Separation of Closely Located Buildings on Aerial Images Using U-Net Neural Network

Roman Larionov, Vladimir Khryashchev, Vladimir Pavlov
P.G. Demidov Yaroslavl State University
Yaroslavl, Russian Federation
r.larionov1@uniyar.ac.ru, v.khryashchev@uniyar.ac.ru, i@yajon.ru

Abstract—**Deep learning and modern type of neural network technologies are increasingly used for the detection, segmentation and classification of different objects in aerial multichannel images. The goal of given research was to develop a deep learning algorithm for automated building detection on four-channel satellite images. It is proposed to use U-Net neural network with two decoders to separate objects, one decoder is trained to segment buildings and structures, and the other detects narrow distances between buildings. It is shown that optimized U-Net can be used to detect such kind of objects efficiently. The model was implemented by means of open Keras library and launched on modern GPUs of high-performance supercomputer NVIDIA DGX-1. Before testing of developed algorithm on Planet aerial image dataset, modified U-Net had been pre-trained on SpaceNet database. The process of image data augmentation is described. The problem of effective building detection on high-resolution aerial photos can be used for urban planning, building control, search of the best locations for future outlets etc.**

I. Introduction

Nowadays remote sensing is widely used in different human activities: agriculture and yield monitoring, dynamics tracking and disaster prediction, forestry, construction, meteorology, measuring the level of urbanization of countries etc. The market of remote sensing data is growing every year. The effectiveness of solving problems in the field of satellite image processing depends on the speed and quality of image processing [1].

In machine learning, image segmentation is reformulated as dividing an image into areas for which a certain uniformity criterion is fulfilled [2], for example, water objects, forests or deserts. Classical algorithms, such as WaterShed, MeanShift, FloodFill, etc., are based on a gradient image map and are not able to segment images with a large number of objects [3]. In [4] the effectiveness of deep learning methods in satellite image segmentation is shown.

This article presents developed convolutional neural network (CNN) as the most effective deep learning method in image processing [5]. CNNs can detect and classify objects in real time while being computationally less expensive and superior in performance compared with other machine learning methods [6]. 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.

In paper [7] there is presented U-Net architecture – a specific type of Feature Pyramid Network, which has shown its effectiveness only in medical image segmentation. Later this model was applied to pixel-wise classification of satellite images [8, 9]. 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.

In the images of large cities, buildings can be located close to each other and merge into one object after segmentation. To solve this problem, the Watershed method is known, in [10] the authors presented Deep Watershed Transform (DWT) based on CNNs. DWT consists of two consecutive convolutional neural networks: Direction Net with the original image and the result of segmentation as inputs and Watershed Transform Net. As a result, an image with separated objects from each other is obtained. In this paper, it is proposed to use U-Net with two decoders to separate objects, one decoder is trained to segment buildings and structures, and the other detects narrow distances between buildings.

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.

II. Databases of satellite images

The GeoEye-1 database [11] 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 [12] contains four-channel (blue, green, red, near-infrared) images from Pleiades-1B satellite. 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.

SpaceNet dataset contains commercial satellite imagery, which is accompanied by markup information. Satellite imagery was provided by Digital Globe, one of the leading providers of high resolution satellite imagery and geospatial data [13]. The company has the world's largest archive of images of the Earth's surface. Images were taken from QuickBird, GeoEye-1 and WorldView satellites (WorldView-1, WorldView-2, WorldView-3 and WorldView-4). Data from various satellites differed in their characteristics, such as spatial resolution, geolocation accuracy, and others. Eight-channel images of 650×650 pixels cover 6 large metropolitan areas: Rio de Janeiro (Brazil), Las Vegas (USA) , Paris (France), Shanghai (China), Khartoum (Sudan) and Atlanta (USA). 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.

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.

III. Data preparation

Numerical experiments for developed deep learning algorithm were performed on normalized satellite images of the Planet database [14]. 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.

