Multiple Objects Localization Using Image Segmentation with U-Net

Dominik Stursa
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
dominik.stursa@upce.cz

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

Daniel Honc
Faculty of Electrical Engineering and Informatics
University of Pardubice
Pardubice, Czech Republic
daniel.honc@upce.cz

***Abstract*—Precise object localization in an industrial environment is a significant task affecting follow-up processes for a pick and place application. One of the solutions to effectively ensure the success of this task is to use modern methods of machine vision. Machine vision is still a highly evolving topic, in which the use of approaches based on convolutional neural networks is rising. And so in this contribution, an innovative engineering approach based on convolutional neural networks is proposed for an object localization task. The approach is based on an atypical image segmentation, where the individual objects are represented by two colored gradient circles. These circles represent significant parts of the object like its center or ending. Each object type (class) is determined by a specific color. By use of a local maxima finder, all circles in an image are transformed to points. With knowledge of these points the coordinates and rotations are calculated. The proposed approach was tested on a legitimate localization problem with 100% precision, more than 99.52% recall on the positioning task and with an average of 6 minutes angle variance per object.**

***Index Terms*—Machine Vision, Object Localization, U-Net, Convolutional Neural Network**

I. Introduction

In recent decades, many manual activities performed by humans have been increasingly replaced by the use of robots. This is especially in areas where human safety needs to be ensured or where work is repeated. Along with the increasing use of robots, the use of machine vision for image-based analysis to ensure automatic robot operations is also growing. The machine vision in industry fields is often used in applications such as automatic inspection [1], robot guidance [2], [3] and process control [4].

One of the most common problems that needs to be solved is the solution of robot guidance for a pick and place application in an industrial environment. In general, the pick and place task is based on a robotic arm (or multiple robot arms) that is able to grab an object and move it to a certain position. From a theoretical point of view, the role of object gripping and placement can be considered as a solved problem. However, from a practical point of view, it still offers many issues that demand a solution [5].

The solution of the pick and place problem is directly affected by various criteria. The common criterion can be the dimension of the space from which we want to grab the objects and in which we want to place them. Therefore the solution is basically limited to the problem of a handling strategy between two flat surfaces (2D to 2D) and between a flat surface and 3D area (2D to 3D) as exemplified in the survey [6].

In this contribution, we deal with precise localization of specific objects of interest, which is the initial part of the pick and place application. This mentioned field is still an extensively researched topic as industrial companies want to save costs according to used technologies. These costs are mainly and directly affected by the type of used sensor. As might be expected, there exists a huge variety of object localization approaches combining different types of sensors with different types of algorithms. A brief summary of state-of-the-art techniques can be found in [7]. Here, the emphasis is placed on quick and precise estimation of the position and orientation of multiple objects on flat, or possibly moving, surface in frames captured by a monocular camera.

II. Problem Formulation

This section serves for proper definition of paper aim. As mentioned before, we deal with the initial challenge of the pick and place problem. Specifically, the proper localization of objects of interest. In this field, a lot of approaches have been proposed for object localization in recent years as described in [8], [9] and [10].

Very recently, due to the growing availability of laser scanners (e.g. Hexagon 3D Scanners [11]), point clouds are more often used for object localization [12]. Obviously, an object localization system, combining the use of laser scanners and advanced point cloud processing algorithms, represents a very robust solution. However, this kind of a solution still has some imperfections. Firstly, these laser scanners are slightly expensive, which is causing an inability for use in low cost productions. Secondly, the performance of laser scanners decreases with the reflexivity of object material (especially for shiny ones). Finally, the frame rate of these sensors is too

Supported from ERDF/ESF. ”Cooperation in Applied Research between the University of Pardubice and companies, in the Field of Positioning, Detection and Simulation Technology for Transport Systems (PosiTrans)” (No. CZ.02.1.01/0.0/0.0/17_049/0008394).

