automation

MDPI

Article

Detection of Novel Objects without Fine-Tuning in Assembly Scenarios by Class-Agnostic Object Detection and Object Re-Identification

Markus Eisenbach *, Henning Franke, Erik Franze, Mona Köhler, Dustin Aganian, Daniel Seichter and Horst-Michael Gross

Neuroinformatics and Cognitive Robotics Lab, Ilmenau University of Technology, 98693 Ilmenau, Germany
* Correspondence: markus.eisenbach@tu-ilmenau.de

Abstract: Object detection is a crucial capability of autonomous agents for human–robot collaboration, as it facilitates the identification of the current processing state. In industrial scenarios, it is uncommon to have comprehensive knowledge of all the objects involved in a given task. Furthermore, training during deployment is not a viable option. Consequently, there is a need for a detector that is able to adapt to novel objects during deployment without the necessity of retraining or fine-tuning on novel data. To achieve this, we propose to exploit the ability of discriminative embeddings learned by an object re-identification model to generalize to unknown categories described by a few shots. To do so, we extract object crops with a class-agnostic detector and then compare the object features with the prototypes of the novel objects. Moreover, we demonstrate that the embedding is also effective for predicting regions of interest, which narrows the search space of the class-agnostic detector and, consequently, increases processing speed. The effectiveness of our approach is evaluated in an assembly scenario, wherein the majority of objects belong to categories distinct from those present in the training datasets. Our experiments demonstrate that, in this scenario, our approach outperforms the current best few-shot object-detection approach DE-ViT, which also does not perform fine-tuning on novel data, in terms of both detection capability and inference speed.

Keywords: object re-identification; class-agnostic detection; adaption to novel categories; assembly scenario

Citation: Eisenbach, M.; Franke, H.; Franze, E.; Köhler, M.; Aganian, D.; Seichter, D.; Gross, H.-M. Detection of Novel Objects without Fine-Tuning in Assembly Scenarios by Class-Agnostic Object Detection and Object Re-Identification. *Automation* **2024**, *5*, 373–406. https://doi.org/10.3390/automation5030023

Academic Editor: Duc Truong Pham

Received: 29 June 2024
Revised: 5 August 2024
Accepted: 13 August 2024
Published: 19 August 2024

1. Introduction

In order to enable autonomous human–robot collaboration in assembly, a robot assistant must be able to recognize the assembly process and identify subtasks. To achieve this, it needs comprehensive scene-understanding capabilities [1], including human action recognition and the detection of tools, workpieces, and objects involved in the current assembly step. The integration of objects for human action recognition has shown high value in previous work [2] and is, therefore, a crucial component for identifying processing steps. When training a detector for industrial or manufacturing scenarios, it is uncommon to have comprehensive knowledge of all the objects that will be relevant in the later application. Even if there is some knowledge, it is not feasible to create a comprehensive dataset of all the objects involved. Therefore, it is preferable to have a detector that does not depend on complete knowledge during training but can adapt to novel objects during deployment without the need for retraining or fine-tuning on novel data. This allows the detector to be used as a standalone product with flexible black-box components for localizing and classifying unknown objects [3]. One potential solution is few-shot object detection (FSOD) [4,5]. However, most FSOD approaches require fine-tuning on target data, which is generally not feasible in the target scenario. In this paper, we present a general framework that addresses the problem of detecting novel objects that are unknown during training, and apply it to the assembly dataset ATTACH [6], which adequately represents the addressed target

scenario. Additionally, we apply it on the assembly dataset IKEA-ASM [7]. To implement a well-generalizing detector for novel objects, we employ a three-step approach: First, we apply a feature-based search for regions of interest as a preprocessing step to reduce the computational load. Second, we apply a class-agnostic detector to generate object proposals. Finally, we assign the novel categories to the proposals by applying object re-identification (see Figures 1 and 2).

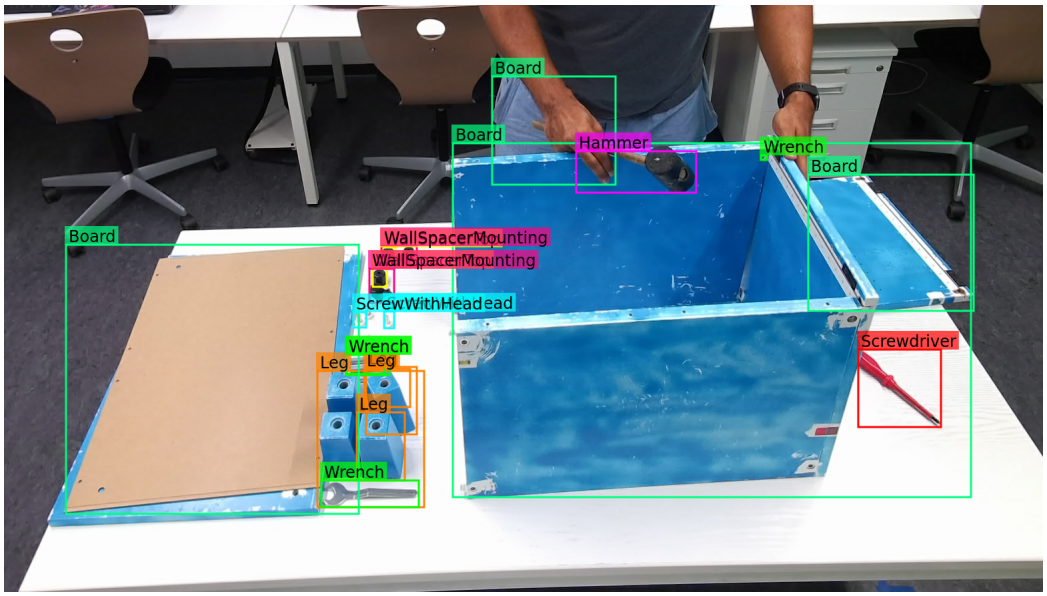


Figure 1. Detection result of our approach on a crop of an HD image of the ATTACH dataset depicting the workplace. This dataset adequately represents the target scenario and contains mainly novel categories that were not included in the training data. The category information is taken from 20 shots per category. The detector is not trained on these shots. However, it is able to adapt to novel categories by employing a class-agnostic detector and an object re-identification model as proposed.

Our contributions are as follows:

- Firstly, we demonstrate that training a class-agnostic detector on a sufficiently large and densely labeled dataset is sufficient to generalize well to novel objects. We compare different models and determine which architectures can detect the most novel objects in an assembly scenario while being fast.
- Secondly, we show that an object re-identification approach can be trained on a 3D object reconstruction dataset and utilized to assign novel categories to proposals without the need for fine-tuning. Our experiments indicate that this training approach outperforms the alternative in related work, wherein classification and proposal generation utilize the same detection dataset.
- Thirdly, we incorporate an additional first-filtering step based on object re-identification before applying the class-agnostic detector. This pre-processing step demonstrates considerable potential to accelerate the overall pipeline and enhance the detection performance by reducing the search space of the detector.

Our proposed processing pipeline comprises dedicated components for class-agnostic object proposal and object re-identification. None of the components require fine-tuning on target domain data. We demonstrate the superiority of the pipeline, designed in this way, over the current best FSOD approach under identical conditions, i.e., identical dataset, hardware, and shots provided to introduce novel categories.

