Cloud Segmentation using Satellite Imagery

SAI NITHISH[1]
Corresponding author: Sai Nithish (e-mail: sainithsih.official@ gmail.com).

"This work was supported in part by Little Place Labs"

ABSTRACT Cloud detection in satellite imagery is a crucial step for meteorology, climate monitoring, and remote sensing applications. In this study, we explore various cloud segmentation techniques on RGB and RGB+NIR satellite images using both classical image processing and deep learning approaches. Despite limited data availability, we investigate augmentation strategies, edge-device deployment considerations, and evaluate open-source pre-trained models such as Cloud-Net and Attention U-Net. Observations highlight the challenges of small datasets, variability in cloud patterns, and potential directions for future research.

INDEX TERMS Cloud Segmentation, Satellite Imagery, Remote Sensing, Deep Learning, Edge Deployment

I. INTRODUCTION
Clouds are visible aggregations of water droplets or ice crystals suspended in the atmosphere. They play a critical role in weather, climate, and the Earth's radiative balance.

A. TYPES OF CLOUDS
- **Cirrus:** high-altitude, wispy clouds.
- **Cumulus:** puffy, low- to mid-altitude clouds.
- **Stratus:** uniform, low-altitude clouds forming layers.
- **Nimbus:** precipitation-bearing clouds.

B. CLOUD IDENTIFICATION
Cloud identification relies on visual cues such as brightness, texture, and spectral reflectance. High albedo (bright white) and specific spectral responses in NIR or RGB channels often indicate cloud presence.

C. CHALLENGES IN CLOUD DETECTION
- Thin or semi-transparent clouds are difficult to detect. Shadows and bright surfaces (snow, roads) can lead to false positives.
- Variability in cloud shape, density, and illumination complicates segmentation.

II. SATELLITE IMAGERY
Satellite sensors capture electromagnetic radiation reflected or emitted from the Earth's surface and atmosphere, varying with surface type, moisture, and atmospheric conditions.

A. IMAGING TECHNIQUES
- Optical Imaging: Captures visible light (RGB).
- Multispectral Imaging: Captures specific bands beyond visible, e.g., NIR.

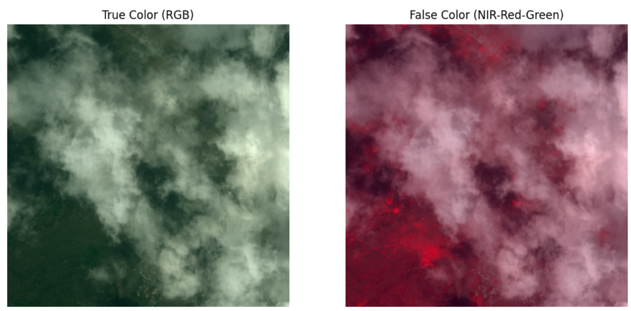

FIGURE 1. Illustration of spectral bands captured by satellites (RGB, NIR).

- Hyperspectral Imaging: Captures dozens to hundreds of narrow spectral bands.

B. SPECTRAL BANDS
Satellite sensors capture multiple spectral bands, including visible and non-visible wavelengths. Common bands include:
- Blue: 450-495 nm, sensitive to water bodies and haze.
- Green: G495-570 nm, captures vegetation and land features.
- Red: 620-750 nm, highlights soil and vegetation stress.
- NIR (Near Infrared): 750-900 nm, excellent for vegetation health, water separation, and cloud detection. Shortwave Infrared (SWIR), Thermal: used for water content, snow, and surface temperature.

Combination of standard red, green, and blue channels (visible spectrum) with the Near Infrared (NIR) band. The RGB channels capture color information, helping distinguish clouds from other features based on brightness and color contrast, while the NIR band enhances cloud detection by highlighting water droplets and ice crystals due to their strong reflectance in this wavelength, improving discrimination from vegetation, soil, and water.

III. CLOUD SEGMENTATION

Cloud segmentation aims to separate cloud pixels from non-cloud pixels. Techniques range from classical thresholding to deep learning models.

