,368,2290,361,2247,375,2205,387,216⌋
                     ↪   2,397,2119,409,2096,413
                ]
            ]
        },
        {
            "area": 929880,
            "bbox": [
                411,
                1168,
                1230,
                756
            ],
            "category_id": 2,
            "id": 4,
            "image_id": 0,
            "iscrowd": 0,
            "segmentation": [
                [
                    428,1409,487,1376,536,1377,589,1376,707,13⌋
                     ↪  65,807,1343,875,1336,904,1332,978,1291⌋
                     ↪  ,1029,1263,1108,1230,1170,1198,1231,11⌋
                     ↪  68,1246,1191,1255,1215,1264,1239,1290,⌋
                     ↪  1253,1317,1265,1348,1289,1347,1329,131⌋
                     ↪  9,1370,1331,1402,1362,1407,1395,1403,1⌋
                     ↪  418,1388,1457,1367,1482,1365,1495,1374⌋
                     ↪  ,1490,1407,1506,1438,1530,1436,1528,14⌋
                     ↪  00,1563,1389,1582,1402,1625,1410,1641,⌋
                     ↪  1412,1599,1438,1579,1457,1589,1479,161⌋
                     ↪  8,1486,1641,1509,1636,1531,1601,1554,1⌋
                     ↪  554,1562,1530,1583,1511,1626,1501,1673⌋
