Technology for Indoor Drone Positioning Based on CNN Detector

V.A. Gorbachev[1], Yu.B. Blokhinov[1], A.D. Nikitin[1], E.E. Andrienko[1]
vadim.gorbachev@gosniias.ru|yury.blokhinov@gosniias.ru
[1]FSUE “GosNIIAS”, Moscow, Russia

The article presents the drone positioning technology in a multi-camera system by using the detection algorithm. Paper describes positioning system and algorithm for calculating 3d drone coordinates based on its image position, detected on images of stationary video cameras. Positioning enables automatically control the drone when precise data from satellite navigation systems are not available, for example, in closed hangars. The developed technology is used to create a complex of automatic visual control of aircraft. The ways of adaptation of neural network detection algorithm to the problem of drone detection are presented. The main attention is paid to the methods of training data preparation. It is shown that high accuracy can be achieved using synthesized images without any real data or manual labelling.

Keywords: *object detection, neural networks, drones, positioning, indoor navigation, multi-camera system, image synthesis.*

1. Introduction

Currently, due to the increase in the aircraft flow in the airspace, the complexity of their timely and high-quality visual inspection during the maintenance at the airport has increased significantly. Significantly increased the total downtime of aircraft during unscheduled inspections, caused, for example, the impact of atmospheric electricity on the surface of the fuselage of the aircraft in flight. External human inspection of hard-to-reach areas of the aircraft, such as the upper fuselage or tail, aimed to identify the effects of lightning today takes a significant time, leading to downtime of aircraft or even flight delays. For companies which have a fleet of more than 200 aircraft, such as Aeroflot, such an event is not uncommon: according to the company, it occurs about 300-400 times a year, leading to significant time and financial losses. Large companies such as Airbus, Lufthansa, EasyJet, American Airlines start applying drones to solve the problems of accelerating the visual inspection of the aircraft. However, currently, the use of drones is carried out in manual mode, which does not allow to completely reveal the potential of the technology. According to experts, the use of programmable drones will significantly reduce the time of inspection of the aircraft and, no less significantly, make the technology itself completely digital.

The article proposes an approach to the creation of automated drone control technology based on its real time positioning using a system of stationary cameras. This technology is necessary to ensure the functioning of the drone control system in enclosed spaces such as aircraft hangars. The development of a special positioning technology is necessary, since the signals of global satellite navigation systems (GPS, GLONASS, etc.) may be partially or completely inaccessible in the hangar where aircraft maintenance is carried out. At the same time, the inertial navigation system of the drone can’t provide sufficient accuracy throughout its flight. Due to the fact that the flight of the drone must be carried out at a short distance from the aircraft (no more than 1.5 meters), ensuring the accuracy of the trajectory is a critical aspect for the safety and applicability of the technology. Visual positioning system is the most preferable in the described conditions, as it is able to provide sufficient accuracy, it does not require the installation of additional equipment on the drone, it is passive, so, it does not emit any radio or other signals except Wi-Fi.

During maintenance, the drone flies over the aircraft on a programmed trajectory and makes a high resolution video of the surface of the fuselage and wings (Fig. 1). Based on the coordinates obtained from the visual positioning system, the onboard drone control system monitors compliance with the choosen trajectory. By results of the automatic analysis of the received videos the decision on existence of damages on a covering of aircraft is made. This technology allows complete automating the process of visual inspection of aircraft [1].

The paper describes the features of creating such a technology in terms of positioning drones through the use of CNN-based detectors.

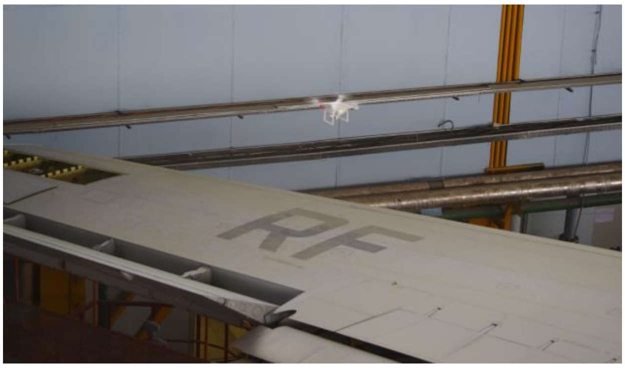

Fig. 1. The drone flight over the aircraft during the tests.

