**Comparison of Different Deep Learning Algorithms for Mines Detection on Satellite Images**

Vladimir Khryashchev, Roman Larionov, 3rd author

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)

**ABSTRACT**

This article presents results of two deep learning algorithms for mines detection on high-resolution aerial photos. Input images contain blue, green, red and near-infrared channels. Before the training process there was implemented the equalization of brightness histogram which allows to cope with the problem of obscuration of satellite images. To implement numerical experiments there were extracted smaller patches. Training and test sets were enlarged methods using various methods of data augmentation. Convolutional neural networks were pretrained on the SpaceNet dataset and tuned on the Planet database. Deep learning algorithms were launched on NVIDIA DGX-1 supercomputer, which was provided by AI-center of P.G Demidov Yaroslavl State University. Special metrics allowed to compare the quality of developed models.

Keywords: mines detection, computer vision, high-resolution aerial photos, convolutional neural networks.

**1. INTRODUCTION**

Remote sensing cannot completely replace ground-based data collection, but can help in monitoring of large areas and hard-to-reach regions. However, nowadays computer vision algorithms for detecting objects on satellite images may implement automatic segmentation of high-resolution aerial photos much more faster than human does. Humans are able to solve a segmentation problem better than a computer, but it takes too much time. In this case it is impossible to obtain results in real time. Despite the rapid development of such algorithms, tasks of image segmentation are particularly relevant.

Today, large number of algorithms for detecting objects on satellite images exists. The main approach of its solution based on machine learning methods which marks each image pixel to corresponding classes of objects. Currently, the greatest effectiveness of solving this problem is achieved by using convolutional neural networks (CNNs). The uniqueness of this method is based on the automatic determination of features in the training process [1].

This paper presents developed CNNs for the task of mines detection. Such networks are capable for real-time detecting and classifying objects. At the same time, there are some requirements for satellite images segmentation:

* Size and type of buildings and structures may significantly vary from cottages and little lakes to huge apartment buildings and wide rivers. The algorithm should detect objects of any size very well.
* Trained model should be invariant to rotations and color distortions of objects. This problem can solved by data augmentation.
* Aerial images have different spatial resolution. It’s important to create deep learning algorithms which have an ability to handle high-resolution aerial photos perfectly. In this case, algorithm can process fragments of images which are extracted by data windowing.

This paper presents CNNs that can be used for mines detection on satellite images. The training process, testing and special metrics for assessing the quality of neural network work are described. Our work continues research, which was presented in [2].

This article consists of six parts. In the first part, a reader gets acquainted with the formulation of problem. The introduction devotes to CNNs as an approach in machine learning and complications of image segmentation. The second part contains an overview of some papers for object detection on high-resolution aerial photos. The third part refers to some available databases which consist of satellite images of nature. The fourth section describes the process of data augmentation and cutting out of fragments before the training of CNNs. Developed deep learning algorithms for mines detection on high-resolution aerial photos and some peculiarities of training these models were considered in the fifth part of this article. The sixth part presents the results of numerical experiments. In the conclusion there is summarized the research. And finally, the last sections contain the acknowledgement and references.

**2. RELATED WORKS**

In recent years, various methods for creating CNN have been proposed, which can produce segmentation for the entire input image. One of the most successful algorithms is based on fully convolutional networks (FCNs). The basic idea of this approach is to use CNN to extract the necessary feature values, while replacing the fully connected layer with a convolution layer with the output in the form of feature maps instead of classification results [1]. This method allows you to train CNN for the segmentation of images of different sizes.

In paper [3] there was proposed a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection. Fast R-CNN builds to efficiently classify object proposals using CNNs. Compared to various deep learning algorithms, Fast R-CNN improves training and testing speed while also increases detection accuracy. Fast R-CNN shows high results of object detection on PASCAL VOC datasets [4].

Following this way, the authors of [5] present the architecture of CNN named Mask R-CNN, using pretrained weights of COCO dataset. This nonsimple method allowed to detect buildings on satellite photos of Inria Aerial Images Dataset [6] exactly.

