This is a preprint of an article published in IEEE Journal on Miniaturization for Air and Space Systems, 2(2) (2021) 98-103
The final authenticated version is available online at: https://doi.org/10.1109/JMASS.2020.3040976

Object detection for unmanned aerial vehicle camera via convolutional neural networks

Ivan.V. Saetchnikov, Elina. A. Tcherniavskaia and Victor.V. Skakun

***Abstract* — The object tracking alongside the image segmentation have recently become of particular significance in satellite and aerial imagery. The latest achievements in this field are closely related to the application of the deep-learning algorithms and, particularly, convolutional neural networks (CNNs). Supplemented by the sufficient amount of the training data CNNs provide the advantageous performance in comparison to the classical methods based on Viola-Jones or Support vector machines. However, the application of CNNs for the object detection on the aerial images faces several general issues that cause classification error. The first one is related to the limited camera shooting angle and spatial resolution. The second one arises from the restricted dataset for specific classes of objects that rarely appear in the captured data. This paper represents a comparative study on the effectiveness of different deep neural networks for detection of the objects with similar patterns on the images within a limited amount of the pre-trained datasets. It has been revealed that YOLO ver. 3 network enables better accuracy and faster analysis than R-CNN, Fast R-CNN, Faster R-CNN, and SSD architectures. This has been demonstrated on example of "Stanford Dataset", "DOTA v-1.5", and "xView 2018 Detection" datasets. The following metrics on the accuracy have been obtained for the YOLO ver. 3 network: 89.12 mAP (Stanford Dataset), 80.20 mAP (DOTA v-1.5), and 78.29 (xView 2018) for testing; and 85.51 mAP (Stanford Dataset), 79.28 (DOTA v-1.5), and 79.92 (xView 2018) on validation with the analysis speed of 26.82 frames per second.**

***Keywords* — Image segmentation, Pattern recognition, Object detection, Neural network architecture.**

I. Introduction

Miniaturization of the microelectronics and expansion of the computing capabilities have boosted the development of the multifunctional systems that can be installed within the modern small unmanned aerial vehicles (UAV). Among the possible application scenarios for both scientific and commercial UAVs a widely demanded but the most resource-intensive is the image/video capturing and processing. This task is commonly characterized by the real-time object detection and tracking that is relevant for a wide range of applications such as remote monitoring, navigation, logistics, telecommunications, etc. A special interest of these systems has been gained for security surveillance [1]-[2]. The adaptive device control, pathfinding and launch [3]-[5], monitoring of the traffic of the self-driving cars in the urban areas [6], and the earth remote sensing [7]-[9] belong to the most frequently reported implementations of the computer vision systems within the mentioned application fields of the UAVs and satellites. Solutions for object detection may belong here to either supervised or unsupervised learning algorithms.

The unsupervised learning implementations are represented by the standard pixel-wise segmentation algorithms and machine learning methods. For example, the face detection problem can be addressed with a combination of the optimized decision trees that compare the intensity of the pixels in the internal nodes [10] whereas a method based on the Haar-like features can be employed for human detection [11]. Although, the key feature of such algorithms is the high processing speed, they typically suffer from the lack of the accuracy for the recognition of the highly correlated objects from different classes and the background-covered objects.

The supervised machine learning methods for object detection include Support Vector Machine (SVM), k-nearest neighbors, and methods based on deep neural networks (dNN) among which the latter become currently a common approach with already 40 different network architectures proposed since 2013 [12]-[13].

The main benefit serving for the widespread utilization of the dNN algorithms is the high accuracy ensured for object detection. At the same time, these solutions demonstrate rather low scalability for multiple classes detection (when a limited number of the pre-trained dataset is provided), as well as longtime constraints related to the network configuration, learning and tuning. Furthermore, the object detection can appear substantially challenging for dNNs compared to the classification task due to the variety of the specific features of the aerial images [14]. Among them one can mention the class imbalances, limited data availability, limited camera shooting angle, and spatial resolution. Taken together this mainly results in the visual similarity of

Ivan V. Saetchnikov is with the Radio Physics Department, Belarusian State University, 220030 Minsk, Belarus (e-mail: saetchnikovivan@gmail.com)

Elina A. Tcherniavskaia is with the Physics Department, Belarusian State University, 220064 Minsk, Belarus.

