This paper considers a model of object detection on aerial photographs and video using a neural network in unmanned aerial systems. The development of artificial intelligence and computer vision systems for unmanned systems (drones, robots) requires the improvement of models for detecting and recognizing objects in images and video streams. The results of video and aerial photography in unmanned aircraft systems are processed by the operator manually but there are objective difficulties associated with the operator's processing of a large number of videos and aerial photographs, so it is advisable to automate this process. Analysis of neural network models has revealed that the YOLOv5x model (USA) is most suitable, as a basic model, for performing the task of object detection on aerial photographs and video. The Microsoft COCO suite (USA) is used to train this model. This set contains more than 200,000 images across 80 categories. To improve the YOLOv5x model, the neural network was trained with a set of VisDrone 2021 images (China) with the choice of such optimal training parameters as the optimization algorithm SGD; the initial learning rate (step) of 0.0005; the number of epochs of 25. As a result, a new model of object detection on aerial photographs and videos with the proposed name VisDroneYOLOv5x was obtained. The effectiveness of the improved model was studied using aerial photographs and videos from the VisDrone 2021 set. To assess the effectiveness of the model, the following indicators were chosen as the main indicators: accuracy, sensitivity, the estimation of average accuracy. Using a convolutional neural network has made it possible to automate the process of object detection on aerial photographs and video in unmanned aerial systems

Keywords: neural network, object detection, VisDrone 2021, Microsoft COCO, YOLOv5x, unmanned aerial system

UDC 004.932
DOI: 10.15587/1729-4061.2022.252876

IMPROVING THE MODEL OF OBJECT DETECTION ON AERIAL PHOTOGRAPHS AND VIDEO IN UNMANNED AERIAL SYSTEMS

Vadym Slyusar
Doctor of Technical Sciences, Professor
Research Institute Group*
Mykhailo Protsenko
PhD, Senior Researcher
Office of Special Forces*
Anton Chernukha
Corresponding author
PhD, Associate Professor
Department of Service and Training
National University of Civil Defence of Ukraine
Chernyshevska str., 94, Kharkiv, Ukraine, 61023
E-mail: an_cher@nuczu.edu.ua
Vasyl Melkin
PhD
Organizational and Scientific Division*
Oleh Biloborodov
Doctor of Technical Sciences
Research Unit*
Mykola Samoilenko
Doctor of Agricultural Sciences, Professor
Department of Viticulture and Horticulture**
Olena Kravchenko
PhD, Associate Professor
Department of Genetics, Animal Feeding and Biotechnology**
Halyna Kalynychenko
PhD, Associate Professor
Department of Livestock Production Technology**
Anton Rohovyi
PhD
Department of Strategic Management***
Mykhaylo Soloshchuk
PhD
Department of Computer Science and Intellectual Property***
*Central Scientific Research Institute of Armament
and Military Equipment of the Armed Forces of Ukraine
Povitroflotsky ave., 28, Kyiv, Ukraine, 03049
**Mykolayiv National Agrarian University
Heorhiya Honhadze str., 9, Mykolayiv, Ukraine, 54020
***National Technical University «Kharkiv Polytechnic Institute»
Kyrpychova str., 2, Kharkiv, Ukraine, 61002

Received date 02.12.2021
Accepted date 13.01.2022
Published date 28.02.2022

How to Cite: *Slyusar, V., Protsenko, M., Chernukha, A., Melkin, V., Biloborodov, O., Samoilenko, M., Kravchenko, O., Kalynychenko, H., Rohovyi, A., Soloshchuk, M. (2022). Improving the model of object detection on aerial photographs and video in unmanned aerial systems. Eastern-European Journal of Enterprise Technologies, 1 (9 (115)), 24–34. doi: https://doi.org/10.15587/1729-4061.2022.252876*

1. Introduction

Safety is a paramount human need. The issue to provide security is associated with the active use of unmanned aerial vehicles (UAVs) to monitor military sites, as well as critical infrastructure facilities. The latter include energy facilities, chemically hazardous industries, and other strategic objects, the disruption of the functioning of which may threaten

vital state interests. A concept of devising an integrated information and analytical system of decision support under the conditions of anthropogenic emergencies has been proposed in [1]. The main factors threatening the safety of a monitored object (MO) include fires (explosions), emissions of hazardous substances, radiation, as well as unauthorized entry of persons into a MO territory. With the help of computer vision, MO reconnaissance is carried out by analyzing aerial photographs and a video stream. Based on artificial intelligence methods, real-time monitoring of traffic flows is carried out [2], real-time detection of vehicles [3]. They create unmanned aircraft systems (UAS), unmanned vehicle systems, and robot control. The development of artificial intelligence systems should be accompanied by the improvement of models for their implementation. One such option may be to use convolutional neural networks (CNNs) to detect MO in aerial photographs and videos.

Therefore, it is a relevant task to improve existing models of object detection in aerial photographs and videos using CNNs.

2. Literature review and problem statement

Work [4] describes a video system for detecting violations of traffic rules. The proposed model implements the detection of three classes of objects on a video sequence: a pedestrian crossing, a car, and a person at a pedestrian crossing. The model also makes it possible to track the trajectory of the vehicle and person at the pedestrian crossing; determine the violation of traffic rules over a certain period. To detect objects in real time, the YOLOv3 neural network was used. The disadvantage of that model is its high computational complexity, lack of adaptation to the detection of objects in the video stream acquired from an unmanned aerial vehicle (UAV).

A traffic video surveillance system is proposed in [5]. The project addresses the concept of vehicle detection with the support of a computer vision algorithm in real time. The proposed system uses the YOLOv4 architecture for faster detection of objects in real time. That model has been tested in a variety of conditions such as rain, low visibility, daylight, snow, and night. The system can automate the process of detecting accidents in real time. However, the task of object detection in the video stream acquired from UAV remains unresolved.

Paper [6] shows that deep learning technique has led to a significant increase in the accuracy of object detection. In many applications, object detection is performed on video data consisting of a sequence of two-dimensional image frames. It is shown that the accuracy of object detection can be significantly improved by using a temporal structure in the sequence of images at the stage of object detection. A new model for object detection is proposed, which takes into consideration the trajectory of movement from neighboring frames, as well as spatial-temporal characteristics. The disadvantage of the model is the inability to use it to detect objects when processing a video stream in UAS.

A prototype of the implementation of a threat detector based on artificial intelligence for video surveillance cameras is considered in [7]. The proposed CNN model processes the stream of images directly from the webcam on the site, classifies objects, and displays the results to the user through a convenient graphical interface. The motion detection module is designed to automatically capture images from video when new motion is detected. Experimental results showed that the average overall accuracy of forecasts for the test set date was 94 %. The disadvantage of the approach used is the lack of practical application for object detection on aerial photographs and video acquired from the optical system of UAV.