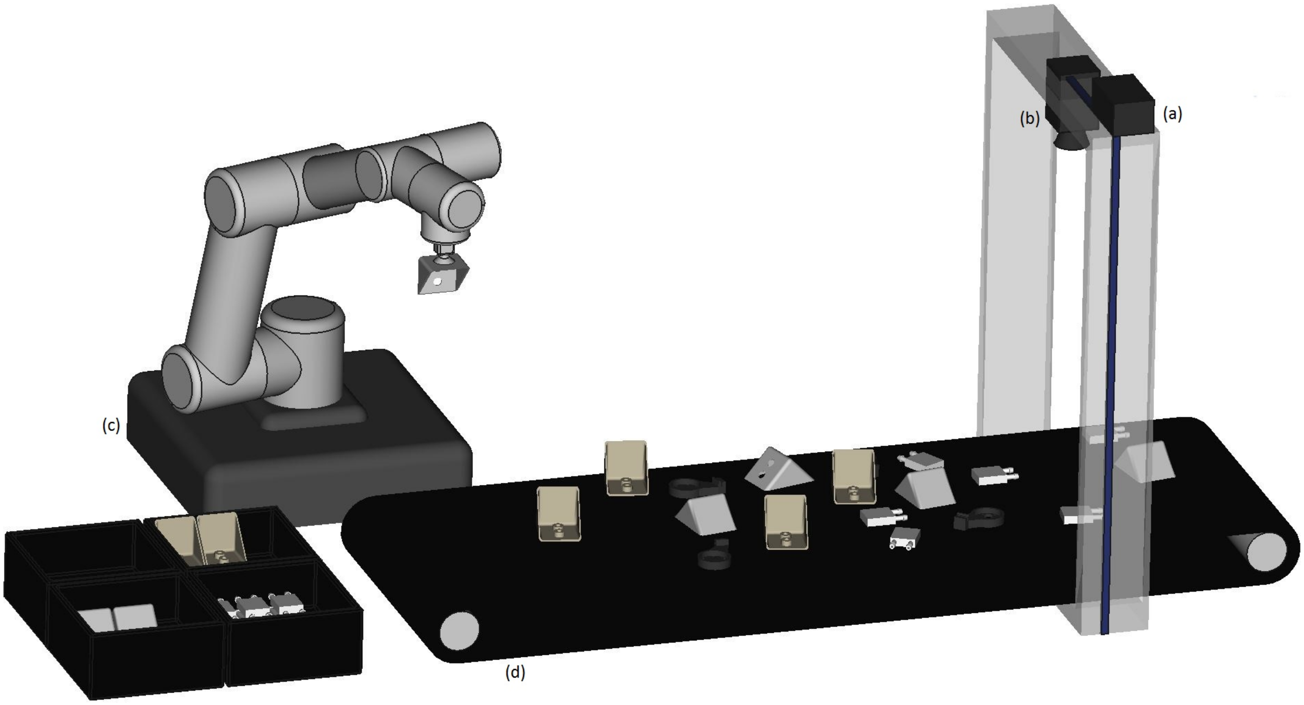

Fig. 1. Example of configuration for pick and place application.

low to be used for objects moving on the surface (e.g. on the conveyor belt).

Thus, our aim is to propose and test a competitive approach which eliminates mentioned shortcomings. In order to satisfy needs for the industrial environment, we need to meet the following parameters. At first, the price of whole device should not exceed a 1000 dollars and its composition should be based on available hardware. Furthermore, the solution shouldn't be sensitive to changing conditions, i.e. changes of light or surface background.

Moreover, the system processing time should be fast enough to provide localization of objects on a moving conveyor belt. To fulfill this requirement the frame rate should be greater than 10 frames per second. In addition to that, our solution should simultaneously provide coordinates and rotations of different objects in a specific application.

