ВЕСТНИК ТОМСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
2024 Управление, вычислительная техника и информатика № 69
Tomsk State University Journal of Control and Computer Science

Original article
UDC 004.93'12
doi: 10.17223/19988605/69/8

Detection of flying objects in images using the YOLOv4-CSP convolutional neural network model

Stepan G. Nebaba[1], Nikolay G. Markov[2]

[1, 2] National Research Tomsk Polytechnic University, Tomsk, Russian Federation
[1] stepanlfx@tpu.ru
[2] markovng@tpu.ru

Abstract. The effectiveness of the YOLOv4-CSP convolutional neural network model in solving the problem of detecting objects moving in airspace is investigated. Images of flying objects of two classes were used as initial data for training and researching the convolutional neural network model: helicopter-type and aircraft-type unmanned aerial vehicles. Images of such objects were obtained in the optical and infrared wavelength ranges. Two datasets were formed from appropriately labeled source images with objects of these two classes. The first dataset was created from optical images, and the second from images obtained in the infrared wavelength range. The YOLOv4-CSP model was trained using training and validation samples from each dataset. Comprehensive studies of the effectiveness of the trained model were carried out using test samples from datasets. It is shown that the accuracy of detecting flying objects in optical images is higher than in images obtained in the infrared range, and the results for the speed of model calculation when analyzing optical and infrared images are close. Recommendations are given for the use of the YOLOv4-CSP model in computer vision systems for airspace monitoring.

Keywords: computer vision system; convolutional neural network YOLOv4-CSP; detection of flying objects; helicopter-type unmanned aerial vehicle; aircraft-type unmanned aerial vehicle.

***For citation*:** Nebaba, S.G., Markov, N.G. (2024) Detection of flying objects in images using the YOLOv4-CSP convolutional neural network model. *Vestnik Tomskogo gosudarstvennogo universiteta. Upravlenie, vychislitelnaja tehnika i informatika – Tomsk State University Journal of Control and Computer Science*. 69. pp. 72–81. doi: 10.17223/19988605/69/8

Научная статья
doi: 10.17223/19988605/69/8

Детектирование летающих объектов на изображениях с помощью модели сверточной нейронной сети YOLOv4-CSP

Степан Геннадьевич Небаба[1], Николай Григорьевич Марков[2]

[1, 2] Национальный исследовательский Томский политехнический университет, Томск, Россия
[1] stepanlfx@tpu.ru
[2] markovng@tpu.ru

Аннотация. Исследуется эффективность модели сверточной нейронной сети YOLOv4-CSP при решении задачи детектирования в воздушном пространстве объектов двух классов: беспилотных летательных аппаратов вертолетного типа и самолетного типа. Изображения объектов получены в оптическом и инфракрасном диапазонах длин волн, из них сформировано два соответствующих датасета. Модель YOLOv4-CSP обучена с использованием обучающей и валидационной выборок из каждого датасета. Проведены комплексные исследования эффективности обученной модели с использованием тестовых выборок из датасетов. Показано, что точность детектирования летающих объектов на оптических изображениях выше, чем на изображениях, полученных в инфракрасном диапазоне, а результаты по скорости вычисления модели при анализе оптических и инфракрасных изображений близки. Даны рекомендации по использованию модели YOLOv4-CSP в системах компьютерного зрения для мониторинга воздушного пространства.

Ключевые слова: система компьютерного зрения; сверточная нейронная сеть YOLOv4-CSP; детектирование летающих объектов; беспилотный летательный аппарат вертолетного типа; беспилотный летательный аппарат самолетного типа.

Для цитирования: Небаба С.Г., Марков Н.Г. Детектирование летающих объектов на изображениях с помощью модели сверточной нейронной сети YOLOv4-CSP // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. 2024. № 69. С. 72–81. doi: 10.17223/19988605/69/8

Introduction

Today, research and development of modern computer vision systems (CVS) are intensively conducted all over the world. Such systems are in demand for solving many applied problems in a variety of areas of human activity, such as controlling autonomous vehicles, monitoring the safety of enterprises and hazardous production facilities [1], processing and analyzing medical images, and in a number of other areas of knowledge [2]. The problem of airspace safety above enterprises with hazardous technological objects and processes deserves special attention. To solve this problem, radar systems for detecting flying objects (FO) are widely used. FO usually refers to various manned and unmanned aerial vehicles, birds, etc. In recent years, monitoring of airspace for the purpose of detecting FO has begun to be carried out using various CVS based on deep learning methods and models. Such CVS can analyze images of FO in airspace obtained in optical, infrared (IR) and other wavelength ranges [3]. Moreover, the analysis of images entering the input of such CVS should most often be carried out in real time [4].

