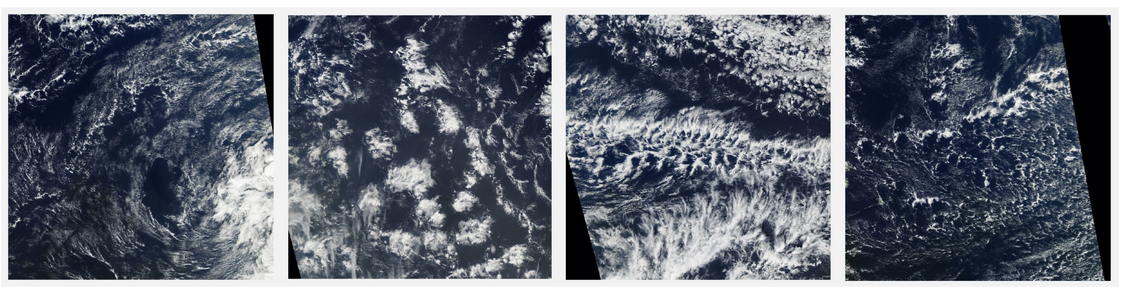
SEGMENTATION OF CLOUD PATTERNS FROM SATELLITE IMAGES TO IMPROVE CLIMATE MODELS

Kirill Vishnyakov[†] and Mukharbek Organokov[‡]
[†]Peter the Great St.Petersburg Polytechnic University, [‡]Université de Strasbourg
kirill.a.vishnyakov@gmail.com, mukharbek.organokov@gmail.com

Introduction

Climate change has been at the top of our minds and at the forefront of important political decision-making for many years [1]. Classification of different types of clouds is substantial for understanding climate change. Human ability to identify patterns is limited and murky boundaries between different forms of clouds lead to obstacles in traditional rule-based algorithms cloud features separation [1]. In these situations, machine learning techniques, particularly deep learning, have demonstrated their ability to mimic the human capacity for identifying patterns in the clouds using satellite images [2]. This work focuses on segmentation of four subjective patterns of clouds organization [2] (see Fig. 1): Sugar, Flower, Fish, Gravel.



(a) Sugar **(b)** Flower **(c)** Fish **(d)** Gravel

Figure 1: Canonical examples of the four cloud organization patterns.

Method

Dataset

The dataset used in the analysis is released on the Kaggle platform [1]. The images with three regions, spanning 21 degrees longitude and 14 degrees latitude, were taken from NASA Worldview [3]. The labels were created on a crowd-sourcing platform [4]. On cloud labeling days at two institutes, 68 scientists screened satellite images, identified areas of cloud patterns in each image, and each image was labeled by approximately 3 different scientists [1]. Ground truth was determined by the union of the areas marked by all labelers for that image, after removing any black band area from the areas [1]. The ground-truth masks presented in the dataset are quite noisy meaning they include a lot of areas that actually do not contain clouds at all. Also, the masks of different classes can overlap. These two facts significantly increase problem complexity. The resolution of original images is 1400×2100, in our experiments we use downscaled images, namely we work with three different image resolutions: 352×512, 512×768, 768×1152. Example of images containing different number of clouds patterns is shown in Fig. 2.

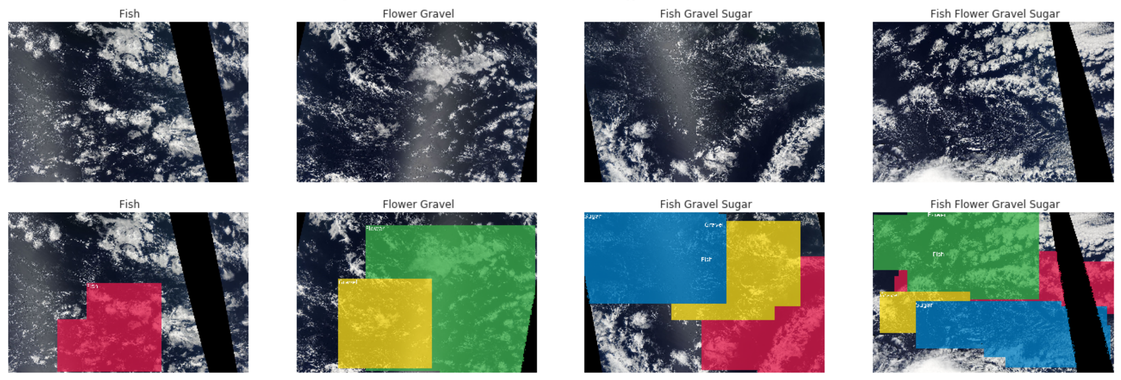


Figure 2: Images of clouds with corresponding masks.