Overall, the proposed solution could be used for the pick and place application. Basically, the whole system for this kind of application is composed of the processing unit (a), camera sensor (b), robotic manipulator (c) and conveyor belt (d) as demonstrated in Fig. 1.

III. Proposed solution

Here we want to present a solution using the processing unit (a), with a proper image processing algorithm, and imaging sensor (b) as also shown in Fig. 1. The imaging sensor provides an area scan which is the source signal for further processing. This signal is then processed by the processing unit which provides particular information about detected objects. In our case, this information is object position and its rotation, which serves for manipulation purposes. This information can be handed over to the superior system, which provides control of the robotic manipulator and conveyor belt. Particular parts are described below.

A. Imaging sensor

As an imaging sensor we decided to use the standard industry monocular RGB camera with sufficient optical system for purposes of sensing with satisfactory frame rate. As the ability of image processing is affected by the source information stability, the correct setting (determined by conditions) in the particular application must be secured.

The camera placement above the sensing area has to provide coverage of the full area of interest. The optical system parameters have to be tuneable. Specifically, the focal point and aperture have to be correctly tuned to provide high image sharpness with balanced colors. A lot of tutorials for these settings are available online. Authors can recommend the tool provided by Basler Corporatioon [13]

B. Processing unit

Selection of the processing unit is conditioned by the method used for the localization. As the aim is to use localization based on the convolutional neural networks (CNN), the processing unit must provide sufficient performance. Fast parallel processing and low cost of the unit is required for our application. These criteria are fulfilled by several devices for developing of neural networks such as NVidia Jetson [14], Google Colar [15], Asus Tinker [16] etc.

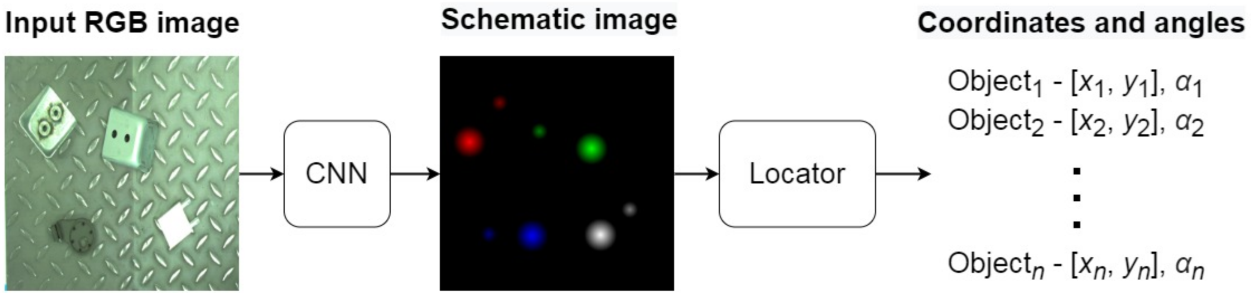
Input RGB image
Schematic image
Coordinates and angles
CNN
Locator
Object$_1$ - [x_1, y_1], α_1
Object$_2$ - [x_2, y_2], α_2
Object$_n$ - [x_n, y_n], α_n

Fig. 2. Diagram of proposed processing approach.

C. Localization algorithm

The processing algorithm directly affects accuracy and performance of the object localization task. The use of convolutional neural networks for this kind of task grows with its increasing progress. Nowadays, CNNs are outperforming traditional machine vision methods for object classification and localization as mentioned in [17], [18].

Based on an ability to accurately predict object position and authors experience [**?**] with CNNs, we propose an approach based on creation of a schematic image leading to image segmentation. This basis for further use in different applications has been established in previous authors' publications [19], [20].

Here, the detected objects are represented by two settings of two radial gradient circles of defined color (representing their class), while the rest of the transformed image remains black. In both settings, the one coordinate of the object is represented by a bigger circle and the second circle represents the other coordinate of the object. Accordingly to used settings, these circles represent different coordinates described as follows.

