Deep Learning-Based Cloud Pattern Segmentation from Satellite Imagery

Utku Efe Akdoğan*, Fatih Alagöz
Boğaziçi University
Email: utkuefeakdogan@gmail.com

***Abstract*—This paper presents an efficient deep learning approach for segmenting four distinct cloud patterns which are Sugar, Flower, Fish, and Gravel from satellite imagery. Using a U-Net architecture with EfficientNet-B1 encoder, we achieved competitive segmentation results while maintaining computational efficiency. Through extensive experimentation with multiple state-of-the-art architectures including MAnet, DeepLab V3+, UNet++, and Feature Pyramid Network (FPN), we demonstrate that architectural complexity does not necessarily correlate with improved performance in this domain. Our findings indicate that the inherent ambiguity in cloud pattern classification, where even expert meteorologists can't always pinpoint the exact boundaries with complete certainty, presents a fundamental challenge that sophisticated architectures cannot readily overcome. Pattern-specific analysis revealed that organized, well-defined patterns like Flower clouds were more accurately segmented, while fine-grained, less organized formations like Sugar patterns proved more challenging, reflecting the inherent complexity in classifying different cloud organizations. This study demonstrates that carefully chosen architectural decisions and optimization strategies can yield robust results in cloud pattern segmentation tasks, while suggesting that future improvements may depend more on addressing fundamental data ambiguities than on increasing model complexity.**

I. Introduction

While traditional physics-based numerical models have been the bedrock of atmospheric sciences, deep learning models have emerged as powerful tools capable of analyzing complex weather and climate data by learning intricate dependencies [9]. Our work on cloud pattern segmentation aligns with this broader trend, focusing on the specific challenge of identifying and classifying distinct cloud formations that play crucial roles in Earth's climate system.

Shallow clouds play a crucial role in Earth's climate system. As noted by Rasp et al. [1], humans excel at detecting interesting patterns in satellite imagery, but gathering sufficient data of subjective features for significant analysis is often challenging. This capability has led to the discovery of new phenomena, particularly in the organization of shallow cumulus convection in trade wind regions, where clouds significantly influence Earth's radiation balance yet are poorly represented in climate models [1].

The Max Planck Institute for Meteorology initiated a project to address this challenge through a combination of crowd-sourcing and deep learning. Initially, a team of 67 scientists screened 10,000 satellite images and classified almost 50,000 mesoscale cloud clusters into four distinct patterns: Sugar, Flower, Fish, and Gravel [1]. This dataset formed the foundation for subsequent machine learning approaches to automate pattern detection and create global climatologies of these four patterns.

The satellite imagery used in this study comes from TERRA and AQUA MODIS visible images, covering three regions chosen for their climatological similarity [1]. Each image spans approximately 14° latitude by 21° longitude, with data collected over an eleven-year period from 2007 to 2017. The dataset was created through a rigorous crowd-sourcing initiative, where multiple experts labeled each image, allowing for the natural handling of subjective classifications through consensus.

This paper presents our approach to the cloud pattern classification challenge, building upon the foundational work of Rasp et al. [1]. By developing efficient deep learning models for this task, we aim to contribute to better understanding of cloud organization patterns, which is crucial for improving climate model projections and reducing uncertainties in climate predictions.

II. Dataset and Cloud Patterns

The dataset comprises satellite images from NASA Worldview, captured by TERRA and AQUA satellites. The regions were selected based on a similarity analysis of atmospheric conditions that resemble those encountered during the DJF season east of Barbados, where these patterns were first identified [1]. Each image spans approximately 14° latitude by 21° longitude, providing a consistent spatial scale for pattern recognition.

A. Cloud Pattern Characteristics

The four distinct cloud patterns present unique challenges for computer vision and pattern recognition tasks [2]:

Sugar clouds represent the most challenging pattern for automated detection, characterized by very fine-grained textures of small cumulus clouds. These patterns typically appear in calm conditions [2], making them particularly difficult to segment due to their lack of distinct boundaries and tendency to form subtle, dendritic structures under certain conditions [1].

Flower patterns present more distinct features, forming large-scale circular structures ranging from 50 to 200 km in diameter with clear cloud-free regions between them. These patterns emerge in windy and stable conditions [2], offering more definitive boundaries for segmentation algorithms compared to other patterns.