The authors of [1–3] believe that such CVSs should be created based on modern deep learning models, primarily based on convolutional neural network (CNN) models [2]. However, on this path it is necessary to solve a number of very complex problems: to select from among the known CNN models the most promising model (models) for implementation in real-time CVS, to conduct a comprehensive analysis of its (their) effectiveness, taking into account the requirements for the accuracy of LO recognition and the speed of calculation this model and, if it meets the requirements of the CVS, solve the problem of software or hardware implementation of such a model as part of the CVS. Moreover, even the preparation of a representative set of images for training and research of a CNN model can be far from a trivial task [5].

The article presents the results of comprehensive studies of the effectiveness of one of the most promising CNN models for creating CVS, YOLOv4-CSP, designed for detection and classification of two classes of aircraft in images: aircraft-type unmanned aerial vehicles (UAVs) and helicopter-type UAVs. At the same time, the problem of recognizing such objects is solved both in optical images (RGB images) and in images obtained in the IR range.

1. The task of detecting flying objects using a CNN model

The task of FO detection in images is the main task solved with the help of CVS when monitoring airspace. It comes down to three subtasks: detecting objects in an image, localizing them, and determining the class of each object. Further, the term «detection» will describe the process of solving all three subtasks. Today, a promising direction in solving such a three-stage detection problem is the use of modern CNN models in CVS [2–4]. When creating a real-time CVS for airspace monitoring, it is necessary to select from among the known CNN models the most promising model(s) for implementation in such a CVS and conduct a comprehensive analysis of its (their) effectiveness, taking into account the requirements for the CVS and, accordingly, to the implemented model CNN.

Let's consider these requirements in more detail. In the images obtained during airspace monitoring using appropriate equipment as part of the CVS, one or more FOs may appear, and in the case of several objects they may belong to different classes. This must be taken into account when initially selecting a CNN model from a set of potentially promising models for solving the problem of FO detection. The main requirements for the CNN model are the contradictory requirements of high speed of model calculations and high accuracy of detection (classification) of FOs in images. It is especially difficult to satisfy the requirement of high computation speed in the case of creating CVS operating in real time [1]. Such CVS must not only detect and classify FOs, but also track their movement in space. The required real-time scale depends on the speed of movement of FO in the airspace. At the same time, as shown in [1, 4], the main computing resources of the real-time CVS are used to calculate the CNN model, which emphasizes the relevance of creating and using the most efficient CNN models in terms of computation speed in such CVSs.

The speed of detecting FOs in an image (the speed of calculating the CNN model), measured as the number of analyzed images per second in FPS (frames per second), should be at least 25 for most real-time CVS [4].

The accuracy of detecting (classifying) objects in an image is usually assessed using the generally accepted metrics $AP_{0,5}$ (Average Precision) and $mAP_{0.5}$ (mean Average Precision) - the average value of $AP_{0.5}$ for all classes of FOs. In accordance with works [2,3], we will assume that the accuracy of detection (classification) of FOs in images using the CNN model is high (the requirement for detection accuracy is met) if the values of the $AP_{0,5}$ metric for each class of FOs and the $mAP_{0,5}$ metric for all classes exceed the threshold value of 60%.

High algorithmic efficiency of the CNN model can be also highlighted as the third requirement. The algorithmic efficiency of the model is assessed based on the calculation of two indicators: the size of the CNN model and its computational complexity. The CNN model size (or the measure of compactness of the architecture), most often measured in MB (megabytes), is the amount of memory of the CVS computing device required to store the weighting coefficients of convolutional layers and intermediate buffers when calculating the CNN model. In turn, the computational complexity of the CNN model is measured in GFLOPs (Giga-FLoating-point OPerations per Second) and is determined by the number of multiplication, addition and comparison operations on floating-point numbers when performing convolution and subsampling procedures in all convolutional and subsampling layers of the model. Note that the threshold values of the model size and computational complexity of the CNN are set by the developers of the real-time CVS. They generally consider that a candidate CNN model satisfies their requirement for algorithmic efficiency if it can be implemented on a mid-range GPU or a modern system-on-chip with a programmable integrated circuit to speed up the computation.

