ISSN 0146-4116, Automatic Control and Computer Sciences, 2023, Vol. 57, No. 5, pp. 459–468. © Allerton Press, Inc., 2023.

Augmenting a Pretrained Object Detection Model with Planar Pose Estimation Capability

A. Lapins[a], *, J. Arents[a], and M. Greitans[a]

[a] *Robotics and Machine Perception Laboratory, Institute of Electronics and Computer Science, Riga, Latvia*

**e-mail: andris.lapins@edi.lv*

Received May 8, 2023; revised June 1, 2023; accepted June 5, 2023

Abstract—This paper presents a 2D pose estimation solution to the bin-picking problem for robotic grasping systems. By extending a pretrained object detection model, namely DETR, with pose and visibility prediction heads we obtain classification, center, 2D rotation and occlusion scores for every detected object. The augmented model is trained and evaluated on synthetically generated images representing the real environment for faster and more flexible acquisition of data. The results show an average angle error of 3.23 deg for cylindrical and cuboid shape objects.

Keywords: bin-picking, DETR, synthetic data, planar orientation

DOI: 10.3103/S0146411623050061

1. INTRODUCTION

The trend towards greater automation of production lines in the modern industrial sector has fuelled ongoing research into how machines can be made more capable of performing tasks that demand human-like intelligence and agility. This necessitates the integration of advanced sensor and robotic technologies with sophisticated data processing techniques to effectively adapt to the inherent variability present in the surrounding environment [1, 2].

One significant challenge in the progression of smart manufacturing is the problem of robotic grasping perception, specifically in scenarios involving multiple, diverse, and overlapping objects, commonly referred to as bin-picking [3, 4]. Modern industrial robots exhibit remarkable precision and repeatability, and the success of bin-picking largely hinges on the accuracy of the perception system. To address this challenge, computer vision applications, particularly those leveraging statistical techniques, such as artificial neural networks, have been widely employed [5].

This paper focuses on the development of a perception system aimed at achieving a precise estimation of object locations and orientations within the bin-picking context. To accomplish this, we propose augmenting a pretrained object detection model with the capability to estimate the planar projections of object poses, which can then be utilized for full 3D pose estimation in conjunction with depth imagery.

The decision to separately estimate 2D rotation is the result of considerations between system complexity and performance. Generally, systems which directly estimate pose in 3D space suffer from poor precision or high inference time as in [6]. The work [7] performs well in precision and time, but at the cost of utilizing a multiple camera setup to train their neural network model. Pretrained object detection models by the years have advanced significantly in speed and precision on RGB images. We hypothesize that a slight extension of functionality, like planar pose estimation, to an already well-built network will not inflict significant performance issues. Afterwards, the time of estimating the normal vector on a surface of an object is an acceptable 0.5 s in average on our current system. In the end, our system has become modular lending itself to an easier upgradability.

In this paper, we begin by discussing related work conducted to achieve similar results in 2D or 3D pose estimation of target objects. Subsequently, we provide a comprehensive overview of our proposed approach for attaining 2D rotation estimation, offering detailed insights into our implementation and experimental setup. Finally, we present our results and draw conclusions based on the findings obtained from our experiments.

2. RELATED WORK

Bin-picking is an active area of research. The method proposed in [8] creates patches of detected objects using a histogram of oriented gradients, which is a feature descriptor to capture information about the underlying texture. Sequentially, the authors use a custom convolutional neural network (CNN) model to predict the 6D pose of a single reflective object which is approximately orthogonal to the image plane.

For more sophisticated 6D pose estimation of objects [6] proposes the use of RANSAC [9] for object pose estimation and an adaptation of the vision transformer (ViT) model [10] to create semantic correspondences as an alternative to other point cloud alignment methods, such as iterative closest point (ICP) [11], between target and reference point cloud samples of objects. As another alternative, [7] employs an extensively studied use of 3D CAD models of objects, where object pose estimation is proposed from an RGB image by classifying the pose to one of the multiple camera viewpoints using Inception-v4 [12].