2. Review of detection algorithms

The proposed technology is based on an algorithm for detecting objects in images (namely, video frames). The most advanced detection algorithms today are algorithms based on deep convolutional neural networks (CNN). Neural network architectures for detection are divided into two main types: single-stage and two-stage. In two-stage approaches, the task of detecting objects is divided into two steps: identifying areas of interest, then classifying the class of object in the area, and predicting the parameters of the bounding box.

The two-stage approach was first introduced in 2014 by Girshik [2]. His work R-CNN (Regions with CNNs) uses a selective search method [3] to detect regions of interest in input images and uses a regional classifier based on DCN (Deformable Convolutive Networks) to self-classify regions of interest. Fast-RCNN [4] improves R-CNN by extracting regions of interest from feature maps. Faster R-CNN [5] is a modification of the method of Fast R-CNN and R-CNN. The method is based on the idea of region proposals. The key difference between Faster R-CNN and its predecessors is that regions are calculated not from the original image, but from the feature map obtained from the convolutional neural network. To do this, a module called Region Proposal Network (RPN) was added. Obtained with the help values are passed to two parallel fully connected branches: bounding box prediction (regression) and classification framework. The outputs of these layers are based on the so called anchor areas (ancor boxes) – several frames for each position of the window, having different sizes and aspect ratios. The regression layer for each such rectangle produces 4 parameters that adjust the position of the bounding rectangle, and the classification layer produces the probability that the rectangle contains an object and the probability that the object in the frame corresponds to each of the classes. Cascade R-CNN [6] solves the problem of increasing the accuracy of the bounding box detection by applying a sequence of detectors with varying thresholds.

In single-stage approaches, there is no stage of finding regions of interest, the regression of bounding boxes and the classification of candidates in anchor areas is performed directly. Because of this, these architectures are more computationally efficient than two-stage architectures, while maintaining a competitive accuracy-performance ratio. SSD (Single Shot Detector) [7], uses a single neural network that performs all the necessary calculations and eliminates the need for resource-intensive methods of region proposals predicting. SSD place the anchors densely over the input image and uses the features of different convolutional layers for the regression and classification of anchor regions. DSSD [8] adds a deconvolution layers inside the SSD to interconnect features from the top and bottom layers. YOLO (You Only Look Once) [9] uses a small number of anchor regions (dividing the input image with a rectangular grid) and is based on the VGG-16 neural network. YOLOv2 [10] improves the performance due to the use of a new method of bounding the regression framework and a new neural network Darknet-19. YOLOv3 [11] continues to improve Darknet-19, offering a deeper neural network with skip connections. Architecture YOLOv2 and YOLOv3 allow to change the balance between accuracy and speed of detection by varying the number of areas able to solve the problem of detection in real time.

A slightly different approach is used by CenterNet [12], a detection algorithm based on methods for key points detection using neural networks. It learns to predict the centers of objects and form a feature map. The parameters of the bounding rectangle are then regressed for the detected centers. Corner Net [13] is another detection algorithm based on key points prediction. Unlike the CenterNet, CornerNet detects an object using a pair of corners of its frame.

3. Indoor positioning system

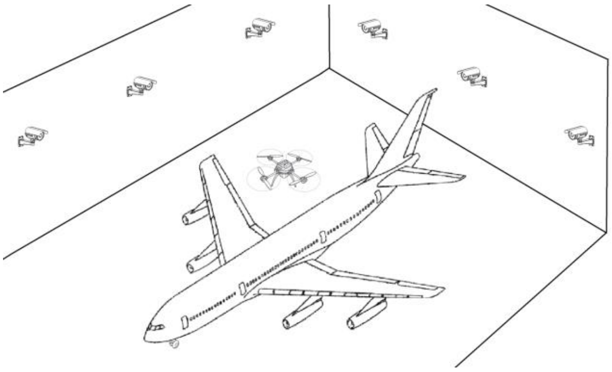

Fig. 2. Indoor cameras-based drone positioning system.

