**Multichannel Satellite Image Segmentation for Automated Building Detection using Deep Learning**

Vladimir Khryashchev, Roman Larionov, Leonid Ivanovsky

P.G. Demidov Yaroslavl State University

[v.khryashchev@uniyar.ac.ru](mailto:v.khryashchev@uniyar.ac.ru), [r.larionov@uniyar.ac.ru](mailto:r.larionov@uniyar.ac.ru), [leon.ivanovsky@yahoo.com](mailto:leon.ivanovsky@yahoo.com)

**ABSTRACT**

The goal of our research was to develop a deep learning algorithm for automated building detection on four-channel satellite images. Modified U-Net neural network can be used to detect needed objects efficiently. The model was implemented by means of Keras library and launched on modern GPUs of high-performance supercomputer NVIDIA DGX-1. Before the testing of developed algorithm on Planet dataset, modified U-Net had been pre-trained on SpaceNet database. Futhermore, in this article the process of data augmentation was also described. The problem of building detection on high-resolution aerial photos can be put into practice for urban planning, building control of some municipal objects, search of the best locations for future outlets etc.

Keywords: deep learning, computer vision, multichannel satellite image segmentation, building detection.

**1. INTRODUCTION**

Today the problem of object detection on satellite images is in the focus of researchers. Automatic image segmentation allows to extract areas of interest such as vehicles or buildings. Most approaches of solving this problem suggest the development of deep learning algorithms.

In machine learning the problem of image segmentation is usually reformulated as a classification of each pixel. The simplest and the slowest way of solving this problem is a manual segmentation of images by experts. However, it is a time-consuming process, which is subjected to human errors. Automatic segmentation makes possible to process images immediately after receiving it. Satellite images segmentation finds its application in urban planning, building control, forest management and meteorology.

This article presents developed convolutional neural network (CNN). CNNs can detect and classify objects in real time while being computationally less expensive and superior in performance compared with other machine learning methods [1]. Essentially, the mathematical structure of CNNs is parallel and perfectly fits the architecture of graphics processing units (GPUs) which consists of thousands of cores to perform several tasks simultaneously. The features in CNNs are formed automatically in the process of training.

The problem of satellite images segmentation is challenging. In recent years there were developed some CNNs, which aim at

satellite images segmentation. For instance, fully convolutional networks (FCNs) show perfect results of object detection on the dataset of PASCAL VOC 2011 challenge [2]. The basic idea of FCNs is the usage of fully connected layers with a convolution layer in the end of segmentation, while other layers extract necessary features from input data [3].

Feature pyramid neural network (FPN) is also successful algorithm of image segmentation. FPNs are based on pyramid architecture of CNNs. These models show acceptable results of object detection on satellite images [4]. For example, FPN allows to get the value of Jaccard index is approximately equal to 0.49 for satellite images from DeepGlobe [5].

In paper [6] there is presented U-Net architecture – a specific type of FPN, which has shown its effectiveness in medical image segmentation. Later this model was applied to pixel-wise classification of satellite images [7]. The main advantage of this architecture is that algorithm can show good results even with a small training datasets. U-Net uses skip-connections to combine low-level and higher-level maps of features.

This article consists of six parts. The first part is devoted to CNNs as an approach in machine learning and peculiarities of image segmentation. It also contains an overview of some papers for object detection on aerial photos. The second part is devoted to the available databases of satellite images. The third section describes the process of data augmentation before the training of deep learning algorithm has been implemented. Developed architectures of CNNs for building detection on aerial photos and some peculiarities of training of models were considered in the fourth part of this article. The fifth part presents the results of numerical experiments for the developed model. In the conclusion there is summarized the research. And finally, the last section represents references.