Metric

The mean Dice coefficient (DC) [5] is chosen as the metric to measure the performance of the model by pixel-wise comparison of the predicted segmentation (Y_{pred}) output and the corresponding ground truth (Y_{gt}). If $Y_{\text{pred}} = \varnothing$ and $Y_{\text{gt}} = \varnothing$, then DC = 1; otherwise, it is defined as:

$$\frac{2|Y_{\text{pred}} \cap Y_{\text{gt}}|}{|Y_{\text{pred}}| + |Y_{\text{gt}}|} \tag{1}$$

The objective of this research is to develop a DL model that ensures higher accuracy. Thus, the goal is to maximize the model on the mean DC so that a perfect model achieves an accuracy of 1.

Architecture and Implementation

This problem was treated as a semantic segmentation task with four different classes. During the past years numerous architectures (see Fig. 3) have been developed to tackle the problem of semantic segmentation including the most famous U-Net [6], LinkNet [7], FPN [8], PAN [9]. In this work, we used U-Net architecture.

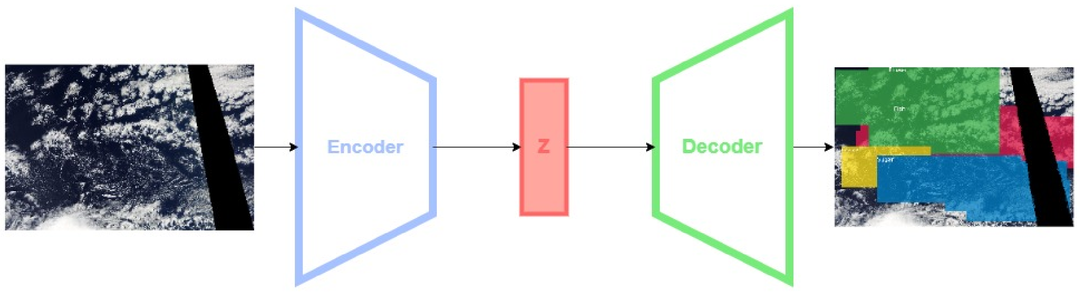


Figure 3: Simplified schematic view of the architecture.

We developed a two-stage (**S1** and **S2**) semantic segmentation pipeline [10] (see Fig. 4) that accepts images with corresponding ground-truth masks and outputs raw (non-normalized) predictions for each of four classes. Then during the post-processing procedure predictions are normalized with sigmoid function to eliminate masks of insufficient size. Thus, final processed predictions were generated.

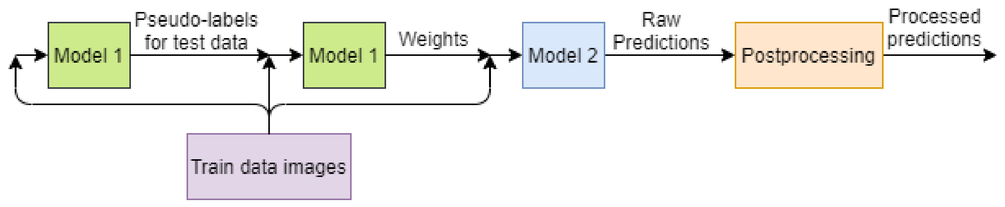


Figure 4: Pipeline.

Experiments

Our main experiments with data included training on two stages with different image resolutions and different set of pseudo-labels (PLs). Recently, a semi-supervised approach became popular and provided improvements in training procedure. We used approach similar to [11] in which authors first trained an EfficientNet model on labeled images and used it to generate PLs for unlabeled images; then a larger EfficientNet was trained on the combination of labeled and PL images [11].