- Case 1 - Bigger circle represents one end of the object and the smaller circle acts for the other end.
- Case 2 - Center of the object is represented by a bigger circle and the end of the object is illustrated as the smaller circle.

When these coordinates are connected it creates the vector, which represents the object position with its rotation. This approach using the first setting is represented by the diagram in Fig. 2. Both particular cases of object representation are illustrated in Fig. 3.

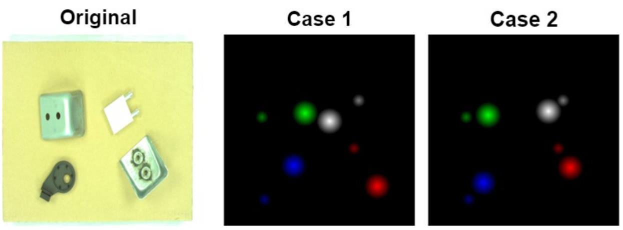

Fig. 3. Cases of object representation in segmented image.

In this proposed approach the Locator is a generic process finding two local maxima of each specific color representing object class. From these coordinates, position and rotation angle of the object is then determined.

As the original image transformation is the complex task, we propose to apply one representative network from the image segmentation neural network group. These networks are often based on the encoder-decoder operation. The encoding involves a feature key extraction and its compression to lower dimensions. The decoding then recovers these feature keys and project them to the segmented image. Typically, the segmentation creates a new image covered with monotonous areas of colors. From our previous experience we believe in the proper segmentation function with segmented objects represented by color gradients.

In recent years a lot of architectures for image segmentation have been developed, e.g. SegNet [21], ResNet [22] or PSPNet [23]. Based on our previous experience, we decided to use the U-Net network architecture, which was introduced in [24].

IV. Experiments procedure

In this section we are going to describe the design of the multiple object localization system and the experiments. The next subsection defines the object localization issue. Hardware implementation of the proposed localization system is then described. Next, the creation of a dataset for neural network training is traced. As following, the U-Net training with its parameters is written down. Finally, function of the locator and the process of obtaining center coordinate and rotation is explained.

A. Object localization issue setup

The special setup was prepared for the demonstration of the proposed solution. The setup was based on a placement of up to four different objects on six different surfaces. All used objects are shown in Fig. 4.

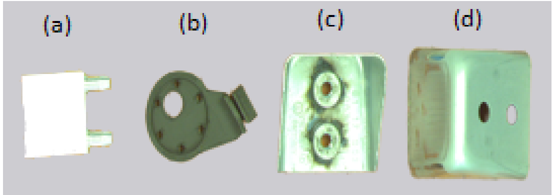

Fig. 4. Objects of interest, selected for localization issue.

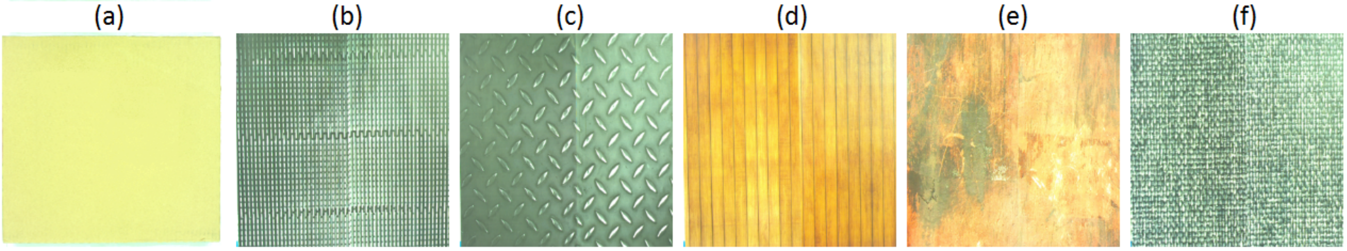

Fig. 5. Background surfaces used in experiments.

