UDK 004.93.1

Borys I. Tymchenko[1], Ph.D student, Institute of Computer Systems,
E-mail: tymchenko.b.i@opu.ua, ORCID: http://orcid.org/0000-0002-2678-7556
Philip O. Marchenko[2], Master student, Faculty of Mathematics, Physics and Information Technology,
E-mail: p.marchenko@stud.onu.edu.ua, ORCID: http://orcid.org/0000-0001-9995-9454
Dmitry V. Spodarets[3], Master of Science, CTO, Head of R&D,
E-mail: dmitry.spodarets@vitechlab.com, ORCID: http://orcid.org/0000-0001-6499-4575
[1]Odessa National Polytechnic University, Avenue Shevchenko, 1, Odesa, Ukraine, 65044
[2]Odessa I.I. Mechnikov National University, Dvoryanskaya str. 2, Odesa, Ukraine, 65014
[3]VITech Lab, Rishelevska St, 33, Odesa, Ukraine, 65000

SEGMENTATION OF CLOUD ORGANIZATION PATTERNS FROM SATELLITE IMAGES USING DEEP NEURAL NETWORKS

Abstract. *Climate change is one of the most important challenges that humanity faces now. The essential part of climate models is the movement of clouds, which affects climate parameters dramatically. Shallow clouds play a huge role in determining the Earth's climate. They're also difficult to understand and to represent in climate models. Unfortunately, the exact modeling of clouds movement is notoriously tricky and requires perfect knowledge of underling physical processes and initial states. Boundaries between different types of clouds are usually blurry and difficult to define with rule-based decision systems. Simplification of the segmentation step is crucial and can help researchers to develop better climate models. Convolutional neural networks have been successfully applied in many similar areas, and for cloud segmentation itself, too. However, there is a high cost of good, pixel-level labeled datasets, so the industry often uses coarse-labeled datasets with the either region or image-level labels. In this paper, we propose an end-to-end deep-learning-based method for classification and segmentation of different types of clouds from a single colored satellite image. Here, we propose the multi-task learning approach to cloud segmentation. Additionally to the segmentation model, we introduce a separate classifier that uses features from the middle layer of the segmentation model. The presented method can use coarse, uneven and overlapping masks for clouds. From the experimental results, the proposed method demonstrates stable results and learns good general features from noisy data. As we observed during the experiments, our model finds types of clouds, which are not annotated on the images but seem to be correctly defined. It is ranked in top three percent competing methods on Understanding Clouds from Satellite Images Dataset.*

Keywords: *deep learning; satellite imaging; deep convolutional neural network; multi-target learning; cloud formations classification; Kaggle; meteorology*

1. Introduction

Today, all issues related to climate change are at the forefront of discussions and different political decisions. Quite a lot of companies pay serious attention to the question of affecting the climate by their by-products, as new and strict standards appear which are designed to preserve nature and regressive climate changes. Neglect of these standards may be a reason for not just only a public censure, but also a reason for significant fines. Thus, the resulting need to maintain the climate in a normal state entailed the need to analyze and build patterns of climate behavior.

One of the most valuable features in determining the Earth's climate model is the be-havior of clouds. However, investigating their be-havior is one of the trickiest parts, as it requires a perfect understanding of underlying processes in the atmosphere. By classifying different types of cloud organizations, researchers hope to improve the phy-sical understanding of these clouds, which in turn will help us build better climate models.

Researchers at Max Planck Institute for meteo rology collected clouds photo, roughly 10 000 21◦× 14◦ (long-lat) Terra and Aqua MODIS visible images from NASA Worldview. Thanks to the crowd-source community at Zooniverse, they created an annotated dataset with four types of clouds (Sugar, Fish, Gravel and Flower) [1]. However, the annotation of types and regions on photos was quite noisy, so researchers asked for a stable and well-generalized solution to classify and segment types of photos on unseen images.

As rule-based approaches to solve the task of classifying and segmentation of cloud types give poor results, machine learning approaches come to help. Convolutional neural networks have been successfully applied in different areas of computer vision, also in classification and segmentation. In this paper, we propose an end-to-end method for the segmentation of four types of clouds from an RGB satellite photo. Also, we propose a pre- and post-processing pipeline which helps our model to learn good discriminative features from the noisy data. This method was ranked 46 of 1538 competing methods (Dice score of 0.66322) on Understanding Clouds from Satellite Images Dataset.

Our code and experiment results are available by the following link:
https://github.com/spsancti/kaggle-clouds/.

© Tymchenko, B. I., Marchenko, Ph. O., Spodarets, D. V., 2020

This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/deed.uk)

2. Related works

To describe a wide range of research works as possible, we review the works for RGB color satellite images as well as other optical sensors such as multispectral/infrared.