As part of the work, an original scheme of the organization of the internal positioning system was developed. Video cameras (4 or more) are placed in the hangar space in a certain way, the orientation parameters of which are pre-determined during the calibration of the system. The cameras are connected to a server that receives and processes video data. The UAV itself is considered as a target object, which is detected on the frames of the received video streams by the detection algorithm. The algorithm parameters are trained to detect the drone of the selected model. In our case, it was a DJI Phantom 3 Advanced drone. Based on the position of the drone on the frames and orientation of the cameras, its spatial position is calculated. The calculated coordinates of the object are transmitted to its on-board control system via Wi-Fi channel. The scheme of the proposed navigation system is shown in Fig. 2.

To calculate the three-dimensional coordinates of the object based on its position in the images, a method is used, which is a special case of block triangulation by the method of ligaments [14]. Since the camera orientation parameters are known, only three unknown 3D coordinate values are calculated. The idea of the method is to minimize the deviation of the projection of the calculated three-dimensional point on the image from the real position of the object (more precisely, the sum of squared errors for all cameras). The projection equations are:

$$x_i = -f_i \frac{a_1(X_g^i - X) + b_1(Y_g^i - Y) + c_1(Z_g^i - Z)}{a_3(X_g^i - X) + b_3(Y_g^i - Y) + c_3(Z_g^i - Z)}, \qquad (1)$$

$$y_i = -f_i \frac{a_2(X_g^i - X) + b_2(Y_g^i - Y) + c_2(Z_g^i - Z)}{a_3(X_g^i - X) + b_3(Y_g^i - Y) + c_3(Z_g^i - Z)}, \qquad (2)$$

where (X_g^i, Y_g^i, Z_g^i) are camera positions, (X, Y, Z) is 3D object position, (x_i, y_i) is its projection on image i, f_i is focus distances,

$$R_i = \begin{pmatrix} a_1^i & a_2^i & a_3^i \\ b_1^i & b_2^i & b_3^i \\ c_1^i & c_2^i & c_3^i \end{pmatrix}$$

is rotation matrix for camera i.

This is a well-known problem, which is solved by the method of iterative approximations. Each increment step of the three-dimensional coordinates ΔX is determined from the solution of the system of equations:

$$A^T A\ \Delta X + A^T B = 0,$$

where A is the matrix of partial differential of projection equations (1),(2) by drone coordinates over all cameras (size 3*3*number of cameras in the system), B is the discrepancy vector (size 2*number of cameras), containing deviations of object projections from real positions on images.

4. Detection algorithm details

As the detection algorithm YOLOv2 [9] CNN architecture was used. This architecture is slightly concede to YOLOv3 in accuracy, but has a higher calculaton speed, and demonstrates one of the best ratios of accuracy and performance, which in our task is of key importance. Performance determines the frequency of control signals delivered to the drone, which directly affects the accuracy of control and maximum safe flight speed.

The network receives a three-channel image as input, and outputs a tensor of size X×X×Y, where X is the number of cells in the input image. The length of the tensor Y depends on the number of classes detected and the number of anchor regions in the cell. For each anchor area, 5 basic parameters are calculated: the coordinates of the upper left corner of the rectangle, the width, the height, and the probability that this rectangle contains any object. In addition, the probability of the object belonging to each selected class is determined. The hyperparameters of the algorithm are the number and size of the anchor areas and the size of the input image.

The image size determines the number of cells for which the features are calculated, since the cell size is fixed and is equal to 32x32 pixels. Therefore, it directly affects the performance and accuracy of the network, as the number of cells increases the number of network filters. On the other hand, if there are more cells, each of them contains fewer objects; the features calculated in it correspond more accurately to each object and allow to build a more reliable prediction. The plot in figure 3 shows the dependence of the FPS, precision, recall and accuracy of the object frame (by the metric Intersection over Union, IoU) on the image resolution. Despite the slight increase in accuracy, 576×576 (18x18 cells) resolution was chosen to improve performance.

To maintain a balance between speed and accuracy, the number of anchors is set to 5, as the higher number of anchor areas decreases performance. K-means clustering of bounding rectangles on our training data set was used to determine anchor sizes.

