AEROSPACE SYSTEMS FOR MONITORING AND CONTROL

UDC 004.032.26.93'12 (045)
DOI 10.18372/2306-1472.76.13149

Volodymyr Kharchenko[1]
Iurii Chyrka[2]

AIRPLANES DETECTION IN AERIAL IMAGES USING YOLO NEURAL NETWORK

National Aviation University
Kosmonavta Komarova avenue 1, Kyiv, 03680, Ukraine
E-mails: [1]knarch@nau.edu.ua; [2]yurasyk88@ukr.net

Abstract

Purpose: *The represented research results are aimed to benchmark performance of state-of-the-art methods of objects detection. There were tested two popular single-stage neural networks based on the "you only looks once" approach.* ***Methods:*** *convolutional neural network, logistic regression, probabilistic theory, stochastic gradient descent.* ***Results:*** *The considered artificial neural network architectures for objects detection has been trained and applied for the particular task of the airplanes detection in aerial images taken from unmanned aerial vehicles and satellites.* ***Discussion:*** *Presented results of experimental verification prove their high detection ability, location precision and real-time processing speed using modern graphics processing unit. The considered neural networks can be easily re-trained for detection of different classes of ground objects.*

Keywords: Convolutional neural network; object detection; real-time processing; unmanned aerial vehicle

1. Introduction

Unmanned aerial vehicles (UAVs) are an important part of many fields of human activity. UAVs can be either remote controlled aircraft (by a pilot at a ground) or can fly autonomously based on some flight program or controlled by a higher-level control system.

UAV technology has many advantages that include low cost, small size, safety, ecological operation, and most of all, the fast and on-demand acquisition of images [1]. Most of all, they are an effective and powerful method of capturing high resolution remote sensing (RS) images [1-3]. Generally speaking, UAVs can be considered as a part of many socio-technical system [4]. The recent rapid advances of UAV technology led to many studies proposing many novel ways for UAV applications and image analysis in relation to corresponding areas including infrastructure surveillance, fire detection, vegetation monitoring, marine surface monitoring, nature changes observation, disasters management, traffic monitoring etc [3,5–8]. Most of them can be generally described as detection, recognition and tracking of various objects of interest.

One of the problems currently facing autonomous UAV operation is performing of mentioned operations in real-time on the basis of the video sequence fed by the attached camera. To solve this problem, modern fast and accurate detection methods must be used.

2. Analysis of the research and publications

Object detection is one of fundamental tasks in computer vision, and refers to the determination of the presence or absence of specific features in image [9]. When features are detected, an object can be further classified as belonging to one of a pre-defined set of classes and then the bounding box around that object or object central point is predicted.

There are three main groups of object detectors: classic, two-stage and one-stage ones. Classic object detectors operate in the sliding window, in which a classifier is applied every time over a predefined image grid. The most known of them are convolutional neural networks for digits recognition, proposed by of LeCun et al [10]. Viola and Jones face detector [11] and the histogram of oriented gradients (HOG) method for pedestrian detection [12]. Later with development of deep learning they have been outperformed by two-stage detectors, described next.

More recent approaches use region proposal methods to first generate a sparse set of candidate proposals that should contain all objects while filtering out the majority of negative locations in an image and then run a classifier on these proposals in order to separate them into foreground classes/background. Such two-stage detection is the dominant paradigm nowadays. Region-based convolutional neural networks (R-CNN) have grown over last years with several improvements [13-15] and numerous extensions [16 - 18].

Yet another modern approach to object detection assumes a single detection stage that is closer to the natural way of objects detection by humans. The three main methods can be mentioned here: SSD [19], YOLO ("You Only Look Once") [20, 21] and RetinaNet [22]. Their effectiveness roughly varies in the same order. SSD has about 10-20% less average precision. YOLO (v3) has almost the same precision as two-stage detectors and RetinaNet so far is the state of the art object detector. However, YOLO v3 is the fastest among them and still has acceptable precision. That is an important factor for autonomous UAVs as it will be described next.