Our paper is organized as follows: In Section 2, we discuss the motivation of our approach and differentiate it from related work. In Section 3, we present our approach for detecting novel objects without fine-tuning. In Section 4, we evaluate the capabilities of the subcomponents in experiments on the assembly dataset ATTACH. Additionally, we

compare our detector with the best current few-shot object-detection approach DE-ViT [8] on the assembly datasets ATTACH and IKEA-ASM. Finally, in Section 6, we conclude our work and present an outlook to future research.

2. Related Work

In order to detect novel objects in an industrial scenario, related work describes several approaches to handle novel categories (Section 2.1). Some approaches are able to classify novel categories without fine-tuning (Section 2.2), which is a prerequisite in the target scenario. This can also be achieved with only a few shots of the novel categories (Section 2.3). Some approaches rely on object re-identification techniques to learn a discriminative embedding to generalize well to novel categories (Section 2.4), which is consistent with our way of handling novel objects. A field, which exhibits great overlap to object re-identification, is content-based image retrieval, which re-identification is sometimes viewed as an application of (Section 2.5).

2.1. Handling Novel Categories

In the target scenario, we must be able to detect objects of novel categories that were not known during training. To illustrate this, an exemplary recognition task in the target scenario is shown in Figure 1 and various objects of novel categories specific for the target scenario are shown in Figure 5.

In related work, novel categories are treated in different ways: One approach is to suppress novel objects as background (open-set detectors, out-of-distribution detectors). Another is to incrementally learn novel categories (open-world detectors). A third is to treat all objects as a single foreground class (class-agnostic detectors). Collectively, all these approaches for dealing with categories are referred to as detection in an open environment [9].

Open-set object detectors [10] address the ability to avoid classifying novel categories as known categories that are part of the training set, and instead treat novel categories as background. Additionally, out-of-distribution object detectors [11] aim to circumvent overconfident false predictions on data that are not present in the training set, which often arise from novel categories. Hence, no bounding boxes are predicted for novel categories.

Open-world object detectors [12] address the open-set problem by classifying all objects into known categories and incrementally learning novel categories based on pseudo-labels for unknown objects. As proposed in [12], most open-world detection approaches train the region proposal network (RPN) on labeled foreground objects and additional background objects pseudo-labeled as a dedicated unknown category. This forces the RPN to be class-agnostic, thereby enabling it to localize objects of novel categories. However, in contrast to common object benchmarks, the real-world target scenario objects are unlikely to be present in the training data, as emphasized by Zhao et al. [13]. Therefore, pseudo-labeling background objects in a sparsely labeled dataset such as COCO [14] will not enhance the model's ability to generalize to the target scenario. Instead, using a sufficiently densely labeled dataset such as LVIS [15] as a more densely labeled version of COCO is just as effective. This comes with an increased number of different categories, which further enhances the generalization of the proposal networks to novel categories [16,17]. Consequently, for training a class-agnostic detector, we utilize the LVIS dataset with 1203 labeled categories instead of the COCO dataset [14] with only 80 labeled categories, which is the most commonly used dataset in related work.

Class-agnostic detectors do not differentiate between known and novel objects. Instead, they treat all objects as a single category and binary distinguish between foreground and background. The concept of training an object-or-not binary classification and bounding box regression from scratch for class-agnostic object detection was first proposed in [18], which influenced modern object detectors. It is also used as first stage in open-set detection [19]. Jaiswal et al. [20] introduce an additional adversarial-learning framework for class-agnostic object detection, which performs slightly better than the approach of [18] on the VOC

and COCO datasets. However, in preliminary experiments, we found that on datasets with many categories both approaches perform on par. Therefore, we decided in favor of the more simplistic approach without adversarial learning. In addition to the dataset, a decision must be made about the model to be used for class-agnostic detection, since the choice of model affects the class-agnostic performance. In particular, transformers perform well [21]. In [22] the Segment Anything Model (SAM) [23] is utilized for class-agnostic object localization as part of an open-world object detector. Consequently, we also investigate the use of transformers and the Segment Anything Model (SAM) for class-agnostic detection.

Another decision to be made about the detector is whether the features that have been extracted for object localization should be reused in subsequent processing steps. While class-agnostic proposals are a necessity to localize objects of novel categories, Han et al. [24] found that class-agnostic proposals harm the classification performance. Therefore, we do not reuse detector features for classifying object proposals. Instead, we use a dedicated object re-identification module, which uses a separate backbone.

2.2. Novel Category Classification without Fine-Tuning

In the target scenario, the detector must be able to identify novel objects without fine-tuning on novel data. In related work, this is often achieved through additional language cues (zero-shot detectors, open-vocabulary detectors).

Zero-shot object detectors [25–28] and open-vocabulary object detectors [29–31] are designed to recognize the novel categories without the need for training on novel category samples. In most cases, this is accomplished by exploiting the zero-shot capability of aligned vision and language models that are available as pre-trained foundation models [32]. However, in the target scenario, it is difficult to sufficiently describe the objects of interest with category words, since highly similar objects can only be differentiated through comprehensive descriptions, e.g., the various screws and assembly parts in Figures 1 and 5. Additionally, the components that constitute the assembled piece are often intricate and cannot be adequately described with words alone. Instead, it is easier to describe the novel categories by providing few visual examples.

2.3. Few-Shot Object Detection without Fine-Tuning

In the target scenario, the knowledge about novel categories is provided by few shots. This concept has been explored in related work under the term few-shot object detection [4,5]. Few-shot object detectors employ either fine-tuning or meta-learning to achieve good performance on novel categories. Only meta-learning approaches are able to generalize to novel categories by training only on known categories [5,33], in this context often referred to as base categories. In FSOD approaches that avoid training on novel data, good generalization is achieved mainly by increasing the discriminative power of the embedding in the classification head [5]. Many approaches that are able to train only on base categories [34–46] perform relatively poorly on novel categories compared to approaches that train on novel category samples. Only TIDE [3] and DE-ViT [8] are able to achieve satisfactory detection performance on novel categories without fine-tuning. Both approaches use transformers as backbone and either use the few-shot images as additional input [3] or the embedding vector of them [8]. Thus, the transformer is able to take the novel samples into consideration for localization and classification. To enable such a pipeline, DE-ViT employs a pretrained vision transformer (ViT) [47] and trains only a projection layer to obtain an embedding space that generalizes to novel categories. This is related to our pipeline. As DE-ViT, we first localize regions of interest by comparing them to the representation of novel samples. Another similar approach is then employed, whereby class-agnostic region proposals are generated and classified according to the cosine similarity with novel category prototypes. In contrast to DE-ViT, we apply a re-identification method, which explicitly learns an embedding on a dataset that shows objects from different perspectives, which more accurately reflects the later application on finding

novel objects seen before, most often from another viewpoint. DE-ViT [8] is currently one of the best performing FSOD methods, particularly in cases with very few shots. Therefore, it is used as a reference approach in our experiments.

2.4. Object Re-Identification

After class-agnostic detection, it is necessary to provide category labels for each of the detected objects. To match objects in the input image with category prototypes extracted from the few shots, we utilize the feature embedding of a re-identification model, which has been trained to embed the representations of identical objects in close proximity and to separate the representations of different objects. Vaguely similar approaches have also been explored in related work, as described below.