These objects were similarly sized and the first object (a) was made of aluminum, colored by white paint. The second object (b) was created by a printer from a black plastic material. The other two objects (c) and (d) were made from a shiny metal. Background surfaces were composed of a plastic plate (a), anti-slip sheets (b and c), small wooden slats (d), an old wooden desk (e) and a carpet (f). All of the used surfaces are shown in Fig. 5.

B. Hardware implementation

Hardware was composed of two elementary systems; the imaging sensor with its lenses; and the processing unit. As an imaging sensor, the industrial camera Basler acA2500-14uc was used. The Computar M3514-MP lens was assembled on the camera and set properly to obtain sharp images of the monitored area. The NVIDIA Jetson nano developer kit was chosen as the processing unit capable of maintaining the segmentation task using U-Net with a frame rate at 18 fps [25].

This developer kit possess a quad-core ARM A57 processor on 1.43 GHz for operating the system and its auxiliary functions; 128-core Maxwell GPU for data processing; 4GB 64-bit LPDDR4 RAM serving the high speed data transfers. Furthermore, the communication and operating interfaces are provided in a wide range. These are for example, two MIPI CSI-2 DPHY lanes for camera connections by flat cables (each capable to transfer 60fps 4k videos), Gigabit Ethernet, four USB 3.0 interfaces, HDMI, etc. The whole specification should be found on an official page [14].

The hardware costs were composed from its particular parts (lenses - \$150, camera - \$550, developer kit - \$100 and accessories - \$150). In total, the costs were \$950 and did not exceed the required costs. However, the total costs can be significantly reduced by using a lower resolution imaging sensor.

C. Dataset creation

Input images taken during the dataset creation in [26] were also used for purposes of this article.

Totally, 1021 images were gathered using different object positions on all of the surfaces. The size of images were transformed to [288 x 288]px for purposes of the proposed localization system.

Target images were created by user interface, where the position vector of each object was defined by user on each input image. Circles were inserted on its specific coordinates by its particular case. This process is illustrated in Fig. 6.

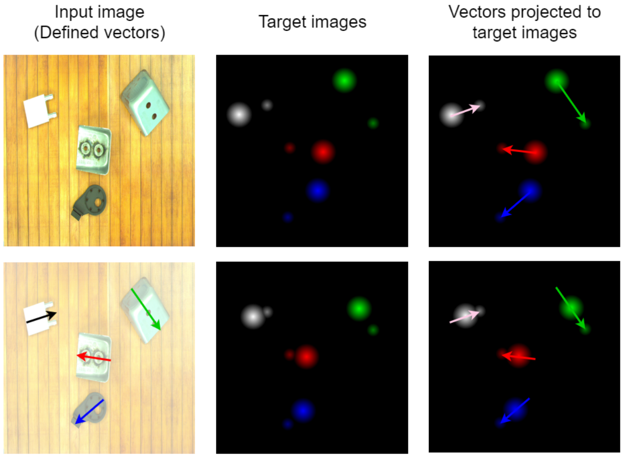

Fig. 6. Illustration of dataset creation process.

Both (input and target) images were equally split into two groups. The group for training contained 815 images and remaining images were assigned to the testing group.

D. U-Net training

In both cases, the training of U-Net topology was performed using the same training parameters. For the optimization strategy the ADAM algorithm was used, which generally provides sufficient performance [27]. Initial weights with Gaussian distribution were randomly set at the beginning of each training. The training was performed ten times for each case. In both cases, the best performing model was then selected for further comparison. Training parameters are summarized in Table I.

TABLE I
TRAINING PARAMETERS

Input shape	288 x 288 x 3
Experiments quantity	2 x 10
Optimizer	ADAM algorithm
Initialization	Normal distribution (mean = 0, std = 0.05)
Learning interruption	500 epochs reached
Learning rate α	0.001

E. Locator

Both the bigger and smaller circle are found by the locator in a segmented image. The two coordinates are returned from the locator for each object. Depending on the case, the coordinates of the center and rotation of each of the objects are calculated.