Generated training and test sets were enlarged using the following techniques [15]:

- 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 [16]. If the border between buildings on satellite images of Planet database did not exceed three pixels, this boundary was marked. Examples of masks with narrow distances between buildings are shown in Fig. 2.

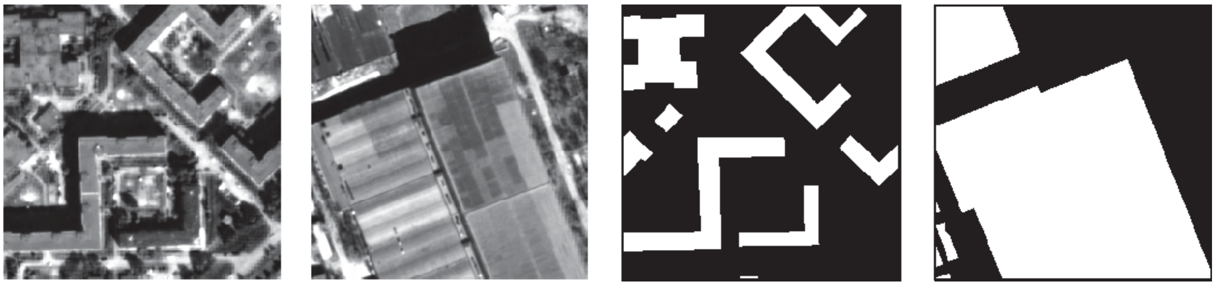

Fig. 1. Examples of patches and corresponding masks

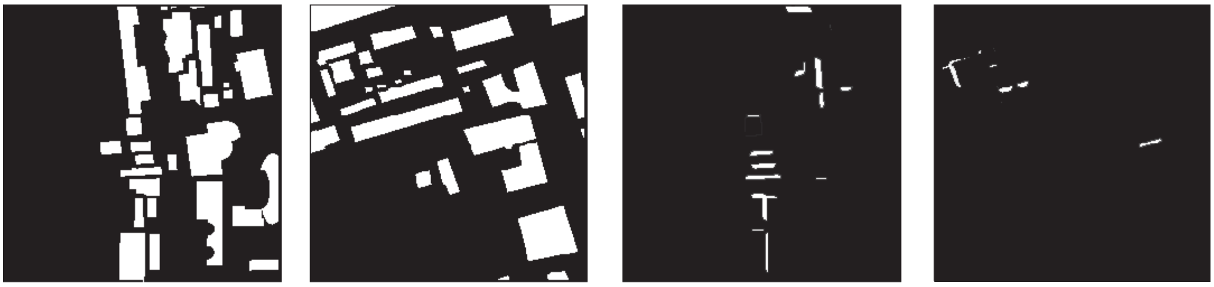

Fig. 2. Examples of buildings and corresponding narrow borders masks

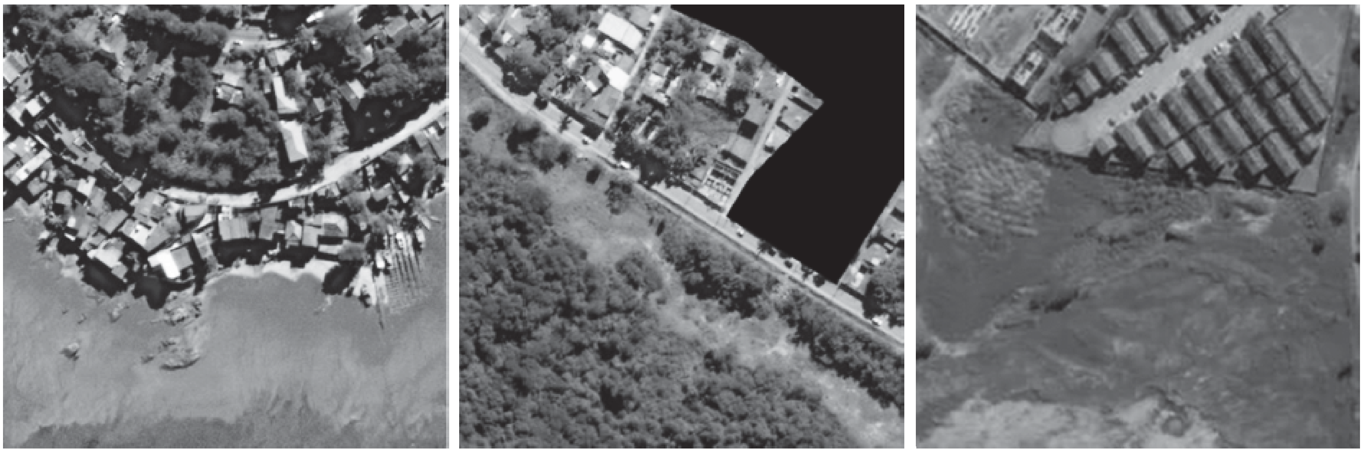

Fig. 3. Examples of images from Spacenet dataset

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). Examples of images from Spacenet database are shown in Fig. 3. 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.

IV. 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 [17, 18].

The neural 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 [7].

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 4.

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 [19]. On the test stage pixels which belong to predicted boundaries were reset on a segmentation mask.

CNN training requires considerable computational resources associated with matrix and tensor operations. Therefore, training and test stages were carried out on several GPUs using the parallel computing technology NVIDIA CUDA. All modern NVIDIA graphics cards support this technology [20].

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 [21]. Moreover, this library allows to train developed models on GPU.

Adaptive moment estimation (Adam) was used for optimization of training process. It's the modification of Alagard method and combines the idea of accumulation of movement and the idea of weaker updating of weights for typical features [22].