**2. DATABASES OF SATALLITE IMAGES**

The SpaceNet database [8] includes 11-bit satellite images of 6 large urban agglomerations: Rio de Janeiro, Las Vegas, Paris, Shanghai, Khartoum and Atalanta. Eight-channel photos are shooted by WorldView-2 and WorldView-3 satellite sensors with a different spatial resolution. The database is divided into subsets, depending on the type of tagged objects. For instance, it contains two subsets of satellite images of 650 × 650 size, which cover areas of 3011 km² and 5555 km², for the task of building detection.

The GeoEye-1 database [9] also includes aerial four-channel (blue, green, red, near-infrared) photos. GeoEye-1 sensor was successfully launched on September 6, 2008 from Vanderberg Air Force Base in the USA. The satellite is capable of acquiring image data at 1.84 m multispectral resolution. Satellite images from GeoEye-1 are used for environmental monitoring, mining, engineering, archaeology and agriculture.

The Pleiades-1B database [10] contains four-channel (blue, green, red, near-infrared) images from Pleiades-1B satellite. This sensor was successfully launched on the 2th of December in 2012. The Pleiades-1B dataset is notable for different angles of shooting. Each image of this database has a spatial resolution of 0.5 m / pixel. Aerial photos from GeoEye-1 are used for engineering and construction projects, monitoring of mining and industrial complexes, natural hazards and rescue operations.

In our research, for training and testing of developed deep learning algorithm there were used 10-bit four-channel satellite images and corresponding binary masks of 18 Russian regions from Planet database. Each of 35 aerial photos of this dataset has a spatial resolution of 0.5 m/pixel and covers areas of 1 km². The angle of deviation from the nadir of satellite image does not exceed 30 degrees. Some aerial photos from Planet database have clouds, but they did not cover more than 70% of its square.

**3. DATA PREPARATION**

Numerical experiments for developed deep learning algorithm were performed on normalized satellite images of the Planet database. Satellite images segmentation concerns the usage of parts of aerial photos, which are fed to the input of CNN, so before the training of CNN each high-resolution photo and mask of dataset have been sliced on parts of 256 × 256 size with the step of 128 by data windowing. The intersection of patches allows to cope with problem of artifacts that occur at the junction of image fragments. As a result, training and test sets of 1457 and 393 images and corresponding masks were formed. Examples of sliced images and masks are shown in Fig. 1. Every little part of sliced images corresponded to the needed small part of big generated mask.



**Figure 1. Examples of patches and corresponding masks.**

Generated training and test sets were enlarged using the following techniques:

* Rotations on 90˚, 180˚ and 270˚ and mirroring of patches. As a result, training and test sets were increased 8 times;
* Applying chromatic distortion;
* Image shifts within 2% of image size, scaling on a coefficient from [1; 1,2] and rotations on small angles from [-15˚, + 15˚].

Since some buildings on aerial photos can be located very close to each other, they can be segmented as one object. In order to prevent this, auxiliary masks were formed [11]. If the border between buildings on satellite images of Planet database did not exceed three pixels, this boundary was marked. Examples of masks with red-marked narrow distances between buildings are shown in Fig. 2.



**Figure 2. Examples of masks with tagged narrow distances between buildings.**

However, extended training and test sets still had insufficient data to train the network from scratch. Therefore these sets were used only for tuning the developed CNN. Before the training of model on the images Planet database, it had been pre-trained on the more extensive SpaceNet dataset. In our research, there were used only 8-channel normalized satellite images of 4 regions: Las Vegas, Paris, Shanghai and Khartoum. These aerial photos contain the following channels: coastal blue, blue, green, yellow, red, red edge and 2 near-infrared (NIR1 and NIR2). In order to use data of the same dimension as photos from the Planet database, satellite images were converted to 4-channel by saving only red, green, blue and NIR1 channels.

**4. DEEP LEARNING ALGORITHM**

In this section there is described the architecture of developed CNN, which is used for building detection on high-resolution aerial photos, and some peculiarities of its training. Our work continues research, which was presented in [12, 13]

**4.1. Convolutional neural network**

The network, which was used for satellite images segmentation in our research, is based on very popular and widespread architecture called U-Net. This CNN was originally developed for segmentation of medical images. Its classical structure is described in [6].