One of the ways how to estimate the planar direction of an object is by using principal component analysis (PCA) over the obtained edges of an object [13]. As pointed out in [14] such a technique suffers from having to manually choose the most appropriate parameters for edge detection, which is a necessary preprocessing step. If tuned incorrectly, the edges of an object can blur together with the surrounding environment, increasing the error of the planar projection angle. Additionally, some objects have a distinct feature representing the direction it points to, but PCA is not able to determine that.

An alternative solution would be the use of neural network models as experimented in [15]. They focus on the tilted bounding box estimation problem, estimating width and height for large objects from aerial imagery. For the neural network, they directly modify the convolutional layers of a YOLO [16] object detection model. Similarly, with [17], they also focus on tilted bounding box estimation, but use VGG-16 [18] and a modified region proposal network [19].

3. PROPOSED APPROACH

To address the challenge of bin picking, we propose a two-stage approach. In the first stage, a neural network is utilized to predict the unit vector of the object's long axis under a 2D rotation, its bounding box and class label only from an RGB image. In the second stage, assuming the surface of the object is approximately planar around the picking center point, the normal vector of the plane can be determined by subtracting the center point from the surrounding points and applying singular value decomposition to find the principal component axes. The full 6DoF pose (up to a rotational symmetry) can then be recovered by back-projecting the unit vector of the long axis into 3D space.

Fig. 1. shows point cloud data within a certain radius around the center of a bottle with its estimated orientation in 3D space.

The present study focuses only on the planar pose estimation part of the problem. The aim is to extend a pretrained object detection model with the capability to predict planar projection of poses for cylindrical and cuboid shape objects.

4. IMPLEMENTATION

Several pretrained object detector architectures like MaskRCNN [20], SOLO [21], DETR [22] and others exist which achieve results of around 40 mAP (mean average precision), which is a widely used metric to evaluate object detection models as defined in [23]. DETR is a state-of-the-art object detection model for computer vision, consisting of a CNN backbone, a transformer [24] and multi-layer perceptrons as final prediction heads. Its design readily lends itself to extension with additional inference heads and has demonstrated the performance of 42 mAP as stated in [22], granting a good ability to predict bounding boxes, which for our purposes makes it particularly useful for estimating the center of an object. For these reasons, DETR was selected as the basis for our implementation.

In the following subsections, we describe the structure of the dataset and the modifications made to DETR to tailor it for our purposes with the aim of improving 2D rotation estimation performance. From now on, we will refer to our adapted DETR model as DETR-POSE-2D.

4.1. Synthetically Generated Dataset

To facilitate the training and evaluation of DETR-POSE-2D, we have leveraged the capabilities of Blender, to generate synthetic images using the framework as described in [25]. By utilizing 3D models, we can obtain large datasets of diverse objects with high flexibility. Additionally, Blender enables the gen-

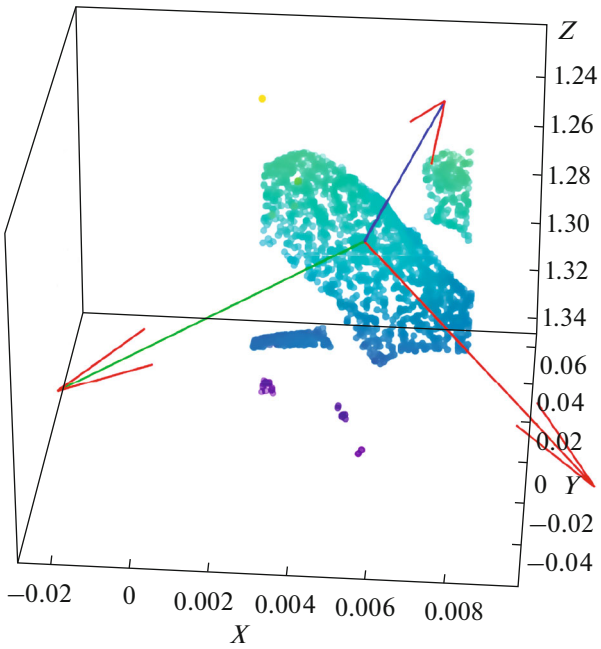

Fig. 1. Estimated orientation of a bottle in 3D space.