Coming back to UAVs operations, the self-controlled flying process can be divided into three stages. First, raw data is recorded during flight via sensors which an UAV is equipped with. Then the real-time data processing is performed by the on-board intelligence system. The final stage supposes autonomous decision-making based on the processed data. All stages should be conducted in a few milliseconds. The crucial part here is the second stage, where the on-board system is supposed to detect and classify surrounding objects in real time.

In this situation, solution comes with usage of single-stage detectors based on CNN. It worth to mention, that one of the superior features of CNNs is their parallel nature that perfectly fits the architecture of a graphical processing unit (GPU), which consists of thousands of cores designed to handle multiple tasks simultaneously. Their combination allows dramatic reduction of computation time while maintaining superb precision.

In view of recent advances in GPU hardware development, price and size of the GPU units have been reduced considerably. This allows to design an integrated software–hardware module capable of real-time processing, which is light and inexpensive enough to be mounted on an UAV. However, before such incorporation, CNN need to be trained and tested on the more powerful equipment.

3. Aim of the paper

In this paper, we consider usage of modern CNN architectures for the detection and classification of objects during autonomous UAV operations in civil applications. That paper shows the example of successful application of YOLO and Tiny YOLO architectures application on real-time airplanes detection on the ground from the video feed during UAV operation test.

4. YOLO neural network

The CNN algorithm considered in this paper has been built on an open-source platform complied from the Darknet framework written in C and CUDA [23] that has an implementation of the YOLO architecture of the 3rd version [21] and it's simplified version from the original paper [20].

The main advantage of YOLO as a single-stage approach is that the single neural network evaluates the whole image. It makes all predictions based on the actual image, not the proposed regions as it goes for two-stage methods. The input image is represented as a tensor of size $n \times m \times 3$, where n and m represent width and height in pixel and 3 denotes 3 color channels). All input images of various sizes are automatically resized to 416×416; therefore, we used a $416 \times 416 \times 3$ input tensor every time for training. Actually, that size can vary in certain range but those particular values give the output feature map with odd number of cells with a single central cell as it will be described later.

The network uses the backbone Darknet-53 that is a 53-layer feature extracting deep neural network. Its structure is shown in the Table 1. Faster or Tiny YOLO architecture has a simplified structure with much less amount of layers, as it is shown in the Table 2.

Table 1

Darknet-53 structure

	Type	Filters	Size	Output
	Convolutional	32	3×3	416×416
	Convolutional	64	3×3/2	208×208
1×	Convolutional	32	1×1	
	Convolutional	64	3×3	
	Residual			208×208
	Convolutional	128	3×3/2	104×104
2×	Convolutional	64	1×1	
	Convolutional	128	3×3	
	Residual			104×104

	Convolutional	256	3×3/2	52×52
	Convolutional	128	1×1	
8×	Convolutional	256	3×3	
	Residual			52×52
	Convolutional	512	3×3/2	26×26
	Convolutional	256	1×1	
8×	Convolutional	512	3×3	
	Residual			26×26
	Convolutional	1024	3×3/2	13×13
	Convolutional	512	1×1	
4×	Convolutional	1024	3×3	
	Residual			13×13

Table 2

Tiny YOLO feature extractor structure

Type	Filters	Size	Output
Convolutional	16	3×3/2	208×208
Convolutional	32	3×3/2	104×104
Convolutional	64	3×3/2	52×52
Convolutional	128	3×3/2	26×26
Convolutional	256	3×3/2	13×13
Convolutional	512	3×3/1	13×13

The logic on the network suppose that the picture is divided onto the grid of equal cells of some size. For 416 × 416 input image, it is a grid of 13 × 13 cells. Each cell is responsible for prediction of a certain amount of bounding boxes that covers this cell.

Each prediction of a bounding box contains the following information: b_x and b_y coordinates of the bounding box, width b_w and height b_h, that are calculated through predictions t_x, t_y, t_w, t_h from cell coordinates c_x, c_y and bounding box prior width p_w and height p_h as

$$
\begin{aligned}
b_x &= \sigma(t_x) + c_x, \\
b_y &= \sigma(t_y) + c_y, \\
b_w &= p_w e^{t_w}, \\
b_h &= p_h e^{t_h}.
\end{aligned} \tag{1}
$$

Besides coordinates, it predicts the probability that the bounding box contains the object of interest and C conditional probabilities of belonging to each of predefined C object classes. We suppose that each cell can predict up to 3 bounding boxes. Therefore, the final output tensor is 13 × 13 × [3* (5 + C)].