The problem solved in this study can be formulated as follows. It is necessary to create real-time CVS based on the CNN model, which should allow detection of two classes of aircraft in images: aircraft-type UAVs and helicopter-type (multi-rotor) UAVs. The task is set to select from the known CNN models the most promising model for detecting FOs in images and the task of its comprehensive study in order to identify the parameters of the model at which it satisfies all three requirements formulated above, and makes it possible to detect not only single FO, but also several objects in the image, including those belonging to various classes.

2. Choosing a CNN model for research

There are two types of CNN models for detecting objects in images: one-stage and two-stage detectors (Figure 1). First, two-stage detectors appeared, the most famous of them is based on the R-CNN model [6]. It provides high detection accuracy, but has a low image analysis speed. The Faster R-CNN model [7] and its extension Mask R-CNN were developed to solve the problem of the low speed of the R-CNN model. However, according to the evaluation of these CNN models from [2], their computational speed is significantly lower than the selected FPS threshold of 25. Moreover, in [2] it is shown that their algorithmic efficiency is quite low. All this does not allow us to consider such two-stage CNN models as a basis for creating real-time CVS.

The analysis of various classes of CNN models carried out in [1, 2] and the above results of our discussion of a number of CNN models suggest that the most suitable for detecting FOs in images, taking into account the specified requirements, are the YOLO (You Only Look Once) class models [8, 9]. CNN models of this class belong to CNN models with a single-stage architecture; they analyze the image in one pass (the stages of detection, localization and classification of objects are performed in parallel, at the same time), which significantly increases the speed of image analysis. Moreover, such CNN models have very high accuracy in object detection [2]. Some YOLO class models have a complex architecture and therefore can exceed threshold values in terms of computational complexity and the size of the memory occupied by the CNN model. Thus, the first models of this family, such as YOLO, YOLOv2 and YOLOv3, are not suitable for CVS with low computing resources [9]. For such CVSs, CNN models with more compact architectures are needed, including, first of all, a small number of convolutional layers. In this regard, for real-time CVS it is proposed to choose and study the effectiveness of YOLO class CNN models with more modern architectures, starting with the YOLOv4 model [8].

The YOLOv4 CNN model consists of 4 blocks, in contrast to the models of two-stage detectors, consisting of 5 blocks (Figure 1) [8]: Input-input image; Backbone (used to build a deeper CNN in order to increase its accuracy); Neck (used to obtain more detailed spatial and semantic information about an object); Dense Prediction (used to determine bounding box coordinates along with a confidence estimate for a feature class). The last block uses the same principle as the earlier CNN model YOLOv3 [9].

