Modification of U-Net neural network

for multichannel satellite images segmentation

**Abstract – Results of training of convolutional neural network for satellite four-channel image segmentation are performed. Input images contain blue, green, red and near-infrared channels. The algorithm was trained to detect areas which contain buildings and structures. Modification of the U-Net with two encoders was used. The values of Sorensen coefficient and Jaccard index were calculated for 16 different urban regions.**

***Keywords: image segmentation, satellite images, convolutional neural network, U-Net***

Introduction

Nowadays the problem of object detection on satellite high-resolution images is very relevant. Automatic image segmentation allows to extract areas of interest quickly and accurately, for example, buildings, agricultural fields, quarries, etc. Most approaches of solving this problem suggest the development of deep learning algorithms [1].

This paper presents developed convolutional neural network (CNN). Such networks are capable for real-time detecting and classifying objects.

Automatic segmentation is an important part for the preliminary processing of images. Today, convolutional neural networks are widely used for this task. In particular, U-Net neural network has shown its effectiveness for medical image segmentation [2]. The mathematical structure of CNN is parallel, so graphics processing units (GPU) are ideal instruments to work with CNN [3].

There are some requirements for satellite images segmentation [4-8]:

* Size and type of buildings and structures may significantly vary from cottages to huge apartment buildings. The algorithm should detect objects of any size very well. The usage of multiple encoders in neural networks could solve this problem.
* The separation of objects with a high density of location. This is achieved by means of careful selection of loss function.
* Trained model should be invariant to rotations and color distortions. This problem can solved by data augmentation.
* Aerial images have different spatial resolution. It’s important to create algorithm which has an ability to generalize perfectly.

This article presents the results of training of developed neural network for satellite multispectral image segmentation in order to detect buildings and structures. Modification of the U-Net architecture with the second encoder is proposed. The research continues works [9-12].

This paper consists of 6 parts. In the first part, a reader gets acquainted with the formulation of problem. The second part of article describes image datasets. Architecture of the neural network is shown in the third part. The fourth part presents results of numerical experiments. And finally, in the conclusion there is summarized the research and supposed future plans.

Databases of satellite images

Neural network was pre-trained on images of Spacenet database. SpaceNet [11] includes satellite images of 6 large urban agglomerations: Rio de Janeiro (Brazil), Las Vegas (USA), Paris (France), Shanghai (China), Khartoum (Sudan) and Atalanta (USA). Eight-channel photos are shooted by WorldView-2 and WorldView-3 satellites with a different spatial resolution. The database is divided into subsets, depending on the type of tagged objects. For instance, the SpaceNet database contains two subsets of satellite images, which cover areas of 3011 km² and 5555 km², for building detection. Examples of images from these subsets are shown in Fig. 1.



Fig. 1. Examples of images from the Spacenet database

Our model was trained on images of 16 different regions. Every image has an appropriate binary mask marked by experts. The dataset covers about 30 km². The images of this database contain blue, green, red and near infrared (NIR) channels with a spatial resolution of 3 m/pixel.

The developed CNN requires input images of 256×256 pixels. So before the launch of deep learning algorithms each image and the appropriate mask of dataset have been cut on two non-intersecting stripes. Each stripe was divided on patches of 256×256 pixels with pixel step of 128 pixels, so patches intersected by half. To increase the training set, there were applied three stages of data augmentation:

1. Rotations on 90, 180, 270 degrees and reflections. The training set has increased 8 times after this stage.
2. Chromatic distortions. Images were translated from RGB color space to HSV color space. Random values were added to HSV coordinates of images. For the NIR channel, instead of chromatic distortions, random values from [-0.06, +0.06] interval were added. (NIR values were normalized in [0, 1]).
3. Random shifts, scales and small degree rotations of patches.

Neural network architecture

In research there was used a modification of the well-known CNN architecture: U-Net [2]. Original U-Net consists of 2 parts: encoder and decoder.

The encoder part is a convolutional neural network of five blocks. Each of these blocks consists of two convolutional layers with 3 × 3 filters, with ReLU activation function and batch normalization applied at each of them, and also of downsampling layer with 2×2 maxpooling. Decoder has the same number of blocks as encoder. The decoder contains the same number of blocks. Each decoder block consists of an upsampling operation with 2×2 filter combining with a corresponding map of features from the encoder, two convolutional layers with 3×3 filter, ReLU activation function and batch normalization applied to each of them. The last layer of the network performs a convolution operation with 1×1 filter, which relates every pixel to a specific class.



Fig. 2. U-Net architecture with 2 encoders

Original U-Net was modified. There were used two encoders for RGB and NIR channels. The outputs of encoders were concatenated before being linked with appropriate layers (fig. 2). As a result, the neural network has 47 convolutional layers, 47 ReLU activation functions, 47 operations of batch normalization, 1 sigmoid activation function, 10 maxpooling operations, 5 upsampling operations and 11 merge operations.

The neural network was created by means of Keras library with Tensorflow framework as backend. Keras is an open source library written in Python. [14].

Numerical results

Developed CNN was launched on NVIDIA DGX-1 supercomputer. As the algorithm of numerical optimization Adam optimizer with a learning rate of 1e-3 was chosen [15]. Lovász-Softmax loss was chosen as the loss function. On each training iteration, the model updated its weights after running through the network of a batch of 16 samples. The training finishes after completing 100 epochs.

The quality of the segmentattion algorithms was evaluated by Sorensen-Dice coefficient (dice) and Jaccard index (IoU).