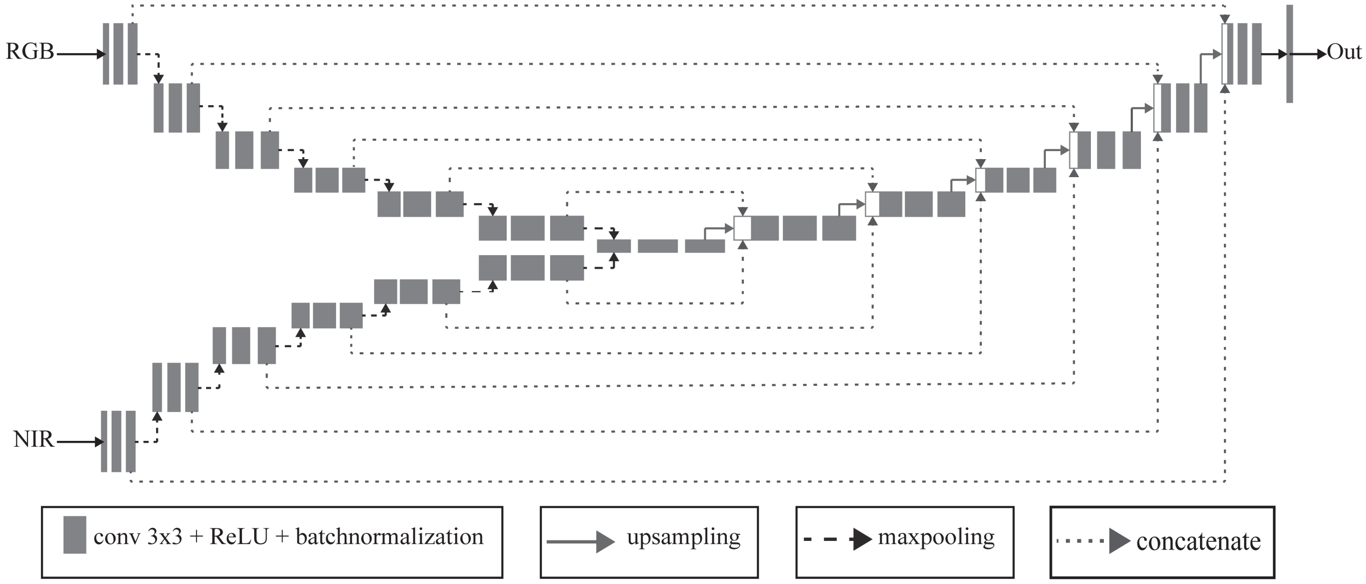

Fig. 4. U-Net with two encoders

V. 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 formulae:

$$DSC = \frac{2I}{S}$$

where $I = |X \cap Y|$ is a power of intersection and $S = |X| + |Y|$ is a sum of powers for real mask X and predictions Y. In our task, numerator I and denominator S can be calculated by following formulas:

$$I = \sum_{\substack{x \in X \\ y \in Y}} xy, \; S = \sum_{\substack{x \in X \\ y \in Y}} (x + y),$$

where x, y are values of pixels from [0, 1] for real masks X and predictions Y respectively. Also to measure the detection capability of developed deep learning algorithm there were used precision (P), recall (R) and F-score (F_1).

As a loss function there was used a sum:

$$Loss = BCE(X,Y) + DL(X,Y)$$

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

$$BCE(X,Y) = -\sum_{x,y} \left(x \log(y) + (1-x)\log(1-y) \right)$$

$$DL(X,Y) = 1 - DSC(X,Y)$$

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 [23]. 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 on the Spacenet database is presented in Table I, U-Net results without boundary detection are presented in Table II and finally, comparison of developed U-Net algorithm with boundary detection (U-Net BD) and Deep Watershed Transform (DWT) are presented in Table III. Dependencies of loss function values on training epochs for U-net BD are shown in Fig. 5.

TABLE I. TEST RESULTS OF ON THE SPACENET DATABASE

Regions	**Metrics**			
	DSC	**P**	**R**	**F1**
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**

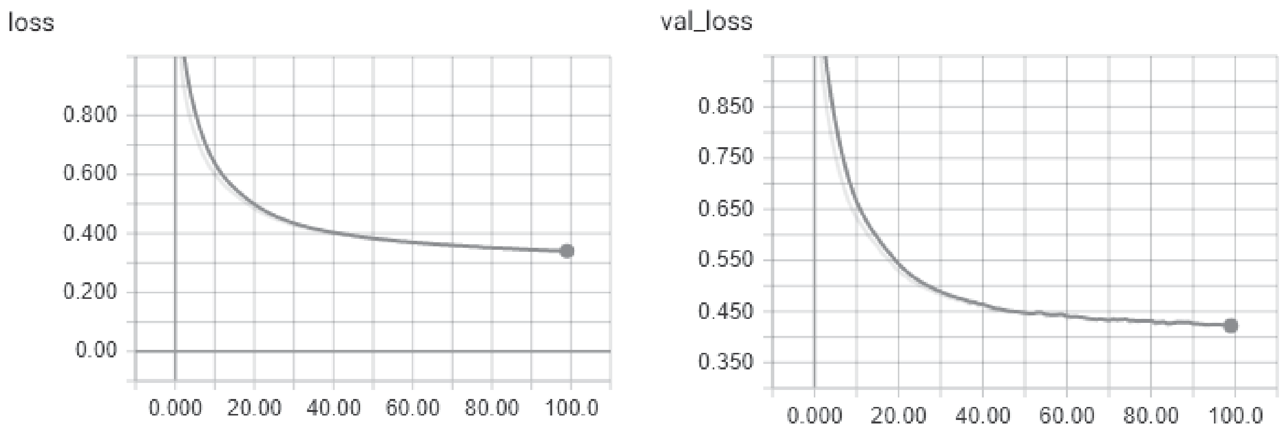

Fig. 5. Dependencies of loss function values on training and validation sets for U-Net BD

