UDC 629.7.014-519.058.6:528.72:004.032.26 **doi: 10.32620/reks.2024.1.08**

Rostyslav TSEKHMYSTRO, Oleksii RUBEL, Vladimir LUKIN

National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine

STUDY OF METHODS FOR SEARCHING AND LOCALIZING OBJECTS IN IMAGES FROM AIRCRAFT USING CONVOLUTIONAL NEURAL NETWORKS

The use of unmanned and manned aerial vehicles for remote object localization and classification is very common. These methods are used in various systems, ranging from territory surveys to law enforcement. Methods of object localization and classification using neural networks require a detailed study and research of the quality of their work on data that has certain specifics, such as vehicle detection. The use of neural networks to detect certain types of objects using images obtained from aircraft can also help in the study of hard-to-reach locations. Therefore, the ***main subject*** *of this paper is the localization and classification of objects in images obtained using digital cameras mounted on aircraft. The main focus is on determining the accuracy of object localization and detection using selected types of neural networks, which are the most important indicators of neural network efficiency. The speed of a neural network is also an equally important characteristic as it directly affects its ability to be used in tasks that require fast object localization, such as video surveillance or automated car control systems. The main* ***goal of this study*** *is to study the accuracy of object localization and classification in images obtained with the help of cameras mounted on aircraft, as well as to study the speed of neural networks and determine the effectiveness of their application in real-world conditions. The* ***objectives of this study*** *are to train YOLO v5, SSD, and Faster RCNNs on the VisDrone dataset and to further study them on the vehicle localization dataset. The main* ***goal of this*** *work is to obtain statistics on the performance of neural networks trained on the VisDrone dataset. On the basis of the obtained statistics, conclusions are drawn about the effectiveness of the considered neural networks. The* ***conclusions*** *are drawn by considering the speed of the model, localization (IoU), and classification (Precision, Recall) metrics. Possible directions for further development of the topic under study are presented as conclusions.*

Keywords: *object localization; YOLOv5s; SSD; FasterRCNN; vehicle classification; aircraft.*

Introduction

Motivation

The tasks of target (object) detection, localization, parameter estimation, and recognition are classical for radars [1], sonars [2], optical [3], and infra-red [4] systems. Numerous approaches have already been proposed and many efficient systems have already been created and tested. However, the development of new technologies and novel applications opens up new tasks and ways to solve them [1, 2]. In particular, this relates to tasks such as face recognition and law enforcement [5], search of dangerous objects, detection of lawbreakers, and traffic jams on roads [6], etc. For these purposes, it has become popular to exploit images of different origin, including those captured by sensors installed on-board unmanned aerial vehicles (UAV) and drones [7].

Such images usually have a rather large size or it is necessary to process each frame of the video. The number of objects to be detected in such images or frames can be quite large (tens) as well, where it is desired not only to detect and localize objects with appropriate accuracy but also to recognize them. Note that the objects might have different sizes and features. Another problem is that data processing has to be performed quickly enough [1, 2], especially if one deals with video processing.

This set of requirements severely restricts the use of traditional approaches for object detection and localization. Most modern approaches that are currently under design or are already used in localization and classification of objects are based on convolutional neural networks [1, 2]. Meanwhile, there are a huge number of CNNs, and it is difficult to compare their performance for the considered application. It is important to study popular neural networks and compare their performance. This study investigates the use of convolutional neural networks for localization and classification of vehicles and people in color images and video frames.

State-of-the-art

By analyzing modern works on this topic, one can notice the trends set in this area. First, it is common to search for the fastest possible methods [7] that allow

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

them to be used in real time without delays. Note that real-time object detection in UAV-based remote sensing is required in different scenarios [8, 9]. Computational efficiency depends on several factors, including the neural network (NN) architecture (than should not be too complicated) and its realization [10].

Another important trend is the accuracy of neural networks [11], which allows for reliable localization and classification of an object. The problem is that alongside the correct detection of objects under interest, some methods are characterized by a high probability of false detections [12]. Therefore, it becomes necessary to spend additional time and effort to remove such falsely detected objects from further consideration.

It is also worth noting that, when developing such neural networks, as well as when using them, it is necessary to choose a trade-off between speed and accuracy [13], since the most accurate networks are often less fast. This means that possible solutions must be considered from different perspectives.

To determine the accuracy parameters of convolutional neural networks for object localization and classification, quantitative parameters (criteria) must be applied. Parameters such as Intersection Over Union (IoU) [14] and classification accuracy, which is characterised by Precision and Recall [15], are mostly used. In this paper, we consider the parameters of start time and inference time. Referring to the publications [9, 10] that investigated neural networks for localization and classification, the above parameters are sufficient to determine the accuracy of a neural network.

Considering modern works on the topic of research, it is obvious that there are a large number of convolutional neural network (CNN) architectures, which provide a wide choice for each application area. At the same time, most studies provide only a general overview of neural networks for localization and classification based on common datasets. Note that there are no commonly accepted datasets for which training and verification must be performed. Therefore, the use of neural networks for localization and classification in the field of unmanned aerial vehicles requires a deeper study of localization and classification accuracy.

Objectives and the approach

Our study focuses on the following three issues:

– to analyze the applicability of four modern types of CNNs to the detection and localization of typical objects in color images and videos acquired from UAVs with a focus on CNN training and their performance in terms of accuracy;

– to study the time expenses needed for CNN operation under the same conditions;

– to compare CNN in terms of the aforementioned metrics and to make conclusions concerning the type or types of CNN to concentrate on in further studies based on the obtained statistical data.

