Cloud Pattern Recognition Using Deep Learning Methods

# Abstract

The main propose of this project is to understand clouds organization given the fix-sized satellite images, which is a part of the research of meteorologists. Our dataset is from an on-going Kaggle competition. The clouds in the image can be detected and classified as one of four patterns, including Fish, Flower, Sugar and Gravel, each of which has important research meaning in Meteorology. In this project, we will apply two kinds of machine learning models as detector and classifier and analyze their application results on the given dataset. All of methods we plan to use are based on deep learning theory and Convolution Neural Network model. We will first use data augmentation to enhance our dataset, and then train the model based on image detection algorithm and image segmentation algorithm separately, with cross-validation embedded, and finally tune the parameter to get the best prediction model.

# Introduction:

Our project is to identify regions in satellite images that contain certain cloud formations, with label names: Fish, Flower, Gravel, and Sugar. It is an on-going Kaggle competition named “Understanding Clouds from Satellite Images”, and the dataset is private for participants only until the end of competition by Nov 18, 2019. The images were downloaded from NASA Worldview, and three regions, spanning 21 degrees longitude and 14 degrees latitude, were chosen. The labels are the result of collaboration of 68 meteorologists and 3 of them label the pattern manually with professional knowledge. [1]

We all regard it as a meaningful participation because climate is no doubt one important topic in nowadays. Building a machine learning model on this dataset to classify different cloud classes can help scientists extend their analysis, so that they can better understand what drives these clouds and what effects they have on climate, eventually lead to better climate prediction. Moreover, it is also a challenging problem. None of the clouds have the same shape, which make it hard to be classified. Also, the shapes of clouds are irregular, we cannot just use a rectangle as boundary. The background of the image can also interfere the result of our model.

# Method

In this project, we will mainly apply two different kinds of models, named Image Detection and Image Semantic Segmentation, and compare their performance.

In terms of Image Detection, we would use a method based on deep learning. The mainstream framework of Image Detection Neural networks, like Faster R-CNN, has four steps. [2] First, we need to preprocess the original images. We need to rearrange the shape of input images and enhance information. Then, we would train a convolutional neural network or use pre-defined anchor boxes to extract and then reshape the Regions of Interest. Later, we would let the features pass through a fully-connected neural network. And finally, we would use a SoftMax layer to do the classification. Along with the SoftMax layer, we will use a linear regression layer to output the bounding box coordinates for the prediction object.

For the part of Image Semantic Segmentation, the most classical model is Fully Convolutional Network (FCN). When classifying images, the last layers of neural networks are fully-connected. However, FCN replaces the layer with transposed convolutional layer, which allows the network to output images. [3] Inspired by FCN, our method will have two steps. We will firstly extract the feature from input images using convolutional neural network, and then apply transposed convolutional layer to recover the spatial information from the feature we get in the first step. Residual learning will be implemented in order to avoid gradient vanishing.

We will improve the model performance by modifying preprocessing, the structure of the model and classification methods. Through a deeper exploration of the problem, we may also use or propose a better framework to solve the problem, and conduct detailed comparative experiments.

# Related Work

**a. Theoretical**

Due to the complexity of the problem, we believe deep learning will be useful to solve this problem. We search the Internet for theoretical explanation of deep learning and conduction of relative methods.

In 2015, Yann el al. [4] published an overview about deep learning, in which algorithms and applications are introduced. They concluded that, different from conventional machine-learning techniques in which experienced engineers are needed to construct pattern-recognition systems, deep-learning methods allows a machine to be fed with raw data and to automatically discover the representations needed for detection or classification. By composing simple but non-linear modules, deep learning can generate higher level of representation with lower one, and therefore, complex functions can be learned after training. Commonly, object detection and image segmentation are two applications which can solve the problem we interest.

Object detection is to detect same kind of objects in different images. This method can be generally divided in 2 steps. The first step is to extract the features of target object in the training set, usually with convolutional neural networks or other extended CNN algorithms. And the next step is to find some part of images with the same feature. [5]

Image segmentation is the process of partitioning images into different segments. This method assigns a label to each pixel such that pixels sharing the same label are similar with respect to some characteristics or computed properties. Then the pixels who have the same label will be viewed as a region, and adjacent regions will have different meaning with respect to the same characteristic. [6]

**b. Practical**

As mentioned before, this is an on-going Kaggle competition. Because of our late participation, there are already some participants who have shared their methods on this topic, which can be easily found in the Notebook section. We take Andrew's kernel [7] and Mobassir’s kernel [8] as our reference, due to highest up-votes and highest predict accuracy, respectively.

Andrew's work is mainly based on image segmentation in Pytorch. After noticing that many labels can be on one picture and overlap with each other, Andrew took the strategy that to train 4 models separately for each label and combine the prediction of these models as finally result. He made a stratified split of original training set with ratio of 0.1 as training and validation, then applied augmentation on split training set. After the preparation of training data, he took dice loss as the loss function and used the pre-training model - resnet50 to train the model, and got his final result 0.645 after optimizing the threshold and parameters.