TABLE II. TEST RESULTS OF TUNING ON THE PLANET DATABASE WITHOUT BOUNDARY DETECTION

Regions	Metrics	
	DSC	F1
1	0.706	0.418
2	0.880	0.591
3	0.571	0.318
4	0.787	0.500
5	0.610	0.424
6	0.703	0.46
7	0.719	0.626
8	0.778	0.287
9	0.992	0.000
10	0.599	0.328
11	0.756	0.358
12	0.883	0.447
13	0.907	0.769
14	0.805	0.587
15	0.712	0.598
16	0.686	0.443
17	0.724	0.418
Average Values	**0.754**	**0.446**

According to results presented in Table II, 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 and the total value reached 0.77. Also F1 value increased from 0.45 to 0.49. It means that developed algorithm is able to separate different buildings better. According to results of Table III DSC value of DWT algorithm is equal 0.75. 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. 6.

TABLE III. RESULTS OF U-NET WITH BOUNDARY DETECTION AND DWT

Regions	U-Net BD		DWT	
	DSC	F1	DSC	F1
1	0.743	0.499	0.724	0.545
2	0.884	0.653	0.879	0.646
3	0.615	0.392	0.613	0.436
4	0.828	0.610	0.805	0.648
5	0.647	0.488	0.563	0.382
6	0.739	0.499	0.648	0.396
7	0.708	0.605	0.589	0.443
8	0.784	0.321	0.743	0.286
9	0.960	0.000	1.0	0.0
10	0.622	0.345	0.596	0.299
11	0.711	0.371	0.724	0.361
12	0.877	0.512	0.880	0.543
13	0.910	0.806	0.881	0.779
14	0.804	0.582	0.804	0.628
15	0.759	0.599	0.762	0.591
16	0.706	0.513	0.704	0.547
17	0.759	0.521	0.799	0.616
Average Values	**0.768**	**0.489**	**0.748**	**0.479**