U-Net is an U-shaped CNN which consists of two parts: the encoder and the decoder. Both parts have six blocks. The encoder represents typical downsampling path of CNN. Each block of encoder consists of 3 convolutional layers with 3 × 3 filter, 3 rectified linear unit (ReLU) activation functions applied to each of them respectively, 3 layers of batch normalization and a maxpooling operation with 2 × 2 filter and step 2. The decoder represents upsampling path which is used for restoration of segmentation mask. Each decoder’s block includes an upsampling operation with 2 × 2 filter combining with a corresponding map of features from the encoder, 3 ReLU activation functions applied to each of them respectively and 3 layers of batch normalization. The last layer of the network is a convolutional K-channel layer, where K is the number of classes and its output is computed by sigmoid function. In our task, K is equal 2 (“building” and “non-building classes”).

Since our images contain four channels, U-Net was modified by adding the second encoder for data from NIR channel. In addition, feature maps from each block of the second encoder were concatenated with corresponding features of the decoder. The architecture of this model is shown in Fig 3.



**Figure 3. U-Net with two encoders.**

To use information about boundaries between buildings there was embedded an auxiliary decoder of the same structure and an output, which used weights of the main CNN, which had been trained on the images of SpaceNet database. The auxiliary decoder was tuned on created masks of boundaries [14]. On the test stage pixels which belong to predicted boundaries were reset on a segmentation mask.

**4.2. Peculiarities of training**

The approach based on the usage of CNNs requires considerable computational resources. Therefore, training and test stages were implemented on a large number of independent streams of GPU using the parallel computing technology NVIDIA CUDA. This cross-platform technology is supported by all modern NVIDIA graphics cards [15].

Keras library with Tensorflow framework as a backend was used for development of CNN. Keras is an open-source library written in Python. It is built on Tensorflow framework and contains various implementations of commonly used neural network building blocks, such as layers, activation functions and optimizers, and ready tools to pre-process images and text data. In other words, Keras offers a higher-level, more intuitive set of abstractions to develop deep learning models [16]. Moreover, this library allows to train developed models on GPU.

As an algorithm of numerical optimization Adam optimizer with learning rate of 1e-3 was chosen. This optimizer combines the best approaches from gradient descent and momentum optimizers and shows optimal and quick convergence for most tasks of machine learning [17].

**5. NUMERICAL RESULTS**

Modified U-Net was launched on NVIDIA DGX-1 supercomputer, which was provided by Artificial Intelligence Center of P.G Demidov Yaroslavl State University.

As a rule, the quality of algorithms for image segmentation is evaluated by special coefficients for comparing the similarity of predicted and true masks. To estimate developed models there was used Sorensen-Dice coefficient (DSC). This index is binary measure of similarity, possesses the value from [0, 1] and can be calculated by the following formula:

(1)

where is a power of intersection and is a sum of powers for real mask and predictions . In our task, numerator and denominator can be calculated by following formulae

, (2)

, (3)

where are values of pixels from for real masks and predictions respectively. Also to measure the quality of developed deep learning algorithm there were used precision (), recall () and F-score ().

As a loss function there was used a sum

and are binary cross-entropy and the value of Dice loss respectively, which are calculated by following formulae:

(4)

. (5)

Combinations of various loss functions for the learning of machine learning algorithms help to get higher quality of segmentation for the most modern tasks and data competitions [18]. The training and tuning on each dataset finishes after completing 100 epochs, while on each training step of epoch a batch of 16 samples was passed through the developed model. Test results of developed model were presented in Table 1, 2, 3.