Work [8] shows that object detection is closely related to the analysis of streaming video. Owing to the rapid development of deep learning neural networks, new CNN models are emerging that are able to resolve this task. The disadvantage of the approach used is the lack of practical application for object detection when processing a video stream in UAS.

Paper [9] proposes a new approach to the detection of YOLO objects (You Only Look Once). Object detection is considered as a regression problem for spatially separated bounding frames and related probabilities of classes. The neural network predicts bounding boxes and probabilities of classes directly from complete images in a single estimate. The model is superior to other detection models such as DPM and R-CNN. Despite this, the issues of automation of the process of object detection in aerial photographs and streaming video in UAS were not considered.

In work [10], a method for recognizing images of objects monitored by a convolutional neural network is proposed. The effectiveness of image recognition of monitored objects by an improved method was tested on a convolutional neural network, which was trained with images of 300 monitored objects. In that case, the decision-making time for the proposed method decreased on average from 0.7 to 0.84 s compared with the artificial neural networks ResNet and ConvNets. The disadvantage of the proposed model is the lack of practical application for object detection when processing a video stream in UAS.

Paper [11] proposes a model of YOLO9000 object detection in real time, which can detect more than 9,000 categories of objects. The advanced YOLOv2 model corresponds to the latest technology in standard detection tasks such as PASCAL VOC and COCO. A method of joint training in the detection and classification of objects is proposed. The YOLO9000 is trained on both the COCO discovery dataset and the ImageNet classification dataset. YOLO9000 predicts detections for 200 classes and 9,000 different categories of objects and works in real time.

Our review of the scientific literature [4–11] has revealed the following shortcomings of known models:

– the lack of CNN models that solve the task of object detection on aerial photographs and streaming video in UAS;

– the process of object detection on aerial photographs and videos in UAS is not automated.

All this suggests that it is expedient to conduct a study on improving the model of object detection on aerial photographs and video in unmanned aircraft systems.

3. The aim and objectives of the study

The purpose of this study is to improve the model of object detection on aerial photographs and video in unmanned aircraft systems using CNN and choosing the parameters of its training. This would make it possible to automate the process of object recognition in aerial photographs and videos.

To accomplish the aim, the following tasks have been set:

– to investigate the effectiveness of object detection on aerial photographs using CNN;

– to evaluate the effectiveness of object detection on aerial photographs and video streams with an improved model VisDroneYOLOv5x.

4. The study materials and methods

A video camera is installed on board the UAV. The video stream is transmitted through the communication channel to the ground control point. To simplify understanding and processing, each frame of the video stream is treated as one digital image (Fig. 1). The task of object detection is to assign each image P to one object (set of objects) B in a certain class:

$$\varphi: P \to B,\ B = \left\{b_k,\ k = \overline{0,|B|-1}\right\}, \quad (1)$$

where $b_k = \left(\left(x_1^k, y_1^k\right), \left(x_2^k, y_2^k\right), \left[s^k, c^k\right]\right)$, $s^k \in R$ is the reliability, c^k is the class of objects (person, plane, car, truck, bus, etc.), k is the number of classes. In fact, the task of finding the location of an object in a frame is a detection task.

The proposed model can work with two types of data: aerial photography and streaming video. In the first case, an RGB JPEG aerial photograph is submitted to the CNN input, and an aerial photograph with marked classes (objects) in the form of rectangular frames is received at the output (Fig. 2).

In the second case, a video in MPEG-4 format is submitted to the CNN input, and a video with marked classes (objects) in the form of rectangular frames (similar to an aerial photograph) is received at the output. Table 1 gives the color of the rectangular frame of the class (object), using an example of 6 classes (total – 80) of the model YOLOv4, YOLOv5x.

Knowing the coordinates of objects in the image (video) makes it possible to solve various more complex problems:

– tracking (tracking of movement);

– prediction of actions;

– simultaneous localization and construction of the SLAM map (simultaneous localization and mapping);

– estimation of distances to objects.

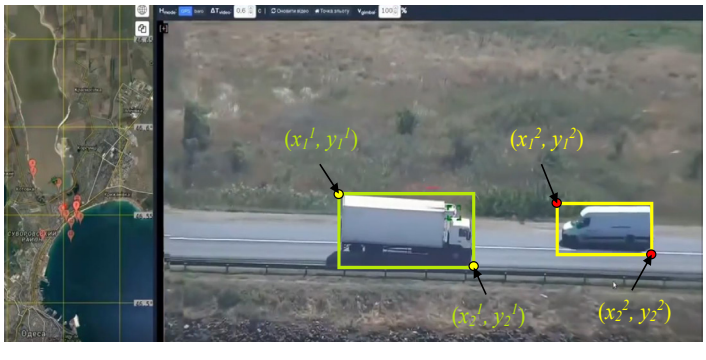


Fig. 1. The process of object detection on an image (a frame of the video stream)

Table 1

Rectangular class (object) frame color for YOLOv4, YOLOv5x model

Class	Class title	Mark	Color of rectangular frame	
			YOLOv4	YOLOv5x
1	Person	person		
2	Airplane	airplane		
3	Bus	bus		
4	Truck	truck		
5	Car	car		
...				
80	Boat	boat		

As the main performance indicators that characterize the detection process, the following were chosen at GitHub: precision, recall, mean average precision.

Precision P is the ratio of correctly detected objects to the total number of correctly and erroneously detected objects [12]:

$$P = \frac{TP}{TP + FP}, \quad (2)$$

where TP (true positive) is the number of correctly detected objects in the image; FP (false positive) is the number of erroneously detected objects in the image.

Recall r is the ratio of correctly detected objects to the total number of detected objects in the images [12]:

$$r = \frac{TP}{TP + FN}, \quad (3)$$

where FN (false negative) is the number of false positives in the background.