One of the methods for cloud detection was designed by Zhu et al. [2] they provided a method called Fmask (Function of the mask) for cloud and cloud shadow detection in Landsat imagery. Fmask uses rules-based approaches on cloud physical properties to separate potential cloud regions from a clear sky. As input data, they used information from seven bands of enhanced thematic mapper (ETM) and enhanced thematic mapper plus (ETM+) sensors. Another approach was provided by Harb et al. [3], they analyzed CBERS medium resolution multispectral data and their algorithm uses a set of literature indices, as well as a set of mathematical operations on the spectral bands, in order to enhance the visibility of the cloud/shadow objects. These methods performed accurate results, nevertheless, these methods highly depend on sensor models (i.e. since they are rule-based methods) and proposed solutions are not scalable to other types of sensors.

To generalize solution, Hu et al. [4] used computer vision techniques to extract several low-level features such as color, texture features, etc. to estimate pixel-level masks using classical ML algorithms. Ozkan et al. tried to apply deep neural networks (such as Feature Pyramid Network) to segment clouds from low-orbit Gokturk-2 and RASAT satellites RGB images.

Not only satellite images were analyzed by researchers. Zhang et al. [5] proposed a new convolutional neural network model, called CloudNet, for accurate ground-based meteorological cloud classification. They built a ground-based cloud data set, called Cirrus Cumulus Stratus Nimbus, which consists of 11 categories under meteorological standards. The total number of cloud images is three times that of the previous database.

3. Problem statement

3.1 The Dataset

The image data used in this research was taken from a single dataset. We used Understanding Clouds from Satellite Images Dataset (UCSID). It consists of 10000 RGB images taken from two polar-orbiting satellites, TERRA and AQUA [6], each of which passes a specific region once a day. Because of the limited field of view of the imagers installed on these satellites, each image is stitched together from two places on orbit. The remaining area that has not been covered by two orbits if filled black. There are compression artifacts, so the black color has some small-valued anomalies.

There are regions in satellite images that contain certain cloud formations, with label names: Fish, Flower, Gravel and Sugar. Each image has at least one cloud formation, and can possibly contain up to all four.

The labels for these regions were created in a crowd-sourcing activity at the Max-Planck-Institute for Meteorology in Hamburg, Germany, and the Laboratoire de météorologie Dynamique in Paris, France. A team of 68 scientists identified areas of cloud patterns in each image, and each image was labeled by approximately 3 different scientists. 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 [7]. All images have a native resolution of 2100x1400 pixels. The sample image from UCSID with the overlayed mask is shown in Fig. [1].

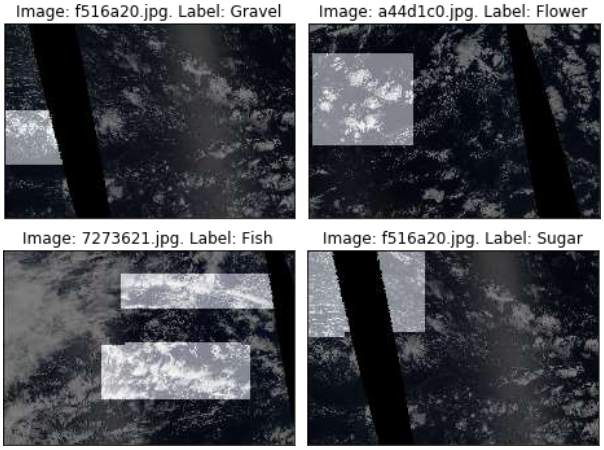


Fig. 1. Example images from UCSID with masks overlayed

Due to the way UCSID was collected, it has a significant amount of noise pixels in cloud masks. As masks are composed of rectangles that overlay cloud completely, there are many pixels, which are labeled as clouds but, in reality, are not. Not all clouds are marked, too. Additionally, because of the concatenation of masks from different annotators, classes can overlap significantly, including cases, where all four classes are assigned to the same pixels.

Most images contain more than one label per sample. The distribution of cloud types per image is shown in Fig. [2].

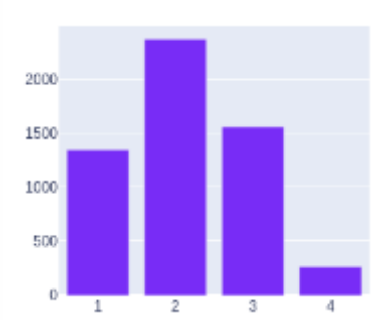


Fig. 2. Number of cloud types per images

We analyze which cloud types are usually found together. The distribution of labels co-occurrence is shown in Fig. [3].