Additionally YOLOv3 predicts boxes at 3 different scales for better detection of small objects. After prediction at the first scale, the feature map from 2 layers previous is taken and upsampled by 2×. After that, a feature map from earlier in the network is taken and merged with upsampled features using element-wise addition. A few more convolutional layers are added to process this combined feature map make a new tensor with predictions, but now it is of size 26 × 26 × [3* (5 + C)]. At the final scale all the same operations are performed once more. Thus predictions for the 3rd scale benefit from all the prior computation as well as fine-grained features from early on in the network. [21]

Finally, duplicate detections are eliminated by non-maximal suppression.

5. CNN training

Although, there are a few YOLO networks trained on several known datasets, the CNN still needs to be trained for better precision when for work with objects specific for our particular tasks. A few task-specific parameters such as batch size, learning rate, decay, iteration number, and detection thresholds should be additionally tuned for the best performance. The number of epochs required for training was determined empirically. “Epoch” means a single run through the entire data set during training of a neural network. For batch training, the input data are fed to the network within batches that includes a fixed number of samples (called as a “batch size”) and weights are updates every time it pass through the learning algorithm. “Learning rate” is a constant used to control the rate of learning or gradient descent. “Decay” refers to the ratio for decreasing learning rate at certain number of epochs.

Our networks (full and tiny) have been trained for only a single object class i.e. “airplane” (C = 1). That gives us a size of the first scale output tensor as 13 × 13 × 18.

Preliminary analysis have shown that images with airplanes taken by UAVs differ significantly from the images available at common databases with a lot of objects. As example, many photos from the PASCAL VOC database are taken from the frontal or side view, while the images from UAV - mostly from the top-down view. That promises a significant gain in performance between a network trained on those databases and the one trained on a custom database containing satellite and UAV-acquired images. For the data augmentation during network training, there were used a few transformations: random scaling up to 60% change of the original image size, changes of hue, saturation and exposure of the image were randomized in the range 0.1, 1.5, 1.5 respectively in the hue saturation value (HSV) model.

The custom database of images with objects of the class "airplanes" was created by capturing satellite images of airplanes grounded on civil and military airfields mostly across Europe from Google Maps. Additionally there were downloaded several photos taken from the air (mostly with not right angles, that better suits to actual operating conditions of low-altitude flying UAV) by browsing them in the Internet. Images from Internet were used due to current restrictions on operating UAVs in airfield proximity.

These images consisted of a variety of airplane types, shapes and color schemes, with a wide range of image scales, resolutions, and compositions. For example, images were selected in a way to show airplanes as close as possible (when an airplane covers almost the whole image) and from large distances (when an airplane is a small spot on the image, but still recognizable by its shape). Images quality varies from HD to Full HD and size of airplane figures on them varies from about ten pixels wide to hundreds pixels. The most of images contains more than one airplane, sometimes with overlapped bounding boxes for tightly placed airplanes. All airplanes have been fully places within frame boundaries. Cases, where only a part of the airplane is present on the image have been excluded from the training set or have not marked as airplanes. The rest simply have a single airplane. All images contains photos in daylight conditions. Image quality also varies in a certain region from high-quality clean images to noisy and distorted by compression artefact images. The training set consists of 204 images containing 1245 airplane objects in total.