Fish patterns create unique skeletal structures that can span up to 1,000 km longitudinally. Their elongated nature and distinctive "fishbone" appearance [1] make them particularly suitable for detection using convolutional neural networks, though their variable size presents challenges for maintaining consistent recognition accuracy.

Gravel patterns exhibit medium-scale granular features marked by arcs or rings approximately 20 km in diameter. These patterns occur in windy conditions [2] and present intermediate-level detection difficulty, with more defined structures than Sugar but less distinctive shapes than Flower or Fish patterns.

B. Data Creation and Labeling Process

The dataset creation process exemplifies modern approaches to handling subjective pattern recognition tasks. Using the Zooniverse platform, 67 scientists participated in cloud labeling days at two research institutes [1]. This resulted in approximately 50,000 classified mesoscale cloud clusters from 10,000 satellite images, with each image receiving annotations from about three different experts.

The labeling interface design prioritized practical considerations crucial for machine learning applications. Rather than requiring precise polygon boundaries, which would be time-consuming and potentially introduce unnecessary noise, rectangular selection tools were employed. This decision acknowledged both the inherently fuzzy boundaries of cloud structures and the need for efficient data collection to build a substantial training dataset [1].

C. Dataset Characteristics and Challenges

Several aspects of this dataset make it particularly valuable for machine learning research. The ground truth construction through multi-expert consensus, where the final masks are derived from the union of areas marked by multiple experts, provides a robust approach to handling subjective classifications. The patterns' inherent scale variance, ranging from fine-grained Sugar to large-scale Fish patterns, requires models to handle multi-scale feature detection effectively. Pattern ambiguity presents another significant challenge, as the dataset includes cases where patterns overlap or gradually transition into one another, reflecting real-world complexity in pattern recognition tasks. Despite these challenges, the patterns show remarkable global coherence across different geographical regions, suggesting that the learned features represent meaningful physical characteristics despite the subjective nature of the classification [1]. These characteristics make the dataset particularly suitable for developing and testing deep learning approaches to pattern recognition in satellite imagery, while also presenting significant challenges that push the boundaries of current computer vision techniques.

III. Methodology

A. Model Architecture

We implemented a U-Net architecture with EfficientNet-B1 encoder, chosen for its optimal balance between performance and computational efficiency. This architectural choice reflects a key insight from recent surveys in weather-related deep learning: while more complex models exist, simpler architectures can often achieve competitive results when properly optimized [8], [9]. Our approach prioritizes fast inference and practical computational requirements, important considerations highlighted in recent weather prediction literature [9].

The U-Net architecture, originally introduced by Ronneberger et al. [3], consists of a contracting path (encoder) and an expansive path (decoder) that gives it its characteristic U-shaped structure. This architecture was specifically designed to work with very few training images while maintaining high segmentation accuracy. The contracting path follows a typical convolutional network pattern, with repeated application of convolutions followed by ReLU activation functions and max pooling operations. Each downsampling step doubles the number of feature channels, enabling the network to capture increasingly complex features.

The expansive path, which is more or less symmetric to the contracting path, combines precise localization with contextual information. Each step in the expansive path consists of an upsampling operation followed by feature channel halving, concatenation with the corresponding feature map from the contracting path, and convolutions with ReLU activations. This symmetric expansion allows the network to propagate context information to higher resolution layers, enabling precise segmentation boundaries.

A key innovation in U-Net's design is its skip connections between corresponding layers in the contracting and expansive paths. These connections allow the network to combine high-resolution features from the contracting path with upsampled features, enabling precise localization. This is particularly important in our cloud pattern segmentation task, where accurate boundary detection between different cloud types is crucial.

Our implementation modifies the original U-Net architecture by replacing the standard convolutional encoder with EfficientNet-B1. This modification maintains the architectural benefits of U-Net while leveraging the improved feature extraction capabilities of EfficientNet. We evaluated several architectures including U-Net++, FPN (Feature Pyramid Network), and different EfficientNet variants (B1-B3), with U-Net + EfficientNet-B1 providing the best balance of speed and accuracy.

