Computers in Industry 161 (2024) 104130

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.sciencedirect.com/journal/computers-in-industry

Addressing challenges in industrial pick and place: A deep learning-based 6 Degrees-of-Freedom pose estimation solution

Elena Govi [a,b,*], Davide Sapienza [a], Samuele Toscani [a], Ivan Cotti [a], Giorgia Franchini [a], Marko Bertogna [c,a]

[a] Department of Physics, Mathematics and Computer Science, University of Modena and Reggio Emilia, 41125, Modena, Italy
[b] Department of Mathematics, Physics and Computer Science, University of Parma, 43124, Parma, Italy
[c] Hipert srl, 41122 Modena, Italy

ARTICLE INFO

Keywords:
Artificial intelligence in industrial automation
Learning from synthetic data
Computer vision for pose estimation
Convolutional Neural Network
Machine learning in real-world applications
Industry 4.0

ABSTRACT

Object picking is a fundamental, long-lasting, and yet unsolved problem in industrial applications. To complete it, 6 Degrees-of-Freedom pose estimation can be crucial. This task, easy for humans, is a challenge for machines as it involves multiple intelligent processes (for example object detection, recognition, pose prediction). Pose estimation has recently made huge steps forward, due to the advent of Deep Learning. However, in real-world applications it is not trivial to compute it: each use-case needs an annotated dataset and a model robust enough to face its specific challenges. In this paper, we present a comprehensive investigation focused on a specific use-case: the picking of four industrial objects by a collaborative robot's arm, addressing challenges related to reflective textures and pose ambiguities of heterogeneous shapes. Thus, Artificial Intelligence is crucial in this process, utilizing Convolutional Neural Networks to discern an object's pose by extracting hierarchical features from a single image. In detail, we propose a new synthetic dataset of industrial objects and a fine-tuning method to close the sim-to-real domain gap. In addition, we improved an existing pipeline for pose estimation and introduced a new version of an existing method, based on Convolutional Neural Networks. Finally, extensive experiments were conducted with a Universal Robot UR5e. Results show our strategy achieves good performances with an average successful picking rate of 75% on these new objects. Considering the lack of available datasets for pose estimation, coupled with the significant time and labor required for annotating new images, we contribute to the scientific community by providing a comprehensive dataset, and the associated generation and estimation pipelines.[1]

1. Introduction

In recent years, the field of robotics has witnessed remarkable advancements, with intelligent machines progressively permeating various industrial domains. In Industry 4.0, a novel type of robot has been introduced: Collaborative Robots (Cobots). The idea of collaborative robots emerged in the mid-1990s, and today, they are widely adopted and continually evolving. Cobots are not required to be confined in a safety cage far away from human workers, they are designed to work alongside humans in a shared workspace. Among the myriad applications, robotic grasping is pivotal in automating industrial processes with cobots, facilitating increased productivity and precision. In computer vision related challenges, the quest for successful learning-based grasping solutions in complex industrial scenarios remains formidable. It takes two main directions: model-based and model-free methods (Kleeberger et al., 2020). In the first, the 3D Computer-Aided Design (CAD) model or the category of the object is known, in the second only a set of images is used. The industrial scenario usually requires the best possible precision with well-known recurrent objects. We focused on a model-based grasping pipeline, where a set of grasps is predefined and the robotic arm works according to the 6 Degrees-of-Freedom (DoF) pose estimation. It consists of, given an image as input, predicting the object position (rotation and translation). 6D pose achieved remarkable results on benchmarks largely due to advancements in artificial intelligence for computer vision. However, in real-world scenarios, the task is still a challenge, because the need for a large annotated dataset often limits practical applications. In addition, in robotics, we could have different environments, for example, light changes and cluttered backgrounds, which could affect learning. As a

* Corresponding author at: Department of Physics, Mathematics and Computer Science, University of Modena and Reggio Emilia, 41125, Modena, Italy.
E-mail address: elena.govi@unimore.it (E. Govi).

[1] https://github.com/EleGo9/6Dpose-Yolov7-Seg-A.

https://doi.org/10.1016/j.compind.2024.104130
Received 1 March 2024; Received in revised form 21 May 2024; Accepted 16 July 2024
Available online 25 July 2024

consequence, deep learning methods must be robust enough to face changing environments, and this is not trivial. For this reason, each use case robotic application must tackle different challenges. In this paper, we will face some of the monocular 6D pose challenges highlighted by Thalhammer et al. (2023), proposing a possible solution in a specific real-world industrial application. In detail, we have four industrial objects with different shapes and materials and we want the cobot to successfully grasp them by predicting the 6D pose estimation. Then, from an image I, the 6D pose estimation is defined by x, y, and z as axis coordinates and ϕ, ψ and θ as corresponding Euler angles. Input images I give rise to a main distinction in 6D pose estimators since they can be: (i) Red Green Blue (RGB) monocular images and (ii) RGB-Depth (RGB-D) images (He et al., 2021, 2020; Cao et al., 2023). The RGB monocular image is an array with shape $(H, W, 3)$ where 3 is the number of channels representing the three colors RGB and W and H are respectively width and height. In the second RGB-D case, additional information is required: depth data. Though RGB-D pose estimation methods achieved the best performances on benchmark datasets, we do not use depth maps in our learning step, only in post-processing. Even if the depth map has proven to be a great geometric source for pose estimation, it still presents many limitations (Marullo et al., 2023; Thalhammer et al., 2023): (i) it requires expensive hardware if compared to RGB images; (ii) it has different noise patterns at different distances or image regions; (iii) it is strongly dependent on sensor types since it varies noise patterns between sensors.

On the other hand, RGB sensors are low cost, low noise, and are less sensitive to different parameters (such as the scenario, the objects' texture or the camera's intrinsic matrix).

As a consequence, RGB approaches gained significant relevance and some methods (Hodan et al., 2020; Sundermeyer et al., 2020b) used RGB input for the training phase, taking into account depth only for the post-processing refinement. Driven by these motivations, our primary objective was to thoroughly investigate the capabilities of the easily accessible and stable RGB modality in the context of 6D robotic grasping tasks, all without relying on any depth input in the learning phase. After that, we used depth in the final picking point computation. A further categorization of 6D pose methods relies on instance-level and category-level 6D pose estimation. In the first, instance-level 6D pose estimation, inputs are an image and the exact 3D model of the object of which we want the pose. In the category-level 6D pose estimation task, the specific 3D model is not required. Instead, the provided 3D models are not the same as represented in the image, but they belong to the same category. The goal is more challenging than instance-level: for example, given the model of a cup, the purpose is to predict the pose from an image of every existing cup. This task was introduced by Wang et al. (2019) in 2019, so it is new and less navigated than the instance-level one. For these reasons, given the industrial scenario where achieving the highest possible performance on objects is imperative and 3D object models are known a priori, we opted for instance-level methods. Focusing on this task, the main ongoing research challenges we met were: domain shift, symmetry handling, and challenging material properties, also cited by Thalhammer et al. (2023). The purpose of this paper is to solve the task of 6D pose estimation for a successful picking in a challenging environment, where four industrial objects are chosen. The entire procedure is presented in the paper, from data generation to real-world experiments. Our main contributions, categorized in Fig. 1, are the following:

1. We proposed a new pipeline for RGB-input 6D pose estimation on challenging objects in a semi-cluttered industrial scenario, improving an existing method by modifying its assumption on noise impedance (Section 3.2.2);
2. We provide a new simulated semi-cluttered dataset (Section 3.1) with single instance objects, and a customizable data-generation pipeline. To bridge the performance gap of a segmentation model between simulation and real scenario, we introduced a new two-step training strategy:: (i) training on simulated data, (ii) fine-tuning on a very small dataset with both simulated and real, augmented images;
3. We evaluated the entire pipeline with a robotic arm and a camera with extensive picking experiments (Section 4.3);
4. We provide the dataset and code, which are publicly available, serving as valuable resources for similar applications in Industry 4.0.

