Object detection for robotic grasping using a cascade of convolutional networks

Vitek Rais
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
vitek.rais@student.upce.cz

Petr Dolezel
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
petr.dolezel@upce.cz

***Abstract*—Robot guidance in industry is a significant issue that needs to be dealt with in modern manufacturing facilities. One of the common tasks in this area is the pick and place problem. For proper implementation of an automatic pick and place application using a robotic arm for object grasping, it is necessary to detect the accurate pose of the objects of interest. In this contribution, a novel engineering approach to object positioning, based on image processing is proposed. In this approach, the operation is composed of a cascade of convolutional neural networks. This cascade consists of 2 different types of networks. The first one is the object detection network called YOLOv5. It is used to process the raw image data from the scene to provide precise localization and determine the position of the objects of interest. After that, crops of the detected objects are created and processed by the second neural network, namely EfficientNet. This classification network is used to determine the rotation angle of the detected objects. The proposed approach provides a precision rate of 0.997 and a recall rate of 0.999 for locating and determining the correct position. For angle classification, EfficientNet provides an accuracy of 0.951. All tests are performed on the testing set of the legitimate positioning problem.**

***Index Terms*—Object detection, Pick and Place, Convolutional neural network, EfficientNet, YOLO**

I. Introduction

Efforts to make industrial processes more effective are very often oriented towards process automation, and thus towards eliminating or at least reducing the necessity of human intervention in these processes. Increasingly significant in the process automation is machine vision [1], a technology that provides automatic analysis of image-based data to extract the necessary information to control these processes. This increasing significance is driven primarily by the significant advances in machine vision, which are mainly due to the rapid development and increased use of deep learning methods [2]. This leads to the increasing use of applications such as automatic inspection [3] or robot guidance [4].

One of the common robot guidance processes is the pick and place problem, and its automation leads to greatly reduced production costs. Solution of this problem should allow a robotic arm to pick up an object located at any position in a given space and move it to a predefined location. This problem can be divided into three basic areas. The first is acquiring the precise position of the objects of interest in the scene. The second is proper grasping of the object. The last one is controlling the movement of the robotic arms. Pick and place solutions and systems can then be grouped into several categories based on various criteria. Examples of many instances can be found in [5].

In this paper we concentrate on the first part of this problem, i.e., detecting the exact position of the object. Namely, it is about registering objects lying on a flat surface and providing precise enough information about its pose in order to control the robotic manipulator. There are several different approaches for this problem, which differ based on the type of sensors or algorithms used. In this case, the focus is on the accuracy of the detection and its speed.

The aim of the contribution is clearly defined in section II. Next section is focused on the used methods and the proposed solution of the problem. After that, a section describing the implementation of the given problem follows. Then the outcomes are summarized. Lastly, some conclusions are discussed at the end of the paper.

II. Problem formulation

The aim of this work is to create an object detector for the pick and place problem. The detector works with only a single type of object of interest, which is a non-trivial metal component used in the automotive industry. This object must be located on a flat surface, which can be e.g. a conveyor belt. The object of interest then lies directly on this surface, does not occur in multiple layers and does not overlap at all. The detector determines the pose of each object. Since the objects always lie entirely on a flat surface, five different object positions can be distinguished when viewed overhead, which can be seen in Fig. 1. Additionally, the rotation angle of the object has to be determined, e.g. for accurate grasping of the object when using a parallel gripper. The rotation angle is not determined precisely. Only an interval, within which the actual rotation angle lies, is found. For example, if the actual rotation angle of the object is 5°, the detector should determine the interval to be $<0°, 10°)$. The size of the interval can be set during the preparation of the dataset for training, see Section IV for more information.