The method of using FCN was supplemented and now it is known as U-Net. In papers [7, 8] there is presented U-Net architecture – a specific type of FCN, which had received a lot of interest for segmentation of biomedical 2D and 3D images. The U-Net architecture uses skip-connections to combine low-level and higher-level feature maps, which provides accurate localization of objects. The authors of [9] hold the similar method to solve the problem of satellite images segmentation. They developed the U-Net like architecture, which is using ResNet-34 weights in the encoder. This algorithm shows excellent results of detecting roads on satellite images of DeepGlobe database [10].

**3. DATABASES OF SATELLITE IMAGES**

DSTL database constains 50 satellite images in GEOTIFF format. For the first time, this dataset was provided in Kaggle competition “DSTL Satellite Imagery Feature Detection” [11]. Three-channel images of DSTL dataset are labeled on 10 different classes: “buildings”, “manmade structures”, “roads”, “tracks”, “trees”, “crops”, “waterway”, “standing water”, “large vehicles” (e.g. lorries, trucks or buses) and “small vehicles” (cars, vans or bikes).

The SpaceNet database [12] includes 11-bit 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 high-resolution aerial 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. In out research the SpaceNet dataset was used for pre-training developed deep learning algorithms.

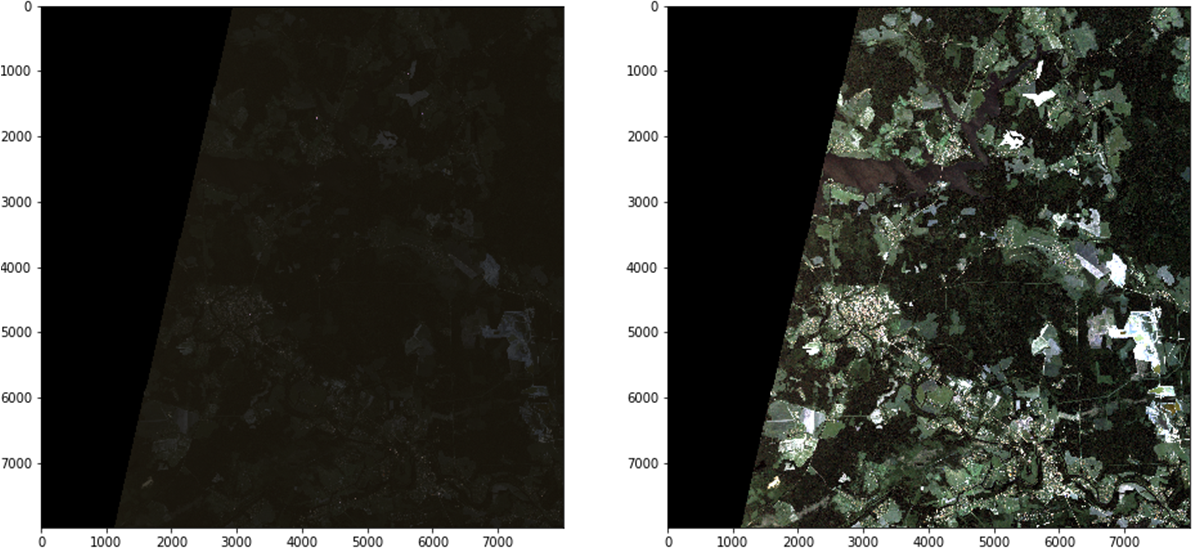
The SkySat-1 database contains color satellite images from SkySat-1 sensor, which was on the 21th of November in 2013 from Yasny space launching site in Russia. The SkySat-1 sensor is licensed to collect high-resolution 0.8m panchromatic and 1.0m multispectral imagery for various applications in energy, defense, agriculture, forestry and environmental monitoring [13]. The SkySat-1 satellite operates in a polar inclined, circular orbit at approximately 450 km above the Earth.