Some open-set approaches [24,48–50] employ class-agnostic proposals in combination with a learned feature space that aims to increase the separability between known and novel categories by applying contrastive learning, so that novel categories lie in low-density regions of the latent space. This is analogous to the optimization objective of re-identification approaches. Some out-of-distribution object-detection approaches [11,51,52] increase the discrimination of in-distribution data, which reflect known categories during training, and out-of-distribution data, which are most often synthesized outliers. While this is also similar to learning a discriminative embedding, it does not force a better distinction between known and novel categories. The similarity of object embeddings to known-category prototypes in latent space, to separate novel from known categories, is used in [53] based on an aligned text-image space, and in [19,54] based on a learned discriminative embedding space. Moreover, in [55,56] unknown category samples are clustered and classified into one of the novel category clusters based on the similarity to the cluster prototypes. This approach is related to how the learned embedding in re-identification approaches is used to match novel category samples with the few shots that build novel category prototypes.

The combination of class-agnostic detection and some kind of object re-identification has also been explored in related work. A two-step approach in which objects are first class-agnostically localized based on the Segment Anything Model (SAM) [23] and then feature vectors of DINO v2 [57] are used for re-identifying the novel categories is described in [58]. However, it should be noted that all experiments are performed on synthesized data, which limits the value of the results. In [59–61] matching object pairs in two images are detected and identified as such without category knowledge. This application includes similar tasks to ours, such as class-agnostic object detection and similarity comparison of object proposals. However, these approaches are only employed on images that are highly similar to those in the training data. The ability to generalize to out-of-domain data is not examined.

The re-identification of objects in industrial and assembly scenarios after class-agnostic detection has been evaluated in [58,62]. Gorlo et al. [58] train and test only on simulated scenes. Tests on real-world images were not part of their experiments, which limits the meaningfulness of the results. Dummel et al. [62] use query images generated from CAD data to retrieve matching images based on a siamese network that was trained with synthetic data. Since this detector was trained on synthetic data only, Dummel et al. [62] observed a large synthetic-to-real gap for novel categories when applying this approach to retrieve real images. In contrast to these two attempts, our approach involves training on real image data, thereby avoiding the observed decline in performance when our pipeline is applied to real assembly images.

2.5. Content-Based Image Retrieval

A field very related to re-identification of arbitrary objects is contend-based image retrieval (CBIR), specifically instance-level image retrieval. The goal of instance-level CBIR is similar to that of re-identification: Finding those images in a large gallery, which display the same object as a query image by ranking them according to their similarity [63].

The main differences to re-identification is that instance-level CBIR models are usually trained for the domain of landmark retrieval [64,65], since, to the best of our knowledge, appropriately labeled datasets of more general domains do not exist. Additionally, CBIR is usually not utilized as a downstream application of an object detector and does not require precisely cropped images of the instances being retrieved.

Since object re-identification models, which we introduce in this paper, do not exist as such, it is necessary to employ an adequate reference approach. For this reason, we decided in favor of the instance-level CBIR model SuperGlobal [66]. SuperGlobal uses a pretrained ResNet backbone from [67] and introduces advanced feature-pooling and re-ranking methods, thereby achieving state-of-the-art retrieval performance.

3. Detection of Novel Objects without Fine-Tuning

A suitable detector in the target scenario must be able to adapt to novel categories based on few shots provided for each category, without the necessity of fine-tuning on novel data. Therefore, we employ a class-agnostic object detector to localize any objects. To provide category labels for each of the detected objects, we apply an object re-identification (ReID) model to match the detected objects with the provided few shots. Furthermore, to reduce the computational load when processing high-resolution images, regions of interest (RoI) proposal is utilized as a preprocessing step.

Figure 2 shows the entire pipeline, consisting of (I) a model for RoI proposal, (II) a class-agnostic object detector, and (III) an object ReID model for matching object proposals with the few shots provided for the novel category.

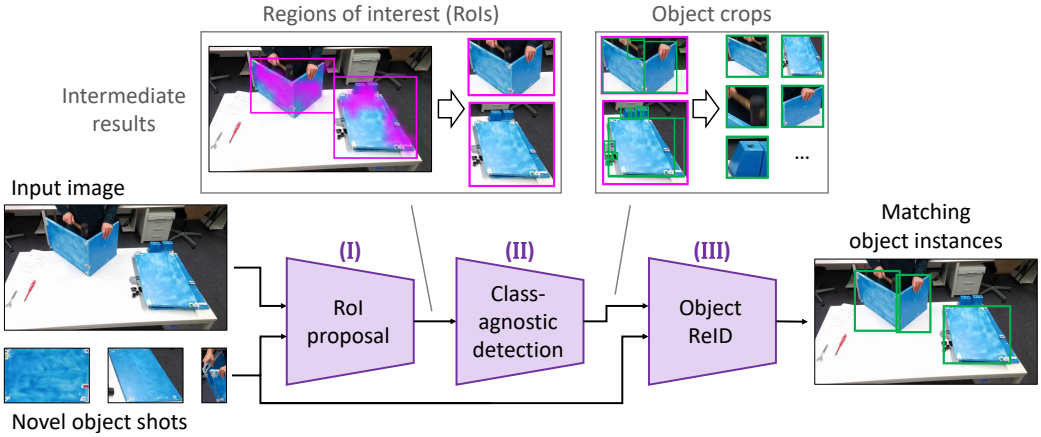


Figure 2. Overall processing pipeline. (I) First, regions of interest (RoIs) are extracted (purple rects). (II) Second, a class-agnostic detector is employed to extract crops for all objects in the scene (green rects). (III) Finally, a re-identification (ReID) model compares the novel object shots with the object proposals to identify matching objects. Figure 3 provides a more detailed view of the stages (I)–(III).

In Section 3.1, we provide a more detailed view on the processing pipeline. Then, we provide details on the three processing stages. In Section 3.2, we describe why we need to predict RoIs in a preprocessing step and how we compute these RoIs based on feature-similarity to the novel objects depicted in the few shots. In Section 3.3, we explain the class-agnostic object detector. In Section 3.4, we explain, how we trained an object re-identification model in order to match the object proposals with the few shots provided for each category.

3.1. Processing Pipeline

A detailed view of our entire pipeline is displayed in Figure 3 and the numbered stages will be continuously referenced in this section.

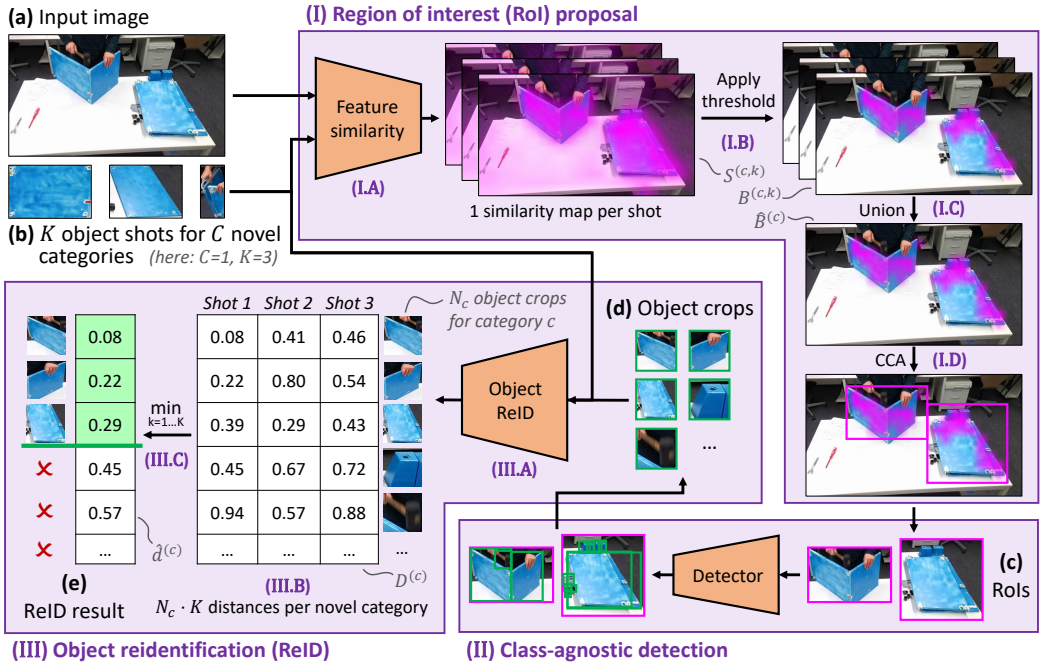


Figure 3. Detailed view of our processing pipeline consisting of three stages, namely (I) regions of interest (RoI) proposal, (II) class-agnostic object detection and (III) object re-identification (ReID). For illustration purposes, we show results for a single novel category ($C = 1$, here: board) represented by $K = 3$ shots. In practice, multiple novel categories can be processed simultaneously. The numbered stages are continuously referenced in the text.