Fig. 2. Generated image from the evaluation dataset with 50 objects.

eration of object annotations including object labels, bounding boxes, 3D orientation and visibility. These annotations serve as ground truth information for our model.

In this study, we have selected two types of objects of cylindrical and cuboid shapes, namely, bottles and cans, representing two of the most common shapes encountered in everyday life. Each image contains a varying number of objects, specifically 5, 10, 20, 30 or 50 objects, with each set of the number of objects in an image being equal in size. All objects are randomly placed within a virtual box, as illustrated in Fig, 2. Although rare, some images consist solely of bottles or only of cans. Our generator outputs data in the COCO format for convenience, as DETR itself was trained on the COCO 2017 dataset [23].

It is important to note that a training dataset consisting solely of generated objects may not perform as well as a training dataset containing a percentage of real objects [25]. Nevertheless, given the scope of our current research, we have opted to defer the integration of real-world images containing objects to future iterations of our model, as it falls beyond the scope of our present investigation.

4.2. DETR-POSE-2D

To adapt the DETR model for the estimation of object rotation in a 2D plane, we incorporated additional components, namely, 2D orientation prediction head and visibility prediction head as MLPs (Multi-Layer Perceptrons), consisting of three linear layers and ReLU activation between them. The

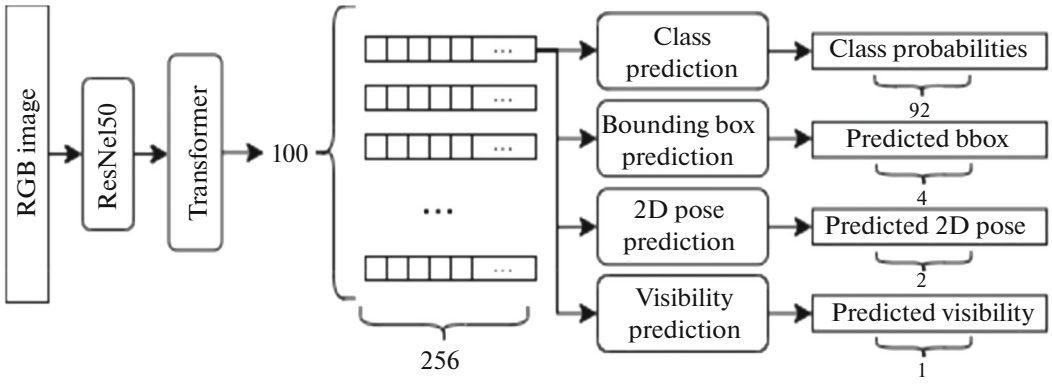

Fig. 3. DETR-POSE-2D architecture.

weights of a pretrained DETR model with ResNet50 as the CNN backbone available from [26] were used as a starting point, due to having the fastest inference time between all other available pretrained DETR models. The visibility head was included to distinguish highly occluded objects in the scene. By sorting all prediction outputs by visibility, the grasping robotic system can prioritize picking the least occluded objects at the top of a bin. The modified architecture is illustrated in Fig. 3, showing how each part of the 256-dimensional transformer output is processed by the prediction heads.

$$L_{\text{model}} = L_{\text{main}} + L_{\text{auxiliary}}, \tag{1}$$

$$L_{\text{main}} = L_{\text{class}} w_{l,\text{class}} + L_{\text{bbox}} w_{l,\text{bbox}} + L_{\text{giou}} w_{l,\text{giou}} + L_{\text{rot}} w_{l,\text{rot}} + L_{\text{vis}} w_{l,\text{vis}}, \tag{2}$$

$$L_{\text{auxiliary}} = \sum_{i=1}^{N-1} L_{\text{aux},i,\text{class}} w_{l,\text{class}} + L_{\text{aux},i,\text{bbox}} w_{l,\text{bbox}} + L_{\text{aux},i,\text{giou}} w_{l,\text{giou}} + L_{\text{aux},i,\text{rot}} w_{l,\text{rot}} + L_{\text{aux},i,\text{vis}} w_{l,\text{vis}}. \tag{3}$$

The final model loss L_{model} is given by the sum in Eq. (1). As shown in Eq. (2), L_{main} is calculated using the model output values from the last decoding layer of the transformer. As shown in Eq. (3), $L_{\text{auxiliary}}$ is computed using the model output values from each of the previous lower transformer decoding layers but using the same loss functions and weights that were used in Eq. (2). For example, $L_{\text{aux},i,\text{giou}}$ is the auxiliary loss of GIoU using the output of the decoding layer i.