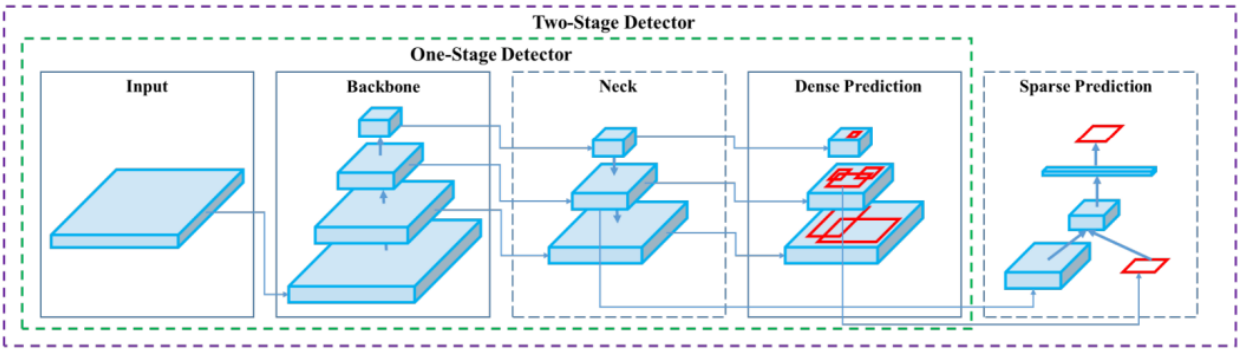

Fig. 1. Single-stage and two-stage detectors

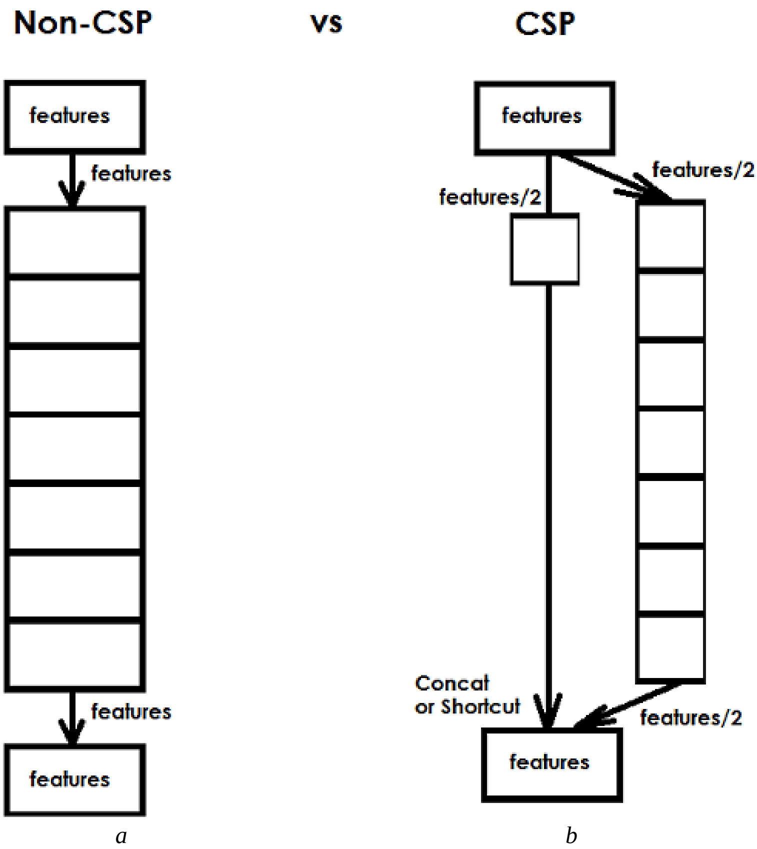

Fig. 2. Backbone block structure: normal connection of layers (*a*); CSP layer connection (*b*)

The YOLOv4-CSP CNN model is a modification of the YOLOv4 model and contains a number of important improvements of intermediate detector blocks. The main ones are the following.

1. Backbone. This block uses a regular CSPDarknet53 connection [10] for the YOLOv4 model (Figure 2a) or a CSP connection for the YOLOv4-CSP CNN model (Figure 2b). A CSP connection is more efficient because its basic idea is:

– a half of the output signal goes along the main path (generating more semantic information with a larger receptive field);

– the second half of the signal takes a detour (preserving more spatial information with a small receptive field).

2. Neck. This block uses additional SPP layers [11], as well as the CNN PAN model for path aggregation (Figure 3).

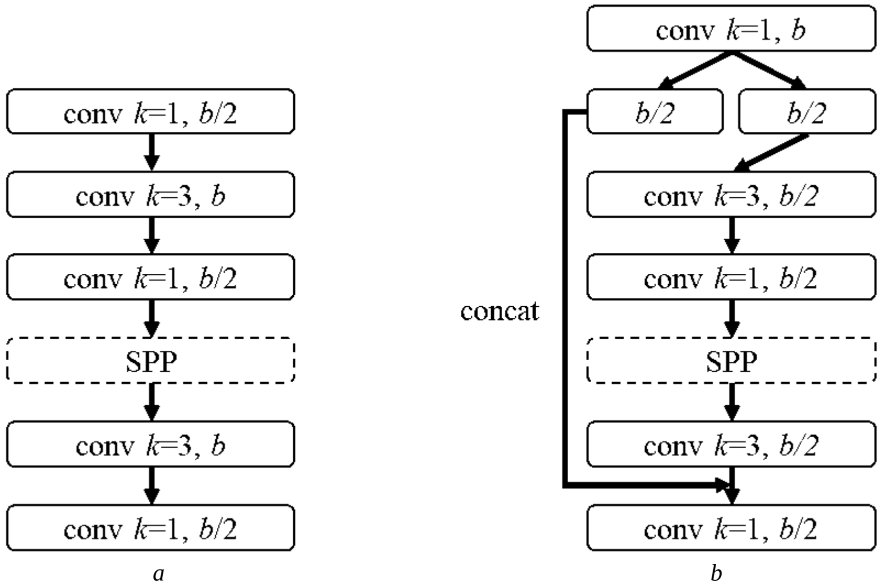

Fig. 3. Neck block structure: YOLOv4 CNN model (*a*); YOLOv4-CSP CNN model (*b*)

Moreover, by applying optimal scaling, efficient CNN models can be obtained from the YOLOv4-CSP model for different input image sizes (Fig. 4).