A. LITERATURE REVIEW

Research on cloud segmentation has progressed from traditional thresholding to modern deep learning and hybrid methods:

1) Traditional Methods

- Thresholding & Spectral Indices: Using indices like NDCI and NDSI to distinguish clouds from other surfaces; simple but limited with thin clouds.
- Object-Based Image Analysis (OBIA): Segments images into meaningful objects based on spectral and spatial properties before classification.

2) Machine Learning Approaches

- Support Vector Machines (SVMs): Classify pixels using handcrafted features; require careful feature selection.
- Random Forests: Handle complex datasets with multiple decision trees but limited in capturing spatial dependencies.

3) Deep Learning Methods

- Fully Convolutional Networks (FCNs): Pixel-wise classification for arbitrary-sized images
- U-Net: Encoder-decoder architecture for precise localization.
- Attention U-Net: Adds attention modules to focus on relevant cloud regions.
- Cloud-Net: Multi-scale architecture tailored for cloud segmentation.
- Transformer-Based Models: Capture long-range dependencies and global context.

4) Hybrid and Ensemble Approaches

- Combines traditional methods with deep learning for better performance.
- Ensemble learning merges predictions from multiple models to improve robustness.

Observations from Literature:

- Data scarcity limits deep learning model performance.
- Model generalization across diverse cloud types is challenging.
- Computational intensity and edge deployment are key considerations.
- Synthetic data, multi-temporal analysis, explainability, and real-time processing are active research areas.

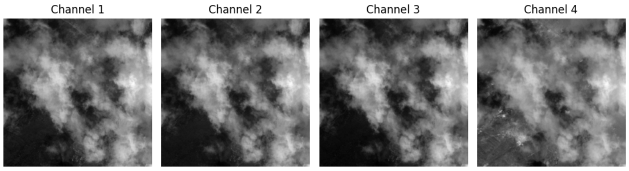

FIGURE 2. Sample RGB+NIR satellite images used in the study.

IV. DATASET

- Total Images: 10 images (5 unique, 4 with clouds).
- Preprocessing: Images converted to 4-channel arrays (RGB+NIR), resized to 512×512.
- Augmentation: Basic flips and rotations applied.

A. PREPARATION

Deep learning models need large datasets to perform good and since we only have handful number of images we can use a couple of transformations to the images to generate the dataset, augumentation techniques include:

- Geometric: flips, rotations, crops, scaling
- Photometric: brightness/contrast, color jitter, Gaussian noise
- Spatial: elastic deformation, affine transforms
- Synthetic generation using GANs

B. BOTTLENECKS AND MITIGATION

Cloud segmentation faces several bottlenecks due to high-resolution satellite imagery, limited computational resources, and deployment constraints. Detailed considerations include:

- Compute: Training and inference with high-resolution images are compute-intensive. Mitigation strategies include tiling images into smaller patches, using GPUs, or leveraging distributed computing.
- Memory: Large model weights and high-resolution batches can exceed memory limits. Techniques such as smaller batch sizes, mixed-precision training, or model checkpointing help reduce memory usage.
- Power/Energy: Edge devices or satellites may have limited power budgets. Optimizations like model quantization (INT8), pruning, knowledge distillation, or using lightweight architectures (MobileNet, EfficientNet) can reduce power consumption.
- Data Scarcity: Limited labeled data restricts model generalization. Mitigation includes aggressive data augmentation, synthetic data generation with GANs, and transfer learning.
- Latency: Real-time or near-real-time applications require low inference time. Efficient architectures and model compression techniques are essential.
- Storage and I/O: High-resolution images require large storage and fast I/O. Solutions include efficient file formats (e.g., HDF5, compressed TIFF) and streaming pipelines.

C. EDGE DEVICES

- Must optimize model size, inference speed, and memory footprint.
- Lightweight segmentation models (MobileNet backbones, quantized U-Nets) are preferred.

