ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 6, Num. 2, 2021

# ANALYSIS OF ALGORITHMS FOR SEARCHING OBJECTS IN IMAGES USING CONVOLUTIONAL NEURAL NETWORK

***Ihor Koval***

*Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine.*
Author's e-mail*: kowalkowaligor2014@gmail.com*

https://doi.org/10.23939/acps2021.__.___
*Submitted on 10.10.2021*


***Abstract*: The problem of finding objects in images using modern computer vision algorithms has been considered. The description of the main types of algorithms and methods for finding objects based on the use of convolutional neural networks has been given. A comparative analysis and modeling of neural network algorithms to solve the problem of finding objects in images has been conducted. The results of testing neural network models with different architectures on data sets VOC2012 and COCO have been presented. The results of the study of the accuracy of recognition depending on different hyperparameters of learning have been analyzed. The change in the value of the time of determining the location of the object depending on the different architectures of the neural network has been investigated.**

***Index Terms*: Object detection, neural networks, deep learning, R-CNN, Fast R-CNN, Faster R-CNN**

## I. INTRODUCTION

Finding objects in images is one of the most important tasks of scene analysis and machine vision. The task of finding objects in images usually consists of two subtasks: the first of them belongs to the class of tasks of detection of an object belonging to one of the given classes in the image; the second - in the classification of the found object belonging to one of the specified classes. The solution of these problems can be carried out in two successive stages, and together, without division into stages in time. The complexity of this task is determined by the high degree of variability of real images and the objects represented on them. Almost everything changes: condition, angle, lighting, color, shape, etc.

Until recently, the problem of this class of problems was solved using various information processing algorithms, including: adaptive gain algorithms [1], algorithms based on the use of gradient histograms [2] and color information [3], cascade classifier algorithms based on the method Viola-Jones [4], [5], who has performed particularly well for detecting human faces in images. Algorithms based on the methods of contour analysis have also become widespread.

Solutions based on conventional neural networks with prior selection of features of object classification, as well as standard deep convolutional neural networks are also known [6]. It should be noted that this approach is quite costly in terms of computation, despite all the advantages that initially provide neural networks in terms of accuracy of classification.

Recently, algorithms based on the use of convolutional neural networks or Regional Convolutional Neural Networks (R-CNN) have become widespread to solve this problem, which are fundamentally focused on solving the problem of finding objects with their simultaneous classification [7]-[11]. In comparison with the methods presented above, these algorithms are fundamentally adapted to solve the problem of finding objects in images. Their initial implementation is based on the use of special pre-processing algorithms - region-proposal-function, which provide a proposal of the so-called areas of attention, which could potentially be objects that interest us. This "specialized" approach proposes to reduce computational costs, and also allows to achieve a minimum time to determine the location of the object and high accuracy of its classification. At present, there are a large number of options for the implementation of such algorithms, which have achieved good results according to these criteria [12].

### *A. THE AIM OF THE WORK*

Thus, the purpose of this study is to conduct a comparative analysis of neural network algorithms of class R-CNN, to search for objects in images.

## II. RESEARCH METHODS AND MATERIALS

The following models of neural networks for solving the problem of objects in images are considered: R-CNN (region-based convolutional neural networks) with the original algorithm region-proposal-function, Fast R-CNN, Faster R-CNN [9]-[11]. Data sets VOC 2012 and COCO were used in the research.

The data set VOC 2012 (20 classes, 11530 images) [13] consists of images of size 500 × 375, each of which has an annotation file that provides coordinates that delimit the boundaries and label of the object class from the list of twenty classes.

The COCO data set (80 classes, 330,000 images) [14] is similar in structure to VOC 2012, with an image size of

640 × 507. In addition to image files, it also contains annotations that contain the coordinates of the object, the class to which it belongs, link to it and more information.

5300 images from each data set were used to train and test the models.

### *A. R-CNN MODEL*

When searching for objects using the R-CNN model [9], the following sequence of steps is performed, shown in Fig. 1.