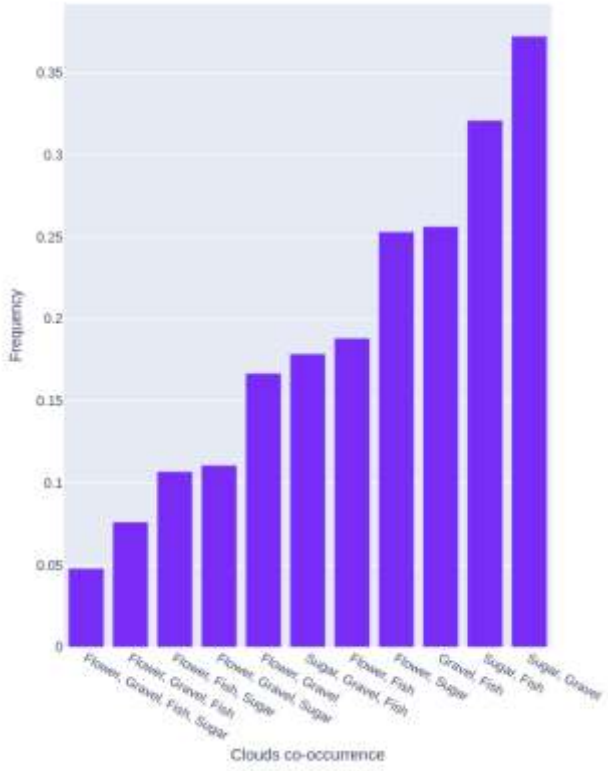


Fig. 3. Frequency diagram of different labels cooccurrence

UCSID is divided into 5546 train, 3698 validation and test images by competition organizers. Additionally, we do not have access to labels of validation and test datasets.

We did no modifications to the dataset distribution (undersampling, oversampling, etc.).

3.2 Evaluation metric

In this research, we used the Dice coefficient, which is the same as the F1-score. 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:

$$D = F_1 = \frac{2 \cdot (X \cap Y)}{(X) + (Y)},$$

where: X is the predicted set of pixels and Y is the ground truth set of pixels.

The Dice coefficient is defined to be 1 when both X and Y are empty. The overall score is the mean of the Dice coefficients for every pair of images and labels in the dataset.

Thus, we take into account the possible difference in relative cloud sizes.

4. Method

4.1 Preprocessing

Model training and validation were performed with preprocessed versions of the original images. As original image resolution is unnecessarily large and evaluation was performed in the resolution of 350x525 pixels, images were resized.

Because UCSID was collected from satellites flying over the same regions, slight contours of sea bed can be seen if photos are averaged. Also, the shape of the black stripe on the photos depends on the satellites' position and region. A possible correlation between place and preferable clouds type could create an unwanted bias during training, so we used aggressive strategies of data augmentation. Also, showing as much variance as possible to the model during training increases its ability to generalize to unseen regions.

4.2 Data augmentation

We used online augmentations, at least one augmentation was applied to the training image before inputting to the CNN. All augmentations were from Albumentations [8] library: coarse dropout; horizontal and vertical flips; shift, scale, rotation; optical distortion; grid distortion, piecewise affine distortion; shift of RGB channels; Gaussian noise; motion blur; median and Gaussian blur; sharpening; embossing; random changes of brightness, contrast, and gamma.

Additionally, we used grid shuffle augmentation. It alleviates the influence of the persistent patterns of the image, simultaneously preserving large enough parts of the clouds to be successfully identified.

4.3 Network architecture

We aim to segment each satellite photograph precisely despite noisy training masks. We build our neural networks using conventional deep CNN, which is built using the encoder-decoder

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

architecture. Unfortunately, training the encoder from scratch is computationally expensive, especially given the small amount of training data and noise in labels. Thus, we use Imagenet-pretrained CNN architectures as initialization for encoder [9].

Here, we propose the multi-task learning approach to cloud segmentation (Fig. 4). Additionally to the decoder, which predicts a segmentation mask, we introduce a separate classifier that uses features from the last layer of the encoder.

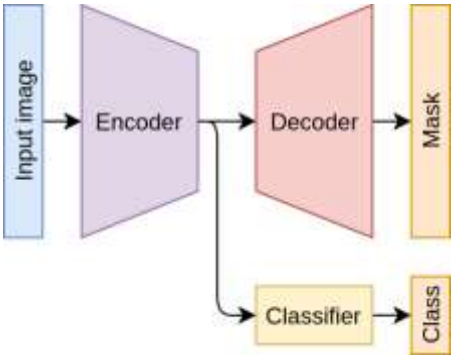


Fig. 4. Auxiliary classifier output during the training phase

The decoder outputs 4 channels, one channel per type of cloud. The classifier also has 4 outputs. The motivation to use separate classifiers lies in the assumption, that labels are less noisy on the image-level. Individually annotated pixels can be noisy but groups of pixels definitely represent the presence of the cloud in the image. The addition of the classifier passes more stable gradients to the encoder, which increases the speed of training and accuracy. The architecture of the classifier is shown in Fig. 5.