To train the newly added rotation prediction and visibility prediction heads, we introduce two additional loss terms—L_{rot} for 2D rotation prediction loss and L_{vis} for visibility prediction loss—into Eq. (2) and, also, Eq. (3) with the corresponding auxiliary loss terms. 2D rotation loss L_{rot} and visibility prediction loss L_{vis} are element-wise Huber losses with $\delta = 1$.

The other loss terms in Eqs. (2) and (3) are classifier loss L_{class}, object detection loss L_{bbox} and generalized intersection over union loss L_{giou} as described in [22]. All loss terms are multiplied by their corresponding weight.

$$R = \begin{cases} Err_{\text{deg}}\left(\hat{P}, P\right) & \text{if} \quad Err_{\text{deg}}\left(\hat{P}, P\right) < Err_{\text{deg}}\left(\hat{P}, P_{\text{can flipped}}\right) \\ Err_{\text{deg}}\left(\hat{P}, P_{\text{can flipped}}\right) & \text{otherwise} \end{cases}, \tag{4}$$

$$P_{\text{can flipped}} = \begin{cases} [-dx, -dy] & \text{if the object class is can} \\ [dx, dy] & \text{otherwise} \end{cases}, \tag{5}$$

$$Err_{\text{deg}}\left(\hat{P}, P\right) = \arccos\left(S_{\text{C}}\left(\hat{P}, P\right)\right) \times \frac{180}{\pi}, \tag{6}$$

$$S_{\mathrm{C}}\left(\hat{P},P\right)=\frac{\sum_{i=1}^{n}\hat{P}_iP_i}{\sqrt{\sum_{i=1}^{n}\hat{P}_i^2\sum_{i=1}^{n}P_i^2}}. \tag{7}$$

For evaluation, model outputs were compared with ground truth labels up to symmetries, as stated in Eqs. (4) and (5). In the case of bottles, this was a rotation about the longitudinal axis. Cans were taken as symmetric about the *YZ* plane in their local coordinate system.

To enhance the performance of DETR-POSE-2D, the can symmetry is not only considered during evaluation but also taken into account when calculating L_{rot}. A copy of 2D rotations is created and all cans of this duplicate have their rotations flipped according to Eq. (5). In the end, the corresponding elements of both sets are compared and the one with the smallest angle error in degrees is chosen for the final evaluation or calculation of loss, as shown in Eq. (4). At first, angle error is calculated using Eq. (7), and then converted to degrees using Eq. (6).

$$F\left(I,\theta\right)=\hat{C},\hat{B},\hat{P},\hat{v}. \tag{8}$$

Equation (8) formally depicts all our model F inputs and outputs. For inputs, I is an RGB image and θ is the model parameters. For the outputs, $\hat{C}$ is a 100 × 92 matrix, which represents the output class probabilities (including for the "no object" class) in percent, $\hat{B}$ is a 100 × 4 matrix, which represents top-left point and bottom-right bounding box point pixel coordinates, $\hat{P}$ is a 100 × 2 matrix, which represents the predicted unit vectors of orientation in 2D, and $\hat{v}$ being a 100 × 1 vector, which represents the predicted visibility for each of the 100 detections in percent.

$$\hat{\sigma}=\underset{\sigma\in S^N}{\arg\min}\left(\sum_{i=1}^{N}\left[-\hat{C}_{\sigma_i,c_i}w_{c,\mathrm{class}}+d_M\left(\hat{B}_{\sigma_i},B_i\right)w_{c,\mathrm{bbox}}-GIoU\left(\hat{B}_{\sigma_i},B_i\right)w_{c,\ \mathrm{giou}}\right]\right). \tag{9}$$

The model outputs are associated with corresponding object labels using the Hungarian matcher algorithm [27], which minimizes the target expression in Eq. (9). $\hat{\sigma}$ represents permutation that minimizes the cost function, σ is a one-to-one mapping of prediction to ground truth and S^N is the set of all possible permutations of possible detections to the actual object in the image.