D. OTHER AVAILABLE DATASETS

- Landsat datasets
- Sentinel-2 datasets

V. EXPERIMENTS: CLOUD SEGMENTATION TECHNIQUES

This section documents my experimental journey in cloud segmentation, highlighting my iterative approach where initial methods failed or performed inconsistently, leading me to develop more robust solutions.

A. MANUAL THRESHOLDING

I first attempted manual thresholding, setting intensity thresholds on normalized RGB channels as seen on Figure 3

- Methodology: I classified pixels as clouds if their brightness exceeded a defined threshold in normalized RGB channels. I tested multiple thresholds to see the effect on cloud detection.
- Advantages: Extremely fast and simple; provided a baseline for comparison.
- Limitations: Failed to detect thin or semi-transparent clouds; misclassified bright surfaces like snow, sand, or urban structures.
- Observations: Only worked for dense clouds; clearly indicated the need for more advanced methods.

B. BRIGHTNESS MASKING

Next, I implemented brightness masking to better isolate bright cloud regions as seen on Figure 4.

- Methodology: I applied a global brightness threshold across RGB or RGB+NIR channels. I experimented with different brightness metrics and thresholds to improve segmentation.
- Advantages: Slightly better detection for bright clouds than manual thresholding.
- Limitations: Still missed thin clouds; false positives persisted in bright non-cloud areas.
- Observations: Marginal improvement over manual thresholding; suggested that incorporating spectral features might enhance results.

C. FLOOD FILLING

To address fragmented cloud masks, I experimented with flood filling as seen on Figure 5.

- Methodology: I selected high-brightness seed pixels and expanded them iteratively to neighboring pixels that met brightness criteria, effectively growing cloud regions.
- Advantages: Connected fragmented cloud regions and improved continuity.
- Limitations: Sensitive to seed selection and brightness thresholds; sometimes included bright non-cloud areas.
- Observations: Showed promise in grouping cloud regions but could not reliably detect thin clouds.

D. PREPROCESSING + FLOOD FILLING

I combined preprocessing with flood filling to improve performance as seen on Figure 6.

- Methodology: I normalized RGB+NIR channels, applied brightness masking, then performed flood filling on the candidate cloud regions.
- Advantages: Improved mask continuity and reduced fragmentation compared to previous methods.
- Limitations: Still struggled with thin or semi-transparent clouds.
- Observations: Effective on the small dataset; indicated that spectral features might further enhance detection.

E. RGB+NIR BASED CLOUD MASKING

After observing limitations in earlier approaches, I developed a custom RGB+NIR cloud masking technique that produced consistent results. This method also utilizes the Normalized Difference Vegetation Index (NDVI) to help exclude vegetation from cloud detection; NDVI highlights healthy vegetation based on the difference between NIR and red reflectance, ensuring that bright vegetation areas are not misclassified as clouds as seen on Figure 7.

- Criteria: Brightness close to white, low color difference (neutral), high NIR reflectance, low NDVI to exclude vegetation.
- Methodology: I evaluated each pixel against these criteria. Preprocessing included normalization and NDVI computation. Pixels satisfying all conditions were classified as clouds.
- Advantages: Successfully detected thin and semi-transparent clouds; reduced false positives; robust across my dataset and tested well on Sentinel-2 data.
- Limitations: Sensitive to threshold selection; may misclassify bright man-made surfaces; dependent on NIR data quality.
- Observations: Outperformed classical methods; proved to be my primary successful method after iterative experiments.