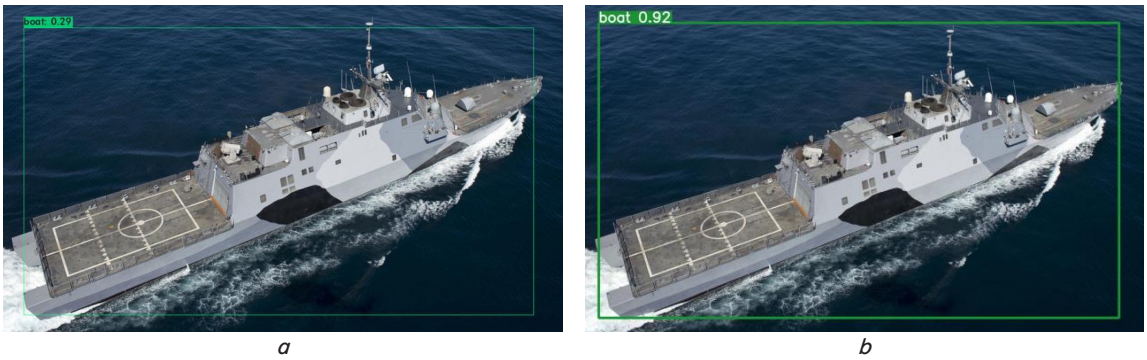
The mean Average Precision (mAP) score is the mean of Average Precision (AP) for each class in the training sample:

$$AP = \frac{1}{11} \sum_{r \in 0,0.1,\ldots,1} P(r). \quad (4)$$

The estimate of the average precision is determined from the formula:

$$mAP = \frac{\sum_k AP_k}{k}, \quad (5)$$

where k is the number of classes.



a *b*

Fig. 2. An example of detecting a vessel-type object on an aerial photograph (a frame of a video stream): *a* – by the YOLOv4 model; *b* – by the YOLOv5x model

Chronology of development of YOLO models:

– YOLOv1 (June 8, 2015): you look only once: unified real-time object detection;

– YOLOv2 (December 25, 2016): YOLO9000: better, faster, more accurate;

– YOLOv3 (April 8, 2018): YOLOv3: gradual improvement;

– YOLOv4 (April 23, 2020): YOLOv4: Optimal speed and accuracy of object detection;

– YOLOv5 (18 May 2020).

A rationale for the architecture for implementing the proposed CNN.

Features of the architecture and principles of implementation of the YOLOv4 CNN are considered in [13, 14]. The model YOLOv5 [15] shows high efficiency for object detection of various shapes and positions both in digital images and videos. YOLOv5 includes models of different sizes: YOLOv5n (small), YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x (large).

The advantages of YOLOv5. YOLOv5 is the first model from the YOLO family to be implemented in the PyTorch framework. Previous models were written on the Darknet, the framework of the creator of the architecture. The Darknet loses to PyTorch in the context of performance, configuration capabilities, and model deployment. In Colab Notebooks with the Tesla P100, the YOLOv5 produces predictions at a rate of 0.007 seconds per image. That's equivalent to 140 frames per second. For comparison, YOLOv4 operates at 50 frames per second.

YOLO is a modern real-time object detector, and YOLOv5 (Fig. 3) is based on YOLOv1-YOLOv4 [15]. Continuous improvements have made it possible to achieve the highest performance on two official sets of object detection datasets: Pascal VOC (Visual Object Classes) [16] and Microsoft COCO (Shared Objects in Context) [17]. The architecture of the YOLOv5 CNN (Fig. 3) is discussed in [15], the task that CNN solves is to detect objects from 80 classes (Microsoft COCO).

The architecture of the YOLOv5 network consists of three parts. The data are first entered into CSPDarknet to extract the features, and then passed to PANet for the merge function. At the end, Yolo Layer displays the detection results (class, score, location, size).

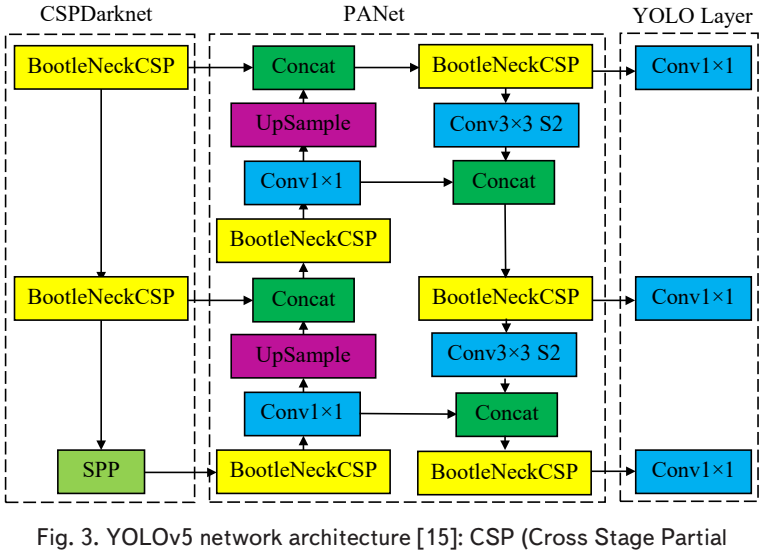


Fig. 3. YOLOv5 network architecture [15]: CSP (Cross Stage Partial Network) – multi-stage partial network; Conv (Convolutional Layer) – convolutional layer; SPP (Spatial Pyramid Pooling) – spatial pyramidal layer; Concat (Concatenate Function) – a layer with a merge function

To solve the problem of automation and increase the efficiency of object detection on aerial photographs and videos in UAS, it is proposed at GitHub to use the YOLOv5x CNN as a basic model. This model is the most accurate of the YOLOv5 line. Improvement of the model of object detection in aerial photographs and in the video stream was carried out by training the YOLOv5x neural network with a set of images VisDrone 2021 (China) [18]. This set includes 10 classes: pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, motor. YOLOv5x training was conducted with the choice of optimal training parameters: optimization algorithm – SGD; the initial learning rate (step) is 0.0005, the number of epochs is 25. The VisDrone 2021 set consists of 400 video files consisting of 265,228 frames and 10,209 images that were acquired from the camera installed on UAV. A feature of this set is that it is assembled under different lighting conditions, different densities of objects, and weather conditions.

Details of work (source code) of YOLOv5, principles, full documentation (books) on training, testing and deployment of the model are given in the official repository [19]. In that repository, there are links to tutorials, the main ones are: training user data; tips for achieving the best training results; registration of scales and displacements; Multi-GPU training assembly of the model; reduction and sparseness of the model; evolution of hyperparameters; transfer learning with frozen layers.

Their study of object detection on aerial photographs and videos, plotting graphs, was conducted in the python 3.7 programming language in the cloud service (machine learning modeling environment) Colab Notebooks PRO version (paid), Tesla T4 GPU runtime 15,110 MB. They used computer ACPI X64 (China) with the operating system Windows 10 Pro, which is equipped with a processor AMD Ryzen 3 1200 Quad-Core Processor 3.10 GHz, graphics card GPU GTX 1050 2 GB, and RAM capacity of 16 GB.

When studying the model, aerial photographs and videos obtained from the UAV were used. According to the NATO classification (STANAG 4670 (ATP 3.3.7)), the UAV used belongs to class I (≤150 kg), category – small (>15 kg). Tactical and technical characteristics of the UAV:

– engine type – internal combustion engine;

– wingspan – 3 m;

– weight with full equipment – 33 kg;

– maximum flight duration – 5 hours;

– maximum flight speed – 140 km/h;

– maximum flight altitude – 2,000 m;

– Full HD video transmission range in real time – 50 km;

– telemetry communication range – 85 km.