For this study, we used our own benchmark, which employs the PyTorch library [16], to implement the structure of the considered neural networks. The implementation of the studied neural networks was performed using publications that indicate the appropriate optimizers and loss functions for their training. The most popular datasets obtained from unmanned aerial vehicles at the time of writing were also used in this research.

1. Selecting the dataset for training and testing

The main point in training neural networks is choosing the right dataset for training and validation. The accuracy of object detection and the quality of classification depend on the choice of dataset.

To conduct the research, we chose two datasets that were similar in mark-up but different in images and content. The VisDrone dataset [17] was used for training because it has many images, which are divided into 10 classes of objects, which, as a result of verification and analysis, were combined into 6 main categories, which the model was trained to predict. The resulting categories consist of the following:

- people, which are combined from the primary categories of pedestrians and stationary people;
- car, which combines cars, pickups, and other vehicles;
- buses (bus);
- trucks (truck);
- bicycles (bicycle);
- tricycles (tricycle), which also includes three-wheeled vehicles with tents.

In the images that present statistics and example images, objects are highlighted in different colors as follows: people - purple, cars - dark green, buses – light blue (mabel), trucks - coral, bicycles - light purple, three-wheeled vehicles - red.

In total, the dataset consists of 6471 images for training and 548 images for testing during the training process. In the training part, blocks for 343205 objects belonging to the 6 classes are identified (the distribution by category is shown in Figure 1a). In the test part, 38759 objects were identified, the distribution of which is shown in Figure 1b. Analyzing the obtained statistics, we can see that most of the objects are cars, and a third of all objects are occupied by people. Such indicators fit our task quite well; therefore, this dataset was chosen as the main one for training.

The objects detected in the images are quite different in size, which allows the model to be trained to vary these sizes. This allows the resulting models to be used at different distances from the monitored objects. This makes it possible to keep the aircraft invisible under different conditions without losing the quality of recognition and classification. Examples of images from the test part of the dataset are shown in Figure 2. The colors that reflect the types of objects are indicated above.

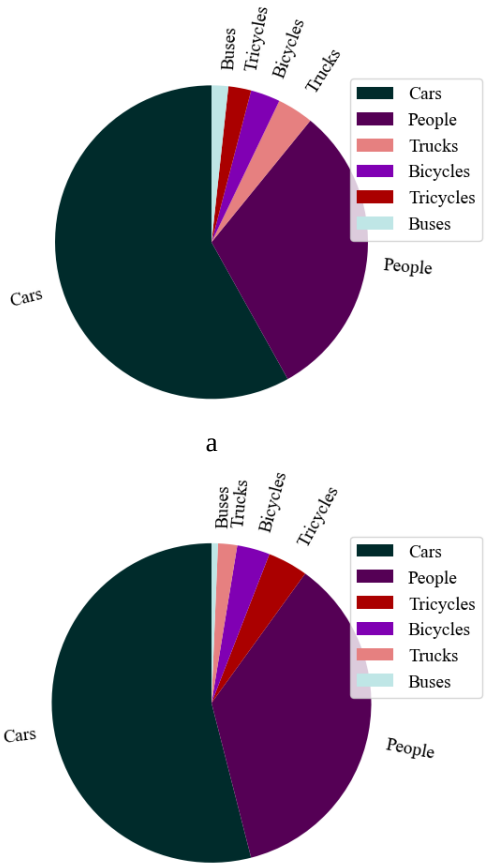

Fig. 1. Distribution of categories in the VisDrone dataset in the training (a) and validation (b) subset

To study the performance of the studied NN, we used a dataset that is also a benchmark for tasks similar to the one described in this article – Unmanned Aerial Vehicle dataset (UAV dataset) [18]. The categories in this dataset were mapped to the list of categories used to train the networks. The dataset [18] has three categories that fit well with the categories described above that trained networks can classify. In particular, the categories are cars, trucks, and buses. The distribution of categories in the dataset is shown in Figure 3.

As one can see from the distribution, the dataset contains only cars, trucks and busses, as the dataset is intended for vehicle search and classification. Therefore, at the stage of model testing, we will also test on a test sample from the VisDrone dataset. Most objects in the dataset are cars. The original version of the dataset contains more than 4000 images; however, for the test part, we selected 1200 images from different subsamples. In total, the test sample contains 130168 objects of different classes.

Fig. 2. Examples of labelled images from the VisDrone dataset

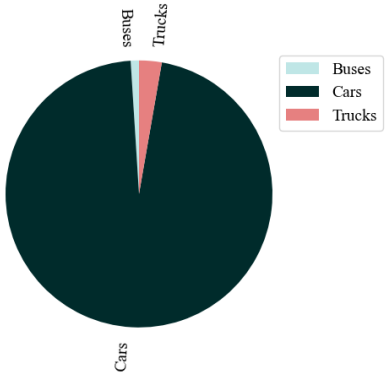

Fig. 3. Distribution of categories in the UAV dataset

Figure 4 shows examples of test images from the UAV dataset, considering the color classification from the training data. The dataset contains a wide variety of test images with different weather and altitudes. This allows for a robust assessment of the model's accuracy under different conditions. The disadvantage of this dataset is insufficient object labelling as the dataset is presented for tracking moving objects; therefore, stationary objects remain unclassified, which is evident in the examples from the dataset in Figure 4.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Fig. 4. Examples of labeled images from the UAV datasets

2. Neural network architectures

To investigate the performance of modern methods for searching and identifying objects in images or video frames, we selected several popular neural networks and built a common infrastructure for the research and training of the selected neural networks. The selected neural networks include the following: SSD, SSDLite [19], YoLo v5 [20], and Faster RCNN [21]. The criteria for choosing these particular networks were as follows:

- the declared accuracy of the network, which is the primary criterion for selection;
- the size of the model, which is also one of the main criteria because the model must work on portable devices; and
- operating time (this is the declared processing time for one image, depending on the device used);
- popularity at the moment and ease of implementation (the network should be easily scalable and can be improved in the future).