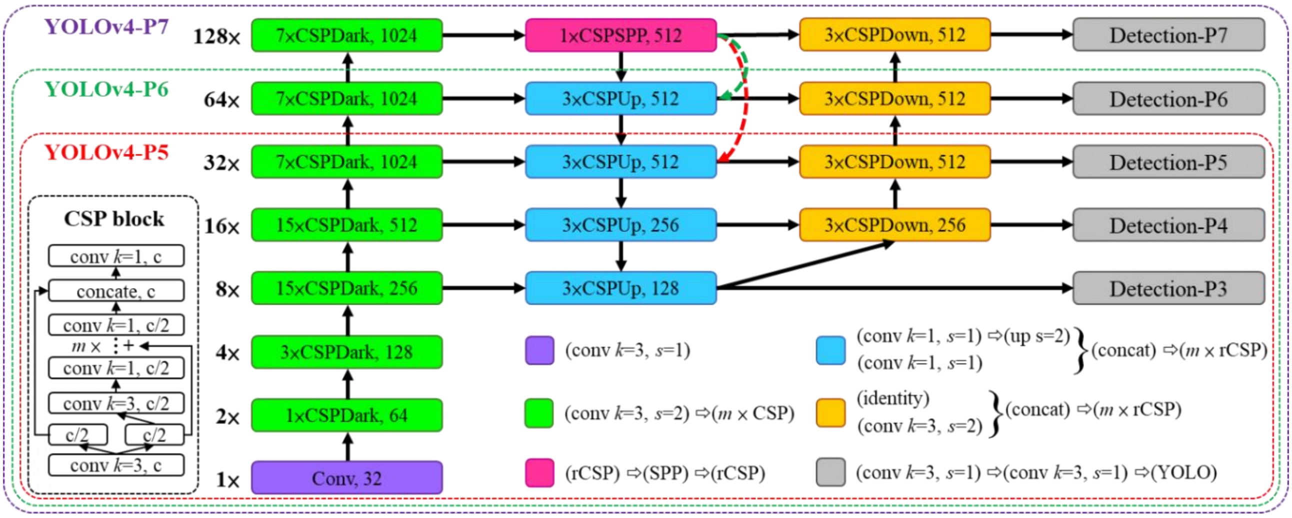

Fig. 4. Examples of YOLOv4-CSP CNN models for input images of various sizes

Compared to previous CNN models of the YOLO class, the YOLOv4-CSP model is more effective for object detection than previous CNN models from this class, which is confirmed by the results of a number of studies on the MS COCO dataset [12]. For example, the average detection accuracy of YOLOv4-CSP (with $mAP_{0.5}$ metric) is increased from 57,9% to 65,7% compared to the YOLOv3 model. All of the above made it possible to select YOLOv4-CSP for potential use in real-time CVS as a promising CNN model. Next, it is necessary to conduct a study of the effectiveness of this model and determine its compliance with the requirements described above for the CVS and, accordingly, for the CNN model implemented in such a system.

3. Training the YOLOv4-CSP CNN model and studying its effectiveness

3.1. Dataset preparation

To study the effectiveness of the YOLOv4-CSP CNN model in solving the problem of detecting FOs, two datasets were marked and prepared based on the results of shooting FOs of two classes: helicopter-type UAVs and aircraft-type UAVs. The size of each dataset is 6230 labeled images. The first dataset is formed on the basis of 1150 optical images (RGB), the second - on the basis of 1160 images obtained in the IR range. To expand the datasets to 6230 images, data augmentation was carried out in each of them using the following algorithms:

– varying the size of images;
– mirroring images horizontally and vertically;
– rotation of images at an angle from 1 to 15 degrees;
– mosaic placement of objects in the image.

70% of the volume of each of these datasets was used for training the YOLOv4-CSP CNN model, 20% was used for validation, and 10% was used for testing/research. The Python programming language and the PyTorch framework were used for the software implementation of the YOLOv4-CSP CNN. Training of the CNN model and testing/research of its effectiveness were carried out on a computer with the following characteristics: Intel Core i9-11900KF processor, 64 GB RAM, NVIDIA Quadro RTX 6000 video card with 24 GB of video memory.