The UAV for aerial photography uses a Sony ILCE-7M2K camera, which has the following characteristics and functions:

– matrix type – 35-mm full-frame CMOS-matrix Exmor™ (35.8×23.9 mm);

– recording format (photo) – RAW, JPEG, JPEG Extra fine, JPEG Fine;

– image size (pixels) – 6,000×4,000 (24 M), 6,000×3,376 (20 M), 3,936×2,624 (10 M), 3,936×2,216 (8.7 M), 3,008×2,000 (6,0 M), 3,008×1,688 (5.1 M), 1,968×1,312 (2.6 M);

– digital zoom – up to 8x;

– dimensions ($W \times H \times D$) – 126.9×95.7×59.7 mm;

– weight – 559 g.

To obtain streaming video, a multi-sensor gyrostabilized suspension USG-212 EO/IR is used, which was designed for use on UAVs and small manned aircraft. The gimbal is equipped with a Sony Full-HD block camera with 30x optical zoom and a high-quality infrared camera. It is hermetically sealed and can be operated in all weather conditions. Optionally, a built-in image processing unit is available, adding functions such as target tracking, coordinate acquisition, and video stabilization. The anti-vibration damping system eliminates the vibrations of the UAV hull. Even when using a 30x zoom, the image remains clear and stable.

Their studies were carried out under the following assumptions and limitations:

– a digital photo and video camera (a digital camera with the ability to shoot photos and videos can be used) is installed on board the UAV; it shoots in the view range during daytime;

– an aerial photograph (video) in digital form is transmitted through the communication channel to the ground control point;

– the process of object detection on aerial photographs (video) is carried out on the computer of the ground control point of UAS.

5. Results of studying the effectiveness of object detection on aerial photographs and videos using CNN

5. 1. Studying the effectiveness of object detection on aerial photographs using CNN

The efficiency of object detection on aerial photographs using CNN of the following models YOLOv4, YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, trained on the Microsoft COCO dataset, VisDrone 2021 was investigated. The comparison was carried out in the cloud service Colab Notebooks runtime Tesla T4 15110 MB. Fig. 4 shows the original aerial photo from the VisDrone 2021 set.

At the first stage, a study was conducted on the detection of objects on an aerial image (from the VisDrone set (VisDrone2019-DET-test-challenge)) in the Colab Notebooks cloud environment using models YOLOv4, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, trained on the Microsoft COCO dataset. The result of object detection for the model YOLOv4, YOLOv5x is shown in Fig. 5, 6, respectively.

The validation parameters of the YOLOv4, YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x models trained on the Microsoft COCO dataset are given in Table 2.

Fig. 4. Aerial photograph from a VisDrone set

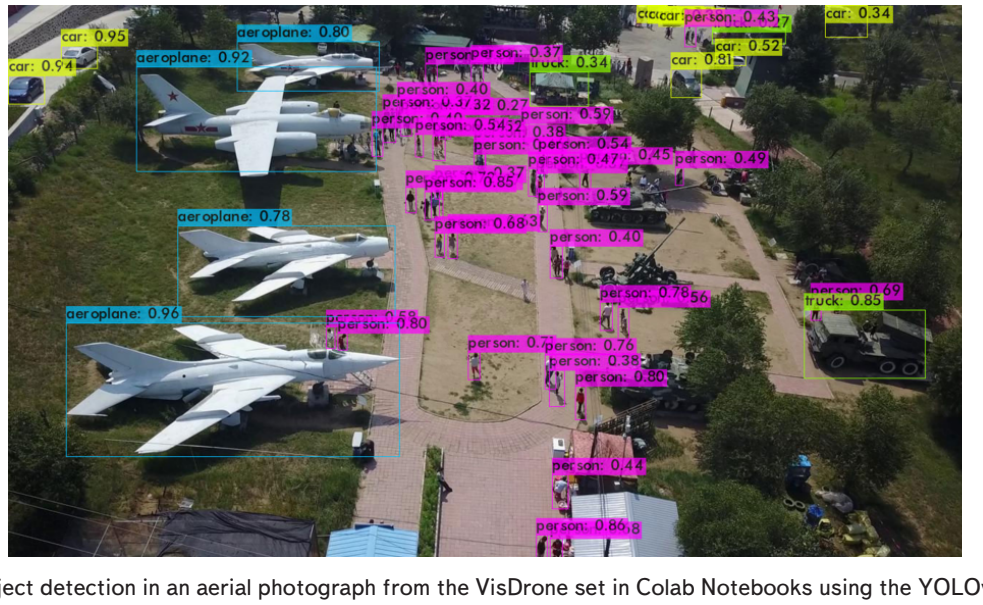


Fig. 5. Object detection in an aerial photograph from the VisDrone set in Colab Notebooks using the YOLOv4 model

Fig. 6. Object detection in an aerial photograph from the VisDrone set in Colab Notebooks using the YOLOv5x model

Table 2

Parameters for checking models YOLOv4, YOLOv5

Model	Number of layers	Number of parameters	Test image size	Average detection time, s
YOLOv4	161	61,640,962	1,360×765	0.166
YOLOv5n	213	1,867,405	1,360×765	0.0096
YOLOv5s	213	7,225,885	1,360×765	0.013
YOLOv5m	290	21,172,173	1,360×765	0.027
YOLOv5l	367	46,533,693	1,360×765	0.037
YOLOv5x	444	86,705,005	1,360×765	0.047

Analysis of their results given in Table 2 shows that the best indicators for the time of detection of aerial photographs with a size of 1,360×765 are demonstrated by the YOLOv5n model – 0.0096 s, the largest detection time was demonstrated by the YOLOv4 model – 0.166 s.

Table 3 gives the result of checking the YOLOv4, YOLOv5 models on the Microsoft COCO 2017 validation set (the number of images for validation is 5,000, the image size is 640×640, the number of tags is 36,335).

Analysis of their results given in Table 3 reveals that the best average precision is demonstrated by the model YOLOv5x – $mAP_{0.5}$=0.683, $mAP_{0.5...95}$=0.496.

Table 3

Parameters for checking YOLOv4, YOLOv5 models on the Microsoft COCO 2017 validation set

Model	Precision (P)	Recall (r)	Average precision	
			$mAP_{0.5}$	$mAP_{0.5...95}$
YOLOv4	0.716	0.602	0.671	0.454
YOLOv5n	0.582	0.427	0.453	0.271
YOLOv5s	0.68	0.498	0.553	0.359
YOLOv5m	0.71	0.581	0.633	0.439
YOLOv5l	0.721	0.607	0.666	0.476
YOLOv5x	0.729	0.63	0.683	0.496

An example of checking the YOLOv5x model on the Microsoft COCO 2017 validation set using Colab Notebooks in the Tesla T4 15110 MB GPU runtime is shown in Fig. 7.