Holding the same idea with Andrew, Mobassir also used image segmentation to train model, however, we need to pay special attention on two main differences. One is that Mobassir defined a “callback” to estimate AUC under PR curve for each class to reduce learning rate, and get early stop if there is no improvement after 5 epochs. And the other is that, instead of using resnet50, Mobassir used efficientnetb3 as pre-training model and implemented fully-connected layer during the training process, which maybe the main reason counting for higher prediction accuracy (0.658).

**c. Our Novelty**

Obviously, for such an image pattern classification question, Deep Learning and Fully Convolution Network are two common solutions. However, there are still something novel we can do to build our answer. If we change the common strategy a little bit, a new solution can be formed by using image detection to find areas where we may interest and then applying classifier to judge the label of this area. This method may seem stupid because one more step is needed, but if we can get higher prediction accuracy, such a minor sacrifice is completely acceptable. And also, in the preparation of training, we can implement Cross-Validation when making the split of training data, which others had not done so.

# Plan of Experiment

**a. Implementation**

The project will be implemented using Pytorch 1.0.1 with python 3.7.2. First of all, we will preprocess the data by normalization and augmentation. And then we will implement object recognition and image segmentation separately to solve the problem. At last, we will evaluate these two models and compare the result, and make a parallel comparison with other existing methods.

**b. Dataset**

The dataset contains 5546 images in train set and 3698 images in test set. For each image, it can have up to 4 masks called Fish, Flower, Gravel and Sugar. In the whole training set, there are 3751 Sugar masks, 2939 Gravel masks, 2781 Fish masks and 2365 Flower masks. And among all of the training images, 2372 of them have 2 masks, 1560 of them have 3 masks, 1348 of them have 1 mask and 266 of them have all of the 4 masks. The ground truth of each image is given by an encoded format. Each image is represented by one vector, and the positions of specific mask are implied by a list of pairs that contain a start position and a run-length. E.g. ‘1 3 10 5’ implies pixels 1,2,3,10,11,12,13,14 are to be included in the mask.

**c. Evaluation**

The result of the model 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:

Where X is the predicted set of pixels and Y is the ground truth. It is defined to be 1 when both X and Y are empty. [9]

# Plan of Project

There are 4 teammates in our team labeled as A, B, C and D. In our project, two different models will be compared, both of them are based on Convolutional Neural Network. So we can divide the coding work into three parts, data preprocessing, model establishment, and model tuning and evaluation. The timeline except the review and poster is from 1st Nov. to 1st Dec. So we can assign work weekly.

In the first week, we will discuss the model details together and then teammates A and B will preprocess the data, do the data augmentation while teammates C and D will start the report sections such as introduction and Problem statement.

In the second and third weeks, we plan to divide our team into two parts, teammate A and C implementing the Object Detection model and teammate B and D implementing the Segmentation model.

During the last week, we will focus on tuning, evaluation work, and the final report. Teammates C and B will tune, evaluate and compare the model and results while teammate A and D will focus on the final report.

After that, we will do the review and the poster presentation design together.

According to the planned timeline, we have enough time to handle the project. The most important part of our project is the establishment of the Neural Network. Since we decided to use Pytorch to build our complex neural network, the workload becomes acceptable for one month. Some of our teammates have experiences in Computer Vision and familiar with Pytorch, which makes our project more feasible. Also, the GPU computation resources can make us evaluating and testing our model more efficiently.

We expect some difficulties would be mainly on the implementation of data augmentation, i.e. how to transform the images with their corresponding masks. And also, tuning and debugging should be a hard challenge.

# Reference:

[1] Understanding Clouds from Satellite Images. *Kaggle*. (2019). Retrieved October 26, 2019, from: https://www.kaggle.com/c/understanding\_cloud\_organization/data

[2] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[3] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[4] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[5] Object detection. *Wikipedia*.(n.d.). Retrieved October 26, 2019, from: https://en.wikipedia.org/wiki/Object\_detection

[6] Image segmentation. *Wikipedia*.(n.d.). Retrieved October 26, 2019, from: https://en.wikipedia.org/wiki/Image\_segmentation

[7] Andrew. (2019, September). Segmentation in PyTorch using convenient tools (version 37). *Kaggle*. Retrieved October 26, 2019, from: https://www.kaggle.com/artgor/segmentation-in-pytorch-using-convenient-tools

[8] Mobassir. (2019, October). Keras efficientnetb2 for classifying cloud (version 13). *Kaggle*. Retrieved October 26, 2019, from: https://www.kaggle.com/mobassir/keras-efficientnetb2-for-classifying-cloud#Libraries

[9] Understanding Clouds from Satellite Images. *Kaggle*. (2019). Retrieved Oct. 26, 2019, from: https://www.kaggle.com/c/understanding\_cloud\_organization/overview/evaluation