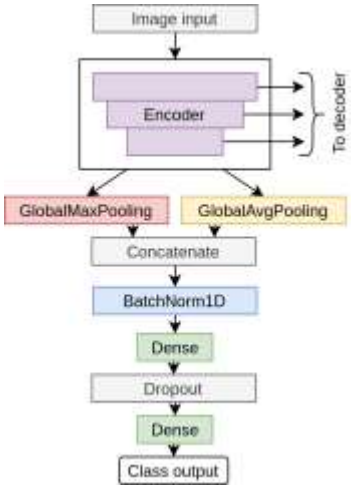


Fig. 5. The architecture of the classifier

For the decoder, we utilize conventional U-Net [10] and Feature Pyramid Network (FPN) [11] architectures. In our experiments, we found, that FPN decoder was consistently better because its predictions are made in ¼ of the image resolution. *This construct reduces the influence of noisy masks, which explains better performance.*

4.4 Training process

We used a single-stage training process and single round pseudo-labeling for several particular models.

4.4.1 Main training

The main training is performed on the 80 % subset of UCSID, while the other 20 % is left as a holdout dataset. The decoder was initialized with random weights [12]. To save pretrained weights in encoder while the decoder is in a random state, we disabled training (froze) of the encoder for five epochs while training decoder and classifier only.

We trained classifier and decoder separately. Joint training of classifier and decoder as described in 4.3 showed worse results. *We observed, that it happened because the classifier was training much faster, than decoder and prevented decoder from training on photos without clouds by stopping the gradients.*

Labels for classifiers were generated from masks. If the ground truth mask was not empty, it was considered a positive label for the classifier and vice versa.

During our experiments, CNNs were trained up to 50 epochs with early stopping [13]. Training stopped automatically in a range from 30 to 40 epochs. We used a combination of novel optimizers: Radam [14], LARS [15] and Lookahead [16]. This combination was called "RangerLars" by deep learning engineers [17]. In this task, it scored consistently better, than the Radam baseline.

We use cosine annealing learning rate schedule to achieve a better any-time performance of our CNNs [18].

4.4.2 Loss functions

To train our models we used different loss functions and their combinations. The basic losses used are:

– Binary Cross-Entropy (BCE) loss which is classically used for tasks of segmentation;

– Focal loss, as we tried to down-weight the contribution of easy examples (non-noisy pixels);

– Soft Dice loss [19], we wanted to use loss which correlates with the target metric;

– The logarithm of the Dice loss for better convergence.

Experimentally we've found that sum reduction of BCE, focal and log dice losses outperforms other losses separately on the validation set. Also, we used trimming for the Focal and BCE loss to smooth out the noise in ground truth labels presented in data. Plots of used losses can be found in Fig. 6.

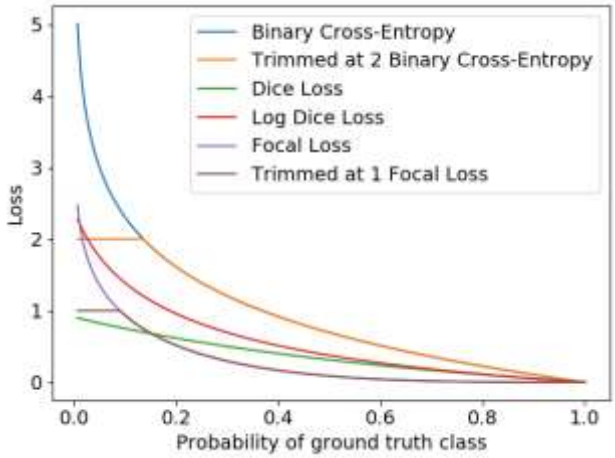


Fig. 6. The plot of different loss functions (best viewed in color)

From this figure it is clearly seen, that trimmed versions of losses penalize network less if its prediction is incorrect, which happens frequently, given the noisy data.

We used Binary Cross-Entropy loss for the classifier, as we found image-level labels to be stable.

During training, we evaluated the performance of the segmentation model using only non-empty masks making the classifier to be responsible for the detection of empty images. Using this technique, we were able to increase the recall of the segmentation model.

At training time, we regularize our models for better robustness. We use conventional methods, e.g., weight decay and dropout.

4.4.2 Pseudo-labeling

We use pseudo-labeling [20] to reduce the effect of noisy labels during the initial stage of training. We used an ensemble of previously trained models to re-label the training set. Then, we used the modified training set to train other CNNs.

While introducing significant confirmation bias [21], this method reduced noise in labels significantly, as now; masks were generated by single labeler with proper procedure. In Fig. 7 we show ground truth and pseudo-labeling masks (boundaries – ground truth, fill – pseudo labels).

As seen from this illustration, pseudo-labels are way smoother and look more consistent with the underlying clouds.