Important criteria for the evaluation of a given detector include not only the detection accuracy but also its speed,

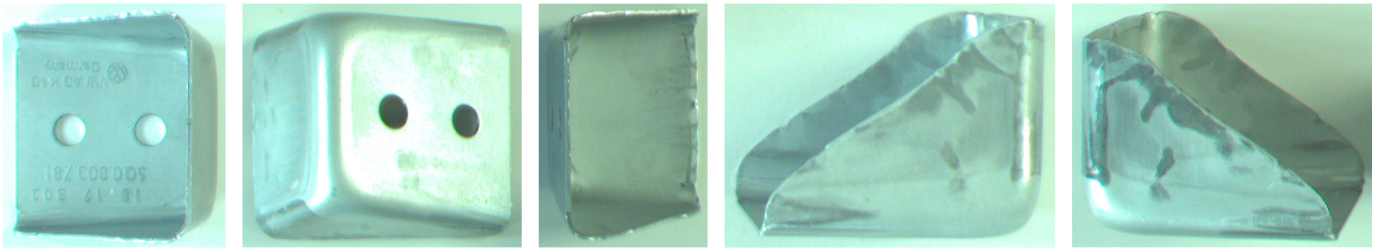

Fig. 1. Possible orientation of the object of interest.

which should be sufficient for the possibility of using the detector in real-time applications. Other important attributes of the detector include the low purchase price of the hardware and its easy configuration.

III. Proposed solution

A system containing two main components has been designed. The first component is responsible for acquiring data from the scene for further processing. The second component is responsible for processing those data obtained by the first component. From this data, it extracts relevant information about the position and angle of rotation of each object of interest. This information is then passed to the control unit, which uses it to control the behaviour of the robotic manipulator.

A. Data acquisition

As a sensor for data acquisition, considering the objectives specified in Section II, a standard industrial RGB camera with a suitable lens is used. The specific lens selection and overall setup depends on the conditions of the particular application. There are many guides and tools available for proper setup. In this paper, the tool from Basler [6] is used.

B. Processing unit

This component is used to process the raw data from the RGB camera, extracting relevant information about the position and rotation angle of each object of interest. For this purpose, an innovative approach combining object detection and object classification algorithms into a cascade is proposed. The algorithms for both steps are based on convolutional neural networks which are currently the most successful. This is demonstrated by the results obtained on various benchmark datasets [7], [8].

The raw image data are first processed using a detection algorithm that determines the location and position of all objects of interest in a given image. Then, all the objects of interest are cropped. These crops are processed using a classification algorithm that determines the interval in which the real rotation angle of a given object is located. Finally, the information from both steps is combined and transformed into the required format, e.g. an annotated image. This approach is well documented in Fig. 2.

The particular algorithm for object detection is selected from the one-stage detectors, mainly because of their very high speed. Even so, there are a large number of specific architectures from this category among the popular ones in recent years, e.g. SSD [9], RetinaNet [10] or YOLO [11] in all its versions. Out of these options, the YOLOv5s [12] algorithm is selected, because it combines very high speed with good detection accuracy. Other advantages include its ease of use and very good detection accuracy even when using a relatively small training set.

There are also many different algorithms for image classification. Popular ones include ResNet [13], DenseNet [14] or SENet [15]. The algorithm from the EfficientNet [16] family is finally chosen. In its design, strong focus is given not only to accuracy, but to the overall efficiency of the classification; thus its name. This makes these algorithms not only highly accurate but also very fast. Out of this family of algorithms, the smallest one, called EfficientNet-B0, is selected.

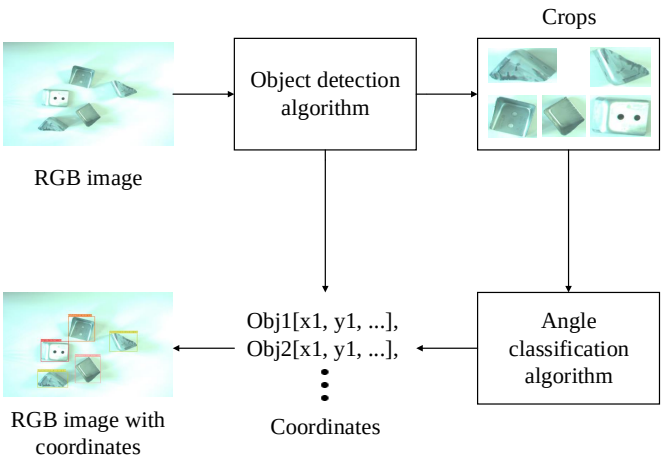

Fig. 2. Proposed processing approach.