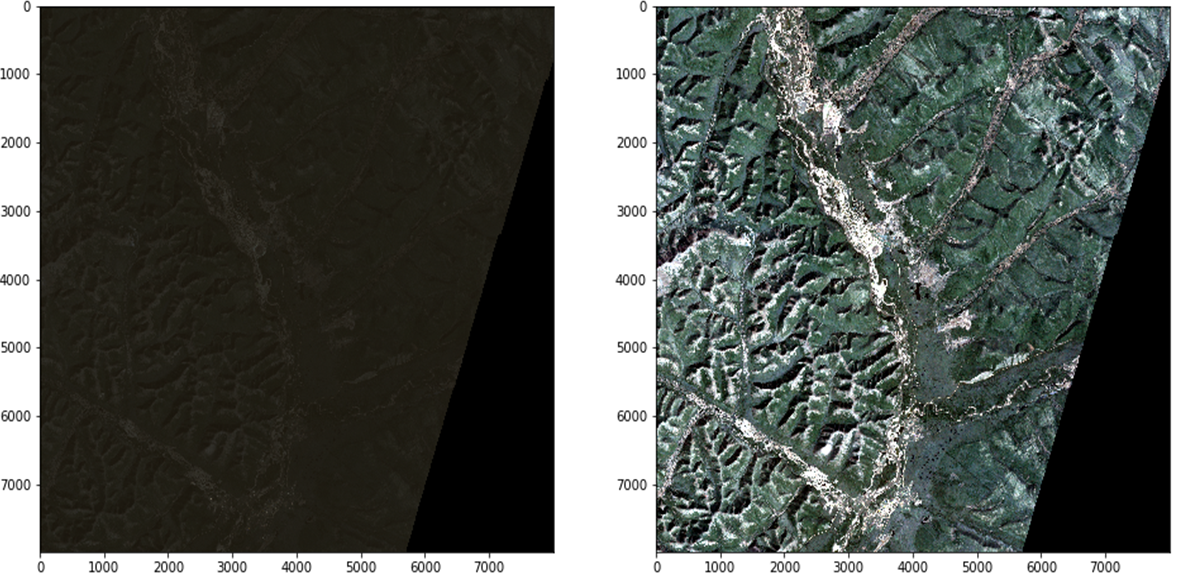
In our research, for training and testing of developed deep learning algorithms there were used 12-bit four-channel (blue, green, red, near-infrared) satellite images from the Planet database. Each of 55 high-resolution aerial photos of this dataset were shot by RapidEye satellite sensor. Images have a spatial resolution of 3 m/pixel. Some satellite images from the Planet database have noisy pixels, such as photographed clouds or glares from building roofs and water.

**4. DATA PREPARATION**

Noisy and very bright pixels may shift the general brightness histogram of image. This side effect leads to the obscuration of full image. It affects the training process of CNNs and as a result a deep learning algorithm might work incorrectly. The equalization of brightness histogram allows to cope with this problem.

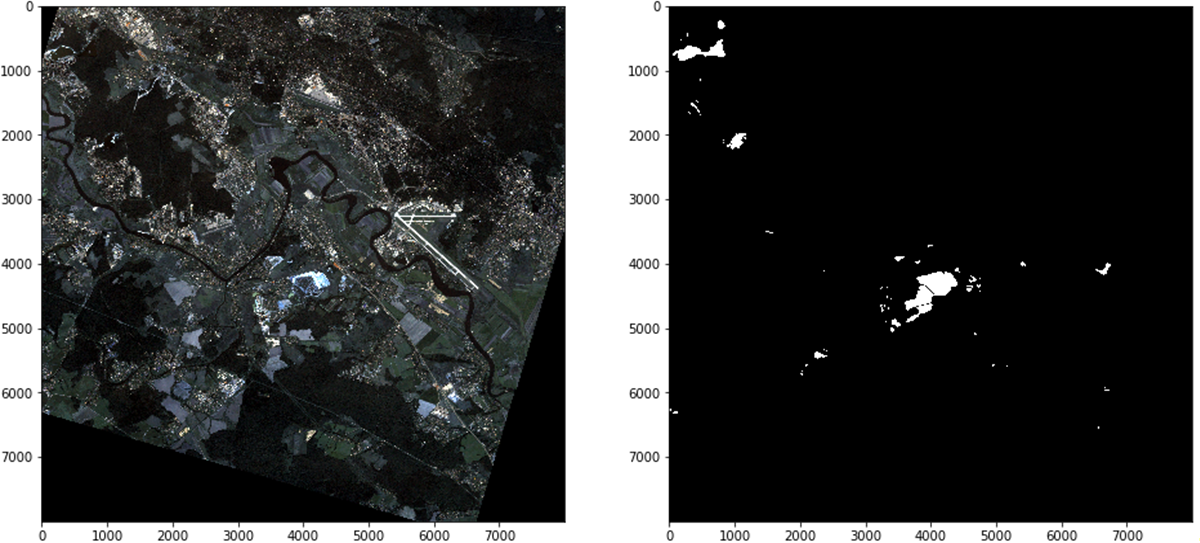
Histogram equalization is a process of image preprocessing which redistribute all pixels of image uniformly. Values of pixels that do not fall into the certain range of brightness are replaced with minimal or maximal threshold values of brightness. To align the histogram and increase its range of brightness, a linear transformation was performed for each pixel of satellite image. Results of equalization of brightness histogram for satellite images of the Planet database are shown in Figure 1. Equalized satellite image and masks from the Planet database are shown in Figure 2.





