http://doi.org/10.35784/iapgos.5862

received: 03.02.2024 | revised: 07.05.2024 | accepted: 19.06.2024 | available online: 30.06.2024

APPLICATION OF RESNET-152 NEURAL NETWORKS TO ANALYZE IMAGES FROM UAV FOR FIRE DETECTION

Nataliia Stelmakh[1], Svitlana Mandrovska[1], Roman Galagan[2]

[1]Department of Computer-Integrated Technologies of Device Production, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine
[2]Department of Automation and Non-Destructive Testing Systems, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine

***Abstract**. Timely detection of fires in the natural environment (including fires on agricultural land) is an urgent task, as their uncontrolled development can cause significant damage. Today, the main approaches to fire detection are human visual analysis of real-time video stream from unmanned aerial vehicles or satellite image analysis. The first approach does not allow automating the fire detection process and contains a human factor, and the second approach does not allow detect the fire in real time. The article is devoted to the issue of the relevance of using neural networks to recognize and detect seat of the fire based on the analysis of images obtained in real time from the cameras of small unmanned aerial vehicles. This ensures the automation of fire detection, increases the efficiency of this process, and provides a rapid response to fires occurrence, which reduces their destructive consequences. In this paper, we propose to use the convolutional neural network ResNet-152. In order to test the performance of the trained neural network model, we specifically used a limited test dataset with characteristics that differ significantly from the training and validation dataset. Thus, the trained neural network was placed in deliberately difficult working conditions. At the same time, we achieved a Precision of 84.6%, Accuracy of 91% and Recall of 97.8%.*

Keywords: UAV, neural network, ResNet-152, computer vision, artificial intelligence, fire detection, firefighting

ZASTOSOWANIE SIECI NEURONOWYCH RESNET-152 DO ANALIZY OBRAZÓW Z UAV DO WYKRYWANIA POŻARU

***Streszczenie.** Wczesne wykrycie pożarów w środowisku naturalnym (w tym pożarów na gruntach rolnych) jest zadaniem pilnym, gdyż ich niekontrolowany rozwój może spowodować znaczne szkody. Obecnie głównymi podejściami do wykrywania pożarów jest wizualna analiza przez człowieka strumienia wideo w czasie rzeczywistym z bezzałogowych statków powietrznych lub analiza obrazu satelitarnego. Pierwsze podejście nie pozwala na automatyzację procesu wykrywania pożaru i uwzględnia czynnik ludzki, natomiast drugie podejście nie pozwala na wykrycie pożaru w czasie rzeczywistym. Artykuł poświęcony jest zagadnieniu przydatności wykorzystania sieci neuronowych do rozpoznawania i wykrywania źródła pożaru na podstawie analizy obrazów uzyskiwanych w czasie rzeczywistym z kamer małych bezzałogowych statków powietrznych. Zapewnia to automatyzację wykrywania pożaru, zwiększa efektywność tego procesu oraz zapewnia szybką reakcję na wystąpienie pożarów, co ogranicza ich niszczycielskie skutki. W artykule proponujemy wykorzystanie splotowej sieci neuronowej ResNet-152. Aby przetestować wydajność wyszkolonego modelu sieci neuronowej wykorzystaliśmy ograniczony testowy zbiór danych, którego charakterystyka znacznie różni się od zbiorów danych treningowych i walidacyjnych. Tym samym wytrenowana sieć neuronowa została poddana celowo trudnym warunkom operacyjnym. Jednocześnie uzyskano parametry "Precision" – 84.6%, "Accuracy" – 91% i "Recall" – 97.8%.*

Słowa kluczowe: UAV, sieć neuronowa, ResNet-152, wizja komputerowa, sztuczna inteligencja, detekcja pożaru

Introduction

The rapid development of technology in the modern world is increasing the efficiency of small-sized unmanned aerial vehicles (UAV), making it possible to automate many processes and thereby expand the range of their applications. Small-sized UAVs are widely used in civil, military, research and development sectors. The definition of "unmanned aerial vehicle" covers all vehicles that can fly without the physical presence of a human pilot on board. The UAV flight can be controlled remotely from the ground or can be performed autonomously according to a given algorithm without direct human intervention [5].