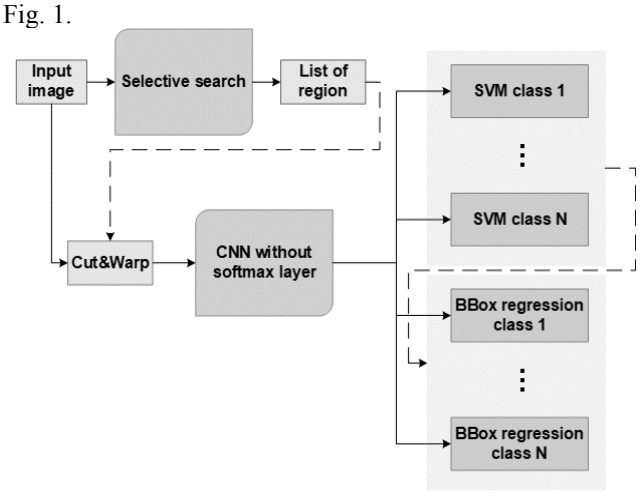


*Fig. 1 Search of the objects in the image, R-CNN*

Step 1. The generation of regions of interest (region proposals), presumably containing the desired objects (usually up to 2000 possible areas) using various algorithms designed to reduce the computational complexity of detecting objects in the image (e.g., Edge Boxes algorithms, Selective search).

In a standard object search scheme, the detector first traverses the input image (or set of images) at different scales by shifting the detection window. Thus, a set of rectangular areas of interest is formed, which may contain an object.

The Edge Boxes algorithm [15] is designed to generate areas of interest in the form of a rectangular frame bounding the object, and is based on the selection of the contours of objects. The contours provide a sparse but informative representation of the input image. The number of contours that are completely contained in the boundary rectangle indicates the probability that the rectangle contains an object. The evaluation of each block is performed by excluding from the set of contours contained in the boundary rectangle (block), those that intersect with the boundaries of the block. Thus, generated areas of interest for the input image are areas containing closed or almost closed contours.

The selective search algorithm [16] is based on the use of the method of hierarchical grouping of similar regions based on the correspondence of color, texture, size or shape and graphs. For these graphs, the vertex is the intensity of the current pixel, and the edges connect a pair of adjacent pixels. The absolute difference in pixel intensity of the vertices is used as the weight of the edge. Using the graph, the fragments are selected, which are then grouped according to the following principle: the edges between two vertices in one group should have less weight, and the edges between two vertices in different groups should have more weight. After the two most similar regions are grouped, a new similarity is calculated between the resulting region and its neighbors. The process of grouping similar areas is repeated until the whole image becomes one area.

Thus, the selective search algorithm is implemented using detailed image segmentation depending on the pixel intensity, using the method of segmentation based on graphs and selective search.

Step 2. At this stage, the formation of a map of features for the input image. The formed areas of interest are scaled to a size comparable to the architecture of the CNN neural network. To do this, an affine transformation is performed, and each area of interest is converted to a square of 227 × 227, as the CNN architecture used requires inputs of a fixed size of 227 × 227 pixels. In this case, before optimization, the bounding box of the area of interest is expanded so that after the transformation to get around the area, which may contain an object, a border with a width of 16 pixels.

The data formed in this way are fed to the input of the convolutional neural network (CNN), the architecture of which is presented in Fig. 2.

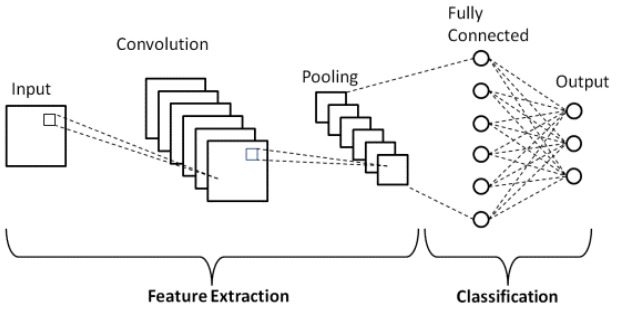


*Fig. 2 CNN Architecture*

At the output of the CNN neural network, a 4096-dimensional feature vector is formed for each area of interest.

Step 3. At this stage, the classification of objects for each area of interest is performed using the generated feature vector based on the method of reference vectors (SVM). To do this, firstly, before performing the classification, for each generated area is applied non-maximum suppression (Non-Maximum Suppression algorithm), based on which only local maxima are marked as the contour of the object. Suppression of non-maxima is necessary to exclude the generated duplicate areas of interest for each object on the original document. A pixel whose value is above the upper limit acquires the maximum value, i.e., the contour is considered valid. If the pixel value does not reach the lower threshold, it is suppressed.