                     ↪  ,1480,1692,1451,1723,1400,1744,1374,17⌋
                     ↪  58,1350,1777,1319,1806,1298,1808,1276,⌋
                     ↪  1808,1250,1846,1250,1860,1241,1882,120⌋
                     ↪  0,1924,1164,1908,1144,1874,1127,1851,1⌋
                     ↪  087,1836,1054,1822,1027,1801,1011,1773⌋
                     ↪  ,1008,1723,975,1701,935,1694,916,1704,⌋
                     ↪  873,1713,845,1723,828,1708,819,1687,80⌋
                     ↪  5,1647,795,1621,773,1595,750,1580,736,⌋
                     ↪  1562,719,1524,691,1486,657,1476,614,14⌋
                     ↪  72,572,1472,513,1472,484,1472,437,1462⌋
                     ↪  ,411,1448
                ]
            ]
        },
        {
            "area": 574659,
            "bbox": [
            1638,996,953,603
        ],
        "category_id": 2,
        "id": 5,
        "image_id": 0,
        "iscrowd": 0,
        "segmentation": [
            [
                1686,1599,1651,1572,1638,1540,1646,1498,16⌋
                 ↪  67,1456,1691,1418,1709,1409,1739,1390,⌋
                 ↪  1796,1349,1815,1327,1828,1291,1847,124⌋