One of the important areas of application of small-sized UAVs is monitoring and detection of fires [13], which arise as a result of human factors, climatic conditions and natural disasters. The use of small-sized UAVs to detect seat of the fire allows:

1. to remotely monitor large areas;
2. to do so for a long time;
3. to work autonomously;
4. to reduce or completely eliminate risks to the life and health of the operator who controls the UAV, since this method eliminates the need for a person to physically be at the disaster site.

An important advantage of using UAVs is the ability to detect fires in an automated manner, which will speed up the localization and elimination of the disaster, and significantly reduce the devastating effects of fire on human life and health, the environment, natural and property resources. In order to provide UAVs with this capability, it is necessary to implement artificial intelligence and machine learning technologies in the process of analyzing and processing images captured by its camera [15].

Effective use of UAVs for early detection seat of the fire requires solving two tasks:

1. to substantiate the optimal technical support of the UAV itself, including the requirements for the computer vision system [14];
2. to design the optimal architecture of the neural network (NN) module, which will be tasked with searching for fires in the images received from the UAV.

1. Materials and methods

The analysis and selection of UAV hardware to be used for fire detection is a primary task that requires a careful approach. When choosing UAV hardware, it is important to consider various aspects such as technical characteristics, reliability, accuracy, and the ability to integrate with existing systems.

Reliability of the hardware is one of the main key characteristics, as UAV must operate stably even in difficult conditions, such as strong winds, rain, or poor visibility. The batteries must have sufficient capacity for long-term operation, and the control system must be reliable and efficient. In addition, when choosing the hardware for a UAV to be used for fire detection, it is important to consider the weight and size of the equipment, as this can affect the UAV's battery life and maneuverability. Lightweight and compact sensors can improve performance and provide longer flight times. Reducing the number of different pieces of equipment on board a UAV can also increase autonomy. For example, thermal cameras are currently used to detect thermal traces of fires. However, it is possible to try to limit ourselves to high-resolution video cameras operating in the visible range, provided that neural network technologies for image processing are implemented, which would allow for high accuracy in classifying and recognizing seat of the fire. Technical solutions using spatial photometry with multiple photodetectors [1] and non-spherical hollow mirrors with CCDs [2] may also be interesting, as they implement the principles of image processing with light scattering effects [3] due to raytracing mechanisms in the visible and near-infrared range.

Another aspect is the ability to integrate additional software into the control system of a small-sized UAV, as the data collected from the aircraft should be easily integrated with existing fire monitoring and management systems to quickly respond to detected threats.

All of these aspects should be carefully considered and balanced when analyzing and selecting the UAV hardware to be used for fire detection. The best choice will be the one that takes into account the specifics of the task at hand and provides efficient, accurate and reliable real-time monitoring fire occurrence.

The second task to be solved in order to increase the efficiency of using UAVs for monitoring fire occurrence is to justify and select the optimal architecture of the NN, which should be integrated into the UAV software and perform analysis and processing of incoming information.

When choosing a NN architecture, it is necessary to take into account the key factors for the task at hand: the need for rapid fire detection, the limited computing capabilities of UAVs, and the specifics of the objects present in the images that need to be recognized. The main choice is between the NN architectures used for object detection (e.g., YOLO (You Only Look Once) [12], SSD (Single Shot MultiBox Detector) [10], Faster R-CNN [16], and others) and those used for classification (e.g., ResNet 50/101/152, VGG 16/19, Inception V3/V4, and others) [9].

In order to justify the choice, it is necessary to take into account the specifics of the task, namely the recognition seat of the fire. Flames, as an object in an image, have a complex and variable shape, their contours can be uncertain and will always be non-static. This makes the task of fire detection difficult for architectures that are focused on recognizing objects with clear contours. Therefore, taking into account the peculiarities of the task, it is more appropriate to choose the architectures used for classification. Moreover, the recognition task will be reduced to binary classification, which consists in dividing images into only two classes: with or without fire. This approach will simplify the task of discover the presence or absence of fire in an image.

In addition, the classification-based architecture is more computationally efficient, as it does not require complex object contour recognition operations. This is especially important when implementing systems on limited computing resources (such as small-sized UAVs), as well as taking into account the fact that the NN must analyze images from a video stream in real time with minimal delay.