B. Training Infrastructure

The model was trained on multiple GPU configurations to optimize performance. Our primary setup utilized dual Tesla T4 GPUs (T4 x2), providing 2,560 CUDA cores per GPU and 32GB total GDDR6 memory (16GB per GPU). Through extensive experimentation with batch sizes, we found that the T4 x2 configuration could handle a maximum batch size of

16, which we used for optimal training efficiency. We also experimented with Tesla P100 GPUs, which despite offering 3,584 CUDA cores per GPU and higher memory bandwidth, were limited to a batch size of 8 due to our specific model configuration and memory requirements. The system was optimized for mixed-precision calculations using PyTorch's DataParallel functionality to efficiently distribute the computational load across available GPUs. Training times varied significantly across different architectures: the baseline U-Net completed training in 6-8 hours and FPN followed, while more complex architectures like U-Net++, DeepLab V3+ and MAnet required 10-12 hours for complete convergence.

C. Training Strategy

Our training process involved extensive experimentation with various architectures and configurations. While ensemble methods have shown promise in cloud segmentation tasks [10], particularly in enhancing predictive capability on unseen samples, we found that the additional computational overhead and complexity outweighed the benefits for our specific application. The baseline U-Net architecture, as described by Ronneberger et al. [3], demonstrated superior performance with the best Dice scores and fastest training times. Its characteristic U-shaped structure with skip connections enabled precise localization while maintaining contextual information, making it particularly effective for segmentation tasks. We also evaluated the Feature Pyramid Network (FPN), introduced by Lin et al. [4], which achieved results close to U-Net but required more training time. While FPN showed promise in handling objects at different scales through its inherent multi-scale, pyramidal hierarchy, the additional computational overhead did not justify its use in our specific application.

Further experiments included MAnet, which incorporated sophisticated attention mechanisms but performed slightly worse than U-Net and had longer training times. Following Li et al. [5], we implemented MAnet with kernel attention and channel attention mechanisms, using decoder channels (256, 128, 64, 32, 16) and integrating with ResNeXt-101 features. While the attention mechanisms were designed to capture long-range dependencies, in practice they added computational complexity without proportional performance gains. We also tested UNet++, which showed promise in medical image segmentation through its nested, dense skip pathways. Following Zhou et al. [7], we implemented the architecture with redesigned skip connections and deep supervision. Although UNet++ demonstrated impressive results on medical imaging tasks with IoU gains of 3.9 points over U-Net in their original implementation, in our cloud pattern segmentation task, the dense skip pathways and deep supervision did not yield proportional improvements, likely due to the inherent ambiguity in cloud pattern boundaries and our computational constraints requiring image resizing.

DeepLabV3+ showed initial promise but proved challenging to stabilize. Following Chen et al. [6], we configured the model with encoder output stride of 16, decoder channels of 256, and atrous rates of (6, 12, 18). Despite its sophisticated atrous spatial pyramid pooling (ASPP) design for multi-scale feature extraction, the model exhibited rapid overfitting and proved difficult to train effectively, ultimately yielding the lowest performance among tested architectures.

For encoder selection, we conducted comprehensive experiments with various configurations. Beyond our initial tests with EfficientNet variants (B1-B3), we explored more recent architectures including EfficientNet-B4 with noisy-student weights. Despite the theoretical advantages of these more sophisticated encoders, the performance improvements were marginal and did not justify the increased computational overhead. We also experimented with ensemble methods, combining predictions from multiple models, but the modest performance gains did not warrant the substantial increase in computational requirements, particularly given our GPU time constraints on the Kaggle platform.

Our final configuration used EfficientNet-B1 with 7.8M parameters, which provided an optimal balance of performance and efficiency. Other tested variants included EfficientNet-rasp (9.2M parameters), EfficientNet-B3 (12M parameters), and ResNet-50 (25.6M parameters), but these showed increased complexity without proportional gains or proved too heavy with risk of overfitting.

For the loss function, we employed a combination of Binary Cross Entropy (BCE) and Dice Loss, which proved more effective than Focal + Dice Loss for our use case. This combination was chosen because BCE Loss provided stable pixel-wise error calculation, while Dice Loss helped handle slight imbalances in segmented areas. Additionally, the dataset showed minimal class imbalance but some segment mask size variations, making this combination particularly suitable. We also experimented with a novel learnable weighted loss function that dynamically adjusted the relative importance of BCE and Dice Loss during training. This approach used learnable parameters with softplus activation for weight normalization and included log-scale adjustments to handle different loss scales. While theoretically promising, this adaptive loss function did not significantly improve performance compared to the simpler fixed-weight combination, possibly due to the added complexity in the optimization landscape. The learnable weights tended to converge to values similar to our manually tuned weights, validating our initial fixed-weight approach.

