Cloud Classification from Satellite Images

University of Massachusetts Amherst

Bryce Bodley-Gomes, Kaivan Shah

The Problem

Shallow clouds play a huge role in determining the Earth's climate. By classifying different types of cloud organization it will help us build better climate models.

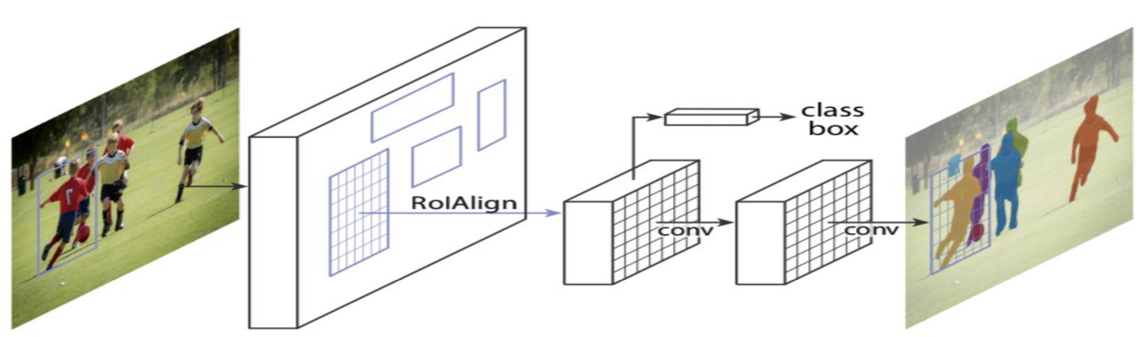
There are many ways in which clouds can organize, but identifying the boundaries between different forms of organization is challenging. The human eye, however, is fairly good at detecting features—such as clouds that resemble flowers.

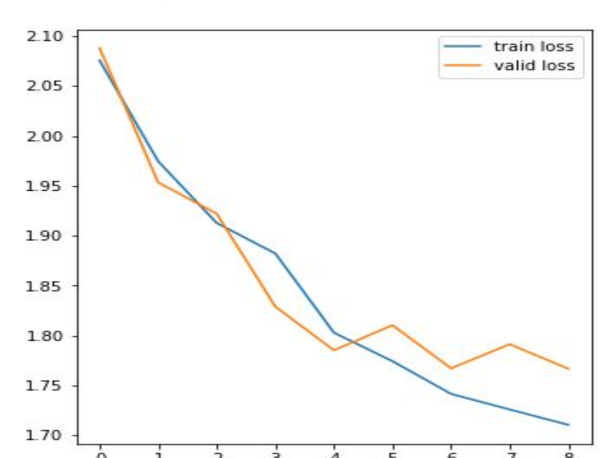
In this project we are supposed to do multiclass segmentation: finding the regions of image dominated by one of the four different cloud patterns (sugar, flower, fish and gravel) in the images.

Our Approach

In this challenge, we build a model that classifies the cloud organization patterns and returns a bounding box (encoded pixels). For this purpose we figured, that Mask RCNN would be a perfect architecture, as it outputs the class label, bounding box coordinates for each object, and an object mask. It's based on Feature Pyramid Network (FPN) and is based on a ResNet101 backbone.

We use Mask RCNN that has pretrained weights for the MS-Coco dataset and then train it on the cloud satellite image dataset provided in this kaggle challenge.





We trained our model for 9 epochs (taking approximately 8-9 hours). Then to amortize overfitting issues, we picked the model version with the minimal validation Loss.

The mask RCNN model generates bounding boxes and segmentation masks for each instance of an object in the image. Due to the rectangular shape of the groundtruth, we have decided to use the ROIs and not the masks given by the MaskRCNN.

Dataset and Output

Dataset: The dataset consists of satellite images that contain certain cloud formations, with label names: Fish, Flower, Gravel, Sugar. For each image in the test set, it needs to be segmented into the regions of each cloud formation label. Each image has at least one cloud formation, and can possibly contain up to all all four.