**a) b)**

**Figure 1. Results of equalization of brightness histogram for images of the Planet database: a) initial aerial photos, b) transformed satellite images**



**a) b)**

**Figure 2. Example of an equalized satellite image and expert markup from the Planet database: a) equalized satellite image, b) true mask**

Numerical experiments for developed deep learning algorithms 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. Before the training of CNN by means of data windowing each high-resolution photo and mask of dataset have been sliced on parts of 512 × 512 and 1024 × 1024 size with the step of 256 and 512 respectively.The intersection of patches allows to cope with problem of artifacts that occur at the junction of image fragments. Information about prepared patches of different size is presented in Table 1, 2.

**Table 1. Prepared patches of 512 × 512 size**

|  |  |  |
| --- | --- | --- |
|  | **Training set** | **Test set** |
| **Total** | 18900 | 9060 |
| **With objects** | 1831 | 652 |
| **Without objects** | 17069 | 8408 |

**Table 2. Prepared patches of 1024 × 1024 size**

|  |  |  |
| --- | --- | --- |
|  | **Training set** | **Test set** |
| **Total** | 12585 | 1525 |
| **With objects** | 2359 | 213 |
| **Without objects** | 10226 | 1312 |

To enlarge training and test sets there were used 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˚].

However, extended training and test sets still had insufficient data to train CNNs from scratch. Therefore these sets were used only for tuning. Before the training of models on the images Planet database, they had been pretrained on the SpaceNet dataset. In our research, there were used only 8-channel normalized satellite images of 4 regions (Las Vegas, Paris, Shanghai and Khartoum) with tagged buildings, because these objects have the biggest variance of size in this dataset. 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.

**5. CONVOLUTIONAL NEURAL NETWORKS**

In this section there are described developed CNNs used for mines detection on high-resolution aerial photos, and some peculiarities of their training.

First of all, for mines detection on satellite images there was developed the special neural network architecture called Mask R-CNN. This deep learning algorithm extends the functionality of well-known method for object detection – Faster R-CNN [14]. The architecture of Mask R-CNN is shown in Fig. 3.



**Figure 3. Architecture of Mask R-CNN.**

In Mask R-CNN a fully connected neural network (FCN) as an additional branch was added to the basic algorithm. FCN allows to make a segmentation at the pixel level in areas of detected objects (RoI) in parallel with existing outputs of Faster R-CNN: classes and bounding boxes.