(I) RoI proposal

First, similarity maps for the input image with each of the few shots (Figure 3b) are created by utilizing a feature embedding (Section 3.2.4, Figure 3(I.A)). These similarity maps are then filtered for entries with high similarity scores exceeding a certain threshold to create binary masks (Section 3.2.5, Figure 3(I.B)). For the K shots provided for each category, the union of the K binary masks belonging to the same category is used as binary mask for this category (Figure 3(I.C)). The initial RoIs (Figure 3c) are computed by applying connected component analysis (CCA) to the remaining foreground regions and drawing bounding boxes around these components (Figure 3(I.D)). If necessary, these bounding boxes are further combined and refined (Section 3.2.6) before image crops are extracted and forwarded to the class-agnostic object detector. This entire process is performed on the original HD image (Figure 3a) and three scaled down versions to achieve a degree of scale invariance and, therefore, improve the quality of results for objects of different sizes.

(II) Class-agnostic detection

Each of the final RoIs may contain several objects (see the magenta and green bounding boxes in Figure 3(II)) and, therefore, must be processed further by a class-agnostic detector. Therefore, the RoIs are cropped from the image and supplied to the class-agnostic object detector (Figure 3(II)) as input, which outputs N_c detection bounding boxes for all RoIs belonging to the same category $c \in \{1, 2, \ldots, C\}$ (Section 3.3). Regardless of the image resolution used to create each RoI, crops are always taken from the original image and detection bounding boxes are collected into one set across all resolutions.

(III) Object re-identification

The extracted object crops (Figure 3d) are then processed by the object re-identification (ReID) model (Figure 3(III.A)), introduced in Section 3.4. It computes descriptive feature vectors for all objects and the $K \cdot C$ few-shot images (Figure 3b). Then, the feature-based distance $D^{(c)}$ between the shots and the objects is computed (Figure 3(III.B)). Finally, for each object i, the distances for shots representing the same category are combined

to get a single distance value $\hat{d}_i^{(c)}$ for each object crop with respect to the object category c (Figure 3(III.C)). However, if the distance $\hat{d}_i^{(c)}$ is not smaller than a certain category-specific threshold, no category is assigned and the object proposal is treated as background (Figure 3e). For mathematical details, we refer to Section 3.4.

In the following, each of the three stages, i.e., region of interest (RoI) proposal, class-agnostic detection, and object re-identification (ReID), is described in more detail.

3.2. Region of Interest Proposal

In the first processing stage, regions containing objects belonging to the novel categories to be detected, must be proposed.

3.2.1. Reason Why This Preprocessing Step Is Necessary

The detection of very small objects is of critical importance in the target scenario. For this reason, HD cameras are commonly employed to capture a high degree of detail. In the case of the ATTACH dataset [6], the images have a resolution of 2560 $\times$ 1440 pixels. Unfortunately, the detector must process the full-size image to locate these small objects. Thus, scaling down the images is not a viable option. However, processing the full-size image is a highly computationally expensive process that requires a considerable amount of GPU memory. Instead, sliding windows combined with batch processing at first glance seem to be a potential solution, but this comes with even higher computational costs due to redundancy. Using a naive sliding-window approach to create sub-images for the object detector from a high-resolution image leads to a significant number of crops that must be processed, which is very slow. As a consequence, the class-agnostic detector is applied to a vast number of crops, which in turn leads to a large number of object proposals that must be processed by the ReID model. This further slows down the total processing time.

3.2.2. How to Reduce the Search Space

As an alternative, we propose to apply a feature-based search for regions of interest as a preprocessing step to reduce the computational load. Since the ReID model is able to efficiently compare the representations of the few shots with representations of image regions, we also utilize a model trained in ReID fashion to propose only a few regions of interest (RoI), which potentially contain objects of the novel categories.

Our hypothesis is that the feature embedding of a ReID model can be utilized to compare the few shots with image regions in order to find regions of interest. Therefore, we propose to apply a model trained like a ReID model (see Section 3.4) to the entire image, as described in Section 3.2.3 and shown in Figure 3(I.A). The resulting feature map $F^{(map)}$ contains feature vectors $f_{i,j}^{(map)}$ at each position (i, j). By comparing these feature vectors $f_{i,j}^{(map)}$ with the feature vectors $f^{(query_{c,k})}$ of the K shots provided for the novel object categories $c \in \{1, 2, \ldots, C\}$, we get a similarity map $S_k^{(c)}$, as explained in Section 3.2.4. The similarity map $S_k^{(c)}$ can be interpreted as a heatmap of regions most similar to the k-th shot describing the novel object category c (see Figure 3(I.A)).

From the computed similarity maps, we derive regions of interest (RoIs) (see Sections 3.2.5 and 3.2.6). These RoIs significantly narrow the search space for the class-agnostic object detector and, as a result, improve the processing speed of subsequent processing steps and simultaneously improve the detection precision.

3.2.3. Application of a Re-Identification Model for Feature Similarity Computation

To compute a similarity map $S_k^{(c)}$, we need to compare the feature vector $f^{(query_{c,k})}$ of the k-th shot of category c with the feature map $F^{(map)}$ derived from the full-size image by applying a re-identification (ReID) model.

A ReID model can only be applied in an efficient manner to the full-size image, if it can be utilized as a fully convolutional network. Therefore, it cannot contain non-local

attention blocks or a special type of pooling, such as GeMP [68]. These prerequisites restrict the models we can choose from. Therefore, we decided in favor of a ResNet-50, which, in contrast to the model described in Section 3.4, does not contain any attention blocks and utilizes regular Global Average Pooling (GAP).

We train the ResNet-50 according to the principle of object ReID (see Section 3.4). During inference, the ReID model is applied to the entire HD image $I^{(input)}$ without cropping and to three versions of lower resolution. Moreover, GAP is removed, such that the final feature map $F^{(map)}$ is utilized in its entirety. This allows for a precise localization of the objects represented by the features. Although this ReID model is applied to the entire HD image, it is a time-saving preprocessing step for the overall pipeline due to its reduced computational expense in comparison to the class-agnostic detector.

3.2.4. Computation of Similarity Maps

A similarity map $S_k^{(c)}$ shows the similarity of image regions to the k-th shot of the novel category c based on feature similarity. It is the basic structure to extract regions of interest (RoIs) that need to be processed in the subsequent stages.