D. Data Augmentation

We employed various augmentation techniques to address the limited availability of human-annotated data and improve model generalization. For geometric transformations, we implemented vertical and horizontal flips to increase dataset diversity, along with random geometric distortions to simulate varying viewing angles. In terms of image processing, we applied noise injection to improve robustness and resolution modifications (downsampling) to reduce training time while maintaining the required 350 x 525 px output dimensions.

These augmentation strategies proved crucial for expanding the effective dataset size, enhancing model generalization capabilities, maintaining computational efficiency during training, and meeting competition requirements for final prediction mask dimensions.

E. Optimization Strategy

Rather than implementing a separate classification model, we utilized random sampling to fine-tune label and pixel thresholds. Through extensive experimentation, we identified optimal ranges for both threshold types: label thresholds between 0.80-0.95 (with 0.01 step size) and pixel thresholds between 0.20-0.45 (with 0.05 step size). To make the optimization computationally tractable, we employed random sampling with 100 combinations. This approach provided a good balance between exploration of the parameter space and computational efficiency.

Algorithm 1 Random Sampling for Threshold Optimization

```
procedure OptimizeThresholds($M$, $D$, $N = 100$)
    $L_{vals} \leftarrow \{0.80, 0.81, \ldots, 0.94, 0.95\}$    ▷ 16 label threshold values
    $P_{vals} \leftarrow \{0.20, 0.25, \ldots, 0.40, 0.45\}$    ▷ 6 pixel threshold values
    $S_{best} \leftarrow 0$    ▷ Best overall Dice score
    $L^* \leftarrow \emptyset$, $P^* \leftarrow \emptyset$
    for $i = 1$ to $N$ do    ▷ Random sampling iterations
        $L_{curr} \leftarrow$ RandomChoice($L_{vals}$, 4)    ▷ 1 threshold for each class (Sugar, Flower, Fish, Gravel)
        $P_{curr} \leftarrow$ RandomChoice($P_{vals}$, 4)    ▷ 1 threshold for each class (Sugar, Flower, Fish, Gravel)
        $S_{dice}, S_{iou} \leftarrow$ Evaluate($M$, $D$, $L_{curr}$, $P_{curr}$)
        if $S_{dice} > S_{best}$ then
            $S_{best} \leftarrow S_{dice}$
            $L^* \leftarrow L_{curr}$
            $P^* \leftarrow P_{curr}$
        end if
    end for
    return $L^*, P^*$
end procedure
```

The algorithm performs random sampling over discrete threshold values. For label thresholds, we consider 16 possible values (0.80 to 0.95 with 0.01 steps), and for pixel thresholds, we consider 6 possible values (0.20 to 0.45 with 0.05 steps). With 100 samples and approximately 1 minute per evaluation, the optimization process takes about 1.67 hours. While this approach doesn't guarantee finding the global optimum, our experiments showed that it consistently finds high-quality solutions that generalize well to the test set. The optimized thresholds are then fixed for inference, adding no additional computational overhead during deployment.

IV. Results and Evaluation

A. Performance Metrics

The model achieved an overall Mean Dice Score of 0.6543 and an overall IoU (Intersection over Union) of 0.6067 across all cloud pattern types. In terms of pattern-specific performance, Flower clouds demonstrated the best performance with Dice and IoU scores of 0.7663 and 0.7310 respectively, likely due to their more distinct and well-defined boundaries. Fish and Gravel clouds showed intermediate performance levels, while Sugar clouds proved more challenging with lower scores of 0.5948 (Dice) and 0.5235 (IoU), primarily due to their fine-grained nature and less organized patterns. These metrics indicate that while the model performs well on clearly structured patterns, it faces more difficulty with subtle, less organized cloud formations.