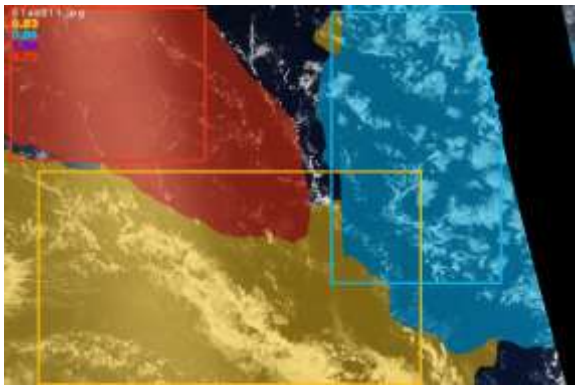


Fig. 7. Example of pseudo-labels on the photo with severe label noise (best viewed in color)

To reduce unwanted bias, we use pseudo-labels only to pretrain CNNs for the first five epochs. It reduced training time significantly and led to better performance on holdout and testing sets.

Additionally, we used the testing set as pseudo-labels, too. To decrease noise in pseudo-labels, we used binarized masks after post-processing. It reduced the influence of overlapping clouds and small noisy masks, generated only by the ensemble itself.

Networks trained this way performed slightly better on holdout set and slightly worse on the final test set than networks that were trained only with ground-truth labels.

4.5 Inference

During inference, we resized testing images to the size, on which models were trained. Predicted masks were resized to the fixed resolution of 350x525 pixels. Also, we apply the sigmoid function to the CNN output to get probabilities.

To obtain the final segmentation mask, we multiply classification output by segmentation mask. This allows us to reduce false-positive results in the post-processing step. The structure of the inference network graph is shown in Fig. 8.

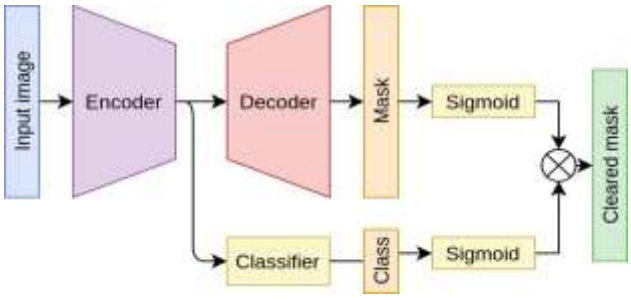


Fig. 8. Inference CNN setup

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

We utilize test-time augmentations, ensembling, and mask post-processing to achieve more stable results.

4.5.1 Test-time augmentations

To reduce the variance of predictions, we utilize test-time augmentations (TTA) [22]: we make predictions on different changed versions of the original images, and then average prediction results. As pictures of clouds can be viewed from any angle and in different time of the day, we utilize the following changes to each original image:

1. Original image;
2. Horizontal flip
3. Vertical flip
4. Multiplicative dimming
5. Multiplicative brightening

Here, we define multiplicative dimming and brightening as a multiplication of image pixel intensity values over a constant factor (0.9 for dimming, and 1.1 for brightening).

4.5.2 Post-processing

Segmentation CNN outputs continuous floating-point numbers for every pixel. To transform them into the binary mask, we threshold them. Using the holdout validation set, we perform a linear search for the threshold that maximizes the Dice coefficient, given fixed predictions of CNN.

Also, as the Dice coefficient is defined to be 1 when prediction and ground truth are empty, even a single false positive pixel makes the Dice coefficient 0. We use minimal size cutoff, by assigning all zeros to the mask, which has area lower than a threshold. To perform a search, we selected areas from 5 % to 20 % of the total area of the segmentation mask.

The example of threshold selection is shown in Fig. 9. We select threshold and cut size, which maximizes the Dice score and lies in the flat region of the plot [28]. We make an assumption, that this type of threshold selection guarantees similar performance on different datasets.

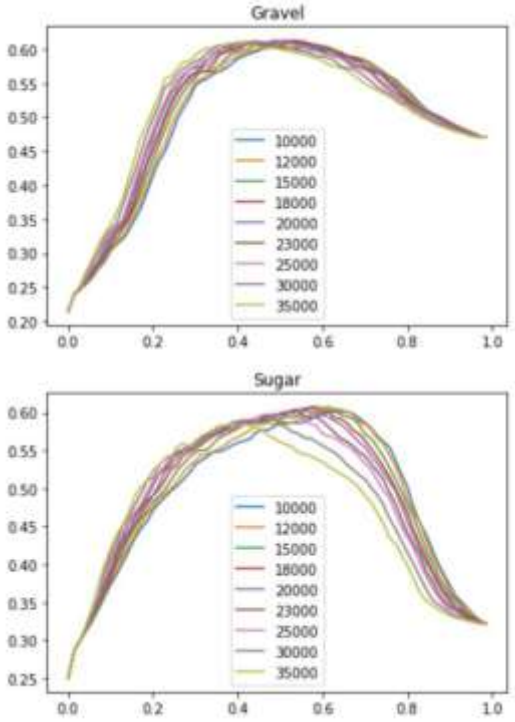


Fig. 9. The dependency of Dice score from the threshold and minimum size cutoff (best viewed in color)

4.5.3 Ensemble

We used single-fold models, with different split, to get a more robust ensemble of them.