Victor V. Skakun is with the Radio Physics Department, Belarusian State University, 220030 Minsk, Belarus.

different types of objects appearing on the captured images, e.g. of various vehicles like small cars, trucks, cargo trucks, utility trucks, buses, etc. Another challenge for object detection with dNN is related to the limited availability of the datasets for specific objects' classes, where in case of the aerial imagery the dataset expansion is extra constrained by significant accompanying expenses. The mentioned problem is often stated along with the class imbalance issue [15], where objects of some classes, which have a high diversity on the images, overlap with the objects with a low diversity causing a significant drop of the resultant accuracy [16-17].

Examples of the dNN application for object detection include different versions of You Looks Only Once (Yolo) architecture (SmallYoloV3, TinyYoloVoc, TinyYoloNet, and DroNet) which provides up to 90% accuracy with a single-shot object detector for vehicle tracking on the road with UAVs [18]; tracking and partial forecasting of the motion direction of the object with Region Proposal Networks (RPN) allowing 75.92% accuracy [19], etc. Adaptation of the R-CNN, Fast R-CNN, Faster R-CNN, and R-FCN architectures for object detection tasks has been recently addressed via the rotation-invariant and Fisher discriminative CNN models [20]. Modification of the Fast R-CNN with 70.4% accuracy for detection, tracking and movement prediction of autonomous vehicles has been also demonstrated with PASCAL VOC 2007 dataset in [21]. The extraction of the features and simultaneous localization of the geospatial objects have been shown with a combination of the RPN with the contextual feature fusion network [22]. The adaptation of the learning phase of the RiCNN architecture proposed in [23] enforces the training samples to share the similar features to achieve rotation invariance. Very recently a study on implementation of the existing deep neural networks to several datasets, including self-developed large-scale detection dataset DIOR covered by 20 aerospace object categories has been reported [24].

Aiming at systemizing different dNN architectures in terms of their accuracy for object detection and study their application for the particular case of limited dataset including objects with similar patterns, this paper represents a comparison study on performance of R-CNN, Fast R-CNN, Faster R-CNN, and YOLO V3 architectures on example of the aerial images captured with UAVs and satellites. Training process has been performed on datasets of small and medium size ("Stanford Dataset", "DOTA v-1.5", and "xView 2018 Detection") with special techniques applied for dataset pre-processing. The possibilities to implement these solutions for on-board computer vision systems of UAV's have been discussed.

II. Methods and Procedures

Since it is challenging to predict which types of dNNs would be effective for the discussed task, the architectures based on different principles of feature extraction have to be compared. Among the known solutions the R-CNN, Fast R-CNN, Faster R-CNN, YOLO ver. 3, and SSD architectures have been reported in literature on benefits in processing speed and detection accuracy and thus have been selected for the study.

A. R-CNN

Region Convolution Neural Network is one of the first approaches used to determine the object on the image. The R-CNN network consists of several sequential steps (Fig. 1):

1) Determination of a set of hypotheses: based on selective search method a list of hypotheses is defined, which includes 2000 different regions partially overlapping with each other.
2) Features extraction using convolutional neural network and its encoding into a vector: each hypothesis is transferred independently and separately from each other to the input of the convolutional neural network.
3) Object classification within each hypothesis: to determine which particular object appears in the region under analysis, the list of separate classification models One vs. Rest is used. In fact, the binary classification problem is solved wherever the specific class is in the intended region exists.

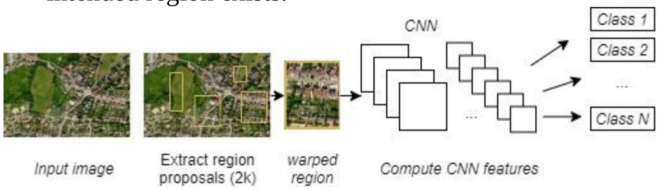

Fig. 1. Architecture of R-CNN network.