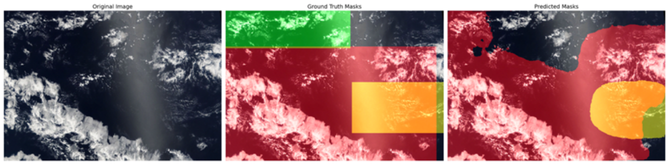

Fig. 1. Comparison of ground truth masks with model predictions

B. Performance Analysis

The model's performance varied significantly across different cloud types, which can be attributed to their inherent characteristics. Flower clouds achieved the highest scores with Dice 0.7663 and IoU 0.7310, demonstrating superior performance due to their distinct and well-separated patterns, while their clearer boundaries made segmentation more reliable. Fish and Gravel clouds showed intermediate performance levels, with Fish clouds achieving a Dice score of 0.6437 and Gravel clouds reaching 0.6125. Sugar clouds proved to be the most challenging, recording the lowest scores with Dice 0.5948 and IoU 0.5235, with this difficulty primarily attributed to their small size and less organized patterns, which required more precise detection capabilities.

C. Model Generalization

The model demonstrated strong generalization capabilities, achieving scores of 0.66479 and 0.66009 on the private and public leaderboards respectively. The close alignment between public and private scores indicates robust generalization, and notably, our approach approached the competition winner's score (0.67175) while maintaining a simpler architecture.

D. Comparison with Winner's Approach

While our approach prioritized simplicity and efficiency, the competition winner employed a more complex strategy that utilized an ensemble of three classification heads and incorporated nine segmentation models. Their approach used segmentation models for both mask predictions and classification, averaged top K pixel probabilities for classification, and required significantly more computational resources and training time.

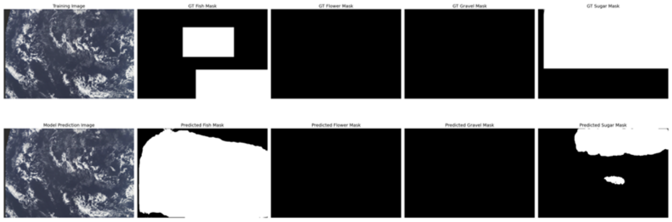

Fig. 2. Visualization of segmentation results across different cloud patterns

E. Performance by Cloud Type

Among the different cloud patterns, Flower clouds achieved the best performance with Dice 0.7663 and IoU 0.7310, followed by Fish clouds with Dice 0.6437 and Gravel clouds with Dice 0.6125, while Sugar clouds proved most challenging with Dice 0.5948 and IoU 0.5235. The model achieved competitive scores on both public (0.66009) and private (0.66479) leaderboards, approaching the competition winner's score of 0.67175 while maintaining a simpler architecture.

V. Conclusion

This study demonstrates that effective cloud pattern segmentation can be achieved through careful architectural choices and optimization strategies, even when faced with inherently ambiguous data and computational constraints. Our findings suggest that model sophistication alone does not guarantee better performance in this domain.

The inherent subjectivity in cloud pattern classification presents a fundamental challenge that even expert meteorologists struggle with. This human-level ambiguity in pattern boundaries and classifications suggests that pursuing marginally higher accuracy metrics through increasingly complex models may not be the most productive direction for improvement.

Our experiments with various architectures revealed that newer, more sophisticated models like MAnet, DeepLab V3+, and U-Net++ did not necessarily translate to better performance in this specific application. While these architectures have shown impressive results in other domains, their advantages were diminished in our context due to several factors. The need to resize images to meet computational constraints reduced the very fine-grained details these advanced architectures are designed to capture. The inherent ambiguity in cloud pattern boundaries limits the potential benefits of more sophisticated boundary detection mechanisms. Furthermore, we observed a significant trade-off between model complexity and training stability, particularly evident in our experiments with DeepLab V3+.

Instead, our results suggest that a well-optimized U-Net architecture with EfficientNet-B1 encoder provides an effective balance between computational efficiency and segmentation accuracy. This simpler approach achieved competitive results (private leaderboard score of 0.66479 and public leaderboard score of 0.66009) while maintaining practical computational requirements and training stability.

These findings have important implications for future work in cloud pattern segmentation. Rather than focusing solely on architectural innovations, more attention might be productively directed toward developing more consistent labeling protocols for cloud patterns, improving data quality and resolution while maintaining computational feasibility, and creating more robust evaluation metrics that account for the inherent ambiguity in pattern classification. Our work demonstrates that in domains with high inherent ambiguity, the path to improvement may lie not in increasingly complex models, but in better understanding and accounting for the fundamental uncertainties in the task itself. This insight could be valuable for other computer vision tasks where ground truth labels involve subjective human judgment.