All the selected networks are quite popular in modern projects and have relatively the same size, claimed accuracy, and image processing time, which will be studied in more detail in Section 4.

2.1. Single Short Detector and Single Short Detector Lite

To train the Single Shot Detector (SSD) model, we chose the standard implementation from PyTorch and created our own script for training and measuring the performance during training and evaluation. The pre-trained VGG16 (Visual Geometry Group) model [22] was chosen as the backbone because it was used in the original implementation. This model can also use other models as a backbone, such as ResNet [23] and EfficientNet [24]. The backbone structure of the model allows the use of a fleet of different models, but the use of vgg16 as a base is a classic SSD implementation.

SmoothedL1Loss [25] was chosen as the loss function in the training scenario. This loss function uses the quadratic difference if the absolute elemental error is less than beta or the L1 additive otherwise:

$$l_n = \begin{cases} \dfrac{0.5(x_n - y_n)^2}{\text{beta}}, & \text{if } |x_n - y_n| < \text{beta}, \\ |x_n - y_n| - 0.5 * \text{beta}, & \text{otherwise}, \end{cases} \quad (1)$$

where x_n is the true information about the object,

y_n is the predicted information about the object,

beta is the threshold of change between L1 and L2 loss (hyperparameter, non-negative, default is 1.0).

The classification loss function in the training scenario is CrossEntropyLoss [26]:

$$l_n = -\sum_{c=1}^{C} w_c \log \frac{\exp(x_{n,c})}{\sum_{i=1}^{C} \exp(x_{n,i})} y_{n,c}, \quad (2)$$

where x_n is the true classification data for the object,

y_n is the classification vector provided by this method,

w_c is the weight for each class,

C is the number of classes.

CrossEntropyLoss is used for unbalanced datasets and allows the adjustment of weights for classes. The input to this function is a vector of the same length as the number of classes that the model can predict, where each element is a class probability.

To match blocks in the training and metrics evaluation mode, we used SSDMatcher, which uses intersection through union to match object frames. Boxes for which no pair is found are discarded. Once the box pairs are detected, the learning algorithm calculates the loss functions described above. Loss functions are used to model back propagation during the learning process. As an optimizer in the infrastructure, we used the SGD optimizer [27].

The Single Short Detector Lite (SSD Lite) model is a lightweight version of the SSD, which is described above. This model was trained to obtain data on the accuracy and speed of this method. Unlike SSD, this method uses the MobileNet_v3_large model [28]. It is faster than VGG16 and has fewer parameters. Of course, this reduces the accuracy of the model in different tasks.

2.2. Faster Regional Convolutional Neural Network

Faster Regional Convolutional Neural Network (Faster RCNN) [21] is a continuation of the Regional Convolutional Neural Network (RCNN) family of models. The first model is a simple Regional

Convolutional Neural Network (RCNN) [29]. Models of this structure use an algorithm to generate regions in which an object is likely to be present. Mostly, they use a selective search algorithm to identify regions of interest (ROI), and these regions are represented as frames for the image. In the initial implementation, the algorithm generated two thousand ROIs. Subsequently, the generated regions are fed to the feature extraction network and the classification subnetwork. The classification subnetwork acts as a thresholding processor by removing regions with no objects.

The next model is Fast RCNN [30]. This model, like the previous one, creates a proposal of probable regions using a Regional Proposal Network (RPN) based on a selective search algorithm. What makes this model different from the previous model is that the network is run for the full image and cuts off the proposed blocks from the network, reshaping and classifying them. This method is called ROIPooling.

The third generation of RCNN models is Faster RCNN. The structure of the model is very different from that of the previous models, but the algorithm is very similar. In this model, the Regional Proposal Network (RPN) is integrated into the neural network, which speeds up this model up to 10 times. In this case, the RPN uses a feature map to create suggestion regions. In our implementation, the mobilenet_v3_large network is used as the algorithm for extracting features from the image.

To train the Faster RCNN, we used a smoothed L1 (1) loss function, which is identical to that used for SSD and SSDLite. This loss function was used twice: the first time as a loss function for the predicted blocks and the second time as a loss function for the region supply network. This model also uses a loss function for classification and objectivity. The classification loss is the cross-entropy according to the SSD model. The objectivity loss function is calculated for the RPN part, which reflects the number of accurately predicted regions regardless of the predicted class. This loss function is a binary cross-entropy function with logits:

$$l_n = -w_n[y_n \log\sigma(x_n) + (1-y_n)\log(1-\sigma(x_n))], \quad (3)$$

where x_n is the input classification vector,

y_n is the true classification vector,

$\sigma(x_n)$ is the probability of each class.

2.3. YOLO v5

YOLO (you only look once) is a family of models based on one-step regression. The architecture of YOLO v5 [20] is based on the updated Darknet layers used in YOLO v3 [31]. For our task, we chose the smallest version of the model from the official git repository.

To train YOLO, we used binary cross-entropy (BCE) with logit loss as a loss function, which combined the sigmoid layer and BCE loss in one class. This loss function is used to classify and measure the performance of the objects. The intersection of union (IoU) [14] is used for regression and box sorting, and the resulting predictions are used to merge the target and predicted boxes. The BCE loss for object detection accuracy is represented by the object prediction score, which is a binary metric that does not pay attention to class. The loss of BCE for classification is shown by the object category prediction accuracy score. Equation 3 shows the original formula for measuring the binary cross-entropy loss:

3. Neural networks training

For the neural network training process, we used the implementations described in the previous section. Each network was trained for 300 epochs, and during the training process, metrics were calculated that reflect the accuracy of the model and its progress in training. The resulting metrics were calculated for the loss functions also described for each of the neural networks. This training process allows tracking the accuracy at each stage of training and selecting the model checkpoints that are as accurately as possible.

