

# Automatic Cloud and Shadow Mask Generation from Resourcesat-2/2A LISS-4 Satellite Images

Team ID: NRCC251050

Dev Kaushal (ep23bt004@iitdh.ac.in)  
Aniruddh Pandav (ee23bt031@iitdh.ac.in)  
Surya Prakash S (cs23bt068@iitdh.ac.in)  
Shreya S Bhat (cs23bt070@iitdh.ac.in)

Indian Institute of Technology Dharwad

**Abstract**—Clouds and shadows significantly hinder the usability of optical remote sensing images by obscuring land features. This paper presents a robust preprocessing and machine learning-based approach for detecting clouds and their shadows in Resourcesat-2/2A LISS-4 imagery, which lacks a SWIR band. The pipeline includes DN to TOA conversion, solar geometry correction, brightness thresholding, and supervised segmentation using deep learning. Evaluation on diverse terrains shows promising generalization with high accuracy, IoU, and F1 scores. The final output includes georeferenced TIFF masks and shapefiles for deployment in GIS workflows.

**Index Terms**—Cloud Masking, TOA Reflectance, LISS-4, Shadow Detection, Remote Sensing, Semantic Segmentation, Deep Learning, Satellite Image Preprocessing

## I. INTRODUCTION

Clouds and their associated shadows represent one of the most persistent challenges in the analysis of optical satellite imagery. These atmospheric artifacts obscure large portions of the Earth's surface, leading to incomplete or inaccurate information extraction. In cloud-covered regions, surface features are entirely hidden, while shadows introduce confusion with naturally dark features such as water bodies, building shadows, and mountain valleys. These issues critically impair downstream applications such as land cover classification, vegetation health assessment, urban expansion mapping, and climate monitoring.

Resourcesat-2 and Resourcesat-2A, launched by ISRO, are high-resolution Earth observation satellites equipped with the LISS-4 (Linear Imaging Self Scanner-4) sensor. LISS-4 operates in three spectral bands—Green (BAND2), Red (BAND3), and Near-Infrared (NIR, BAND4)—with a spatial resolution of 5.8 meters. These bands are essential for extracting vegetation indices, soil moisture content, and land surface characteristics.

However, LISS-4 lacks the Short-Wave Infrared (SWIR) band, which is traditionally used in cloud and snow discrimination due to its unique absorption characteristics in high-albedo and moist areas. This absence makes the task of delineating clouds from spectrally similar features like snow, sand dunes, and brightly lit rooftops significantly more difficult. Similarly, identifying shadows without thermal or SWIR bands requires

careful modeling based on brightness, spectral signatures, and spatial morphology.

In this work, we present a robust pipeline for automatic detection and masking of clouds and their shadows from LISS-4 imagery. The proposed approach includes radiometric corrections (converting Digital Numbers to Top-of-Atmosphere reflectance), solar geometry correction, spectral and brightness-based thresholding, and semantic segmentation using supervised deep learning. The final outputs include 8-bit georeferenced TIFF masks aligned with the original images, and ESRI-compatible shapefiles for integration into GIS platforms. Our approach attempts to be generalizable across varied terrains.

This work addresses not only the technical challenges posed by the absence of the SWIR band but also focuses on developing a scalable and reproducible method that can aid large-scale EO (Earth Observation) analytics.

## II. OBJECTIVE

The goal of this project is to design a generalizable, efficient pipeline that classifies each pixel of LISS-4 images into: **No Cloud (0)**, **Cloud (1)**, and **Shadow (2)**. Key goals include:

- DN to TOA reflectance conversion using calibration parameters and solar geometry.
- Semi-automated training data creation using spectral indices, morphological filters, and QGIS-based manual labeling.
- Training a supervised segmentation model and evaluating it using standard metrics (IoU, F1, etc.).
- Exporting predictions as 8-bit georeferenced TIFFs and ESRI shapefiles.
- Generalizing the model across seasons and terrains.

## III. DATASET DESCRIPTION AND LABELLING

The dataset comprises LISS-4 multispectral images acquired from Resourcesat-2 and Resourcesat-2A platforms. These images cover a wide range of geographic and seasonal variations, including deserts, forests, snow-covered regions, croplands, and urban areas, ensuring that the model can generalize well across diverse terrains.