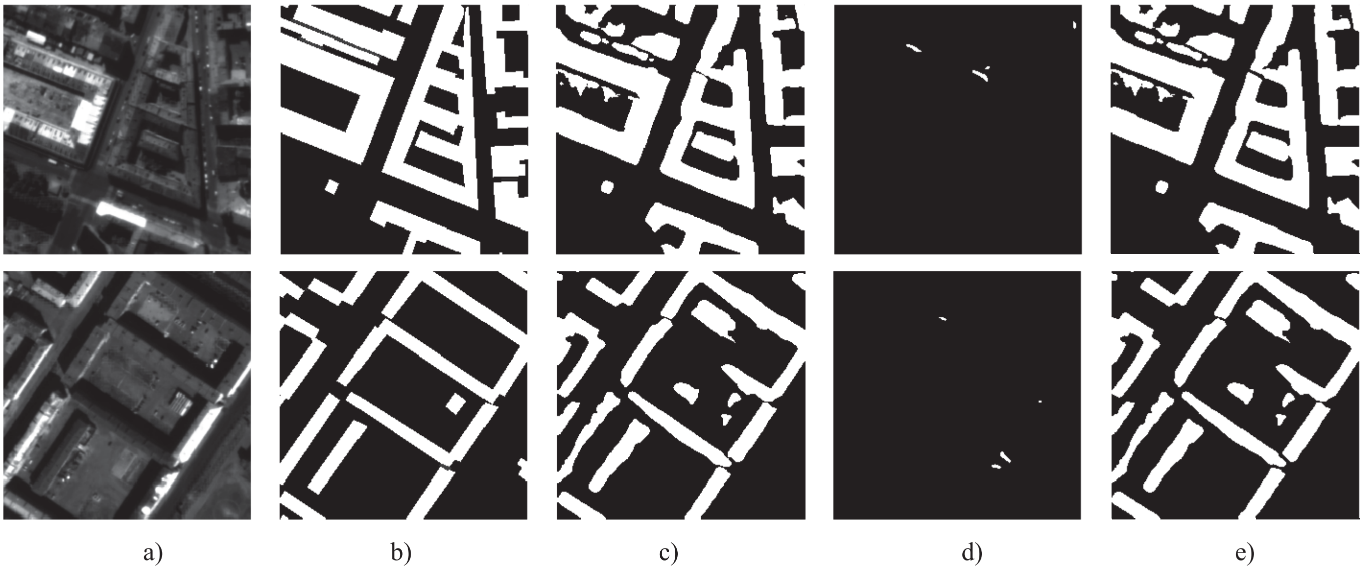

Fig. 6. Test results a) input images, b) manual marked masks with tagged buildings, c) results of segmentation excluding boundary detection, d) narrow boundaries segmentation, e) final results of segmentation

VI. 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.

Acknowledgment

This work was supported by a grant from the P.G. Demidov Yaroslavl State University № ОП-2Г-09-2019

References

[1] X. X. Zhu, “Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources”, *IEEE Geoscience and Remote Sensing Magazine,* 2017, vol. 5, no. 4, pp. 8-36.

[2] X. Chen, S. Xiang, C. Liu, C. Pan, “Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks”, *IEEE Geoscience and Remote Sensing Letters*, 2015, vol. 11, no 10 pp. 1797–1801.

[3] R.Venkatesan, B. Li, “*Convolutional neural networks in visual computing: a concise guide*”, CRC Press, 2017, 187 p.

[4] E. Shelhamer, J. Long, T. Darrell, “Fully Convolutional Networks for Semantic Segmentation”, Web: https://arxiv.org/abs/1605.06211.

[5] A. Geron, “Hands-On Machine Learning with Scikit-Learn and Tensorflow: Concepts, Tools and Techniques to Build Intelligent Systems”, O’Reilly Media, 2019, 856 p.

[6] S. Seferbekov, V. Iglovikov, A. Buslaev, A. Shvets, “Feature Pyramid Network for Multi-Class Land Segmentation”, Web: arXiv: https://arxiv.org/abs/1806.03510.

[7] O. Ronneberger, P. Fischer, T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, LNCS, 2015, vol. 9351, pp. 234–241.

[8] V. Khryashchev, L. Ivanovsky, V. Pavlov, A. Ostrovskaya, A. Rubtsov, “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, 2018, pp. 172-179.