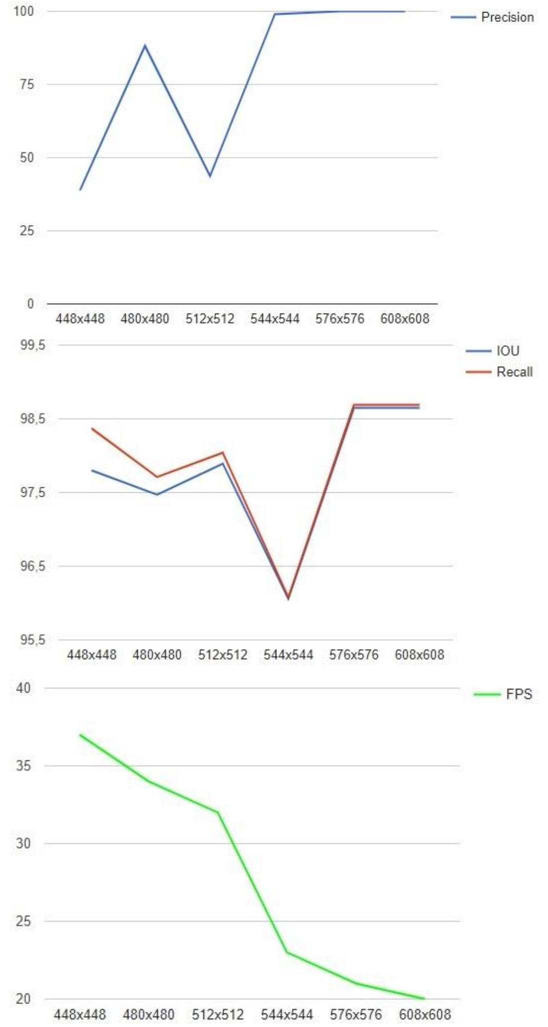

Fig. 3. Dependency of FPS, precision, recall and IoU on image resolution.

5. Automated data preparation

Since the work uses AI detection algorithms, training data is required to learn them. In our case, data are images with annotation: the type of object and its coordinates (bounding box) in the image. The CNN detectors used are very flexible but have a very large number of parameters. In this regard, a lot of training data is required. In order to avoid time-consuming manual data labelling, automatic synthesis of images was used for training and testing the algorithm.

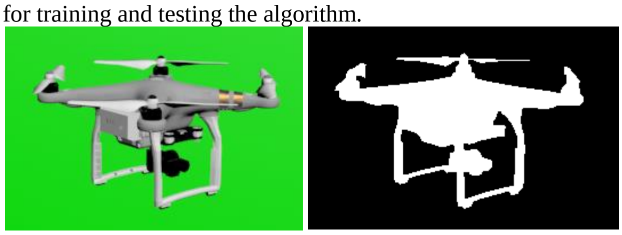

Fig. 4. Rendered 3D drone model and its mask.

The data were synthesized based on the rendering of the existing three-dimensional model of the drone (Fig. 4). Special 3D environments were not used during data endering, as their preparation requires additional manual labor of designers. Instead, the process was structured as follows. The model of the drone in different angles was rendered in a 3D modeling system on a uniform-colored background. The object in the image was automatically cut out, and its mask was built. Then the image and mask were subjected to random transformations: rotation, scaling, displacement, reflection, perspective transformation, blurring, salt/pepper noise, shift of color channel values (Fig. 5). After that, the image of the object on his mask was ovelayed on arbitrary backgrounds. Both random images and images from the test hangar where the subsequent testing was carried out were used as backgrounds. In order to make such insertion look natural and the network did not remember overlay artifacts as informative features of the object, local smoothing of objects with a Gaussian filter with randomized intensity was performed. In addition, objects from the Coil-100 collection were added to the images to increase the discriminative ability of the network [15].

To prepare the test data and expand the training base through real images, the real drone flight and video capture in the test hangar were carried out (Fig. 6). To get rid of manual annotation video files, the following automatic labellng algorithm was used. Optical flow maps were calculated for each video frame. The area with the maximum magnitude of the optical flow was selected on the maps. Since normally there were no other moving objects in the experimental scene, this area was thought to correspond to a drone. Sometimes due to the presence of foreign moving objects and shadows, as well as inaccuracy of segmentation, such labelling contained several errors. An experimental study was devoted to the estimate of the influence of different types of training data on the detection results.

