Cairo University
Faculty of Engineering
Computer Engineering Department

AI Powered Cloud Masking

Satellite Imagery Project Report

By Team 13

Name	**Sec**	**BN**
Rawan Mostafa Mahmoud	1	18
Sara Bisheer Fekry	1	19
Menna Mohamed Abdelbaset	2	26
Fatma Ebrahim sobhy	2	1

Table of Content:

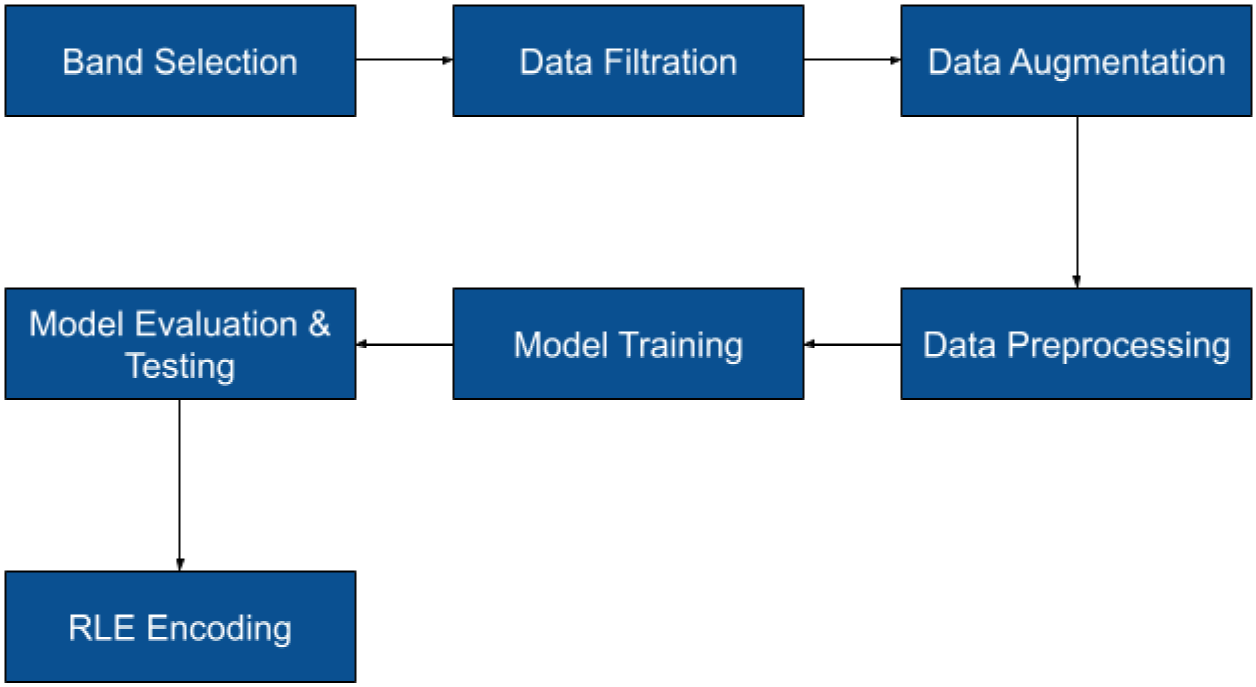
Literature Review

Detecting cloud and clear-sky pixels is a key preprocessing step in remote sensing. Multiple approaches have been explored, ranging from rule-based and machine learning methods to deep learning.

1. How have previous research papers addressed this problem?
    - The paper *Ready-to-Use Methods for the Detection of Clouds(2016)* developed both threshold-based decision rules and traditional classifiers such as **Support Vector Machines (SVM), Naive Bayes, and Random Forest** for detecting clouds, shadows, and snow. These models relied on carefully selected spectral features and band ratios, and showed promising results on well-curated datasets. While interpretable and lightweight, they have limited generalization to unseen regions and atmospheric conditions.
    - The paper *Self-Configuring nnU-Nets Detect Clouds(2022) in Satellite Images* introduced a deep learning framework capable of automatically adapting to various datasets for cloud segmentation tasks. Applied to Sentinel-2 and Landsat-8 imagery, the nnU-Net achieved a Jaccard index of 0.882 on over 10,000 unseen Sentinel-2 image patches.
    - The paper *Cloud masking for Sentinel-2 using deep learning and temporal features(2024)* combined a **U-Net** structure with a **RegNetY_006** The U-Net architecture facilitates fine-grained segmentation, while RegNetY provides strong feature extraction capabilities. This combination results in robust performance across varying cloud types and lighting conditions.
    - Additionally, the *Complete Guide to Data Augmentation* outlines techniques such as image flipping, color jitter, and noise injection, which are increasingly applied to remote sensing data to help ML models generalize better when labeled samples are limited.

2. Are there existing models or approaches that you can leverage as a starting point?
    - We started with U-Net for cloud detection due to its strong ability for pixel-wise segmentation, making it well-suited for identifying clouds in satellite imagery. Additionally, we explored machine learning models like Naive Bayes and SVM to act as our baseline and also offer a simpler approach for cloud detection based on selected spectral features.

Project Pipeline



Preprocessing Module

1- Automatic Data Filtration

This was our first trial when we found out about the noise in the data. After the EDA phase, we noticed that a lot of cloudy images have black corresponding masks, and some non-cloudy images have white masks which causes confusion and decreases the model accuracy.

So, our first approach was to remove masks that are all-black and masks that are all-white from the dataset, but this of course removed correct and incorrect samples so it wasn't a very robust approach.

2- Prediction-Based Manual Data Filtration

First, we trained the model on all of the data after augmentation and got an average accuracy (around 83%) so we thought of this as a good start for our model to help us filter the data.

We predicted the masks for the dataset, calculated the dice score of each image, sorted them ascendingly and started viewing the images from lowest dice scores and removing image-mask pairs that don't make sense. This approach helped us improve the accuracy significantly.