[9] T Zhao, Y. Yang, H. Niu, D. Wang, “Comparing U-Net convolutional network with mask R-CNN in the performances of pomegranate tree canopy segmentation”, in *Proceedings of SPIE*, 2018, vol. 10780, pp. 1-9.

[10] M. Bai, R. Urtasun, “Deep Watershed Transform for Instance Segmentation”, Web: https://arxiv.org/abs/1611.08303

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

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

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

[14] L. Zhang, L. Zhang and B. Du, "Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art," in IEEE Geoscience and Remote Sensing Magazine, vol. 4, no. 2, pp. 22-40, June 2016.

[15] L. Weijia, H. Conghui, F. Jiarui, Z. Juepeng, “Semantic Segmentation-Based Building Footprint Extraction Using Very High-Resolution Satellite Images and Multi-Source GIS Data”, *Remote Sensing,* vol. 11, no. 4, pp. 1-19.

[16] S. Ohleyer, “Building segmentation on satellite images”, Web: https://project.inria.fr/aerialimagelabeling/files/2018/01/fp_ohleyer_compressed.pdf

[17] L. Ivanovsky, V. Khryashchev, V. Pavlov, A. Ostrovskaya, “Building Detection on Aerial Images Using U-NET Neural Networks”, *In Proceedings of the 24rd Conference of Open Innovations Association FRUCT’24*, Moscow, Russia, 2019, pp. 116-122.

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

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

[20] N. Wilt, “*The CUDA Handbook. A Comprehensive Guide to GPU Programming*”, Addison-Wesley Professional, 2013, 520 p.

[21] R. Atienza, “*Advanced Deep Learning with Keras*”, Packt Publishing, UK, 369 p.

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

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