Automatic CLOUD and SHADOW mask generation from Resourcesat-2/2A Liss4 Satellite Images

(An Offline EO Data processing Challenge using Open source packages)

Registration ID: NRCC251168

All Team Members Details:

Shreyas Goswami, Bhoomicam Pvt. Ltd., TiDES, IIT Roorkee)
Ishfaqul Haque, IIT Roorkee
Akash K, IIT Roorkee
Akash Pandey, IIT Roorkee

1. Objective

To develop a classifier model to identify and mask out cloud, cloud shadows, and no-cloud classes from Resourcesat-2/2A LISS 4 satellite product (L2).

2. Dataset Description

Data set Ids:

Sl. No.	Product ID	OTS Product ID
1	248465831	R2F01SEP2024069380009900053SSANSTUC00GTDC
2	2477501261	R2F05AUG2024068996010800054SSANSTUC00GTDD
3	2461261111	R2F15JUN2024068272009300044SSANSTUC00GTDD
4	2483602441	R2F27AUG2024069309009800054SSANSTUC00GTDA
5	2448113381	R2F30APR2024067624010300067SSANSTUC00GTDB
6	2450769191	RAF08MAY2024038491010700053SSANSTUC00GTDA
7	2487391111	RAF10SEP2024040267010800054SSANSTUC00GTDC
8	2487824261	RAF16SEP2024040353009000054SSANSTUC00GTDA
9	248073641	RAF18AUG2024039941008900053SSANSTUC00GTDB
10	2471116221	RAF19JUL2024039514010700054SSANSTUC00GTDD
11	2463262291	RAF22JUN2024039131009200052SSANSTUC00GTDA
12	248379371	RAF28AUG2024040083009100054SSANSTUC00GTDB
13	2475771291	RAF30JUL2024039671009000054SSANSTUC00GTDA
14	2511545171	R2F01JAN2025071113010900054SSANSTUC00GTDD
15	2518244301	R2F02FEB2025071568010100068SSANSTUC00GTDB
16	251448921	R2F18JAN2025071363009800054SSANSTUC00GTDA
17	2410431171	RAF13NOV2024041177009200044SSANSTUC00GTDC
18	2552552251	RAF18MAY2025043819011000054SSANSTUC00GTDC

19	253565761	RAF23MAR202504302400940004５SSANSTUC00GTDC
20	2519118331	RAF25JAN202504222000970005５SSANSTUC00GTDB

Number of samples: train, validation, test:

A total of 20 images were used to generate training samples, which were split into 80/20 for training and validation. Model performance was assessed using an independent test set derived from 13 separate images.

Training Set:	Validation Set:
Class 0: 8930 samples	Class 0: 2233 samples
Class 1: 8302 samples	Class 1: 2075 samples
Class 2: 9725 samples	Class 2: 2432 samples

Test Set (generated using the 13 images provided for testing):

Class 0: 5273 samples

Class 1: 5567 samples

Class 2: 4728 samples

Pre-processing steps:

The LISS4 level 2 data product obtained from Bhoonidhi portal contains band values in digital number (DN) format, which needs to be converted into top of atmosphere reflectance product (TOA) before any further analysis can be done.

Conversion from DN values to TOA reflectance values was performed using the following methodology:

a. **Conversion from DN to radiance:** Using the saturation ratio values for band obtained from the product metadata file, the spectral radiance values were computed by multiplying them with the DN values and dividing the product by 1024 (10bit data).
b. **Conversion from Radiance to TOA:** Spectral radiance values were converted to TOA reflectance values using the formula:

$$\rho_{\lambda i} = \frac{\pi L_i d^2}{E_0 Cos\theta},$$

where $\rho_{\lambda i}$ is TOA reflectance for ith band, L_i is the spectral radiance for ith band, d is the Earth-Sun distance in astronomical units (AU), E_0 is the ex-atmospheric solar irradiance, θ is the solar zenith angle.

L_i was calculated in the previous step, θ was obtained from the product metadata file, E_0 values were used as per the provided TOA_help folder depending on the satellite sensor, d value was obtained using the 'ephem' python package which provides earth sun distance depending upon the day of the year.