Each dataset folder includes:

- **Multispectral TIFF images** for BAND2 (Green), BAND3 (Red), and BAND4 (NIR), which serve as the core input for the model.
- A **metadata text file** (BAND\_META.txt) containing key acquisition parameters such as Sun Elevation Angle and Date of Pass.
- An **Earth-Sun distance table** in Excel format, which is used to compute Top-of-Atmosphere (TOA) reflectance based on the acquisition date.

However, one major challenge encountered in this project was that the dataset was entirely **unlabelled**. No ground truth cloud or shadow masks were provided, which made supervised learning difficult in the initial phases.

To overcome this, a hybrid labelling strategy was adopted. We used a combination of threshold-based heuristics applied to TOA reflectance values (e.g., brightness and NIR suppression) to generate initial pseudo-labels. These were then refined using manual annotation in QGIS, an open-source Geographic Information System. This iterative process helped us build a high-quality training dataset for model development.

#### A. Labelling Strategy

Given the absence of labelled data, we constructed training masks through a two-stage strategy:

- 1) **Automatic Pre-labeling:** Initial masks were generated by applying a combination of spectral indices and thresholding techniques to the TOA reflectance images. Specifically, we used:

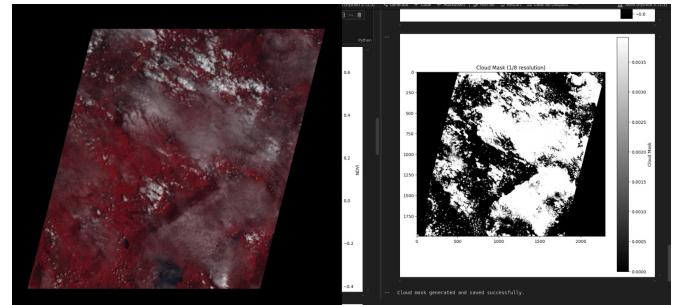
- **NDVI (Normalized Difference Vegetation Index)** to distinguish vegetated regions and suppress false positives in cloud or shadow detection.
- **Brightness Index (BI)** to identify highly reflective regions, which often correspond to clouds or snow.
- **Raw TOA reflectance values** from BAND2 (Green), BAND3 (Red), and BAND4 (NIR), which were averaged or used in ratios to characterize brightness or darkness levels in specific spectral regions.

Cloud pixels typically exhibit high reflectance across all bands and high BI, whereas shadows are characterized by low reflectance, particularly in the NIR band and reduced NDVI. These heuristic rules were applied to each scene individually, and thresholds were tuned empirically based on terrain, season, and illumination conditions.

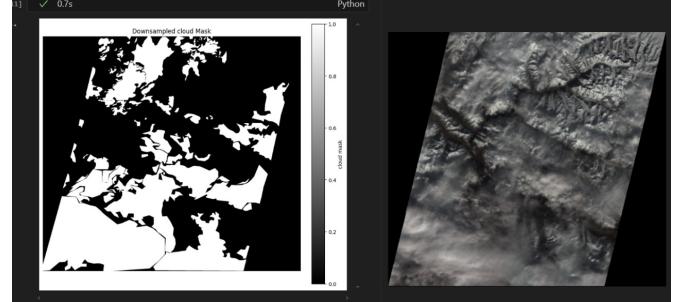
- 2) **Manual Refinement in QGIS:** The pre-labelled masks were imported into QGIS for inspection. Using the composite RGB imagery and false-color combinations (e.g., NIR-Red-Green), manual polygon annotations were created to correct false positives and negatives. This step was crucial for accurately labelling ambiguous regions like snowfields, thin clouds, and terrain shadows.

Figure 1 shows examples of the manual labelling process in QGIS, highlighting the refinement of cloud and shadow masks over challenging terrain.

This approach, although labor-intensive, was necessary to create a reliable ground truth dataset for supervised training especially for desert, snow, and other terrains.