This paper focuses on estimating 6D pose at the instance level using RGB input, with a particular emphasis on implicit representations of pose. In Section 2, we will explore various commonly utilized strategies. Section 3 introduces the challenges of industrial scenarios and the proposed pipeline to address them. Section 4 details the experiments, including their settings, metrics, and results, which are further discussed in Section 4.4. Finally, Section 5 presents the conclusions and suggestions for future work.

2. Related works

Early approaches. Initially, the 6D pose estimation task was solved by geometric feature extractors and template-matching techniques (Hinterstoisser et al., 2011; Huttenlocher et al., 1993). However, these methods lacked robustness against changes in lighting and background conditions.

Deep learning approaches. In recent years, Convolutional Neural Networks (CNNs) led to a breakthrough in this field, overcoming other strategies. Nowadays several CNN-based algorithms have been proposed. PoseNet (Kendall et al., 2015) was the first method based on CNNs. It consists of a CNN-trained end-to-end to regress the camera's orientation and position.

Subsequent methods, such as PoseCNN (Xiang et al., 2017), directly regress the pose through a two-stage process. These regression-based methods are efficient and fast but may struggle with capturing spatial relationships and geometric constraints.

On the contrary, the other two types (classification and 2D–3D correspondences) can provide a more generic scheme of implementation. Despite this, end-to-end regression networks remain a focal point of ongoing research. For example, a recent work (Lin et al., 2022), directly regresses the pose while concurrently capitalizing on geometric constraints through the prediction of keypoint offsets. Another method, proposed by Tekin et al. (2018), follows the architecture of YOLOv2 (Redmon et al., 2016), a fully convolutional network typically used for 2D object detection.

The method proposed in Tekin et al. (2018) approaches the 6D pose estimation problem by predicting 2D image coordinates of virtual 3D control points associated with the 3D models of the objects of interest. It selects 9 control points for each object, including the 8 corners of the tight 3D bounding box and the centroid of the 3D model. The network is trained to predict both precise 2D locations and high-confidence values in regions where the object is present. PVNET (Peng et al., 2019) adopts a different approach; rather than regressing image coordinates of keypoints, it forecasts unit vectors representing directions from each pixel to the keypoints. Each pixel's direction is associated with the keypoint they voted for. Then the most voted keypoints are computed by random sample consensus (RANSAC) algorithm (Fischler and Bolles, 1981). This voting scheme creates a vector-field representation for keypoint localization, enforcing spatial relations between object parts, and even inferring the location of invisible parts. Although such a paradigm offers reasonably accurate estimates, it has limitations. First, two sets of correspondences could have the same average error, describing entirely different poses. Secondly, these approaches are not differentiable with respect to the 6D pose estimation parameters, thereby learning capabilities are restricted. Geometry-Guided Direct Regression (GDR) (Wang et al., 2021) is a hybrid between them, trying to regress dense 2D–3D

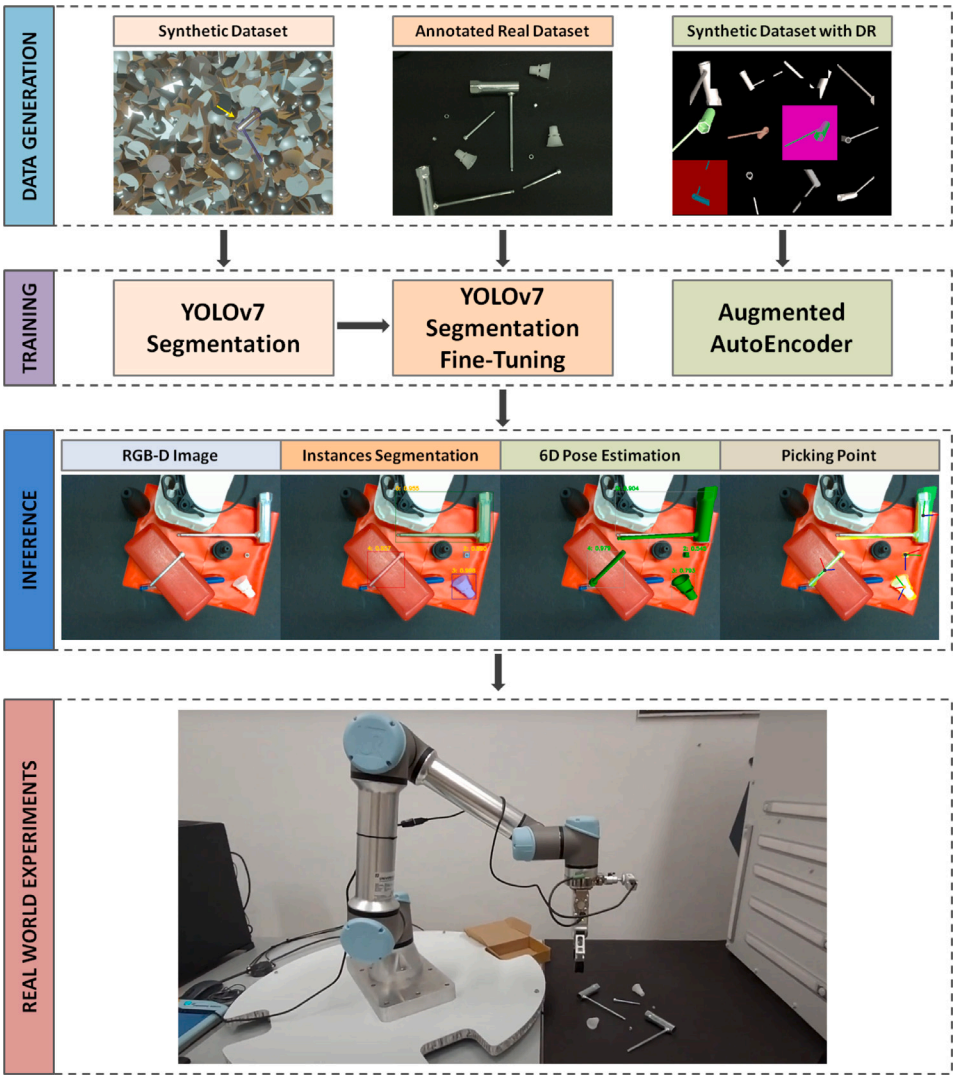


Fig. 1. Industrial use case: main contributions on data generation, pipeline, and picking experiments.

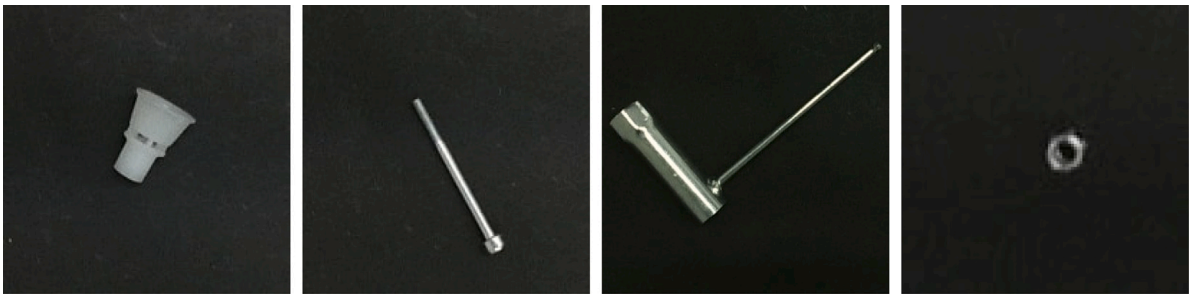
correspondence maps and to learn the Perspective-n-Point (PnP) optimization by convolutions. Another different approach (Su et al., 2022) is based on defining the matching of dense 2D–3D correspondence as a hierarchical classification task. It is an RGB two-stage method where a unique descriptor is assigned to the 3D vertex, then the dense correspondence between pixels and vertices is predicted. However, handling pose ambiguities remains a challenge.