To train SSD model, we set the learning rate to 1×10^{-3} and the weight decay to 5×10^{-4}. This neural network was validated every epoch, and its loss functions were calculated. The data obtained are presented in Figures 5, 6 (one step-one epoch). Figure 5 shows the dependency of the regression loss function versus the validation step. Figure 6 shows the dependency of the classification logarithm versus the validation step with parameters similar to the loss function for boxes.

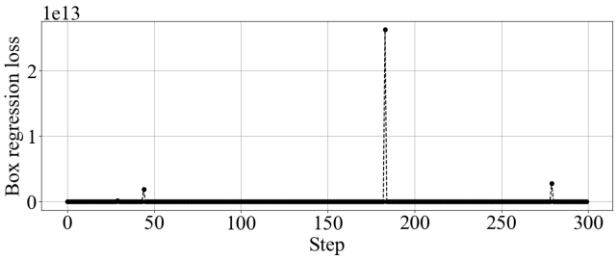

Fig. 5. Dependence of the regression function of losses for boxes on the validation step for SSD network

To train the SSD Lite network, we used a training process similar to that for SSD, which is described above. Figure 7 shows a plot of the regression loss function versus the validation step (1 validation step per epoch). Figure 8 shows a graph of the classification

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

logarithm versus the validation step with parameters similar to the loss function for boxes.

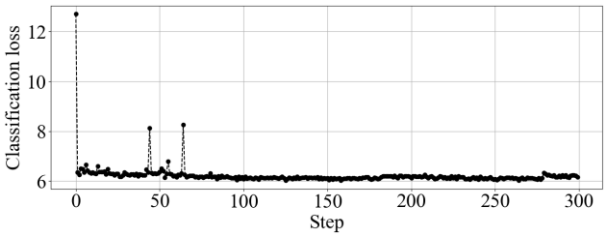

Fig. 6. Dependence of the classification loss for boxes on the validation step for SSD network

Analyzing the data obtained, it is noticeable that the reduced backbone affects the quality and performs better than the enlarged version for this task. It is also noticeable on the speed of operation, the analysis of which will be presented in Section 4.

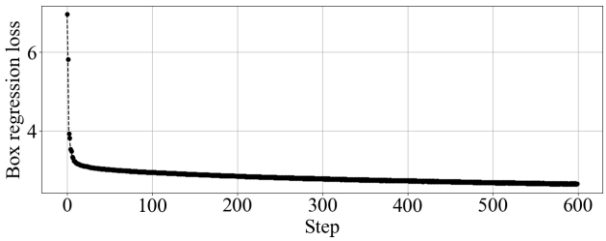

Fig. 7. Dependence of the regression function of losses for boxes on the validation step for SSDLite network

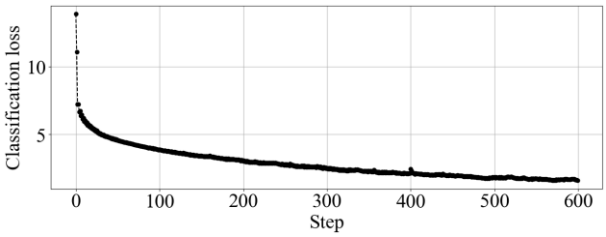

Fig. 8. Dependence of the classification logarithm for boxes in the validation step for SSDLite network

To train Faster RCNN (network structure described on 2.2), we used a stochastic gradient descent (SGD) with a learning rate of 1×10^{-3} and a weight decay of 5×10^{-4}. The network was trained for 300 epochs and validated twice for each epoch. Figure 9 shows a graph of the regression loss function for the blocks from the regional offer network, and Figure 10 shows a similar loss dependence for the predicted blocks. Figure 11 shows the dependence of the loss function for the objectivity from the RPN, and Figure 12 presents the dependence of the classification loss for the predicted values.

When analyzing the obtained values, we can conclude that the investigated method is more accurate than the previous methods (SSD and SSD Lite networks) and has better convergence of the loss function. There is also a sharp decrease in most of the loss functions, which reflects the model's fairly good ability to learn on the selected dataset.

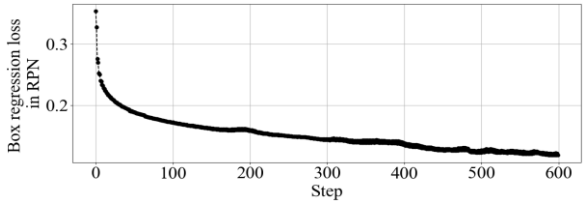

Fig. 9. Dependence of regression losses for predicted frameworks in RPN on the validation step for Faster RCNN network

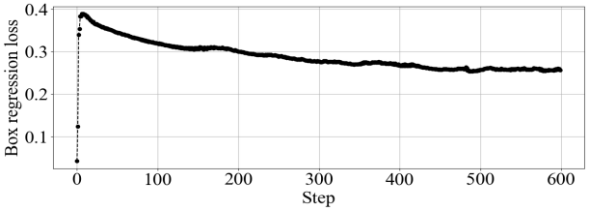

Fig. 10. Dependence of the regression losses for the predicted framework for validation step for Faster RCNN network

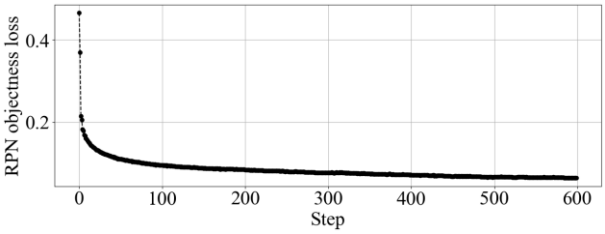

Fig. 11. Dependence of the loss function for predicted objects in RPN during the validation step for Faster RCNN network