(a) Cloud scene and corresponding mask



(b) Snow-like terrain and manual labeling in QGIS

Fig. 1: Examples of annotation strategies. Top: automatic cloud masking based on brightness and TOA thresholds. Bottom: manual polygon-based annotation in QGIS to handle complex snow scenarios.

## IV. PREPROCESSING PIPELINE

The following steps are applied:

- 1) **Band Selection:** Load BAND2, BAND3, and BAND4 TIFFs.
- 2) **Solar Geometry Extraction:** Parse metadata to extract Sun Elevation and Date-of-Pass; compute DOY and retrieve Earth-Sun distance.
- 3) **Radiance Conversion:** DN values converted to spectral radiance using:

$$L = L_{\min} + \left( \frac{DN}{1023} \right) (L_{\max} - L_{\min})$$

- 4) **TOA Reflectance:**

$$\rho = \frac{\pi \cdot L \cdot d^2}{E_{\text{sun}} \cdot \cos(\theta_s)}$$

## V. TRAINING DATASET AND CONFIGURATION

- **Framework:** PyTorch
- **Optimizer:** Adam with learning rate  $1 \times 10^{-4}$
- **Batch Size:** 8
- **Epochs:** 100
- **Hardware:** NVIDIA RTX 3060 GPU (12GB VRAM)
- **Cloud Patches:** 204,553 labeled patches of size  $128 \times 128$
- **Shadow Patches:** 57,811 labeled patches of size  $128 \times 128$

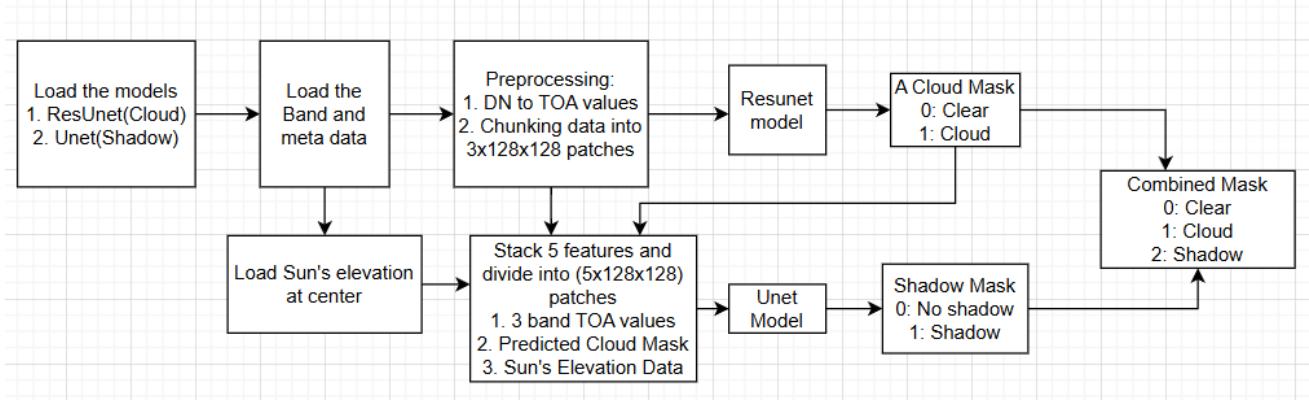


Fig. 2: End-to-end cloud and shadow segmentation pipeline

## VI. METHODOLOGY

This section outlines the complete pipeline for segmentation of clouds and shadows from satellite images. The process includes TOA reflectance computation, patch-based dataset preparation, model architectures, training strategy, and evaluation methodology. The pipeline can be seen in Figure 2.

### A. Patch Extraction and Dataset Preparation

The TOA reflectance arrays are divided into smaller fixed-size patches for training. A sliding window with size  $128 \times 128$  and optional overlap is used to chunk large scenes into manageable patches.

#### a) Cloud Dataset::

- Input: (TOA\_B2, TOA\_B3, TOA\_B4) → shape (3, 128, 128)
- Label: Binary cloud mask where 0 = clear, 1 = cloud

b) *Shadow Dataset*:: After cloud prediction, the resulting cloud mask is stacked with TOA bands and sun elevation to form the input for shadow detection:

- Input: (TOA\_B2, TOA\_B3, TOA\_B4, Cloud Mask, Sun Elevation) → shape (5, 128, 128)
- Label: Binary shadow mask where 0 = clear/cloud, 1 = shadow

Both datasets are saved as .npy files for efficient loading.