In Eq. (9) association-matching terms $\hat{C}_{\sigma_i,c_i}$ is the c_ith probability of the σ_i vector from $\hat{C}$ matrix, $d_M\left(\hat{B}_{\sigma_i},B_i\right)$ is the Manhattan distance algorithm between the predicted and the ground truth bounding box and $GIoU\left(\hat{B}_{\sigma_i},B_i\right)$ is the GIoU algorithm. $w_{c,\mathrm{class}}$, $w_{c,\mathrm{bbox}}$, and $w_{c,\mathrm{giou}}$ are their corresponding weights.

$$d_{\mathrm{E}}\left(\hat{P}_{\sigma_i},P_i\right)w_{c,\mathrm{rot}}, \tag{10}$$

$$d_{\mathrm{M}}\left(\hat{P}_{\sigma_i},P_i\right)w_{c,\mathrm{rot}}, \tag{11}$$

$$d_{\mathrm{E}}\left(\hat{v}_{\sigma_i},v_i\right)w_{c,\mathrm{vis}}. \tag{12}$$

Equations (10)–(12) show expressions to be added to the cost function of Eq. (9), during experiments, where Eq. (10) shows the Euclidean distance between predicted and ground truth unit vectors of 2D rotation, Eq. (11) shows Manhattan distance between predicted and ground truth unit vectors of 2D rotation and Eq. (12) shows the Euclidean distance between predicted and ground truth visibilities. $w_{c,\mathrm{rot}}$ and $w_{c,\mathrm{vis}}$ are their corresponding weights.

5. EXPERIMENTAL SETUP

DETR-POSE-2D was trained and evaluated on one A100 GPU for 20 epochs and a batch size of 16 images. The dataset comprises 9500 RGB images for training and 500 RGB images for evaluation. The average training time was approximately one and a half hours on the whole training set and the average inference time of one image from the evaluation set is roughly 0.02 seconds. Generated training and evaluation datasets were shuffled at the input of the model.

Fig. 4. Our available grasping system.

5.1. Evaluation Metrics

To evaluate the performance of the model, we compute the average error for each type of output, namely, class prediction, center prediction, rotation prediction and visibility prediction, over the evaluation dataset.

$$Avg\left(\hat{C}_{\hat{\sigma}}, C\right) = \frac{1}{N}\sum_{i=i}^{N}\hat{C}_{\hat{\sigma},i} - C_i, \tag{13}$$

$$Avg\left(\hat{v}_{\hat{\sigma}}, v\right) = \frac{1}{N}\sum_{i=i}^{N}\hat{v}_{\hat{\sigma},i} - v_i. \tag{14}$$

The average class error is given in Eq. (13), where $\hat{C}_{\hat{\sigma}}$ is a matrix of predicted classes matched and ordered to the corresponding ground truth C. The average center error is computed as the average Euclidean distance in pixels between the predicted and ground truth values. The average visibility error is given in Eq. (14), where $\hat{v}_{\hat{\sigma}}$ is a matrix of predicted visibility matched and ordered to the corresponding ground truth v. The average angle error is calculated using cosine similarity shown in Eq. (7), where $\hat{P}$ is the predicted and P is the ground truth unit vector of 2D rotation. For visualization purposes, output of Eq. (7) is expressed in degrees, as shown in Eq. (6).

5.2. Scenarios

Experiments consist of five scenarios, but the only difference between them is the target expression used in Hungarian matcher. Namely, the base implementation is given by Eq. (9), base implementation with added Eq. (10) and $w_{c,\text{rot}} = 1$, base implementation with added Eq. (11) and $w_{c,\text{rot}} = 1$, base implementation with added Eqs. (10) and (12) and $w_{c,\text{rot}} = 1$ and $w_{c,\text{vis}} = 1$ and, lastly, base implementation with added Eq. (10) and $w_{c,\text{rot}} = 5$.

5.3. Robotic System

At our disposal, we have a grasping robotic system to empirically test bin-picking capabilities, shown in Fig. 4.The perception system consists of an RGBD Zivid One M camera pointing down at some angle to the black box filled with bottles and cans.