The YOLOv4 YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l models were checked in the same fashion.

5. 2. Evaluation of the effectiveness of object detection on aerial photographs and video stream with the improved model VisDroneYOLOv5x

For training, validation, and testing the models YOLOv4, YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x, aerial photographs from the VisDrone 2021 set were used.

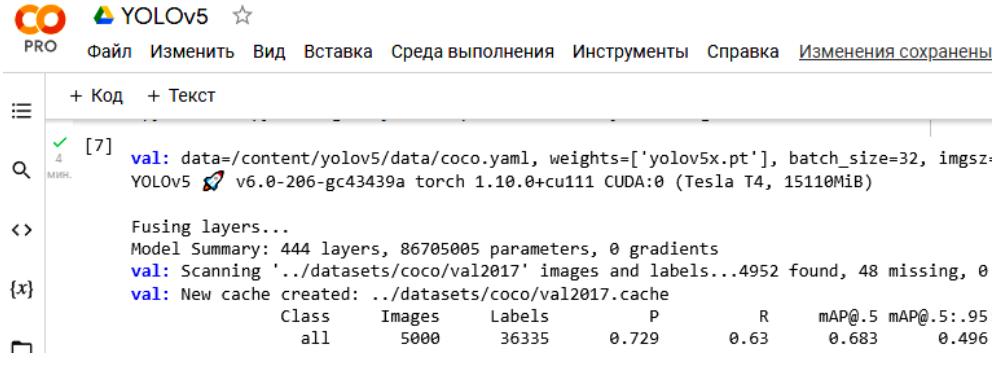


Fig. 7. Example of checking the model YOLOv5x

Training sample – 6,471, validation – 548, test sample – 1,610 aerial photographs of RGB type, JPEG format, dimension 640×640. The total number of classes for object detection on aerial photographs and video stream was 10 (pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, motor).

Research procedure (simulation).

Step 1. Loading (cloning) the YOLOv5 model, checking the PyTorch framework and the GPU.

To train the new dataset, the architecture (code) of the YOLOv5 model is cloned into the Colab Notebooks machine learning simulation environment, PRO version Fig. 8.

```
!git clone https://github.com/ultralytics/yolov5  # clone
%cd yolov5
%pip install -qr requirements.txt  # install

import torch
from yolov5 import utils
display = utils.notebook_init()  # checks
```

Fig. 8. Cloning the YOLOv5x model, checking the PyTorch framework and GPU

In addition, their code can be cloned into other environments (frameworks) of machine learning modeling: Kaggle; DockerHub; Deep Learning Amazon Web Services (AWS); Google Cloud Platform (GCP), and others.

Step 2. Enter training parameters.

Training of the proposed model YOLOv5x was carried out using the optimal values of the parameters, which were obtained experimentally:

– image size – 640 (640×640);
– batch size – 8;
– duration of training (number of epochs) – 25;
– data set – VisDrone;
– model – YOLOv5x.

Enter training parameters (image size, batch size number of epochs, data set, model) Fig. 9.

Default settings:

– initial learning rate – 0.0005;
– optimization algorithm – SGD.

Step 3. Loading the data set VisDrone Fig. 10.

This step also unzips and converts the data.

Step 4. Loading the YOLOv5x model Fig. 11.

```
# Train YOLOv5x on !
!python train.py --img 640 --batch 8 --epochs 25 --data VisDrone.yaml --weights yolov5x.pt --cache
```

Fig. 9. Entering training parameters

```
Dataset not found, missing paths: ['/content/datasets/VisDrone/VisDrone2019-DET-val/images']
Downloading https://github.com/ultralytics/yolov5/releases/download/v1.0/VisDrone2019-DET-train.zip
100% 1.44G/1.44G [00:54<00:00, 28.2MB/s]
Unzipping ../datasets/VisDrone/VisDrone2019-DET-train.zip...
Downloading https://github.com/ultralytics/yolov5/releases/download/v1.0/VisDrone2019-DET-val.zip
100% 77.9M/77.9M [00:00<00:00, 95.9MB/s]
Unzipping ../datasets/VisDrone/VisDrone2019-DET-val.zip...
Downloading https://github.com/ultralytics/yolov5/releases/download/v1.0/VisDrone2019-DET-test-dev.zip
100% 297M/297M [00:03<00:00, 98.3MB/s]
Unzipping ../datasets/VisDrone/VisDrone2019-DET-test-dev.zip...
Downloading https://github.com/ultralytics/yolov5/releases/download/v1.0/VisDrone2019-DET-test-challenge.zip
100% 292M/292M [00:03<00:00, 84.5MB/s]
Unzipping ../datasets/VisDrone/VisDrone2019-DET-test-challenge.zip...
Converting ../datasets/VisDrone/VisDrone2019-DET-train: 6471it [00:40, 158.87it/s]
Converting ../datasets/VisDrone/VisDrone2019-DET-val: 548it [00:04, 120.66it/s]
Converting ../datasets/VisDrone/VisDrone2019-DET-test-dev: 1610it [00:08, 183.80it/s]
Dataset autodownload success, saved to /content/datasets
```

Fig. 10. Downloading the VisDrone data set

```
Downloading https://github.com/ultralytics/yolov5/releases/download/v6.0/yolov5x.pt
100% 166M/166M [00:01<00:00, 133MB/s]

Overriding model.yaml nc=80 with nc=10

                 from  n    params  module                       arguments
  0                -1  1      8800  models.common.Conv           [3, 80, 6, 2, 2]
  1                -1  1    115520  models.common.Conv           [80, 160, 3, 2]
  2                -1  4    309120  models.common.C3             [160, 160, 4]
  3                -1  1    461440  models.common.Conv           [160, 320, 3, 2]
  4                -1  8   2259200  models.common.C3             [320, 320, 8]
  5                -1  1   1844480  models.common.Conv           [320, 640, 3, 2]
  6                -1 12  13125120  models.common.C3             [640, 640, 12]
  7                -1  1   7375360  models.common.Conv           [640, 1280, 3, 2]
  8                -1  4  19676160  models.common.C3             [1280, 1280, 4]
  9                -1  1   4099840  models.common.SPPF           [1280, 1280, 5]
 10                -1  1    820480  models.common.Conv           [1280, 640, 1, 1]
 11                -1  1         0  torch.nn.modules.upsampling  [None, 2, 'nearest']
 12           [-1, 6]  1         0  models.common.Concat         [1]
 13                -1  4   5332480  models.common.C3             [1280, 640, 4, False]
 14                -1  1    205440  models.common.Conv           [640, 320, 1, 1]
 15                -1  1         0  torch.nn.modules.upsampling  [None, 2, 'nearest']
 16           [-1, 4]  1         0  models.common.Concat         [1]
 17                -1  4   1335040  models.common.C3             [640, 320, 4, False]
 18                -1  1    922240  models.common.Conv           [320, 320, 3, 2]
 19          [-1, 14]  1         0  models.common.Concat         [1]
 20                -1  4   4922880  models.common.C3             [640, 640, 4, False]
 21                -1  1   3687680  models.common.Conv           [640, 640, 3, 2]
 22          [-1, 10]  1         0  models.common.Concat         [1]
 23                -1  4  19676160  models.common.C3             [1280, 1280, 4, False]
 24  [17, 20, 23]  1    100935  models.yolo.Detect           [10, [[10, 13, 16, 30
```

Fig. 11. Downloading the YOLOv5x model

The total number of model parameters was 86 million (86,278,375) for the VisDrone dataset.