V. Results and discussion

For correct object localization determination, we compared the center coordinates of the predicted object with the actual object center coordinates. If the predicted coordinates were in a radius of 2 pixels and the object class was the same, then this was determined as the correct prediction.

The best performing model was selected in a maximum of two steps. At first, the total counts of correct predictions were compared. If the number was the same for multiple models, the sums of the rotation deviations of these competing models were compared. Thus, the best performing model was selected for each case.

As the highly used metrics for multi-class classification we chose two typically used indicators - precision and recall.

The proposed system showed an ability to not predict any ”false” objects. These predictions were also independent of the background surface. As such, there were not any false predictions in either case. Accordingly, the precision of the best performing model was 100% for both cases.

On the contrary, the system was not able to find all of the true objects. This ability is referred to as the recall, which is mathematically represented by the equation (1).

$$Recall = \frac{TP}{TP + FN} \qquad (1)$$

In the equations above, TP means true positive, FP means false positive and FN means false negative.

For both cases, the recall values of each object are written in Table II.

TABLE II
Comparison of both cases recall percentage

Case	Object1 (a)	Object2 (b)	Object3 (c)	Object4 (d)
Case 1	100.00%	98.71%	98.14%	99.34%
Case 2	100.00%	100.00%	98.75%	99.34%

As the results show, the second case model is performing better classification in all object classes than the model of the first case. Overall, both presented cases were performing well in both evaluated metrics. The recall macro average, defined as the average value of all of the object recalls, was 99.05% for case 1, and 99.52% for case 2.

After the classification issue, the comparison of true and predicted object rotations was performed. The average angle variance per component value is included in Table III.

TABLE III
Average angle variance per component

Case	Object1 (a)	Object2 (b)	Object3 (c)	Object4 (d)
Case 1	0.126°	0.066°	0.054°	0.104°
Case 2	0.069°	0.068°	0.138°	0.110°

All objects average angle variance was 0.088 degrees for case 1 and 0.097 degrees for case 2.

The second case provides better performance for the object localization task and bigger rotation error for the object. It needs to be mentioned that, the rotation error is affected by distance of points, where a closer distance should lead to a bigger error on a 1 pixel mismatch. As the rotation error was not more than one tenth of a degree, the rotation estimation can be considered as successful.

The proposed solution was able to process more than 15 frames per second with the hardware described in section IV-B.

VI. Conclusions

In this contribution, we proposed an innovative engineering approach to multiple object localization for pick and place applications. The proposed approach consists of two parts. The first is the creation of the schematic image, where the U-Net topology was used for this image segmentation task. In the schematic image, the objects were represented by two settings of two radial gradient circles of defined color (representing their class), while the rest of the transformed image remained black. The second part was about using the locator to maintain center coordinates and rotation of an object.

The solution was tested on a legitimate positioning problem, where multiple objects were placed on various surfaces. In this problem, both circle setting cases were compared in object localization tasks. In the case, where the bigger gradient circle was in the center of the object, better localization performance was realized, and its precision was 100% and recall was 99.52%. The rotation error was 0.097 degrees in total average angle variance.

In the future work, the proposed approach should face more complex tasks. For instance, the objects could be represented only by places where they can be gripped, which could provide direct information for robot manipulation purposes. We also plan to work on the U-Net architecture optimization, which should lead to reducing of the system computational complexity.

References

[1] P. Surya Prasad and B. Prabhakara Rao, “Review on machine vision based insulator inspection systems for power distribution system,” *Journal of Engineering Science and Technology Review*, vol. 9, no. 5, pp. 135–141, 2016.

[2] D. Moru and D. Borro, “A machine vision algorithm for quality control inspection of gears,” *International Journal of Advanced Manufacturing Technology*, vol. 106, no. 1-2, pp. 105–123, 2020.