IV. Experiment setting

An experimental workplace is arranged to perform a testing process - see Fig. 3. The sensor and the processing unit are the system's two primary components. As a sensor, a standard industrial camera Basler acA2500-14uc [17] is used. This camera records in RGB format with a sufficient frame rate of 14 fps and a high resolution of 5 MP (2590 x 1942 px). The camera is equipped with a Computar H0514-MP2 [18] lens. It is set to capture a space of 300 mm x 420 mm from

TABLE I
PARAMETERS OF THE DATASET

-	Training set	Validation set	Testing set	Total
Images	350	100	50	500
Objects of interest	1471	364	204	2039

a distance of 500 mm, with the optical axis perpendicular to the workspace surface.

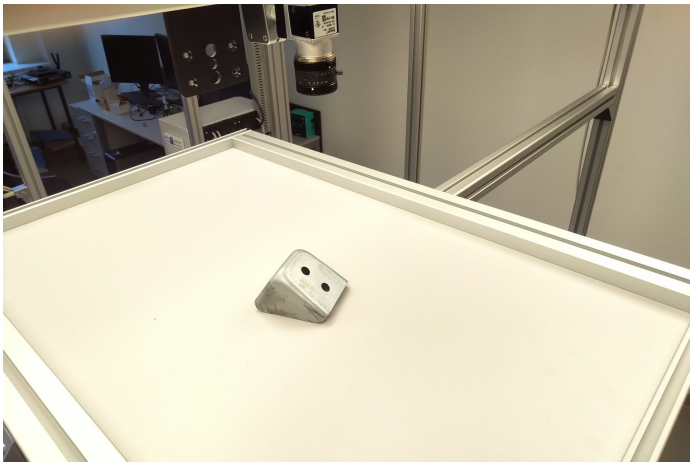

Fig. 3. Experimental workplace for object positioning.

The Lenovo Legion 5-15ARH05 [19] computer is selected as the processing unit. It has a NVIDIA GeForce GTX 1650 Ti graphics card. It also comes with an AMD Ryzen 5 4600H processor and 16 GB of RAM. All the used components are well accessible, but it would definitely be possible to use a more affordable camera and lens. Furthermore, it might be more suitable to use an embedded AI computing device as a processing unit.

A. Dataset for training

The dataset is prepared using the workplace shown in Fig. 3. A total of 500 images are created, each containing 2 to 5 objects of interest. The distribution of the number of objects in the images is as follows: only 2 objects contain 8% of the images, 3 objects contain 16%, 4 objects contain 36% and 5 objects contain 40% of the images. The rotation angles of the objects of interest are evenly distributed between 0 and 360 degrees for all five possible orientations. These images are then randomly divided into training, validation and testing sets in the ratio of 70 : 20 : 10 using stratified sampling. The overall information is summarized in Table I.

Next, target images need to be prepared for training, validation and testing of the YOLOv5 network for object detection. For each input image, an output image is also created, in which all objects of interest are labeled and their positions are specified. This process is performed manually by the application MATLAB Image Labeler [20]. The examples of input-target pairs are shown in Fig. 4.

To create the target images for training, validation, and testing of the EfficientNet classification network, the already

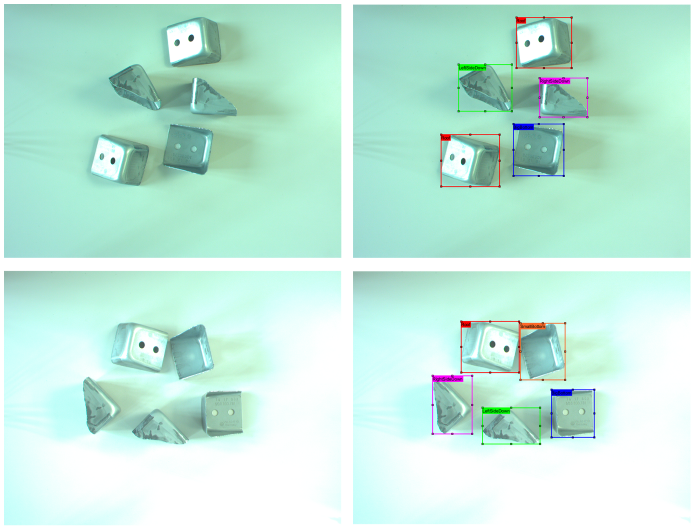

Fig. 4. Examples of input-target pairs.