To enable the computation of similarity maps, first, the input image $I^{(input)} \in \mathbb{R}^{W \times H \times 3}$ with a spatial resolution of $W \times H$ is processed by the fully convolutional ReID model without GAP $\mathcal{M}_{CNN}$ to compute a feature map $F^{(map)} \in \mathbb{R}^{W' \times H' \times D}$ (see Equation (1)).

$$F^{(map)} = \mathcal{M}_{CNN}\left(I^{(input)}\right) \tag{1}$$

In our case of a ResNet-50 as described for ReID in [69], the feature vectors $f_{i,j}^{(map)} \in \mathbb{R}^D$ at each position (i, j) in the feature map have a dimension of $D = 2048$ and the spatial resolution of the feature map $F^{(map)}$ is $\frac{W}{16} \times \frac{H}{16}$.

Each of the C novel object categories is represented by a set of K shots. Hence, one similarity map must be extracted for each of the $C \cdot K$ shots. Therefore, the images $I^{(query_{c,k})}$ of the $C \cdot K$ shots are processed by the ReID model with GAP $\mathcal{M}_{CNN+GAP}$ to obtain a single feature vector $f^{(query_{c,k})} \in \mathbb{R}^D$ of length $D = 2048$ for each shot (see Equation (2)).

$$f^{(query_{c,k})} = \mathcal{M}_{CNN+GAP}\left(I^{(query_{c,k})}\right) \tag{2}$$

Next, we utilize the cosine similarity to compare the feature vectors $f_{i,j}^{(map)}$ at each position (i, j) in the feature map $F^{(map)}$ with the feature vectors $f^{(query_{c,k})}$ of the K shots provided for the novel object categories $c \in \{1, 2, \ldots, C\}$ as described in Equation (3) to get a similarity map $S^{(c,k)}$ for each of the K shots for each of the C categories.

$$s_{i,j}^{(c,k)} = \cos\left(f^{(query_{c,k})}, f_{i,j}^{(map)}\right) = \frac{f^{(query_{c,k})} \cdot f_{i,j}^{(map)}}{\left\| f^{(query_{c,k})} \right\| \cdot \left\| f_{i,j}^{(map)} \right\|} \tag{3}$$

The similarity map $S^{(c,k)}$ can be interpreted as a heatmap of regions most similar to the k-th shot describing the novel object category c (see Figure 3(I.A)). Therefore, each similarity map $S^{(c,k)}$ allows to determine the matching regions for each of the $C \cdot K$ shots.

3.2.5. Binarization of Similarity Maps

First, we need to binarize the similarity map $S^{(c,k)}$ for each shot in order to be able to search for connected components based on a binary mask $B^{(c,k)}$. Therefore, we apply a shot-specific similarity threshold $\tau_s^{(c,k)}$ at each position (i, j) (see Equation (4)).

$$b_{i,j}^{(c,k)} = \begin{cases} 1 & \text{if } s_{i,j}^{(c,k)} \geq \tau_s^{(c,k)} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Why We Need a Shot-Specific Threshold

To distinguish interesting regions from the background, we tested multiple strategies for binarization, including the use of a static similarity threshold, a fixed quantile of locations and shot-specific dataset statistics. The use of a static threshold proved challenging due to the significant differences in similarity scores for both matching and non-matching regions across different shots. Additionally, the application of a fixed quantile of locations is limited by the inherent property of larger objects occupying more space than smaller ones. The sole viable solution is to utilize dataset statistics to account for the distribution of similarity values of each shot.

How to Use Dataset Statistics to Specify a Shot-Specific Threshold for Binarization

To specify a shot-specific threshold $\tau_s^{(c,k)}$, we use the distribution of similarity scores for each shot by comparing the feature vectors $f^{(query_{c,k})}$ of the shots with background images. Therefore, we prepared a small number M of images $\mathcal{I}^{(bg)} \in \left\{I_1^{(bg)}, I_2^{(bg)}, \ldots, I_M^{(bg)}\right\}$, comprising only the background and otherwise exhibiting the same distribution as the target application. It is reasonable to posit that images of this nature will be available in the majority of real-world assembly scenarios with static camera settings or fixed locations.

The background images $\mathcal{I}^{(bg)}$ are then processed by the fully convolutional ReID model $\mathcal{M}_{CNN}$ (see Equation (5)) to get feature maps $F_m^{(bg)}$, with $m \in \{1, 2, \ldots, M\}$, for the M background images.

$$F_m^{(bg)} = \mathcal{M}_{CNN}\left(I_m^{(bg)}\right) \tag{5}$$

Then, feature vectors $f_{m,i,j}^{(bg)}$ are extracted for each location (i, j) in each feature map $F_m^{(bg)}$, resulting in a set of background feature vectors $\mathcal{F}^{(bg)} \in \left\{f_{m,i,j}^{(bg)} \; \forall \; m, i, j\right\}$.

The set of cosine similarities $\mathcal{S}^{(c,k,bg)} \in \left\{s_{m,i,j}^{(c,k)} \; \forall \; m, i, j\right\}$ between the background feature vectors $\mathcal{F}^{(bg)}$ and the feature vector $f^{(query_{c,k})}$ of the k-th shot of category c (see Equation (6)) is used to calculate an individual similarity threshold value $\tau_s^{(c,k)}$ for each shot.

$$s_{m,i,j}^{(c,k)} = \cos\left(f^{(query_{c,k})}, f_{m,i,j}^{(bg)}\right) = \frac{f^{(query_{c,k})} \cdot f_{m,i,j}^{(bg)}}{\left\|f^{(query_{c,k})}\right\| \cdot \left\|f_{m,i,j}^{(bg)}\right\|} \tag{6}$$

The threshold $\tau_s^{(c,k)}$ is specified as the first p-quantile of this set of similarities $\mathcal{S}^{(c,k,bg)}$ between the feature vector $f^{(query_{c,k})}$ of the k-th shot of category c and the background feature vectors $\mathcal{F}^{(bg)}$ (see Equation (7)), where p is a hyperparameter.

$$\tau_s^{(c,k)} = \text{quantile}\left(\mathcal{S}^{(c,k,bg)}, 1, p\right) \tag{7}$$

The result of applying the threshold is a binary mask $B^{(c,k)}$ of foreground and background areas for each shot (see Figure 3(I.B)). Finally, to prepare the extraction of regions of interest for each category, the union of the k binary masks $B^{(c,k)}$ belonging to the same category c is computed (see Equation (8)).

$$\hat{B}^{(c)} = \bigcup_{k=1}^{K} B^{(c,k)} \tag{8}$$

3.2.6. Region of Interest Creation, Combination and Refinement

The initial regions of interest (RoIs) are created from binary mask $\hat{B}^{(c)}$ for each category c using connected component analysis (CCA). To further refine them, we combine clusters of multiple small boxes in close proximity into one larger box of fixed resolution R_1, illustrated in Figure 4. This offers multiple advantages:

1. Most bounding boxes being the same size allows for more efficient batching during inference with the object-detection model without resizing and padding.
2. Object detectors tend to struggle on very large images and on very small image crops, finding a suitable middle ground is advantageous.
3. Our object detector of choice is DINO [70], implemented in MMDetection [71]. This implementation requires images to be of a certain minimum size to function at all, we found 256 × 256 to be a reliable lower bound.