                 ↪  7,1859,1224,1878,1194,1904,1158,1920,1⌋
                 ↪  129,1937,1093,1964,1060,2009,1032,2062⌋
                 ↪  ,1009,2140,1000,2175,1009,2241,1026,22⌋
                 ↪  75,1028,2285,996,2315,996,2352,1009,23⌋
                 ↪  97,1024,2460,1040,2509,1023,2553,1015,⌋
                 ↪  2586,1024,2591,1042,2591,1068,2591,108⌋
                 ↪  9,2591,1114,2557,1139,2532,1146,2525,1⌋
                 ↪  169,2517,1192,2498,1194,2458,1194,2430⌋
                 ↪  ,1220,2384,1217,2352,1198,2296,1169,22⌋
                 ↪  74,1167,2226,1169,2201,1163,2173,1131,⌋
                 ↪  2125,1123,2102,1125,2017,1140,1998,115⌋
                 ↪  8,1984,1180,1959,1199,1924,1205,1906,1⌋
                 ↪  213,1889,1232,1878,1255,1880,1283,1887⌋
                 ↪  ,1306,1887,1331,1891,1355,1899,1374,19⌋
                 ↪  10,1399,1891,1416,1885,1433,1887,1449,⌋
                 ↪  1897,1468,1899,1487,1885,1511,1857,152⌋
                 ↪  7,1826,1542,1771,1568,1749,1578,1724,1⌋
                 ↪  597
            ]
        ]
    },
        {
            "area": 213248,
            "bbox": [
                308,979,544,392
            ],
            "category_id": 2,