Augmented autoencoder. Our application brings back template-matching methods, to overcome this issue. This approach can appear as a step behind, however, we did not use standard template matching techniques, but a revised version of the Augmented Autoencoder (AAE) (Sundermeyer et al., 2020b).

We relied on CNN-based methods for extracting a meaningful latent space for object pose. First, the continuous space of rotation Special Orthogonal 3-dimensional Group (SO(3)) has been discretized as in Kehl et al. (2017). Unfortunately, moving to a classification network, even rather coarse intervals of 5 degrees lead to over 50.000 possible classes, dramatically increasing the complexity of the classification problem and hindering convergence. Another possible option to face regression and classification issues is to (i) implicitly describe the image with the pose and, successively, (ii) compute similarities between the latent representation and a codebook describing a discretized version of SO(3). The implicit descriptor has the structure of an encoder trained in a denoising Autoencoder (AE) and was proposed by Sundermeyer et al. (2020b,a). We chose this method according to our specific constraints:

- AAE bridge the domain gap through the Domain Randomization phase, as claimed by the authors in Sundermeyer et al. (2020b);
- AAE learns invariant features to solve the ambiguities and symmetries, thanks to the fact that it is trained on the reconstruction loss function, as highlighted by the authors (Sundermeyer et al., 2020b);
- AAE does not use a depth map during inference. Therefore, depth noise does not affect the learning. In the original paper, the Iterative Closest Point algorithm is proposed as a refinement step, using depth maps, but it is not considered in this work.

The risk of taking into account the latent space of a denoising AE is that we do not know exactly what it represents. This is a huge limitation and to solve it in the original paper (Sundermeyer et al., 2020b), a strong hypothesis is assumed and extended also to data augmentation: *the Denoising AE produces latent representations which are invariant to noise because it facilitates the reconstruction of de-noised images* called Hypothesis 1. In this paper, we want to discuss this hypothesis and its possible extensions. In addition, we present experiments that explore how different pipelines affect the latent space and, consequently, 6D pose prediction. Moreover, we think that, given the difficulty of latent



(a) Nozzle. (b) Bolt. (c) Spark plug key. (d) Nut.

Fig. 2. Four challenging industrial objects.

space interpretation, the AE was not further explored but the representation of a pose through a latent encoding is a compelling starting point.

3. Methods and applications

3.1. Industrial scenario challenges

Since objects of our use-case do not match with existing 6D dataset (Hinterstoisser et al., 2012; Hodan et al., 2017; Xiang et al., 2017; Kaskman et al., 2019; Tyree et al., 2022), we create our custom one.

In this case, we have $N = 4$ industrial objects: a spark plug key, a nut, a nozzle, and a bolt (Fig. 2). We choose them because they are typical industrial objects, commissioned by a company, and simultaneously present numerous critical aspects. First, we faced the problem of the dataset: 6D pose real datasets are limited by the high cost in terms of time and labor for annotating. While large annotated datasets are essential for training models with millions of parameters, the process of translation and rotation labeling is notably expensive. This issue is clearly more relevant in 6D pose estimation than in other vision tasks. For this reason, Physically-based Rendering (PBR) for automated dataset generation is increasing: synthetic data can be easily generated with low cost and high efficiency thanks to modern simulators. A lot of progress has been made with respect to monocular object pose estimation (Thalhammer et al., 2021; Nguyen et al., 2022; Wang et al., 2020; Hu et al., 2022; Lu et al., 2022). However, the methods that rely only on PBR as input, experimented with lower performances than methods trained on real data. The problem of domain shift remains one of the major challenges of general robotics, with particular effort in the task of 6D pose estimation. For these reasons, we first created labeled synthetic images using CAD models of the objects, generated with UnityEngine. Secondly, we propose some solutions to bridge the domain gap between reality and simulation. The generation code is available.[2]

New challenging issues we faced are:

1. Semi-cluttered scenario: we look for a pipeline able to distinguish an object placed on other similar background objects;
2. Reflecting textures: most of the objects used in 6D pose estimation benchmarks are opaque. However, especially for industrial applications, handling metallic materials is a crucial point. They are challenging because they reflect light and their appearance is strongly dependent on the light and camera position in the environment. Some of our objects (spark plug key, nut, and bolt) have a metallic surface, and the light changes given by this material could affect the learning of the algorithm. We need a method robust to these effects.
3. Symmetries and self-occlusions: ambiguities arise when an object has the same visual appearance for different poses. Template matching methods do not suffer from ambiguities, because they do not require learning a representation or regressing a pose. On the contrary, deep learning approaches are widely affected by symmetries. Each of our four considered objects present pose ambiguities given by object symmetries or self-occlusion induced symmetries. This means that identical training images could have different rotation labels assigned which disturbs the learning process. We looked for a pose ambiguity-invariant method.

To solve the issue 1, a semi-cluttered geometric background dataset has been created with UnityEngine. The geometric background is composed of different 3D geometric objects (cubes, spheres and cylinders) of different metallic/matt materials, placed randomly on each image. The background can be considered semi-cluttered, because the geometric solids are only in the background, and always behind the main object, which is never occluded or partially occluded.

In addition, light directions and light colors change randomly too, creating different metallic effects. This helps solve issue 2. There are no identical images. Each image contains one of the four target objects placed in a random translation and rotation.

Some examples are provided in Fig. 3. Target objects are segmented and indicated with an arrow in these example images.

The issue 3 is related to the method choice and, therefore is further explained in the following section, with the pipeline explanation.