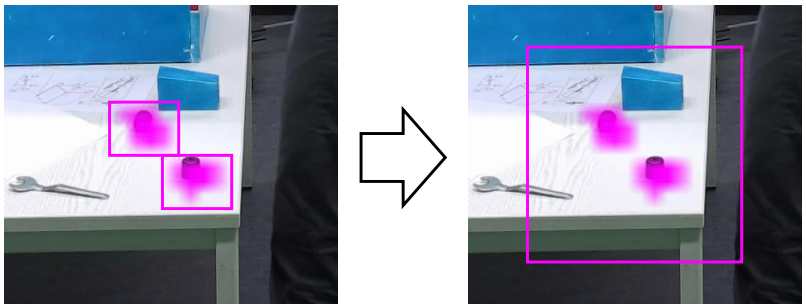


Figure 4. Example for combining multiple small RoIs into a single one, with $R_1 = 256 \times 256$.

Due to the third fact, single small boxes are also enlarged to a resolution R_2 of at least 256 × 256. Additionally, all pairs of boxes, which when framed by one larger bounding box already take up more than a certain percentage threshold τ_b of that boxes area, will be replaced by it. The influence of these hyperparameters is examined in Section 4.3.

3.3. Class-Agnostic Object Detection

In the context of our processing pipeline, the deployment of an object detector is essential for the purpose of extracting cropped images of objects (Figure 3(II)) as a preliminary step in the process of ReID. Given that the target domain objects are unknown during training, the object detector must be class-agnostic.

As evidenced by findings in related work [13,16,17] (see Section 2.1), in order to obtain a powerful class-agnostic object detector, it is crucial to train a detector on a large and densely labeled dataset with many different categories. Therefore, we decided in favor of the LVIS dataset [15] with 1203 densely labeled categories as training dataset. We trained several models as binary object-or-not region proposal networks from scratch, as proposed in [18], including the training pipeline of Dhamija et al. [10], which specifically addresses open-set scenarios. Our experiments in Section 4.1.4 demonstrate that the choice of training dataset is a critical factor in achieving high recall in novel object detection, even outperforming approaches that specifically train detectors for open-set scenarios but using the inferior COCO [14] dataset.

In addition to achieving a high recall on novel categories, the class-agnostic object detector must be capable of minimizing computational demands for subsequent processing steps and be configured to achieve a high inference speed. These two factors are examined in Sections 4.1.5 and 4.1.6.

3.4. Object Re-Identification

After the class-agnostic object detector has generated object proposals, cropped images are extracted for each proposal. The extracted object crops (Figure 3d) are then processed by the object re-identification (ReID) model (Figure 3(III.A)) to get feature vectors $f^{(obj_i)}$ describing the objects. This ReID model also processes the $K \cdot C$ few-shot images (Figure 3b) already used in the first step to also describe them with feature vectors $f^{(query_{c,k})}$. Next, for each of the C novel categories, a distance matrix $D^{(c)} \in \mathbb{R}^{K \times N_c}$ is computed (Equation (9)) between the K few-shot image feature vectors $F^{(query_c)}$ and the feature vectors $F^{(obj)}$ computed for the N_c object crops for category c (Figure 3(III.B)),

$$\text{with } F^{(query_c)} = \left\{ f^{(query_{c,1})}, f^{(query_{c,2})}, \ldots, f^{(query_{c,K})} \right\}$$

$$\text{and} \qquad F^{(obj)} = \left\{ f^{(obj_1)}, f^{(obj_2)}, \ldots, f^{(obj_{N_c})} \right\}.$$

$$d_{k,i}^{(c)} = 1 - \cos\left(f^{(query_{c,k})}, f^{(obj_i)} \right) = 1 - \frac{f^{(query_{c,k})} \cdot f^{(obj_i)}}{\left\| f^{(query_{c,k})} \right\| \cdot \left\| f^{(obj_i)} \right\|} \tag{9}$$

Then, the rows of each distance matrix $D^{(c)}$, corresponding to K queries depicting the same object category c, are combined to create a single distance value $\hat{d}_i^{(c)}$ for each object crop with respect to the object category c (Figure 3e). There are several possible methods to combine these rows. We found that choosing the minimum distance (Figure 3(III.C)) of each of the N_c bounding boxes to any of the K queries (Equation (10)) works well as it prefers the most fitting shot of each category.

$$\hat{d}_i^{(c)} = \min_{k=1}^{K} d_{k,i}^{(c)} \tag{10}$$

However, if the distance $\hat{d}_i^{(c)}$ is not smaller than a certain category-specific threshold $\tau_{ReID}^{(c)}$, no category is assigned and the object proposal is treated as background (Figure 3e). Finding the category-specific threshold $\tau_{ReID}^{(c)}$ is performed analogously to finding a shot-specific threshold for the binarization of similarity maps in the first stage (Figure 3(I.B)), described in Section 3.2.5.

How to Learn Discriminative Embeddings

The key idea behind our approach to provide category information for class-agnostic object proposals is to leverage the ability of discriminative embeddings learned by re-identification (ReID) models to generalize to unknown instances described by a few shots.

In ReID, unique object instances are represented by a number of images each, with each image depicting one and only one object. These images are split into two partitions, gallery and queries. The goal of ReID is to select those gallery images depicting the same object instance as each individual query image. To that end, a backbone model is used to process the images and analogous to image classification the model's final feature map is condensed into a single feature vector describing each image using some global pooling method. The vector space containing these feature vectors is interpreted as an embedding space with the goal being that feature vectors created from images depicting the same instance end up close together under some metric while being distinct from vectors of images depicting different instances. The metric used is usually the cosine similarity over the normalized feature vectors, i.e., their angle (Equation (9)). The model is trained by using an added classification layer that is discarded during inference and further contrastive loss functions. For the purpose of the classification task during training, every object instance in the training dataset is treated as its own class [72]. In order to ensure that the embedding is a bottleneck and thus a condensed representation of the input, it is necessary that ReID datasets contain more object instances than the dimensionality of the embedding [73]. For instance, assuming a typical embedding dimension for a ReID model of 2048, the training dataset must contain sufficiently more than 2048 object instances. Otherwise, the trained ReID model may tend to overfit on the training domain.

Training a general object ReID model comes with a major challenge: ReID pipelines are generally only used to distinguish instances of specific object classes, such as persons or cars, and not arbitrary objects. To solve this challenge, we must provide an adequate training dataset, which allows the ReID model to learn to distinguish instances of arbitrary object classes. Unfortunately, there is a lack of existing object ReID datasets, forcing us to adapt datasets created for another task to our needs. In Section 4.2.1, we provide further details on how we achieved to derive an appropriate dataset from 3D reconstruction datasets.

To train a model capable of general object ReID, we adapt the person ReID pipeline and model from [72] and trained it on the derived dataset. The model is a ResNet-50 incorporating a number of non-local attention blocks [74] and a generalized-mean pool-

ing (GeMP) layer [68]. For this stage, we deliberately chose a more advanced model, since the number of object proposals it is applied to is reduced by previous processing stages, and, thus, does not affect the overall inference speed as much.

4. Experiments

In order to create a powerful pipeline to detect novel objects, represented by a few shots, we propose to use a class-agnostic detector in combination with an object re-identification model. In Section 4.1, we evaluate different models to obtain a suitable class-agnostic detector. Then, we experiment with different training techniques to obtain a powerful object re-identification model in Section 4.2. Finally, we evaluate the entire processing pipeline in the assembly scenario in Section 4.3 and compare it to DE-ViT [8], the currently best few-shot object detector, which does not require fine-tuning on novel categories.