            "id": 6,
            "image_id": 0,
            "iscrowd": 0,
            "segmentation": [
                [
                    363,1371,407,1371,460,1366,515,1363,545,13⌋
                     ↪   45,581,1325,630,1333,697,1334,757,1338⌋
                     ↪   ,814,1303,831,1261,852,1218,839,1155,8⌋
                     ↪   23,1154,763,1180,708,1202,685,1206,671⌋
                     ↪   ,1206,668,1174,672,1136,674,1110,716,1⌋
                     ↪   073,749,1028,716,979,653,984,561,1026,⌋
                     ↪   498,1034,452,1025,436,1034,465,1075,50⌋
                     ↪   4,1099,501,1157,503,1169,488,1196,471,⌋
                     ↪   1217,451,1237,433,1234,396,1251,358,12⌋
                     ↪   73,315,1303,308,1327,326,1366
                ]
            ]
        },
        {
            "area": 180940,
            "bbox": [
                0,915,436,415
            ],
            "category_id": 2,
            "id": 7,
            "image_id": 0,
            "iscrowd": 0,
            "segmentation": [
                [
                    164,1330,194,1298,242,1276,297,1253,347,12⌋
                     ↪  37,411,1204,436,1169,424,1116,330,1100⌋
                     ↪  ,251,1100,183,1081,179,1050,227,1012,2⌋
                     ↪  13,984,185,927,147,915,113,927,69,957,⌋
                     ↪  32,989,0,996,0,1221
                ]
            ]
        },
        {
            "area": 37024,
            "bbox": [
                1828,899,356,104
            ],
            "category_id": 2,
            "id": 8,
            "image_id": 0,
            "iscrowd": 0,
            "segmentation": [
                [
                    1839,976,1899,956,1945,956,1981,979,1992,1⌋
                     ↪  003,2070,1003,2128,999,2167,982,2184,9⌋
                     ↪  42,2124,945,2071,960,2071,935,2061,903⌋
                     ↪  ,1988,903,1925,902,1879,899,1832,939,1⌋
                     ↪  828,950
                ]
            ]
        },
        {
            "area": 13083,
        "bbox": [
            2198,873,147,89
        ],
        "category_id": 2,
        "id": 9,
        "image_id": 0,
        "iscrowd": 0,
        "segmentation": [
            [
                2212,917,2231,892,2254,873,2267,879,2315,8⌋
                 ↪   96,2335,889,2345,906,2314,935,2285,962⌋
                 ↪   ,2224,947,2198,947
