NASA MODIS IMAGERY CLOUD IDENTIFICATION

Dongwei Fu
University of Colorado, Boulder
Boulder, CO 80309
dofu3785@colorado.edu

ABSTRACT

Clouds play an important role in moderating the Earth's climate. Satellite imagery, with its advantage of high spatial coverage is one of the most effective methods for studying clouds. In this project, I plan to investigate the problem of classification and segmentation of shallow cloud features in satellite imagery using Convolutional Neural Network (CNN) technique.

1. Introduction

Climate change is one of the most pressing challenges human-beings are currently facing. Cloud covers approximately 70% of the Earth's surface, playing a critical role in regulating the planet's climate by reflecting sunlight and trapping heat, which affects weather patterns and the global energy balance. Understanding cloud formations is essential for accurate climate modeling and predictions. Satellite imagery is the most effective approach for studying shallow cloud patterns, structures, and formation because it provides extensive spatial coverage. In this project, we will be using dataset curated by researchers from Max Planck Institute for Meteorology [1][2]. The dataset is provided as an online Machine learning competition posted on Kaggle [1]. It consists of hand labelled RGB satellite imagery taken from the Moderate Resolution Imaging Spectroradiometer (MODIS) onboard NASA's two polar-orbiting satellites, Terra and Aqua. The cloud labels are divided into four classes: Fish, Flower, Sugar and Gravel. Each class (label has its distinct cloud features.

2. Related work

Satellite image segmentation can be viewed as combining classification and localization. Through image segmentation, it partitions the image into smaller segments, and then tries to understand what is given at a pixel level and identifies the shapes and boundaries of the object. The final output of image segmentation is a mask where each element indicates which class the pixel belongs to. One of the most widely used image segmentation model architecture is U-Net [3]. Originally developed for biomedical image analysis, U-Net gets its name from its architecture, a U-shaped structure consists of two main components: an encoder network and a decoder network. The encoder compresses the input image by downsampling, and the decoder then upsamples the compressed features and restores the original spatial resolution. This design ensures that both contextual and detailed information are preserved, making it highly effective for tasks requiring precise localization of objects in images.

3. Proposed work

3.1 Dataset Exploratory Data Analysis

The dataset consists of 5546 (training) and 3698 (testing) RGB satellite imagery with dimensions of 1400 pixels by 2100 pixels taken from the Moderate Resolution Imaging Spectroradiometer (MODIS) onboard NASA's two polar-orbiting satellites, Terra and Aqua.

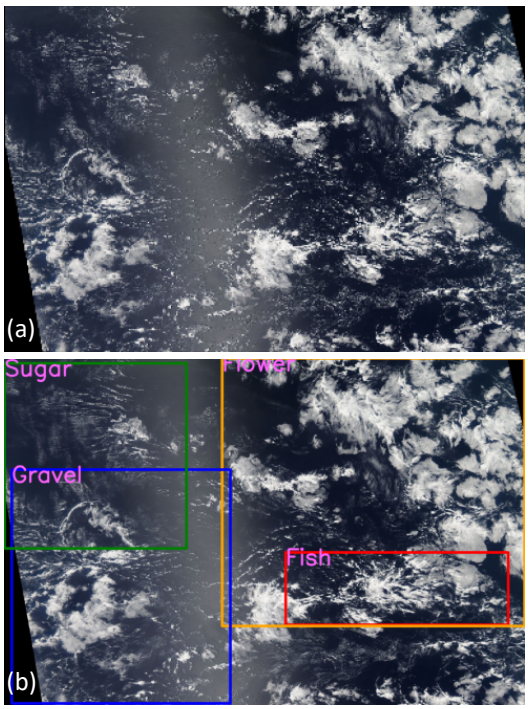

Fig 1. (a) Training Image 015aa06.jpg (MODIS true color RGB Image). (b) bounding box indicating cloud class label(s).

The images contain cloud structures from four class: Fish, Flower, Sugar and Gravel. Each class (label) has its distinct cloud features: For example, "Sugar" categorizes the dusting of very fine clouds with little evidence of self-organization, whereas "Flower" categorizes large-scale stratiform clouds that appears in bouquets with separation between each other; "Fish" refers to large-scale skeletal networks of clouds that are separate from other cloud formations, and finally "Gravel" refers to arcs of randomly interacting cells with granularity [2]. Fig. 1 shows one instance where all four categories appeared in one MODIS imagery. Note to the lower left corner of the imagery there are fill values (black), as there are gaps between different Aqua-MODIS ascending node orbits (daytime orbits) on a daily basis (MODIS has a wide swath of 2330 kilometers, providing near-global coverage every 1-2 days).