### B. Model Architectures

a) *Cloud Segmentation Model: ResUNet*: We use a ResUNet architecture for binary cloud segmentation. The encoder is ResNet34 pre-trained on ImageNet; the decoder follows U-Net structure with skip connections.

b) *Shadow Segmentation Model: Custom U-Net*: We use a custom U-Net architecture for shadow prediction, with 5-channel input and dropout for regularization.

### C. Training Strategy

For both models:

- 70% of patches used for training, 15% for validation, 15% for testing
- Optimizer: Adam with learning rate  $1 \times 10^{-4}$

- Batch size: 16

- Epochs: 10

- Checkpoints saved after each epoch

Training loss and validation accuracy are monitored across epochs to ensure convergence. Model with best validation performance is selected for inference.

### D. Evaluation Metrics

Both models are evaluated using the following metrics on the held-out test set:

- **Accuracy**: Overall pixel-wise correctness
- **Precision**: Fraction of predicted positive pixels that are correct
- **Recall**: Fraction of actual positive pixels correctly predicted
- **F1 Score**: Harmonic mean of precision and recall
- **IoU (Jaccard Index)**: Overlap between predicted and ground truth regions
- **Confusion Matrix**: Aggregated TP, FP, TN, FN across pixels

Metrics are computed separately for cloud and shadow classes. Shadow metrics are evaluated only in the second stage using the U-Net predictions, and cloud predictions from ResUNet are reused for that step.

### E. Training Curves and Performance Metrics

This section presents the training behavior and final evaluation metrics of both the cloud segmentation and shadow segmentation models.

1) *Training Curves*: The training and validation losses were monitored across 10 epochs for both models.

- The time taken for cloud model to train: 934 minutes.
- The time taken for shadow model to train: 234 minutes.

The plots shown in Figures 4a and 4b illustrate how the models progressively learned from the dataset. Both models demonstrate stable convergence without signs of overfitting.

- The best performing model chosen for cloud Segmentation: 9th epoch(8th in the image)
- The best performing model chosen for Shadow Segmentation: 9th epoch(8th in the image)

TABLE I: Model Architecture Components

**(a) ResUNet for Cloud Segmentation**

Component	Details
Input	Patch (3, 128, 128) with BAND2, BAND3, BAND4
Encoder	ResNet34 (ImageNet), 4 downsampling blocks
Decoder	U-Net-style decoder with upsampling
Skip Connections	Encoder → decoder feature concatenation
Output Layer	$1 \times 1$ conv → logits (2, 128, 128)
Activations	ReLU, CrossEntropyLoss (implicit softmax)
Parameters	24,436,514

**(b) Custom U-Net for Shadow Segmentation**

Component	Details
Input	Patch (5, 128, 128): TOA B2, B3, B4 + cloud mask + sun elevation
Encoder	3-level encoder with DoubleConv + MaxPool
Decoder	Symmetric decoder with upsampling
Skip Connections	Encoder → decoder feature concatenation
Dropout	0.2 rate after each conv block
Output Layer	$1 \times 1$ conv → logits (2, 128, 128)
Loss	Weighted CrossEntropyLoss
Parameters	7,704,194