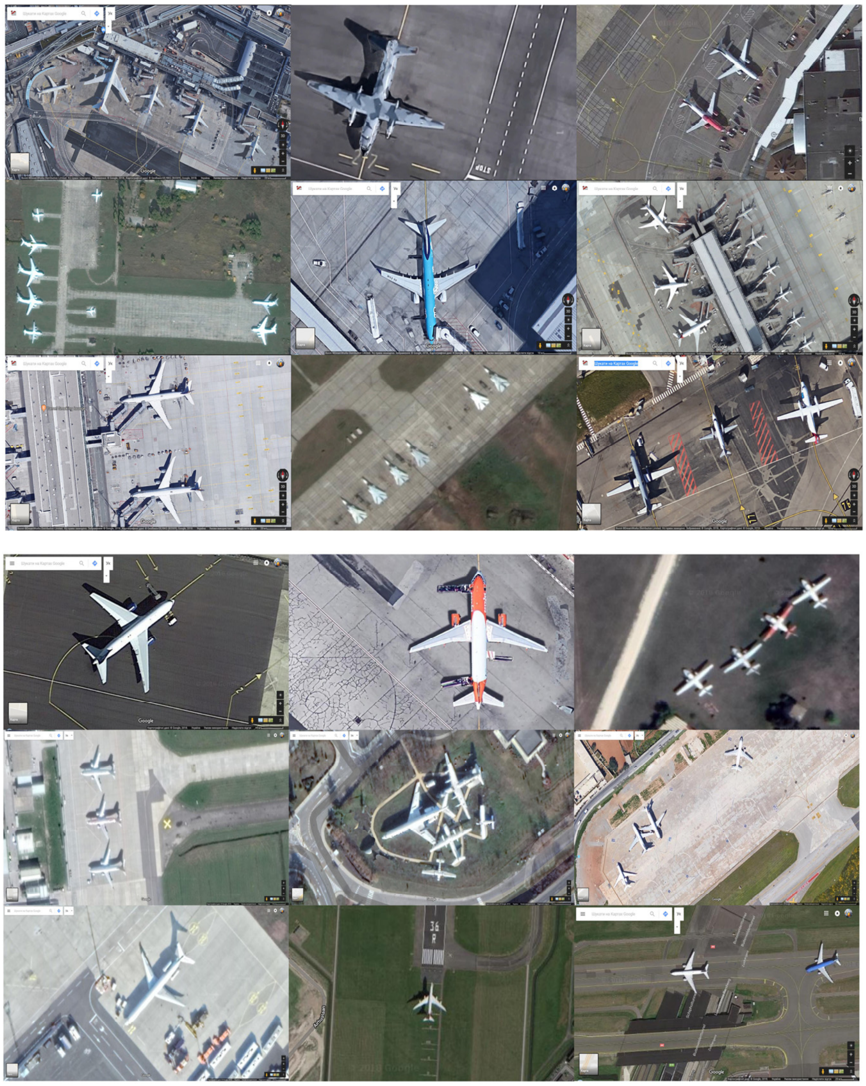

Fig. 1. Training and validation sets of pictures with "airplanes".

The open-source tool known as YOLO mark [24] was used to label all airplane instances in the dataset and specify ground truth bounding boxes. Training has been carried out with the next parameters, batch size 64, momentum = 0.5, decay = 0.0005, base learning rate = 0.001, and maximum iteration number = 50000. Learning rate has been changed during training in the next way: for YOLOv3 100% of base on steps 0-39999, 10% on steps 40000-44999, 1% on steps 45000-49999; for tiny YOLO 10% of base on steps 0-99, 100% on steps 100-19999, 10% on steps 20000-29999, 1% on steps 30000-49999.

6. Experimental results

For validation of the trained neural networks a new data set of 50 images similar to the ones in the training set and containing a total of 203 airplanes was used. Both trained neural networks have been tested with different sizes of an input image from 224 × 224 to 608 × 608 with the step 32 pixels. As far as the network is convolutional, the input tensor for it can be different sizes with a small restriction to be divided by 32. Such variability is an additional tool in finding a reasonable trade-off between desired speed and accuracy. Table 3 shows results for various cases for both YOLOv3 and tiny YOLO networks. There were carried out preliminary tests in order to avoid overfitting of the model. The best results (in terms of average precision) on the validation test have been obtained with the weights after 34000 iterations for YOLO v3 and after 29300 iterations for tiny YOLO.