Fig. 2 shows the percentage of each class (label) in the training dataset. The Sugar category has the highest percentage at 31.7%, whereas Flower has the lowest percentage at 20.0%. Overall, the four classes are well balanced, and each class makes up nearly equal shares of the dataset. However, it is also worth noting that for the encoded mask (that indicate the class labels) for each image, nearly half (46.6%) of the masks are empty. Finally, the correlation heatmap between each categorical label is provided in Fig. 3. We

can see that cross-label correlation among different labels are generally low.

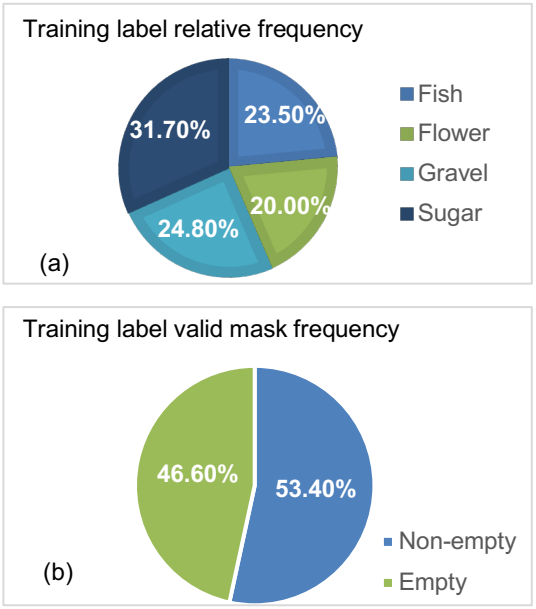

Fig 2. (a) Relative frequency of each label (for valid masks), (b) Percentage of non-empty and empty masks in the training dataset.

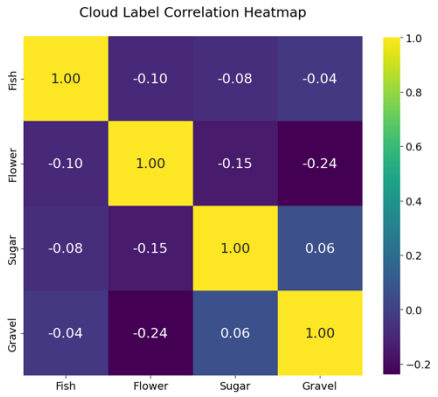

Fig 3. Heatmap of cloud labels from training dataset.

3.2 Data Preprocessing

The images at its original resolution of 1400 x 2100 pixels can be computational heavy for model training. Therefore, data augmentation was performed prior to training. Augmentation artificially expands the size of the dataset by creating modified data which improves the performance of the model to generalize. Albumentations library has been used for augmentation. This library efficiently implements an abundant variation in image transformation that are performance optimized. The training dataset images were split into batches of 32 and horizontal, vertical flips and random rotation (less than 45 degrees) were performed.

Pixel encoding technique was followed to participate in the submission of the competition since the image sizes were too large for Kaggle submission system [1]. As a result, instead of submitting an exhaustive list of indices for segmentation, pairs of values were submitted, which contained the start position and the run length of the image pixels. For example, a pair value of (1, 3) indicates that the pixel starts at 1 and run 3 pixels. The competition also required a space-delimited list of pairs. The predicted encodings were scaled by 0.25 per side, which scaled down the images of size 1400 × 2100 pixels in both train and test set to 320×480 pixels, hence allowing the scope to achieve reasonable submission evaluation times. This was achieved through run-length encoding of the segmentation mask.

3.3 Evaluation Metric

The evaluation metric used in this paper is the Dice coefficient [1, 4, 5]. It was applied to compare the pixel-wise agreement within a predicted segmentation and corresponding ground truth, using the following equation:

$$\frac{2 * |X \cap Y|}{|X| + |Y|} \qquad (1)$$

Here X defines the set of pixels predicted, whereas Y defines the ground truth of the training set. When X and Y are empty, the dice coefficient is defined as 1. The Kaggle competition Leaderboard score provides the mean of Dice coefficients for each (Image, Label) pair in the test data.