In our ensemble, we used models with a diverse set of encoder architectures. Additionally, we selected models that had different distributions of errors relative to images on the holdout set.

Because models were trained with different settings (input shape, interpolation, color scheme, etc.), we obtained prediction masks from every model separately. To obtain final predictions, we average obtained masks w.r.t. to channels and apply post-processing to the result. The post-processed mask is then used to evaluate the ensemble.

Our best ensemble consisted of the following encoder architectures:

1. ResNet-34 [23]
2. ResNet-50
3. EfficientNet-B0 [24]
4. EfficientNet-B1
5. EfficientNet-B2
6. DenseNet-121 [25]
7. DenseNet-169

We used the Catalyst framework [26] based on PyTorch [27] with GPU support. Evaluation of the

whole ensemble was performed on Nvidia V100 GPU in 40 minutes, processing 1.5 seconds per image.

5. Results

Our test stage was split into two parts: local testing and Kaggle testing. As we found locally, the ensembling method is the best one, and we evaluated it on Kaggle validation and test datasets.

On a local dataset of 1110 images, ensembling with TTA performed the best, as well as it performed better on the testing dataset of 3346 images as it has a better ability to generalize on unseen images.

Ensembles scored 0.66732/0.66322 validation/test Dice score for an ensemble without test-set pseudo-labeling and 0.66988/0.65982 with it.

The ensembles showed their stability in the final scoring, keeping consistent rank (60 and 46 of 1538) on validation and testing datasets, respectively.

Conclusions. In this paper, we proposed an end-to-end deep learning-based method for the segmentation of different types of clouds from RGB satellite images with a preprocessing and post-processing pipeline. We have trained the ensemble from several networks on augmented data and made a hyperparameter selection for better model performance.

We have used an ensemble of 7 CNN architectures (ResNet-34, ResNet-50, EfficientNet-B0, EfficientNet-B1, EfficientNet-B2, DenseNet-121,DenseNet-169) and made transfer learning for our final solution.

From the experimental results, the proposed method demonstrates stable results and learns good general features from noisy data. As we observed during the experiments, our model finds types of clouds, which are not annotated on the images but seem to be correctly defined by the model.

Future work can extend this method with the interpretation of the model predictions, as well as for the whole ensemble.

References

1. Rasp S., Schulz H., Bony S. & Stevens B. (2019) “Combining crowd-sourcing and deep learning to explore the mesoscale organization of shallow convection”, eprint arXiv:1906.01906, USA.

2. Zhe Z., Shi Q., Binbin H. & Chengbing D. (2018). “Cloud and Cloud Shadow Detection for Landsat Images: The Fundamental Basis for Analyzing Landsat Time Series”, Remote Sensing Time Series Image Processing, USA.

3. Harb M., Gamba P. & Dell'Acqua F. (2016). “Automatic Delineation of Clouds and Their Shadows in Landsat and CBERS (HRCC) Data”, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* USA. DOI:10.1109/JSTARS.2016.2514274.

4. Hu X., Wang Y. & Shan J. (2015). “Automatic Recognition of Cloud Images by Using Visual Saliency Features”, *IEEE Geoscience and Remote Sensing Letters,* Vol. 12, No. 8, pp. 1760-1764. DOI: 10.1109/LGRS.2015.2424531.

5. Zhang, Jinglin Z., Liu P., Feng Z. & Qianqian S. (2018). “CloudNet: Ground-Based Cloud Classification With Deep Convolutional Neural Network” Geophysical Research Letters, USA. DOI: 10.1029/2018GL077787.

6. (2020). “Understanding Clouds from Satellite Images | Kaggle”. [Digital resource] – Available at: https://www.kaggle.com/c/understanding_cloud_organization. – Active link – 15.03.2020.

7. (2020). “Sugar, Flower Fish, or Gravel?” [Digital resource]/ – Available at: https://www.zooniverse.org/projects/raspstephan/sugar-flower-fish-or-gravel. – Active link – 15.03.2020.

8. (2020). “Albumentations: fast image augmentation library and easy to use wrapper around other libraries”. [Digital resource] – Available at: https://github.com/albumentations-team/albumentations. – Active link – 15.03.2020.

9. Iglovikov V. & Shvets A. (2018). “TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation”, eprint arXiv: 1801.05746, USA.

10. Ronneberger O., Fischer P. & Brox T. (2015). “U-Net: Convolutional Networks for Biomedical Image Segmentation”, eprint arXiv: 1505.04597, USA.

11. Lin T., Dollár P., Girshick R., He K., Hariharan B. & Belongie S. (2016). “Feature Pyramid Networks for Object Detection”, eprint arXiv: 1612.03144, USA.

12. He K., Zhang X., Ren S. & Sun J. (2015). “Delving Deep into Rectifiers: Surpassing Human-

Level Performance on ImageNet Classification", eprint arXiv: 1502.01852, USA.

