Clearing the Haze from Climate Models

Cloud Organization Classification

Prerna Jain prernaj@gatech.edu
Arnav Jeurkar ajeurkar3@gatech.edu
Priyank Madria priyankmadria@gatech.edu
Shubham Singhal ssinghal37@gatech.edu

ABSTRACT

Cloud climate feedback is one of the most challenging research areas in climate science. Traditional rule-based algorithms fail to create boundaries between different forms of clouds and thus there is a need to improve classification methods to help scientists understand how clouds will help to predict future climate change. We aim to leverage deep learning models to improve the identification of cloud organization patterns and store them within a database called Eva. This can be interpreted as a multiclass segmentation task that can be implemented using various deep learning frameworks.

KEYWORDS

Deep Learning, Computer Vision, Eva, Segmentation, Object detection, Classification

WHAT IS THE PROBLEM

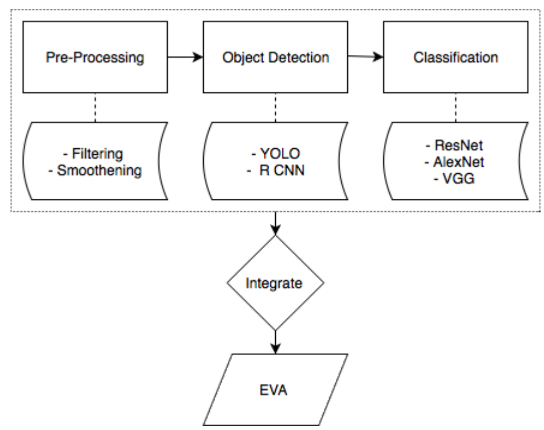
This project is aimed at applying deep learning to better quantify and understand the role of clouds in climate models. This research will guide the development of next-generation models which could reduce uncertainties in climate projections. We aim to learn a model to identify patterns of cloud organization from satellite images. This involves detecting sections within the image which have a potential organization pattern (sugar, gravel, fish, or flower), classifying the organization, and improving the section boundaries to only enclose the pattern.

In addition, we also aim to incorporate our data and model into Exploratory Video Analytics (Eva) framework which is currently primarily used to optimize query processing for surveillance video data. By creating a catalog containing our images and model along with user-defined functions (UDF) to query this model, we will expand the EVA framework to process image data as well.

WHY IS THIS PROBLEM IMPORTANT?

Climate change has been increasingly driven by human influence since the past few decades. Its manifestation in the form of adverse meteorological phenomena is one of the important facets which directly impacts lives on a great scale. Precipitation—a key link between the atmosphere, oceans, land surface, and the cryosphere—is increasing over the northern middle–high latitudes, especially in autumn. Over the United States, annual precipitation has decreased in the Southwest but increased over the Great Plains, Midwest, and the Northeast; U.S.-averaged precipitation has increased by about 4% since 1900. The combination of warmer temperatures and reduced precipitation in some regions has increased the risk of drought and drought-related impacts. Areas that receive limited precipitation, sometimes called drylands, are also increasing in area. At any given moment, about two-thirds of our planet is covered by clouds. Shallow clouds play a huge role in determining the Earth's climate. They are also difficult to understand and to represent in climate models. By classifying different types of cloud organization, we hope to improve our physical understanding of these clouds, which in turn will help us build better climate models.

HOW WILL THE TEAM SOLVE THIS PROBLEM?



Our image processing pipeline will be implemented using the following workflow:

1. Preprocessing:
Cloud images are taken from satellite videos which contain cloud shapes and potentially noise and other extraneous variables. We are going to remove those variables as best as we can through the implementation of either:[2]

a.) Sobel Filter - Sobel filter is based on convolving the image with 3x3 kernel to calculate gradients. We run the convolution horizontally and vertically to find gradients at each pixel that correspond to its edges.

b.) Canny Edge Detector - Canny edge detection uses Gaussian filtering to remove noise and hysteresis thresholding to detect edges in an image.

2. Segmentation[4]:
After preprocessing, we are required to generate the model which will segment the image to find boundaries around the objects. This will give us borders around the images which can be used to encapsulate specific cloud images. From there, we are going to try and mask (depending on how much noise is actually there) the images and even work towards contorting the boundaries to fit around the image but being careful not to overfit the boundaries. Proposed methods include:

a.) YOLO[3] - YOLO divides the image in grids and tries to find similar pixels to detect objects. It uses a single neural network with prediction probabilities to score each of the images.

b.) Mask R CNN[1] - Mask R CNN is the extension of Faster RCNN which uses convnet to find feature maps in the image.

Those feature maps further passed through region proposed network which detects the bounding boxes around the object.