3.2. Pipeline

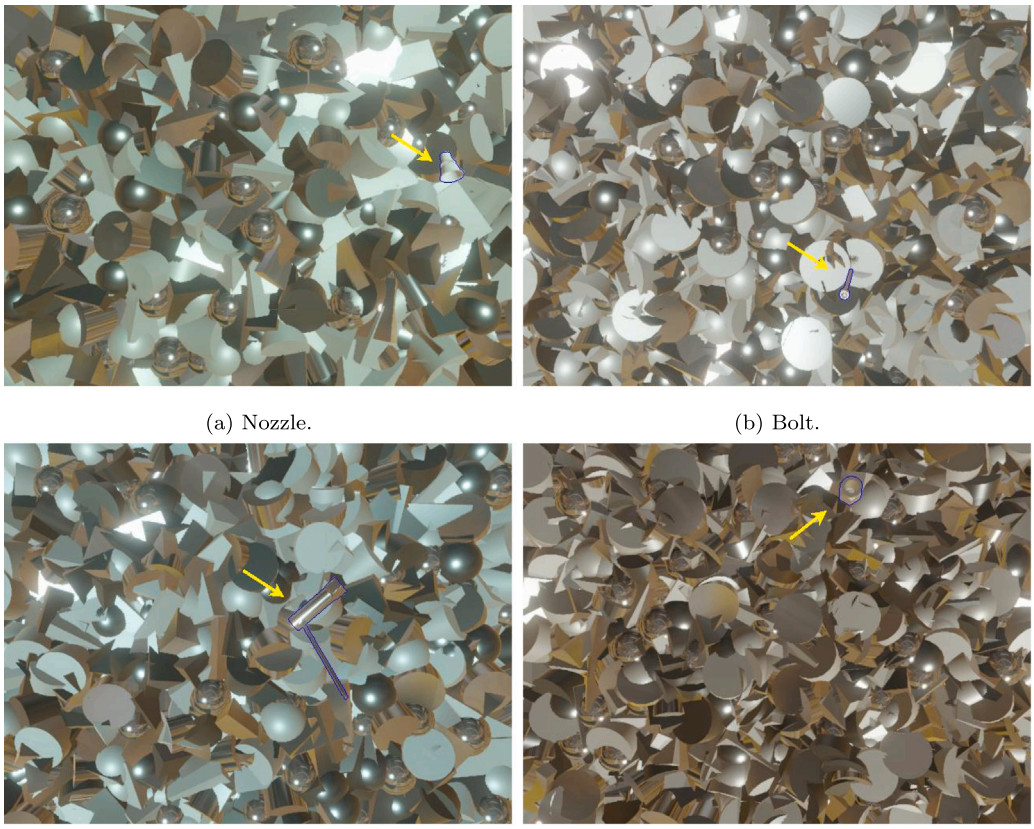
According to considerations done in previous sections, the 6D pose estimation problem should be formalized as follows (Hodan et al., 2018): there are N objects $\mathcal{O} = \{\mathcal{O}_i : i = 1, \dots, N\}$ with their corresponding 3D CAD models $\mathcal{M} = \{\mathcal{M}_i : i = 1, \dots, N\}$; given an RGB image I we want to estimate the pose $\mathbf{P}$ of each object O_i in the scene I. The pose is defined by a 4×4 matrix composed by the rotation 3×3 matrix $\mathbf{R}$ and the translation 3×1 vector $\mathbf{t}$, obtaining in this way:

$$\mathbf{P} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

The choice of AAE (Sundermeyer et al., 2020b), explained in Section 3.2.2, solves problems of reflecting textures and symmetries: data augmentation during the training phase allows AAE to generalize with different light effects and the method is invariant with respect to pose-ambiguities, as explained by the authors. However, the original "2D detection + AAE" pipeline presents several difficulties with thin, elongated objects, as shown in Section 4.

For this reason, we propose a new pipeline that addresses this issue by employing segmentation rather than relying on 2D detection, as shown in Fig. 4. For each object, three sets of images have been generated with geometric backgrounds: 3500, 750, and 500 images

[2] https://github.com/EleGo9/Gen4Industry



(a) Nozzle. (b) Bolt.

(c) Spark plug key. (d) Nut.

Fig. 3. Our objects represented in our geometric background for the simulated dataset.

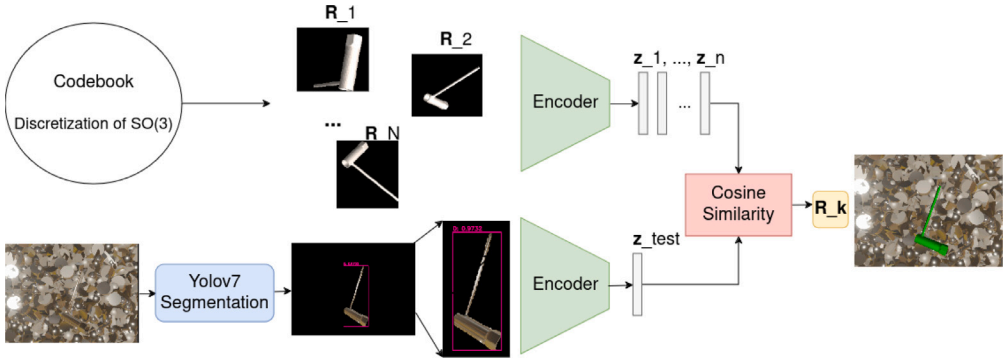


Fig. 4. Our modified pipeline.

respectively for train, validation, and test set. This results in a total of 4750 images per object and a dataset comprising 19 000 images. The geometric dataset is used for the segmentation training phase, while 6D pose method generates its own dataset during training, with a domain randomization strategy. After training phases, the geometric dataset is also used for ablation study and validation of the entire pipeline.

3.2.1. YOLOv7-segmentation

The original pipeline proposed by Sundermeyer et al. (2020b) consists of two stages: 2D detection with Single-Shot Detector (SSD) (Liu et al., 2016) and RetinaNet (Lin et al., 2017). For the segmentation phase, we trained the YOLOv7-segmentation (Wang et al., 2022) on our custom data, as shown in Fig. 5. We used weights trained on Common Objects in Context COCO (Lin et al., 2014) as initialization. We used Stochastic Gradient Descent (SGD) with momentum as optimizer and 0.01 as learning rate with a decay factor of 0.005. Even if the scores on synthetic data were promising, to overcome the issue of domain shift, we fine-tuned the network. After the best pipeline had been identified, we improved it with a refined segmentation model. In details:

- We manually annotated a small real dataset consisting of only 30 images acquired with a Real-Sense D435i camera;
- We apply a random augmentation to them (noise addition, blurring effects, and contrast/brightness modifications);
- We fine-tune our model for a few epochs (100), initializing from the weights learned with the simulated dataset but using a new dataset. This new dataset comprises 330 simulated images from the previous dataset and an additional 330 images from real augmented data. The time required for the second training is minimal, taking less than one hour.

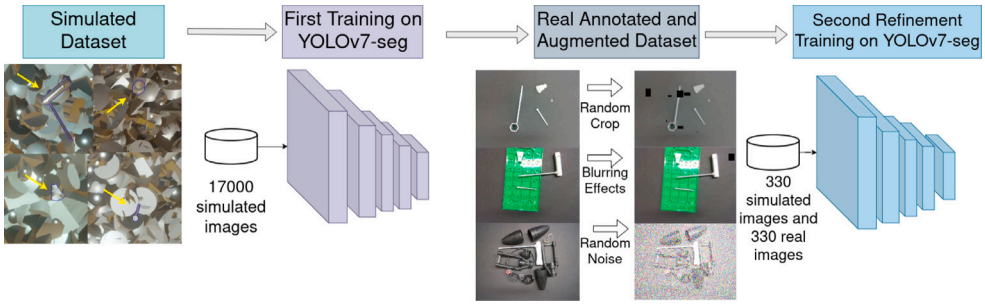


Fig. 5. Resume of the Segmentation Phase.

In Section 4 we detail our experiments based on this network. Its notable capability lies in its ability to generalize across different scenarios, objects, and lighting conditions, achieved through learning from a modest set of only 30 real images and over 15,000 synthetic images. Both the simulated and real datasets are easily created and adaptable to a wide range of objects.

3.2.2. Augmented autoencoder

AAE (Sundermeyer et al., 2020b) is based on the AE structure, to obtain an image representation in a low-dimensional Euclidean space. During the training phase, AAE applies a technique called Domain Randomization (DR), modifying the original image x with some operations (multiplication, sum, inversion, occlusion) and adding real images as background, obtaining a new randomly augmented image f_{aug}. Then, an encoder(ϕ)-decoder(ψ) architecture, based on convolutional and deconvolutional layers, reconstructs the original input $x = \psi(\phi(f_{aug}(x)))$ and, with backpropagation, the network's parameters are learned by minimizing the per-sample loss function: $\mathcal{L} = \sum_{i \in D} \|x_i - \hat{x}_i\|_2$. After the training phase, which differs for each type of object, a *codebook* is created (offline) by generating a latent representation $z_i \in \mathbb{R}^l$ of each one of the n object views, and associating them with their correspondent rotation $\mathbf{R}_i$. Then, during the test phase, the cropped image x_{test} with the target object, received by AAE receives as input, goes through the encoder, which returns its latent space features z_{test}. Finally, the cosine similarity is computed between the input latent representation vector z_{test} and all z_i from the object's codebook, returning rotation matrix with the highest similarity as 3D object orientation.