Input: RGB+NIR image
Normalize channels
Compute NDVI
For each pixel:
  If (Brightness $\approx$ white) AND (low color difference) AND (high NIR) AND (low NDVI):
    classify as cloud
Output: Cloud mask

F. CLOUD-NET

Encouraged by the success of RGB+NIR masking, I explored deep learning with Cloud-Net [cite Cloud-Net paper], which

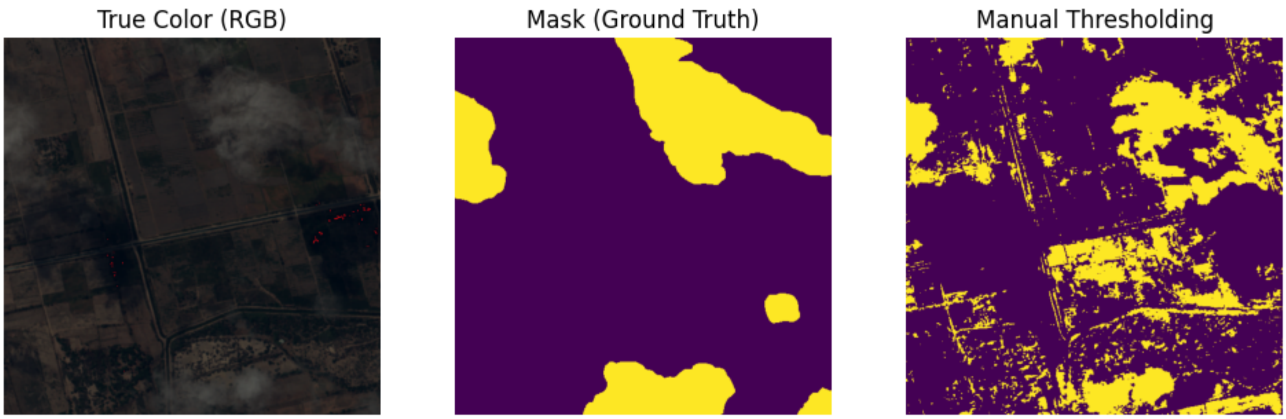

FIGURE 3. Result of Manual Thresholding

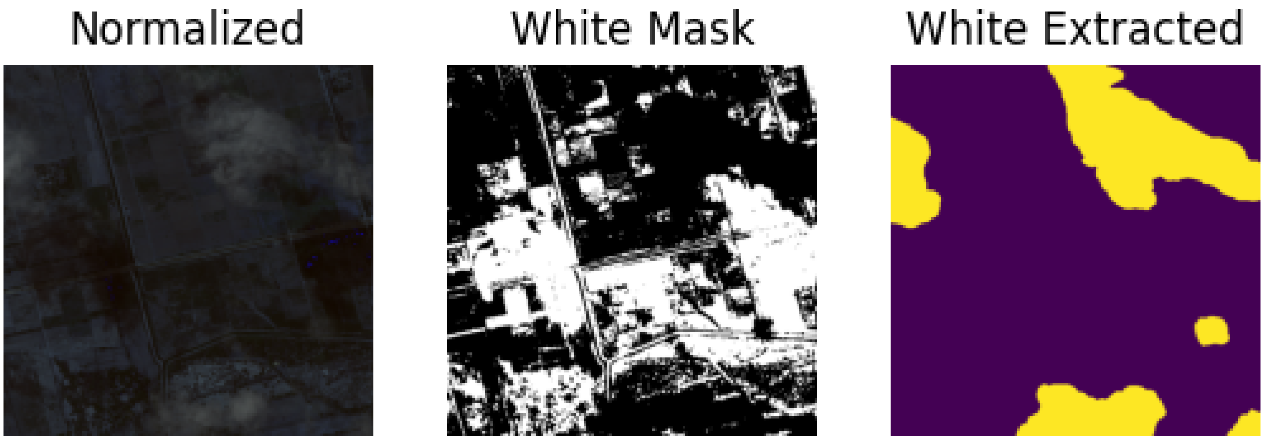

FIGURE 4. Result of Brightness Masking

FIGURE 5. Result of Flood Filling

is a CNN architecture designed specifically for cloud segmentation as seen on Figure 8.