Two sets of PLs were generated:

- **Only confident PLs**;
- **All images from test data selected for PL**.

Our hyperparameters configuration is the following:

- **Encoder**: EfficientNet-B0 [12];
- **Augmentations**: HorizontalFlip, VerticalFlip, ShiftScaleRotate, GridDistortion, OpticalDistortion, and RandomBrightnessContrast [13] – all with 50% probabilities;
- **Optimizer**: Adam [14];
- **Scheduler**: ReduceLROnPlateau;
- **Stopping criteria**: Early Stopping with patience = 5 and min_delta = 5e-4
- **Post-processing**:activation threshold=0.4, min_mask_size – depends on image resolution;
  - 352×512 – 2.5k pixels;
  - 512×768 – 5k pixels;
  - 768×1152 – 11k pixels.
- **Loss**: On **S1** stage we trained with BCE Dice Loss, on the **S2** stage we trained with Symmetric Lovasz [15, 16] Loss.

Results

In this work we have studied how choice of PLs and training with different image resolutions affects results given the noisy dataset. All results can be split in 3 following groups:

- **S1** trained with train data + set of confident PLs, **S2** trained only on train data.
- **S1** trained with train data + PLs of **all** test images, **S2** trained only on train data.
- Both, **S1** and **S2** stages trained with train data + PLs of **all** test images.

	352×512	512×768	768×1152
Pseudo-labels confident **only stage 1 stage**	0.64958	0.65486	0.65087
Pseudo-labels all **only stage 1**	0.64941	0.65356	0.65553
Pseudo-labels all **stage 1 + stage 2**	0.64840	0.65603	0.65389
Pseudo-labels confident **stage 1 + stage 2**	0.64518	0.64925	0.64763
352×512	0.65109	0.65318	
	0.65590	0.65582	
	0.65663	0.65394	
	0.65313	0.64665	
512×768	0.65041		
	0.65118		
	0.65141		
	0.64528		
768×1152			

Table 1: Summary of the results.

Conclusion

We have developed a two-stage segmentation pipeline with post-processing procedure which identifies location of four cloud organization patterns. We have studied how choice of PLs and image resolution affects the final results. Surprisingly, training on the biggest image resolution (768×1152) didn't yield the best results. Moreover, training with the confident PLs consistently couldn't outperform the result obtained from training with a set of all test images chosen as PLs.

GitHub

References

[1] Kaggle Understanding Clouds from Satellite Images, https://www.kaggle.com/c/understanding_cloud_organization [Accessed: 2020-06-30]
[2] Rasp, S., Schulz, H., Bony, S., et al. 2019, arXiv:1906.01906
[3] NASA Worldview, 1999, https://worldview.earthdata.nasa.gov/ [Accessed: 2020-06-30]
[4] Rasp, S., Schulz, H., Stevens, B., 2019 https://www.zooniverse.org/projects/raspstephan/sugar-flower-fish-or-gravel
[5] Dice, L. Ecology, 26(3), 297-302, 1945
[6] Ronneberger, O., Fischer, P., & Brox, T. 2015, arXiv:1505.04597
[7] Chaurasia, A. & Culurciello, E. 2017, arXiv:1707.03718
[8] Lin, T.-Y., Dollár, P., Girshick, R., et al. 2016, arXiv:1612.03144
[9] Li, H., Xiong, P., An, J., et al. 2018, arXiv:1805.10180
[10] Vishnyakov, K., Organokov, M., "Understanding Clouds from Satellite Images", GitHub repository, https://github.com/LightnessOfBeing/kaggle-understanding-cloud-organization
[11] Xie, Q., Luong, M.-T., Hovy, E., et al. 2019, arXiv:1911.04252
[12] Tan, M. & Le, Q. V. 2019, arXiv:1905.11946
[13] Buslaev, A. et al. Information. 11(2):125. 2020
[14] Kingma, D. P. & Ba, J. 2014, arXiv:1412.6980
[15] Charoenphakdee, N., Lee, J., & Sugiyama, M. 2019, arXiv:1901.09314
[16] Berman, M., Rannen Triki, A., & Blaschko, M. B. 2017, arXiv:1705.08790