Step 5. Caching the training and validation data set (Fig. 12).

Step 6. Training the model by the training sample and checking for precision (P), recall (R), mean averaged precision $mAP_{0.5}$, $mAP_{0.5...0.95}$ using the validation sample of 548 images Fig. 13.

For epoch 1/24 precision P=0.524; recall (r) R=0.235; averaged precision $mAP_{0.5}$=0.199, $mAP_{0.5...0.95}$=0.0949).

Step 7. Evaluation of precision, recall, average precision of the model on the validation sample.

As a result, a new model with the proposed name VisDroneYOLOv5x was obtained. The results of checking the precision and recall of this neural network are shown in Fig. 14. Fig. 14 shows that for the model VisDroneYOLOv5x, the greatest precision for the class of car was $mAP_{0.5}$=0.787.

The weight of each category is related to the number of tags shown in Fig. 15 for the VisDrone 2021 set when examining the VisDroneYOLOv5x model.

Fig. 15 shows that for the VisDrone 2021 dataset, the largest number of tags is demonstrated by the class of car, followed by the class of pedestrian.

Fig. 16 shows the deployment (example of object detection) of a test sample by the proposed VisDroneYOLOv5x model.

Fig. 16 demonstrates that even with high saturation of objects, the proposed model VisDroneYOLOv5x copes with the task of detecting 10 classes.

Fig. 17 shows an example of object detection by the VisDroneYOLOv5x model on the video acquired from UAV (the distance to the objects is 2.38 km, the height is 333.4 m, the zoom multiplicity is 30).

The proposed model VisDroneYOLOv5x was compared with the models YOLOv4, YOLOv3. To evaluate the VisDroneYOLOv5x model for convergence, adequacy, and validity, 548 aerial photographs from the VisDrone 2021 set were used as a validation sample.

Convergence. CNN shows convergence provided that with each epoch the error decreases. The convergence of the CNN model is influenced by three components: the completeness of the database (aerial photographs); the correct choice of architecture; selection of CNN training parameters.

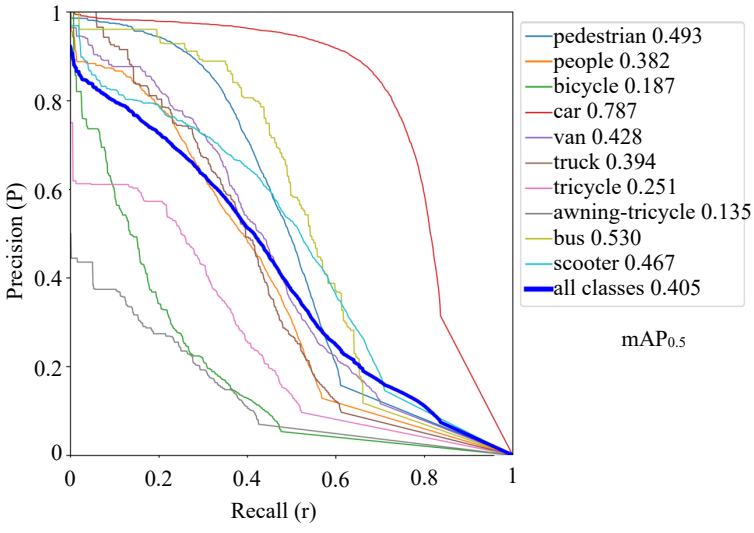


Fig. 14. Plots of changes in precision, recall, on the validation sample for the model VisDroneYOLOv5x

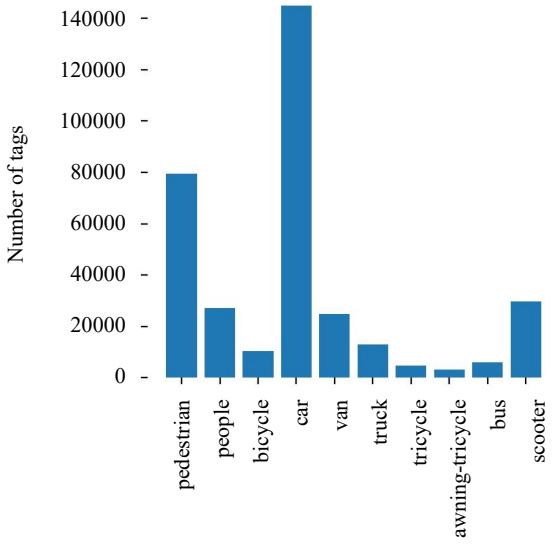


Fig. 15. Number of tags of each category for the VisDrone 2021 dataset when examining the VisDroneYOLOv5x model

```
train: New cache created: /content/datasets/VisDrone/VisDrone2019-DET-train/labels.cache
train: Caching images (5.2GB ram): 100% 6471/6471 [01:13<00:00, 87.99it/s]
val: Scanning '/content/datasets/VisDrone/VisDrone2019-DET-val/labels' images and labels.
val: New cache created: /content/datasets/VisDrone/VisDrone2019-DET-val/labels.cache
val: Caching images (0.4GB ram): 100% 548/548 [00:05<00:00, 101.67it/s]
Plotting labels to runs/train/exp/labels.jpg...
'silent_list' object has no attribute 'patches'
```

Fig. 12. Caching the data set

```
Starting training for 25 epochs...

     Epoch   gpu_mem       box       obj       cls    labels  img_size
      0/24     7.69G    0.1191     0.153   0.04539       673       640: 100% 809/809 [18:01<00:00,
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 35/35
                  all        548      38759      0.367      0.219      0.144     0.0661

     Epoch   gpu_mem       box       obj       cls    labels  img_size
      1/24     7.69G    0.1049    0.1719   0.03364       556       640: 100% 809/809 [17:42<00:00,
                Class     Images     Labels          P          R     mAP@.5 mAP@.5:.95: 100% 35/35
                  all        548      38759      0.524      0.235      0.199     0.0949
```

Fig. 13. Training the model

Fig. 18 shows the estimation of convergence (the dependence of a detection error on the epoch) in the VisDroneYOLOv5x model.