Today, various NN architectures are developing at a rapid pace, newer and more efficient architectures are emerging. Comparing the capabilities and performance of different NN architectures is not an easy task that must be additionally solved for each individual classification task. Although articles by various researchers are being compared different NN architectures by such metrics as accuracy and error rate, the following aspects should be taken into account:

1. the comparison results obtained during research on some datasets may not correlate with the results obtained on others;
2. the comparison results will be affected by the size of the sample for training and validation, as well as the selected parameters;
3. some NN architectures are subject to changes and modifications over time, so you should not rely on studies conducted a long time ago.

To solve the problem of fire detection, we considered several NN architectures, including VGG and ResNet, which have proven to be effective in image classification tasks. We analyzed various literature sources and information resources to compare the effectiveness of VGG and ResNet architectures in classification tasks, for example [4, 11, 12]. Based on this analysis, as well as the well-known facts about different NN architectures, we chose ResNet. This is because the VGG architecture has more parameters and is more complex, and its accuracy may be lower in some cases [13]. Although the ResNet architecture is deeper than VGG, it is less complex. Among the variants of ResNet architectures, we chose the ResNet-152 in the face of uncertainty about the training and testing dataset. This choice is the first iteration in our research on this topic. ResNet-152 consists of 152 layers and is more complex than ResNet-50 and ResNet-101, but it has higher accuracy [6], reliability, scalability, and the ability to detect more complex patterns. Due to its deep structure, ResNet-152 can learn more complex and abstract image features, which is extremely important for our research. After all, the images on which the NN will be trained and tested will contain different landscapes, weather conditions, fire-like situations (e.g., sunset, red sky, forest in autumn with red leaves), have different scales, display different shapes of fire, complex structure of flames, smoke, etc. Therefore, although ResNet-152 will have a longer learning times and require more computing resources, reliable detection seat of the fire in automatic mode is a key aspect, as it affects the timeliness of response and the ability to localize the fire, which will ensure safety for people and their property.

A graphical representation of the ResNet-152 architecture is shown in Fig. 1. Analyzing and understanding the fundamental structure of the ResNet-152 neural network will allow us to create more efficient code for its implementation in software.

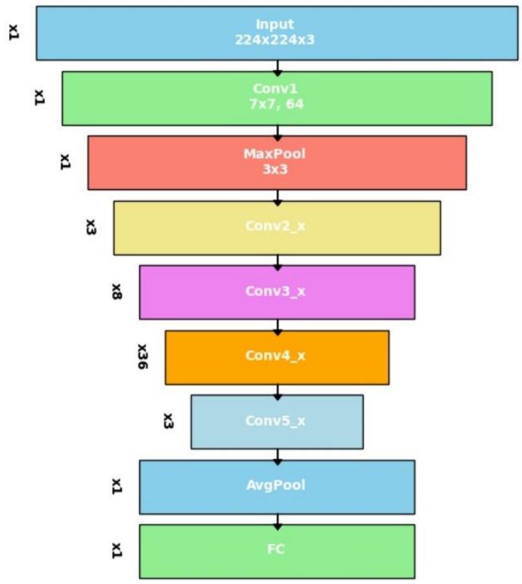

Fig. 1. Visualization of the ResNet-152 architecture

The first layer of the ResNet-152 architecture is the Input layer, which is responsible for accepting images of specified sizes (for example, 224x224) with three RGB channels (red, green, and blue).

The next layer of the architecture is the first convolution layer Conv1, which is responsible for performing convolution with a 7×7 kernel and produces 64 channels. This layer reduces the image dimension to 112×112 pixels.

This is followed by the MaxPool layer, which applies the max-pooling operation to reduce the dimensionality of the input data, reducing its dimensionality to 56×56 pixels.

After the MaxPool layer are the Residual Blocks, which represent a key feature of the ResNet-152 architecture. Their structure consists of several convolutional layers, in which the output from one layer is added to the input of the next layer. This structure helps to avoid the problem of gradients disappearing when training deep NNs. The layers are represented as follows:

- Conv2_x: consists of 3 residual blocks;
- Conv3_x: consists of 8 residual blocks;
- Conv4_x: consists of 36 residual blocks;
- Conv5_x: consists of 3 residual blocks.

The penultimate layer is Average Pooling (AvgPool), which performs the average convolution operation.