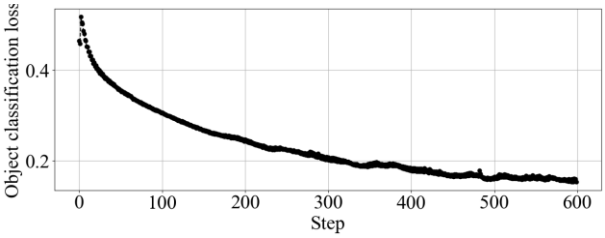

Fig. 12. Dependence of object classification loss function in the validation step for Faster RCNN network

For training YOLO v5 model, we used the Adam complex optimizer [32] with a learning rate of 0.01. No weight decay was used for the bias and normalization layers, and weight decay with a value of 1×10^{-5} was used for the batch normalization layers. The model was trained for 300 epochs, and each epoch was evaluated using metrics. Figure 13 shows the dependence of the loss function for object detection accuracy for the trained model, and Figure 14 shows the dependence of the loss function for classification on the epoch.

When analyzing the obtained dependencies of the loss functions on the training step, we observe a sharp drop in the values of the loss function, which reflects the ability of the model to learn on the selected dataset, similar to the Faster RCNN model, which, judging by the results, is competitive with the networks studied in this subsection. In addition, the loss functions converge quite well until the last stages, where they show almost identical values, which means that the convergence function has reached a plateau. The obtained results of the loss function are the smallest of the studied NN variants.

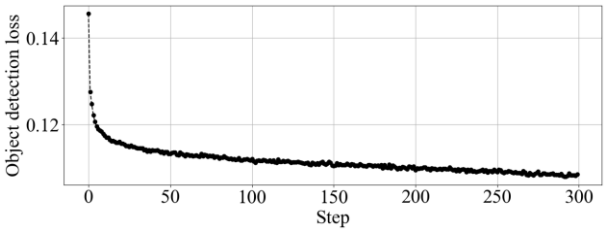

Fig. 13. Dependence of the loss function for object detection accuracy on the epoch of YOLO v5

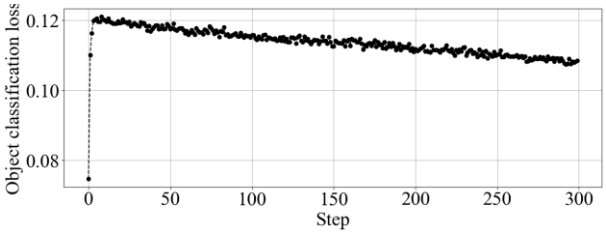

Fig. 14. Dependence of the loss function for object classification accuracy on the epoch of YOLO v5

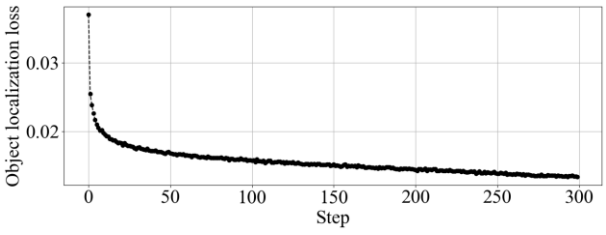

Fig. 15. Dependence of the object localization loss function on the epoch of YOLO v5

4. Analysis of the results

4.1. Determining localization accuracy metrics

The metric most commonly used to reflect the accuracy of object localization using neural networks is intersection over union (IoU [14). This metric calculates the ratio of the intersection area of the predicted block and that marked in the process of creating the dataset to their total area. The data are displayed in a percentage gradation, where 1 is the most similar blocks and 0 is the blocks that do not intersect at all.

This relationship can be represented by the following formula:

$$\mathrm{IoU} = \frac{|\mathrm{A} \cap \mathrm{B}|}{|\mathrm{A} \cup \mathrm{B}|}, \tag{4}$$

where A is the coordinates of the predicted block,

B is the coordinates of the marked block.

Using the previously obtained models, we evaluated this metric for each model and visually interpreted the results using the graph presented in Figure 17. The data obtained are also presented in Table 1 for comparison by numerical parameters.

Analyzing the obtained localization accuracy data, we can see that YOLOv5 shows the best result among the studied networks, and the FasterRCNN network has a fairly similar result. The structure of the SSD models shows a rather poor result according to the IoU metric, which indicates that it is inappropriate to use these models in real-life conditions for the studied task.

4.2. Determination of the object classification accuracy metrics

To determine the accuracy of the object classification (or objectness), we used the UAV dataset, in which all labelled data were combined into one class. This type of accuracy determination is useful for models that do not need to be accurately classified into multiple types. In the UAV dataset, only vehicles are labeled, therefore, when calculating classification accuracy parameters, classes can be neglected.

In this test, we measured the objectivity metrics for the trained Precision and Recall models [15]. These indicators reflect the accuracy of the prediction regardless of the class. For each model, we calculated Precision and Recall. Precision shows the proportion of relevant instances among the retrieved instances, and Recall shows the proportion of relevant instances that were retrieved. In mathematical terms, these metrics can be expressed as (5) and (6):

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

$$P = \frac{TP}{FP - TP}, \quad (5)$$

where TP is the number of correctly predicted classes,
FP is the number of negatively predicted classes.

$$R = \frac{TP}{FN - TP}, \quad (6)$$

where TP is the number of correctly predicted classes,
FN is the number of negatively predicted classes that should be positive.