While recent research has demonstrated the potential benefits of model ensembling in cloud segmentation tasks [10], our work shows that carefully optimized single-model architectures can achieve competitive results while maintaining practical computational requirements. This finding is particularly relevant for operational deployment scenarios where infrastructure simplicity and inference speed are crucial considerations.

Author Contributions

U.E.A. conducted the research, performed the experiments, analyzed the results, and wrote the manuscript. F.A. supervised the research, provided guidance on the methodology, and reviewed the manuscript. All authors have read and agreed to the published version of the manuscript.

Data Availability

The dataset used in this study was provided through the "Understanding Clouds from Satellite Images" Kaggle competition. In accordance with the competition rules, the data consists of satellite images from NASA Worldview, captured by TERRA and AQUA satellites, with expert-labeled cloud pattern annotations. Due to competition data usage restrictions, we cannot directly redistribute the dataset. Interested researchers can access the data by participating in the competition at kaggle.com/competitions/understanding_cloud_organization.

Supplementary Information

No supplementary materials were used in this study.

Competing Interests

The authors declare no competing financial or non-financial interests.

References

[1] S. Rasp, H. Schulz, S. Bony, and B. Stevens, "Combining Crowd-Sourcing and Deep Learning to Explore the Mesoscale Organization of Shallow Convection," Bulletin of the American Meteorological Society, vol. 100, no. 11, pp. 2237-2252, 2019, doi: https://doi.org/10.48550/arXiv.1906.01906.

[2] S. Bony, H. Schulz, J. Vial, and B. Stevens, "Sugar, Gravel, Fish, and Flowers: Dependence of Mesoscale Patterns of Trade-Wind Clouds on Environmental Conditions," Geophysical Research Letters, vol. 48, no. 1, p. e2019GL085988, 2020, doi: 10.1029/2019GL085988.

[3] 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), vol. 9351, pp. 234-241, 2015, doi: https://doi.org/10.48550/arXiv.1505.04597.

[4] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, ”Feature Pyramid Networks for Object Detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2117-2125, 2017, doi: https://doi.org/10.48550/arXiv.1612.03144.

[5] R. Li, S. Zheng, C. Duan, C. Zhang, J. Su, and P.M. Atkinson, ”Multi-Attention-Network for Semantic Segmentation of Fine Resolution Remote Sensing Images,” IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 1, pp. 426-439, 2021, doi: https://doi.org/10.48550/arXiv.2009.02130.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A.L. Yuille, ”DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,” IEEE TPAMI, 2017, doi: https://doi.org/10.48550/arXiv.1606.00915.

[7] Z. Zhou, M.M.R. Siddiquee, N. Tajbakhsh, and J. Liang, ”UNet++: A Nested U-Net Architecture for Medical Image Segmentation,” IEEE Transactions on Medical Imaging, vol. 39, no. 6, pp. 1856-1867, 2020, doi: https://doi.org/10.48550/arXiv.1807.10165.

[8] F. Loddo, P. Dario, U. Michelucci, and S. EL Ghazouali, ”BenchCloudVision: A Benchmark Analysis of Deep Learning Approaches for Cloud Detection and Segmentation in Remote Sensing Imagery,” arXiv:2402.13918v3, Mar 2024, doi: https://doi.org/10.48550/arXiv.2402.13918.

[9] J. Shi, A. Shirali, B. Jin, S. Zhou, W. Hu, R. Rangaraj, S. Wang, J. Han, Z. Wang, U. Lall, Y. Wu, L. Bobadilla, and G. Narasimhan, ”Deep Learning and Foundation Models for Weather Prediction: A Survey,” arXiv:2501.06907v1, Jan 2024, doi: https://doi.org/10.48550/arXiv.2501.06907.

[10] J. Zhang and M. Han, ”Ensemble Learning of Lightweight Neural Networks for Severe Convective Cloud Identification in Satellite Remote Sensing Images,” Remote Sensing, vol. 16, no. 12, p. 2070, 2024, doi: https://doi.org/10.3390/rs16122070.