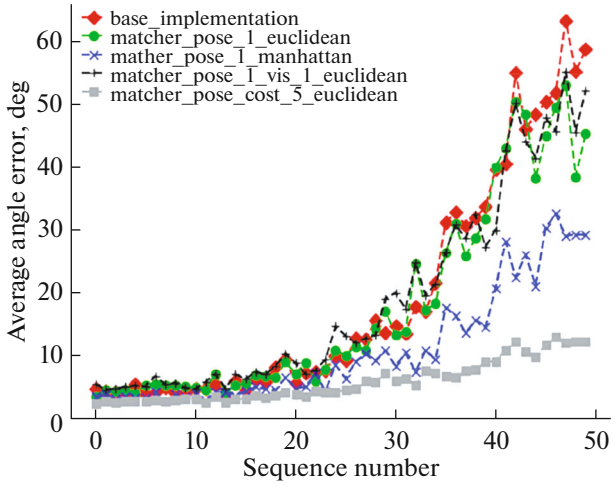

Fig. 5. Average angle error in degrees w.r.t the most visible object to the least visible object in an image according to true visibility.

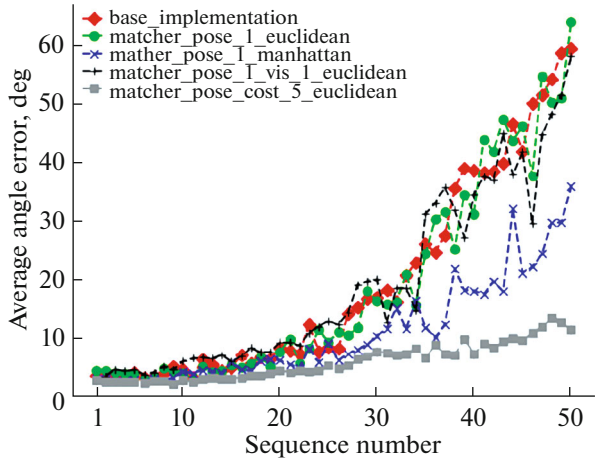

Fig. 6. Average angle error in degrees w.r.t the most visible object to the least visible object in an image according to predicted visibility.

6. RESULTS

Figure 5 displays the average angle error in degrees of all objects in the evaluation dataset in relation to their order of detection sorted by the ground truth visibility.

Figure 6 shows a similar graph but order of detection is sorted by the predicted visibility. By comparing both graphs we will be able to assess what angle error to expect when a robot will grasp the most visible objects by prediction. Each mentioned figure includes one plot of each scenario described in the Experimental Setup section.

Figure 7 depicts DETR-POSE-2D output visualization, showing the predicted label, classification score, bounding box, visibility score and the pointing direction from the center of the bounding box of each detected object with a threshold of 0.99 for the classification score and 0.99 for the visibility score. It is worth noting that the predicted visibility score is not normalized, making the score go over the range of [0; 1].

Table 1 additionally shows average error values for other outputs of our model, namely, center prediction error, class prediction error and visibility prediction error. By looking at this table, it is shown how one change can improve one type of output of the model but might worsen a different one.

Fig. 7. DETR-POSE-2D output visualization with a class confidence threshold of 0.99 and visibility score threshold of 0.99.

7. CONCLUSIONS

In this study, the aim was to demonstrate the feasibility of extending an existing object detection model to obtain planar pose estimation. Our approach differs from the research mentioned in related works by:

(1) Instead of detecting tilted bounding boxes or the pose of an object in 3D using point clouds or multiple cameras, we focus on planar projection estimation directly from a single RGB image.

(2) Inserting new fully connected prediction heads after an unchanged DETR CNN-transformer backbone, without any addition of preprocessing and postprocessing steps resulting in a significantly differing practical implementation.

The pretrained model DETR was extended with a 2D rotation prediction and visibility prediction head, and the final loss function was modified accordingly with the 2D prediction loss and visibility loss. Additionally, corresponding expressions were added to calculate the difference between the ground truth and predicted values of 2D rotation and visibility to improve the matching inside of the Hungarian matcher.