13. Rich C., Lawrence S. & Giles C. (2000). "Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping", Advances in Neural Information Processing Systems. 13. pp. 402-408, USA.

14. Liu L., Jiang H., He P., Chen W., Liu X., Gao J. & Han J. (2019). "On the Variance of the Adaptive Learning Rate and Beyond", eprint arXiv: 1908.03265, USA.

15. You Y., Gitman I. & Ginsburg B. (2017). "Large Batch Training of Convolutional Networks", eprint arXiv: 1708.03888, USA.

16. Zhang M., Lucas J., Hinton G., Ba J. & Han J. (2019). "Lookahead Optimizer: k steps forward, 1 step back", eprint arXiv: 1907.08610, USA.

17. (2020). "Mgrankin/over9000: Over9000 optimizer" [Digital resource]. – Available at: https://github.com/mgrankin/over9000. – Active link – 15.03.2020.

18. Loshchilov I. & Hutter F. (2016). "SGDR: Stochastic Gradient Descent with Warm Restarts", eprint arXiv: 1608.03983, USA.

19. Jeroen Bertels, Tom Eelbode, Maxim Berman, Dirk Vandermeulen, Frederik Maes, Raf Bisschops & Matthew Blaschko. (2019). "Optimizing the Dice Score and Jaccard Index for Medical Image Segmentation: Theory & Practice", eprints arXiv: 1911.01685, USA.

20. Dong-Hyun L. (2013). "Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks", ICML 2013 Workshop: Challenges in Representation Learning (WREPL), USA.

21. Arazo E., Ortego D., Albert P., O'Connor N. & McGuinness K. (2020). "Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning", eprints arXiv: 1908.02983, USA.

22. Moshkov N., Mathe B., Kertesz-Farkas A., Hollandi R. & Horvath P. (2020). "Test-time augmentation for deep learning-based cell segmentation on microscopy images", bioRxiv 814962, USA, DOI: 10.1101/814962.

23. He K., Zhang X., Ren S. & Sun J. (2015). "Deep Residual Learning for Image Recognition", eprints arXiv: 1512.03385, USA.

24. Huang G., Liu Z., van der Maaten L. & Weinberger K. (2016). "Densely Connected Convolutional Networks", eprints arXiv: 1608.06993, USA.

25. Tan M. & Le Q. (2019). "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", eprints arXiv: 1905.11946, USA.

26. (2018). "Accelerated DL R&D". [Digital resource]. – Available at: https://github.com/catalyst-team/catalyst. – Active link – 15.03.2020.

27. (2017). "PyTorch". [Digital resource] – Available at: https://pytorch.org. – Active link – 15.03.2020.

28. Tymchenko B., Hramatik A., Tulchiy H. & Antoshchuk S. (2019). "Classifying Mixed Patterns of Proteins in Microscopic Images with Deep Neural Networks", Herald of advanced information technology, Vol. 2, No. 1, pp. 29-36. DOI: 10.15276/hait.02.2019.3.

Received 29.01.2020
Received after revision 12.02.2020
Accepted 14.02.2020

УДК 004.93.1

[1]**Тимченко, Борис Ігорович** аспірант, інститут комп'ютерних систем,
E-mail: tymchenko.b.i@opu.ua, ORCID: http://orcid.org/0000-0002-2678-7556
[2]**Марченко, Філіп Олександрович** магістр, факультет математики, фізики та інформаційних технологій, E-mail: p.marchenko@stud.onu.edu.ua, ORCID: http://orcid.org/0000-0001-9995-9454
[3]**Сподарець, Дмитро Володимирович,** E-mail: dmitry.spodarets@vitechlab.com,
ORCID: http://orcid.org/0000-0001-6499-4575
[1]Одеський національний політехнічний університет, просп. Шевченка, 1, м. Одеса, Україна, 65044
[2]Одеський національний університет ім. І. І. Мечникова, вул. Дворянська, 2, м. Одеса, Україна, 65014
[3]VITech Lab, вул. Рішельєвська, 33, м. Одеса, Україна, 65000