We notice that the definition of f_{aug} has to be taken very carefully. Indeed, Sundermeyer et al. (2020b) presented Hypothesis 1, previously explained in Section 2, for noise. It was validated with a toy dataset that it also holds for geometric transformations. However, the toy dataset consists of binary images against a uniformly black background. We argue that this proof cannot be directly extrapolated to the case of the Original AAE, primarily due to the incorporation of the Visual Object Classes (VOC) dataset (Everingham et al., 2009) as background. Specifically, noise addition's impact differs from introducing a background with structured objects (as illustrated in Fig. 6).

To contextualize and motivate this discussion, some considerations are necessary. In the field of AE recently there has been great interest in the Deep Image Prior technique of Ulyanov et al. (2020). Among denoising applications, the authors explain the concept of impedance property to noise, i.e., noise is reconstructed with great difficulty by the AE, while structured images, such as natural images, do not suffer from this impedance, making their reconstruction easier. This occurs precisely because only the structured information of an image is stored in the latent space. For this reason, while adding noise to the image does not affect its latent space, adding a structured background, such as geometric figures or natural images, can lead to background-related information being stored in latent space, leading to a deterioration in the performance of the method. Therefore, we avoid the latent space to represent background structured objects, by removing them. To do this, we substitute 2D detection with segmentation, masking the image

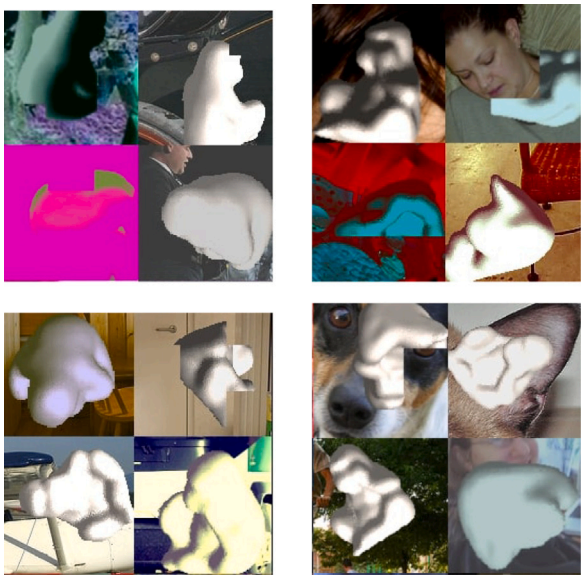


Fig. 6. Examples of images created for the training dataset.

before AAE prediction. Since the first phase includes segmentation, we tried also a different AAE training where the Augmentation phase is changed. We save only data augmentation such as lights and colors effects or occlusions, while we only use black background images instead of the VOC dataset (Everingham et al., 2009) used by the paper (Sundermeyer et al., 2020b), introducing a Less Augmented Autoencoder (LessAAE). Segmentation as the first phase turned out to be the best solution for our application, significantly improving results in Section 4. For comparison, in line with the paper, we use YOLOv4 (pipeline proposed by Sapienza et al. 2023) and YOLOv7 as 2D detectors, since they demonstrated better results than RetinaNet and SSD in Bochkovskiy et al. (2020) and Wang et al. (2022). Then we applied Original AAE with VOC dataset images as background. In our new pipeline, we use YOLOv7 for instance segmentation and a LessAAE. This new pipeline presents many advantages, as shown in Section 4.

3.2.3. Picking point computation

The primary aim of this work was to improve object picking for our industrial scenario. Therefore, to validate our pipeline, we conducted real-world picking experiments. Concerning them, once the 6D pose of an object has been estimated through the proposed pipeline, we identified a priori, for each object's type, a single picking point, allowing the robotic arm to perform the grasping. Typically the origin of the 3D CAD model reference system and the chosen picking point differ from each other. We must therefore find the correct transformation matrix, to properly project the named picking point in the camera reference system. Given its predicted rotation $\mathbf{R}_{pp}$ and translation $\mathbf{t}_{pp}$, generally

explained in Section 3.2, the 4 × 4 pose matrix, called $\mathbf{P}_{pp}$ is described as follows:

$$\mathbf{P}_{pp} = \begin{bmatrix} \mathbf{R}_{pp} & \mathbf{t}_{pp} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

To enhance picking performance we chose to use depth information obtained from the RealSense D435i. This decision is based on the observation that the pipeline's depth estimation compared to the RealSense's depth measurement has an average error of 16.51%, with a standard deviation of 8.94%. This error is computed across 51 experiments. The reading of the depth allows us to achieve significantly better performance.

4. Experimental results

4.1. Evaluation metrics

The most widely used and known metric in 6D pose estimation problems is the Average Distance to the correspondent(-closest) model point (ADD(-S)) error (Hinterstoisser et al., 2012).

$$ADD(-S) = \begin{cases} ADD - S, & \text{if } obj \text{ sym} \\ ADD, & \text{if } obj \text{ asym} \end{cases}$$

Given the object model $\mathcal{M}$, the estimated pose $\hat{\mathbf{P}}$ and the ground-truth $\overline{\mathbf{P}}$

$$e_{ADD} = avg_{x \in \mathcal{M}} \|\hat{\mathbf{P}}x - \overline{\mathbf{P}}x\|$$

If the model $\mathcal{M}$ has ambiguous poses, i.e. it presents geometric or self-occlusion-induced symmetries, the error is calculated as the ADD-S:

$$e_{ADD-S} = avg_{\mathbf{x}_1 \in \mathcal{M}} \min_{\mathbf{x}_2 \in \mathcal{M}} \|\hat{\mathbf{P}}x_1 - \overline{\mathbf{P}}x_2\|$$

e_{ADD-S} is not pose-ambiguity invariant, but only yields relatively small errors even for distinguishable views. Each estimated pose is considered correct if $e < \theta = k_m * d$, where k_m is a constant and d = object diameter. The threshold $k_m * d$ could be arbitrarily chosen, depending on the applications, as claimed by Hodaň et al. (2016). k_m defines the percentage of diameter to be used as the threshold. We experimented three different values of k_m: 0.1 (as used by Hinterstoisser et al. 2012), 0.2, 0.3. As measure of correctness, the Recall percentage is used.

4.2. Results

In the industrial use case described in Section 3.1, we take into consideration the four objects depicted in Fig. 3 for the pipelines.

All combinations of methods result in six different pipelines. The results in the ablation study from Table 1 show that segmentation as the first stage is in most cases a winning option. For the spark plug key, the composition of segmentation and LessAAE brings a significant improvement. For the other objects, the best pipeline is composed by segmentation and original AAE. Only the nut presents an outlier result. On average, the best pipeline is our customized one (YOLOv7-segmentation and LessAAE) with 26.23% of success, followed by YOLOv7-segmentation with AAE which achieves 21.75%. The first-row pipeline, considered as a baseline, achieves only 14.44%. This highlights one more time our consideration (Section 3.2.2) on the latent space: if there are background objects, the latent representation could be affected by them. With the segmentation phase, we avoided this problem. In particular, we can observe from Table 1 that the improvement in segmentation usage becomes more pronounced when objects have an elongated and thin shape (such as the spark plug key and the bolt). Considering only the bounding box, in thin objects, there are many background pixels that the AAE will receive as input and, as a consequence, wrong information could be represented in the latent space. Therefore, in this type of object, it becomes powerful to eliminate the background within the bounding box to avoid it affecting the latent space of the AAE. In the case of the nut, segmentation yields better results for $k_m = 0.1$, although they remain very low. This is probably due to the small size of the object, and thus, the nut is represented by very few pixels. Despite this, 6D pose estimation for this object is less relevant for the successful completeness of a picking, given the shape of the nut and that its picking point is in the center of the bounding box.