Because this metric is measured for a single class in this case, TPs are represented as objects that match the target objects. FP are objects that do not match the target objects. Finally, FNs are objects that do not have a pair in the target set. The intersection over union metric (IoU [14]) with a threshold of 0.5 was used to match objects.

The accuracy and memorization rates are shown in Table 1.

Table 1
Accuracy, recall, and IoU metrics for the trained models

Metric	SSDLite	SSD	FasterRCNN	Yolo v5s
Precision	0.048	0.349	0.528	0.540
Recall	0.0059	0.055	0.627	0.928
IoU	0.01	0.03	0.645	0.672

Analyzing the results, we can conclude that the SSDLite and SSD models predict many objects that cannot be compared with the target objects. In addition, these models have poor prediction speed for the target objects. Faster RCNNs have fairly good accuracy rates, which show that the model predicts more than 50% of the objects accurately, but it generates slightly less than 40% of the objects that are not represented in the target set. The best performance in this case was shown by the Yolo v5s model. It predicts more than 50% of the objects correctly and has some false positive objects (less than 8% of bad cases). Figure 16 shows the visual results for all models. In the image, green rectangles show the objects that are matched to the target objects, and red rectangles show the objects that are not matched to the target objects. Figure 17 shows a visual representation of the accuracy and recall scores.

4.3. Comparison of the inference time

Speed is a critical metric for wrapper models. Comparing the inference time is used to select a model for different tasks, choosing the best option between the best accuracy or speed. To measure the time, we used 100 generated arrays of size 3×640×640 representing RGB images. We also measured the start time because this parameter is also important for using the model. To measure the time, we used PyTorch with NVidia CUDA support, and all measurements were performed on an NVIDIA GTX1660 GPU with an Intel i9-10900 processor and 32 GB of DDR4 RAM. Table 2 shows the results of this comparison. Analyzing the results, we can conclude that YOLO v5s is up to 5 times faster than other models, which is another advantage of this model.

Conclusions

On the basis of the results obtained in the process of training neural networks and at the stage of their testing, we have obtained estimates of the effectiveness of each model and the feasibility of their application in real systems for the studied problem. Thus, during training, it is noticeable that the YOLO v5 and Faster RCNN networks show better results according to all metrics compared with networks with the SSD structure. During the testing process, we determined the recall and accuracy metrics for each of the obtained networks. In this sense, YOLO v5 and Faster RCNN are also favorites although YOLO v5 shows slightly better results. We also measured the localization metric (IoU) and proved that the best results were observed for the same network. Based on the data obtained, as well as using the statistics on the processing time of one image, which is much shorter for the YOLO v5 model, it can be argued that the YOLO v5 model is the most efficient for the task under study.

The results obtained in this study are sufficient to consider YOLO v5 as the best neural network for localizing and classifying objects in images taken from unmanned aerial vehicles among the studied neural networks. Metrics are also presented, and their analysis allows you to choose the best neural network for your own needs.

In the future, it is also advisable to consider these models in the context of imperfect or poor image quality, which is caused by distortions due to various factors [33, 34], such as low sensor resolution or compression of visual data [35], for faster transmission by communication. It is also worth considering datasets for narrower training, e.g., only for transport detection.

Authors' contributions: conception – **Oleksii Rubel**, **Rostyslav Tsekhmystro**; methodology – **Oleksii Rubel**, **Rostyslav Tsekhmystro**; problem formulation – **Oleksii Rubel**; analysis – **Rostyslav Tsekhmystro**; model development – **Rostyslav Tsekhmystro**; software – **Rostyslav Tsekhmystro**; validation – **Vladimir Lukin**, **Oleksii Rubel**; analysis of results – **Rostyslav Tsekhmystro**, **Oleksii Rubel**, **Vladimir Lukin**; visualization – **Rostyslav Tsekhmystro**; writing – **Rostyslav Tsekhmystro**; revision and editing – **Oleksii Rubel**, **Vladimir Lukin**.

a b c d

Fig. 16. Prediction results for SSDLite (a), SSD (b), Faster RCNN (c), and Yolo v5s (d)

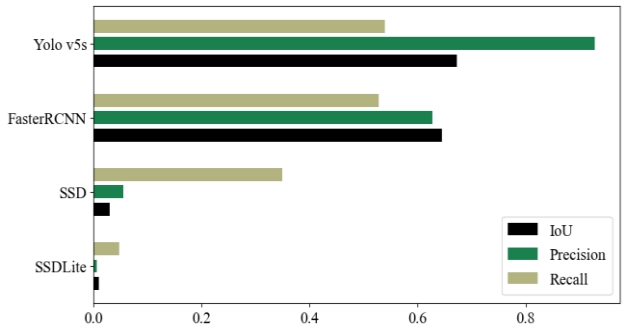

Fig. 17. Results (metric values) for the trained model

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was conducted without financial support.

Data availability

The manuscript contains no associated data.

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

Table 2

Comparison of inference and start times for trained models (in seconds)

Metric	SSDLite	SSD	FasterRCNN	YOLO v5s
Initialization time, s	0.705	2.37	0.720	4.088
First prediction time, s	0.665	0.661	0.651	0.149
Average inference time, s	0.029	0.038	0.040	0.011

Use of Artificial Intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

All authors have read and agreed to the published version of this manuscript.

References

1. Wang, L., Tang, J., & Liao, Q. A Study on Radar Target Detection Based on Deep Neural Networks. *IEEE Sensors Letters*, 2019, vol. 3, no. 3, article no. 7000504. DOI: 10.1109/LSENS. 2019.2896072.

2. Yu, S. Sonar Image Target Detection Based on Deep Learning. *Mathematical Problems in Engineering*, 2022, vol. 2022, article no. 5294151. DOI: 10.1155/2022/5294151.