To assess the quality of classification, an indicator is equal to the ratio of the cross-sectional area of the rectangle (area of interest) obtained from the detection and the rectangle from the markup to the area of their union (Intersection over Union, IoU) is used. Thus, it is considered that the object is detected correctly if this indicator exceeds a certain threshold, otherwise it is considered that the object is not detected. IoU is calculated by the following formula:

$$\mathrm{IoU} = \frac{\mathrm{AoO}}{\mathrm{AoU}},$$

where AoO (Area of Overlap) - the area of intersection of the true rectangle and the predicted; AoU (Area of Union) - the area of union of a true rectangle and the intended.

### *B. R-CNN REGION-PROPOSAL-FUNCTION ALGORITHM*

For the R-CNN model, an original algorithm for determining regions of interest Region-proposal-function is implemented. In its architecture, this algorithm is close to that proposed in [9], but slightly simpler and better in speed.

In general, it should be noted that the R-CNN model is quite slow due to the fact that the algorithms for generating areas of interest create many constraint rectangles, which are processed by a neural network for the analysis of which it takes considerable time.

### *C. FAST R-CNN MODEL*

Despite the good results obtained with its use, the performance of R-CNN is not so high, especially for deeper than CaffeNet networks (such as VGG19). In addition, when using this model, it is necessary to store a large amount of data. The scheme of the algorithm for the model Fast R-CNN [10] is presented in Fig. 3.

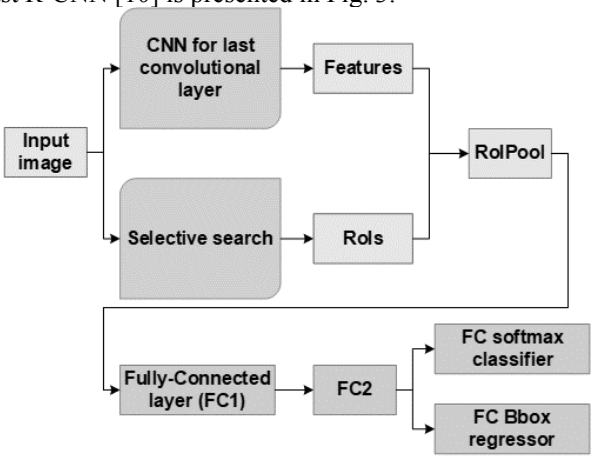


*Fig. 3 Search of the objects in the image, Fast R-CNN*

Step 1. The generation of regions of interest (region proposals), which presumably contain an object (for example, the Edge Boxes algorithm).

Step 2. The feature map for the input image is formed. To do this, the input of the neural network CNN is full of input image, but the last layer of the max-pool is replaced by RoI pooling. The RoI pooling layer receives at the input a feature map obtained from the last convolutional layer of the neural network, and the generated area of interest (in image coordinates). The area of interest is converted from image coordinates to coordinates on the feature map and the resulting rectangle is superimposed on the grid W × H with specified dimensions, the area of interest width w and height h is converted into a grid having H × W cells of size h / H × w / W (for example, for VGG19 W = H = 7). Thus, RoI pooling converts the feature vector of an arbitrary rectangle from the input image into a feature vector of a fixed size. Then Max Pooling is applied to each such cell to select only one value, thus, the resulting matrix of features H × W is formed.

Step 3. The boundaries of the area of interest are specified using a regression model (Bounding Box Regression). The obtained areas of interest and feature vectors are fed to the input of two new fully bound layers. The first is used to specify the boundaries of a rectangle, and the second is used to classify an object located inside that rectangle (Step 4). To refine the bounding box, the Bounding Box Regression learns to adjust the predicted bounding box using CNN functions. For a given predicted coordinate of the bounding box $p = (p_x, p_y, p_w, p_h)$, where $p_x$ is the x coordinate of the frame center, $p_y$ is the y coordinate of the frame center, $p_h$ is the width of the bounding box, $p_w$ is the length of the bounding box, and the corresponding coordinates of the rectangle truth $g = (g_x, g_y, g_h, g_w)$, where $g_x$ is the x coordinate of the truth rectangle, $g_y$ is the y coordinate of the truth rectangle, $g_h$ is the width of the truth rectangle, $g_w$ is the length of the truth rectangle. The regressor is configured for a scale-invariant transformation between two centers and a log-scale transformation between latitude and altitude, shown in Fig. 4.