The Sorensen coefficient takes values from [0, 1] and shows the degree of similarity between two sets. The Sorensen coefficient is calculated by means of following formula:

,

where is a power of intersection and is a sum of powers for expert markup and predicted masks.

For our task, the numerator and denominator can be calculated by following formula:

,

,

where are values of pixels from [0,1] for real mask and predictions respectively.

IoU determines the degree of similarity of two compared objects. The index is determined by the formula:

,

where and are classes of expert markup and predictions respectively.

IoU takes values from [0,1]. If the index of this coefficient tends to 1, then objects are identical, whereas the value of 0 means that objects are not equal.

The comparison of training results for original U-Net with four‑channel images as inputs and modification with two encoders for RGB and NIR channels is shown on fig. 3.



Fig.3. Values of Sorensen coefficients for original U-Net and modified U-Net for images of 16 regions

During preliminary training on Spacenet dataset, Sorensen coefficient reached the value of 0.84 and Jaccard index increased to 0.77.

The values of Sørensen coefficients and Jaccard indices for all 16 regions for the algorithm with pre-training and without this process are shown in Fig. 4.



Fig. 4. Values of Sorensen coefficients and Jaccard indices of pre-trained model and the algorithm without pre-training

Examples of input images and the result of segmentaion are shown on Fig. 5.



Fig. 5. Examples of input images and results of segmentation for developed algorithm.

Conclusion

This paper describes the training process of developed convolutional neural network intended for the segmentation of satellite images. U-Net architecture with two encoders was proposed to work with four-channel images. The algorithm was pre-trained on Spacenet dataset. The value of Sorensen coefficient was equal to 0.78 and Jaccard index was equal to 0.67.

Other directions of research may include:

* Categorization of buildings based on area size and encoders for separate classes.
* Usage of object boundaries as the third class for detection.

Developed algorithm may be put into practice for assessing the level of urbanization of various regions and tracking the construction of large objects.

ACKNOWLEDGMENT

The article was prepared with the financial support of the Ministry of Education of the Russian Federation as part of the research project No. 14.575.21.0167 connected with the implementation of applied scientific research on the following topic: «Development of applied solutions for processing and integration of large volumes of diverse operational, retrospective and the thematic data of Earth's remote sensing in the unified geospace using smart digital technologies and artificial intelligence» (identifier RFMEFI57517X0167). The authors are also grateful to the AI-centre of P.G. Demidov Yaroslavl State University for providing an access to NVIDIA DGX-1 supercomputer.

REFERENCES

1. Y. Goodfellow, Y. Bengio, A. Courville. *Deep Learning*. The MIT Press, 2016, 800 p.
2. O. Ronneberger, P. Fischer, T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, LNCS, vol. 9351, 2015, pp. 234–241.
3. S. Seferbekov, V. Iglovikov, A. Buslaev, A. Shvets. Feature Pyramid Network for Multi-Class Land Segmentation. Web: https://arxiv.org/pdf/1806.03510.pdf.
4. Zhang, Z., Liu, Q., & Wang, Y. (2018). “Road extraction by deep residual u-net”. IEEE Geoscience and Remote Sensing Letters, 15(5), pp. 749-753.
5. Zhang, Liangpei, Lefei Zhang, and Bo Du. “Deep learning for remote sensing data: A technical tutorial on the state of the art.” IEEE Geoscience and Remote Sensing Magazine 4.2 (2016): 22-40.
6. Y. Chen, H. Jiang, C. Li, X. Jia and P. Ghamisi, “Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks,” in IEEE Transactions on Geoscience and Remote Sensing, vol. 54, no. 10, pp. 6232-6251, Oct. 2016.
7. X. X. Zhu et al., “Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources,” in IEEE Geoscience and Remote Sensing Magazine, vol. 5, no. 4, pp. 8-36, Dec. 2017.
8. Iglovikov, V., Mushinskiy, S., & Osin, V. (2017). “Satellite imagery feature detection using deep convolutional neural network: A kaggle competition”. arXiv preprint arXiv:1706.06169.
9. V. V. Khryashchev, V. A. Pavlov, A. Priorov and A. A. Ostrovskaya, "Deep Learning for Region Detection in High-Resolution Aerial Images," 2018 IEEE East-West Design & Test Symposium (EWDTS), Kazan, 2018, pp. 1-5.
10. V.Khryashchev, L.Ivanovsky, V.Pavlov, A.Ostrovskaya, A.Rubtsov Comparison of Different Convolutional Neural Network Architectures for Satellite Image Segmentation, Proceedings of the FRUCT’23, Bologna, Italy, 13–16 November 2018. pp.172–179.
11. V. V. Khryashchev, A. A. Ostrovskaya, V. A. Pavlov and A. S. Semenov, "Optimization Of Convolutional Neural Network For Object Recognition On Satellite Images," Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO), Minsk, 2018, pp. 1-5.
12. L. Ivanovsky, V. Khryashchev, V. Pavlov and A. Ostrovskaya, "Building Detection on Aerial Images Using U-NET Neural Networks," 2019 24th Conference of Open Innovations Association (FRUCT), Moscow, Russia, 2019, pp. 116-122.
13. SpaceNet Database, Web: http://explore.digitalglobe.com/spacenet.
14. A. Gulli, S. Pal., *Deep Learning with Keras*, Packt Publishing, 2017, 320 p.
15. D. P. Kingma, J. Ba, “Adam: A Method for Stochastic Optimization”, Web: https://arxiv.org/abs/1412.6980.