B. Fast R-CNN

The Fast R-CNN network is an advanced model of the R-CNN (the corresponding architecture is presented in Fig. 2) and differs:

1) Features map extraction is performed for the whole image.
2) Hypothesis search follows the selective search algorithm.
3) Each hypothesis is matched to a location on the feature map, i.e. a single set of selected features is used for each hypothesis.
4) Classification of each hypothesis and correction of the bounding box coordinates is performed (may be run in e parallel since the SVM classification algorithm is not used).

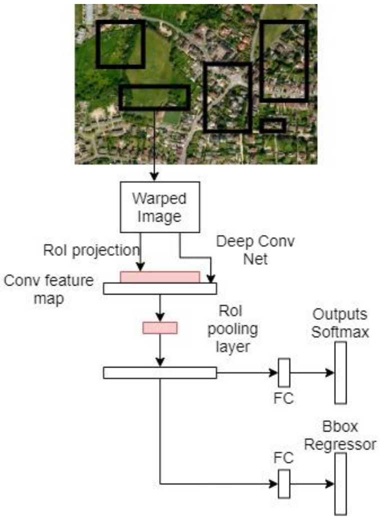

Fig. 2. General architecture of the Fast R-CNN network.

C. Faster R-CNN

In Faster R-CNN method the selective search algorithm is represented by two modules: hypotheses determination and its post-processing. The first module has been implemented using the Region Proposal Network (RPN). The second similarly to the Fast R-CNN (starting from the Region of Interest layer). The detection process using Faster R-CNN includes the following steps:

1) Images feature map extraction.
2) Generation of hypotheses based on the feature map, determination of the approximate coordinates.
3) Comparison of the hypotheses coordinates with the feature map using RoI.
4) Hypotheses classification and additional refinement of the coordinates.

The major improvement takes place precisely at the stage of hypotheses generation via an auxiliary neural network referred to as Region Proposal Network (see Fig. 3).

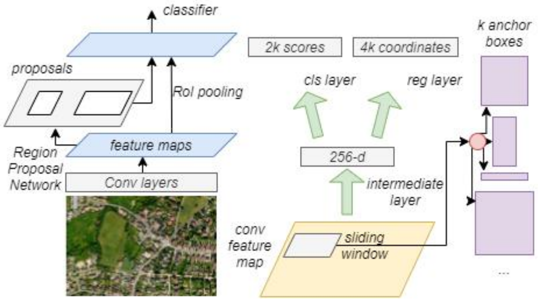

Fig. 3. Regional Proposal Network.

A summary of the evolution of the R-CNN architectures is shown in Table I.

D. YOLO

The YOLO network architecture (Fig 4.) is applied to the entire image at once by splitting it into a grid wherein n bounding boxes are selected and assigned with the probability and offset value. The predicted probability is compared with a pre-defined threshold value, whereupon a decision on the object presence within the image is made.

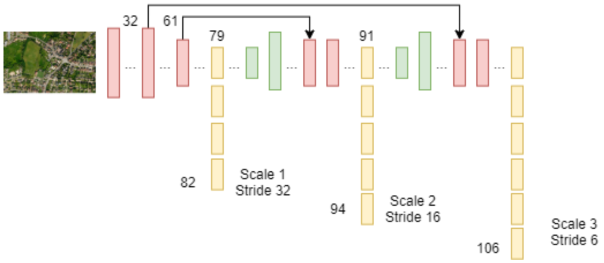

Fig. 4. Regional Proposal Network.

Based on the YOLO concept several versions of the network architecture have been developed. The versions differ from each other with respect to the used convolutional networks for classification, implementation of residual learning techniques, up-sampling and multi-core presentation of each grid’s sample set. For example, the YOLO v3 network applies detection 1×1 kernels to feature maps of three different sizes in several parts of the network.