The dataset size is ~6GB and consists of the following images:

- **Training Data**
  - 5546 training images (with areas labeled as encoded_pixels)
  - We split our training data with 90% train and 10% validation (fairly standard).
- **Testing Data**
  - 3697 test images. Kaggle withholds the labels for the test images.

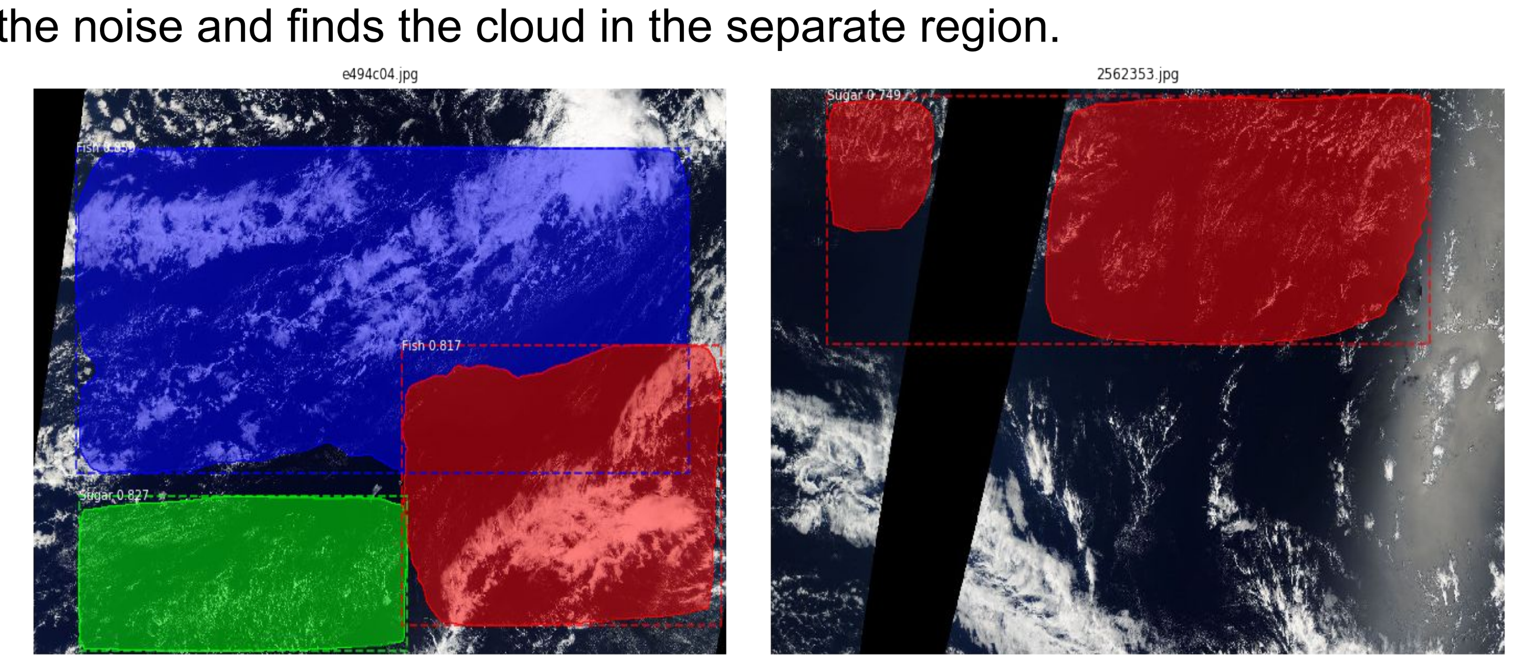
Output: The output is the label name and the encoded pixels of the box on the image containing that said label.