For segmentation, YOLOv7-Seg has been trained for 500 epochs. The backbone and head of YOLOv7-Seg are based on the YOLOv7 detector (Wang et al., 2022). We computed precision, recall, and mean Average Precision (mAP) on the validation set with a threshold of 0.5, which respectively achieve the following scores: 0.997, 0.997, 0.994. Details for each class are presented in Table 2. In the fine-tuning phase, we trained the same architecture for only 100 epochs and a small dataset, comprising 330 real images and 330 simulated images. The network's hyperparameters remain unchanged. While the mAP decreased on the new training, qualitative results and real experiments were significantly improved after this step. This is evident in Fig. 7: Fig. 7(a) shows the drop in F1-score performance, while Fig. 7(b) compares two real images predicted by the YOLOv7 on simulated data (on the top) and YOLOv7 fine-tuned (on the bottom). Predictions of the second network show a great qualitative improvement. This is because the network with the fine-tuning phase is more able to generalize and less fitted on geometric simulated images. As a consequence, the metrics decrease on the validation but increase on real-world images. A proof of this improvement is shown by segmentation performances on real images at Table 4. In detail, we computed recall (R), precision (P), and mAP on 48 annotated real images comprising 64 instances.

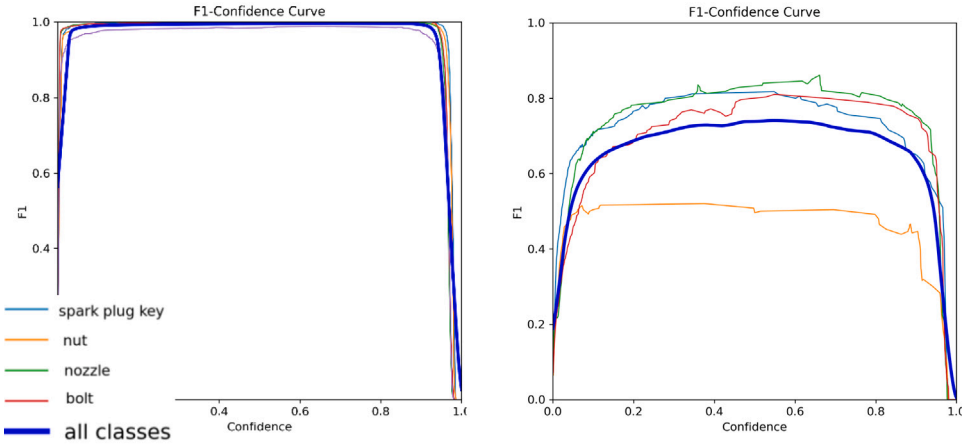
Details on the AAE and LessAAE hyperparameters are summed up in Table 3, to show that the only significant changes are the different choices of background images.

4.3. Real-world experiments

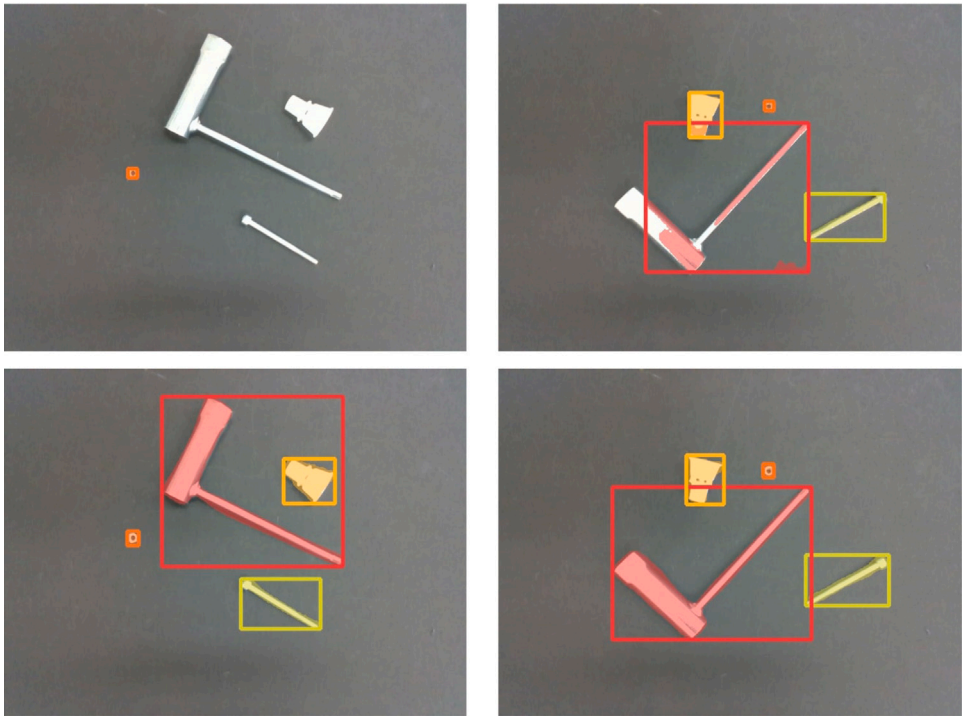
Our dataset and improved pipeline have been tested in inference with real-world experiments on the four mentioned objects (Section 3), using an UR5e cobot from Universal Robot, on which a Real-Sense D435i camera was installed. First, the RGB-D frame is acquired by the stereo-camera (Section 3.2.1), then the pipeline (segmentation and 6D pose estimation) is applied to the RGB image only. If more objects are segmented by the YOLOv7, considering the fact we were not working on a cluttered scenario, the object with the highest confidence score is chosen as the first to be picked. After this, the picking point of the chosen object is computed, also considering the reading of the depth from the depth frame. Finally the information related to translation and rotation are passed to the cobot. To give an exhaustive representation of the results, we changed the experimental environment by considering different lights (high or low level), different camera settings and single or multi-object disposition (Fig. 8). In details we conducted 16 attempts per object for the two pipelines (64 attempts in total). We selected two different camera settings and two different light conditions. The scenes were distinguished in plain and with-objects backgrounds. In total, the different options proposed in these experiments were the following:

- $L = 2$ light options: low or strong;
- $B = 2$ backgrounds: black or with objects (semi-cluttered);
- $C = 2$ camera settings with different saturation and brightness values;

Examples of different lights and camera settings are provided in Fig. 8, where the first two images were acquired with camera setting II and respectively low light, strong light. The third image is acquired with camera setting I and low light, while the fourth image is with camera setting I and strong light. All possible combinations were tried ($L \times B \times C$). Table 5 presents quantitative results, while Fig. 9 shows qualitative results.



(a) F1 score curve of the YOLOv7 trained only on simulated data (on the left) and YOLOv7 fine-tuned on real augmented data (on the right).



(b) Two real images respectively predicted by YOLOv7 trained only on simulated data (on the top) and by YOLOv7 fine-tuned on real augmented data (on the bottom) .

Fig. 7. In Figure (a) F1 score, in Figure (b) qualitative comparison on two real images.

Fig. 8. Examples of changing light and camera settings for experiments.

Table 1
Ablation study: ADD(-S) recall results across different pipelines for the four objects with a geometric background.