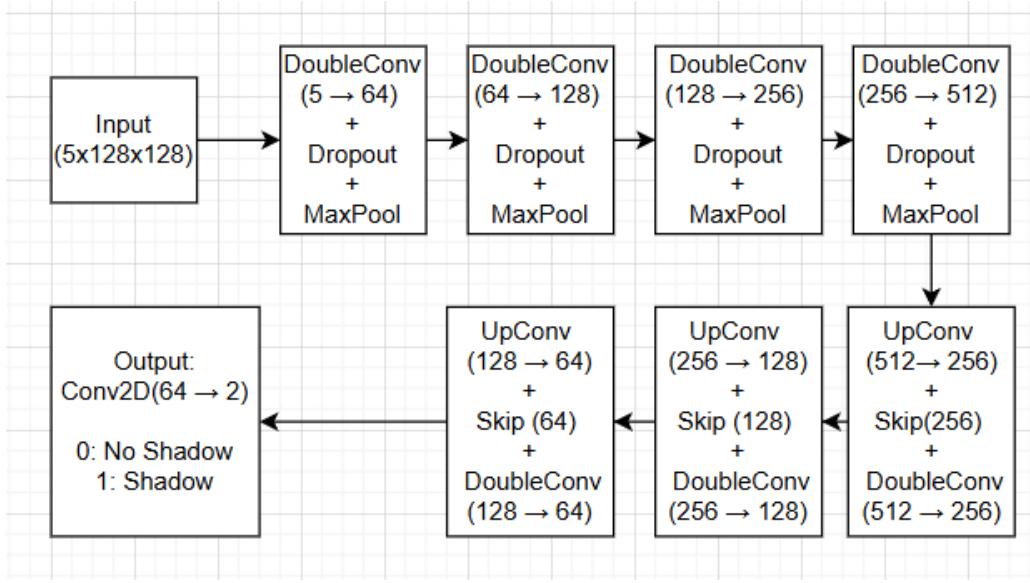
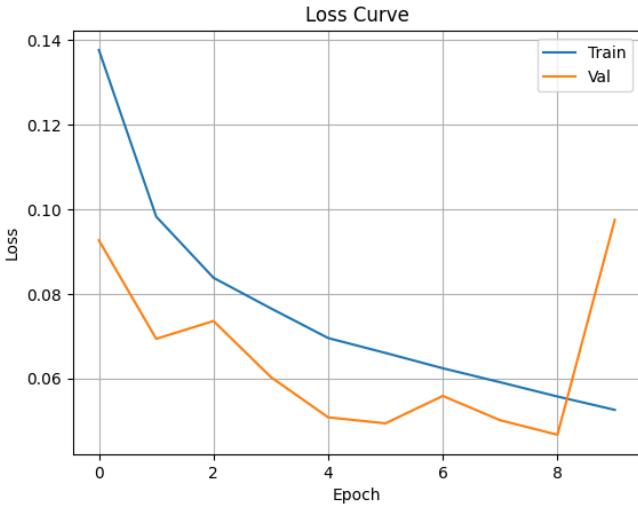
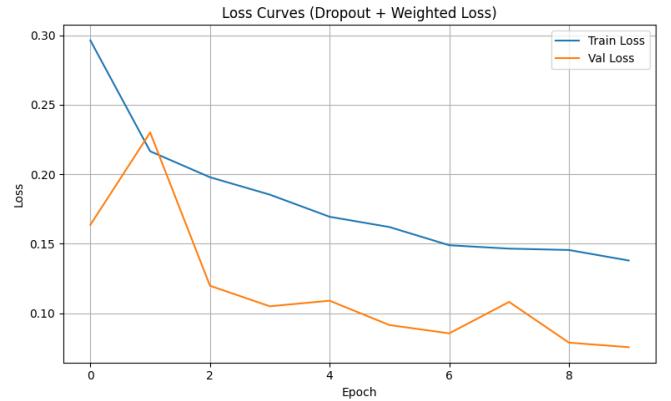


Fig. 3: Shadow model architecture



(a) ResUNet (Cloud Segmentation)



(b) U-Net (Shadow Segmentation)

Fig. 4: Training and Validation Loss Curves for Cloud and Shadow Segmentation Models

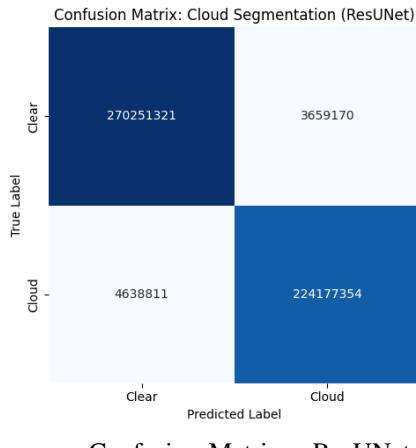
Fig. 5: Evaluation Metrics and Confusion Matrices