E. Single Shot Detector (SSD)

Single Shot Detector (Fig. 5) consists of two shots: first one is intended to extract the feature maps whereas the second one applies convolutional filters to detect objects. To extract features and generate a feature map a VGG networks has been used. The latter contained four sets of convolutional layers with ReLu, where size of each next layer reduces by half in comparison to the previous one. Detection has been performed with a separate convolutional filter, which makes 4 predictions for each cell. Hereby, each filter output includes N + 4 channels N scores for each class and one boundary box.

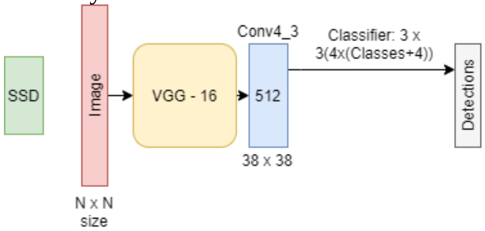

Fig. 5. Single Shot Detector (SSD).

The distinctive feature of the SSD network is the possibility for objects separation into classes in a single run using a given grid of windows on the image pyramid.

III. Results

A. Segmentation of aerospace images

DSTL dataset has been chosen as a basic set for image segmentation task since it partly includes the pre-marked images. Dataset consists of RGB, multispectral part (400-1040 nm) and a mid-infrared part (1195-2365 nm).

Training dataset out of 425 images includes 10 pre-labeled classes: trucks, cars, rivers, lakes, agricultural lands, roads, etc. Only 25 images from the whole dataset were pre-marked. Here the different classes had uneven distribution over the images, e.g. rivers and lakes totally cover over 20% of the area, agriculture lands – over 30%, cars – less than 0.5%. The data has been pre-processed including the normalization and resizing to ensure the same resolution of 1024×1024 px. For this, a custom ImresNet network, which consists of 16 layers with 32 neurons in the first hidden layer and 64 in the others, has been used

The architecture of four CNNs (U-Net, DeepLab, FullyConvolutional, GlobalConvolutional) have been optimized according to the following paraments: number of the feature maps, size of the convolution kernel, pooling type, dropout probabilities, and activation function. Then the networks have been trained by Adam optimization algorithm with learning speed of 10^{-3}, hyperparameters β_1=0.9, β_2=0.999; decomposition of 10^{-5}, and 30 epochs on 1000 frames batches [25].

Categorical cross entropy with modulation coefficient has been selected as a loss function to compensate the high classes imbalance. In order to evaluate the segmentation accuracy an intersection of real labels and predicted areas has been calculated according to Jacard Index Metrix. Dropout probabilities have been optimized for each network type in order to prevent over-learning and for regularization purposes. The corresponding optimized dropout values are represented in Table II.

TABLE I
DROPOUT PROBABILITY VALUE FOR IMPLEMENTED CNNS FOR SATELLITE IMAGES SEGMENTATION

Network architecture	Dropout probability value
U-Net	0.60
FullConv	0.25
DeepLab	0.25
GCNet	0.3, 0.2, 0.1, 0.05

The weighted Jacard indexes for image segmentation into 10 classes for both training and validation phases are presented in Table III. It has been determined that within the training phase the best result with the Jacard index of 0.684 is demonstrated by the DeepLab network, whereas the worst one is exhibited by the Fully Convolutional network. For the testing phase U-Net network with the Jacard index of 0.663 shows the most accurate result.

TABLE II
RESULTS OF SATELLITE IMAGE SEGMENTATION

Network architecture	Jacard index on testing set	Jacard index on training set	Number of weights
U-Net	0.663	0.672	10 234 124
FullConv	0.522	0.554	24 194 852
DeepLab	0.582	0.684	40 284 743
GCNet	0.647	0.679	8 296 163

The comparison of the networks according to the weightiness, that determines the possibility to implement these networks for real-time segmentation (Table III), demonstrates the benefit of the GCNet architecture. Generalizing the ability, speediness and learning efficiency, the U-Net network is determined as an optimal configuration.

B. *Object detection of aerospace images.*

"Stanford Campus", "DOTA v-1.5" and "xView 2018 election" have been selected as datasets to study the performance of the dNNs for objects detection on the aerial images. All datasets include the images captured with the same camera shooting angle, classes with high similarity of the captured pattern and limited size of the pre-labeled training dataset.