The last layer of the ResNet-152 architecture is the Fully Connected (FC), which performs the function of converting features from the previous layers into a vector used for classification.

A key feature of ResNet-152 is that each residual block contains a "skip connection" that allows the model to transfer information across layers without losing significance

The standard implementation of ResNet-152 has 1000 output classes (for the ImageNet dataset), while in our work this layer is adapted to two main classes – the presence of a fire and its absence.

Fig. 2 shows a block diagram of the fire detection and recognition system based on the ResNet-152 architecture, which contains the following main modules: adjustment module, search and recognition module, data transmission module, and display module.

The adjustment module performs preliminary processing of input images, including scaling and lighting changes, as well as setting minimizing and corrective filters at the image adjustment stage, which will significantly improve their quality. Next, the input image from the photo or video stream is passed through the search and recognition modules, after which the processing results are transferred to the display module.

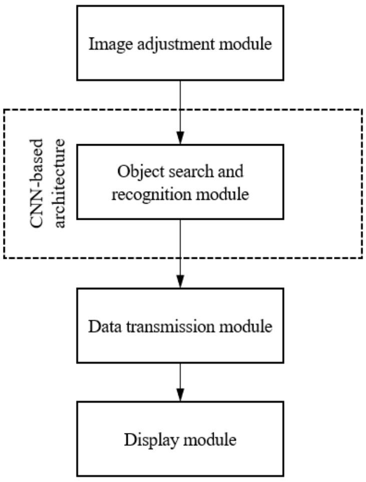

Fig. 2. Block diagram of the proposed system for object search and recognition

All the modules described above are created within the same software. In Fig. 2, the connections between them are shown conventionally and determine the direction of the program execution flow and the sequence of information processing.

2. Experiment and results

Creating and preparing a database is a key step in training NNs. During the formation of the database used for training the NNs, images were selected according to the following criteria:

- images of fires at different angles, distances, and lighting conditions;
- images with no fires or other objects that could cause false alarms;
- images that may contain objects that look like a fire or that may cause false alarms, such as fog, which may look like smoke, or a bright sunset, which may look like fire.

Taking into account the specifics of the NN operation and the tentative field of use (agriculture and agribusiness), it is necessary to collect as many possible image variations as possible during the image search: different seasons, different stages of plant growth, different crops grown in the fields, different weather conditions, time of day, etc.

Another requirement for the images included in the dataset was that they should capture the fire or landscape from above, not from the ground. This is an important aspect, as the style of fire images from above (viewpoint above the fire) and from the ground (viewpoint from the side) are significantly different. If you use images from the ground when training NN, it can lead to false NN training, since the UAV will always record the fire from above (and even if the UAV sees the fire from a distance, the viewpoint will still be higher than when compared to the ground position).

To create the image database, data was collected from online resources, such as Google image search, news sites, images from videos of fires, other UAVs, etc.

The database collected for the NN testing includes 2736 images, of which 1254 do not contain fires and 1482 contain fires. An example of the collected images can be seen in Fig. 3.

Fig. 3. An example of data preparation for training and validation NN

The resulting database was divided into training and validation samples in the following ratio: 80% of the data for training and 20% for validation.

For the correct operation of the NN, the source images were prepared as follows:

- the size of the images was reduced to a single value of 256×256 pixels;
- images were saved in a format supported by the NN framework.

Finally, metadata was created:

- classes of objects to be recognized were defined ("fire" and "no fire");
- a configuration file was created, which contains the path to the images, the number of classes, the size of the images, the paths to the training and validation files, etc.

Fig. 4 shows a diagram that displays all stage of our research of NN model for detecting seat of the fire.