3. Model Architecture and methodology

For the given problem of accurately masking out cloud, shadow, and no-cloud classes, a Multi-Layer Perceptron (MLP) model was used with a pixel-based classification approach.

A Multilayer Perceptron is a type of an artificial neural network consisting of multiple fully connected layers with non-linear activation functions. Each layer transforms the input data space into increasingly abstract representations, allowing the network to model complex relationships between input features and output classes, helping identify non-linear class boundaries between the class samples which are otherwise not detected in simpler models.

A pixel-based classification approach was adopted for this task instead of using image patches. This decision was based on the practical constraint i.e., generating a large number of labelled image patches, along with their corresponding ground truth masks, would be time consuming. Additionally, training CNN-based models, that are most effective with spatial context which is available in image patches, typically requires large amount of patch-level data and longer training times. In contrast, pixel-level sample extraction enables faster data preparation, model training, and initial comparisons and tuning.

Choice of the classifier model:

The choice of MLP was motivated by the following factors:

a) **Ability to Model Non-Linear Class Boundaries**

Cloud, shadow, and no-cloud pixels often exhibit overlapping spectral characteristics. MLPs are well-suited to handle such complexity through layered transformations.

b) **Efficiency with Scaled Inputs**

MLPs work well when the input features are normalized or standardized. In this project, spectral features were scaled using a Standard Scaler, which further improved training stability and convergence.

c) **Simplicity and Flexibility**

Unlike convolutional neural networks (CNNs), MLPs are computationally lighter and easier to train on small to medium-sized pixel-level datasets — making them ideal for the problem statement without requiring large image patches or spatial context.

d) **Better Performance**

Comparative testing showed that the MLP model outperformed other machine learning based classifiers like Random Forest in terms of classification accuracy and F1-score on the test set. This indicates that MLP could better generalize the underlying patterns in the data.

Methodology:

i. A total of 20 labelled LISS4 images which were provided for training purposes were used to extract pixel-wise samples for the three target classes – No cloud, Cloud and Shadow.

ii. 80/20 split was applied to form training and validation sets for the training and tuning of a pixel-based MLP classifier model. A class-balanced weighted loss function was used to mitigate class imbalances.

iii. The trained MLP was evaluated on 13 test images provided using the test samples generated from these images which remained unseen to the trained model. Accuracy metrics and classification masks were generated for the test dataset provided.

iv. Additional evaluations were conducted both at the individual image level and using a combined test set for overall performance assessment.

Model architecture:

Multi-Layer Perceptron model for pixel-wise classification.

1. **Input features:**

3 spectral bands: SAMPLE_1 (Green band), SAMPLE_2 (Red band), SAMPLE_3 (NIR band)

2. **Architecture Details:**

Layer Type	**Input Neurons**	**Output Neurons**	**Activation Function**	**Dropout**
Fully connected layer	3	256	ReLU	0.2
Fully connected layer	256	128	ReLU	0.2
Fully connected layer	128	64	ReLU	0.2
Fully connected layer	64	32	ReLU	-
Output layer	32	3 (Number of classes)	Raw logits (predictions with argmax)	-

The model architecture used consists of four hidden layers with ReLU activations and dropout regularization, making it capable of learning non-linear patterns while avoiding overfitting.

The output layer outputs raw logits, which are then assigned class labels using argmax function for choosing the class with maximum value.

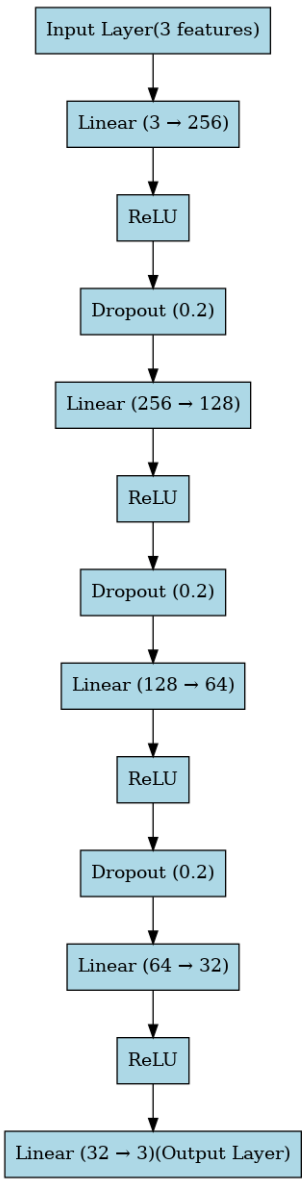
3. **Activation Function:**

All hidden layers use ReLU as the activation function.