Table 3

YOLOv3 and Tiny YOLO performance

Size	mAP	IoU	FPS
YOLOv3			
608 × 608	90.91%	85.44%	13
416 × 416	90.73%	84.56%	23
320 × 320	90.64%	81.83%	34
224 × 224	81.72%	78.32%	51
Tiny YOLO			
608 × 608	90.46%	72.19%	65
416 × 416	80.77%	66.26%	100
320 × 320	71.16%	59.74%	120
224 × 224	47.75%	52.65%	136

The table contains the next information: a set of sizes of input tensor; mAP is a mean average precision that means an averaged value on precision/recall curve calculated over 11 points [0, 0.1, …, 1]; IoU is an intersection over union that is the ratio of an intersection area of a ground true bounding box and a predicted bounding box to area of union of these boxes; FPS means processing speed frames per second obtained with our low-performance GPU NVIDIA 1060 6Gb. Those parameters can be interpreted in the next way: mAP is an indicator of detection effectiveness, IoU is an indicator of bounding boxes positioning precision, FPS is a common performance and speed indicator.

One can see that YOLOv3 has quite high precision level that is achievable even for a typical 25fps video. It has slight dependence on the input tensor size. However, the smallest acceptable size 224 × 224 has significant drop in probability of detection and can be used only for fast processing. Tiny YOLO generally is much faster but way less precise, especially in terms of IoU. When with the biggest input size it can achieve a detection level comparable with YOLOv3, precision of objects location (IoU) is still quite poor.

It should be mentioned, that the trained YOLO v3 CNN was able to detect an airplane in the image, even if its contours were obscured by another object, for example, a tower on the ground, or in pretty different conditions, for example photos of airplanes in the air taken from the bottom, but size in pixels of the airplane must be relatively big. If an airplane is not fully shown in the image it recognize it only when most of it is present, that has been expected.

On the other hand, when size of the airplane image is distorted by compression artifacts, or when it has a livery or a shape not present in the training set it usually is missed by the network, especially at low size of the airplane in the picture. Obviously, that can be overcome with more thorough database, carefully selected, graded and cleaned of repetitive examples.

7. Conclusions

Presented results have shown the high level of detection and classification accuracy of YOLO architecture in the particular application of airplanes detection, even with relatively small training database. That approach have been tested using UAVs of National aviation university and can be applied for detection of variety of ground objects, moreover, for multiple classes detection simultaneously.

The YOLOv3 and tiny YOLO CNN have shown average precision 90.91% and 90.46% at most on our database. Both have performance level that is enough for real-time detection on relatively low-performance GPUs.

References

[1] Ammour N., Alhichri H., Bazi Y., Benjdira B., Alajlan N. and Zuair M. (2017) Deep Learning Approach for Car Detection in UAV Imagery. *Remote Sensing*, No. 9(312), pp. 1–15. doi: 10.3390/rs9040312

[2] Maria G., Baccaglini E., Brevi D., Gavelli M., Scopigno R. (2016) A drone-based image processing system for car detection in a smart transport infrastructure. Proc. 18th Mediterranean Electrotechnical Conf. (MELECON). Limassol, Cyprus. doi: 10.1109/MELCON.2016.7495454

[3] C. Castiblanco, J. Rodriguez, I. Mondragon, C. Parra, and J. Colorado Air Drones for Explosive Landmines Detection. in *ROBOT2013: First Iberian Robotics Conference* (M. A. Armada, A. Sanfeliu, M. Ferre Eds.) Springer International Publishing, pp. 107–114. doi: 10.1007/978-3-319-03653-3_9

[4] Kharchenko V., Shmelova T., Sikirda, Y. (2012) Pryynyattya rishen' operatorom aeronavihatsiynoyi systemy. Monohrafiya [DecisionMaking of Operator in Air Navigation System.Monograph]. Kirovograd, KFA of NAU Publ., 292 p. (In Ukrainian)

[5] Xu Y., Yu G., Wang Y., Wu X., and Ma Y. (2017) Car Detection from Low-Altitude UAV Imagery with the Faster R-CNN. *Journal of Advanced Transportation*, Vol. 2017. doi: 10.1155/2017/2823617

[6] Lee J., Wang J., Crandall D., Sabanovic S. and Fox G. (2017) Real-Time Object Detection for Unmanned Aerial Vehicles based on Cloud-based Convolutional Neural Networks. Proc. IEEE International Conference on Robotic Computing (IRC). Taichung, Taiwan. doi: 10.1109/IRC.2017.77