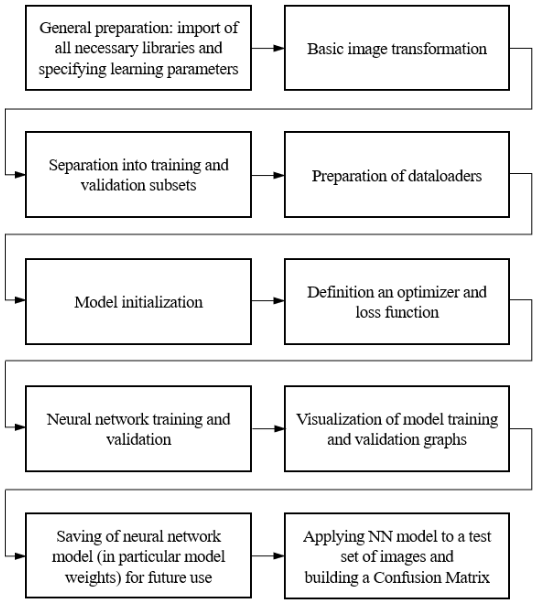

Fig. 4. Stage of our research of NN model for detecting seat of the fire

To work with the NN, we used the Python programming language, the PyTorch deep learning framework, and the NumPy, matplotlib, scikit-learn, and torchvision libraries. To write the code, train and test the NN, we used the software and hardware capabilities of the Kaggle platform. As a result, was created a program that allows working with the selected NN architecture and performing image classification to search for fires.

The results of training and further operation of the NN are influenced by many factors, some of which are the settings of the parameters, in particular: batch size (the size of a batch of data used for one gradient update during training), learning rate (determines the step size when updating model weights), number of training epochs (the number of complete passes through the entire training dataset during training) and the percentage of training data from the total amount of data. Table 1 shows the values that were used training and validation of the selected NN model.

Table 1. Model parameters for training and validation

Parameter	Values
batch_size	32
learning_rate	0.001
num_epochs	10
train_percentage	0.8

In order to evaluate the quality of the NN, we analyze the loss and accuracy for each of the epochs on the training and validation sets. The results for the selected NN architecture, the collected dataset, and the configured parameters are shown in Fig. 5.

Based on the results, we can draw the following conclusions:

- the loss on the training set (Train Loss) decreases over time, which indicates that the model is well trained and improves with each epoch;
- the percentage of correct classifications on the training set (Train Accuracy) reaches 99.5% in the last epoch;
- the dynamics of changes in losses on the validation set (Validation Loss) indicates that the model is generalizing its knowledge to new data;
- the percentage of correct classifications on the validation set (Validation Accuracy) reaches 99.27%, which indicates high accuracy on the validation data.

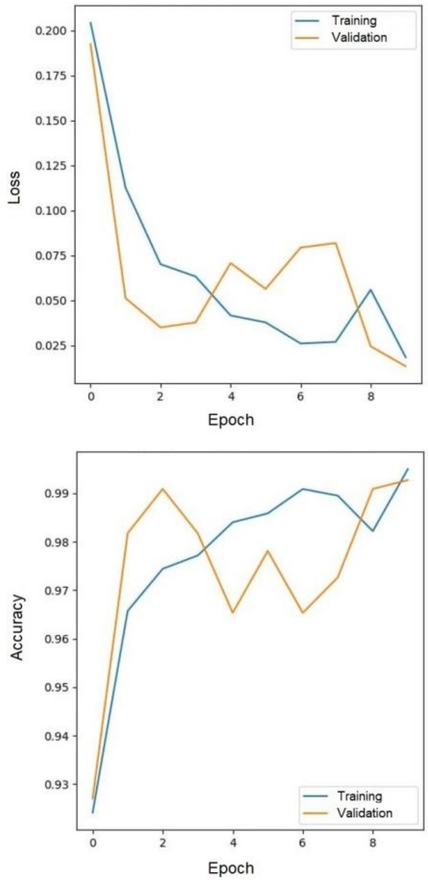

Fig. 5. Loss and accuracy plots for the NN on the training and validation samples

Analyzing the graphs in Fig. 5, we can identify the following trend: it is usually assumed that the model will perform better with an increase in the number of epochs, but it is important to watch for signs of overfitting on the training data.

As can be seen from the plots of the model's loss and accuracy on the training and validation set, the NN model has learned well, as it achieves high accuracy on both the training and validation set. Also, the difference between the losses on the training and validation set is small, which is also a successful sign. So, the model is well trained and generalizes its knowledge to the test data. It is also important to monitor metrics such as accuracy and loss to avoid overtraining and to ensure good performance on new data. In this case, overlearning did not occur.