Dice coefficients are slightly different from the more popular evaluation metric: accuracy of a model. They are used to quantify the performance of image segmentation methods. Some ground truth regions are annotated in the images, and then an automated algorithm is allowed to do it. The algorithm is validated by calculating the dice score, which is a measure of how similar the objects are and is calculated by the overlap of the two segmentations divided by the total size of the two objects [5]. Dice coefficient works better in segmentation because of its ease of differentiation, as a result it is preferable over Jaccard's index, another evaluation metric similar to Dice coefficient [6].

The loss function for one image sample is defined as the summary of binary cross entropy loss and dice coefficient loss,

$$Loss = -(Loss_{BCE}(yt, yp) + Loss_{dice}(yt, yp)) \qquad (2)$$

Whereas *yt* is the ground truth, *yp* is the model's prediction.

3.4 Segmentation Model Architecture

As mentioned in Section 2, in this project I will focus on using U-Net model architecture, along with various backbones.

EfficientNet

EfficientNet is a family of convolutional neural networks (CNNs) introduced by Google, which revolutionized the approach to scaling deep learning models. The core innovation in EfficientNet is its compound scaling method, which uniformly scales a model's depth, width, and resolution using a simple scaling coefficient. Traditional methods typically scale one dimension at a time, such as increasing network depth or input resolution, often leading to suboptimal results. EfficientNet, however, applies a balance across these three dimensions, allowing for better performance without overburdening computational resources. The baseline EfficientNet-B0 model serves as a foundation, and its variations (B1 through B7) gradually increase in complexity and capability.

EfficientNet models are built using mobile inverted bottleneck convolution (MBConv) blocks and a scaling formula that applies different degrees of scaling to the model's dimensions. EfficientNet-B0 is the baseline model, and higher versions (like

EfficientNet-B7) apply the compound scaling formula to progressively increase the network's complexity and performance. Despite its simplicity, EfficientNet has been shown to significantly improve model efficiency, reducing computational cost without sacrificing accuracy. This makes it a popular choice for tasks like image classification, object detection, and segmentation.

While UNet is highly effective for tasks like medical image segmentation due to its unique encoder-decoder structure, EfficientNet provides several advantages in terms of performance and efficiency when adapted for similar tasks. One of the major advantages is EfficientNet's ability to handle larger and more complex datasets while maintaining lower computational requirements. The compound scaling strategy allows EfficientNet to improve accuracy without drastically increasing the number of parameters or computational load, which is crucial in resource-constrained environments.

In this project, due to the limited computational resources, I will mainly be using the baseline EfficientNet backbone combined with UNet decoder. The model architecture along with the encoder backbones are provided by Segmentation Models [7].

3.5 Proposed Timeline

10/01/24 – 10/10/24: Choose baseline model and perform model training and initial model evaluation. Fine tune model.

10/11/24 – 10/15/24: Finalize analysis and write final project report and presentation.

4. Preliminary Results and Evaluation

The first model run was performed using UNet with EfficientNetB0 as the encoder backbone. Learning rate was initialized at 0.0002. Fig. 4 shows the Dice coefficient and the model loss for both training and validation dataset.

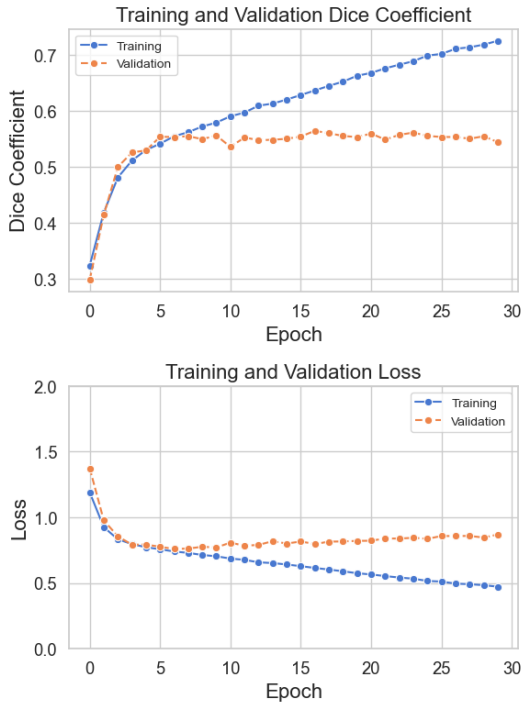

Fig. 4. Training and Validation (a) Dice Coefficient and (b) Loss.