3.2. Results of FO detection in images

The number of epochs was set equal to 200, and learning rate was set equal to 0.001. These parameters were not changed during training and validation processes for the CNN model. The Adam algorithm was used as an optimizer.

When testing the trained CNN model YOLOv4-CSP and conducting studies of its effectiveness on datasets with optical images and images in the IR range, the following model parameters were changed: input image size (416x416, 512x512, 608x608 pixels), mini-batch size (4, 8) and activation function (Leaky ReLU, Mish).

Figure 5 shows, as an example, the results of detection using the trained and validated CNN model YOLOv4-CSP FO of each class on optical images.

a

b

Fig. 5. Results of FO detection on optical images: Helicopter-type UAV (*a*); Aircraft-type UAV (*b*)

The first experiment was conducted to determine the effect on the accuracy of FO detection using this model depending on the size of mini-batches, the size of the input optical images, as well as the approaches used to train the model, implemented in the form of BoF (Bag of Freebies) procedures and BoS (Bag of Specials). The Leaky ReLU activation function was used in the experiment. BoF and BoS are procedures that implement methods and techniques that change the training strategy or cost of training a CNN model to improve the accuracy of detecting objects in images with its help. These methods are implemented in the model in the form of procedures (various plugins and post-processing modules), which can significantly increase the accuracy of object detection at the cost of a slight increase in the cost of training the CNN model [8]. The results of first experiment are shown in table 1.

Table 1

Accuracy of FO detection in optical images for various mini-batch sizes, input images and usage of BoF/BoS procedures

mini-batch size	input image size	BoF/BoS	$mAP_{0.5}$, %
4	416 × 416	–	61,5
4	416 × 416	+	62,7
8	416 × 416	–	62,3
8	416 × 416	+	63,2
4	512 × 512	–	62,9
4	512 × 512	+	64,7
8	512 × 512	–	63,1
8	512 × 512	+	64,8

From table 1 it follows that the use of BoF/BoS procedures in conducted experiment significantly increases the accuracy of FO detection. In this case, the size of the mini-batch has little effect on it, but the accuracy increases with increasing size. The accuracy of FO detection increases with increasing size of input images.

In the second experiment, the accuracy of FO detection in optical images was determined using the YOLOv4-CSP CNN model using different activation functions and for different sizes of input images. The BoF/BoS procedures were also used, the mini-batch size remained unchanged and was equal to 8. The results of the experiment are shown in table 2.

Table 2

Accuracy of FO detection in optical images for various activation functions and input image sizes

activation function	input image size	$mAP_{0.5}$, %
Leaky ReLU	416 × 416	63,2
Mish	416 × 416	63,7
Leaky ReLU	512 × 512	64,8
Mish	512 × 512	65,2
Leaky ReLU	608 × 608	65,1
Mish	608 × 608	66,1

In the third experiment, the detection rate of FO was determined using the YOLOv4-CSP CNN model for various sizes of input optical images. In this case, BoF/BoS procedures were used, and the mini-batch size remained unchanged and was equal to 8. The results of the experiment are shown in table 3.

Table 3

Speed of FO detection in optical images for different input image sizes

input image size	Frames per second (FPS)	Average detection time, ms
416 × 416	43	23,2
512 × 512	35	28,6
608 × 608	29	34,5

Here, the detection rate was defined as calculated frames per second and the average time of FO detection per analysis of one image. It was calculated by averaging the results over the detection time of 623 optical images from the test sample of the first dataset.

In experiments №№ 4–6, only the initial data for training, validating and testing the YOLOv4-CSP CNN model was changed: instead of optical images, images obtained in the IR range were fed to the input of the model. The results of these experiments are presented in table 4, table 5 and table 6. The results in table 5 are obtained using a mini-batch size of 8 and using the BoF/BoS procedures. The average FO detection time required to analyze one image was calculated by averaging the results over the detection time of 623 such images from the test sample of the second dataset.