The "Stanford Campus" dataset consists of 8 unique scenes [see Table IV] and accounts in total 60 videos taken from UAVs with duration times from 2 to 11 min. The appearance frequency for objects in the scenes is represented in Table IV. Every 20th frame of the videos (in total 3000 images) containing 41800 objects has been put into the training set. The selected images have been resized to resolution of 1400×1100 px with ImresNet network that has been discussed in the segmentation part. In order to ensure generalization each video has been split in half into validation and testing parts (700 images and 7500 objects), with a 10% part added to the training part.

TABLE III
STRUCTURE OF THE STANFORD CAMPUS DATASET

№	Videos	Bikes	Pedestrian	Skateboarder	Cart	Car	Bus
1	9	51.94	43.36	2.55	0.29	1.08	0.78
2	4	56.04	42.46	0.67	0	0.17	0.67
3	12	4.22	64.02	0.60	0.40	29.5	1.25
4	4	18.89	80.61	0.17	0.17	0.17	0
5	7	32.89	63.94	1.63	0.34	0.83	0.37
6	5	56.30	33.13	2.33	3.10	4.71	0.42
7	4	12.50	87.50	0	0	0	0
8	15	27.68	70.01	1.29	0.43	0.50	0.09

The "Dota-v 1.5" dataset has been used in part (1927 images with resolution ranging from 800×800 to 4000×4000 px.). The data includes 127 241 instances representing 11 classes: baseball diamond, tennis court, basketball court, ground track field, soccer ball field, swimming pool, storage tank, roundabout, large vehicles, small vehicles, harbor and ships. Among them there are several groups out of two, three or four classes that have similar or slightly variant visual patterns.

In total 647 images with 0.3-meter resolution have been selected from the "xView" dataset. The final dataset contains 19 object classes: small car, truck, cargo truck, utility truck, bus, trailer, cargo car, dump truck, pickup truck, truck tractor, haul truck, cargo plane, aircraft hangar, fixed-wing aircraft, helicopter, tugboat, yacht, sailboat and container ship. The pre-processing includes random crop of the 416×416 px areas from the images, augmentation with image flips, shifts and random 45º rotation. Part of the images have few annotations (< 5) that sophisticates the training procedure.

The following architectures SSD, R-CNN, Fast R-CNN, Faster R-CNN, and YOLO v3 have been studied on effectiveness. For R-CNN generation already the half of the necessary proposal regions (1000) has been defined sufficient to reach the required accuracy which significantly accelerates the process of objects detection. Here the support vector machine is used as a classifier. In case of Faster R-CNN, the network has been modified by implementing the residual learning strategy within the ResNet model for classification tasks. This has been applied to address the rather low diversity of the detected objects in terms of feature depth. This optimization solution allowed to skip several items and thus to prevent overfitting of the network. The YOLO v3 network has been modified by adding the unsampled layers to extract small object's features, which are highly important in detecting objects with similar patterns. Within the YOLO v3 the Darknet 53 network has been implemented instead of Darknet 19 with the grid of 13×13 px size with 7 anchors. The size of the convolutional kernel has been decreased to 13 as default, so that the 13×13 region is employed to detect the large objects, 26×26 – the medium, and 56×56 – the small ones. For the SSD network the grid was set to [5 x 5], zoom level – to 1.0 and the ratio of 1:1 as this set of parameters was determined to demonstrate the best result.

Since the background of the images contains much more classes to be detected than the foreground has, the Focal Loss algorithm and advanced cross-entropy loss function with the adjusting parameter *y* have been applied. This allows to minimize risks of false detection, when patterns on the background contribute to the noise component within the training process. The performance of several mechanisms for weights optimization, and namely Adagrad, RMSprop, Adadelta, and Adam, have been compared. As the result, advanced version of stochastic gradient descent (SGD) Adam has been finally chosen due to the possibility to avoid a zero-shift at the initial moments. The following parameters for Adam method have been used: learning rate = 0.01, weight decay = 0.001, momentum = 0.7, *y* value = 4, α = 0.30.

