Final Project Report: Automated Cloud and Shadow Detection in LISS-4 Satellite Imagery

Registration ID: NRCC251024

Date: July 6, 2025

1. Introduction

1.1. Project Overview

This project helps to demonstrate an automated system for detecting clouds and their shadows in LISS-4 satellite imagery. Clouds are a common problem in optical satellite images because they block the view of the Earth's surface, making further analysis difficult. The goal was to create a reliable model that can automatically "mask" out these unusable areas, saving significant time and manual effort, in the process.

1.2. Chosen Methodology

We used a modern deep learning technique called Semantic Segmentation. This allows to segregate areas or segments in the image with different colours. We trained a model to look at a satellite image and colour in every single pixel based on whether it was a cloud, a shadow, or clear ground.

The specific model architecture we used is the U-Net, which is famous for its precision in this kind of task. We experimented with several versions of the U-Net and found that the one using a ResNet50 backbone (its core feature extractor) gave the best results.

2. Data Pre-processing

Before the model could be trained, the raw satellite data needed to be carefully prepared.

2.1. Radiometric Correction

The raw pixel values from a satellite are called Digital Numbers (DN). To make the data scientifically accurate and comparable across different images, we converted these DNs into **Top-of-Atmosphere (TOA) Reflectance**. This important step corrects for variations in the sun's brightness and position. This was achieved by:

- Reading satellite-specific calibration data from provided CSV files (LISS.csv, d_values.csv), created from LISS data shared through excel file.
- Parsing scene-specific metadata (Band_meta.txt) to get values like the sun's elevation angle, gain, and offset.
- Applying the full TOA Reflectance formula to each band of the image.

2.2. Training Data Generation

To teach our model, we needed thousands of labelled examples. We automated this process by:

- **Chipping:** The large, original satellite scenes were cut into smaller 512x512 pixel image chips. To avoid crashing the system due to high memory usage, we used an efficient windowed reading technique, processing each large image in horizontal strips.

- **Pseudo-Labelling:** For each image chip, we created a corresponding "mask" where every pixel was labelled as cloud, shadow, or background based on simple brightness rules. This step created the initial dataset for the model to learn from.

3. Model Training and Evaluation

This is the core phase where the model learns to identify clouds and shadows.

3.1. Model Comparison

To ensure we selected the best possible model, we ran a systematic experiment comparing three different configurations:

1. U-Net with a **ResNet34** backbone
2. U-Net with a **ResNet50** backbone
3. U-Net with an **EfficientNet-B4** backbone

Each model was trained for 15 epochs, and its performance was tracked on a separate validation dataset.

3.2. Evaluation Metrics

The performance of each model was measured using a standard set of metrics: **IoU (Intersection over Union)**, **F1-Score, Precision, Recall**, **Accuracy.**

The comparison is stored in a csv file which is attached in the deliverables.

3.3. Results

The **U-Net with a ResNet50 backbone** was the clear winner, achieving the highest performance across the board. The final validation scores for this model were:

- **Best Validation IoU:** 0.8504
- **Best Validation F1-Score:** 0.9192
- **Final Validation Accuracy:** 98.50%

The training progress was plotted on graphs, which showed a healthy learning curve where the model consistently improved without significant overfitting.

4. Inference and Final Outputs

The best-performing model (U-Net with ResNet50) was then used to process the final, unseen test set.

4.1. Inference Process

For each large image in the test set, a sliding-window technique was used. The model analyzed the image tile by tile, and the results were stitched back together to create a single, full-resolution prediction mask. This process also involved the full TOA reflectance correction using each test scene's unique metadata.

4.2. Final Deliverables

For each input test scene, the script generated two industry-standard GIS files:

1. **Georeferenced GeoTIFF (.tif):** An 8-bit mask file where pixels are labeled (0: No-Cloud, 1: Cloud, 2: Shadow). This file is perfectly aligned with the original satellite image and can be used as an overlay in GIS software.
2. **ESRI Shapefile (.shp):** A vector file containing precise polygons that outline every individual cloud and shadow. This is useful for calculating the area covered by clouds or for other spatial analyses.