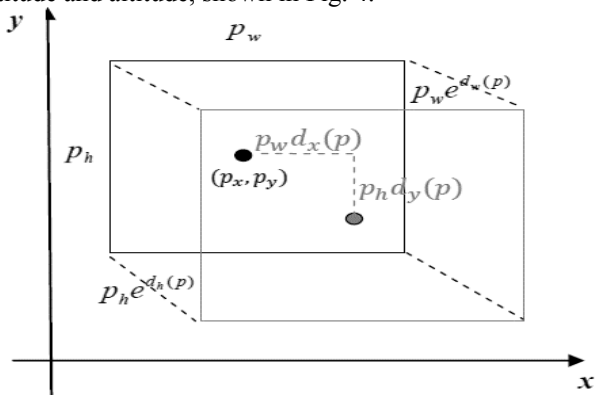


*Fig. 4 Transformation between predicted and true constraints*

$$g_x = p_w d_x(p) + p_x,$$
$$g_y = p_h d_y(p) + p_y,$$
$$g_w = p_w e^{d_w(p)},$$
$$g_h = p_h e^{d_h(p)},$$

where $d_x(p)$ and $d_y(p)$ – functions of scale-invariant transformation of the center with coordinates x and y, $d_w(p)$ and $d_h(p)$ – log-scale transformation functions between width w and height h.

Step 4. The classification of objects contained in the presumed areas of interest is performed. To classify objects, a softmax layer with K + 1 outputs is used (where K + 1 is a value that characterizes the number of object classes, taking into account the presence of the background on the original document). Steps 3 and 4 are performed in parallel.

### *D. FASTER R-CNN MODEL*

Compared to Fast R-CNN, Faster R-CNN [11] uses a special Region Proposal Network (RPN) system instead of using an external algorithm to generate areas of interest where the object is likely to be located (e.g., Edge Boxes). The scheme of work for the Faster R-CNN model is presented in Fig. 5.

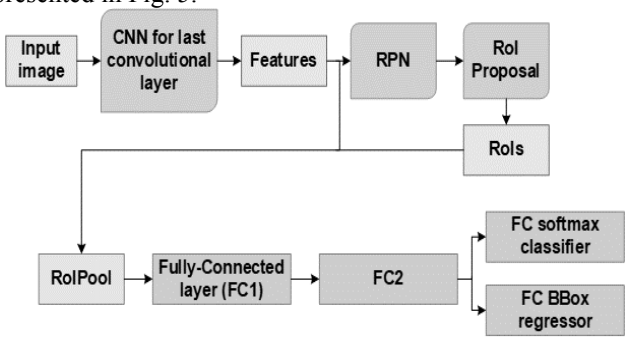


*Fig. 5 Search of the objects in the image, Faster R-CNN*

Step 1. The formation of a map of signs on the basis of the input image is carried out. To do this, the input image is fed to the input of the convolutional neural network CNN, the architecture of which is presented in Fig. 2.

Step 2. Generation of areas of interest that may contain the object. Using the neural network Region Proposal Network (RPN), the generated map of features is processed by a variable window of a given size 3 × 3. Thus, a set of 9 areas of interest is generated, which have the same center, but with different proportions and scales. For each position of the window the vector of signs of small dimension is extracted. In addition, for each of the generated areas, the value of IoU is calculated, based on the value of which a decision is made about the current area of interest.

Step 3. The conversion of the vector of features of the area of interest (arbitrary rectangle) from the input image to the vector of features of a fixed dimension, using the layer RoI pooling.

Step 4. The boundaries of the area of interest are specified using a regression model (Bounding Box Regression). The obtained areas of interest and the obtained feature vectors are fed to the input of two new fully bound layers. The first of these layers is used to specify the boundaries of the rectangle, and the second to classify the object located inside this rectangle (Step 5).