TABLE II: Pixel-wise metrics for cloud segmentation using ResUNet

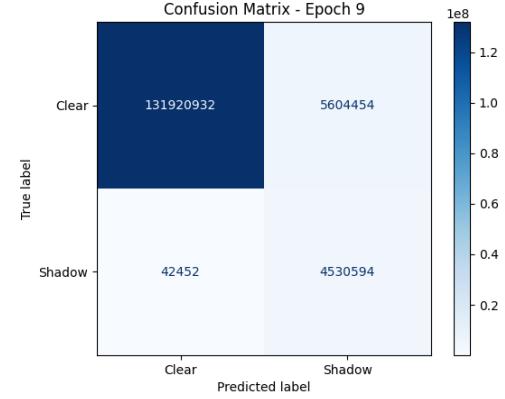
Metric	Clear	Cloud
Precision	1.00	0.92
Recall	0.93	0.99
F1 Score	0.96	0.96
IoU	—	0.918
<b>Overall Acc</b>		0.96

TABLE III: Pixel-wise metrics for shadow segmentation using U-Net

Metric	Clear	Shadow
Precision	1.00	0.45
Recall	0.96	0.99
F1 Score	0.98	0.62
<b>Overall Acc</b>		0.96



Confusion Matrix – ResUNet



Confusion Matrix – U-Net

2) *Evaluation Metrics:* After training, both models were evaluated on the test set. Table II and Table III summarize the pixel-level classification performance using Accuracy, Precision, Recall, and F1 Score for the cloud and shadow models, respectively.

## VII. RESOURCES USED

- **System:** Windows 11, Ryzen 7, 16 GB RAM
- **Software:** Python 3.10, PyTorch, Rasterio, QGIS
- **Model Training time:**
  - **Cloud Model:** 934 minutes
  - **Shadow Model:** 234 minutes

## VIII. ANALYSIS

- **Cloud vs Snow Confusion:** Our cloud model performs well on snow covered terrain as can be seen in the following image.

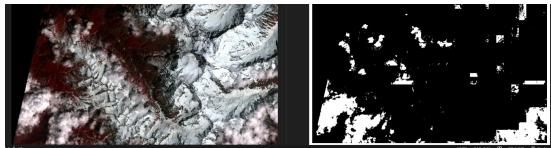


Fig. 6: Model correctly classifying snow as non-cloud

- **Shadow Detection:** Shadows over water and building regions remain challenging. By looking at the training

curves it looks the model is clearly learning. If the quality of the training dataset is improved the model is expected to perform better.

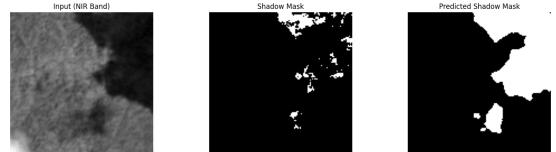


Fig. 7: Model outperforming the thresholding method

As can be seen in the given image, the thresholding wasn't able to identify the shadow in that region but our shadow model did. There is a huge class imbalance as our training dataset just had 3.09% pixel coverage.

- **Overfitting:** Minimal, based on consistent validation loss and performance.

## IX. INNOVATION

- We used two different models for cloud and shadow detection. The output of the cloud segmentation model was passed as an input to the shadow model to capture the spatial relationship between clouds and their shadows.
- Including the cloud mask and sun elevation angle as explicit input channels helped the model reason better

with physical priors, improving prediction accuracy for shadows.

- Using separate models helped reduce the negative impact of class imbalance. A dedicated shadow model performs better than joint models on rare classes.
- We used inverse frequency weighting in the loss function to emphasize the minority shadow class and mitigate bias toward dominant clear pixels.
- Large GeoTIFFs were converted into aligned 128x128 patches with spatial consistency preserved. Patches containing no information were excluded to optimize training efficiency.