After training and validation, the NN model was applied to a test set of images, which was created using data from a highly unbalanced dataset [8] that contain normal images and images with fire. It should be noted that this dataset was collected other researchers [13] with other characteristic features that can be used to identify fires. Along with images of fires in open areas, it contains many images of just bonfire. Also, among the images that do not contain a fire, there are many that show indoor spaces, urban development, etc. Therefore, we selected the most relevant images from this dataset for our needs, which correlate with what a UAV can see in real conditions. A certain number of additional images were also added to balance the data. A total of 100 images were selected for testing: 45 with fire and 55 without fire. As a result of applying the NN to the new data, a Confusion Matrix was obtained (Table 2), where the cells reflect the correlation between the actual classes and the model predictions.

Table 2. Confusion Matrix

		True Class	
		Fire	Normal (no fire)
Predicted Class	Fire	TP = 44	FP = 8
	Normal (no fire)	FN = 1	TN = 47

3. Discussions

The model Accuracy calculated based on Confusion Matrix is 91%, Precision – 84.6%, and the Recall – 97.8%. Of course, this number of test images is not enough to make final conclusions about the model's performance. However, we can draw some intermediate conclusions.

It can be noted that the NN coped well with the classification of quite complex images that do not contain a fire, but they show natural conditions that look like a fire. For example: sunset, northern lights, a field of yellow color with a smoke plume from a tractor resembling smoke of fire, etc. (Fig. 6).

Fig. 6. Samples of difficult images that NN right classify as normal (without fire)

A high Recall metric (taking into account the number of errors when a fire was not correctly identified) means that the model is well trained to identify fires and makes virtually no mistakes. If we talk about the Accuracy metric (which takes into account the number of all errors in the model's predictions), the main number of false predictions is due to the fact that situations without fire are recognized as containing fire. But this can be explained. As mentioned earlier, we used for testing a dataset, for which the method of image selection was different from the way we did it. If looking at Fig. 7, you will see several examples of images on which the NN made a mistake. Some images are very similar to fires (sunset, autumn forest), so the prediction errors in these images are objective (although Fig. 6 shows an example where the NN did not make a mistake in classifying the image with a sunset), but some images (such as mountains) are not. Errors in the images with mountains occurred because we did not use images with mountainous terrain for training at all, but due to their complex shape, mountains can really resemble a fire.

Fig. 7. Samples of difficult images that NN wrong classify as not normal (with fire)

Thus, the results obtained indicate the high efficiency of the NN model, but in the future, it is necessary to expand our dataset for training and validation with images with more weather conditions and landscapes.

4. Conclusions

The implementation of the ResNet-152 for fire detection has shown high efficiency and accuracy. Integration of this NN with the software of a small-sized UAV will automate the process of early detection seat of the fire and speed up the response time of the relevant services.

Also, due to modern technologies, in tandem with the developed NN, it will be possible to collect and process data indicating the extent of the fire and transmit a specific location, calculate the area and determine the best ways to the danger zone. This will optimize the process of firefighting by the relevant services, as they will have access to up-to-date information in real time and will be able to adjust their action plan.

Further research can be divided into two direction. The first direction is to increase the dataset by expanding it with new images. This will allow better training of the NN and increase its accuracy. This direction also involves a comparative analysis of the effectiveness of different NN architectures. Although in this study we chose the ResNet-152 architecture, which demonstrated excellent results, a comparative analysis may allow us to justify the use of another more efficient and reliable NN.

The second direction of research is the integration of the developed NN into a system for recognizing seat of the fire from small-sized UAV images. In this case, the following steps should be performed:

1. Justification and selection of circuitry (microcontroller architecture, FPGA) to be installed on the UAV. The justification should be made taking into account the selected architecture of the NN.
2. Development of interface specifications for IOS and Android platforms, which will determine how the NN will interact with the UAV control system. Consideration should be given to the use of standard communication protocols, data formats, and input and output data processing. It is also important that the interface is efficient, understandable to the end user and provides the required data transfer speed, because the lower data processing delay will lead to faster fire detection, and therefore the devastating consequences of the disaster will be much less.
3. The next step is to set up the integrated fire detection system and test it in real conditions, which involves debugging all system components, including the hardware, general software, and the control system itself.
4. Final stage of the NN integration into the control system of a small-sized UAV is continuous optimization and improvement, which includes the development of updates and expansion of the NN capabilities depending on the needs of the end user.