- Methodology: Multi-scale CNN on 512×512 RGB+NIR patches; I used pretrained weights to evaluate performance without full training.
- Advantages: Captured complex cloud structures; reduced manual effort.
- Limitations: Struggled to distinguish sea waves from

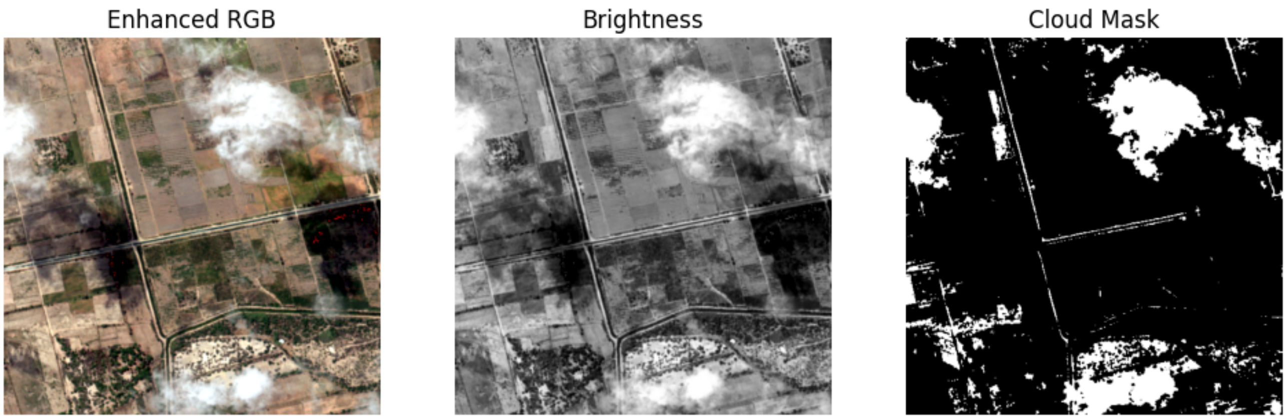

FIGURE 6. Result of Preprocessing + Flood Filling

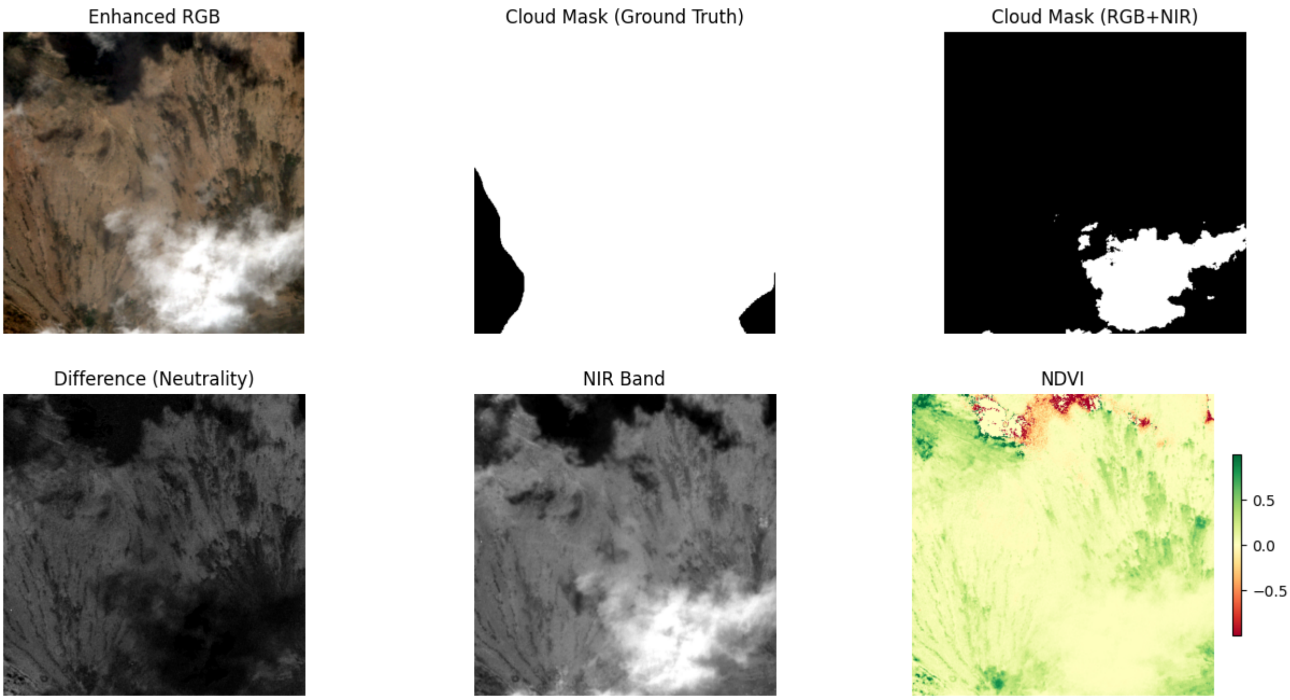

FIGURE 7. Result of RGB+NIR Based Cloud Masking

clouds; failed on indistinct clouds; high computational requirements.
- Observations: Worked for some images but not consistently; highlighted the limitations of transfer learning with small, domain-specific datasets.

G. ATTENTION U-NET

I planned to implement Attention U-Net for improved feature focus.

- Status: Implementation was incomplete due to time constraints; testing was very limited.

VI. RESULTS

My experiments illustrate a progression from simple to complex methods:

- Classical approaches often failed on thin or indistinct clouds.
- RGB+NIR based cloud masking achieved consistent segmentation and performed well on external Sentinel-2 data.