## X. CONCLUSION AND FUTURE WORK

The proposed model performs robustly across a wide range of LISS-4 scenes. Results are suitable for downstream geospatial workflows, and outputs are provided in both TIFF and ESRI formats. Here is a list of things that we wanted to experiment with but couldn't due to time and computational limitations:

- Incorporating synthetic data for rare terrain types.
- The shadow annotations in the provided training dataset were often ambiguous and inconsistent and less in size. In several instances, it was difficult even for a human observer to distinguish terrain from shadow. Simple thresholding techniques also failed due to low contrast and labeling noise. In contrast, the test dataset had visibly better-defined shadows, indicating a domain shift. This discrepancy is illustrated in the image below. We believe that using a more accurately labeled and consistent dataset for training would significantly improve model performance.

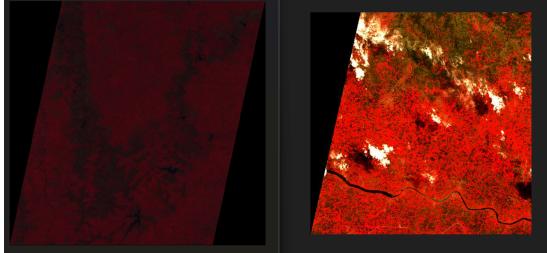


Fig. 8: Training image vs test images

- The current shadow segmentation model uses only the cloud mask corresponding to the same patch as input. However, shadows can be cast by clouds located in adjacent patches. Incorporating predicted cloud masks from neighboring patches as additional context could help the model better learn spatial dependencies and improve generalization, especially near patch boundaries.
- A larger patch size such as 512x512 could be used for shadow segmentation instead of 128x128. This would allow the model to capture more contextual information and spatial relationships between clouds and shadows within a single patch, potentially improving detection accuracy—especially for elongated or soft-edged shadows.

- Our cloud model prioritizes precision by focusing only on dense, well-defined clouds during training. As a result, hazy or thin clouds—often ignored during manual labeling—are sometimes missed by the model. The performance can be further improved by fine-tuning the model on such ambiguous edge cases, enabling better generalization to real-world scenarios with varying cloud opacity.
- Exploring lightweight models for on-device inference.

## ACKNOWLEDGEMENTS

We thank the organizers of the ISRO EO Challenge for providing the dataset and guidance. We also acknowledge the use of open-source software including PyTorch, Rasterio, and QGIS.

## REFERENCES

- [1] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in *Proc. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.
- [2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [3] Z. Zhang, Q. Liu, and Y. Wang, “Road extraction by deep residual U-Net,” *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, 2018.
- [4] J. E. Jeppesen, S. R. Shah, and B. L. Bingham, “Cloud-Net: An end-to-end cloud detection algorithm for Landsat 8 imagery,” *Remote Sensing*, vol. 11, no. 4, p. 467, 2019.
- [5] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl, “Multilabel classification of satellite images with pixel-wise explanations,” in *Proc. IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2019, pp. 1809–1812.
- [6] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in \**Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)\*, 2015, pp. 3431–3440.*
- [7] L. Oreopoulos, M. J. Wilson, and T. Várnai, “Implementation on Landsat Data of a Simple Cloud-Mask Algorithm Developed for MODIS Land Bands,” *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 597–601, Jul. 2011, doi: 10.1109/LGRS.2010.2095409.
- [8] Y. Luo, A. P. Trishchenko, and K. V. Khlopenkov, “Developing clear-sky, cloud and cloud shadow mask for producing clear-sky composites at 250-meter spatial resolution for the seven MODIS land bands over Canada and North America,” *Remote Sensing of Environment*, vol. 112, no. 12, pp. 4167–4185, 2008, doi: 10.1016/j.rse.2008.06.010.
- [9] S. Ozkan, M. Efendioglu, and C. Demirpolat, “Cloud Detection from RGB Color Remote Sensing Images with Deep Pyramid Networks,” in *Proc. IGARSS 2018 – IEEE Int. Geoscience and Remote Sensing Symposium*, Valencia, Spain, 2018, pp. 6939–6942, doi: 10.1109/IGARSS.2018.8519570.
- [10] “How to convert Landsat DNs to Top of Atmosphere (TOA) Reflectance,” Yale Center for Geospatial Solutions, accessed July 6, 2025. <https://yceo.yale.edu/how-convert-landsat-dns-top-atmosphere-toa-reflectance>.