Detector	6D estimator	ADD(-S) recall				
		Spark plug key		Nut	Nozzle	Bolt
YOLOv4-BBs	AAE	$k_m = 0.1$	4.92%	1.68%	42.37%	8.78%
		$k_m = 0.2$	21.41%	8.70%	61.04%	19.18%
		$k_m = 0.3$	41.9%	12.08%	69.28%	28.16%
YOLOv7-BBs	AAE	$k_m = 0.1$	4.61%	1.51%	50.8	12.95%
		$k_m = 0.2$	20.08%	4.53	71.0%	24.69%
		$k_m = 0.3$	40.36%	7.8	82.8%	43.21%
YOLOv7-seg	AAE	$k_m = 0.1$	7.82%	2.73%	**61.6%**	**14.86%**
		$k_m = 0.2$	29.26%	8.4%	**78.80%**	**29.91%**
		$k_m = 0.3$	47.09%	13.94%	**86.80%**	**39.95%**
YOLOv4-BBs	LessAAE	$k_m = 0.1$	1.93%	1.12%	7.43%	0.20%
		$k_m = 0.2$	14.99%	**8.99%**	16.67%	0.41%
		$k_m = 0.3$	28.69%	**14.05%**	22.29%	0.61%
YOLOv7-BBs	LessAAE	$k_m = 0.1$	2.2%	0.0%	6.00%	0.0%
		$k_m = 0.2$	17.87%	0.76%	10.00%	0.23%
		$k_m = 0.3$	33.53%	1.26%	13.00%	0.47%
YOLOv7-seg	LessAAE	$k_m = 0.1$	**39.48%**	**3.03%**	57.00%	5.42%
		$k_m = 0.2$	**78.58%**	8.78%	74.20%	13.25%
		$k_m = 0.3$	**88.98%**	12.42%	82.39%	23.90%

Table 2
Mean Average Precision (mAP) with threshold $\theta = 0.5$ for detection and segmentation by YOLOv7.

mAP $\theta = 0.5$	Box	Mask
Spark plug key	0.993	0.964
Nozzle	0.987	0.849
Nut	0.995	0.995
Bolt	0.995	0.988

Table 3
Details on hyperparameters of 6D estimators for our industrial use-case.

	LatentSpaceDim	N iter	Background imgs	Optimizer	BatchSize
AAE	256	50 000	VOC Dataset	Adam	32
LessAAE	256	40 000	Black image	Adam	32

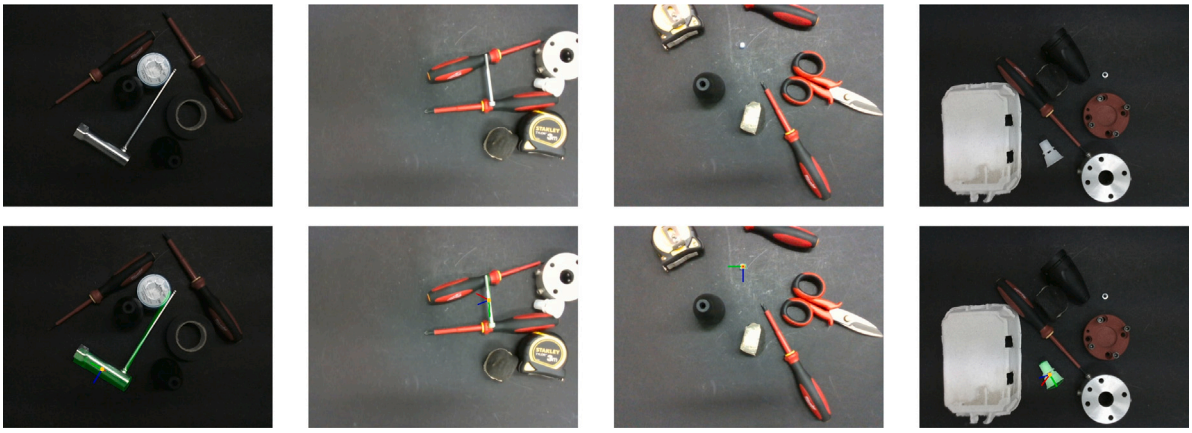


Fig. 9. Qualitative results with the four objects in a random semi-cluttered scenario with low and high light-levels. On the top: original images; on the bottom: the prediction of the picking point.

4.4. Discussion

The average success of picking achieves the 75%. Our customized pipeline, our synthetic dataset, and the addition of a sim-to-real fine-tuning procedure during the segmentation phase achieve make it possible to reach this percentage of success. First, we claim that segmentation reaches better results than detection in this application, given the ablation study in Table 1. In particular, our pipeline (in the last row) gains an improvement of +11.79% with respect to the original pipeline (in the first row), on simulated test images. Furthermore, we observed that the fine-tuning step on real data improves by +65.7% the segmentation mAP on real images. In addition, the final results in Table 5 confirm the superiority with +21.875% of our LessAAE approach with respect to the standard AAE. The experiments that gave rise to Tables 1 and 5 are the same, for a total of 64 picking attempts. However, in some cases even if the segmentation was wrong, the picking was successful (2% cases) and vice versa (8%). In Table 5, only the fine-tuned YOLOv7 is considered for segmentation, because YOLOv7-only-sim

Table 4
Correct segmentation in real-world experiments.

Objects	Correct segmentation					
	YOLOv7 only sim			**YOLOv7 with real**		
	R	P	mAP	R	P	mAP
Spark plug key	21.1%	14.8%	15.5%	89.5%	89.5%	93.9%
Nozzle	26.7%	28.6%	25.5%	93.3%	93.3%	92.4%
Nut	41.7%	25.0%	37.9%	66.7%	100%	83.3%
Bolt	0%	0%	0%	76.9%	58.8%	72.1%
Average	22.4%	17.1%	19.7%	**81.6%**	**85.4%**	**85.4%**

Table 5
The percentages of single object picking. The Table shows the results of the two different pipelines for the four objects separately and the average over different camera settings and light conditions.

Objects	Successful picking	
	Real+AAE	Real+LessAAE
Spark plug key	62.5%	87.5%
Nozzle	62.5%	62.5%
Nut	75.0%	75.0%
Bolt	12.5%	75.0%
Average	53.125%	**75.0%**

shows lower performances. Then, the two pipelines' difference lies in the pose estimation step, with the original AAE and the LessAAE. After the ablation study on the simulated dataset and the picking experiments on real scenarios, we claim that our proposed pipeline achieves the best performance. Nevertheless, there is a 25.0% of failures. In most of this percentage, the reason why it fails is given by misunderstanding of other background objects or low lights. This means that in some limit-situations the model should be improved. Since we proposed an original use case with new challenging industrial objects, it is impossible to have a comparison with other works. However, we provide an extensive analysis of results with different pipelines. Moreover, we make the data generation, the dataset, and the pipeline code available, to give everyone the possibility to compare and achieve better results in the future. Given the lack of 6D pose data, the dataset and data generation pipeline are significant for the scientific community. In addition, our pipeline brings improvements to an existing method and sheds light on the concept of implicitly capturing an object's pose through the latent space of an AE, in contrast with the current prevalent 2D–3D correspondence approach.

5. Conclusions

In conclusion, this paper introduces a novel industrial use case addressing the challenge of grasping known objects by predicting their 6D pose from an RGB frame. In particular, we introduce objects that are challenging yet widely used in the industrial domain. We proposed a new pipeline that significantly improves the performance of an existing algorithm for 6D pose estimation: AAE. Moreover, we introduced a novel version of AAE called LessAAE, which significantly helps the learning procedure. To mitigate the issue of domain shift, we also proposed a training procedure that uses thousands of simulated images and only 30 real images, manually annotated. To evaluate the performance of our approach, we conducted extensive experiments, which included simulated test images and real-world grasping experiments utilizing a robotic arm. The results demonstrated the improvement provided by our method in both simulated and real-world scenarios. These findings highlight the potential of our approach in the field of robotic grasping and provide a foundation for future research and practical applications of these objects. By improving an existing approach with mathematical considerations about its assumptions, introducing an innovative dataset, and a well-structured pipeline, we achieved more reliable and versatile object-grasping capabilities in robotics, with applications for industrial automation, manufacturing, and various other industries. Our future plans include pursuing two varied directions.

