**A Report On**

An Offline EO Data Processing Challenge  
using Open-Source Packages

Automatic CLOUD and SHADOW mask generation

from Resourcesat-2/2A Liss4 Satellite Images

Registration Id: NRCC251067

Bhawuk Arora, Poornima College of Engineering

Mukul, Poornima College of Engineering

# **1. Objective**

The primary objective of this project is to design and implement an automated and generalized system for the accurate identification and segmentation of cloud and shadow regions in optical satellite imagery. The system uses deep learning techniques and is capable of processing LISS-4 (Resourcesat-2/2A) RGB imagery to generate cloud and shadow masks, and outputs georeferenced raster and shapefile formats for downstream Earth observation applications.

2. Dataset Description

To Include:

Data set Ids (e.g.Resourcesat-2/2A)

Number of samples: train, validation, test

Preprocessing steps (DNto TOA , normalization, sun angle correction )

Satellite: Resourcesat-2/2A, LISS-4 sensor Bands: Band 2 (Red), Band 3 (Green), Band 4 (NIR) File Format: GeoTIFF Samples Used: Training: 20 scenes Testing: 10 scenes (provided later

Preprocessing Steps: -

Convert DN to TOA reflectance using metadata values - Adjust for sunlight angle using sun elevation - Normalize values to [0, 1] - Generate 3 class masks manually (No Cloud = 0, Cloud = 1, Shadow = 2

3. Model Architecture / Algorithm Pipeline

Details of your methodology / model being used :

If and as applicable

Number and types of layers (Conv, Pooling, FC)

Activation functions

Use of transfer learning (e.g., ResNet, VGG16)

Total parameters

Optional: Include a diagram or table of model architecture.

Model Used: U-Net U-Net is a convolutional neural network specifically designed for biomedical image segmentation and is well-suited for pixel-wise classification. Architecture Summary: - Encoder Path: - Conv2D → BatchNorm → ReLU ×2 → MaxPool - Filters: [64, 128, 256, 512] • • • • • Decoder Path: Upsample (Transposed Conv) → Concatenate Skip Connection → Conv2D ×2 Filters: [512, 256, 128, 64] Output Layer: Conv2D (1×1 kernel) → 3 Output Classes Key Details: - Activation: ReLU - Final Activation (Inference): Softmax - Loss Function: CrossEntropyLoss- Skip Connections: Allow gradient flow and spatial recovery - Total Trainable Parameters: ~31 million



4. Training Configuration( if applicable) , Assumptions and Constraints

Framework used (TensorFlow, PyTorch, etc.)

Loss function

Optimizer (e.g., Adam, SGD)

Learning rate

Number of epochs

Batch size

Hardware used (CPU/GPU)

• • • • • • • Software: Python, PyTorch, Jupyter Notebook Run On: Windows 10 (4 GB RAM), CPU only Loss Function: CrossEntropyLoss Optimizer: Adam Learning Rate: 0.0001 Epochs: 20 Batch Size: 1 or 2 (to avoid memory issues)

5. Resources used ( compute Hardware and software packages )

OS , CPU, RAM ,…

Open source packages / COTS packages

• • • • • • Windows 10 (64-bit) Python 3.10 (Anaconda) Jupyter Notebook Libraries: rasterio, numpy, opencv-python matplotlib, torch, torchvision shapely, geopandas (for shapefiles)

5. Evaluation Metrics

Classification: Accuracy, Precision, Recall, F1-Score, Confusion Matrix

6. Results

a. Quantitative

Metric values on test set

Table of class-wise performance

Confusion matrix

b. Visual Outputs

Sample predictions vs. ground truth

Heatmaps or Class Activation Maps (CAM)

Overlay masks (for segmentation tasks)

c. Model training graphs on training

7. Analysis

Insights: Which classes performed well/poorly?

Common errors (misclassifications, false positives)

Overfitting/underfitting evidence (training vs. validation loss curves)

Comparison with baseline models (if any)

8. Conclusion and future improvements

Summary of performance

Suitability for deployment/use

Limitations