The model's performance, as indicated by the training and validation Dice Coefficient and Loss curves, reveals some key insights into its learning behavior. The training Dice Coefficient steadily improves throughout the 30 epochs, suggesting that the EfficientNetB0 backbone effectively learns features from the training data. Similarly, the training loss decreases progressively, indicating that the model optimizes well on the training set. However, the validation Dice Coefficient plateaus early around epoch 6 and shows minimal improvement afterward, while the validation loss stabilizes without significant reduction after an initial decline. This disparity between training and validation performance suggests that the model is learning well on the training data but struggles to generalize to unseen validation data.

The early plateau in validation Dice Coefficient and the stagnation of validation loss point to potential overfitting, where the model memorizes the training data without effectively generalizing. Possible reasons for this behavior include the model's complexity in relation to the dataset size, class imbalance in the segmentation task, or the need for further regularization techniques such as dropout or weight decay. Adjusting the learning rate or employing learning rate schedules could also help prevent the model from plateauing too early.

Fig. 5 shows one test case using the baseline model. Fig. 5 shows for test dataset image ID '04326a2.jpg', all four cloud features were identified. Initial visual inspection shows that the model predictions are reasonable, as can be seen in Fig. 5, in the right section of the image the "Flowers" and "Fish" cloud formations are correctly identified. Yet "sugar" and "gravel" in the left showed some overlap. This will be further studied in the following assessments.

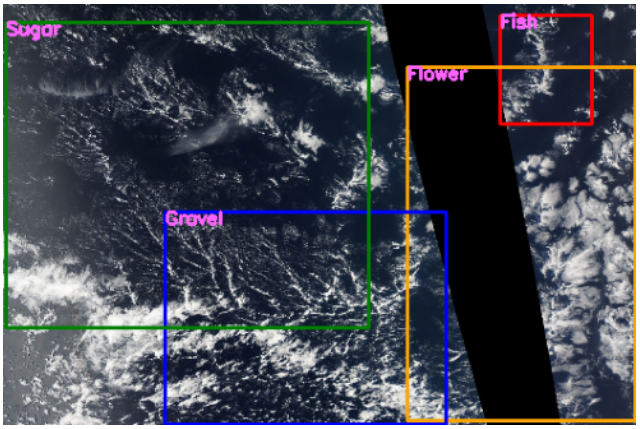

Fig. 5. Test case of model prediction on test image '04326a2.jpg'

5. Conclusion

So far we have successfully constructed the baseline model architecture for identifying cloud structures in satellite images. We utilized segmentation model package to construct a hybrid U-Net model with EfficientNet backbones and trained the model on preprocessed training images (after image augmentation). Preliminary results indicate that the baseline model has skill at identifying the four cloud labels from satellite images, with training Dice coefficient of 0.726 and validation Dice coefficient of 0.544. Upon visual inspection of the test cases ran on test images, the model was capable of detecting and identifying the four cloud labels from test images.

6. References

[1] Rasp S., H. Schulz, R. Walter, and M. Demkin. 2019. Understanding Clouds from Satellite Images. Kaggle. https://kaggle.com/competitions/understanding_cloud_organization

[2] Rasp, S., H. Schulz, S. Bony, and B. Stevens. 2020. Combining Crowdsourcing and Deep Learning to Explore the Mesoscale Organization of Shallow Convection. Bull. Amer. Meteor. Soc., 101, E1980–E1995, https://doi.org/10.1175/BAMS-D-19-0324.1.

[3] Ronneberger, O., Fischer, P., & Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. Lecture Notes in Computer Science, 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

297–302, https://doi.org/10.2307/1932409.

[4] Zou, Kelly H., et al 2004. Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index. Academic Radiology, vol. 11, no. 2, 1 Feb. 2004, pp. 178–189, www.ncbi.nlm.nih.gov/pmc/articles/PMC1415224/, https://doi.org/10.1016/S1076-6332(03)00671-8.

[5] Dice, Lee R. 1945, Measures of the Amount of Ecologic Association between Species. Ecology, vol. 26, no. 3, pp. 297–302, https://doi.org/10.2307/1932409.

[6] Bertels, Jeroen, et al., 2019. Optimizing the Dice Score and Jaccard Index for Medical Image Segmentation: Theory and Practice. Lecture Notes in Computer Science, pp. 92–100, https://doi.org/10.1007/978-3-030-32245-8_11.

[7] Iakubovskii, P. 2019. Segmentation Models. GitHub repository. Retrieved from https://github.com/qubvel/segmentation_models