The dNNs have been implemented in the Python on Intel Core i7 6700HQ and GPU NVIDIA Tesla K80 processing units using the TensorFlow, Pytorch, Numpy, Keras, and matplotlib libraries. During the training and validation phases the mean Average Precision (mAP) value has been calculated for all selected dNNs for all classes whereupon the weighted accuracy has been calculated (see Table V).

TABLE IV
RESULTS OF OBJECT DETECTION ON STANFORD CAMPUS DATASET

Algorithm	Frames per second	mAP test	mAP validation
Stanford Campus Dataset			
SSD	22.83	78.24	82.87
R-CNN	14.92	80.49	85.26
Fast R-CNN	13.23	81.18	83.23
Faster R-CNN	15.65	84.78	**86.12**
YOLO v3	**26.82**	**87.12**	85.51

For testing phase, the best result has been obtained for the YOLO v3 network with mAP = 87.12. The best training accuracy has been shown by Faster R-CNN network with mAP = 86.12. The given results on image processing speed in frames per second have been used to characterize the dNN effectiveness for real-time detection. It has been determined that the fastest network is YOLO v3 (fps= 26.82) whereas the slowest is Fast R-CNN (fps = 13.23).

Table VI represents the object detection results for Dota v1.5 and xView 2018 datasets. As previously mentioned, the Dota v1.5 and xView 2018 datasets contain only images, and thus the ability for real-time processing is not considered. Taking into account the rate of methods convergence, generalization ability and speediness, the best result on detection of the objects with similar patterns has been shown by YOLO v3 deep neural network.

TABLE V
RESULTS OF OBJECT DETECTION ON DOTA V1.5 AND XVIEW 2018 DATASETS

Algorithm	mAP test	mAP validation
Dota v1.5		
SSD	73.19	76.98
R-CNN	70.17	71.12
Fast R-CNN	74.02	75.64
Faster R-CNN	78.93	**79.92**
YOLO v3	**80.20.**	79.28
xView 2018 Detection		
SSD	71.52	75.63
R-CNN	75.91	74.39
Fast R-CNN	76.01	79.23
Faster R-CNN	77.33	79.58
YOLO v3	**78.29**	**79.92**

An example of the detected objects of two classes: pedestrians (black) and bikes (blue) defined by the YOLO v3-based network on the image from Stanford Campus Dataset is represented in Fig. 6a; an example of the detected objects of large vehicles (red) and small vehicles (blue) on the image from "Dota-v 1.5" is represented in Fig. 6b and example of the detected objects of cargo plane (red) and fixed-wing aircraft (blue) on the image from "xView" is represented in Fig. 6b

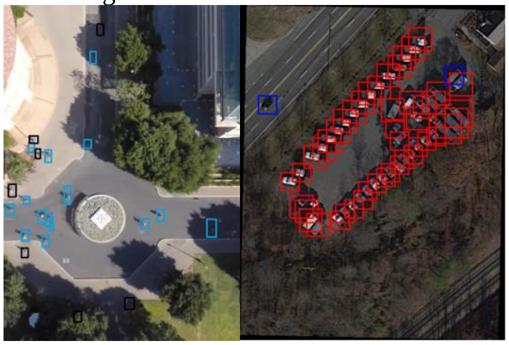
a) b)

Fig. 6. Examples of detected objects by the YOLO v3-based network on the images from a) Stanford Campus Dataset; b) Dota-v 1.5, of a) pedestrians (black) and bikes (blue); b) large vehicles (red) and small vehicles (blue); c) cargo plane (red) and fixed-wing aircraft (blue) classes.

This shows the possibility to implement the methods based on deep neural networks, especially YOLO v3-based network, within the on-board computer vision systems for unmanned aerial vehicles.

IV. CONCLUSION

The comparative study on the performance of different architectures of deep neural networks for both pattern recognition on aerospace images and object detection from unmanned aerial vehicles with limited amount of the pre-trained datasets has been performed.