Table 4

Accuracy of FO detection in IR images for various mini-batch sizes, input images and usage of BoF/BoS procedures

mini-batch size	input image size	BoF/BoS	$mAP_{0.5}$, %
4	416 × 416	–	55,3
4	416 × 416	+	56,9
8	416 × 416	–	56,8
8	416 × 416	+	57,2
4	512 × 512	–	57,6
4	512 × 512	+	59,1
8	512 × 512	–	58,4
8	512 × 512	+	59,5

Table 5

Accuracy of FO detection in IR images for various activation functions and input image sizes

activation function	input image size	$mAP_{0.5}$, %
Leaky ReLU	416 × 416	56,9
Mish	416 × 416	57,2
Leaky ReLU	512 × 512	59,5
Mish	512 × 512	60,1
Leaky ReLU	608 × 608	60,3
Mish	608 × 608	60,6

Table 6

Speed of FO detection in IR images for different input image sizes

input image size	Frames per second (FPS)	Average detection time, ms
416 × 416	42	23,8
512 × 512	36	27,7
608 × 608	27	37,0

4. Analysis of the obtained results

As it follows from table 1 and table 4, the accuracy of FO detection in images using the YOLOv4-CSP CNN model in the case of using the BoF/BoS procedures changes slightly with increasing mini-batch size. Without the use of these procedures, regardless of the wavelength range in which images of objects are obtained, the size of the mini-batch significantly affects the accuracy of FO detection: the larger its value, the more accurate the detection. This is consistent with the results obtained when detecting various objects of a different physical nature in images [11].

The experimental results in table 2 and table 5 show that the accuracy of FO detection in images using the YOLOv4-CSP CNN model is higher when using the Mish activation function for different input image sizes. This function is smooth, non-monotonic, bounded below and unbounded above, so these properties allow it to obtain better results compared to the Leaky ReLU activation function. However, existing implementations of the Mish activation function depend on support for CUDA hardware instructions, making it less universally usable than the Leaky ReLU activation function. This should be taken into account when developing CVS.

From the experimental results shown in table 3 and table 6, it follows that the best speed of detecting FOs in images using the CNN model under study can be obtained with an input image size of 416x416 pixels. As the size of the input images increases, the detection time per image increases, and the number of frames processed per second decreases. At the same time, the research results indicate that with the considered sizes of input images (416x416, 512x512 and 608x608 pixels), the CVS created on the basis of the YOLOv4-CSP CNN model for FO detection will work in real time (the FPS parameter value in all cases is above 25).

Analysis of the results obtained shows that the accuracy of FO detection using the $mAP_{0.5}$ metric using the YOLOv4-CSP CNN model on optical images is 5–6% higher than on images obtained in the IR range.

This is explained by the fact that the model, when generating feature maps from optical images, also takes into account the characteristics of FO colors. Moreover, exceeding the threshold for the detection accuracy of FO, equal to 60%, occurs in all studied cases of optical image analysis (table 1 and table 2). However, for IR images, exceeding this threshold was detected only for three cases: with input image sizes of 512x512 and 608x608 pixels and the Mish activation function, as well as for input images with sizes of 608x608 pixels and the Leaky ReLU activation function (table 5). It is the results obtained on the accuracy of FO detection that make it possible to formulate recommendations for the use of the YOLOv4-CSP CNN model as the basis of real-time CVS. Thus, for optical images, taking into account the specifics of the created CVS, it can be recommended to select a model with parameters from a fairly wide set. For images obtained in the IR range, it is possible to use in the CVS only three versions of the model, which were identified in the experiments.

Conclusion

The most important task in airspace control is the task of detecting and classifying various FO. In recent years, FO images obtained in the optical and IR wavelength ranges have been increasingly analyzed to solve this problem. For this purpose, CVSs are created based on modern CNN models. We selected the YOLOv4-CSP model, which is part of the YOLO class, as a promising CNN model for potential use in real-time CVS.

Comprehensive studies have been carried out on the effectiveness of the selected CNN model in detecting two classes of FO in images: helicopter-type UAVs and aircraft-type UAVs. It was revealed that the accuracy of detecting such FOs using the YOLOv4-CSP CNN model is positively influenced by the use of BoF/BoS procedures and an increase in the size of input images. To a lesser extent, the accuracy is influenced by the choice of activation function (the best results were obtained using the Mish activation function) and mini-batch size (a larger mini-batch slightly improves detection accuracy). The obtained estimates of the speed of FO detection in images indicate the fundamental possibility of creating a CVS based on the YOLOv4-CSP CNN model, which detects FO in real time.