3. Bondžulić, B., Stojanović, N., Lukin, V., Stankevich, S. A., Bujaković, D., & Kryvenko, S. Target acquisition performance in the presence of JPEG image compression. *Defence Technology*, 2023. DOI: 10.1016/j.dt.2023.12.006.

4. Zhao, M., Li, W., Li, L., Hu, J., Ma, P., & Tao, R. Single-Frame Infrared Small-Target Detection: A survey. *IEEE Geoscience and Remote Sensing Magazine*, 2022, vol. 10, no. 2, pp. 87-119. DOI: 10.1109/MGRS.2022.3145502.

5. Lei, J., Lay, T., Weiland, C., & Lu, C. Combination of Spatiotemporal ICA and Euclidean Features for Face Recognition. Artificial Intelligence in Theory and Practice. *IFIP AI 2006. IFIP International Federation for Information Processing*, 2006, vol. 217, pp. 395-403. DOI: 10.1007/978-0-387-34747-9_41.

6. *Ford Blue Cruise Version 1.2 Hands-Off Review: More Automation, Improved Operation.* Available at: https://www.motortrend.com/reviews/ford-bluecruise-version-1-2-first-drive-review/. (accessed 5 Jan. 2024).

7. Cao, Z., Kooistra, L., Wang, W., Guo, L. & Valente, J. Real-Time Object Detection Based on UAV Remote Sensing: A Systematic Literature Review. *Drones*, 2023, no. 7, article no. 620. DOI: 10.3390/drones7100620.

8. Feng, J. & Yi, C. Lightweight Detection Network for Arbitrary-Oriented Vehicles in UAV Imagery via Global Attentive Relation and Multi-Path Fusion. *Drone*, 2022, vol. 6, no. 5, article no. 108. DOI: 10.3390/drones6050108.

9. Alsamhi, S. H., Shvetsov, A. V., Kumar, S., Shvetsova, S. V., Alhartomi, M. A., Hawbani, A., Rajput, N. S., Srivastava, S., Saif, A., & Nyangaresi, V. O. UAV Computing-Assisted Search and Rescue Mission Framework for Disaster and Harsh Environment Mitigation, *Drones*, 2022, vol. 6, no. 7, article no. 154. DOI: 10.3390/drones6070154.

10. Aposporis, P. Object Detection Methods for Improving UAV Autonomy and Remote Sensing Applications. *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2020, pp. 845-853. DOI: 10.1109/ASONAM49781.2020.9381377.

11. Zhao, C., Liu, R.W., Qu, J., & Gao, R. Deep Learning-Based Object Detection in Maritime Unmanned Aerial Vehicle Imagery: Review and Experimental Comparisons. *Engineering Applications of Artificial Intelligence*, 2024, vol. 128, article no. 107513. DOI: 10.1016/j.engappai.2023.107513.

12. Kong, M., Roh, M., Kim, Lee, J., Kim, J., & Lee, G. Object detection method for ship safety plans using deep learning. *Ocean Engineering*, 2022, vol. 246, article no. 110587. DOI: 10.1016/j.oceaneng. 2022.110587.

13. Lyu, M., Zhao, Y., Huang, C., & Huang H. Unmanned Aerial Vehicles for Search and Rescue: A Survey. *Remote Sensing*, 2023, no. 15, article no. 3266. DOI: 10.3390/rs15133266.

14. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 658-666. DOI: 10.48550/arXiv.1902.09630.

15. Ting, K. M. Precision and Recall. *Encyclopedia of Machine Learning*, 2010. 781 p. DOI: 10.1007/978-0-387-30164-8_652.

16. *PYTORCH DOCUMENTATION*. Available at: https://pytorch.org/docs/stable/index.html#pytorch-documentation. (accessed 5 Jan. 2024).

17. Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., & Ling, H. Detection and Tracking Meet Drones Challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022, vol. 44, no.

11, pp. 7380-7399. DOI: 10.1109/TPAMI. 2021.3119563.

18. Du, D., Qi, Y., Yu, H., Yang, Y., Duan, K., Li, G., Zhang, W., Huang, Q., & Tian, Q. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. *European Conference on Computer Vision (ECCV)*, 2018, vol. 128, pp. 1141-1159. DOI: 10.1007/s11263-019-01266-1.

19. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., & Berg, A. SSD: Single Shot MultiBox Detector. *Lecture Notes in Computer Science*, 2016, vol. 9905, pp. 21-37. DOI: 10.1007/978-3-319-46448-0_2

20. *YOLOv5 by Ultralytics*. Available at: https://github.com/ultralytics/yolov5. (accessed 5 Jan. 2024).

21. Shaoqing, R., Kaiming, H., Girshick, R., & Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, vol. 39, pp. 1137-1149. DOI: 10.1109/TPAMI.2016.2577031.

22. Simonyan, K., & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations (ICLR 2015)*, 2015, pp. 1-14. DOI: 10.48550/arXiv.1409.1556.

23. He, K., Zhang, X., Ren, S., & Sun, J. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778. DOI: 10.1109/CVPR.2016.90.

24. Tan, M., & Le, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 6105-6114. DOI: 10.48550/arXiv.1905.11946.

25. *SMOOTHL1LOSS*. Available at: https://pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html. (accessed 5 Jan. 2024).

26. *CROSSENTROPYLOSS*. Available at: https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html. (accessed 5 Jan. 2024).

27. Sutskever, I., Martens, J., Dahl, G. & Hinton, G. On the importance of initialization and momentum in deep learning. *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 1139-1147.