4.1. Class-Agnostic Detection

In order to obtain a suitable class-agnostic object detector, we conducted a series of experiments. In Section 4.1.1, we introduce the out-of-domain detection test dataset, which closely resembles the target scenario. In Section 4.1.2, we describe the evaluation protocol. In Section 4.1.4, we compare different models trained on class-agnostic object detection and derive which architectures are the most promising candidates. Based on these, we evaluate additional factors that influence the processing time of our pipeline: We examine the number of predicted proposals (Section 4.1.5), as well as the model size, input size, and batch size (Section 4.1.6).

4.1.1. Out-of-Domain Detection Test Dataset

For evaluating the capability of the class-agnostic detector in the target scenario, we choose the ATTACH dataset [6]. The ATTACH dataset was proposed for human action recognition and, therefore, provides only annotations for extracted human skeletons and the actions performed. Therefore, we generated additional object annotations as described below. This dataset shows scenes of persons assembling a small piece of furniture. From a static camera configuration, the working place is observed from three perspectives. The objects include tools, the assembly instruction, parts of the furniture, and many small objects, such as screws. Only the tools hammer, screwdriver, and wrench are known categories in LVIS, which is the training dataset for the class-agnostic object detector. All other objects are from novel categories that are not contained in the training dataset.

Object Annotations

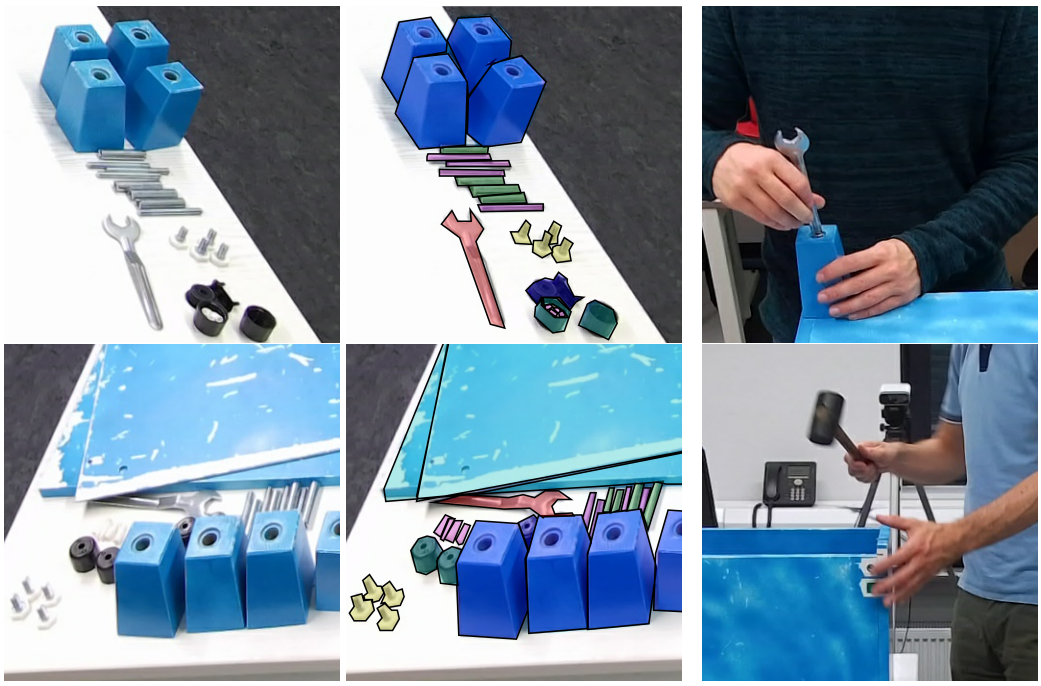
A total of 2401 images from 30 videos in the ATTACH database were selected for annotation. These images incorporated all three camera views and 30 different individuals performing the assembly. The extracted images are selected at 300-image intervals, or one image every ten seconds, in order to enhance the variability. Additionally, some scenes displaying the utilization of tools are labeled at a frequency of one image per second. The object categories have been selected in accordance with the action labels included in the ATTACH dataset, which encompasses 12 different categories. Therefore, all objects associated with the assembly are annotated. Each object is annotated with a polygon using the Labelme tool. All annotations, along with the HD images, are publicly available at (https://www.tu-ilmenau.de/neurob/data-sets-code/object-reid).

These annotations were primarily created for training human action-recognition approaches that include objects. In the context of object detection, the annotated images are highly redundant: Although only one frame is annotated every ten seconds, the majority of objects remain stationary on the table since they are not in use. Consequently, further selection of distinct images was required in order to obtain meaningful benchmarks, as described below.

Benchmarks for Detection of Novel Objects

For our experiments, we selected images for two benchmarks, namely the table benchmark and the in-hand benchmark. We made sure to include as many different perspectives, occlusions, and configurations of objects as possible. Since the assemblies were guided by building instructions, many scenes are very similar. In order to avoid high redundancy, we had to reduce the number of benchmark images considerably.

The table benchmark consists of 38 crops of size 350 × 350 from 20 of the annotated images of the ATTACH dataset showing many small objects and tools lying on a table. The in-hand benchmark consists of 40 crops of size 350 × 350 from 20 of the annotated images of the ATTACH dataset where the assembling persons are using a tool in their hands. Figure 5 shows sample images of both benchmarks.



(**a**) Sample images of table benchmark and labeled version (**b**) In-hand benchmark

Figure 5. Sample images for class-agnostic object detector benchmarks. These images are part of the ATTACH dataset [6] for human action recognition. For benchmarking the object detectors, we annotated the objects in the scene associated with the assembly.

Assuming that LVIS was used as the training dataset, novel categories account for 93.3% of annotated instances (541 out of 580) in the table benchmark. Only 6.7% of the annotated instances show objects of categories included in LVIS, as listed above. Therefore, we use the table benchmark as the primary benchmark to compare the capabilities of different class-agnostic detection models to generate proposals for novel objects. Later on, in Section 4.3, the table benchmark is also used to evaluate the entire detection pipeline including RoI proposasl and object ReID. The in-hand benchmark primarily shows whether the objects can be detected in case of motion blur, which we identified as a challenging task.

4.1.2. Evaluation Protocol

We allow each detector to predict a limited number of object proposals and measure the recall, which counts the number of labeled objects that are included in the proposals, using the common intersection-over-union threshold of 0.5 as bounding box overlap measure. Therefore, to maximize the recall, within the permitted number of proposals, the class-agnostic detector needs to predict fitting bounding boxes for as many ground truth objects as possible. Note that the recall measure does not account for false positives, i.e., it does not decrease as more bounding boxes are predicted for other potential unlabeled objects.

In our comparison of models for class-agnostic object detection, we included EMSANet [75], Segment Anything Model (SAM) incorporating a vision transformer [23], RTMDet [76], DINO v1 incorporating a ResNet-50 model and a transformer [70], DETR [77], Deformable DETR [78], as well as Faster R-CNN with two training pipelines that specifically address the open-set scenario [10,12].

4.1.3. Implementation Details and Model Training

By utilizing the MMDetection framework [71], we trained DINO v1 [70] and RTMDet [76] on the LVIS dataset [15] for 20 epochs on 4 A100 GPUs using a batch size of 16, the AdamW optimizer [79], and a learning rate schedule that decreased the learning rate by a factor of 10 after 11 epochs. As data augmentation, we used random flip, resize, and crop. Faster R-CNN was trained using the publicly available code of [10], which specifically addresses the open-set scenarios and is implemented in Detectron2 [80]. For EMSANet [75], we also used the publicly available code.