3- Data Augmentation

After the filtration stage, the dataset size was reduced, so we decided to add augmentation to our pipeline, we are using the following augmentation pipeline:

```
Resize(256, 256),
HorizontalFlip(p=0.5),
VerticalFlip(p=0.5),
RandomRotate90(p=0.5),
```

With 3 augmentations per image with the above probabilities.
Data augmentations have improved our average dice score around 3%
(Check more at the trials section)

4- Spectral Band Enhancement

As a trial to increase our performance, we decided to add 2 artificial indices as 2 extra channels along with red and infrared, which are:

1- $NDVI\ (Normalized\ Difference\ Vegetation\ Index)$

$$NDVI = \frac{(NIR - Red)}{(NIR + Red)}$$

Vegetation reflects NIR strongly, but clouds do not.
Red light is absorbed by vegetation, but not by clouds.
So:

- Vegetated areas → high NDVI (close to +1)
- Cloud-covered or non-vegetated areas → low NDVI (close to 0 or negative)

2- $NDWI \quad (Normalized\ Difference\ Water\ Index)$

$$NDWI \quad = \frac{(Green - NIR)}{(Green + NIR)}$$

- Clouds reflect green fairly strongly
- NDWI values tend to be high for clouds (but still different from water).
- Helps distinguish clouds vs. water and clouds vs. land.

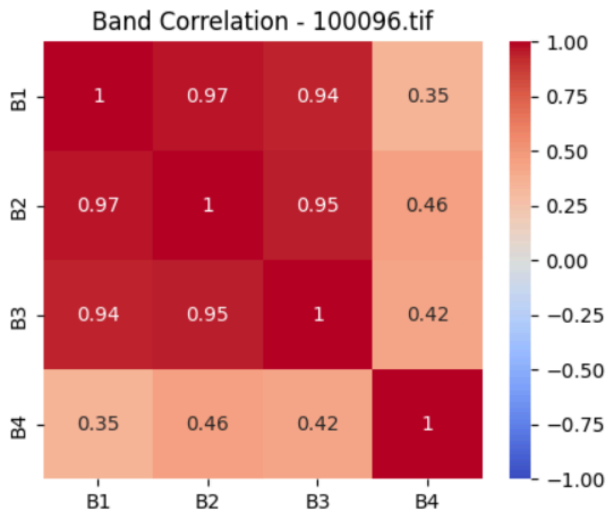
Exploratory Data Analysis

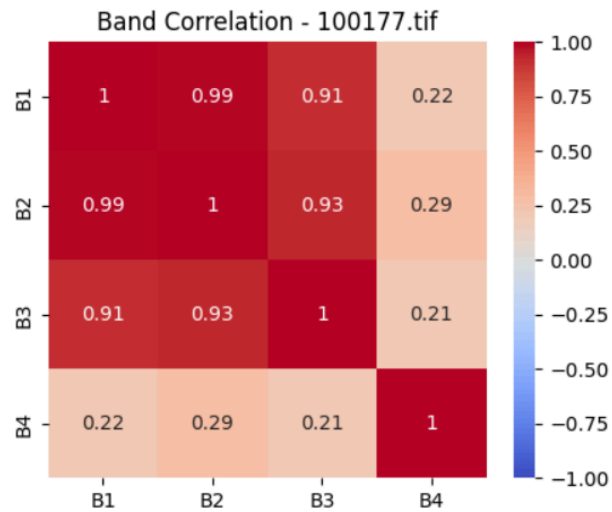
1- Correlation Test

We needed to perform band selections due to the following reasons:

- The model is very complex and takes a long training time if we use the 4 bands
- We won't be able to perform data augmentation if we use the 4 bands because kaggle working directory is only 19.5 GB which won't fit the augmented 4 band images
- We noticed a strong correlation between the R G B bands, so we thought that it would be redundant to use the 3 bands

Here's the band correlation matrix of some sample images:





As we can see, the R G B bands are strongly correlated, and the three of them weakly correlated to the infrared band, so we decided for most of our trials that we'll use the red band as a representative for the R G B bands, along with infrared as our two selected bands.

Model Selection & Training Module

Machine Learning Models Trials

Bayes Classifier Model

Trial1 :

- ➔ Trained on original data with 80:10:10 train:val:test split
- ➔ Images are 512x512
- ➔ All 4 bands were used
- ➔ No Augmentation
- ➔ No data filtration

Trial2 :

- ➔ Trained on original data with 80:10:10 train:val:test split
- ➔ Images were resized to 128x128
- ➔ Using 2 bands only (blue and infrared)
- ➔ Augmented train data with 10 augmentations per image
- ➔ Automated filtration (removing all black or white masks)

SVM Model

Trial1 :

- Trained on original data with 80:10:10 train:val:test split
- Images are 512x512
- Using 2 bands only (blue and infrared)
- No Augmentation
- No data filtration

Deep Learning Models Trials

See https://smp.readthedocs.io/en/latest/models.html#segmentation-models

1- Unet++

Trial 1:

- 80:10:10 train:val:test split
- Images are 512x512
- All 4 bands were used
- No Augmentation
- No data filtration
- lr=1e-4, 10 epochs

Trial 2:

- 80:10:10 train:val:test split
- Images were resized to 256x256
- Using 2 bands only (red and infrared)
- Augmented train data with 10 augmentations per image
- Automated filtration (removing all black or white masks)
- lr=1e-4, 10 epochs

Trial 3:

- 80:10:10 train:val:test split
- Images were resized to 128x128

- ➔ Using 2 bands only (red and infrared)
- ➔ Augmented train data with 10 augmentations per image
- ➔ Automated filtration (removing all black or white masks)
- ➔ lr=1e-4, 10 epochs

Trial 4:

- ➔ 80:10:10 train:val:test split
- ➔ Images were resized to 128x128
- ➔ Using 2 bands only (red and infrared)
- ➔ Augmented train data with 10 augmentations per image
- ➔ Automated filtration (removing all black or white masks)
- ➔ lr=1e-4, 10 epochs