TABLE II
PARAMETERS OF THE TRAINING YOLOV5S NETWORK

Input shape	640 x 640 x 3
Training algorithm	SGD algorithm
Maximum epochs	300
Stopping criterion	Maximum epochs reached
Learning rate α	0.01
Final OneCycleLR learning rate	0.01
Momentum	0.937
Optimizer weight decay	0.0005

trained YOLOv5 detection network is used. The YOLOv5 network is used to create crops of all objects of interest from the images of each set. These are then grouped according to object position and the interval where the object's rotation angle is located. In this case, 12 equally sized intervals are considered, with the first interval being $<0°$, $30°$). This means that the given network resolves a total of 60 classes.

B. Neural networks training

First, the YOLOv5 network is trained. The settings recommended by the authors are used for most parameters, and the most important ones are summarized in the Table II. Due to the relatively small size of the training set, data augmentation techniques are used. For color augmentations, the HSV color model is used. The hue is modified in the interval +-1.5%, saturation in the interval +-70%, and value in the interval +-40%. Moreover, images are translated by up to 10% in each direction. The last used techniques are left-right image flipping which is applied with probability 50%, and image mosaic which is used every time. Before training, the weights are initialized randomly, and no pre-trained model is used. As an optimizer, the default SGD algorithm is used, which proves to be a more suitable variant for this network [21], [22].

Consequently, the training of the EfficientNet network is performed. As in training the YOLOv5 network, data augmentation techniques are used. In this case, extra care is

TABLE III
PARAMETERS OF THE TRAINING EFFICIENTNETB0 NETWORK

Input shape	224 x 224 x 3
Training algorithm	ADAM algorithm
Maximum epochs	500
Stopping criterion	val_accuracy is not improved for 50 epochs
Actual epochs	197
Loss function	Sparse Categorical Crossentropy

TABLE IV
RESULTS OF YOLOV5S NETWORK

Dataset	mAP.50	mAP.50:.05:.95	Precision	Recall
Validation	0.9950	0.9946	0.9989	0.9989
Testing	0.9950	0.9920	0.9970	0.9990

needed. It is essential to avoid any techniques that affect the orientation of the images. Such techniques lead to devaluation of the obtained results. Therefore, only contrast adjustment is used in the interval +-25%, random zoom of the image is used in the interval +-10% while preserving the aspect ratio, and random translation is used by up to 10% in each direction. The ADAM algorithm is selected as the optimizer due to its acceptable performance in most cases [23]. Sparse Categorical Crossentropy is used as the loss function. The pre-trained model is not used, and the pre-training weights are set randomly. The most important training parameters are summarized in the Table III.

V. RESULTS

In this section, the performance of the proposed detector is evaluated. At first, the metrics used to evaluate the performance of each network are presented. Then, the performance of the entire detector is summarized, especially in terms of detection speed. The evaluation of the individual networks is performed on both the validation and testing sets. Different metrics are chosen for each network due to the different categories of these networks.

The metric chosen for the YOLOv5 network is the mean average precision (mAP) in two variants, namely mAP.50 and mAP.50:.05:.95. This metric is properly defined on the MS COCO dataset [24], which is commonly used as benchmark dataset for evaluating object detection algorithms. However, two additional metrics, precision and recall, are added. Results of the YOLOv5 network are summarized in Table IV.

For the evaluation of the EfficientNet network, the (top 1) accuracy metric is chosen, as is usual for classification networks. In addition, one more metric, total loss, is added. Results of the EfficientNet network are summarized in Table V.

TABLE V
RESULTS OF EFFICIENTNETB0 NETWORK

Dataset	Accuracy	Total loss
Training	0.9932	0.0193
Validation	0.9175	0.4211
Testing	0.9509	0.2442

Using the hardware described in section IV, the proposed detector is able to process up to 3 frames per second.

According to the results, the designed positioning system performs excellently on the training images and the detection network works excellently even on the test images. For the angle classification network, a fairly significant decrease in accuracy for both the validation and testing sets compared to the training set is observed. This means that an overfitting problem occurs. Despite this, the accuracy above 90% on the validation set and even above 95% on the testing set is a more than acceptable result. A detection speed of around 3 FPS is sufficient for some real-time applications, but for some specific applications, this speed may not be satisfactory.