Step 5. The classification of objects contained in the proposed areas of interest is performed. To classify objects, a softmax layer with K + 1 outputs is used (where K + 1 is a value that characterizes the number of object classes, taking into account the presence of the background on the original document). Steps 4 and 5 are performed in parallel.

## III. RESEARCH RESULT

In the course of research, R-CNN, Fast-RCNN, Faster-RCNN models were implemented using Python 3.9 and using the Pytorch machine learning library (version 1.9) with GPU support. Training and testing of neural networks were performed on data without the use of augmentation. Detection time and accuracy of object classification in the image were used as indicators of efficiency of the received models.

When learning networks, transfer learning technology was used, which allows the use of ready-made neural networks to solve new types of problems for which networks are not previously amenable to learning. Convolutional neural networks VGG19 and ResNet152 were used to implement the learning transfer, truncated to the penultimate fully connected layer, from which the feature vector of dimension 4096 for VGG19 and 2048 for ResNet152 was removed. The resulting vector was applied to a new fully connected layer, which is responsible for classifying the object in the image.

When implementing the R-CNN model using a new fully connected layer with softmax activation function as a classifier, a neural network architecture was used without specifying bounding boxes (without regression).

During the experiments, the influence of two algorithms for generating areas of interest was investigated, namely Selective search and the original algorithm Regional-proposal-function. The results of experimental studies are presented in Table 1.

*Table 1*

**R-CNN results (VOC2012 data set, Adam optimization method)**

| Algorithm | CNN model | Number of epochs | Accuracy | Detection time, sec |
|---|---|---|---|---|
| Regional-proposal-function | VGG19 | 50 | 0.7475 | 8.373 |
| | | 100 | 0.8846 | 8.442 |
| | | 150 | 0.8936 | 8.343 |
| | ResNet152 | 50 | 0.8362 | 8.151 |
| | | 100 | 0.9034 | 8.183 |
| | | 150 | 0.9452 | 8.051 |
| Selective search | VGG19 | 50 | 0.7962 | 10.840 |
| | | 100 | 0.8699 | 10.701 |
| | | 150 | 0.9198 | 10.866 |
| | ResNet152 | 50 | 0.8279 | 10.715 |
| | | 100 | 0.9358 | 10.322 |
| | | 150 | 0.9681 | 10.376 |

The obtained results show that the object detection time using the Regional-proposal-function is less than using the Selective search. This is explained by the fact that to generate areas of interest using the first algorithm takes less time with equal learning parameters. This also shows that the proposed modification of the algorithm Regional-proposal-function is faster, but in some cases the result of its application with CNN neural networks is slightly worse.

A study of network quality indicators depending on the optimization method was also conducted. The influence of two methods was studied: SGD (stochastic gradient descent) and the Adam algorithm [17] (stochastic optimization algorithm). The results presented in Table 2, show that the accuracy of object recognition when learning the network by the Adam method was higher than when using the SGD method.

In Table 3-Table 5 there are results of Fast R-CNN and Faster R-CNN models trained by the Adam method. The results indicate that the Faster R-CNN model is faster than other neural network architectures in terms of test images.

*Table 2*

**Study of the impact of optimization methods (R-CNN, data set VOC2012, number of learning epochs 150)**

| SGD method | Adam method | Accuracy | Detection time, sec |
|---|---|---|---|
| + | – | 0.9467 | 10.583 |
| – | + | 0.9681 | 10.376 |

*Table 3*

**The result of Fast R-CNN (VOC2012 data set, Adam optimization method)**

| CNN model | Number of epochs | Accuracy | Detection time, sec |
|---|---|---|---|
| VGG19 | 50 | 0.7621 | 3.212 |
| | 100 | 0.8037 | 3.454 |
| | 150 | 0.9232 | 3.294 |
| ResNet152 | 50 | 0.8372 | 3.345 |
| | 100 | 0.9224 | 3.664 |
| | 150 | 0.9528 | 3.552 |

*Table 4*

**The result of Faster R-CNN (VOC2012 data set, Adam optimization method)**

| CNN model | Number of epochs | Accuracy | Detection time, sec |
|---|---|---|---|
| VGG19 | 50 | 0.6882 | 1.283 |
| | 100 | 0.7325 | 1.364 |
| | 150 | 0.8537 | 1.298 |
| ResNet152 | 50 | 0.7851 | 1.382 |
| | 100 | 0.8683 | 1.482 |
| | 150 | 0.9028 | 1.372 |

The graphs presented in Fig. 6 illustrate changes in the loss function and the accuracy value of the object classification when using the Faster R-CNN, when learning on the COCO data set. Obviously, this neural network achieves high accuracy in classification accuracy and low values of the loss function.