**Table 1. Test results on the SpaceNet database**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Regions** | **Metrics** | | | |
| **DSC** | **P** | **R** |  |
| Khartoum | 0,814 | 0,441 | 0,505 | 0,471 |
| Paris | 0,821 | 0,629 | 0,646 | 0,637 |
| Shanghai | 0,790 | 0,510 | 0,556 | 0,532 |
| Vegas | 0,896 | 0,795 | 0,825 | 0,810 |
| Average values | **0,830** | **0,753** | **0,777** | **0,594** |

**Table 2. Test results of tuning on the Planet database without boundary detection**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Regions** | **Metrics** | | | |
| **DSC** | **P** | **R** |  |
| Region01 | 0,706 | 0,363 | 0,494 | 0,418 |
| Region02 | 0,880 | 0,526 | 0,673 | 0,591 |
| Region03 | 0,571 | 0,303 | 0,334 | 0,318 |
| Region04 | 0,787 | 0,426 | 0,606 | 0,500 |
| Region06 | 0,610 | 0,435 | 0,414 | 0,424 |
| Region07 | 0,703 | 0,464 | 0,459 | 0,46 |
| Region08 | 0,719 | 0,645 | 0,614 | 0,626 |
| Region09 | 0,778 | 0,465 | 0,208 | 0,287 |
| Region10 | 0,992 | 0,000 | 0,000 | 0,000 |
| Region11 | 0,599 | 0,322 | 0,334 | 0,328 |
| Region12 | 0,756 | 0,347 | 0,368 | 0,358 |
| Region13 | 0,883 | 0,412 | 0,489 | 0,447 |
| Region14 | 0,907 | 0,750 | 0,789 | 0,769 |
| Region15 | 0,805 | 0,649 | 0,536 | 0,587 |
| Region16 | 0,712 | 0,589 | 0,607 | 0,598 |
| Region17 | 0,686 | 0,366 | 0,560 | 0,443 |
| Region18 | 0,724 | 0,337 | 0,548 | 0,418 |
| Average values | **0,754** | **0,435** | **0,472** | **0,446** |

**Table 3. Test results on the Planet database with boundary detection**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Regions** | **Metrics** | | | |
| **DSC** | **P** | **R** |  |
| Region01 | 0,743 | 0,483 | 0,516 | 0,499 |
| Region02 | 0,884 | 0,631 | 0,676 | 0,653 |
| Region03 | 0,615 | 0,412 | 0,373 | 0,392 |
| Region04 | 0,828 | 0,573 | 0,652 | 0,610 |
| Region06 | 0,647 | 0,497 | 0,479 | 0,488 |
| Region07 | 0,739 | 0,489 | 0,510 | 0,499 |
| Region08 | 0,708 | 0,649 | 0,567 | 0,605 |
| Region09 | 0,784 | 0,393 | 0,270 | 0,321 |
| Region10 | 0,960 | 0,000 | 0,000 | 0,000 |
| Region11 | 0,622 | 0,361 | 0,331 | 0,345 |
| Region12 | 0,711 | 0,391 | 0,353 | 0,371 |
| Region13 | 0,877 | 0,500 | 0,525 | 0,512 |
| Region14 | 0,910 | 0,796 | 0,816 | 0,806 |
| Region15 | 0,804 | 0,630 | 0,541 | 0,582 |
| Region16 | 0,759 | 0,593 | 0,605 | 0,599 |
| Region17 | 0,706 | 0,469 | 0,567 | 0,513 |
| Region18 | 0,759 | 0,486 | 0,560 | 0,521 |
| Average values | **0,768** | **0,491** | **0,491** | **0,489** |

According to results presented in Table 2, minimal and average value of DSC for modified U-Net without boundary detection is approximately equal to 0,6 and 0,75 respectively. In addition, the second encoder for boundary detection allows to increase these values on 0.02. Thus developed deep learning algorithm for building detection on satellite images shows acceptable results. Examples of input images, masks with tagged buildings and narrow distances between them and results of deep learning algorithm are shown in Fig. 4.