Trial 5:

- ➔ 80:10:10 train:val:test split
- ➔ Images were resized to 256x256
- ➔ Using 2 bands only (red and infrared)
- ➔ Augmented train data with 3 augmentations per image
- ➔ No filtration
- ➔ lr=1e-4, 10 epochs

Trial 6:

- ➔ 80:10:10 train:val:test split
- ➔ Images were resized to 256x256
- ➔ Using 2 bands only (red and infrared)
- ➔ Augmented train data with 3 augmentations per image
- ➔ Manual filtration (using model predictions)
- ➔ lr=1e-4, 10 epochs

After testset release, we did more filtration because most of our predictions were mostly white, so we thought this was because the training data was biased with white masks, so we thought of filtering the data more and did the following trials:

Trial 7:

- 80:10 train:test split (to use more data for training)
- Images were resized to 256x256
- Using 2 bands only (red and infrared)
- Augmented train data with 3 augmentations per image
- More Manual white filtration (using model predictions)
- lr=1e-3, 10 epochs

We did more filtration trials which were to remove all white masks from the dataset:

Trial 8:

- 100% training data
- Images were resized to 256x256
- Using 2 bands only (red and infrared)
- Augmented train data with 3 augmentations per image
- Severe white masks filtration
- lr=1e-3, 12 epochs

2- Unet:

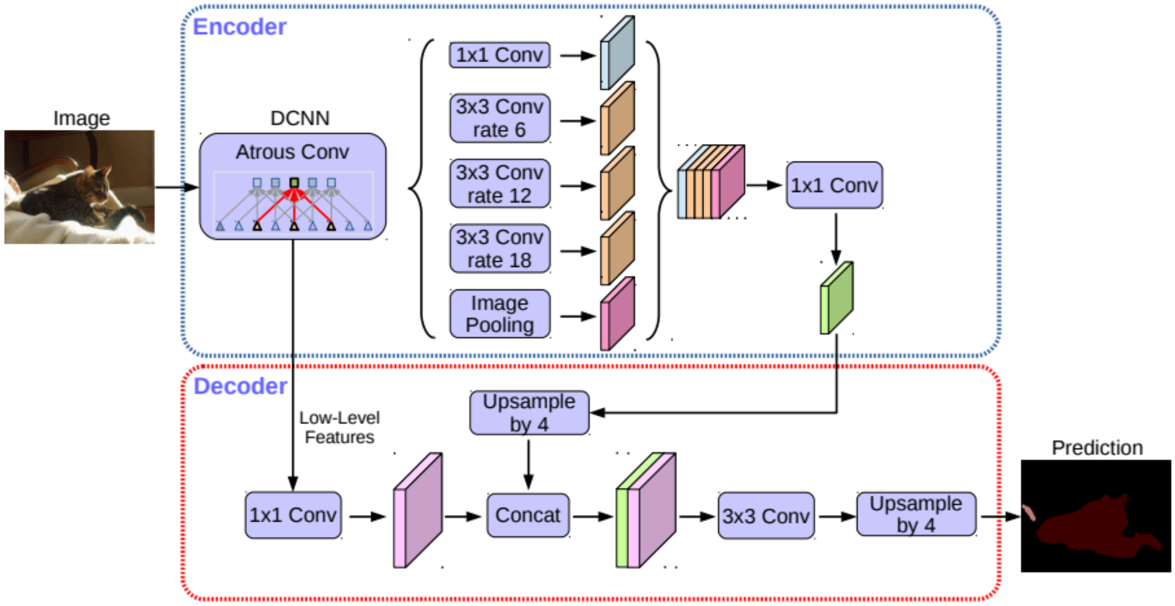
Trial 1:

- 80:10:10 train:val:test split
- Images are 512x512
- All 4 bands were used
- No Augmentation
- No data filtration
- lr=1e-4, 10 epochs

Trial 2:

- 80:10:10 train:val:test split
- Images were resized to 256x256
- Using 2 bands only (red and infrared)
- Augmented train data with 10 augmentations per image
- Automated filtration (removing all black or white masks)
- lr=1e-4, 10 epochs

3- DeepLabV3Plus:



Reference: [Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation]

Trial 1:

- ➔ 80:10:10 train:val:test split
- ➔ Images were resized to 256x256
- ➔ Using 2 bands only (red and infrared)
- ➔ Augmented train data with 3 augmentations per image
- ➔ Manual filtration (using model predictions)
- ➔ lr=0.01, 10 epochs

Trial 2: (the submitted model)

- ➔ 80:10:10 train:val:test split
- ➔ Images were resized to 256x256
- ➔ Using 2 bands only (red and infrared)
- ➔ Augmented train data with 3 augmentations per image
- ➔ Manual filtration (using model predictions)
- ➔ lr = 1e-4, 20 epochs

4- FPN

Trial 1:

- ➔ 80:10:10 train:val:test split
- ➔ Images were resized to 256x256
- ➔ Using 2 bands only (red and infrared)
- ➔ Augmented train data with 3 augmentations per image
- ➔ Manual filtration (using model predictions)
- ➔ lr = 1e-3, 20 epochs

Trial 2:

- ➔ 80:10:10 train:val:test split
- ➔ Images were resized to 256x256
- ➔ Using 4 bands (red and infrared) in addition to the two aforementioned indices (NDWI and NDVI)
- ➔ No augmentation
- ➔ Manual filtration (using model predictions)
- ➔ lr = 1e-3, 20 epochs

Trial 3:

- ➔ 80:10 train:test split (to increase the training set size)
- ➔ Images were resized to 256x256
- ➔ Using 2 bands only (red and infrared)
- ➔ Augmented train data with 3 augmentations per image
- ➔ More Manual white filtration (using model predictions)
- ➔ lr = 1e-3, 20 epochs

Trial 4:

- ➔ 80:10 train:test split (to increase the training set size)
- ➔ Images were resized to 256x256
- ➔ Using 2 bands only (red and infrared)
- ➔ Augmented train data with 3 augmentations per image
- ➔ Severe white masks filtration
- ➔ lr = 1e-3, 20 epochs

Performance Analysis Module

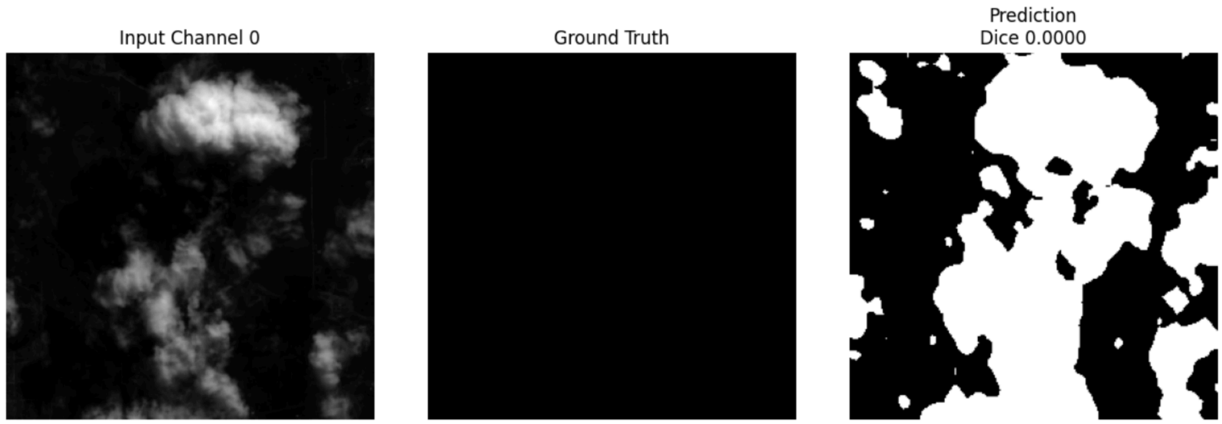
These are the maximum scores that could be achieved by the tried models

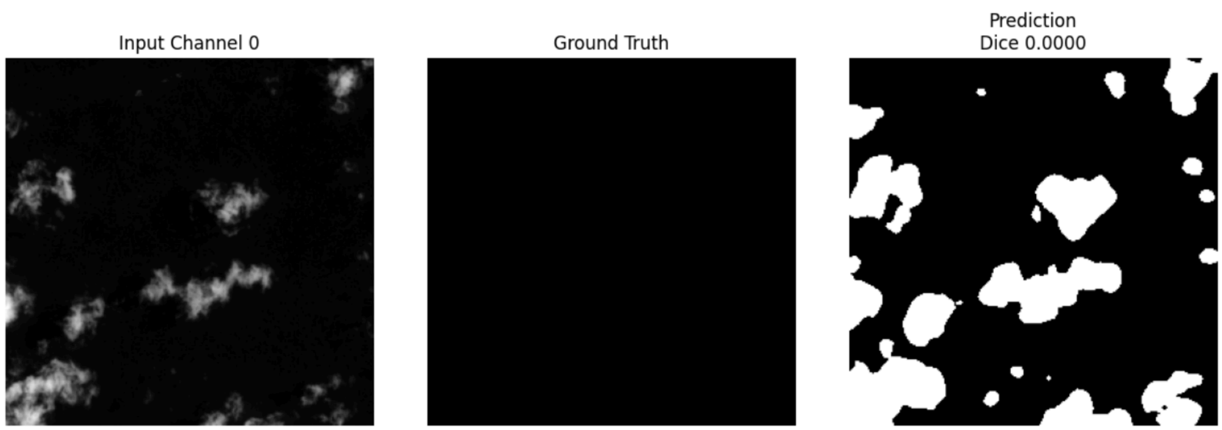
Model	Avg Dice Score	Private Kaggle Dice score
Bayes Classifier	0.5692	Not submitted
Unet++	0.8757	0.54836
Unet	0.8174	Not submitted
DeepLabV3Plus	0.880054	0.56186
FPN	0.8879	0.5252

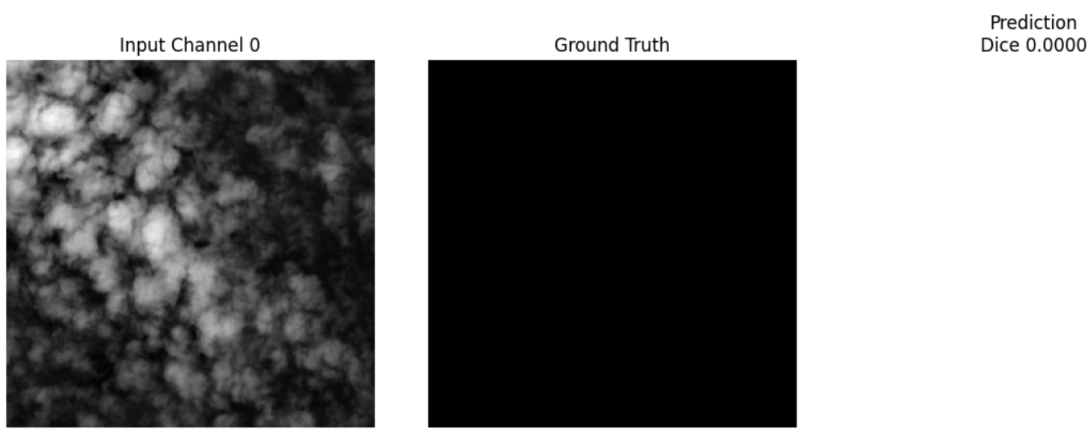
Note: We couldn't compute the Dice Score for the SVM model because it was difficult to parallelize and run efficiently on the GPU but we observed sample results which were not very promising.