*Table 5*

**The result of Faster R-CNN (COCO data set, Adam optimization method)**

| CNN model | Number of epochs | Accuracy | Detection time, sec |
|---|---|---|---|
| VGG19 | 50 | 0.8867 | 1.684 |
| | 100 | 0.9054 | 1.627 |
| | 150 | 0.9547 | 1.739 |
| ResNet152 | 50 | 0.9234 | 1.542 |
| | 100 | 0.9648 | 1.536 |
| | 150 | 0.9832 | 1.622 |

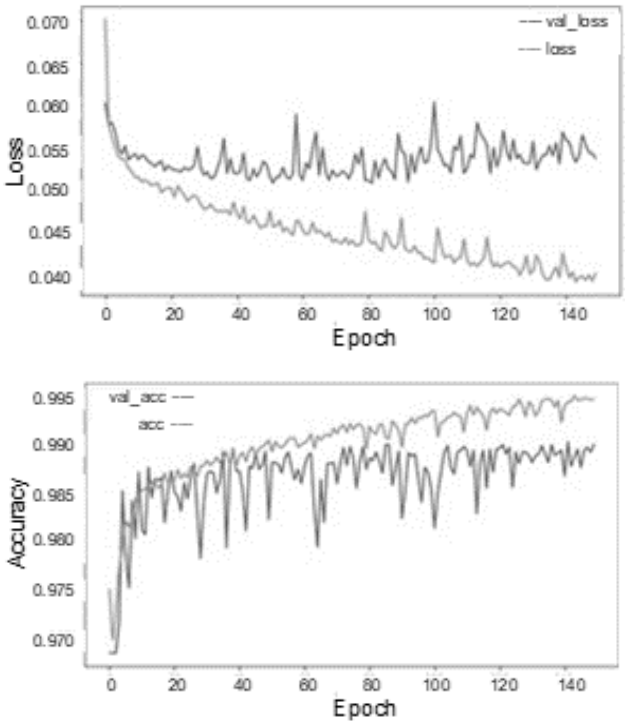


*Fig. 6 Faster R-CNN, graph of the loss function and graph of the accuracy of object recognition.*

In Fig. 7,Fig. 8 there are examples of object detection (road sign).

*(a)*

*(b)*

*Fig. 7 Example of object detection in the image using R-CNN: (a) - the first five examples of the location of the bounding box with the highest probability of containing the object, (b) - the last five examples of the location of the bounding box with the lowest probability contain an object.*


*Fig. 8 Example of detecting an object in the image (left) using Fast R-CNN, (right) using Faster R-CNN.*

## IV. CONCLUSION

The article considers the tasks of searching for objects in images using models of convolutional neural networks R-CNN, Fast R-CNN, Faster R-CNN. In the course of the research a comparative analysis of the regional-proposal algorithm implemented within the R-CNN model with the selective search algorithm was performed. Based on the presented results, we can conclude that the use of the proposed approach to the generation of areas of interest accelerates the process of finding an object in the image. The results of testing the modifications of different models of R-CNN show that Faster R-CNN significantly exceeds (6 times compared with R-CNN and 2,5 times with Fast R-CNN) the speed of other models through the use as an algorithm for generating areas of interest of a special network for the formation of appropriate proposals. In general, the obtained results of modeling and testing these models indicate a high accuracy (in total 85,7%) of object detection in the image for all considered algorithms.

## References

[1] Butenko, V. V. (2015) 'Finding objects in the image using the adaptive gain algorithm', *Young scientist*, 4, pp. 52–56. Available at: https://moluch.ru/archive/84/15604/ (Accessed: 10 October 2021).