СЕГМЕНТАЦІЯ ПАТЕРНІВ ОРГАНІЗАЦІЇ ХМАР НА СУПУТНИКОВИХ ЗОБРАЖЕННЯХ З ВИКОРИСТАННЯМ ГЛИБОКИХ НЕЙРОННИХ МЕРЕЖ

Анотація. *Зміна клімату є однією з найважливіших проблем, що стоять зараз перед людством. Важливою частиною кліматичних моделей є рух хмар, який різко впливає на параметри клімату. Невисокі хмари грають величезну роль у визначенні клімату Землі. Їхній рух також складно зрозуміти і представити в кліматичних моделях. На жаль, точне моделювання руху хмар є складною задачею та вимагає досконалого знання основних фізичних процесів і початкових станів. Межі між різними типами хмар зазвичай розмиті і їх важко визначити за допомогою систем прийняття рішень на основі правил. Спрощення етапу сегментації має вирішальне значення і може допомогти дослідникам в розробці більш досконалих кліматичних моделей. Конволюційні нейронні мережі були успішно застосовані в багатьох аналогічних областях, а також для самої сегментації хмар. Проте, існує проблема високої вартості хороших наборів даних з маркуванням на рівні пікселів, тому в галузі часто використовуються грубі набори даних з мітками рівня зображення. У цій статті ми пропонуємо комплексний метод на основі глибокого навчання для класифікації та сегментації хмар різних типів з одного кольорового супутникового зображення. Тут ми пропонуємо багатозадачний підхід до сегментації хмар. На додаток до моделі сегментації, ми вводимо окремий класифікатор, який використовує ознаки з середнього рівня моделі сегментації. Представлений метод може використовувати грубі, нерівні і такі, що перекриваються маски для хмар. Виходячи з результатів експериментів, запропонований метод демонструє стабільні результати і вивчає хороші загальні характеристики з зашумлених даних. Як ми спостерігали під час експериментів, наша модель правильно знаходить хмари, які не анотовані на зображеннях. Запропонований метод входить в топ 3 % серед конкуруючих методів на наборі даних Understanding Clouds from Satellite Images.*

Keywords: *глибоке навчання; супутникова зйомка; глибокі конволюційні нейронні мережі; багатоцільове навчання; класифікація хмарних утворень; Kaggle; метеорологія*

УДК 004.93.1

[1]**Тимченко, Борис Игоревич** аспирант, институт компьютерных систем,
E-mail: tymchenko.b.i@opu.ua, ORCID: http://orcid.org/0000-0002-2678-7556
[2]**Марченко, Филипп Александрович** магистр, факультет математики, физики и информационных технологий, E-mail: p.marchenko@stud.onu.edu.ua, ORCID: http://orcid.org/0000-0001-9995-9454
[3]**Сподарец, Дмитрий Владимирович,** E-mail: dmitry.spodarets@vitechlab.com,
ORCID: http://orcid.org/0000-0001-6499-4575
[1]Одеський национальный политехнический университет, просп. Шевченко, 1, г. Одесса, Украина, 65044
[2]Одеський национальный университет им. И. И. Мечникова, ул. Дворянская, 2, г. Одесса, Украина, 65014
[3]VITech Lab, ул. Ришельевська, 33, г. Одесса, Украина, 65000

СЕГМЕНТАЦИЯ ПАТТЕРНОВ ОРГАНИЗАЦИИ ОБЛАКОВ НА СПУТНИКОВЫХ СНИМКАХ С ИСПОЛЬЗОВАНИЕМ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ

Аннотация. *Изменение климата является одной из важнейших проблем, стоящих сейчас перед человечеством. Важной частью климатических моделей является движение облаков, который сильно влияет на параметры климата. Низкие облака играют огромную роль в определения климата Земли. Их движение сложно понять и представить в климатических моделях. К сожалению, точное моделирование движения облаков является сложной задачей и требует точного знания основных физических процессов и начальных состояний. Границы между разными типами облаков, обычно размыты, их трудно определить с помощью систем принятия решений на основе правил. Упрощение этапа сегментации имеет решающее значение и может помочь исследователям в разработке более совершенных климатических моделей. Сверхточные нейронные сети успешно применяются во многих аналогичных областях, а также для самой сегментации облаков. Однако существует проблема высокой стоимости хороших наборов данных с маркировкой на уровне пикселей, поэтому в отрасли часто используют грубые наборы данных с разметкой уровня изображения. В этой статье мы предлагаем комплексный метод на основе глубокого обучения для классификации и сегментации облаков разных типов с одного цветного спутникового изображения. Здесь мы предлагаем многозадачный подход к сегментации облаков. В дополнение к модели сегментации, мы вводим отдельный классификатор, который использует признаки с среднего уровня модели сегментации. Представленный метод может использовать грубые, неровные и пересекающиеся метки для облаков. Исходя из результатов экспериментов, предложенный метод демонстрирует стабильные результаты и изучает,*

извлекает общие характеристики с зашумленных данных. Как мы наблюдали во время экспериментов, наша модель правильно находит облака, которые НЕ аннотированные на изображениях. Предложенный метод входит в топ 3 % среди конкурирующих методов на наборе данных Understanding Clouds from Satellite Images.

***Ключевые слова**: глубокое обучение; спутниковая съемка; глубокие сверхточные нейронные сети; многоцзадачное обучение; классификация облачных образований; Kaggle; метеорология*



Tymchenko, Borys Igorovych Ph. D, Senior Computer Vision Engineer
Research field: computer vision, deep learning, deep convolutional neural networks



Marchenko, Philip Oleksandrovych Master student, Machine Learning Engineer
Research field: machine learning, computer vision, optimization methods, deep learning



Spodarets, Dmytro Volodymyrovych
Master of Science, CTO
Research field: software infrastructure, machine learning, computer vision