VI. CONCLUSION

An innovative engineering approach to object detection for pick and place applications using a robotic manipulator for object grasping is proposed in this contribution. For this purpose, it is necessary to detect, not only its location, but the accurate pose of the object of interest. The proposed approach is based on the cascade of convolutional neural networks of different types. Firstly, the YOLOv5 network is used for the localization of all objects of interest and to determine their positions. Then, crops of all detected objects are created. These are the inputs to the EfficientNet neural network. This network is responsible for determining the rotation angle interval in which the actual rotation angle is located. The outputs of the detector are the merged information from both networks. For this particular problem, YOLOv5 provides a precision rate of 0.9970 and a recall rate of 0.9990 with the testing set. The EfficientNet provides an accuracy of 0.9509.

The presented contribution should be understood as a first step in the development of a robust object detector. In the near future, there will be work on enhancements to the system in several ways. First of all, the process of determining the rotation angle should be optimized so the size of the obtained interval is minimized while preserving high accuracy. The next work will be focused on optimizing the neural network architectures and the method of transferring information between them. This reduces the computational complexity and overall increases the detection speed.

REFERENCES

[1] H. Golnabi and A. Asadpour, "Design and application of industrial machine vision systems," *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 6, pp. 630–637, 2007.

[2] T. Serre, "Deep learning: the good, the bad, and the ugly," *Annual review of vision science*, vol. 5, no. 1, pp. 399–426, 2019.

[3] S. Cubero, N. Aleixos, E. Moltó, J. Gómez-Sanchis, and J. Blasco, "Advances in machine vision applications for automatic inspection and quality evaluation of fruits and vegetables," *Food and bioprocess technology*, vol. 4, no. 4, pp. 487–504, 2011.

[4] L. Pérez, Í. Rodríguez, N. Rodríguez, R. Usamentiaga, and D. F. García, "Robot guidance using machine vision techniques in industrial environments: A comparative review," *Sensors*, vol. 16, no. 3, p. 335, 2016.

[5] A. Björnsson, M. Jonsson, and K. Johansen, "Automated material handling in composite manufacturing using pick-and-place systems–a review," *Robotics and Computer-Integrated Manufacturing*, vol. 51, pp. 222–229, 2018.

[6] Basler, "Lens selector by basler," https://www.baslerweb.com/en/sales-support/tools/lens-selector/, 2022, accessed: 2022-12-25.

[7] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[8] Z. Zou, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *CoRR*, vol. abs/1905.05055, 2019. [Online]. Available: http://arxiv.org/abs/1905.05055

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

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

[12] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, TaoXie, J. Fang, imyhxy, K. Michael, Lorna, A. V, D. Montes, J. Nadar, Laughing, tkianai, yxNONG, P. Skalski, Z. Wang, A. Hogan, C. Fati, L. Mammana, AlexWang1900, D. Patel, D. Yiwei, F. You, J. Hajek, L. Diaconu, and M. T. Minh, "ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference," Feb. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6222936

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[14] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[15] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[16] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[17] Basler, "Basler ace aca2500-14uc - area scan camera," https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ace/aca2500-14uc/, 2023, accessed: 2023-01-04.

[18] Computar, "Mpz machine vision series h0514-mp2," https://computar.com/product/551/H0514-MP2, 2023, accessed: 2023-01-04.

[19] Lenovo, "Legion 5 (15, amd)," https://www.lenovo.com/cz/cs/laptops/legion-laptops/legion-5-series/Lenovo-Legion-5-15ARH05/p/88GMY501444, 2023, accessed: 2023-01-04.

[20] MathWorks, "Image labeler," https://www.mathworks.com/help/vision/ref/imagelabeler-app.html, 2023, accessed: 2023-01-09.

[21] T.-K. Nguyen, L. T. Vu, V. Q. Vu, T.-D. Hoang, S.-H. Liang, and M.-Q. Tran, "Analysis of object detection models on duckietown robot based on yolov5 architectures," *International Journal of iRobotics*, vol. 4, no. 4, pp. 17–22, 2021.

[22] D. Fu, L. Gao, T. Hu, S. Wang, and W. Liu, "Research on safety helmet detection algorithm of power workers based on improved yolov5," in *Journal of Physics: Conference Series*, vol. 2171, no. 1. IOP Publishing, 2022, p. 012006.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.