4. **Output layer:**

Output layer outputs logit values for three output classes: No clouds, Clouds, and Shadow. The logits are then used to assign the output class for each pixel using argmax function.

5. **Model architecture diagram:**



4. Training Configuration

Configuration	Value
Framework	PyTorch
Loss Function	CrossEntropyLoss (with class weights for imbalance)
Optimizer	Adam
Learning Rate	0.001
Epochs	Up to 500 (Early Stopping with patience = 20)
Batch Size	220
Class Balancing	Yes
Feature Scaling	StandardScaler
Dataset Split	80% train / 20% validation (stratified)
Input Feature Size	3
Output Classes	Automatically determined from training labels
Dropout	0.2 in 3 hidden layers

5. Resources used

CPU and system configuration:

OS	Ubuntu 22.04.5 LTS
Architecture	x86_64
CPU op-mode(s)	32-bit, 64-bit
CPU(s)	128
Model Name	AMD EPYC 9534 64-Core Processor
CPU Family	25
Model	17
Thread(s) per Core	2
Core(s) per Socket	64
Socket(s)	1

Stepping	1
Frequency Boost	Enabled
CPU Max MHz	3718.07
CPU Min MHz	1500.00

RAM:

8 modules of 32 GB DDR5 Samsung each, totalling 258 GB of RAM

Size	Type	Speed	Manufacturer
32 GB	DDR5	4800 MT/s	Samsung

GPU:

Property	Value
GPU Model	NVIDIA RTX 5000 Ada Generation
CUDA Version	12.8
Total GPU Memory	32,760 MB (≈ 32 GB)

Python packages used:

The following open-source python packages were used for this project -

- datetime
- ephem
- math
- os
- zipfile
- rioxarray
- xarray
- dask
- torch (PyTorch)
- sklearn (scikit-learn)
- joblib
- numpy
- matplotlib
- geopandas
- pandas
- rasterio
- shapely
- tqdm
- seaborn
- graphviz

6. Evaluation Metrics

The following metrics were used to assess the accuracy of the classification results -

1. **Classification Accuracy**

   Classification accuracy is the proportion of total predictions (pixels, objects, or samples) that were correctly classified.

   Formula:

   Classification accuracy = No. of correctly classified samples / Total number of samples

2. **Precision**

   Precision refers to percentage of the total number of samples that were accurately classified as positive out of all the predicted positive samples. Out of all the predicted positive cases, how many were actually positive.

   Formula:

   Precision = TP / (TP + FP),

   where TP = True Positives, and FP = False Positives.

3. **Recall**

   Recall is the total number of samples that were accurately classified as positives among the actual positive samples.

   Formula:

   Recall = TP / (TP + FN),

   where TP = True Positives, and FN = False Negatives.

4. **F1-Score**

   F1-Score is defined as the harmonic mean of Precision and Recall, providing a balance between the two.

   Formula:

   F1-Score = 2 * Precision * Recall / (Precision + Recall)

5. **Confusion Matrix**

   A table used to evaluate the performance of a classification model by comparing the actual and the predicted classifications.

   Structure of the matrix (for binary classification):

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

7. Results

a. Quantitative

Accuracy assessment on the validation set:

Class	Precision	Recall	F1-Score	Samples
0 (No cloud)	0.86	0.76	0.81	2233
1 (Cloud)	0.89	0.93	0.91	2075
2 (Shadow)	0.85	0.91	0.88	2432
Accuracy	**0.87**			
Overall metrics	**0.87**	**0.87**	**0.86**	**6740**

Confusion Matrix for the validation set:

	Predicted: 0	Predicted: 1	Predicted: 2
Actual: 0	1707	186	340
Actual: 1	104	1920	51
Actual: 2	182	41	2209

Accuracy assessment on the test set:

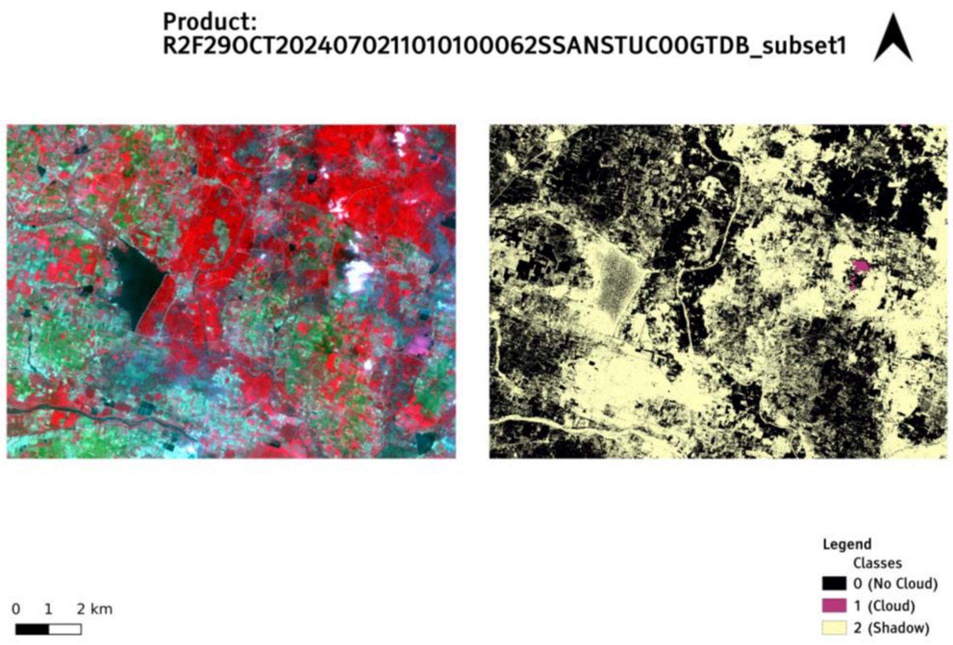
Class	Precision	Recall	F1-Score	Samples
0 (No Cloud)	0.6487	0.6624	0.6555	5273
1 (Cloud)	0.9118	0.7358	0.8144	5567
2 (Shadow)	0.6712	0.8080	0.7333	4728
Accuracy	**0.7328**			
Overall Metrics	**0.7496**	**0.7328**	**0.7359**	**15568**

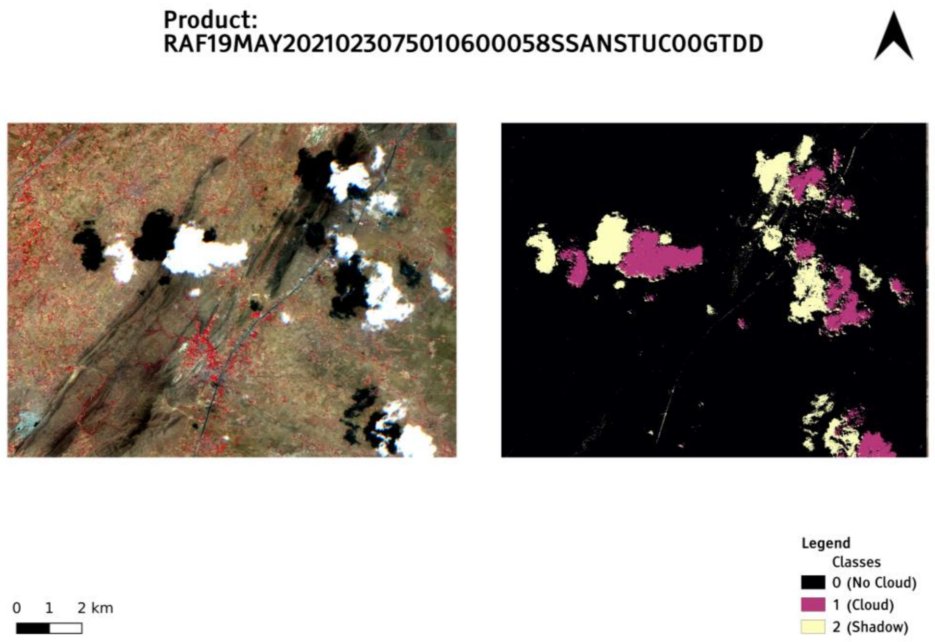
Confusion Matrix for the test set:

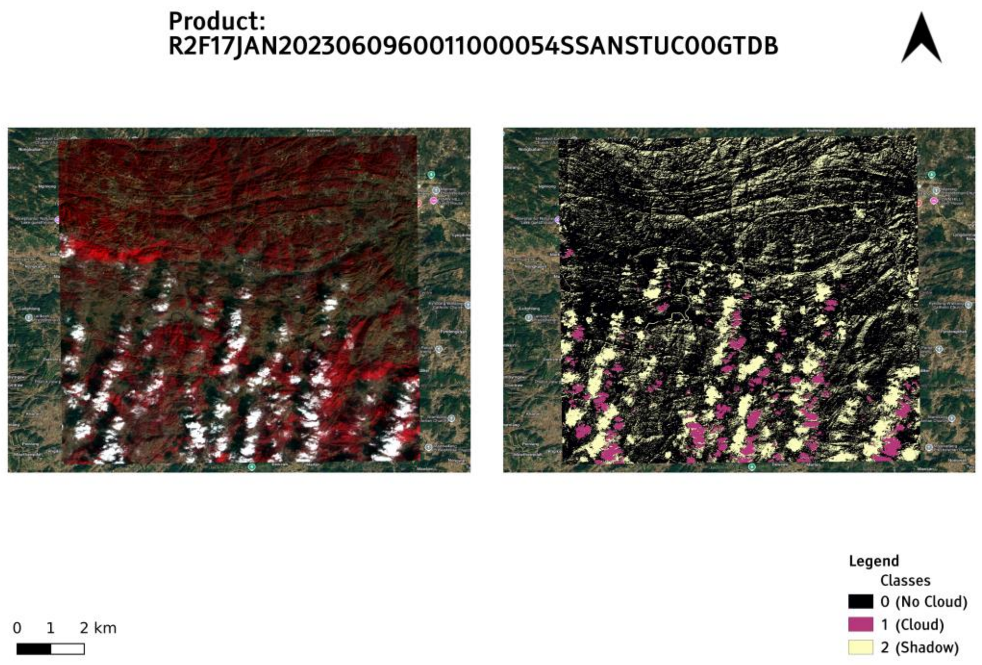
	Predicted: 0	Predicted: 1	Predicted: 2
Actual: 0	3493	396	1384
Actual: 1	984	4096	487
Actual: 2	908	0	3820

b. Visual Outputs

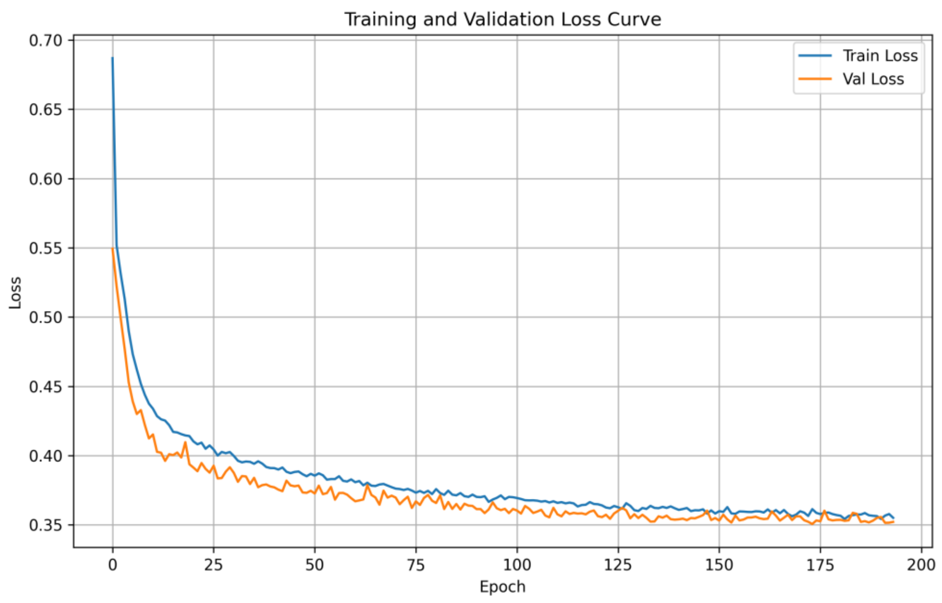
Portions of the output masks and their corresponding False Color Composite (FCC) images are shown below.