Faster R-CNN represents a sequence of two machine learning algorithms: a region-proposal network (RPN) and Fast R-CNN. RPN makes predictions about possible locations of bounding boxes for areas of interest (RoI) on satellite images using sliding windows and «anchors» - special rectangular frames of various size and different ration of sides that surround objects of interest. The presence or absence of objects inside each frame is determined due to the value of IoU metrics («intersection-over-union»): if the value of IoU is more than 0.5 then it is considered that the object fell into the frame. As RPN architecture, we used Feature Pyramid Network (FPN) [15]. The main advantage of this approach is to improve the quality of detection, taking into account a wide range of possible sizes of objects. Feature maps of lower and upper layers of FPN have pros and cons: lower layers have high resolution, but low semantic, generalizing ability, whereas upper layers have low resolution, but good generalizing ability to extract needed unobvious features.

Fast R-CNN, extracts a set of frames which surround objects of interest more exactly and classifies detected objects simultaneously. The key components of this deep learning algorithm are the spatially pyramidal layer with RoIPool operation, which extracts a feature map for each RoI and RoIAlign layer, which determines the exact spatial location of frames for detected objects on images.

Also there was developed UNet-like architecture – U-ResNet34 based on models from paper [9]. U-ResNet34 is a U-Net neural network, where ResNet34 used as an encoder and decoder was copied from the classic U-Net architecture.

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 [16].

TensorFlow is an open-source software library for high performance numerical computation on CPU or GPU. It is a symbolic math library, and is also used for machine learning applications such as neural networks. In this field Tensorflow aims at fast detection and classification of images, achieving the quality of human perception The library contains various implementations of commonly used neural network building blocks and ready tools to preprocess images and text data [17].

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 [18].

**6. NUMERICAL RESULTS**

The training and testing of Mask R-CNN and U-ResNet34 were carried out 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 metrics which evaluate the similarity of predicted and true masks. In our research there was used IoU coefficient, a binary measure of similarity predicted and true masks. Also to measure the quality of developed deep learning algorithms there were used precision (), recall () and F-score (). During the training, on the assumption of maximal value of , for each CNN there was chosen a threshold value to form predicted masks with detected mines. If the value of possibility for a pixel is higher than the threshold value, it belongs to the class of interest.

As a loss function for real masks and predictions there was used a sum of binary cross entropy (BCE) and the value of Dice loss (DL):

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 [19].

Mines vary in size greatly, from tens to hundreds of meters in linear dimensions. Therefore, it is necessary to choose the right sizes for anchors to cover small and large objects of interest. For Mask R-CNN there were selected two sets of anchors:

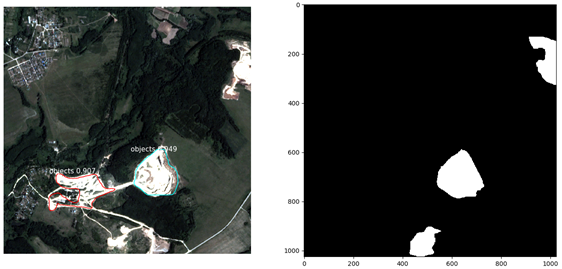
* Anchor set 1 (AS 1): 64, 128, 256, 512 and 1024;
* Anchor set 2 (AS 2): 32, 64, 128, 256 and 512.

For Mask R-CNN there were used training and test sets of patches of 1024 × 1024 size which were enlarged using all techniques described earlier. The training and tuning processes finish after completing 113 epochs. Test results for Mask R-CNN with different sets of anchors on the Planet database were presented in Table 3.

**Table 3. Test results for Mask R-CNN on the Planet database.**

|  |  |  |
| --- | --- | --- |
| **Metrics** | **Mask R-CNN (AS 1)** | **Mask R-CNN (AS 2)** |
|  | 0,753 | 0,757 |
|  | 0,252 | 0,142 |
|  | 0,189 | 0,081 |
|  | 0,380 | 0,584 |

Mask R-CNN with the set of larger anchors is shown better results: the value of reached 0,252 in comparison with 0,142 which was given on the set of smaller anchors. This peculiarity can be explained by the fact that mines are big enough on satellite images so the deep learning algorithm with little bounding boxes are not able to detect whole objects of interest. The example of an input image with detected objects and corresponding true mask of the Planet database for Mask R-CNN is shown in Fig. 4.