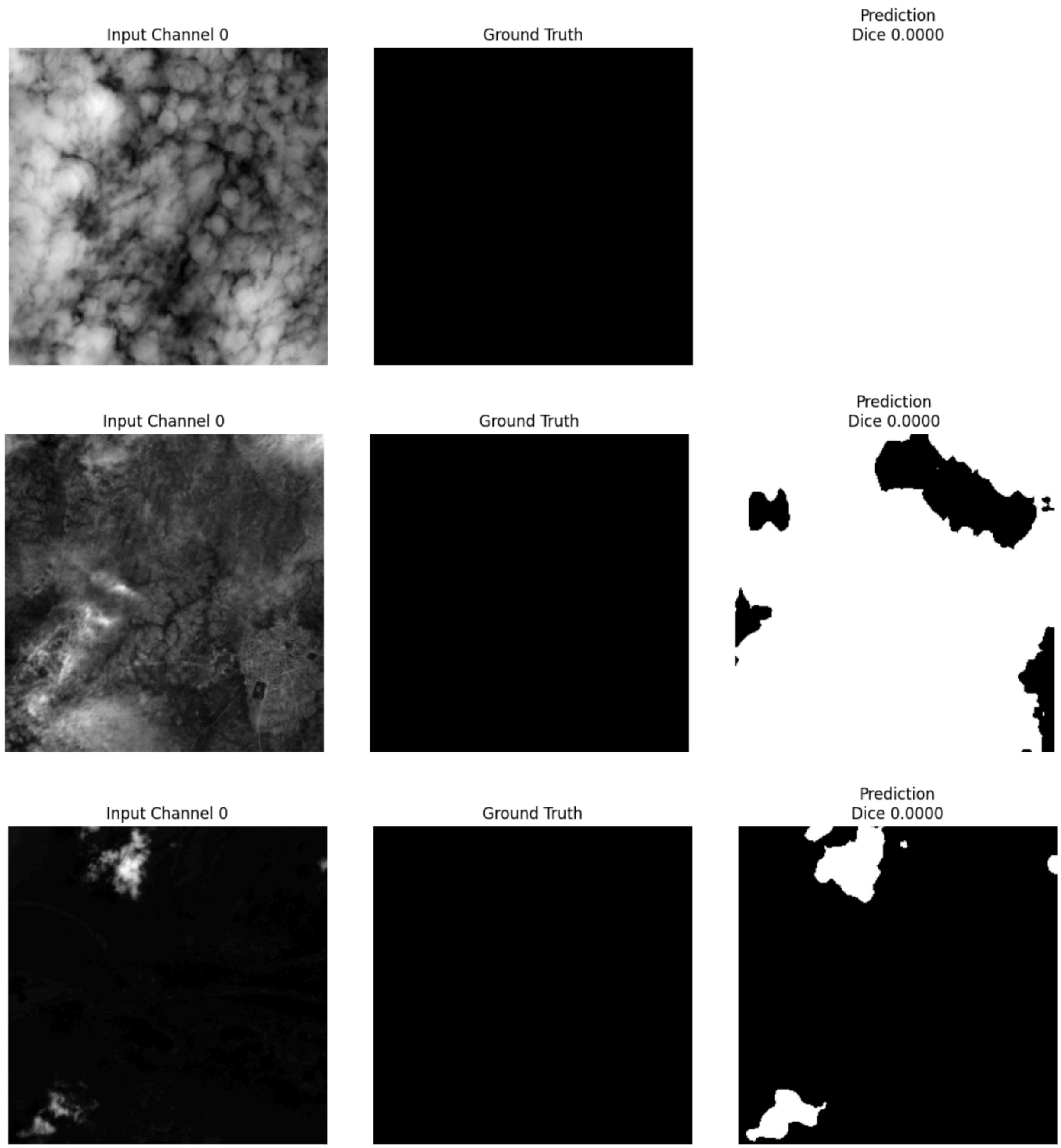
Bad samples

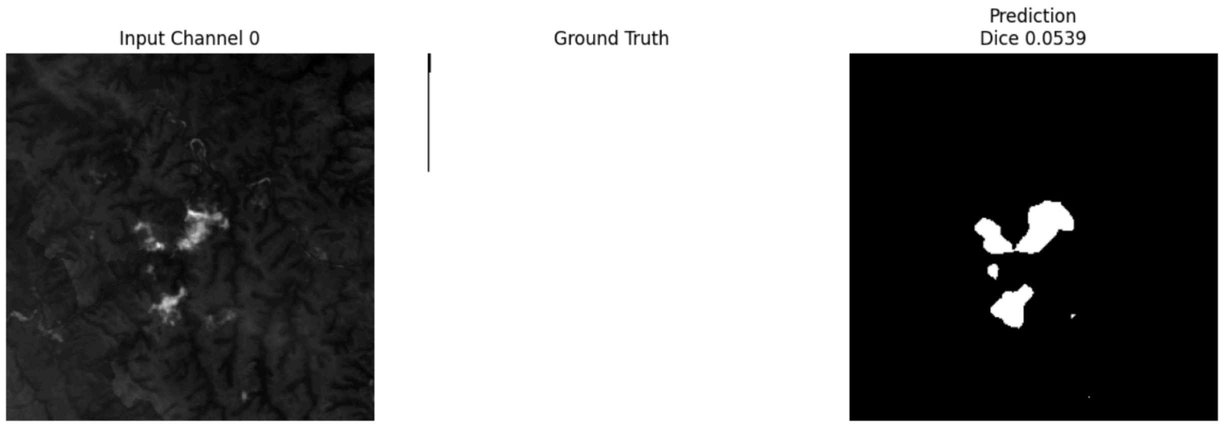
1. **Errors due to mislabeled dataset:**