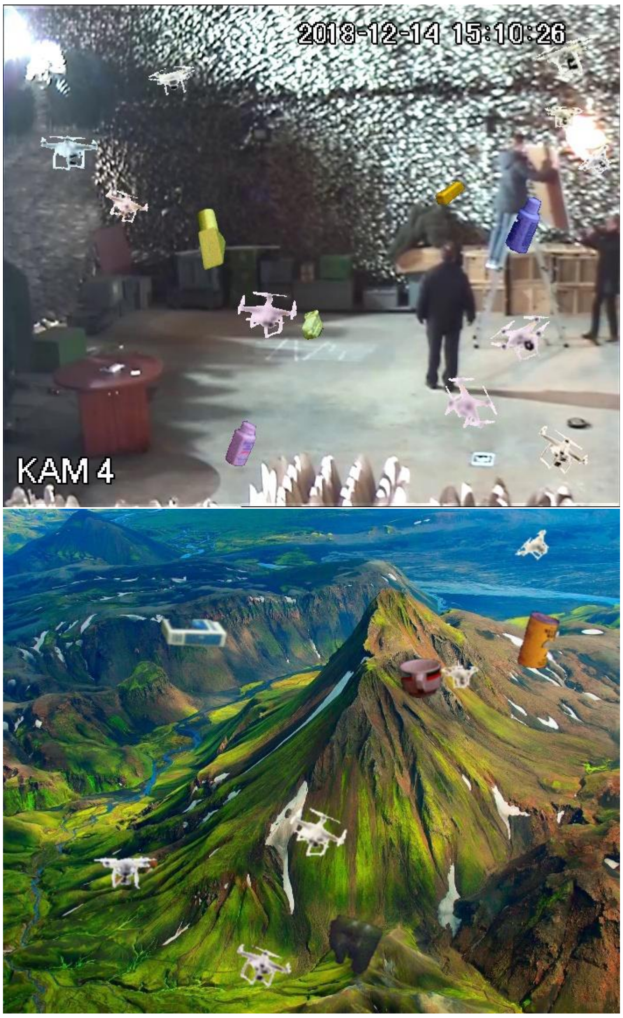

Fig. 5. Examples of training samples. Image having real hangar image as a background and random image.

6. Experiments

The accuracy of the detection algorithms was tested in a series of computational experiments on various training collections. We had three main data collections: *synthetic*, where images of drones were obtained by rendering 3D models, and the backgrounds are taken arbitrarily; *semi-synthetic*, where backgrounds for rendering was real images of the hangar in which the experiment was carried out; and *autolabelled real*, obtained by automated labelling drone videos (using optical flow). Incorrectly labelled data was manually deleted. Testing data was the part of the real data collection that was not used for training. The obtained precision, recall and IoU for different sets of training data are shown in table 1.

According to the results of the experiments, the following conclusions can be drawn. First, it is possible to train the algorithm with high accuracy on fully synthetic data, which was the purpose of the work. Secondly, the smoothing of objects when overlaying them on the background image plays a crucial role. Without smoothing, artifacts at the boundaries of objects become too important feature for the neural network, and it overfits to detect only artificial objects. Third, the use of a large number of random backgrounds was better than the use of a small number of real backgrounds from the test hangar. Despite the fact that the background images on the test data were similar to the training ones (but not the same), the network overfits, that means it has a low generalizing ability and does not cope with new scenes. Fourth, the inclusions of random objects (distractor) in the training images allowed significantly improve the accuracy of the work. Although these objects are not labelled in the test data, the network has learned to better distinguish drones from any other objects (see table 1).