The analysis of Fig. 18 reveals that the proposed model VisDroneYOLOv5x has convergence.

Adequacy. A neural network is adequate if the learning outcomes converge to close values, a necessary condition that there is a dependence between the output and input data that is implemented by the neural network.

The most recommended way to test a neural network model for adequacy is to compare the results with known models.

Fig. 16. An example of object detection in the test sample by the proposed VisDroneYOLOv5x model

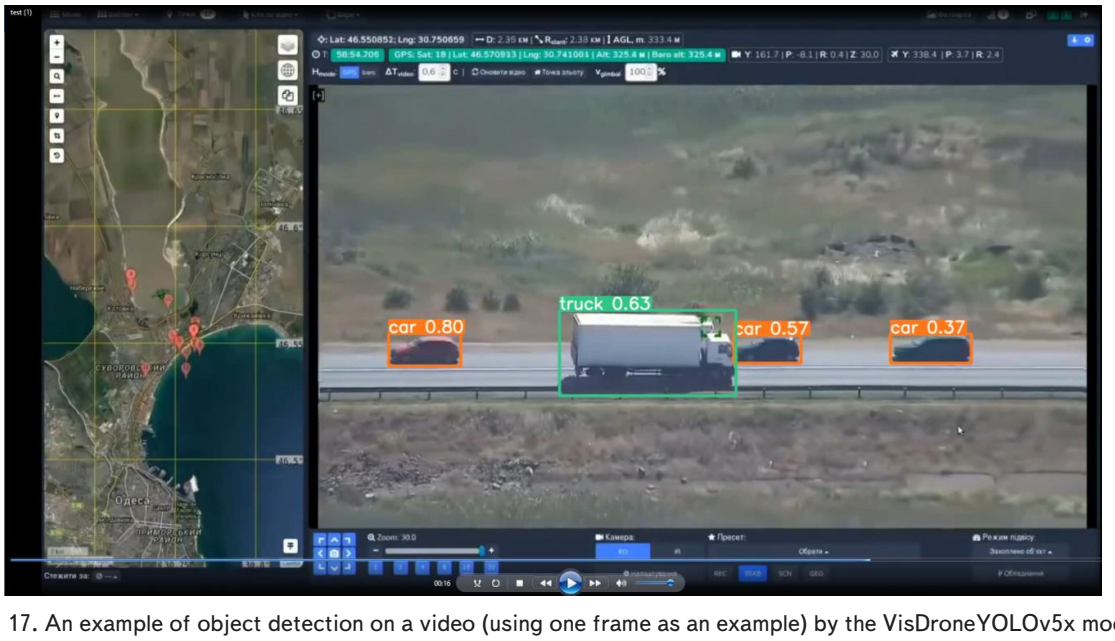


Fig. 17. An example of object detection on a video (using one frame as an example) by the VisDroneYOLOv5x model

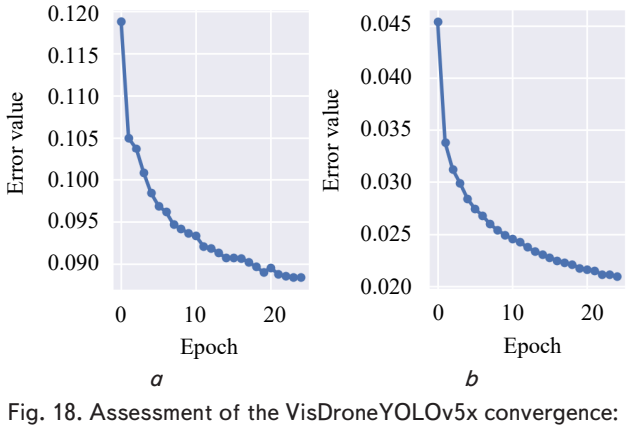


Fig. 18. Assessment of the VisDroneYOLOv5x convergence: *a* – dependence of detection error on epoch for a batch; *b* – dependence of the average detection error on epoch for all (ten) classes

The results of the test on a validation sample (548 aerial photographs) are given in Table 4. Table 4 demonstrates that in comparison with the models YOLOv3, YOLOv4, the proposed model VisDroneYOLOv5x shows the best performance: precision P=0.510; recall r=0.403; average precision $mAP_{0.5}$=0.403, $mAP_{0.5...0.95}$=0.235.

Table 4

Results of model precision and recall assessment

Model	Precision (P)	Recall (r)	Averaged precision	
			$mAP_{0.5}$	$mAP_{0.5...0.95}$
YOLOv3	0.426	0.356	0.352	0.202
YOLOv4	0.472	0.384	0.373	0.211
Proposed model VisDroneYOLOv5x	0.510	0.403	0.403	0.235

6. Discussion of results of studying object detection in aerial photographs and videos using CNN

A study of the efficiency of object detection on aerial photographs (Table 2) using CNN of the following models YOLOv4 (Fig. 5), YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x (Fig. 6) showed that the best indicators for the time of detection of aerial photographs with a size of 1,360×765 are demonstrated by the model YOLOv5n – 0.0096 s. The longest detection time (Table 2) was shown by the YOLOv4 model – 0.166 s. The best indicators of average precision are demonstrated by the YOLOv5x model (Fig. 7) – $mAP_{0.5}$=0.683, $mAP_{0.5...0.95}$=0.496 (Table 3).

It is proposed to use the YOLOv5x CNN (Fig. 11) to detect objects in aerial photographs (Fig. 6) and video (Fig. 17). To increase the efficiency of the neural network, this model was trained (Fig. 13) by the VisDrone set with the selection of optimal parameters (Fig. 9): the duration of training (number of epochs) – 25; batch size – 8; initial learning rate – 0.0005; optimization algorithm – SGD. As a result, a new model with the proposed name VisDroneYOLOv5x was obtained.

The use of the VisDroneYOLOv5x CNN makes it possible to automate the process of object detection on aerial photographs and videos.

Using the proposed model makes it possible to solve the following problems [4–11]:

– computational complexity of object detection on aerial photographs and videos acquired from UAVs;

– the lack of models of neural networks that solve the problem of object detection on aerial photographs and videos.

Limitations of the proposed model include:

– the detection of objects on aerial photographs and videos is carried out within 10 classes;

– the orientation of objects on aerial photographs is not taken into consideration;

– the CNN's broadcast invariance is not taken into consideration.

The limitations of the proposed model are that it is adapted to detect objects in aerial photography and video for ten classes. CNN training was conducted on aerial photographs of high contrast, clarity. For other aerial photographs, the precision and recall of object detection by class may vary, which requires additional copy paste research.