[7] Leira F. S., Johansen T. A., Fossen T. I. (2015) Automatic Detection, Classification and Tracking of Objects in the Ocean Surface from UAVs Using a Thermal Camera. Proc. IEEE Aerospace Conference. Big Sky, MT, USA. doi: 10.1109/AERO.2015.7119238

[8] Lee J.-N., Kwak K.-C. (2014) A Trends Analysis of Image Processing in Unmanned Aerial Vehicle. *International Journal of Computer and Information Engineering*, No. 2(8), pp. 271–274. doi: 10.1999/1307-6892/9997348

[9] Dhawad S. R., Itkarkar R. R. (2016) Car Detecting Method using high Resolution images. *International Journal on Recent and Innovation Trends in Computing and Communication*, No. 2(4), pp. 197–203.

[10] LeCun Y., Jackel L., Bottou L., Brunot A., Cortes C., Denker J., Drucker H., Guyon I., Müller U., Säckinger E., Simard P., Vapnik V. (1995) Comparison of learning algorithms for handwritten digits recognition. Proc. 1995 Int. Conf. on Artificial Neural Networks. Paris, France, pp. 53-60.

[11] Viola P. and Jones M. (2001) Rapid object detection using a boosted cascade of simple features. Proc. Computer Vision and Pattern Recognition (CVPR-2001). Kauai, HI, USA, pp. I-511 – I-518. doi: 10.1109/CVPR.2001.990517

[12] Dalal N. and Triggs B. (2005) Histograms of oriented gradients for human detection. Proc. Computer Vision and Pattern Recognition (CVPR-2005). San Diego, CA, USA, Vol. 1, pp. 886–891. doi: 10.1109/CVPR.2005.177

[13] Girshick R., Donahue J., Darrell T., and Malik J. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. Proc. Computer Vision and Pattern Recognition (CVPR-2014). Columbus, OH, USA, pp. 580–587. doi: 10.1109/CVPR.2014.81

[14] Girshick R. (2015). Fast R-CNN. Proc. Computer Vision (ICCV-2015). Santiago, Chile, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.

[15] Ren S., He K., Girshick R., Sun J. (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, No. 6(39), pp. 1137–1149. doi: 10.1109/TPAMI.2016.2577031

[16] Lin T.-Y., Doll´ar P., Girshick R., He K., Hariharan B., and Belongie S. (2017) Feature pyramid networks for object detection. Proc. Computer Vision and Pattern Recognition (CVPR-2017). Honolulu, HI, USA, pp. 936–944. doi: 10.1109/CVPR.2017.106

[17] He K., Gkioxari G., Doll´ar P., and Girshick R. (2017) Mask R-CNN. Proc. Computer Vision (ICCV-2017). Venice, Italy, pp. 2980–2988. doi: 10.1109/ICCV.2017.322

[18] He K., Zhang X., Ren S., and Sun J. (2016) Deep residual learning for image recognition. Proc. Computer Vision and Pattern Recognition (CVPR-2016). Las Vegas, NV, USA, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[19] Liu W., Anguelov D., Erhan D., Szegedy C., and Reed S. (2016) SSD: Single shot multibox detector. Proc. European Conference on Computer vision Computer Vision and Pattern Recognition

(ECCV-2016). Amsterdam, The Netherlands. doi: 10.1007/978-3-319-46448-0_2

[20] Redmon J., Divvala S., Girshick R., and Farhadi A. (2016) You only look once: Unified, real-time object detection. Proc. Computer Vision and Pattern Recognition (CVPR-2016). Las Vegas, NV, USA, pp. 779–788. doi: 10.1109/CVPR.2016.91

[21] Redmon J. and Farhadi A. YOLOv3: An Incremental Improvement. Available at: https://pjreddie.com/media/files/papers/YOLOv3.pdf (Accessed 06.04.18)

[22] Lin T.-Y., Goyal P., Girshick R., He K., Doll´ar P. (2017) Focal Loss for Dense Object Detection. Proc. Computer Vision (ICCV-2017). Venice, Italy, pp. 2999–3007. doi: 10.1109/ICCV.2017.324