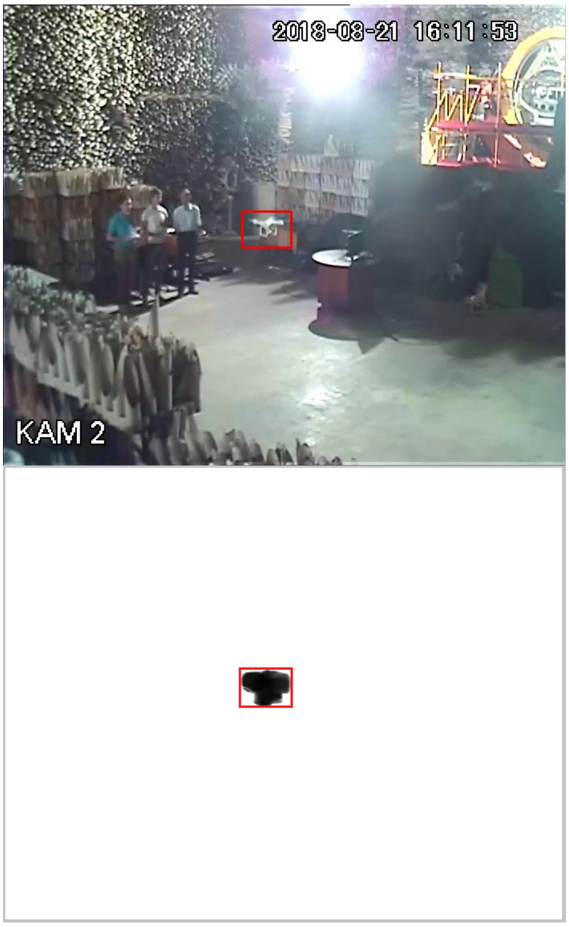

Fig. 6. Real video frame from experimental hangar and corresponding optical flow map.

In addition, during the experiments it was found that when training the network on the data obtained by the above-described autolabelling method, the accuracy was worse than on synthetic data. This is due to the fact that optical flow map is blurred, and the resulting bounding box is greater than the real object bounding box (Fig. 6). Also, the available real data are not sufficiently diverse.

Table 1. Detector testing results

Train Data	Precision	Recall	IoU
Synthetic	**100%**	**98.69%**	**98.65%**
Synthetic without disctractors	27.53%	97.71%	97.63%
Synthetic without smoothing	18.25%	86.60%	86.41%
Semi-Synthetic	41.13%	92.48%	91.58%
Semi-Synthetic without smoothing	45.56%	93.32%	98.64%
Autolabelled real data	33.92%	37.91%	37.89%

7. Conclusion

The paper describes the indoor drone positioning technology based on stationary visual sensors and the algorithm of drone detection. Given camera orientation and detection results the 3D position is reconstructed using a special algorithm of iterative minimization of the total reprojection error. The ways of adaptation of the CNN-based detector to the subject area were investigated. Both the automated process of creating training data and hyperparameter tuning are described. The influence of the data generation methods on the result is studied, in particular the inclusion of distracting objects in the data, artifacts of object overlay, the use of various background images. Conducted experiments showed that it is possible to train high accuracy detector exclusively on automatically synthesized images obtained using the renderings of a three-dimensional model of the drone without any real samples.

8. Acknowledgements

This work was supported by the Russian Foundation for Basic Research, project no. 17-08-00191 a.

9. References

[1] Yu.B. Blokhinov, V.A. Gorbachev, A.D. Nikitin, S.V. Skryabin. Technology for Visual Inspection of Aircraft Surfaces using Programmable Unmanned Aerial Vehicles. Journal of Computer and Systems Sciences International. Received by the editor 28.06.2019.

[2] R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, p. 580–587, 2014.

[3] J. R. Uijlings, K. E. Van De Sande, T. Gevers, A. W. Smeulders. Selective search for object recognition. International journal of computer vision, 104(2):154–171, 2013.

[4] R. Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, p. 1440–1448, 2015.

[5] S. Ren, K. He, R. Girshick, J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, p. 91–99, 2015.

[6] Z. Cai and N. Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 6154–6162, 2018.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. ECCV, p. 21–37. Springer, 2016

[8] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A. C. Berg. DSSD: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659, 2017.

[9] J. Redmon, S. Divvala, R. Girshick, A. Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, p. 779–788, 2016.

[10] J. Redmon, A. Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, p. 7263–7271, 2017.

[11] J. Redmon, A. Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[12] X. Zhou, D. Wang, P. Krähenbühl. Object as Points. arXiv preprint arXiv:1904.07850v2, 2019.

[13] H. Law, J. Deng. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European conference on computer vision, p. 734–750, 2018.

[14] A.P. Mikhailov, A.G. Chibunichev. Photogrammetry – MIIGAIK Publishing, Moscow, 2016, 294 p. (In Russian language)

[15] S. A. Nene, S. K. Nayar, H. Murase. Columbia Object Image Library (COIL-100). Technical Report CUCS-006-96. February, 1996.