One aspect focuses on theoretical exploration, aiming to extensively delve into the latent space and enhance its capacity for pose representation. Conversely, we will work to improve our simulated dataset by generating new realistic images in a new totally cluttered scenario and bridge the domain gap between reality and simulation.

Formatting of funding sources

This research was made possible through the support of Cifarelli Spa. The findings and opinions presented in this work are those of the authors and do not necessarily reflect the views of Cifarelli Spa.

This work was partly supported by the "National Group of Computing Science (GNCS-INdAM)". The publication was created with the co-financing of the European Union-FSE-REACT-EU, PON Research and Innovation 2014–2020 DM1062/2021.

CRediT authorship contribution statement

Elena Govi: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Davide Sapienza:** Writing – review & editing, Validation, Resources, Project administration, Investigation, Formal analysis, Conceptualization. **Samuele Toscani:** Writing – review & editing, Validation, Software, Resources, Investigation, Formal analysis. **Ivan Cotti:** Software, Data curation. **Giorgia Franchini:** Writing – review & editing, Supervision. **Marko Bertogna:** Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Cao, H., Dirnberger, L., Bernardini, D., Piazza, C., Caccamo, M., 2023. 6IMPOSE: bridging the reality gap in 6D pose estimation for robotic grasping. Front. Robotics AI 10, 1176492.

Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A., 2009. The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. 88, 303–308.

Fischler, M.A., Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24 (6), 381–395.

He, Y., Huang, H., Fan, H., Chen, Q., Sun, J., 2021. Ffb6d: A full flow bidirectional fusion network for 6d pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3003–3013.

He, Y., Sun, W., Huang, H., Liu, J., Fan, H., Sun, J., 2020. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11632–11641.

Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V., 2011. Gradient response maps for real-time detection of textureless objects. IEEE Trans. Pattern Anal. Mach. Intell. 34 (5), 876–888.

Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N., 2012. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision. Daejeon, Korea, November 5-9, 2012, Revised Selected Papers, Part I 11, Springer, pp. 548–562.

Hodan, T., Barath, D., Matas, J., 2020. Epos: Estimating 6d pose of objects with symmetries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11703–11712.

Hodan, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., Zabulis, X., 2017. T-LESS: An RGB-d dataset for 6D pose estimation of texture-less objects. In: 2017 IEEE Winter Conference on Applications of Computer Vision. WACV, IEEE, pp. 880–888.

Hodaň, T., Matas, J., Obdržálek, Š., 2016. On evaluation of 6D object pose estimation. In: Computer Vision–ECCV 2016 Workshops: Amsterdam, the Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14. Springer, pp. 606–619.

Hodan, T., Michel, F., Brachmann, E., Kehl, W., GlentBuch, A., Kraft, D., Drost, B., Vidal, J., Ihrke, S., Zabulis, X., et al., 2018. Bop: Benchmark for 6d object pose estimation. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 19–34.

Hu, Y., Fua, P., Salzmann, M., 2022. Perspective flow aggregation for data-limited 6d object pose estimation. In: European Conference on Computer Vision. Springer, pp. 89–106.

Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J., 1993. Comparing images using the hausdorff distance. IEEE Trans. Pattern Anal. Mach. Intell. 15 (9), 850–863.

Kaskman, R., Zakharov, S., Shugurov, I., Ilic, S., 2019. Homebreweddb: Rgb-d dataset for 6d pose estimation of 3d objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.

Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N., 2017. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1521–1529.

Kendall, A., Grimes, M., Cipolla, R., 2015. Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2938–2946.

Kleeberger, K., Bormann, R., Kraus, W., Huber, M.F., 2020. A survey on learning-based robotic grasping. Curr. Robotics Rep. 1, 239–249.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2980–2988.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. Springer, pp. 740–755.

Lin, S., Wang, Z., Ling, Y., Tao, Y., Yang, C., 2022. E2EK: End-to-end regression network based on keypoint for 6D pose estimation. IEEE Robot. Autom. Lett. 7 (3), 6526–6533.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C., 2016. Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer, pp. 21–37.

Lu, Z., Zhang, Y., Doherty, K., Severinsen, O., Yang, E., Leonard, J., 2022. SLAM-supported self-training for 6D object pose estimation. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 2833–2840.

Marullo, G., Tanzi, L., Piazzolla, P., Vezzetti, E., 2023. 6D object position estimation from 2D images: a literature review. Multimedia Tools Appl. 82 (16), 24605–24643.

Nguyen, V.N., Hu, Y., Xiao, Y., Salzmann, M., Lepetit, V., 2022. Templates for 3d object pose estimation revisited: Generalization to new objects and robustness to occlusions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6771–6780.

Peng, S., Liu, Y., Huang, Q., Zhou, X., Bao, H., 2019. Pvnet: Pixel-wise voting network for 6dof pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4561–4570.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788.

Sapienza, D., Govi, E., Aldhaheri, S., Bertogna, M., Roura, E., Pairet, È., Verucchi, M., Ardón, P., 2023. Model-based underwater 6D pose estimation from RGB. IEEE Robot. Autom. Lett..

Su, Y., Saleh, M., Fetzer, T., Rambach, J., Navab, N., Busam, B., Stricker, D., Tombari, F., 2022. Zebrapose: Coarse to fine surface encoding for 6dof object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6738–6748.

Sundermeyer, M., Durner, M., Puang, E.Y., Marton, Z.-C., Vaskevicius, N., Arras, K.O., Triebel, R., 2020a. Multi-path learning for object pose estimation across domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13916–13925.

Sundermeyer, M., Marton, Z.-C., Durner, M., Triebel, R., 2020b. Augmented autoencoders: Implicit 3d orientation learning for 6d object detection. Int. J. Comput. Vis. 128, 714–729.

Tekin, B., Sinha, S.N., Fua, P., 2018. Real-time seamless single shot 6d object pose prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 292–301.

Thalhammer, S., Bauer, D., Hönig, P., Weibel, J.-B., García-Rodríguez, J., Vincze, M., 2023. Challenges for monocular 6D object pose estimation in robotics. arXiv: 2307.12172.

Thalhammer, S., Leitner, M., Patten, T., Vincze, M., 2021. Pyrapose: Feature pyramids for fast and accurate object pose estimation under domain shift. In: 2021 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 13909–13915.

Tyree, S., Tremblay, J., To, T., Cheng, J., Mosier, T., Smith, J., Birchfield, S., 2022. 6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 13081–13088.

Ulyanov, D., Vedaldi, A., Lempitsky, V., 2020. Deep image prior. Int. J. Comput. Vis. (128), 1867–1888.

Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M., 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696.

Wang, G., Manhardt, F., Shao, J., Ji, X., Navab, N., Tombari, F., 2020. Self6d: Self-supervised monocular 6d object pose estimation. In: Computer Vision–ECCV 2020: 16th European Conference. Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16, Springer, pp. 108–125.

Wang, G., Manhardt, F., Tombari, F., Ji, X., 2021. Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16611–16621.

Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J., 2019. Normalized object coordinate space for category-level 6d object pose and size estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2642–2651.

Xiang, Y., Schmidt, T., Narayanan, V., Fox, D., 2017. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv: 1711.00199.