- Cloud-Net was effective for clear clouds but misclassified ambiguous regions like sea waves.
- Attention U-Net testing remains incomplete.

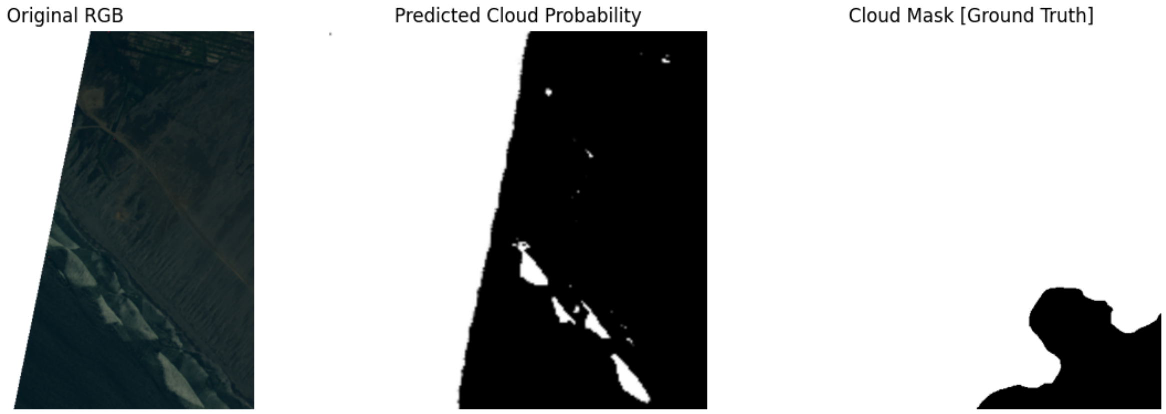

FIGURE 8. Result of RGB+NIR Based Cloud Masking

VII. CONCLUSION

This study highlights an iterative experimentation process. Initial simple methods failed or produced inconsistent results, which led to the development of RGB+NIR based cloud masking that succeeded. Deep learning models like Cloud-Net showed promise but had mixed results due to dataset and feature constraints.

VIII. FUTURE SCOPE

- Expand datasets and apply advanced augmentations or synthetic data.
- Complete and optimize Attention U-Net for improved detection.
- Explore edge deployment with lightweight or quantized models.
- Integrate additional spectral bands for enhanced cloud differentiation.

ACKNOWLEDGMENT

LLMs were used for basic code generation and research purposes. All code used in this study is available at: https://github.com/RSaiNithish/LPL-Cloud-Segmentation.