	Image_Label	EncodedPixels
0	0011165.jpg_Fish	264918 937 266318 937 267718 937 269118 937 27...
1	0011165.jpg_Flower	1355565 1002 1356965 1002 1358365 1002 1359765...
4	002be4f.jpg_Fish	233813 878 235213 878 236613 878 238010 881 23...
5	002be4f.jpg_Flower	1339279 519 1340679 519 1342079 519 1343479 51...
7	002be4f.jpg_Sugar	67495 350 68895 350 70295 350 71695 350 73095 ...

Our Results

Accuracy Scoring: This competition is evaluated on the mean Dice coefficient. The Dice coefficient can be used to compare the pixel-wise agreement between a predicted segmentation and its corresponding ground truth. The formula is given by: $2*|X \cap Y|/(|X|+|Y|)$

Training Results: After we trained our model we intended to try to analyze why current models' accuracies were so low (we predicted around 50-65% based on other kaggle teams results). Our actual accuracy which we obtained was 58.89%. In comparison, the best team obtained accuracy of 67.175% using UNET architecture with classification head. Thus, while our resulting accuracy is low, we can consider it somewhat comparable to the leaderboards UNET based architectures. In the images below, our Mask RCNN finds the labels, mask and boxes for the images. In image1, it can differentiate between 3 different type of clouds while in image2 it avoids the noise and finds the cloud in the separate region.

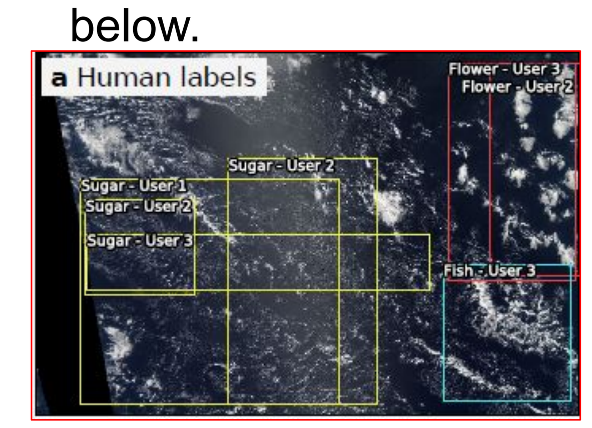


Current Work and Analysis

- **Mispredictions**
  We aim to see if there is misclassification of a specific class due to similarity of cloud types displayed images. We aim to do this by analyzing our predicted labels in the validation set and comparing them to the corresponding labels provided with the dataset (the gold labels). Our preliminary analysis shows that this is only a minor issue with our current model.
- **Under or Over classification**
  We also aim to check if we get false positives due to predicting cloud types in images where there are no regions containing a specified cloud type. This can be implemented by scanning through our dataset and checking for images where we predicted labels that are NOT contained in the gold labels for the selected image.
- **Overly large areas picked**
  Currently due to simplicity in our initial model design, we are using the bounding boxes to create encoded runs. We aim to revise our code to use the ROI to provide finer grained predictions. This should probably bring us up into the 60-65% accuracy which would be more inline with the other teams' work.

Conclusion

Cloud classification is hard primarily due to imprecise definitions on where to segment an image to most accurately discriminate segments of an image where one cloud type dominates. Even humans have clear difficulties consistently labeling the data as shown by the image below.



Also due to the connected nature of clouds, the bounding boxes drawn with ML are often, overly large bounding boxes (as is shown in our results to the left). While our model overbounds regions of interest, even the top scoring models perform similar errors quite often.

As such, there is still clearly much work that needs to be done to develop better segmentation models for learning on non-discrete shapes such as clouds.

Resources/References

[1] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn, 2017.

[2] https://www.kaggle.com/c/understanding_cloud_organization/

[3] Stephan Rasp, Hauke Schulz, Sandrine Bony, and Bjorn Stevens. Combining crowd-sourcing and deep learning to understand meso-scale organization of shallow convection, 2019