    

    

**a) b) c) d) e)**

Figure 4.: Test results a) input images, b) masks with tagged buildings and narrow distances between them, c) results of segmentation excluding boundary detection, d) narrow boundaries segmentation, e) final results of segmentation.

**6. CONCLUSION**

The article shows how modified U-Net with boundary detection for objects of interest can be effectively used for the task of building detection on high-resolution aerial photos. The developed algorithm was pre-trained on the SpaceNet dataset and tuned on the Planet database. The training and test sets were collected and enlarged using various methods of data augmentation. Using the special metrics of similarity between expert markup and predicted masks there was shown that modified U-Net got acceptable results: the average value of Sorensen-Dice coefficient (DSC) was approximately equal to 0.77. For learning of model there was used supercomputer NVIDIA DGX-1.

**7. ACKNOWLEDGMENTS**

The authors are also grateful to the AI-center of P.G. Demidov Yaroslavl State University for providing an access to NVIDIA DGX-1 supercomputer.

**REFERENCES**

[1] Shanmugamani, R. 2018. Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras. Packt Publishing.

[2] PASCAL VOC 2011 challenge. Web: http://host.robots.ox.ac.uk/pascal/VOC/voc2011/index.html.

[3] Shelhamer, E., Long, J., Darrell, T. Fully Convolutional Networks for Semantic Segmentation. Web: https://arxiv.org/pdf/1605.06211.pdf.

[4] Seferbekov, S., Iglovikov, V., Buslaev, A., Shvets, A. Feature Pyramid Network for Multi-Class Land Segmentation. Web: https://arxiv.org/pdf/1806.03510.pdf.

[5] DeepGlobe. CVPR 2018 – Satellite Challenge, Web: http://deepglobe.org.

[6] Ronneberger, O., Fischer, P., Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, LNCS, vol. 9351, 234–341.

[7] Khryashchev, V., Ivanovsky, L., Pavlov ,V., Ostrovskaya, A. and Rubtsov, A. 2018. Comparison of Different Convolutional Neural Network Architectures for Satellite Image Segmentation. In Proceedings of the 23rd Conference of Open Innovations Association FRUCT’23, Bologna, Italy. 172-179.

[8] SpaceNet Database, Web: http://explore.digitalglobe.com/spacenet.

[9] GeoEye-1 Satellite Images. Web: https://www.satimagingcorp.com/gallery/geoeye-1/.

[10] Pleiades-1B Satellite Sensor. Web: https://www.satimagingcorp.com/satellite-sensors/pleiades-1b/.

[11] Ohleyer, S. Building segmentation on satellite images. Web: https://project.inria.fr/aerialimagelabeling/files/2018/01/fp\_ohleyer\_compressed.pdf

[12] Ivanovsky, L., Khryashchev, V., Pavlov, V., Ostrovskaya A. 2019. Building Detection on Aerial Images Using U-NET Neural Networks. In Proceedings of the 24rd Conference of Open Innovations Association FRUCT’24, Moscow, Russia.

[13] Khryashchev, V., Ivanovsky, L., Ostrovskaya A., Semenov, A. 2019. Application of Satellite Image Segmentation for Urban Planning Optimization. In Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering, Hong Kong. 171-175.

[14] Marmanis, D., Schindler, K., Wegner, J. D., Galliani, S. Classification with an Edge: Improving Semantic Image Segmentation with Boundary Detection. Web: https://arxiv.org/pdf/1612.01337.pdf

[15] Wilt, N. 2013. The CUDA Handbook. A Comprehensive Guide to GPU Programming. Addison-Wesley Professional.

[16] Atienza, R. 2018. Advanced Deep Learning with Keras. Packt Publishing, UK.

[17] Kingma, D., Ba, J. Adam: A Method for Stochastic Optimization. Web: https://arxiv.org/abs/1412.6980.

[18] Losses for Image Segmentation. Web: https://lars76.github.io/neural-networks/object-detection/losses-for-segmentation/.