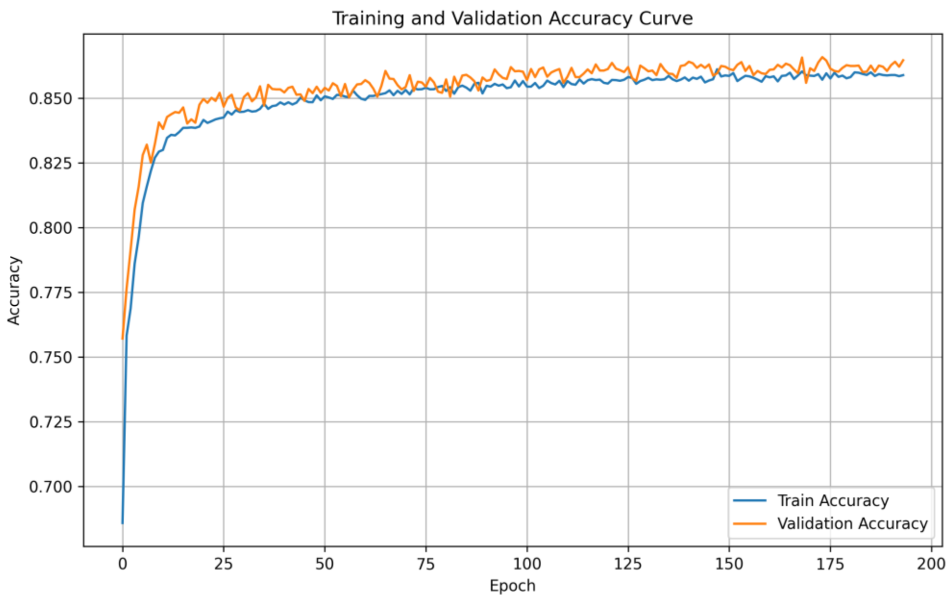


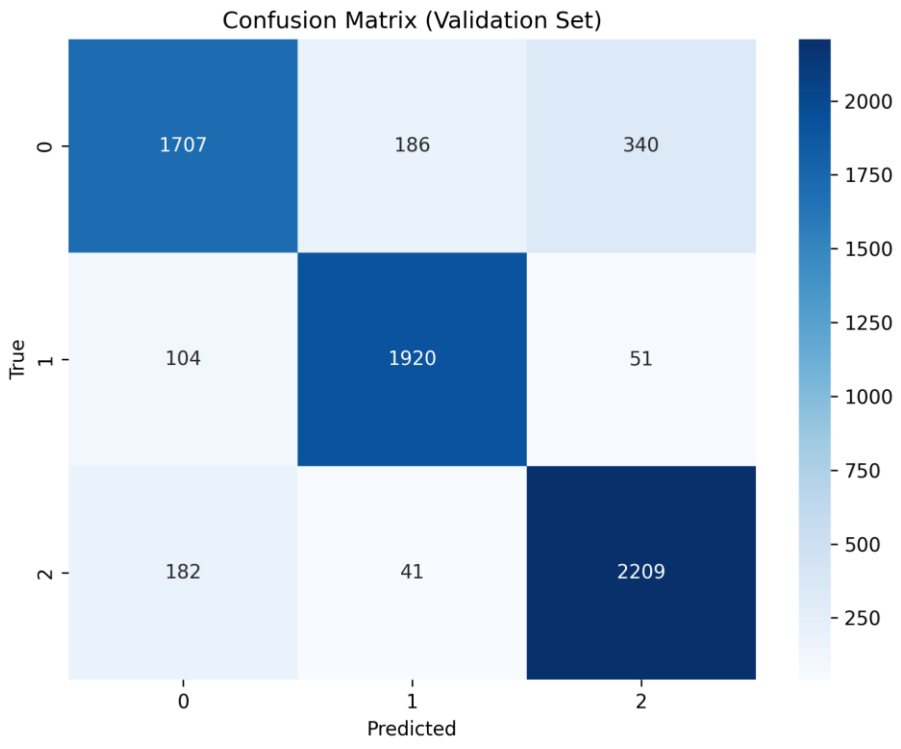




c. Training graphs







True \ Predicted	0	1	2
0	1707	186	340
1	104	1920	51
2	182	41	2209

8. Analysis

Model Performance:

Confusion matrix of the model on the test dataset shows that the model detects clouds well with high precision but has moderate recall, indicating some missed cloud regions.

No Cloud regions suffer from confusion, particularly with shadow areas. Shadow detection has higher recall than precision, meaning more areas are labelled as shadow, but with some false positives - areas that visually resemble shadows like roads, water bodies. This is evident in the images of the classification masks shared in the previous section:

RAF19MAY2021023075010600058SSANSTUC00GTDD_subset1,
R2F29OCT2024070211010100062SSANSTUC00GTDB_subset1.

Relatively low misclassification is seen between Cloud and Shadow classes, suggesting the model distinguishes them reasonably well.

The decrease in the accuracy metrics on the test dataset in comparison to the validation set is expected due to many unseen features, complex cloud structures and diverse terrain features present in the test dataset. High confusion between No Cloud and Shadow classes is observed in the confusion matrix for both, the test and validation sets. This is due to the spectral similarities between the two classes, the intensities of the shadows and their intra-class variations. There is also confusion seen in Cloud class with No Cloud class, especially due to diverse cloud patterns and thin layer layers of clouds that often get confused with No Cloud class features.

Training accuracy and loss curves:

Accuracy curve for the model training shows training and validation accuracies steadily increase and plateau, with minimal divergence, indicating no significant overfitting and good generalization. Loss curve also shows both training and validation losses decrease consistently, stabilizing near 0.35. No major gaps between them shows that the model training was stable.

Comparison with baseline models:

The trained MLP classifier model was also compared with a baseline Random Forest (RF) model trained with the same dataset. Comparisons for some of the results of the classifications on the test data are shown below.

Image	MLP Accuracy	RF Accuracy	Observation
RAF05FEB2025042370010400049SSANSTUC00GTDC	84.45%	79.95%	MLP outperforms RF clearly
RAF19MAY2021023075010600058SSANSTUC00GTDD_subset1	89.53%	92.13%	RF slightly better here
RAF18APR2025043393010400065SSANSTUC00GTDD_subset1	25.04%	23.57%	Both models struggle significantly
R2F30OCT2024070225010600057SSANSTUC00GTDB_subset1	88.88%	86.53%	MLP slightly better
R2F06SEP2021053879009200045SSANSTUC00GTDD_subset1	75.61%	65.82%	MLP outperforms RF substantially

The results show better overall performance of MLP in comparison to RF, showing its better suitability for the given task.

9. Conclusion and future improvements

The MLP classifier model shows best results for the Cloud classification. There is a significant confusion between the No Cloud and Shadow classes due to spectral similarities in three bands of the LISS4 dataset. Confusions between Cloud and No Cloud classes is also seen in the cases of thin and hazy cloud structures that allow land signatures to pass, creating confusions. The model training was stable. MLP classifier was compared with RF, showing better results. Drop in accuracy metrics for the test dataset as compared with the validation dataset suggests the need for further training for better generalisation of the classifier model.

There are several limitations of the classifier model. The model was trained on just the original bands (NIR, Green and Red) of the LISS4 dataset. Feature engineering with different band ratios and indices might help in improving the results further. The model does not take into account the spatial context of the images as of now. Use of texture features and image patches might help the model identify the spatial structures and patterns of the Cloud and No Cloud class features, helping in better classification. Current model also lacks the use of DEM for Shadow class distinguishability. Incorporating the same along with solar elevation and azimuth details might provide an added context for the model to learn and identify the shadow patterns from the different features.

Additionally, further training of the model with more diverse datasets in different seasons of the year will better help the model learn the class features better. With such a large dataset and spatial context added, CNNs might better identify and understand the underlying class patterns, helping identify the classes more effectively.