**a) b)**

**Figure 4. Test results for Mask R-CNN: a) input image with detected objects, b) true mask**

The second model, U-ResNet34, was trained on two sets of patches of 512 × 512 size. Sets of fragments were enlarged by methods of data augmentation in two ways:

1. Only flips
2. flips and SSR.

For developed deep learning algorithm the training and tuning processes finish after completing 80 epochs, while on each training step of epoch a batch of 16 samples was passed through the developed model. Test results for U-ResNet34 on the Planet database were presented in Table 4.

**Table 4. Test results for developed models on the Planet database.**

|  |  |  |
| --- | --- | --- |
| **Metrics** | **U-ResNet34 (flips)** | **U-ResNet34 (flips + SSR)** |
|  | 0,772 | 0,765 |
|  | 0,408 | 0,357 |
|  | 0,375 | 0,289 |
|  | 0,447 | 0,465 |

According to results presented in Tables 3, 4, the best value of for U-ResNet34 reached 0,408 and exceed the same metric for Mask R-CNN on 0,156. Thus, U-ResNet34 works better in the task of mines detection on high-resolution aerial photos. However, the usage of image shifts and rotations on small angles do not improve the quality of deep learning algorithm: an amount of false positive objects increased vastly. Small sandy areas usually acted as false positive objects. The example of an input image, true and predicted masks of the Planet database for U-ResNet34 is shown in Fig. 5.

**a) b) c)**

**Figure 5. Test results for U-ResNet34: a) input image, b) true mask, c) predicted mask**

**7. CONCLUSION**

This article presents numerical experiments for developed deep learning algorithms: Mask R-CNN and U-ResNet34. The training and testing process were performed on high-resolution aerial photos of the Planet database. Before the training process there was implemented the equalization of brightness histogram which allows to cope with the problem of obscuration of satellite images. To implement numerical experiments there were extracted smaller patches. Training and test sets were enlarged methods using various methods of data augmentation. The developed algorithms were pretrained on the SpaceNet dataset and tuned on the Planet database. According to test results U-ResNet34 works better in the task of mines detection on high-resolution aerial photos. For learning of model there was used supercomputer NVIDIA DGX-1.

**ACKNOWLEDGMENT**

The authors are grateful to the AI-centre of P.G. Demidov Yaroslavl State University for providing an access to NVIDIA DGX-1 supercomputer to carry out numerical experiments.

**REFERENCES**

[1] Goodfellow, Y., Bengio, Y., Courville, A. 2016. Deep Learning. The MIT Press.

[2] 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.

[3] Girshick, R. Fast R-CNN. Web: https://arxiv.org/pdf/1504.08083.pdf.

[4] The PASCAL Visual Object Classes Homepage. Web: http://host.robots.ox.ac.uk/pascal/VOC/.

[5] He, K., Gkioxari, G., Dollar, P., Girshick, R. Mask R-CNN. Web: https://arxiv.org/pdf/1703.06870.pdf.

[6] E. Maggiori, Y. Tarabalka, G. Charpiat, P. Alliez, “Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark”, *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. 2017.

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

[8] Patravali, J., Jain, S., Chilamkurthy, S. 2017. 2D-3D Fully Convolutional Neural Networks for Cardiac MR Segmentation In Qure.AI.

[9] Buslaev, A. Seferbekov, S., Iglovikov, V., Shvets, A. 2018. Fully Convolutional Network for Automatic Road Extraction from Satellite Imagery. CoRR, abs/1806.05182.

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

[11] DSTL Satellite Imagery Feature Detection, Web: https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection.

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

[13] SkySat-1 Satellite Images. Web: https://www.satimagingcorp.com/satellite-sensors/skysat-1/.

[14] Ren, S., He, K., Girshick, R., Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Web: https://arxiv.org/pdf/1506.01497.pdf.

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

[16] Tuomanen, B. 2018. Hands-On GPU Programming with Python and CUDA: Explore high-performance parallel computing with CUDA. Packt Publishing.

[17] Hope, T., Resheff, Y.S., Leder, I. 2017. Learning TensorFlow: A Guide to Building Deep Learning Systems. O’Reilly Media.

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

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