In order to prevent the over-learning of the neural networks for image segmentation task the supplementary dropout layers with empirical optimization have been introduced. The shift and rotate augmentation procedures have been utilized to artificially increase the dataset. The best weighted average result in terms of speediness, learning efficiency, and generalization ability has been showed by the GCNet network with Jacard index equal to 0.647 in the testing phase and 0.679 in the training phase.

The accuracy of the region proposal network (R-CNN) together with its advanced versions, SSD and YOLO v3 architectures, for object detection in aerial imaging has been studied. The networks have been trained and optimized in terms of loss function, optimization algorithm, number of input layers and neurons, and accuracy metrics. The weighted mean Average Precision (mAP) has been selected as the performance criterion. On example of “Stanford Campus” dataset it has been defined that the YOLO ver. 3 network enables the best accuracy (mAP = 87.12) and the fastest analysis (26.82 frames per second) compared to the R-CNN, Fast R-CNN, Faster R-CNN, and SSD architectures. The tests with “Dota v1.5” and “xView 2018 Detection” have further confirmed the advantages of the YOLO ver. 3 network with the absolute accuracy values of 80.20 and 78.29 mAP, accordingly.

The obtained results in accuracy and, especially, in processing speed indicate clear feasibility for implementation of the proposed network based on YOLO ver. 3 architecture for an airborne device within the unmanned aerial vehicle for real-time object detection. Particularly, the pre-trained dNN launched on Nvidia Jetson Nano computer supplemented by the transmitter module would be capable for real-time transmission of images including the edges of the detected object and accuracy data. Moreover, the implementation of the dNN algorithm within the on-board system would allow to minimize the impact of the Air-to-Ground video transmission pipeline on the decision-making process. Functionalities of Nvidia Jetson such as Nano DeepStream SDK (streaming pipelines for AI-based video) and integration of various types of ML frameworks like Tensorflow and Pytorch would enable additional optimization of the dNN and their test on board.

Acknowledgment

This paper is an extended version of the work “Pattern recognition on aerospace images using deep neural networks” [25] with the same author list that has been presented at the IEEE International Workshop on Metrology for Aerospace in 2020.

References

[1] Haddal, Chad C., and Jeremiah Gertler, "Homeland security: Unmanned aerial vehicles and border surveillance," *CRS Report for Congress Washington*, pp. 3-4, 2010.

[2] M. A. Ma'sum et al., "Simulation of intelligent Unmanned Aerial Vehicle (UAV) For military surveillance," *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pp. 161-166, 2013.

[3] E. Frew et al., "Vision-based road-following using a small autonomous aircraft," *IEEE Aerospace Conference Proceedings*, Big Sky, MT, vol. 5, pp. 3006-3015, 2004.

[4] S. G. Fowers, D. Lee, B. J. Tippetts, K. D. Lillywhite, A. W. Dennis and J. K. Archibald, "Vision Aided Stabilization and the Development of a Quad-Rotor Micro UAV," *International Symposium on Computational Intelligence in Robotics and Automation*, pp. 143-148, 2007.

[5] Xu, G., Zhang, Y., Ji, S., Cheng, Y. and Tian, Y., “Research on computer vision-based for UAV autonomous landing on a ship,” *Pattern Recognition Letters*, vol. 30, no. 6, pp. 600-605, 2009.

[6] Chen, N., Chen, Y., You, Y., Ling, H., Liang, P. and Zimmermann, R., “Dynamic urban surveillance video stream processing using fog computing,” in *Second International Conference on Multimedia Big Data (BigMM)*, pp. 105-112, 2016.

[7] Honkavaara, E., Saari, H., Kaivosoja, J., Pölönen, I., Hakala, T., Litkey, P., Mäkynen, J. and Pesonen, L., “Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight UAV spectral camera for precision agriculture,” *Remote Sensing*, vol. 5, no. 10, pp.5006-5039, 2013.

[8] Saari, H., Pellikka, I., Pesonen, L., Tuominen, S., Heikkilä, J., Holmlund, C., Mäkynen, J., Ojala, K. and Antila, T., “Unmanned Aerial Vehicle (UAV) operated spectral camera system for forest and agriculture applications,” *Remote Sensing for Agriculture, Ecosystems, and Hydrology XIII*, vol. 8174, no. 1, p. 81, 2011.