To advance the proposed model, it is planned:

– to increase the base of marked aerial photographs for a training sample [20, 21];

– to explore the proposed and other models (YOLOX, YOLOP, etc.) for different conditions of aerial photography;

– to optimize the proposed model for computational complexity, to increase the speed of performance.

The VisDroneYOLOv5x model is proposed to be used at the ground control point of UAS when processing aerial photographs, orthophoto maps, and videos. In addition, a given model should be used in systems with artificial intelligence; in facility control systems; when designing artificial intelligence in robots; in unmanned vehicle systems.

7. Conclusions

1. The indicators of efficiency of models YOLOv4, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x have been studied. The verification of the effectiveness of these models involved the Microsoft COCO 2017 validation set (the number of images for validation is 5,000). It has been established that the best performance is shown by the YOLOv5x model: $mAP_{0.5}$=0.683, $mAP_{0.5...0.95}$=0.496. The lowest average precision is demonstrated by the YOLOv5n model: $mAP_{0.5}$=0.453, $mAP_{0.5...0.95}$=0.271.

2. The effectiveness of the proposed VisDroneYOLOv5x model based on the VisDrone 2021 set (number of aerial photographs: training sample – 6,471; validation – 548; test sample – 1,610) was evaluated. In comparison with the models YOLOv4, YOLOv5s, YOLOv5m, YOLOv5l, the proposed model produces the best indicators: precision P=0.510; recall r=0.403; average precision $mAP_{0.5}$=0.403, $mAP_{0.5...0.95}$=0.235. The obtained values of the performance indicators of the VisDroneYOLOv5x model allow us to assert the correctness of the choice of the CNN architecture and the selection of its training parameters: the initial learning rate is 0.0005; the duration of training (number of epochs) is 25; the optimization algorithm is SGD.

References

1. Kuznetsova, Y., Somochkin, M. (2021). The concept of creating an intellectual core of an integrated information and analytical system for action in emergencies of man-made nature. Innovative Technologies and Scientific Solutions for Industries, 4 (18), 40–49. doi: https://doi.org/10.30837/itssi.2021.18.040
2. Peng, F., Zheng, L., Cui, X., Wang, Z. (2021). Traffic flow statistics algorithm based on YOLOv3. 2021 International Conference on Communications, Information System and Computer Engineering (CISCE). doi: https://doi.org/10.1109/cisce52179.2021.9445932
3. Bin Zuraimi, M. A., Kamaru Zaman, F. H. (2021). Vehicle Detection and Tracking using YOLO and DeepSORT. 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE). doi: https://doi.org/10.1109/iscaie51753.2021.9431784
4. Fedosov, V. P., Ibadov, S. R., Ibadov, R. R., Kucheryavenko, S. V. (2021). Method For Detecting Violation at a Pedestrian Crossing Using a Convolutional Neuaral Network. 2021 Radiation and Scattering of Electromagnetic Waves (RSEMW). doi: https://doi.org/10.1109/rsemw52378.2021.9494089
5. Sindhu, V. S. (2021). Vehicle Identification from Traffic Video Surveillance Using YOLOv4. 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS). doi: https://doi.org/10.1109/iciccs51141.2021.9432144
6. Kim, J., Koh, J., Lee, B., Yang, S., Choi J. (2021). Video Object Detection Using Object's Motion Context and Spatio-Temporal Feature Aggregation. 2020 25th International Conference on Pattern Recognition (ICPR). doi: https://doi.org/10.1109/icpr48806.2021.9412715
7. Ahmed, A. A., Echi, M. (2021). Hawk-Eye: An AI-Powered Threat Detector for Intelligent Surveillance Cameras. IEEE Access, 9, 63283–63293. doi: https://doi.org/10.1109/access.2021.3074319
8. Zhao, Z.-Q., Zheng, P., Xu, S.-T., Wu, X. (2019). Object Detection With Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems, 30 (11), 3212–3232. doi: https://doi.org/10.1109/tnnls.2018.2876865

9. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: https://doi.org/10.1109/cvpr.2016.91
10. Slyusar, V., Protsenko, M., Chernukha, A., Gornostal, S., Rudakov, S., Shevchenko, S. et. al. (2021). Construction of an advanced method for recognizing monitored objects by a convolutional neural network using a discrete wavelet transform. Eastern-European Journal of Enterprise Technologies, 4 (9 (112)), 65–77. doi: https://doi.org/10.15587/1729-4061.2021.238601
11. Redmon, J., Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: https://doi.org/10.1109/cvpr.2017.690
12. Knysh, B., Kulyk, Y. (2021). Improving a model of object recognition in images based on a convolutional neural network. Eastern-European Journal of Enterprise Technologies, 3 (9 (111)), 40–50. doi: https://doi.org/10.15587/1729-4061.2021.233786
13. Huang, Z., Wang, J., Fu, X., Yu, T., Guo, Y., Wang, R. (2020). DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection. Information Sciences, 522, 241–258. doi: https://doi.org/10.1016/j.ins.2020.02.067
14. Bochkovskiy, A., Wang, C.-Y., Liao, M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv.org. Available at: https://arxiv.org/pdf/2004.10934.pdf
15. Xu, R., Lin, H., Lu, K., Cao, L., Liu, Y. (2021). A Forest Fire Detection System Based on Ensemble Learning. Forests, 12 (2), 217. doi: https://doi.org/10.3390/f12020217
16. Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A. (2014). The Pascal Visual Object Classes Challenge: A Retrospective. International Journal of Computer Vision, 111 (1), 98–136. doi: https://doi.org/10.1007/s11263-014-0733-5 5
17. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D. et. al. (2014). Microsoft COCO: Common Objects in Context. Lecture Notes in Computer Science, 740–755. doi: https://doi.org/10.1007/978-3-319-10602-1_48
18. Zhu, P., Wen, L., Bian, X., Ling, H., Hu, Q. (2018). Vision meets drones: A challenge. arXiv.org. Available at: https://arxiv.org/pdf/1804.07437.pdf
19. Ultralytics Yolov5 and Vision AI. Available at: https://github.com/ultralytics/yolov5
20. Slyusar, V., Protsenko, M., Chernukha, A., Kovalov, P., Borodych, P., Shevchenko, S. et. al. (2021). Improvement of the model of object recognition in aero photographs using deep convolutional neural networks. Eastern-European Journal of Enterprise Technologies, 5 (2 (113)), 6–21. doi: https://doi.org/10.15587/1729-4061.2021.243094
21. Slyusar, V., Protsenko, M., Chernukha, A., Melkin, V., Petrova, O., Kravtsov, M. et. al. (2021). Improving a neural network model for semantic segmentation of images of monitored objects in aerial photographs. Eastern-European Journal of Enterprise Technologies, 6 (2 (114)), 86–95. doi: https://doi.org/10.15587/1729-4061.2021.248390