            ]
        ]
    },
    {
        "area": 18444,
        "bbox": [
            2375,862,212,87
        ],
        "category_id": 2,
        "id": 10,
        "image_id": 0,
        "iscrowd": 0,
        "segmentation": [
            [
                2425,872,2378,906,2375,945,2467,949,2463,9⌋
                 ↪   27,2524,949,2535,920,2548,912,2587,895⌋
                 ↪   ,2520,867,2464,862
            ]
        ]
    }
    ],
    "categories": [
        {
            "id": 0,
            "name": "Clouds",
            "supercategory": "none"
        },
        {
            "id": 1,
            "name": "Arvore",
            "supercategory": "Clouds"
        },
        {
            "id": 2,
            "name": "Estratocumuliformes",
            "supercategory": "Clouds"
        }
    ],
    "images": [
        {
            "date_captured": "2022-12-12T19:14:15+00:00",
            "file_name": "2021-04-09-10-00_jpg.rf.4e9d3b485fe32b9325a2f491f5dfff7a.jpg",
            "height": 1944,
            "id": 0,
            "license": 1,
            "width": 2592
        }
    ],
    "info": {
        "contributor": "",
        "date_created": "2022-12-12T19:14:15+00:00",
        "description": "Exported from roboflow.ai",
        "url": "https://public.roboflow.ai/object-detection/undefined",
        "version": "1",
        "year": "2022"
    },
    "licenses": [
        {
            "id": 1,
            "name": "Public Domain",
            "url": "https://creativecommons.org/publicdomain/zero/1.0/"
        }
    ]
}
```

D Appendix D: Additional Result Images

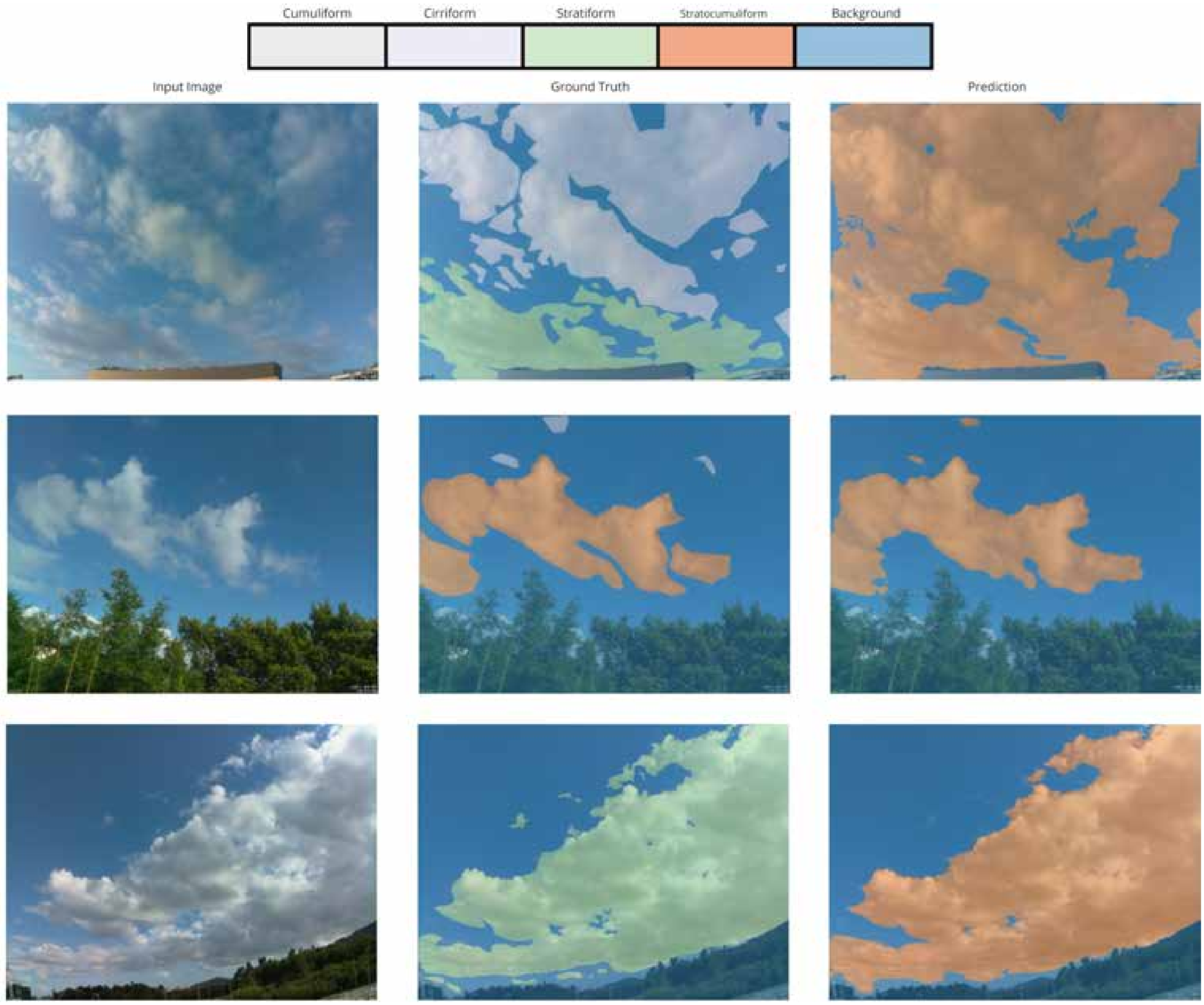


Figure 29: Experiment 36 sample results

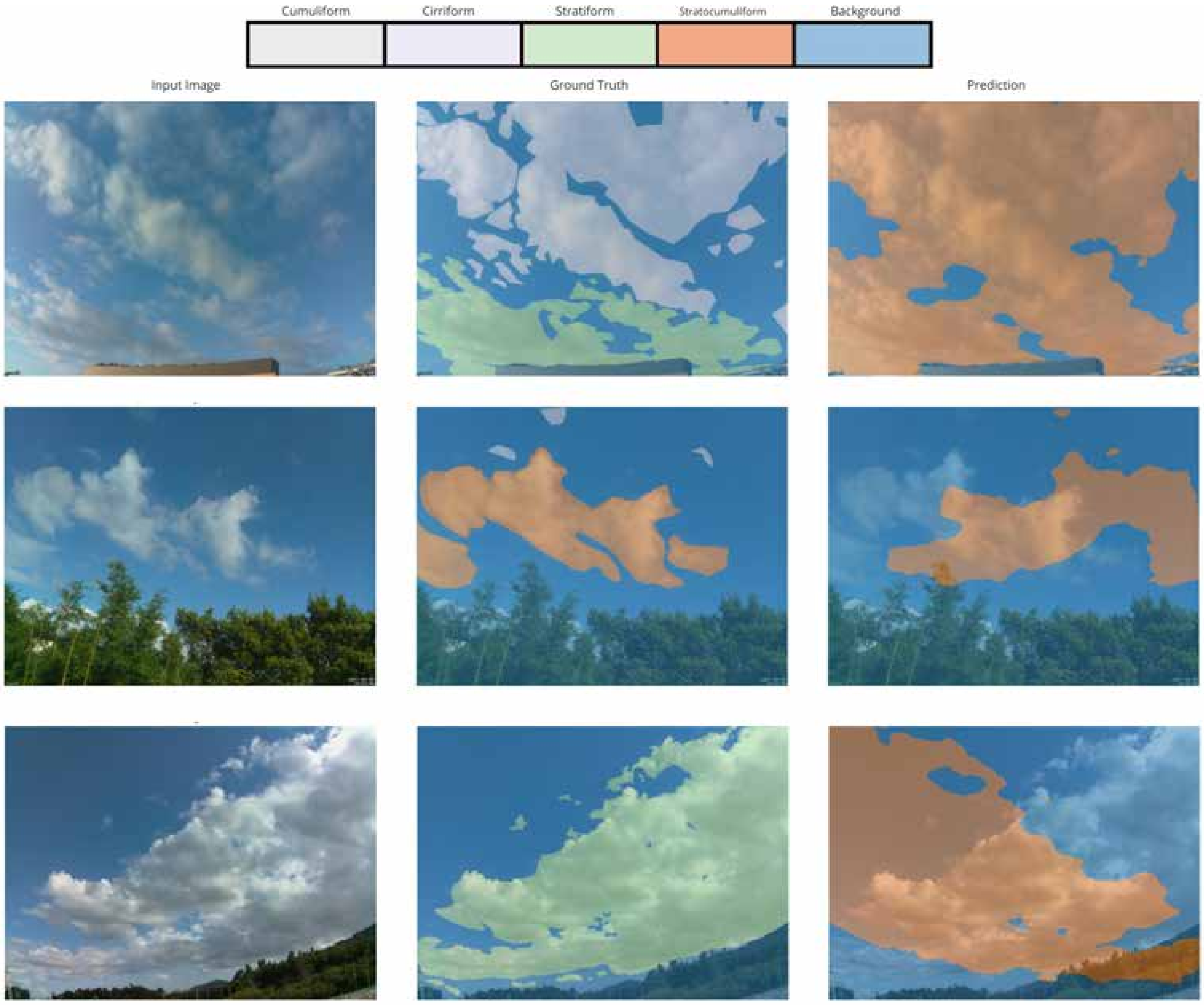


Figure 30: Experiment 37 sample results

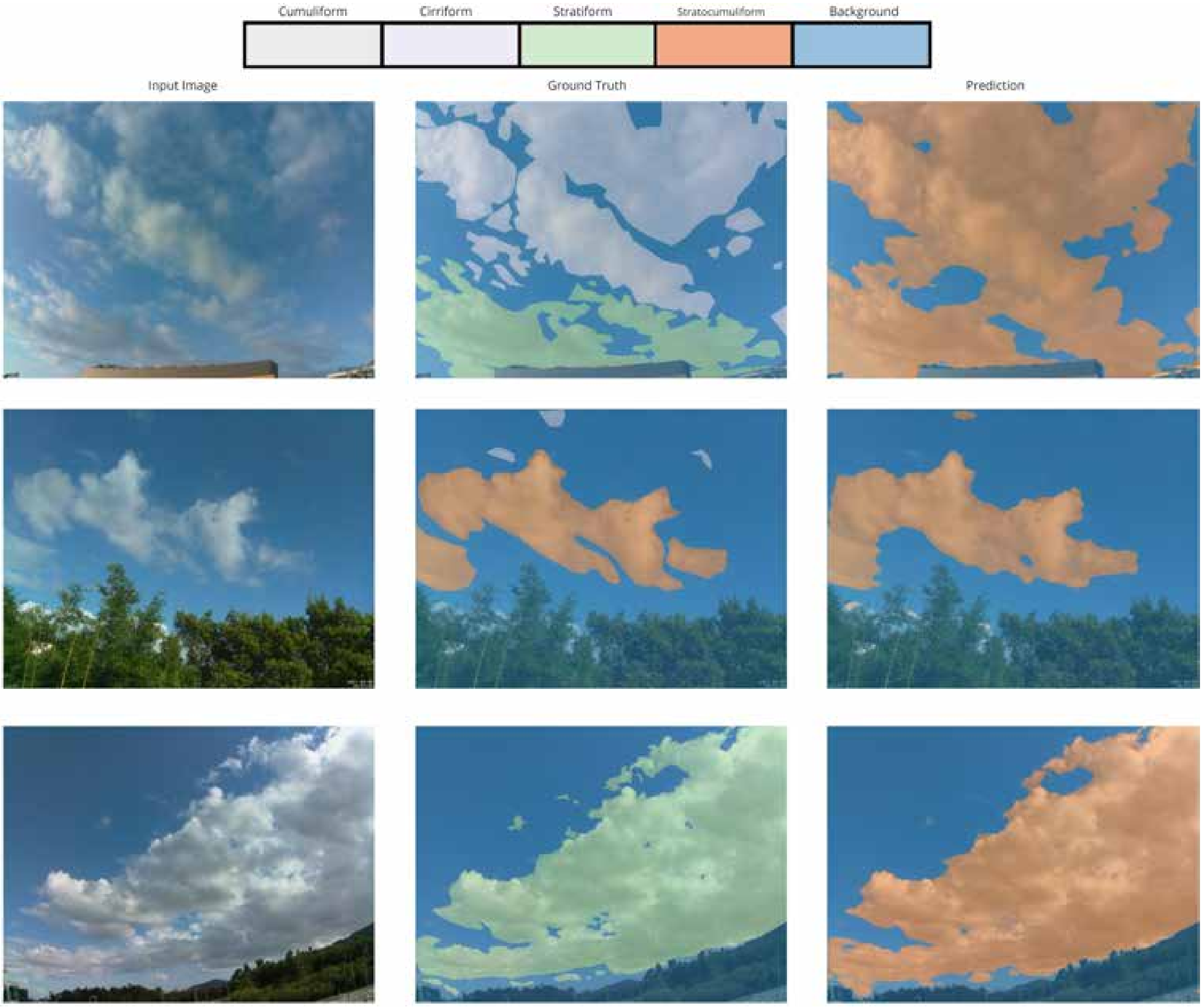


Figure 31: Experiment 38 sample results

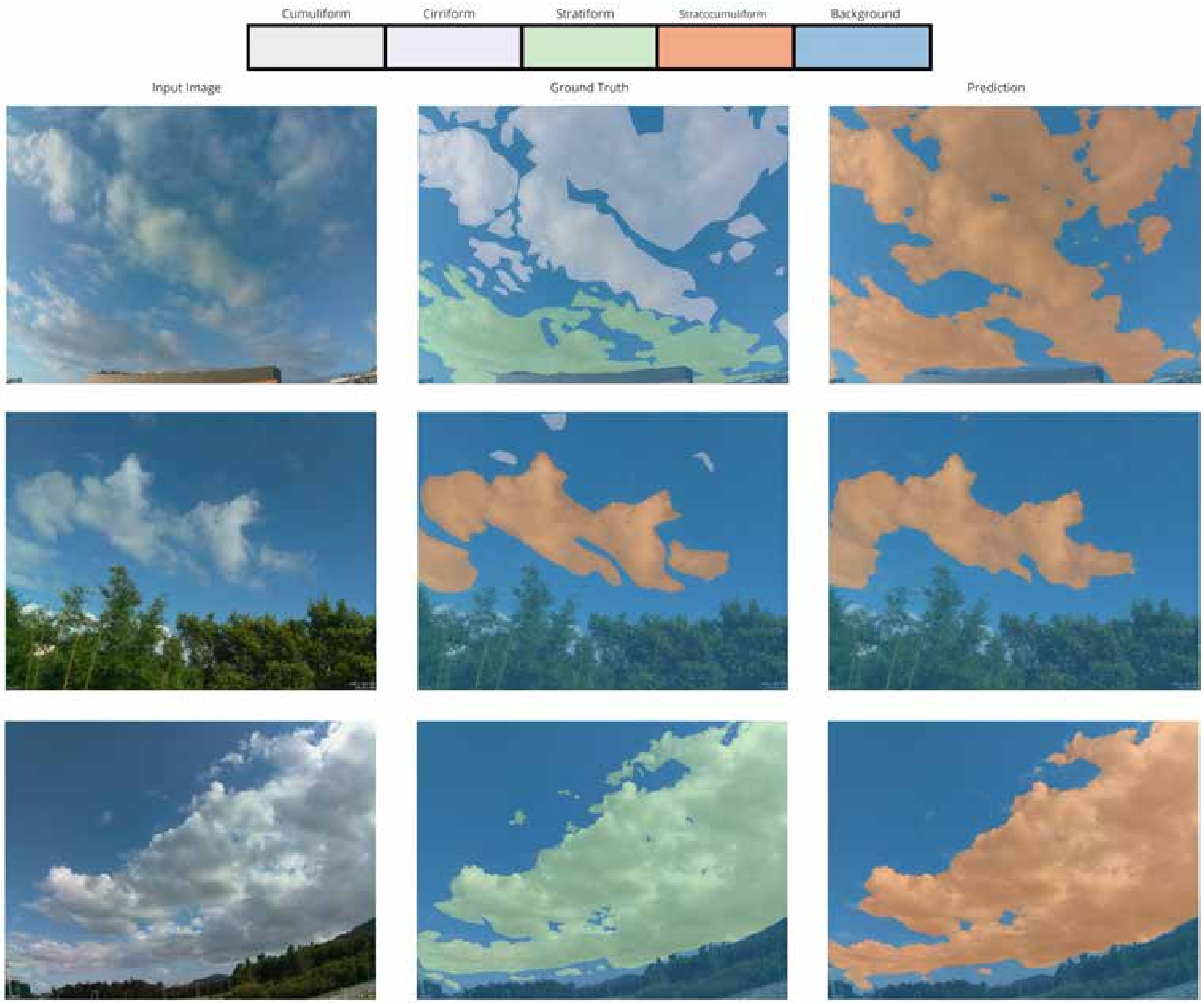


Figure 32: Experiment 39 sample results

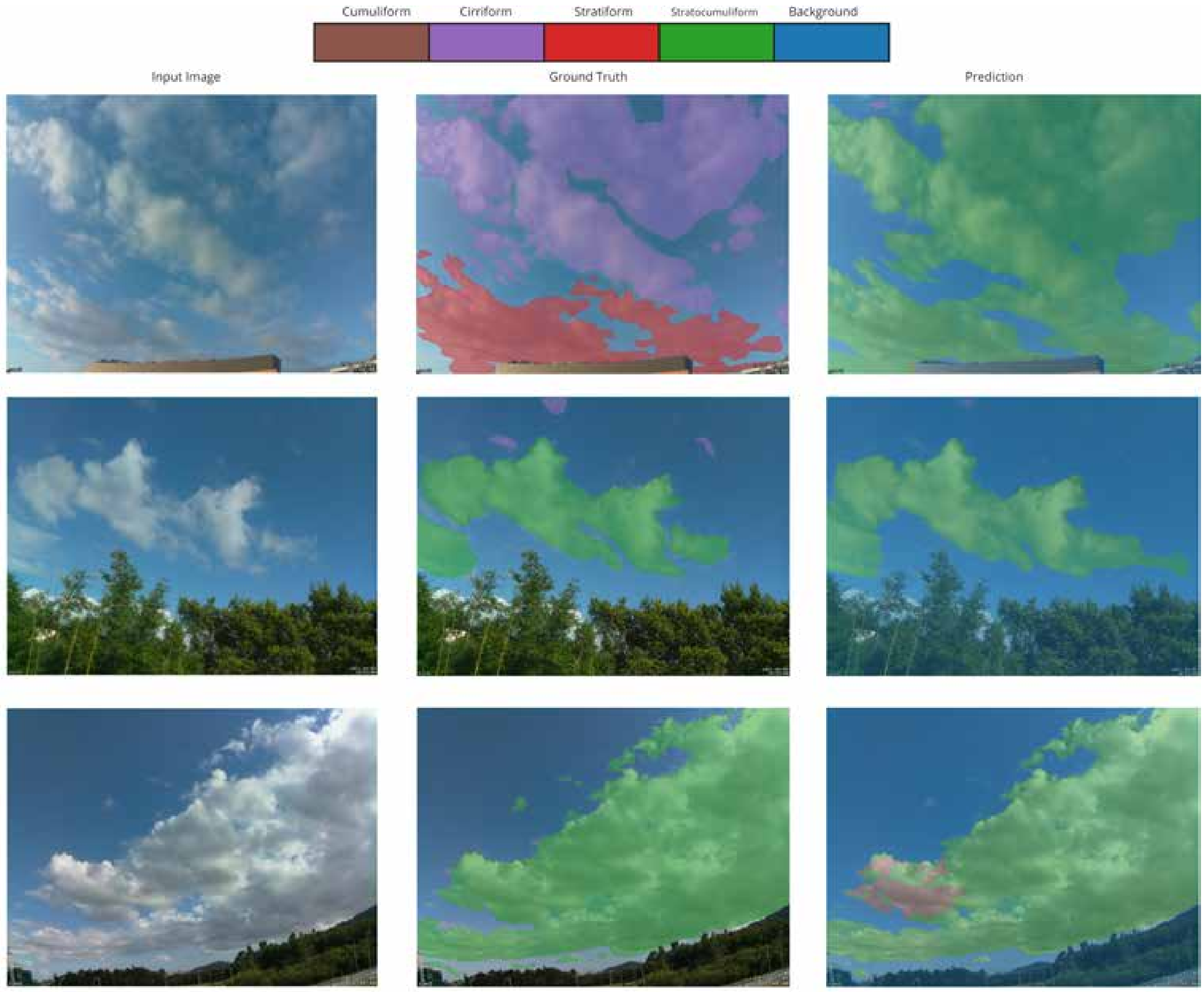


Figure 33: Experiment 41 sample results

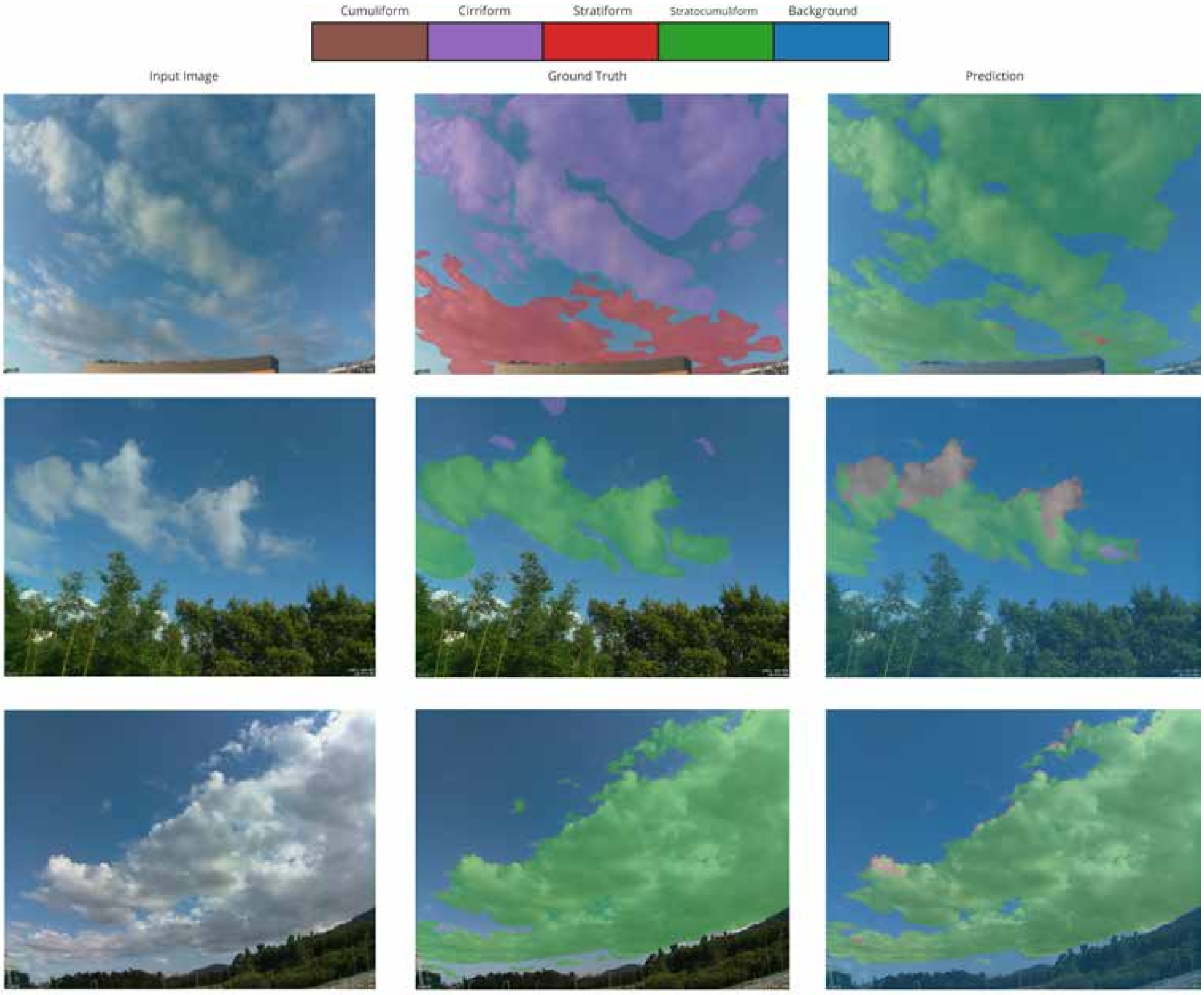


Figure 34: Experiment 42 sample results

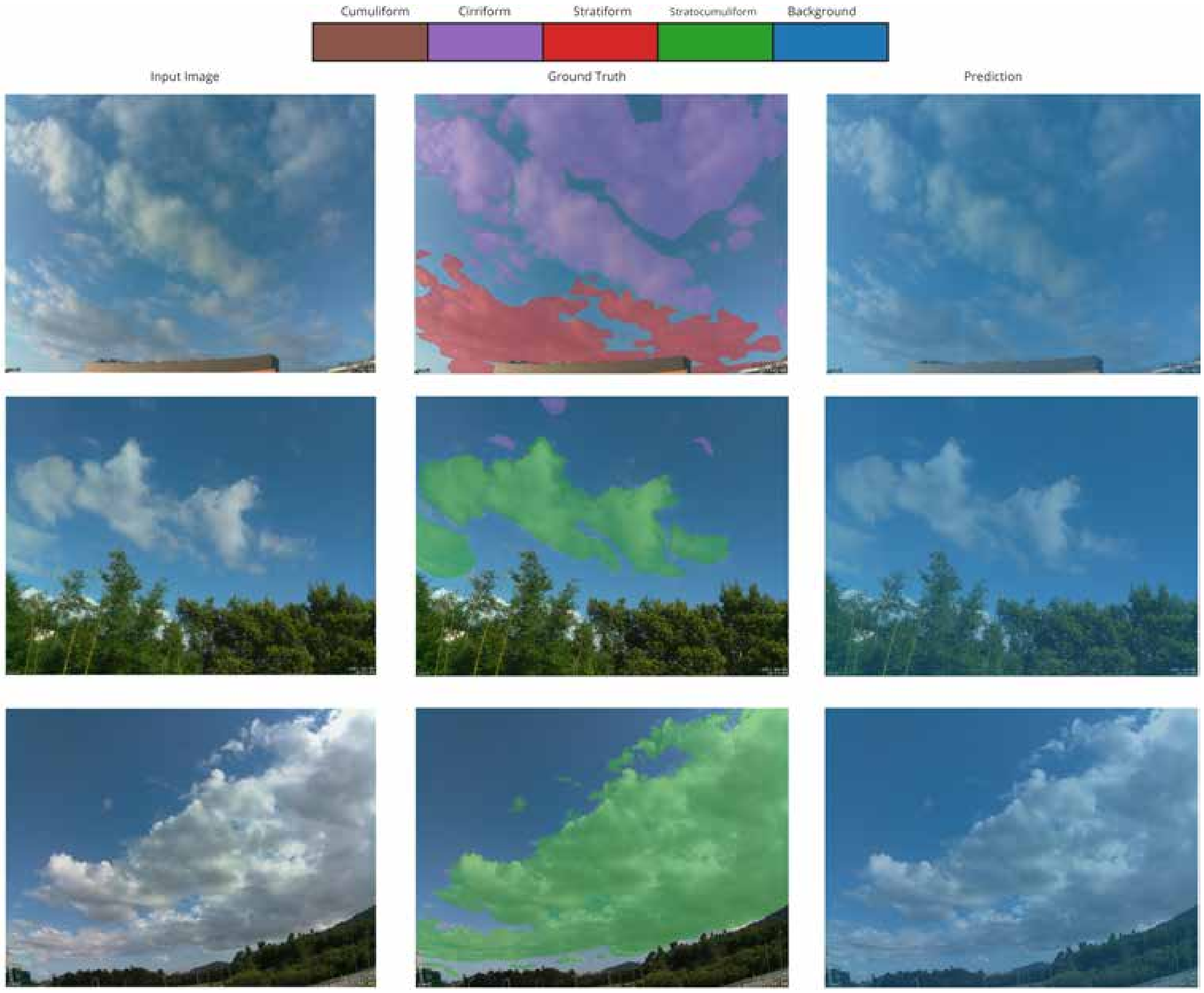


Figure 35: Experiment 43 sample results

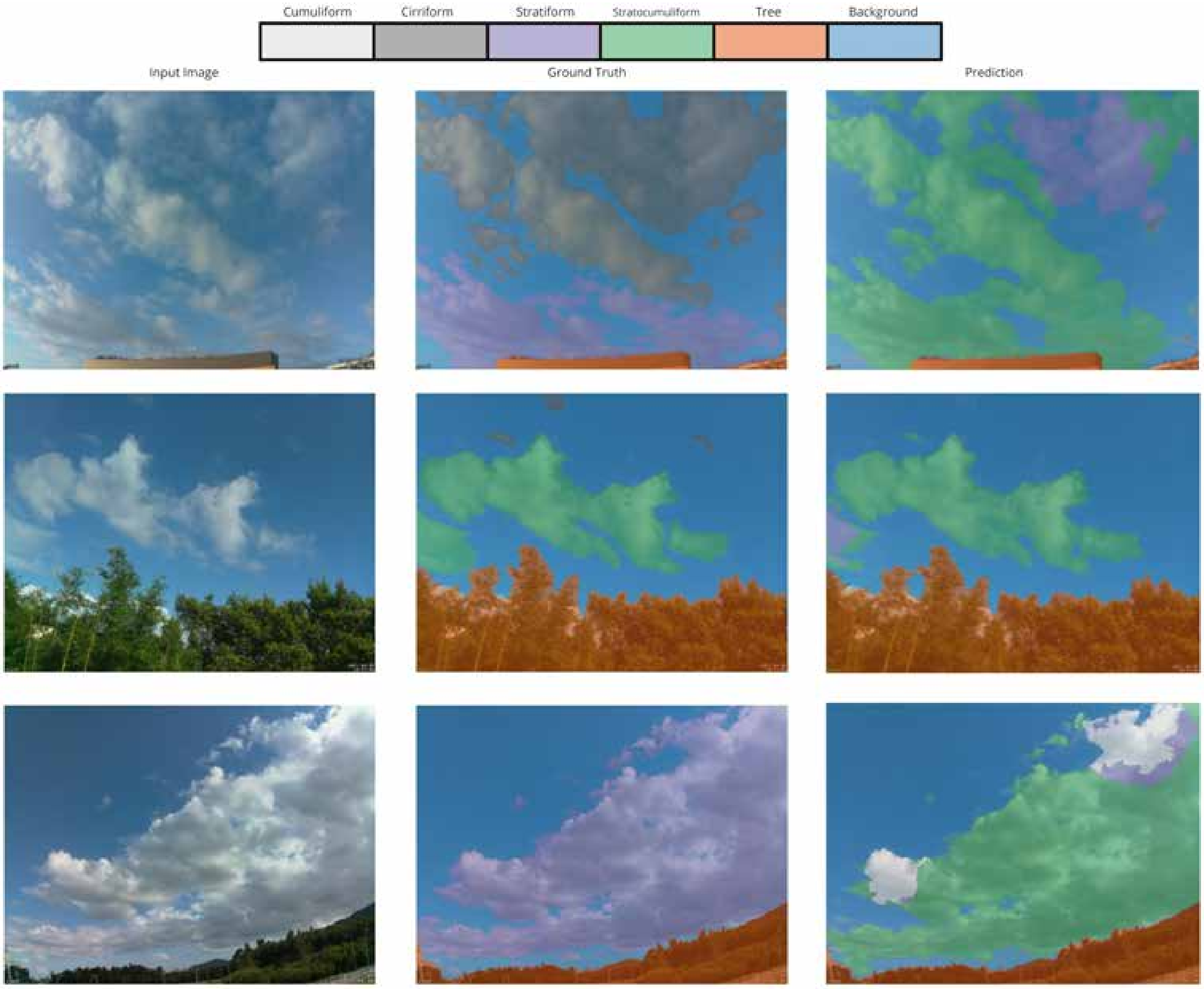


Figure 36: Experiment 45 sample results

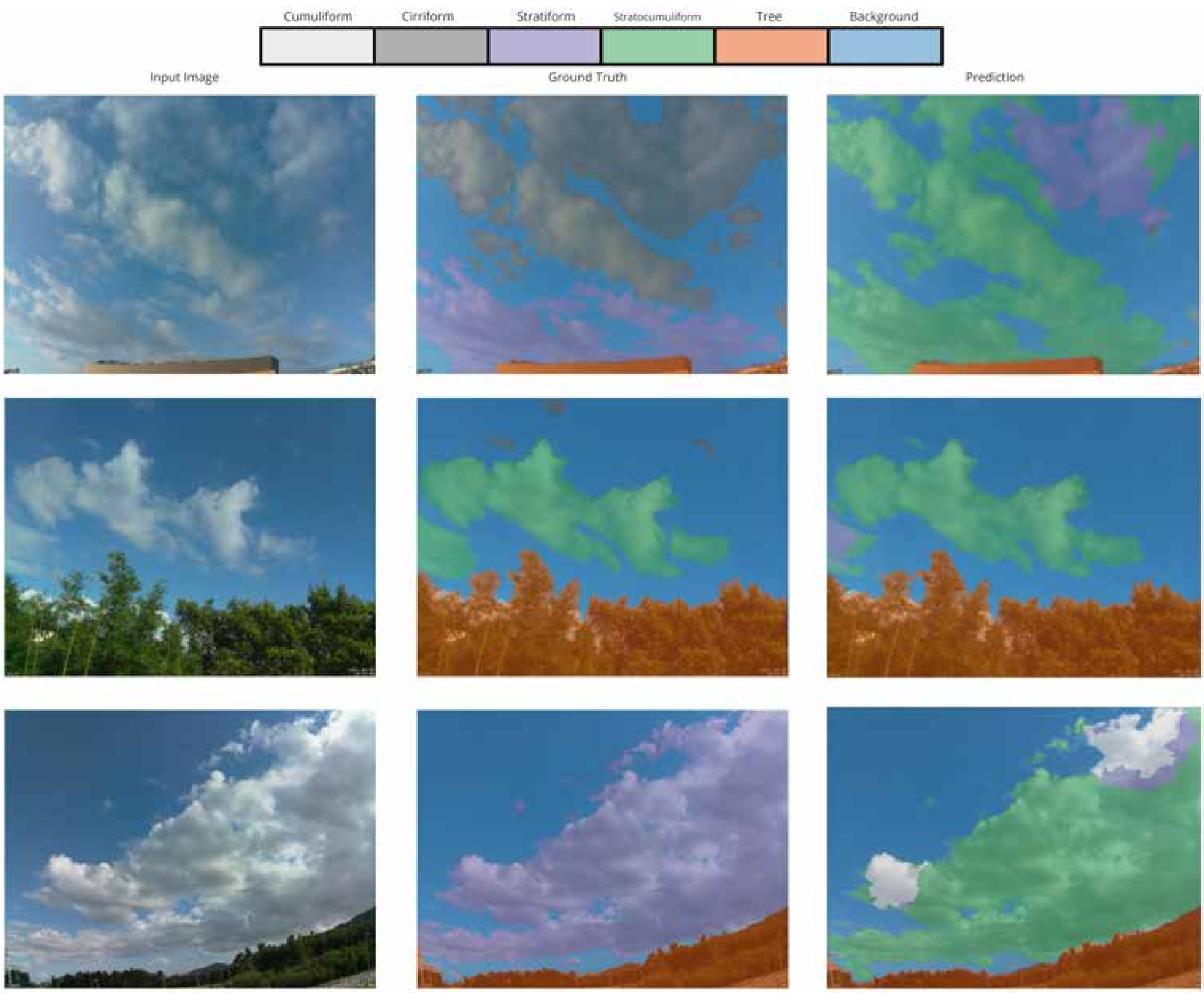


Figure 37: Experiment 46 sample results