3. Classification:
Once we determine masks, we will feed that information to the classification model for which we'll be using the following Neural Networks

a.) Deep Neural Network - Deep Neural network can be trained with labeled data to learn the patterns in the image to classify testing images. We might use deep neural networks to test our cloud images once we train the network. Some tools we can use or implement are:

b.) Mask R CNN[1] - As we discussed above Mask R CNN can detect boundaries around the objects, the next layer in Mask R CNN is a connected layer that classifies the object.

c.) YOLO[3] - A layer within the YOLO can also be deployed to classify objects once detected. YOLO detects objects by applying localization and classification to each grid.

d) Both Mask RCNN and YOLO uses Intersections over Unions to threshold object detection and classification. This is a powerful technique to avoid detecting redundant objects within the main object.

4. Hyperparameter Tuning
We will be experimenting with machine learning techniques like cross validation to optimize our models to improve accuracy and avoid overfitting and underfitting. From this, we will then also tune such parameters depending on the model we use (i.e. Resnet, VGG)

5. Integration with EVA
We will be extending the catalog and UDF to query images and object detection. Some options are:

a.) Extending the catalog - Adding the dataset of clouds to improve over the existing catalog. Currently, there are only cars in the database.

b.) Adding UDF - Integrating another UDF which could query the image dataset to detect objects in that. This query will help to access our cloud information.

VALIDATION OF IMPLEMENTATION

To validate our implementation, we are using 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.This can be used to compare the pixel-wise agreement between a predicted segmentation and its corresponding ground truth. The formula is given by:

$$\frac{2*|X\cap Y|}{|X|+|Y|}$$

Here X is the predicted set of pixels and Y is the ground truth. The Dice coefficient is defined to be 1 when both X and Y are empty. This <Image, Label> pair is what is being tested for each pair in the test set. The final is done through the mean of the Dice Coefficients. [5] Dice coefficient = 1 would indicate that the predicted mask exactly matches the ground truth mask.

We'll also be exploring the possibility of querying the EVA catalog using UDF to retrieve objects from cloud frames. To have a successful Eva implementation, the UDF that we create will correctly access our information in the catalog when retrieving. Furthermore, we will also make sure that our catalog is correctly set up and contains the entirety of the model.

REQUIRED RESOURCES

Software
- Python
- Keras
- Tensorflow
- Eva
- Pytorch
- YOLO
- R CNN

Hardware
- CPU
- GPU

Dataset
- Kaggle Cloud dataset

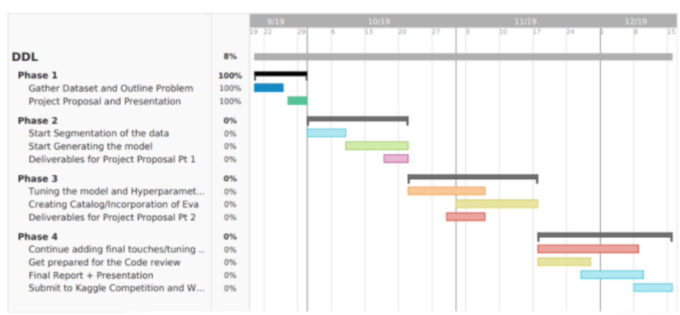
GOALS

The goals we aim for this project is as follows:

Sub-optimal Goal: Complete Kaggle competition with reasonable accuracy from our model. This will have very limited to no access to the Eva.
Goal: Create a catalog in Eva with our data and also at least match the highest accuracy (measured through Dice Coefficient) on Kaggle.
125%: Create a successful UDF within Eva with our data within the catalog. Also, we would beat the highest accuracy achieved so far within the Kaggle Competition and hopefully win. Potentially look into incorporating YOLO within Eva.

TIMELINE

The timeline for our project is shown below.



REFERENCES

[1] Gkioxari, Piotr, Girshick, & Ross. (1970, January 1). Mask R-CNN. Retrieved from http://openaccess.thecvf.com/content_iccv_2017/html/He_Mask_R-CNN_ICCV_2017_paper.html.

[2] Gupta, Gaurav & chandel, ruchika. (2013). Image Filtering Algorithms and Techniques: A Review.

[3] Sharma, P. (2019, May 6). A Practical Guide to Object Detection using the Popular YOLO Framework. Retrieved from https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framewor-python/.

[4] Sharma, P. (2019, July 22). Step-by-Step Tutorial on Image Segmentation Techniques in Python. Retrieved from https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/.

[5] Understanding Clouds from Satellite Images. (n.d.). Retrieved from https://www.kaggle.com/c/understanding_cloud_organization/overview/evaluation.