Table 1 shows that similarly high results were obtained for the least occluded object, regardless of the matcher target expression. The use of Manhattan distance provided better results than the use of Euclidean distance. Similarly, comparing scenarios of $w_{c,\mathrm{rot}} = 1$ and $w_{c,\mathrm{rot}} = 5$, increasing the weight of the pose error term in the matcher objective function improved the accuracy of the corresponding output.

As shown in Table 1, there exists a trade-off between classification and 2D rotation prediction accuracy. Future work could be directed at solving this issue, perhaps by fine-tuning the weights in the matcher target expression or using different model architectures.

By taking the average angle error of both bottles and cans of scenario 5 from Table 1 this research demonstrates that augmenting an existing pretrained model can achieve an overall average angle error of 3.23 deg.

Table 1. Average (avg) error values for all types of output from DETR-POSE-2D depending on the scenario

Scenario	Avg angle error, deg		Avg error, pixels	Avg error, %	
	for bottles	for cans	centre	class	visibility
1.	7.56	7.73	0.06	7.18	0.04
2.	6.64	8.43	0.06	7.90	0.05
3.	4.14	5.74	0.07	8.52	0.04
4.	7.13	9.85	0.06	11.97	0.04
5.	2.23	4.23	0.13	9.67	0.05

FUNDING

This research was partially supported by Latvian state research program no. VPP-EM-FOTONIKA-2022/1-0001 "Viedo materiālu, fotonikas, tehnoloģiju un inženierijas ekosistēma."

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. Di Pasquale, V., Franciosi, C., Iannone, R., and Miranda, S., Special issue: Smart manufacturing for sustainability: Trends and research challenges, *J. Ind. Eng. Manage.*, 2022, vol. 15, no. 1, pp. 863–865. https://doi.org/10.3926/jiem.3864
2. Lee, Y., Kumaraguru, S., Jain, S., Robinson, S., Helu, M., Hatim, Q., Rachuri, S., Dornfeld, D., Saldana, C., and Kumara, S., A classification scheme for smart manufacturing systems' performance metrics, *Smart Sustainable Manuf. Syst.*, 2017, vol. 1, no. 1, p. 20160012. https://doi.org/10.1520/ssms20160012
3. Torres, P., Arents, J., Marques, H., and Marques, P., Bin-picking solution for randomly placed automotive connectors based on machine learning techniques, *Electronics*, 2022, vol. 11, no. 3, p. 476. https://doi.org/10.3390/electronics11030476
4. Lee, S. and Lee, Ye., Real-time industrial bin-picking with a hybrid deep learning-engineering approach, *2020 IEEE Int. Conf. on Big Data and Smart Computing (BigComp)*, Busan, Korea (South), 2020, IEEE, 2020. https://doi.org/10.1109/bigcomp48618.2020.00015
5. Janis, A. and Greitans, M., Smart industrial robot control trends, challenges and opportunities within manufacturing, *Appl. Sci.*, 2022, vol. 12, no. 2, p. 937. https://doi.org/10.3390/app12020937
6. Goodwin, W., Vaze, S., Havoutis, I., and Posner, I., Zero-shot category-level object pose estimation, *Computer Vision—ECCV 2022*, Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., and Hassner, T., Eds., Lecture Notes in Computer Science, vol. 13699, Cham: Springer, 2022, pp. 516–532. https://doi.org/10.1007/978-3-031-19842-7_30
7. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., and Navab, N., SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again, *2017 IEEE Int. Conf. on Computer Vision (ICCV)*, Venice, 2017, IEEE, 2017, pp. 1530–1538. https://doi.org/10.1109/iccv.2017.169
8. Kozák, V., Sushkov, R., Kulich, M., and Přeučil, L., Data-driven object pose estimation in a practical bin-picking application, *Sensors*, 2021, vol. 21, no. 18, p. 6093. https://doi.org/10.3390/s21186093
9. Fischler, M.A. and Bolles, R.C., Random sample consensus, *Commun. ACM*, 1981, vol. 24, no. 6, pp. 381–395. https://doi.org/10.1145/358669.358692
10. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N., An image is worth 16x16 words: Transformers for image recognition at scale, 2020. https://doi.org/10.48550/arXiv.2010.11929
11. Zhang, J., Yao, Yu., and Deng, B., Fast and robust iterative closest point, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021, vol. 44, no. 7, pp. 1–1. https://doi.org/10.1109/tpami.2021.3054619
12. Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A., Inception-v4, Inception-ResNet and the impact of residual connections on learning, *Proc. AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1. https://doi.org/10.1609/aaai.v31i1.11231
13. Wei, Yi. and Marshall, S., Principal component analysis in application to object orientation, *Geo-Spatial Inf. Sci.*, 2000, vol. 3, no. 3, pp. 76–78. https://doi.org/10.1007/bf02826615
14. De Silva, A., Object detection and correction using computer vision, The Repository at St. Cloud State, 2020. https://repository.stcloudstate.edu/cgi/viewcontent.cg=i?article=1040&context=csit_etds. Cited May 3, 2023.
15. Zhang, H. and Liu, J., Direction estimation of aerial image object based on neural network, *Remote Sensing*, 2022, vol. 14, no. 15, p. 3523. https://doi.org/10.3390/rs14153523
16. Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., You only look once: Unified, real-time object detection, *2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 2016, IEEE, 2016, pp. 779–788. https://doi.org/10.1109/cvpr.2016.91