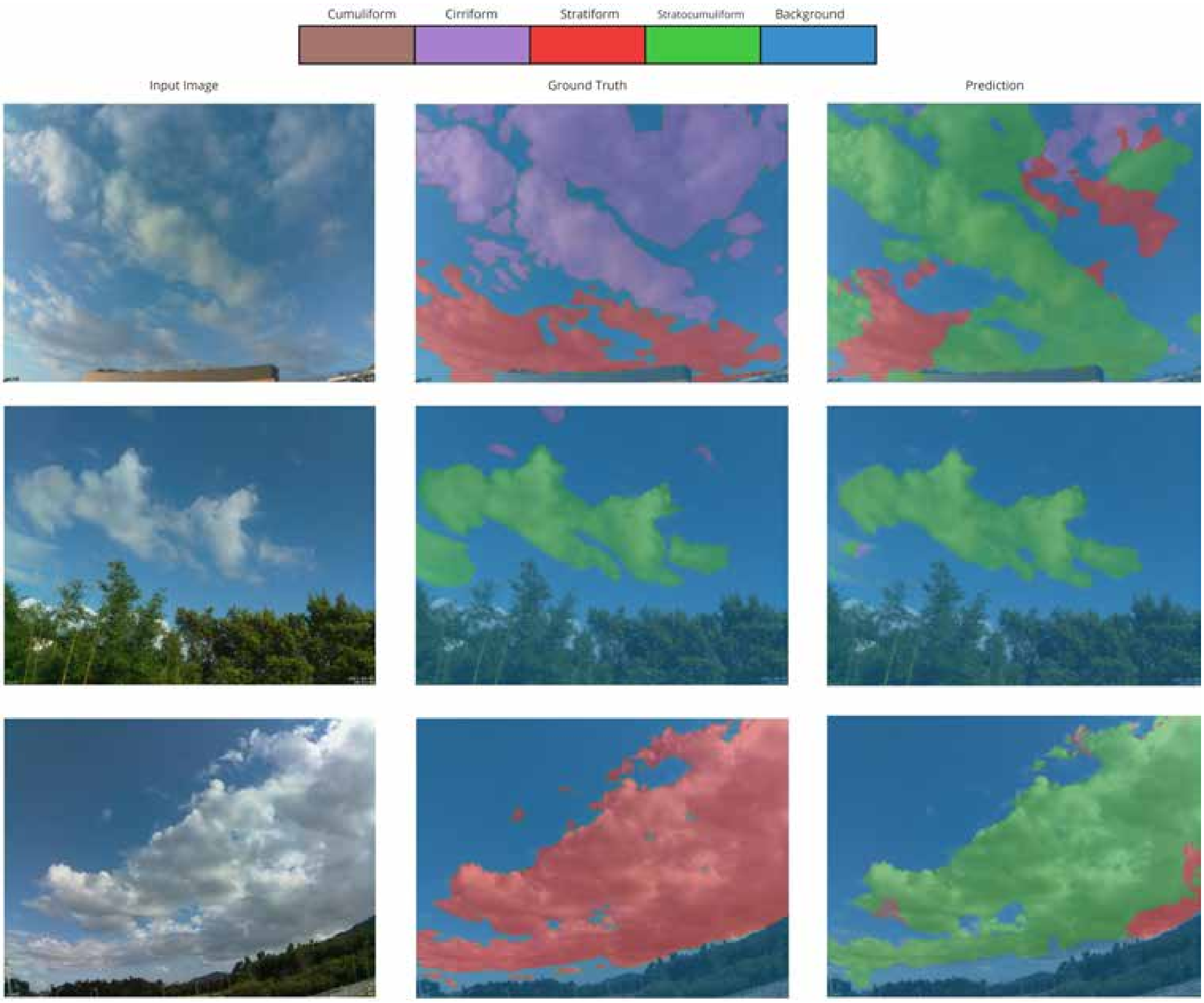


Figure 38: Experiment 47 sample results

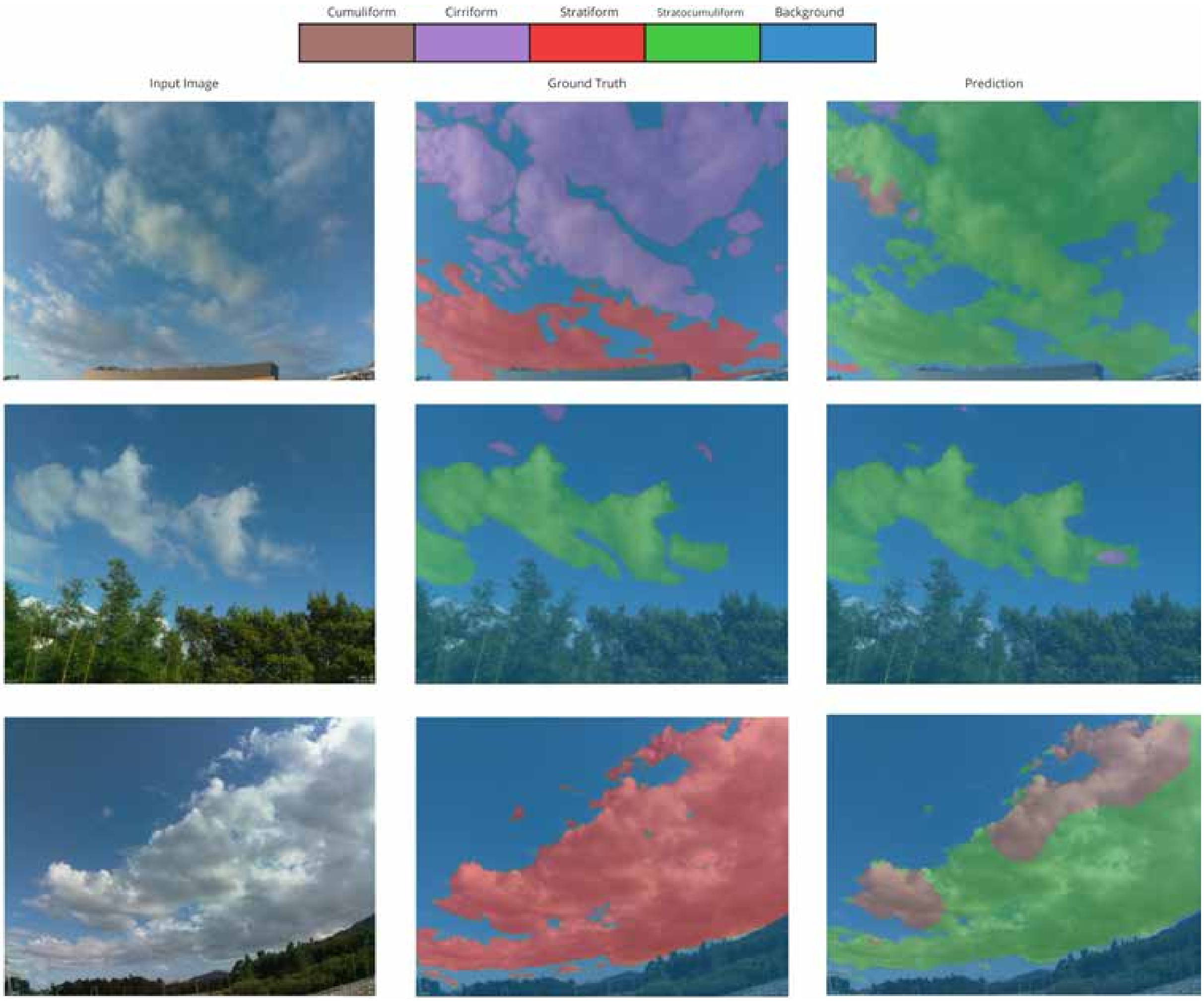


Figure 39: Experiment 48 sample results

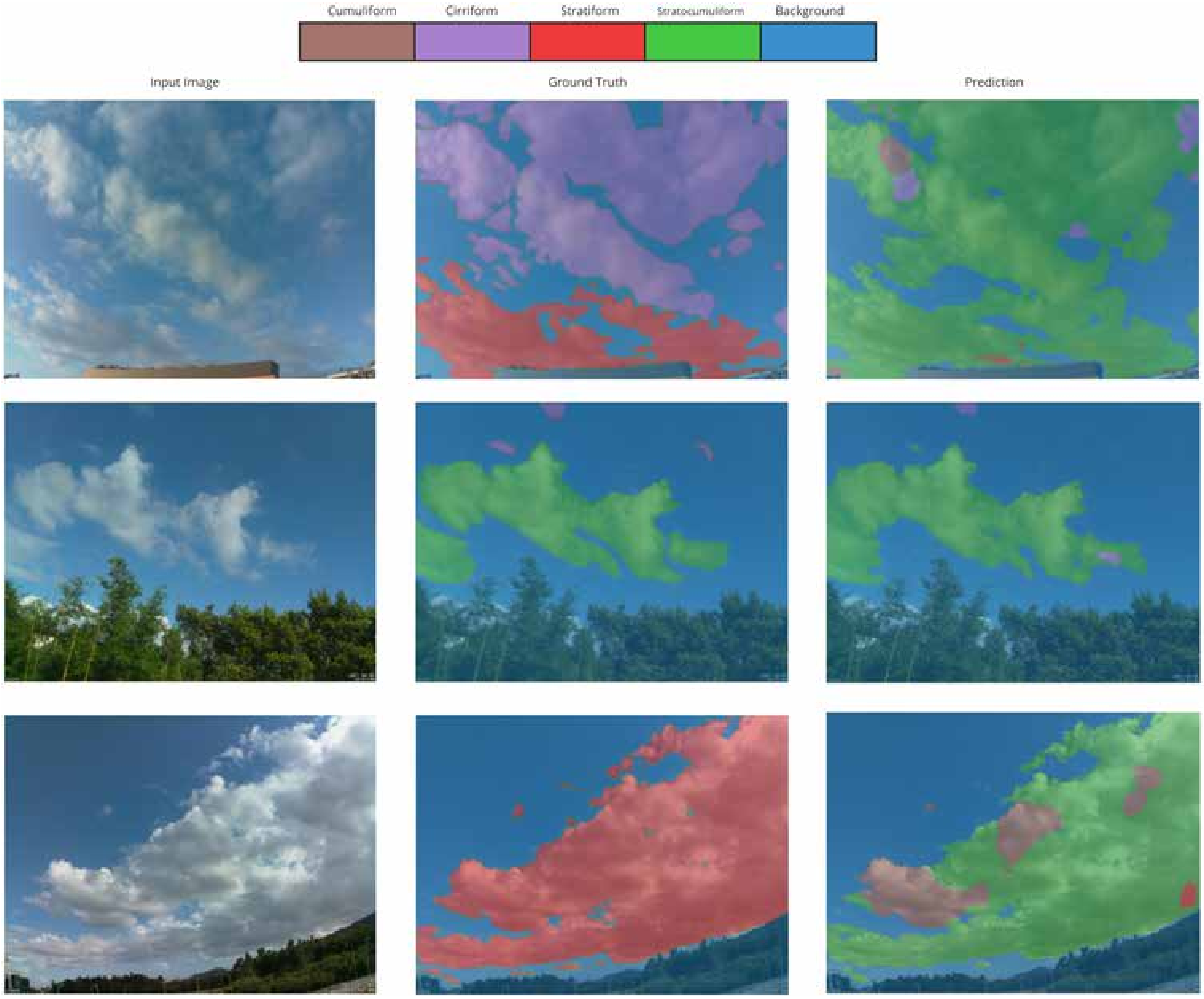


Figure 40: Experiment 49 sample results

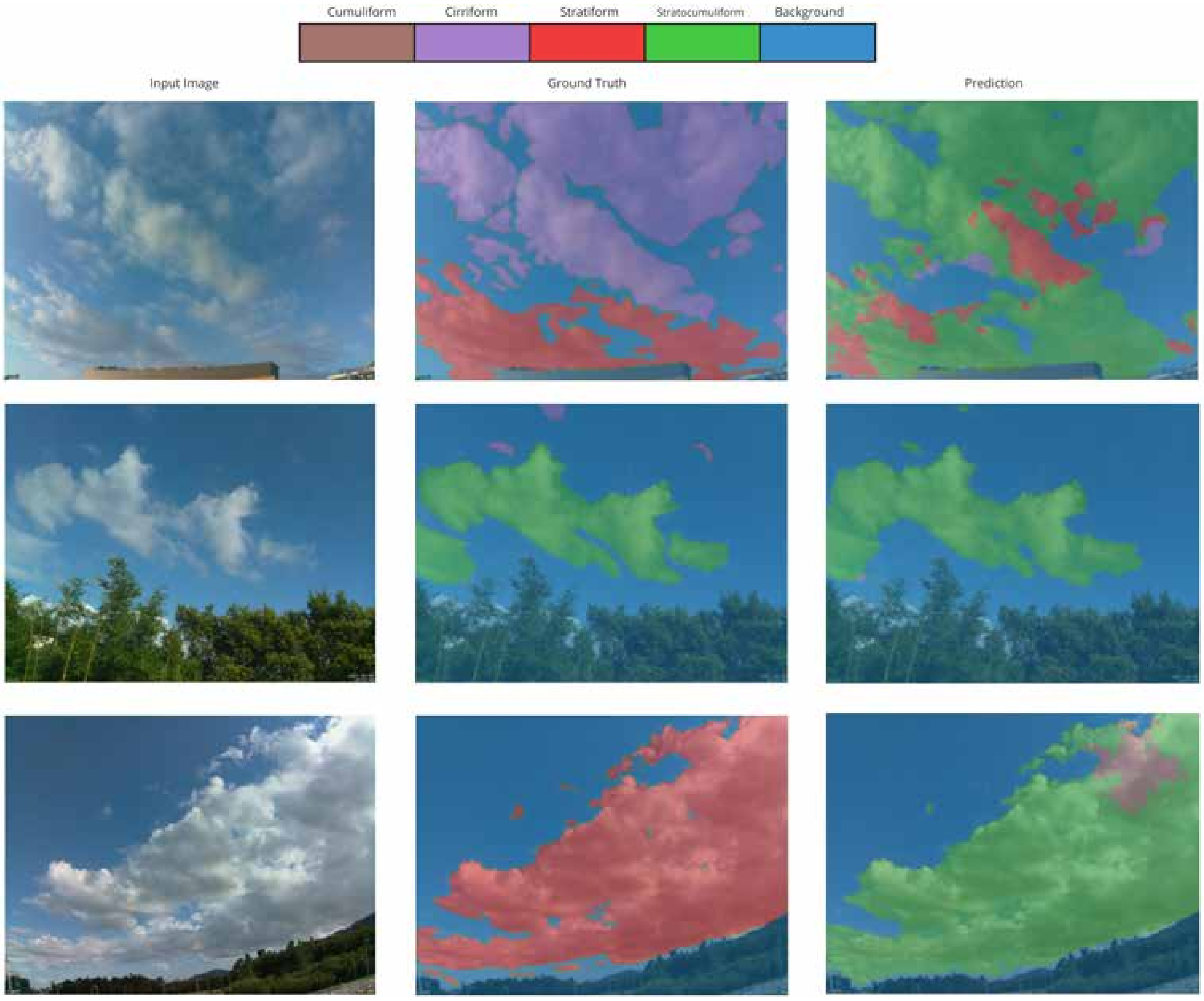


Figure 41: Experiment 50 sample results

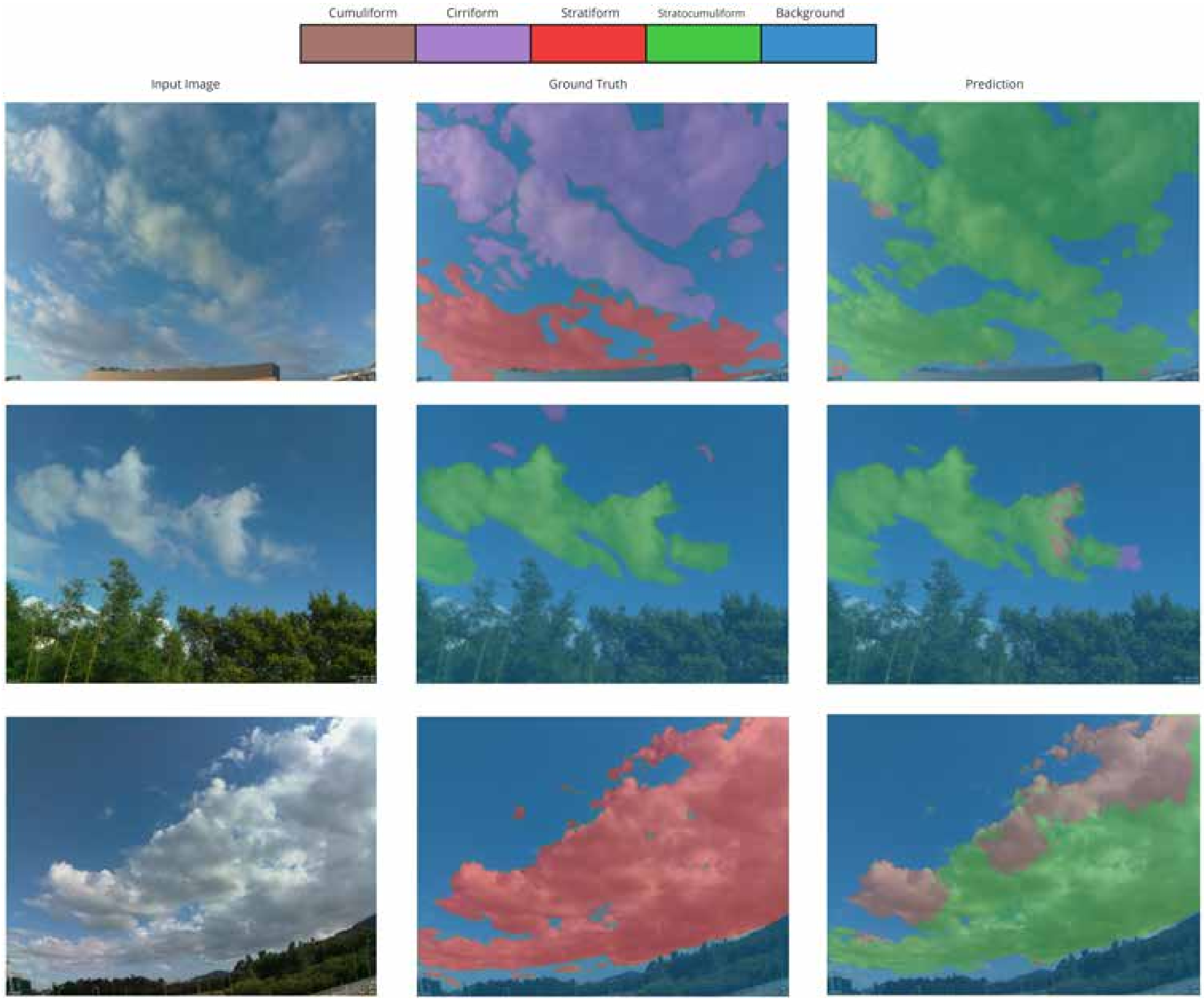


Figure 42: Experiment 51 sample results

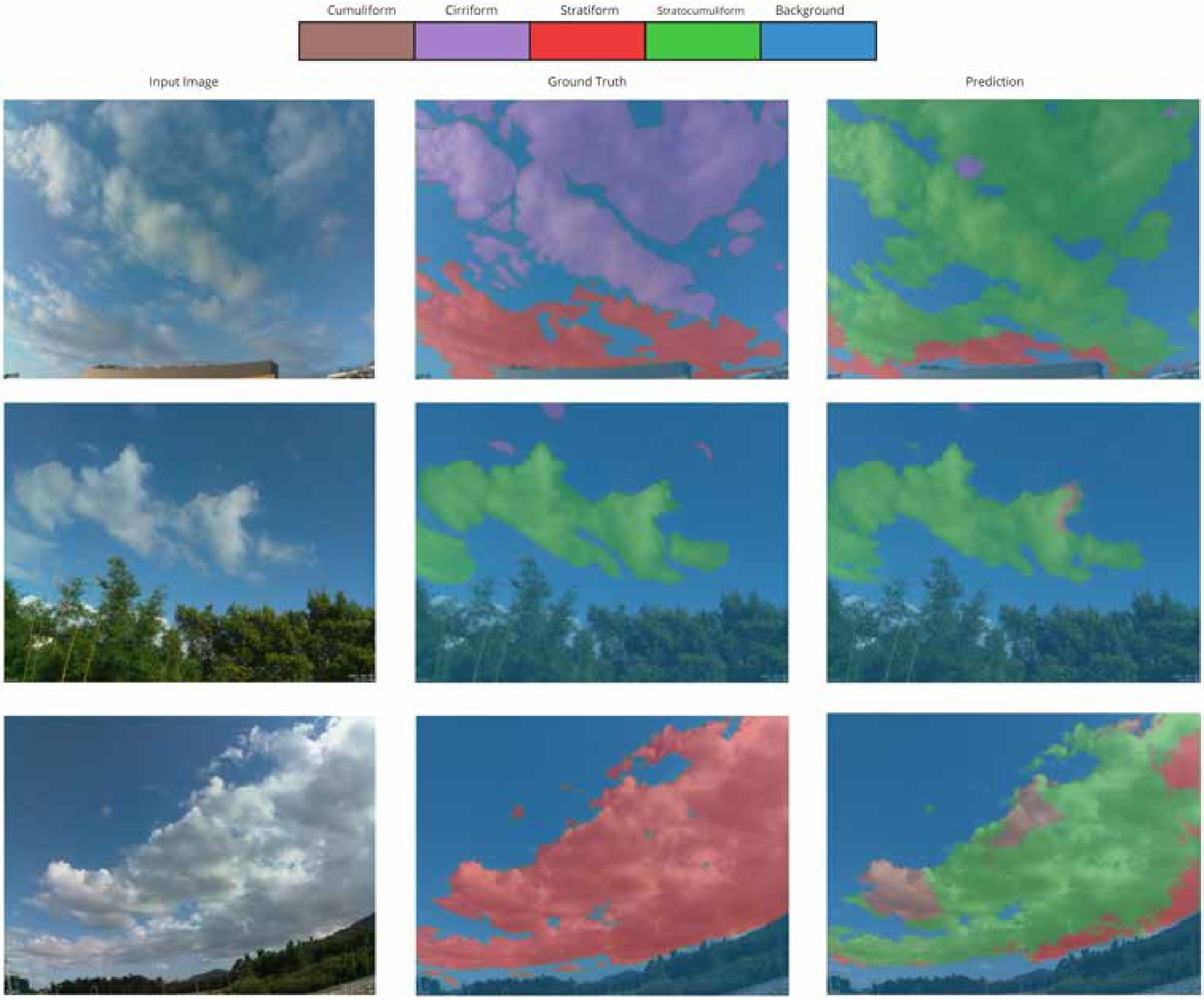


Figure 43: Experiment 52 sample results

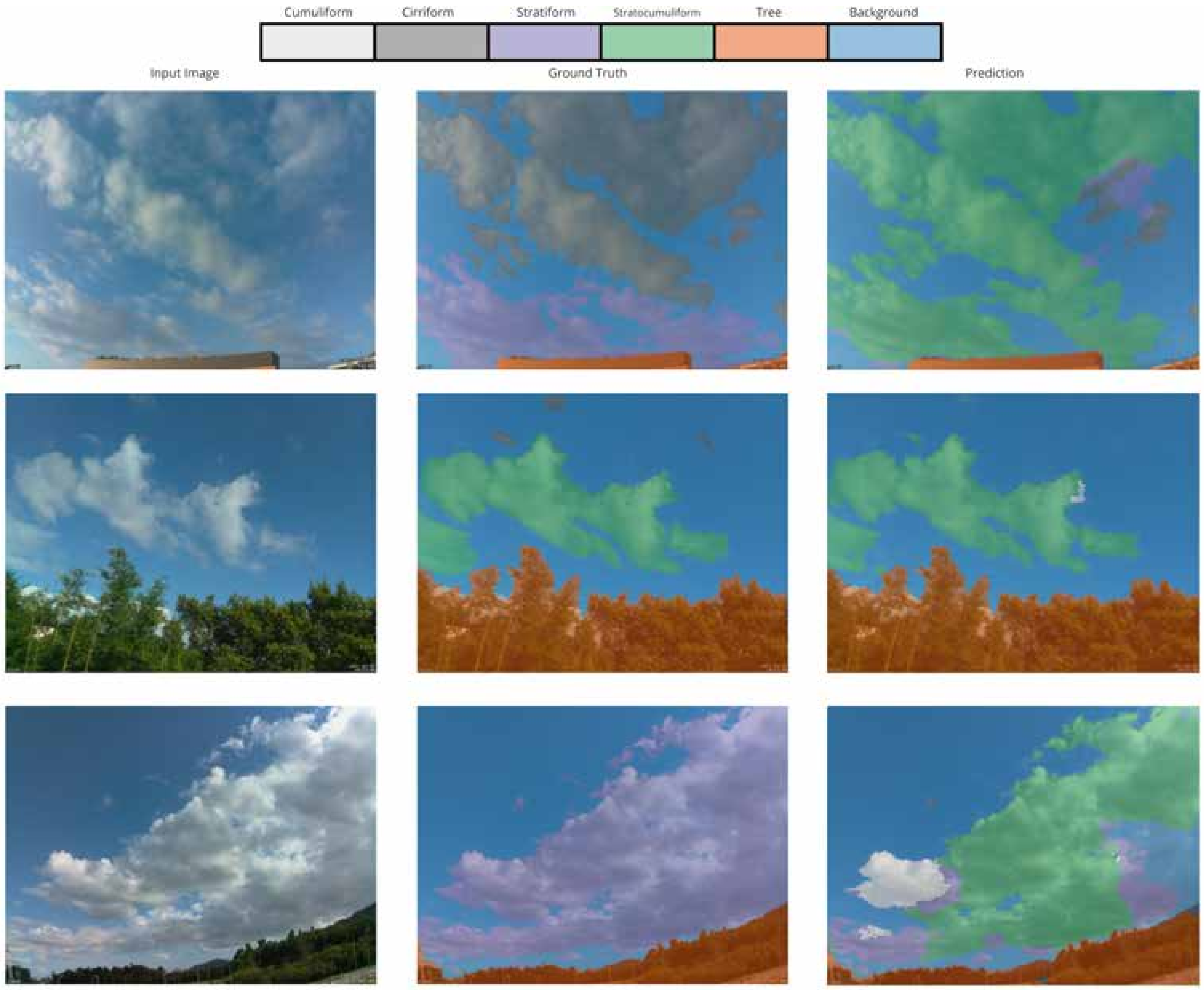


Figure 44: Experiment 54 sample results

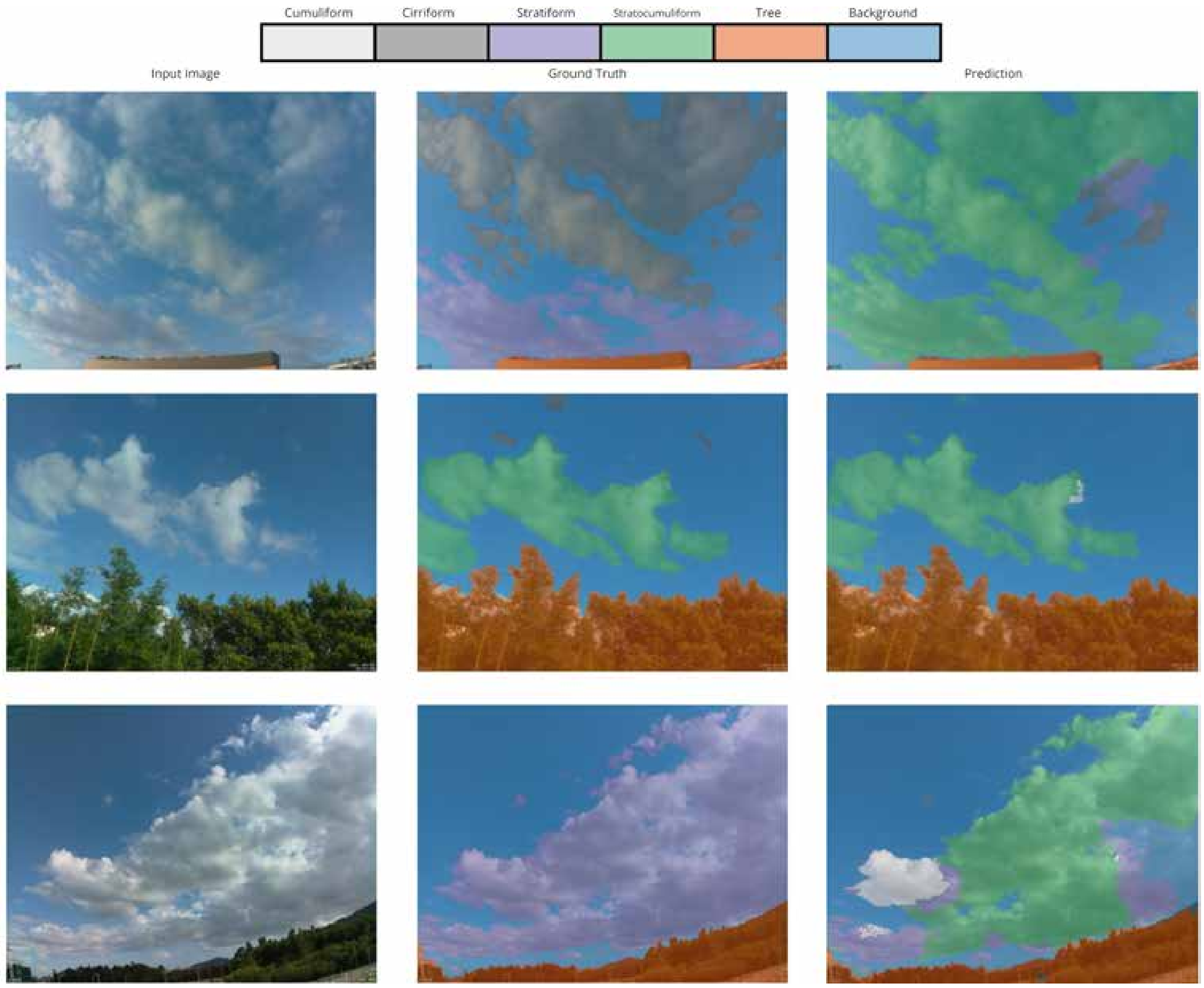


Figure 45: Experiment 55 sample results

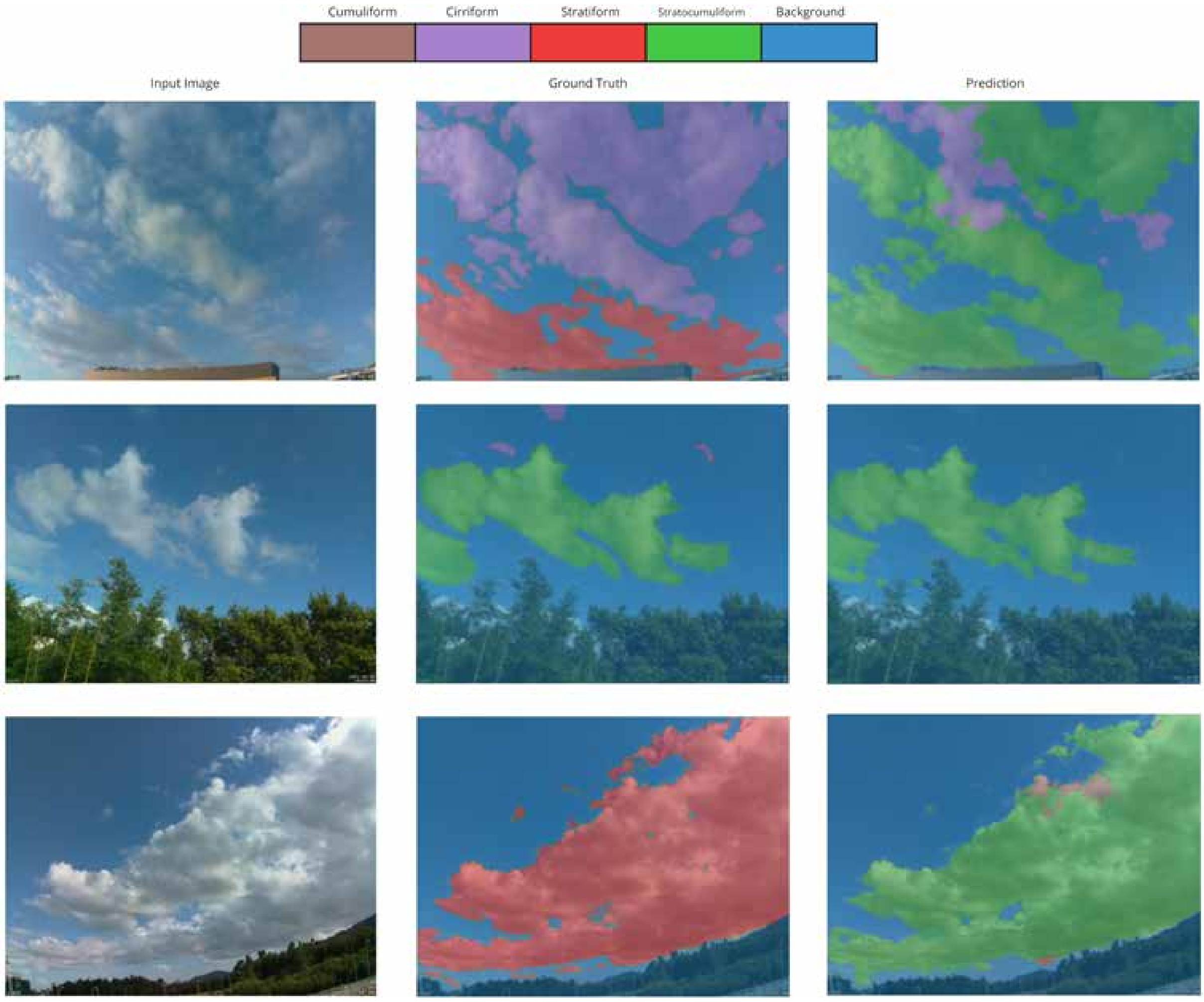


Figure 46: Experiment 56 sample results

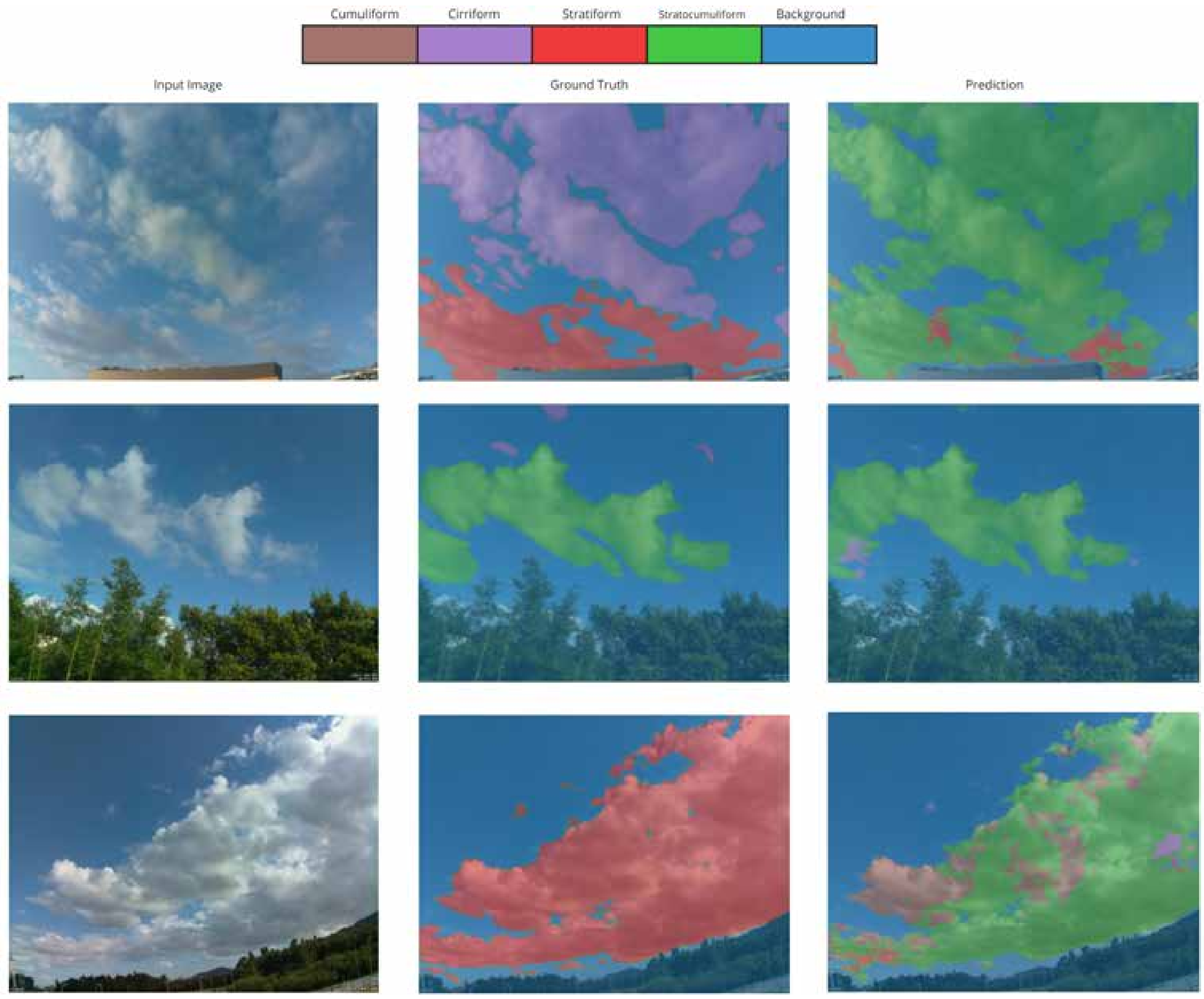


Figure 47: Experiment 57 sample results

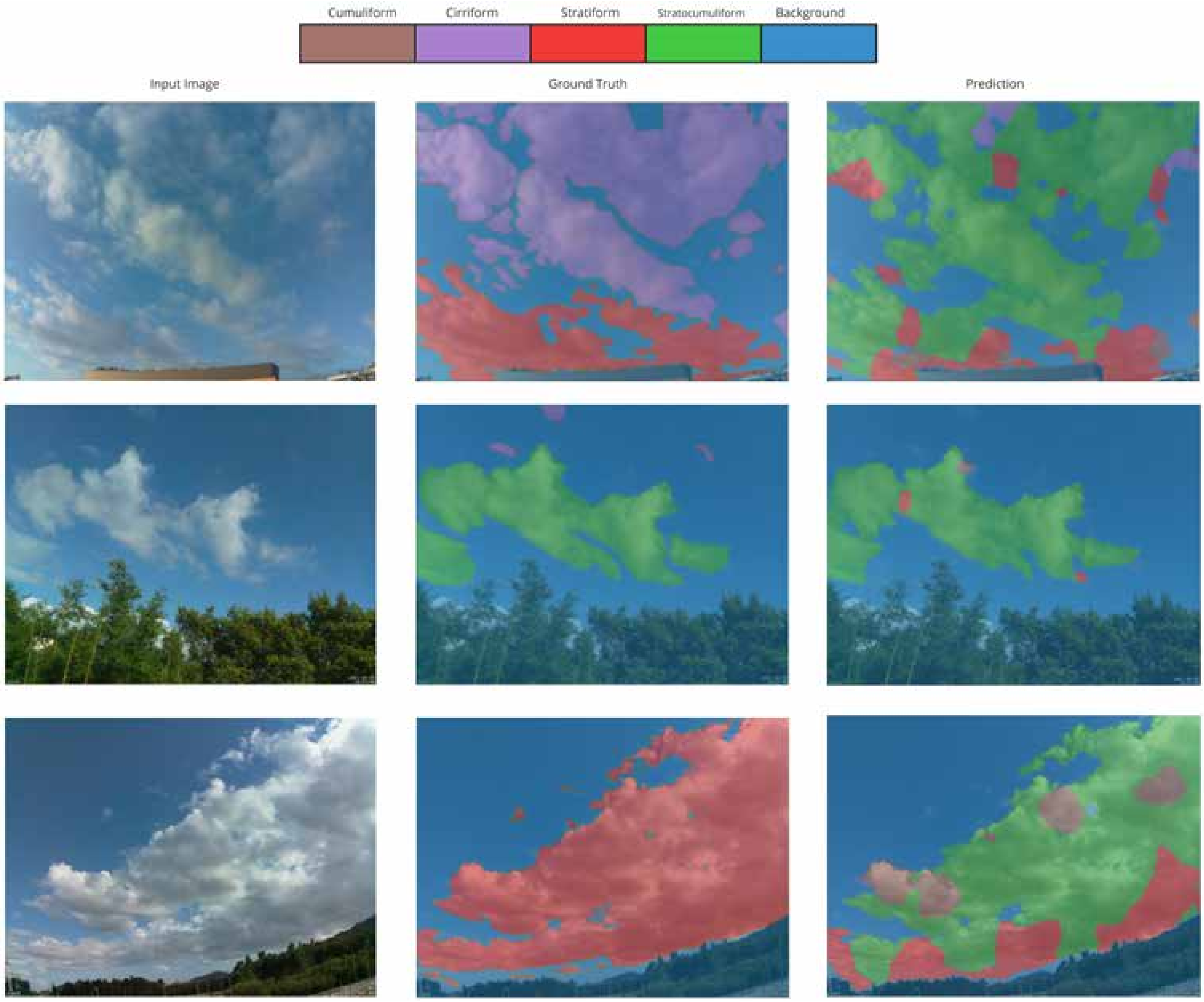


Figure 48: Experiment 58 sample results

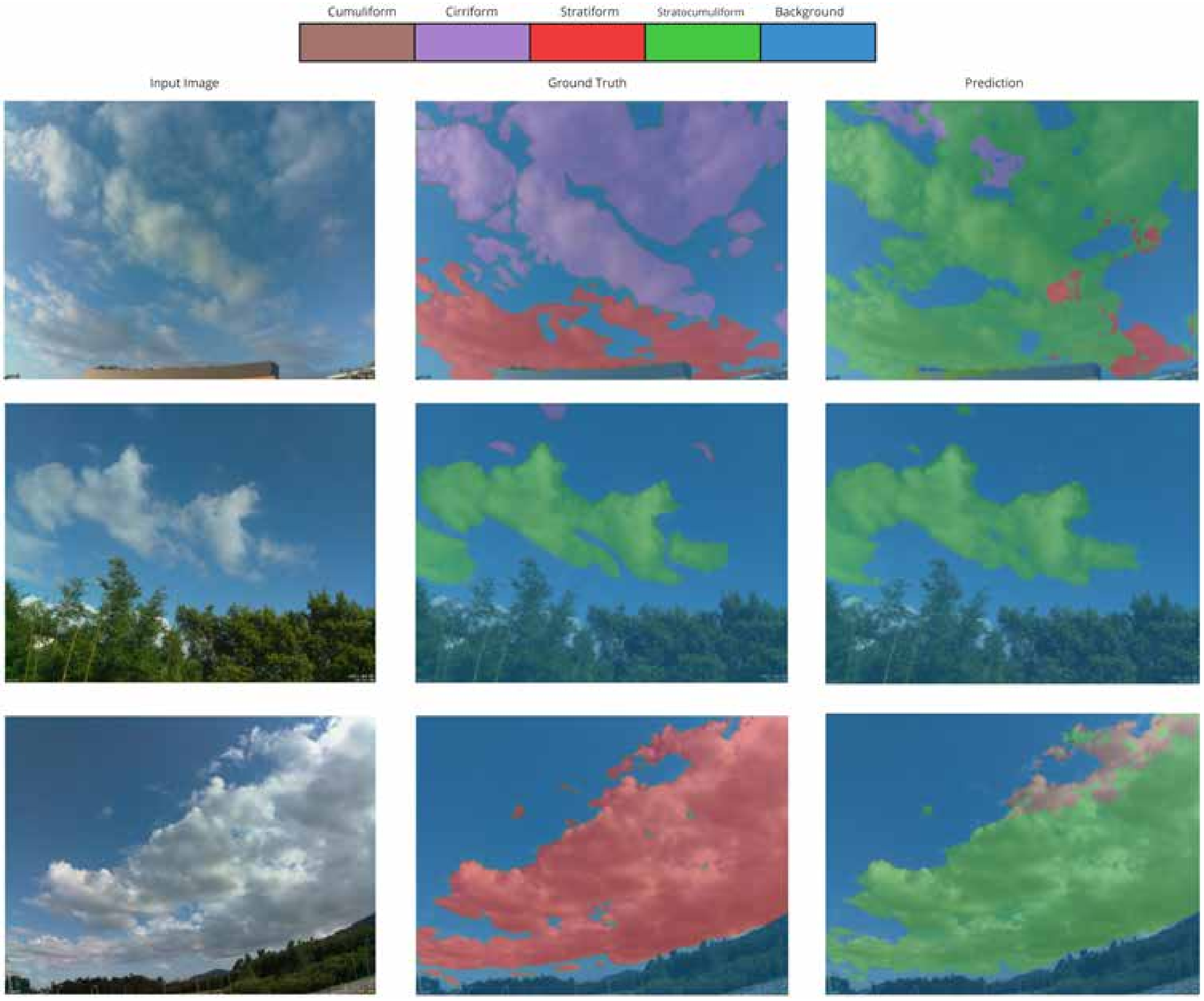


Figure 49: Experiment 59 sample results

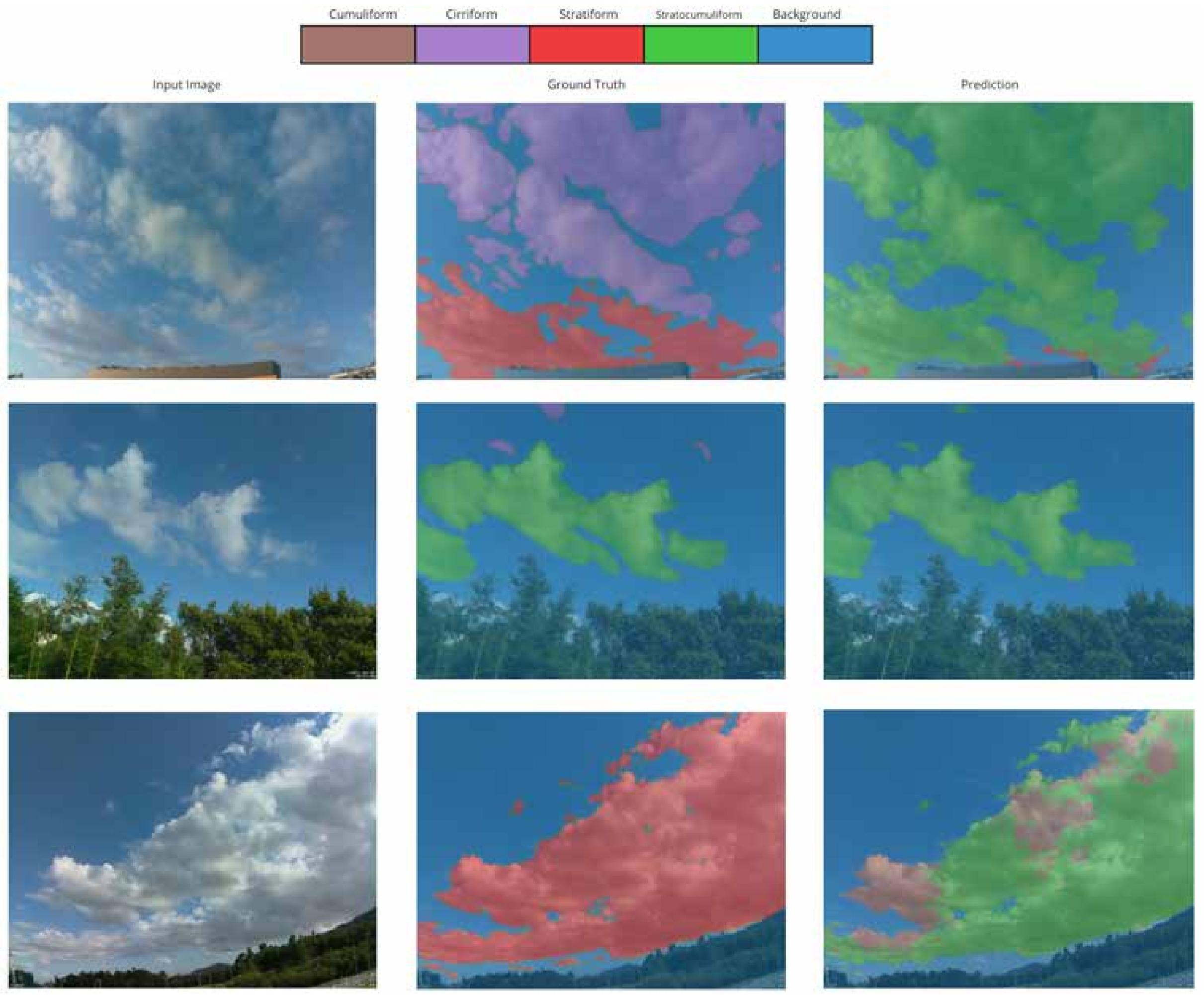


Figure 50: Experiment 60 sample results

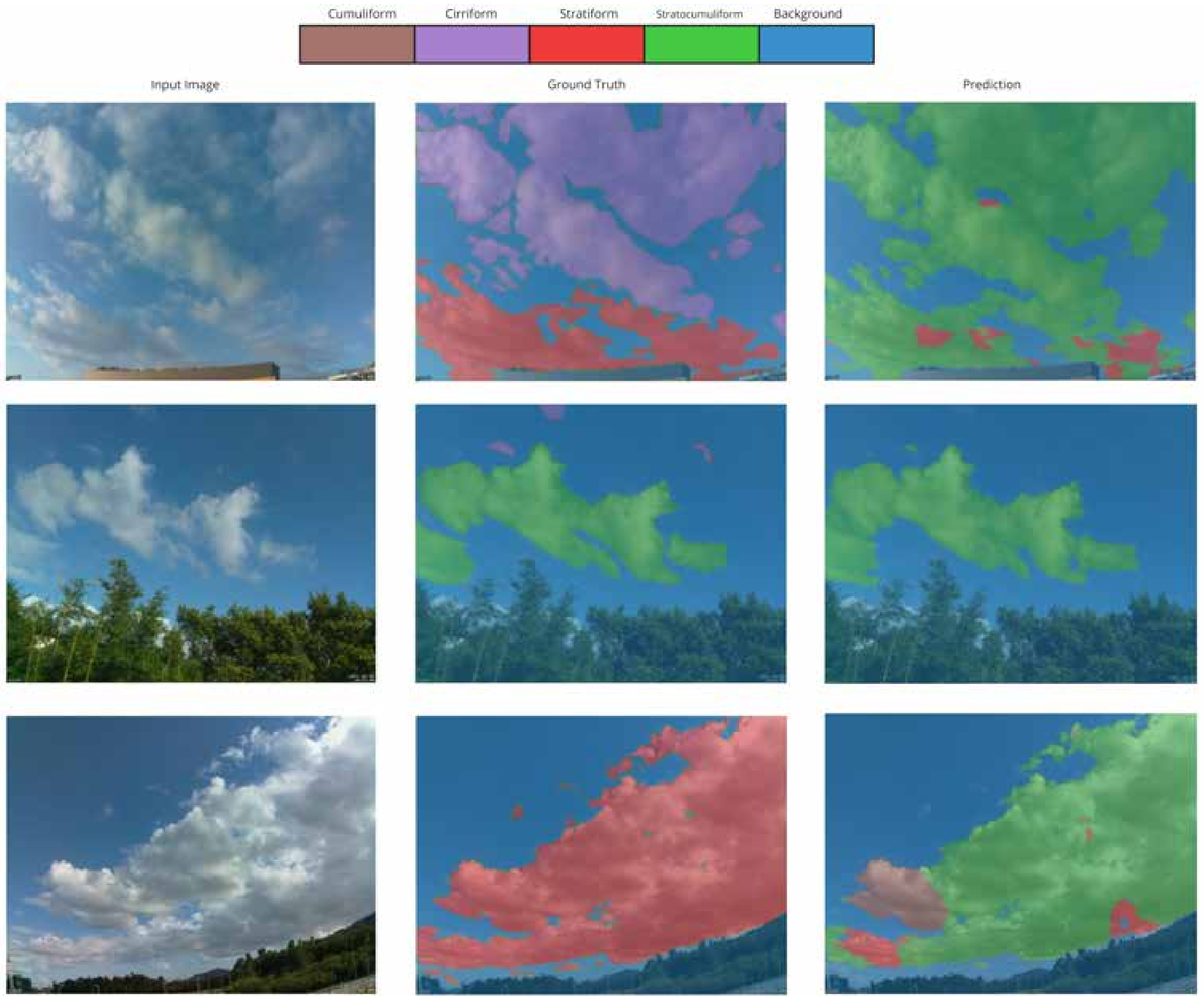


Figure 51: Experiment 61 sample results