References

[1] Bezugla N. et al.: Biological Tissues Axial Anisotropy Spatial Photometry. Bezuglyi M. et al. (eds): Advanced System Development Technologies I. Studies in Systems, Decision and Control 511, Springer 2024 [https://doi.org/10.1007/978-3-031-44347-3_5].
[2] Bezuglyi M. et al.: Ellipsoidal Reflectors for Biological Media Light Scattering Photometry. Bezuglyi M. et al. (eds): Advanced System Development Technologies I. Studies in Systems, Decision and Control 511, Springer 2024 [https://doi.org/10.1007/978-3-031-44347-3_4].
[3] Bondariev D. et al.: Optical Properties of Light-Scattering Standards for CCD Photometry. Sensors 23, 2023, 7700 [https://doi.org/10.3390/s23187700].
[4] Cabezas M. et al.: Detection of invasive species in wetlands: practical DL with heavily imbalanced data. Remote Sensing 12(20), 2020, 3431 [https://doi.org/10.3390/rs12203431].
[5] Eisenbeiss H.: A mini unmanned aerial vehicle (UAV): system overview and image acquisition. International Archives of Photogrammetry. Remote Sensing and Spatial Information Sciences 36(5), 2004, 1–7.
[6] Hasanah S. A. et al.: A Deep Learning Review of ResNet Architecture for Lung Disease Identification in CXR Image. Applied Sciences 13(24), 2023, 13111 [https://doi.org/10.3390/app132413111].
[7] He K. et al.: Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, 770–778.
[8] https://github.com/UIA-CAIR/Fire-Detection-Image-Dataset (available: 29.03.2024).
[9] Khan A. et al.: A survey of the recent architectures of deep convolutional neural networks. Artificial intelligence review 53, 2020, 5455–5516.
[10] Liu W. et al.: SSD: Single shot multibox detector. 14th European Conference Computer Vision–ECCV, Part I 14, 2016, 21–37.
[11] Mascarenhas S., Agarwal M.: A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON), 2021, 96–99 [https://doi.org/10.1109/CENTCON52345.2021.9687944].
[12] Redmon J. et al.: You only look once: Unified, real-time object detection. IEEE conference on computer vision and pattern recognition, 2016, 779–788.
[13] Sharma J. et al.: Deep convolutional neural networks for fire detection in images. Communications in Computer and Information Science 744, 2017 [https://doi.org/10.1007/978-3-319-65172-9_16].
[14] Stelmakh N. et al.: Features of the implementation of computer vision in the problems of automated product quality control. Informatyka, Automatyka, Pomiary w Gospodarce i Ochronie Środowiska 13(1), 2023, 38–41 [http://doi.org/10.35784/iapgos.3434].
[15] Wonjae L. et al.: Deep neural networks for wild fire detection with unmanned aerial vehicle. IEEE international conference on consumer electronics (ICCE), 2017, 252–253 [https://doi.org/10.1109/ICCE.2017.7889305].
[16] Zhang L. et al.: Is faster R-CNN doing well for pedestrian detection? 14th European Conference Computer Vision–ECCV, Part II 14, 2016, 443–457.

Ph.D. Nataliia Stelmakh
e-mail: n.stelmakh@kpi.ua

Associate professor in department of Device Production at the Faculty of Instrumentation Engineering, National Technical University of Ukraine "Kyiv Polytechnic Institute".
Author and co-author of more than 50 scientific papers, 12 patents for utility models. Research interests: automation and computer-integrated technologies, assembly of devices and preparation of production, computer vision.

https://orcid.org/0000-0003-1876-2794

M.Sc. Svitlana Mandrovska
e-mail: mandrovskasvitlana@gmail.com

Svitlana Mandrovska received a master's degree in automation and computer-integrated technologies from the National Technical University of Ukraine "Kyiv Polytechnic Institute", Faculty of Instrumentation (Ukraine).
Research interests: study of computer vision for production processes, machine learning, neural networks.

https://orcid.org/0009-0005-2354-9965

Ph.D. Roman Galagan
e-mail: r.galagan@kpi.ua

Associate professor in Department of Automation and Non-Destructive Testing Systems the Faculty of Instrumentation Engineering, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute".
Author and co-author of more than 40 scientific papers, 1 monograph and 1 textbook. Research interests: programming, machine learning, non-destructive testing, computer vision, data analysis.

https://orcid.org/0000-0001-7470-8392