[23] Darknet: Open source neural networks in c. Available at: https://github.com/AlexeyAB/darknet (Accessed 05.04.18)

[24] YoloMark. Available at: https://github.com/AlexeyAB/Yolo_mark (Accessed 05.04.18)

В.П. Харченко[1], Ю.Д. Чирка[2]

Виявлення літаків на зображеннях з повітря з використанням нейронної мережі YOLO

Національний авіаційний університет, просп. Космонавта Комарова, 1, Київ, 03680, Україна
E-mails: [1]kharch@nau.edu.ua; [2]yurasyk88@ukr.net

Мета: Представлені результати дослідження спрямовані на тестування ефективності найсучасніших методів виявлення об'єктів. Було перевірено дві популярні одноступеневі нейронні мережі, що базуються на підході "ви дивитеся лише один раз". **Методи дослідження:** згорткова нейронна мережа, логістична регресія, імовірнісна теорія, стохастичний градієнтний спуск. **Результати:** Розглянуті архітектури штучних нейронних мереж для виявлення об'єктів були навчені та застосовані для конкретного завдання виявлення літальних апаратів на зображеннях з повітря, знятих з безпілотних літальних апаратів та супутників. **Обговорення:** Представлені результати експериментальної перевірки підтверджують високу здатність цих методів до виявлення, їх високу точність визначення місцезнаходження та швидкість обробки в реальному часі за допомогою сучасного графічного процесора. Розглянуті нейронні мережі можуть бути легко перенавчені для виявлення різних класів наземних об'єктів.

Ключові слова: безпілотний літальний апарат; виявлення об’єктів; згорткова нейронна мережа; обробка в реальному часі

В.П. Харченко[1], Ю.Д. Чирка[2]

Обнаружение самолетов на изображениях с воздуха с использованием нейронной сети YOLO

Национальный авиационный университет, просп. Космонавта Комарова, 1, Киев, 03680, Украина
E-mails: [1]kharch@nau.edu.ua; [2]yurasyk88@ukr.net

Цель: Представленные результаты исследования направлены на тестирование эффективности современных методов обнаружения объектов. Было проверено две популярные одношаговые нейронные сети, основанные на подходе "вы смотрите только один раз». **Методы исследования:** сверточная нейронная сеть, логистическая регрессия, вероятностная теория, стохастический градиентный спуск. **Результаты:** Рассмотренные архитектуры искусственных нейронных сетей для обнаружения объектов были обучены и применены для конкретной задачи обнаружения летательных аппаратов на изображениях с воздуха, снятых с беспилотных летательных аппаратов и спутников. **Обсуждение:** Представлены результаты экспериментальной проверки подтверждают высокую способность этих методов к обнаружению, их высокую точность определения местоположения и скорость обработки в реальном времени с помощью современного графического процессора. Рассмотрены нейронные сети могут быть легко переобучены для обнаружения различных классов наземных объектов.

Ключевые слова: беспилотный летательный аппарат; обнаружение объектов; обработка в реальном времени; сверточная нейронная сеть.

Kharchenko Volodymyr. Doctor of Engineering Sciences. Professor.
Vice-Rector on Scientific Work of the National Aviation University, Kyiv, Ukraine.
Winner of the State Prize of Ukraine in Science and Technology, Honorable Worker of Science and Technology of Ukraine.
Education: Kyiv Institute of Civil Aviation Engineers, Kyiv, Ukraine.
Research area: management of complex socio-technical systems, air navigation systems and automatic decision-making systems aimed at avoidance conflict situations, space information technology design, air navigation services in Ukraine provided by CNS/ATM systems.
Publications: 550.
E-mail: kharch@nau.edu.ua

Chyrka Iurii (1988). Candidate of Engineering Sciences. Senior researcher.
National Aviation University.
Education: National Aviation University, Kyiv, Ukraine (2011).
Research area: computer vision, machine learning, control systems, radar signals processing, acoustic holography, and applied statistics.
Publications: 45.
E-mail: yurasyk88@ukr.net