[2] Dalal, N. and Triggs, B. (2005) 'Histograms of oriented gradients for human detection', *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, San Diego, CA, USA, 1, pp. 886–893. doi: 10.1109/CVPR.2005.177.

[3] Artemov, A. A., Kavalerov, M. V. and Kuznetsov, G. S. (2011) 'The problem of finding objects in images using computer vision based on color information', *Bulletin of PNRPU. Electrical engineering, information technology, control systems*, 5, pp. 70-79. Available at: https://cyberleninka.ru/article/n/problema-poiska-obektov-na-izobrazheniyah-s-pomoschyu-kompyuternogo-zreniya-na-osnove-informatsii-o-tsvete/viewer (Accessed: 10 October 2021).

[4] Akimov, A. V. and Sirota, A. A. (2016) 'Models and algorithms for artificial data multiplication for training face recognition algorithms using the Viola-Jones method', *Computer Optics*, 6, pp. 899–906. Available at: https://readera.org/modeli-i-algoritmy-iskusstvennogo-razmnozhenija-dannyh-dlja-obuchenija-algoritmov-14059619 (Accessed: 10 October 2021).

[5] Viola, P. and Jones, M. (2004) 'Robust real time face detection', *International Journal of Computer Vision*, 57(2), pp. 137–154. doi: 10.1023/B:VISI.0000013087.49260.fb.

[6] Towards data science (2018) *R-CNN, Fast R-CNN, Faster R-CNN, YOLO – Object Detection Algorithms*. Available at: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e (Accessed: 10 October 2021).

[7] Kushnir, D. and Paramud, Y. (2019) 'Methods for real-time object searching and recognizing in video images on ios mobile platform', *Computer systems and network*, 1(1), pp. 24-34. doi: 10.23939/csn2019.01.024.

[8] Michelucci, U. (2019) *Advanced Applied Deep Learning: Convolutional Neural Networks and Object Detection*, 1st edition, TOELT LLC, Dübendorf, Switzerland, September 29 2019, 303 p.

[9] Girshick, R., Darrell, J. and Malik, T. (2015) 'Region-Based Convolutional Networks for Accurate Object Detection and Segmentation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, pp. 142-158. doi: 10.1109/TPAMI.2015.2437384.

[10] Girshick, R. (2015) 'Fast R-CNN', *International Conference on Computer Vision (ICC)*, pp. 1440-1448. doi: 10.1109/ICCV.2015.169.

[11] Girshick, R., Shaoqing, R. and Kaiming, H. (2015) 'Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks', *Neural Information Processing Systems (NIPS)*, 39(6), pp. 1137-1149. doi: 10.1109/TPAMI.2016.2577031.

[12] Wang, Y., Wang, C., Zhan, H., Yingbo, G. and Wei, S. (2019) 'Automatic Ship Detection Based on RetinaNet Using Multi-Resolution', *Remote Sensing*, 11(5), 531 p. doi: 10.3390/rs11050531.

[13] Pascal 2 (2012) *The PASCAL Visual Object Classes Challenge 2012 (VOC2012)*. Available at: http://host.robots.ox.ac.uk/pascal/VOC/voc2012/index.html (Accessed: 10 October 2021).

[14] Microsoft (2021) *COCO: Common Objects in Context*. Available at: http://cocodataset.org/#home (Accessed: 10 October 2021).

[15] Zitnick, C. and Dollár, P. (2014) 'Edge boxes: Locating object proposals from edges', *Computer Vision*, 5, pp. 391–405. doi: 10.1007/978-3-319-10602-1_26.

[16] Sande, J., Gevers, K. and Smeulders, T. (2013) 'Selective Search for Object Recognition', *International Journal of Computer Vision*, 104, pp. 154-171. doi: 10.1007/s11263-013-0620-5.

[17] Chopra, R., England, A. and Noordeen Alaudeen, M. (2019) *Data Science with Python: Combine Python with machine learning principles to discover hidden patterns in raw data*, Packt Publishing Ltd, July 2019, 426 p.

**Ihor Koval** received a B.S. in computer engineering at Lviv Polytechnic National University in 2020. Since 2020 he has obtained a M.S. in system programming at Lviv Polytechnic National University.

His research interests include neural networks, automating processes, system programming and Internet of Things (IoT).