It has been shown that the accuracy of FO detection using this CNN model in optical images is 5–6% higher than in images obtained in the IR range. Exceeding the threshold for the detection accuracy of FO, equal to 60%, occurs in all studied cases of optical image analysis. However, for IR images, exceeding this threshold was detected only for three variants of the model. The results on the accuracy of FO detection make it possible to formulate recommendations for the use of the YOLOv4-CSP CNN model as the basis for real-time CVS.

References

1. Zoev, I.V., Markov, N.G. & Ryzhova, S.E. (2019) Intelligent computer vision system for unmanned aerial vehicles for monitoring technological objects of oil and gas industry. *Bulletin of the Tomsk Polytechnic University. Geo Assets Engineering*. 330(11). pp. 34–49. DOI: 10.18799/24131830/2019/11/2346
2. Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. & Farhan, L. (2021) Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*. 8(1). pp. 53. DOI: 10.1186/Fs40537-021-00444-8
3. Makarenko, S.I., Timoshenko, A.V. & Vasilchenko, A.S. (2020) Counter Unmanned Aerial Vehicles. Part 1. Unmanned aerial vehicle as an object of detection and destruction. *Systems of Control, Communication and Security*. 1. pp. 109–146. DOI: 10.24411/2410-9916-2020-10105
4. Klekovkin, V.A. & Markov, N.G. (2023) Model of convolutional neural network LeNet5 for detection and classification of air space objects in images. *Bulletin of the Tomsk Polytechnic University. Industrial Cybernetics*. 1(2). pp. 11–16. DOI: 10.18799/29495407/2023/2/17
5. Nebaba, S.G. & Markov, N.G. (2022) Effectiveness of moving objects detecting and tracking in airspace by images in near-infrared. *Light & Engineering*. 30(2). pp. 62–69. DOI: 10.33383/2021-102
6. Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 580–587. DOI: 10.48550/arXiv.1311.2524

7. 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*. 39(6). pp. 1137–1149. DOI: 10.48550/arXiv.1506.01497
8. Bochkovskiy, A., Wang, C.Y. & Liao, H.Y.M. (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection. *Journal arXiv, preprint arXiv*: 2004.10934v1. DOI: 10.48550/arXiv.2004.10934
9. Redmon, J. & Ali, F. (2018) Yolov3: An incremental improvement. *Journal arXiv, preprint arXiv*:1804.02767. DOI: 10.48550/arXiv.1804.02767
10. Wang, C.Y., Mark Liao, H.Y., Yeh, I.H., Wu, Y.H., Chen, P.Y. & Hsieh, J.W. (2019) CSPNet: A New Backbone that can Enhance Learning Capability of CNN. *Journal arXiv, preprint arXiv*: 1911.11929v1. DOI: 10.48550/arXiv.1911.11929
11. He, K., Zhang, X., Ren, S. & Sun, J. (2015) Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *Journal arXiv, preprint arXiv*: 1406.4729v4. DOI: 10.48550/arXiv.1406.4729
12. Wang, C.Y., Bochkovskiy, A. & Liao, H.Y.M. (2021) Scaled-YOLOv4: Scaling Cross Stage Partial Network. *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*. DOI: 10.48550/arXiv.2011.08036

Information about the authors:

Nebaba Stepan G. (Candidate of Technical Sciences, Associate Professor, National Research Tomsk Polytechnic University, Tomsk, Russian Federation). E-mail: stepanlfx@tpu.ru
Markov Nikolay G. (Doctor of Technical Sciences, Professor, National Research Tomsk Polytechnic University, Tomsk, Russian Federation). E-mail: markovng@tpu.ru

Contribution of the authors: the authors contributed equally to this article. The authors declare no conflicts of interests.

***Информация об авторах*:**

Небаба Степан Геннадьевич – кандидат технических наук, доцент отделения информационных технологий Инженерной школы информационных технологий и робототехники Национального исследовательского Томского политехнического университета (Томск, Россия). E-mail: stepanlfx@tpu.ru
Марков Николай Григорьевич – доктор технических наук, профессор отделения информационных технологий Инженерной школы информационных технологий и робототехники Национального исследовательского Томского политехнического университета (Томск, Россия). E-mail: markovng@tpu.ru

Вклад авторов: все авторы сделали эквивалентный вклад в подготовку публикации. Авторы заявляют об отсутствии конфликта интересов.

Received 08.07.2024; accepted for publication 02.12.2024

Поступила в редакцию 08.07.2024; принята к публикации 02.12.2024