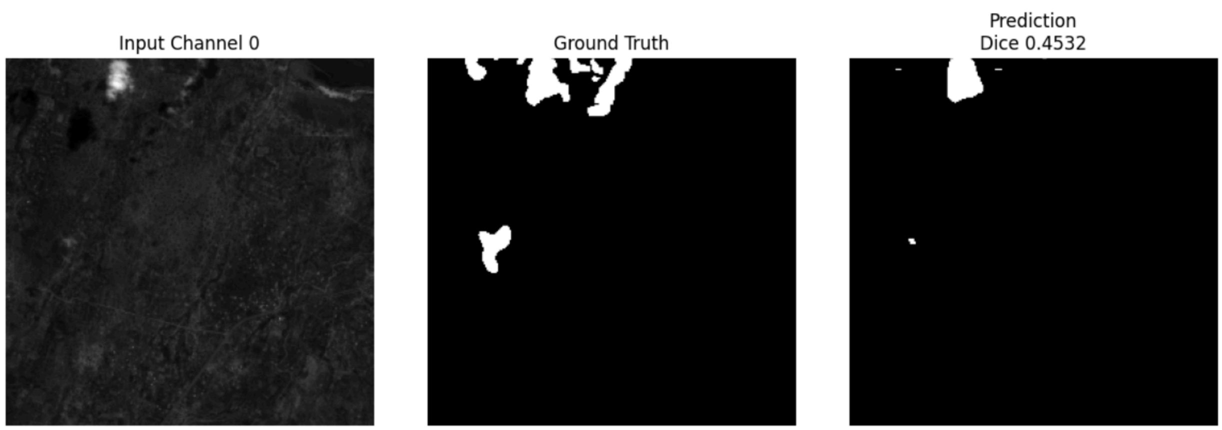


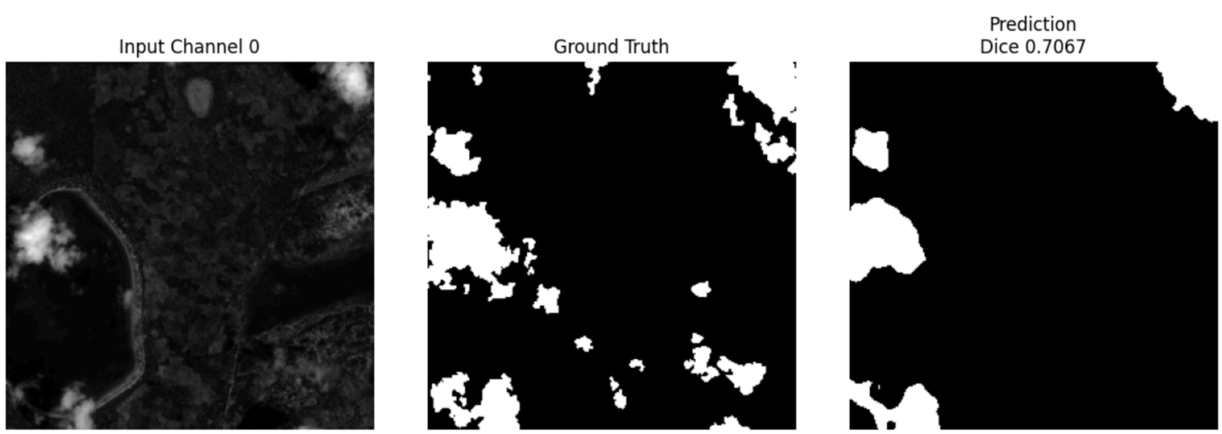



Input Channel 0
Ground Truth
Prediction
Dice 0.0000
Input Channel 0
Ground Truth
Prediction
Dice 0.0000
Input Channel 0
Ground Truth
Prediction
Dice 0.0000