[9] Rovira-Más, F., Zhang, Q. and Reid, J.F., “Stereo vision three-dimensional terrain maps for precision agriculture,” *Computers and Electronics in Agriculture*, vol. 60, no. 2, pp. 133-143, 2008.

[10] Markuš, Nenad & Frljak, Miroslav & Pandžić, Igor & Ahlberg, Jörgen & Forchheimer, Rober, “A method for object detection based on pixel intensity comparisons,” *arXiv:1305.4537v3*, 2013.

[11] S. Han, W. Shen, and Z. Liu, “Deep Drone, “Object Detection and Tracking for Smart Drones on Embedded System,” *Stanford University*, pp. 1–8, 2016.

[12] S. U. Nisa, M. Imran, "A Critical Review of Object Detection using Convolution Neural Network," *2nd International Conference on Communication, Computing and Digital systems (C-CODE)*, pp. 154-159, 2019.

[13] Ke Li, Gang Wan, Gong Cheng, Liqiu Meng, Junwei Han, "Object detection in optical remote sensing images: A survey and a new benchmark,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 159, pp. 296-307, 2020.

[14] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587, 2014.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, “Microsoft coco: Common objects in context,” *European Conference on Computer Vision*, Springer, vol 8693, no. 1, 2014.

[16] Dumitru Erhan, Christian Szegedy, Alexander Toshev, Dragomir Anguelov, “Scalable Object Detection using Deep Neural Networks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2147-2154, 2014.

[17] Andrii Hlavcheva, Daria Kuchuk, Heorhii Yaloveha “Application of Deep Learning in the Processing of the Aerospace System's Multispectral Images” in *Handbook of Research on Artificial Intelligence Applications in the Aviation and Aerospace Industries*, pp.134 – 147, 2019.

[18] W. Luo, B. Yang and R. Urtasun, "Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net," *Conference on Computer Vision and Pattern Recognition*, pp. 3569-3577, 2018.

[19] R. Hänsch, S. Kaiser, and O. Helwich, “Near Real-time Object Detection in RGBD Data,” *International Conference on Computer Vision Theory and Applications*, pp. 179–186, 2017.

[20] G. Cheng, J. Han, P. Zhou, D. Xu, etc., “Learning Rotation-Invariant and Fisher Discriminative Convolutional Neural Networks for Object Detection,” *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 265-278, 2019.

[21] M. Radovic, O. Adarkwa, and Q. Wang, “Object Recognition in Aerial Images Using Convolutional Neural Networks,” *J. Imaging*, vol. 3, no. 4, p. 21, 2017.

[22] K. Li, G. Cheng, S. Bu and X. You, "Rotation-Insensitive and Context-Augmented Object Detection in Remote Sensing Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 4, pp. 2337-2348, 2018.

[23] G. Cheng, P. Zhou, J. Han, etc., “Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images,”. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7405-7415, 2016.

[24] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, Xindong Wu, “Object Detection with Deep Learning: A Review,” *Transaction on neural networks and learning systems*”, pp. 1-21, 2019.

[25] I. Saetchnikov, V. Skakun and E. Tcherniavskaia, “Pattern recognition on aerospace images using deep neural networks” in 2020 IEEE 7th International Workshop, 2020, pp. 336-340.

Ivan V. Saetchnikov graduated with honors from the Belarusian State University in 2015. His research is focused on deep neural networks, computer vision and aerospace image processing.

Elina A. Tcherniavskaia got a doctor degree in 1991 from the Moscow State University. Currently, she is a professor at the chair of Nuclear physics at the Belarusian State University. Among her research interests are optoelectronics, nanophotonics, bioinformatics, machine learning algorithms, neural networks, methods for data processing, models and methods for diagnostics of complex heterogeneous systems.

Victor V. Skakun defended his PhD from the Belarusian State University in 2009. Since 2015, he is a head of the department of System Analysis and Computer Modelling. His research interests include models, methods, algorithms for fluorescence spectroscopy data analysis; development of photon counting detectors; database design and development.