[3] J. Fu, L. Zong, Y. Li, K. Li, B. Yang, and X. Liu, “Model adaption object detection system for robot,” vol. 2020-July, 2020, pp. 3659–3664.

[4] M. He, “Optimal control of sulphur flotation process based on machine vision,” *IFAC-PapersOnLine*, vol. 51, no. 21, pp. 169–174, 2018.

[5] M. Alonso, A. Izaguirre, and M. Graña, “Current research trends in robot grasping and bin picking,” *Advances in Intelligent Systems and Computing*, vol. 771, pp. 367–376, 2019.

[6] A. Bjornsson, 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.

[7] M. Fujita, Y. Domae, A. Noda, G. Garcia Ricardez, T. Nagatani, A. Zeng, S. Song, A. Rodriguez, A. Causo, I. Chen, and T. Ogasawara, “What are the important technologies for bin picking? technology analysis of robots in competitions based on a set of performance metrics,” *Advanced Robotics*, vol. 34, no. 7-8, pp. 560–574, 2020.

[8] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, “A survey of deep learning-based object detection,” *IEEE Access*, vol. 7, pp. 128 837–128 868, 2019.

[9] H. Y. Kuo, H. R. Su, S. H. Lai, and C. C. Wu, “3d object detection and pose estimation from depth image for robotic bin picking,” in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug 2014, pp. 1264–1269.

[10] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, “Std: Sparse-to-dense 3d object detector for point cloud,” vol. 2019-October, 2019, pp. 1951–1960.

[11] *Industrial 3D Laser Scanners*, 2021 (accessed January 7, 2021), https://www.hexagonmi.com/products/3d-laser-scanners/.

[12] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, “Registration with the point cloud library: A modular framework for aligning in 3-d,” *IEEE Robotics Automation Magazine*, vol. 22, no. 4, pp. 110–124, Dec 2015.

[13] *Tool For Lens Selection*, 2021 (accessed January 11, 2021), https://www.baslerweb.com/en/products/tools/lens-selector/.

[14] NVIDIA, “Nvidia jetson nano developer board,” https://developer.nvidia.com/EMBEDDED/jetson-nano-developer-kit, 2021, 2021-01-11.

[15] GOOGLE, “Google edge tpu coral dev board,” https://coral.ai/products/dev-board/, 2021, 2020-01-11.

[16] ASUS, “Tinker edge t,” https://tinker-board.asus.com/product/tinker-edge-t.html, 2021, 2020-01-11.

[17] P. Sharma and A. Singh, “Era of deep neural networks: A review,” in *8th International Conference on Computing, Communications and Networking Technologies, ICCCNT 2017*, 2017.

[18] Y. Xu, X. Zhou, S. Chen, and F. Li, “Deep learning for multiple object tracking: a survey,” *IET Computer Vision*, vol. 13, no. 4, pp. 355–368, 2019.

[19] D. Stursa, B. Zanon, and P. Dolezel, “Novel approach for person detection based on image segmentation neural network,” *Advances in Intelligent Systems and Computing*, vol. 1268 AISC, pp. 166–175, 2021, cited By 0.

[20] P. Skrabanek, P. Dolezel, Z. Nemec, and D. Stursa, “Person detection for an orthogonally placed monocular camera,” *Journal of Advanced Transportation*, vol. 2020, 2020.

[21] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Dec 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” vol. 2016-December, 2016, pp. 770–778.

[23] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” vol. 2017-January, 2017, pp. 6230–6239.

[24] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[25] NVIDIA, “Jetson nano: Deep learning inference benchmarks,” https://developer.nvidia.com/embedded/jetson-nano-dl-inference-benchmarks, 2021, 2021-01-12.

[26] P. Dolezel, D. Stursa, and D. Honc, “Rapid 2d positioning of multiple complex objects for pick and place application using convolutional neural network,” 2020, pp. 213–217.

[27] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980