17. Chen, Yo., Gong, W., Chen, C., and Li, W., Learning orientation-estimation convolutional neural network for building detection in optical remote sensing image, *2018 Digital Image Computing: Techniques and Applications (DICTA)*, Canberra, 2018, IEEE, 2018, pp. 1–8. https://doi.org/10.1109/dicta.2018.8615859
18. Simonyan, K. and Zisserman, A., Very deep convolutional networks for large-scale image recognition, 2014. https://doi.org/10.48550/arXiv.1409.1556
19. Ren, Sh., He, K., Girshick, R., and Sun, J., Faster R-CNN: Towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015, vol. 39, no. 6, pp. 1137–1149. https://doi.org/10.1109/tpami.2016.2577031
20. He, K., Gkioxari, G., Dollár, P., and Girshick, R., Mask R-CNN, *2017 IEEE Int. Conf. on Computer Vision (ICCV)*, Venice, 2017, IEEE, 2017, pp. 2980–2988. https://doi.org/10.1109/iccv.2017.322
21. Wang, X., Kong, T., Shen, C., Jiang, Y., and Li, L., SOLO: Segmenting objects by locations, *Computer Vision—ECCV 2020*, Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.M., Eds., Lecture Notes in Computer Science, vol. 12363, Cham: Springer, 2020, pp. 649–665. https://doi.org/10.1007/978-3-030-58523-5_38
22. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S., End-to-end object detection with transformers, *Computer Vision—ECCV 2020*, Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.M., Eds., Lecture Notes in Computer Science, vol. 12363, Cham: Springer, 2020, pp. 213–229. https://doi.org/10.1007/978-3-030-58452-8_13
23. Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C., Microsoft COCO: Common objects in context, *Computer Vision—ECCV 2014*, Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., Eds., Lecture Notes in Computer Science, vol. 8693, Cham: Springer, 2023, pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48
24. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I., Attention is all you need, *Adv. Neural Inf. Process. Syst.*, 2017, vol. 30.
25. Arents, J., Lesser, B., Bizuns, A., Kadikis, R., Buls, E., and Greitans, M., Synthetic data of randomly piled, similar objects for deep learning-based object detection, *Image Analysis and Processing—ICIAP 2022*, Sclaroff, S., Distante, C., Leo, M., Farinella, G.M., and Tombari, F., Eds., Lecture Notes in Computer Science, vol. 13232, Cham: Springer, 2022, pp. 706–717. https://doi.org/10.1007/978-3-031-06430-2_59
26. facebookresearch, 2020. Detection Transformer (DETR) (v0.2): Pretrained model, GitHub. https://dl.fbaipublicfiles.com/detr/detr-r50-e632da11.pth.
27. Kuhn, H., The Hungarian method for the assignment problem, *Naval Res. Logist. Q.*, 1955, vol. 2, nos. 1–2, pp. 83–97. https://doi.org/10.1002/nav.3800020109