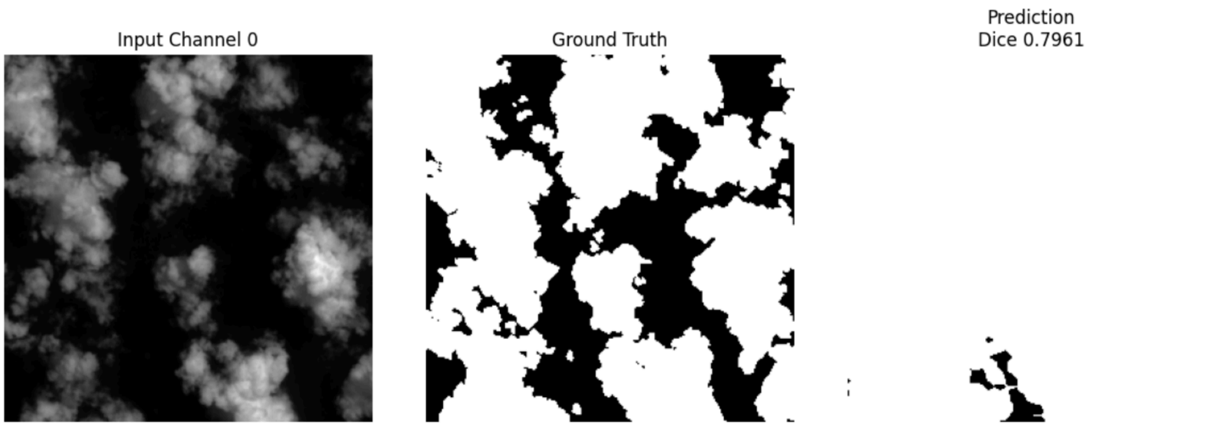


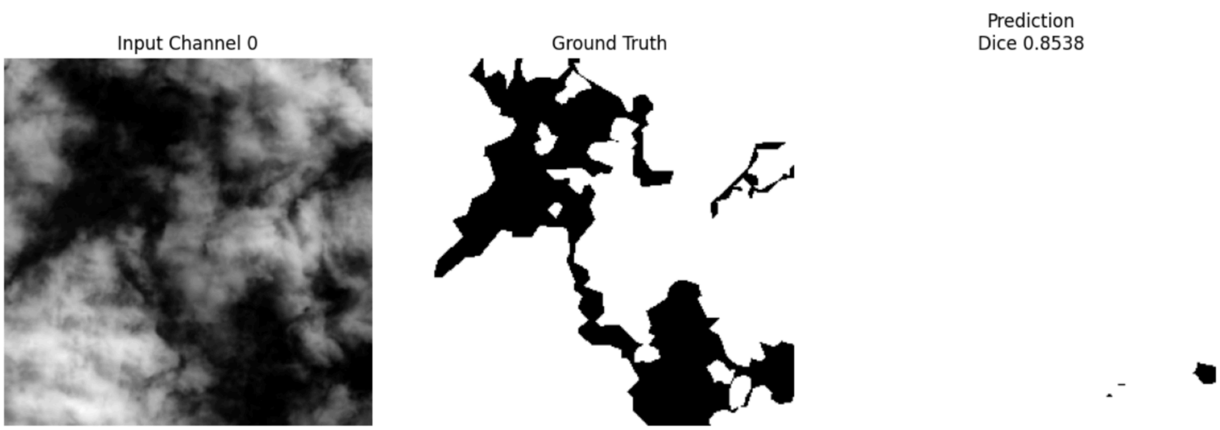
2. Errors due to the lack of prediction details:



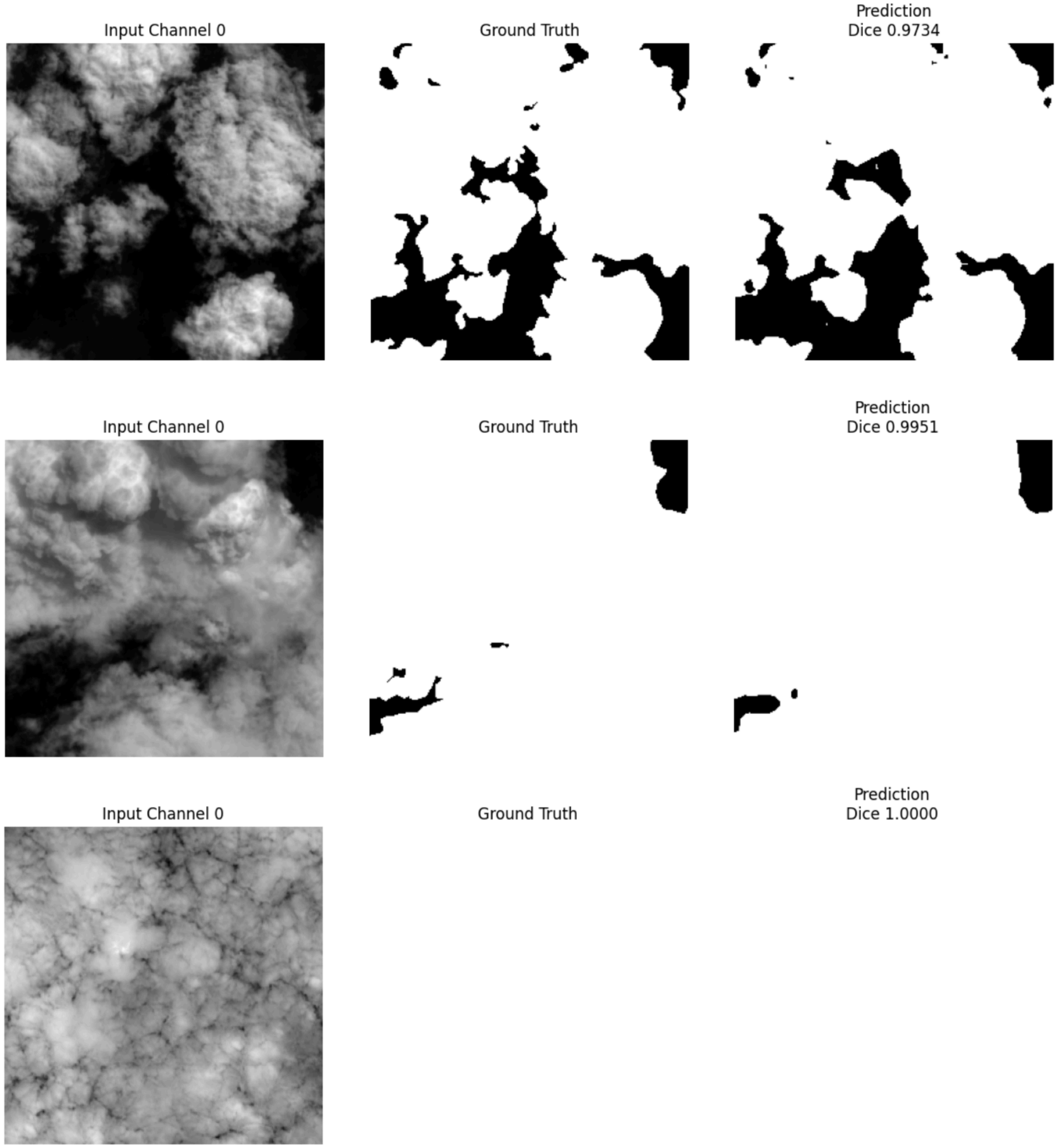


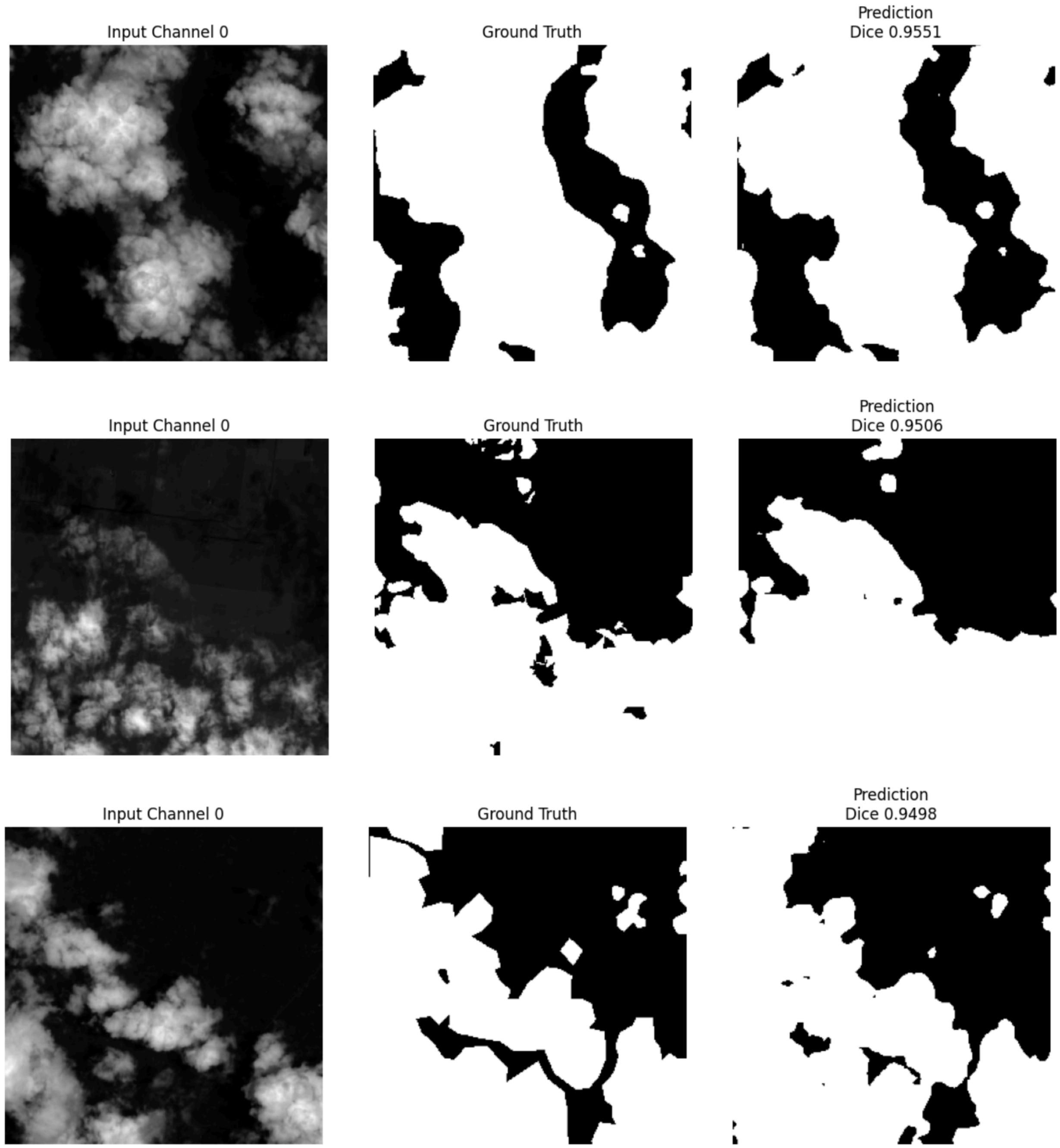
3. **Errors due to the imbalanced dataset since most of the masks are totally white:**

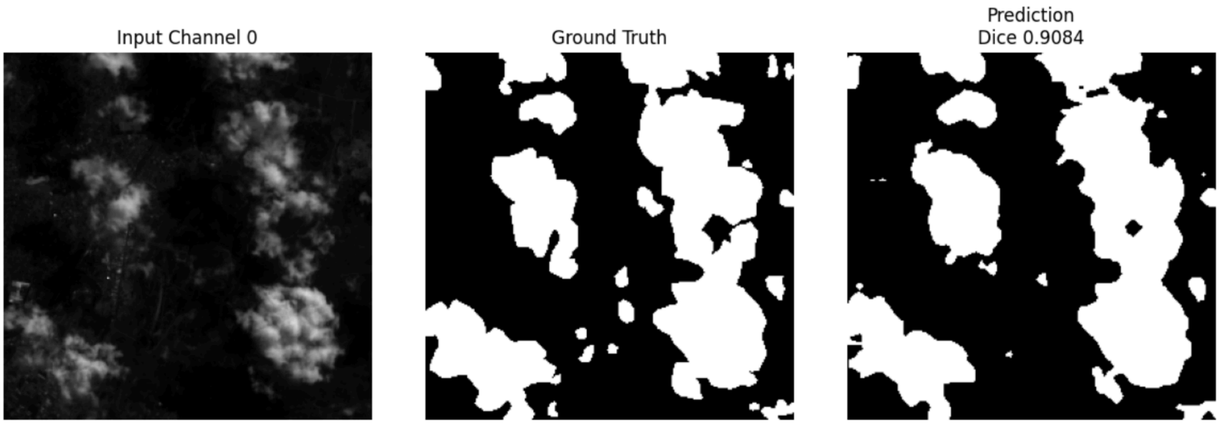




Good samples




Input Channel 0
Ground Truth
Prediction
Dice 0.9551
Input Channel 0
Ground Truth
Prediction
Dice 0.9506
Input Channel 0
Ground Truth
Prediction
Dice 0.9498



Enhancements and Future Work

- ➔ We plan to improve accuracy by further tuning our current U-Net and ResNet-based models and exploring ways to better capture temporal features from image sequences. This could help the model perform more reliably under varying cloud conditions.
- ➔ We plan to apply more data augmentation techniques, like random cropping and adding noise, to help the model generalize better.
- ➔ We also aim to reduce our need for labeled data by exploring semi-supervised learning methods.
- ➔ We plan to optimize the system for faster performance and efficiency, making it more suitable for real-time production use.

Workload distribution table

Name	ID	Workload
Rawan Mostafa Mahmoud	9210423	• Deep Learning Model trials • Data Filtration
Sara Bisheer Fekry	9210453	• Deep Learning Model trials • Data Augmentation
Menna Mohamed Abdelbaset	9211242	• Deep Learning Model trials • Model Evaluation
Fatma Ebrahim sobhy	9210799	• Machine Learning Model Trials • Band Processing and Selection