28. Howard, A., Sandler, M., Chu, G., Chen, L., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V. & Le, Q. Searching for MobileNetV3. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314-1324. DOI: 10.1109/ICCV.2019.00140.

29. Girshick, R., Donahue, J., Darrell, T., & Malik, J. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, pp. 142-158. DOI: 10.1109/TPAMI.2015.2437384.

30. Girshick, R. Fast R-CNN. *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440-1448. DOI: 10.1109/ICCV.2015.169.

31. Redmon, J. YOLOv3: An Incremental Improvement, 2018. DOI: 10.48550/arXiv.1804.02767. (unpablished).

32. Kingma, D. P., & Ba, J. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, 2014. DOI: 10.48550/arXiv.1412.6980.

33. Rubel, A., Rubel, O., Tsekhmystro, R., Rebrov, V., & Lukin V. Automatic Decision Undertaking on Expedience of Image Denoising Based on Filter Efficiency Prediction. *Proceedings of ISSOIA Conference*, 2022, pp. 504-524. DOI: 10.1007/978-981-99-4098-1_44.

34. Tsymbal, O. V., Lukin, V. V., Ponomarenko, N. N., Zelensky, A. A., Egiazarian, K. O., & Astola, J. T. Three-state Locally Adaptive Texture Preserving Filter for Radar and Optical Image Processing. *EURASIP Journal on Applied Signal Processing*, 2005, no. 8, pp. 1185-1204. DOI: 10.1155/ASP.2005.1185.

35. Proskura, G. A., Rubel, O. S., & Lukin, V. V. On classifier learning methodologies with application to compressed remote sensing images. *Radioelectronic and Computer Systems*, 2022, no. 3, pp. 174-189. DOI: 10.32620/reks.2022.3.13.

Received 13.01.2024, Accepted 20.02.2024

ДОСЛІДЖЕННЯ МЕТОДІВ ПОШУКУ ТА ЛОКАЛІЗАЦІЇ ОБ'ЄКТІВ НА ЗОБРАЖЕННЯХ З ЛІТАЛЬНИХ АПАРАТІВ ЗА ДОПОМОГОЮ ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖ

Ростислав Цехмистро, Олексій Рубель, Володимир Лукін

Використання безпілотних та пілотованих літальних апаратів для задач дистанційної локалізації та класифікації об'єктів є дуже поширеним в наш час. Зазначені методи використовуються в різних за сферою застосування системах, від досліджень територій до забезпечення правопорядку. Методи локалізації та класифікації об'єктів за допомогою нейронних мереж потребують детального вивчення та дослідження

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

якості їх роботи на даних, які мають визначену специфікацію, наприклад детектування транспорту. Використання нейронних мереж для детектування окремих типів об'єктів за допомогою зображень, що отримані з літальних апаратів може допомогти також в проблемах дослідження важкодоступних локацій. Саме тому основною **темою** даної роботи є локалізація та класифікація об'єктів на зображеннях, що отримані за допомогою камер, які встановлені на літальних апаратах. Основну **увагу привернуто** до визначення точності локалізації та детектування об'єктів за допомогою обраних нейронних мереж, що є найважливішим показником в ефективності нейронної мережі. Також не менш важливою оцінкою є швидкість роботи нейронної мережі, адже це напряму впливає на можливість її використання в задачах, де потрібна швидка локалізація об'єкту, наприклад відеоспостереження чи системи автоматичного керування автомобілем. Основною **метою роботи** є дослідження точності локалізації та класифікації об'єктів на зображеннях, отриманих за допомогою камер що встановлені на літальних апаратах, а також дослідження швидкості роботи нейронних мереж та визначення ефективності їх застосування в реальних умовах. **Задачами** роботи є навчання YOLO v5, SSD та Faster RCNN на наборі даних VisDrone та їх подальше дослідження на наборі даних для локалізації техніки. Основний **результат роботи** - отримання статистики роботи нейронних мереж, що навчені на наборі даних VisDrone. На основі отриманої статистики надані висновки щодо ефективності застосування отриманих нейронних мереж. **Висновки** зроблені з врахуванням швидкості роботи моделі, метрик по локалізації (IoU) та класифікації (Precision, Recall). Також в якості висновків надано можливі напрямки подальшого розвитку досліджуваної теми.

Ключові слова: локалізація об'єктів; YOLOv5s, SSD, FasterRCNN, класифікація техніки, літальні апарати.

Цехмистро Ростислав Вікторович – асп. каф. інформаційно-комунікаційних технологій ім. О. О. Зеленського, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Рубель Олексій Сергійович – канд. техн. наук, доц. каф. інформаційно-комунікаційних технологій ім. О. О. Зеленського, Національний аерокосмічний університет ім. М. Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Лукін Володимир Васильович – д-р техн. наук, проф., зав. каф. інформаційно-комунікаційних технологій ім. О. О. Зеленського, Національний аерокосмічний університет ім. М.Є. Жуковського «Харківський авіаційний інститут», Харків, Україна.

Rostyslav Tsekhmystro – PhD Student of the Department of Information and Communication Technologies named after O. O. Zelensky, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine,
e-mail: rostyslav.tsekhmystro@gmail.com, ORCID: 0000-0002-1515-7065.

Oleksii Rubel – Candidate of Technical Science, Associate Professor at the Department of Information-Communication Technologies named after O. O. Zelensky, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine,
e-mail: s.rubel@khai.edu, ORCID: 0000-0001-6206-3988.

Vladimir Lukin – Doctor of Technical Sciences, Professor, Head of the Department of Information-Communication Technologies named after O. O. Zelensky, National Aerospace University "Kharkiv Aviation Institute", Kharkiv, Ukraine,
e-mail: v.lukin@khai.edu, ORCID: 0000-0002-1443-9685.