For the best-performing DINO v1 [70] model, we used the following hyperparameters: The learning rate was 0.00001 for the backbone and 0.0001 for the remaining network. The weight decay was 0.0001.

The results for SAM [23] and all models trained on COCO were produced by using publicly available trained netwoks.

For further implementation details we refer to the source code that is publicly available (https://www.tu-ilmenau.de/neurob/data-sets-code/object-reid).

4.1.4. Model Comparison

Figure 6 shows the achieved recall on the table benchmark and the inference speed on an A100-PCIE-40GB GPU for each model. The inference speed was measured using the MMDet inference measuring tool whenever applicable, in order to time the optimized computation of models. For EMSANet and the Segment Anything Model (SAM), we needed to measure the optimized ONNX models. In order to determine the comparability of the two variants of inference measurement, we also measured the inference speed of Faster R-CNN, DINO, and RTMDet based on optimized ONNX models. We obtained similar inference speeds, with a maximum deviation of ±5 FPS.

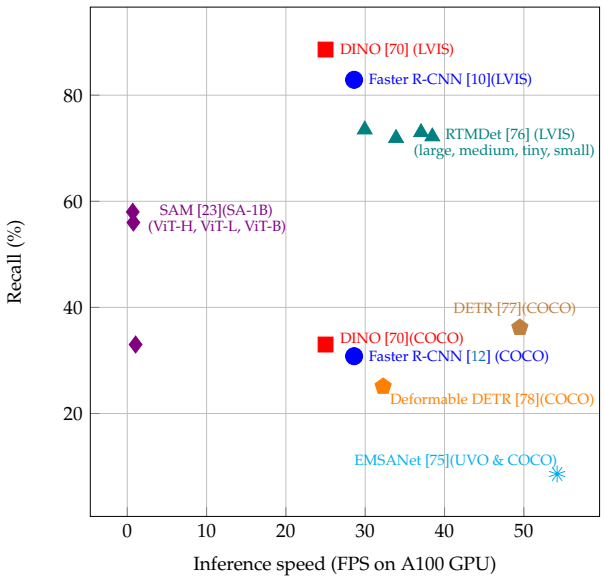


Figure 6. Model comparison for class-agnostic detection on the table benchmark. Shown is the novel category-detection capability in terms of recall vs. the inference speed ([10,23,70,75–78]).

The training dataset for each model is listed in parentheses. The limit of permitted proposal outputs was set to 1000 for each detector. The images were resized to a resolution

of 640 × 640. To produce the results of detectors trained on COCO, we used trained networks provided by the MMDetection framework or the public repositories of the authors, respectively. For training on LVIS and on COCO combined with UFO, we used the provided training pipeline by MMDetection, Detectron, or the GitHub repositories, but changed the training dataset.

All models trained on the densely labeled LVIS dataset achieve a significantly larger recall than the models trained on other datasets. Note that while LVIS includes many categories, only three of them overlap with categories in the test dataset ATTACH, and they account for only 6.7% of annotations in the table benchmark. COCO categories do not overlap with ATTACH categories. Since most of the categories are novel regardless of the training dataset, the huge performance gap between detectors trained on LVIS and detectors trained on other datasets cannot be explained by known categories, but is the result of a better generalization. For the models Faster R-CNN with open-set training pipeline and DINO, Figure 6 lists both the performance when trained on LVIS and the results from the provided checkpoint from the MMDetection framework, where they were trained on COCO. For these two models, the significance of the choice of training dataset becomes clear. Note that we were not able to achieve these good results when training DETR and Deformable DETR on LVIS. The results did not surpass the respective COCO baselines. We hypothesise that their more complicated training pipeline prevented them from benefiting from the larger and more densely labeled training dataset. The Segment Anything Model (SAM) is trained on the large SA-1B dataset. The larger SAMs achieve a significantly higher recall than the models trained on COCO, but cannot reach the performance of models trained on LVIS.

For the instance segmentation models Segment Anything Model (SAM) and EMSANet, we observed that these models outputted only a few segmentation masks, which we used for bounding box extraction. They are not able to provide several potential bounding boxes for a specific image region, and, thus, are unable to predict bounding boxes for both an object and its parts. In contrast, object detectors provide several overlapping proposals, which enables them to predict a greater number of bounding boxes that describe an object on several levels of detail. This is beneficial in cases where the category knowledge is not present beforehand. Furthermore, the instance segmentation models performed relatively poorly on small objects, either by missing them or not distinguishing the instances of nearby similar-appearing objects. Since small objects make up a large proportion of labeled instances in the table benchmark, the recall is significantly influenced by missing small objects.

Regarding inference speed, SAM is clearly performing worse due to the ViT it is using as backbone. All the other approaches use a ConvNet or a ConvNet in combination with a transformer and are, therefore, able to process between 25 and 55 frames of size 640 × 640 per second, with EMSANet and DETR being the fastest. Among the models that achieve the highest recall, RTMDet is fastest, followed by Faster R-CNN and DINO.

Comparable, if not slightly superior, results were observed in the in-hand benchmark due to the absence of small objects, although the appearance of tools held in the hand was noticeably affected by motion blur. In 80% of cases, a bounding box was predicted for the tool held in the hand, which was either a hammer, a wrench, or a screwdriver. In instances where the tool was grasped in the middle, the tool was represented by two bounding boxes, which is correct. However, this makes re-identification more challenging. In some instances of motion blur, the hammer is represented by two bounding boxes for the handle and head. However, the bounding box for the entire object is missing. Notwithstanding the aforementioned findings, the class-agnostic object detectors that performed best in the table benchmark are also capable of reliably detecting tools held in hand in the presence of motion blur.

4.1.5. Number of Predicted Object Proposals

The number of predicted object proposals exerts a linear influence on the processing time of the subsequent re-identification stage and will, therefore, slow down the overall pipeline in case of many predicted proposals.

Therefore, we also need to take into account whether the class-agnostic detectors are able to achieve a high recall with a smaller amount of permitted proposals. We observed that Faster R-CNN [10], DINO [70], and RTMDet [76] exhibited comparable performance in terms of recall for novel object detection and inference speed when permitted to generate a substantial number of object proposals. Table 1 shows the recall for different numbers of permitted proposals for the two best-performing approaches.

Table 1. Influence of permitted proposals for class-agnostic detection performance on the table benchmark. The training dataset is listed in parentheses. The input size was 640 × 640.

Approach	Permitted Proposals	Recall [%]
DINO [70] (LVIS)	1000	88.6
DINO [70] (LVIS)	500	86.8
DINO [70] (LVIS)	300	85.4
DINO [70] (LVIS)	100	78.4
DINO [70] (LVIS)	50	70.5
Faster R-CNN [10] (LVIS)	1000	82.9
Faster R-CNN [10] (LVIS)	500	77.8
Faster R-CNN [10] (LVIS)	300	67.0
Faster R-CNN [10] (LVIS)	100	34.9
Faster R-CNN [10] (LVIS)	50	21.1

Faster R-CNN performs similar to DINO when it is allowed to predict many proposals. When the number of object proposals is reduced based on the predicted confidence score, DINO is performing better. The 100 most confident predictions of DINO achieve a similar recall as the 500 most confident predictions of Faster R-CNN. When both detectors are permitted to predict only 100 proposals, DINO clearly outperforms Faster R-CNN. Therefore, DINO is able to predict much more reliable confidence scores than Faster R-CNN. We hypothesize that this is achieved by the transformer component included in DINO, which is not contained in the pure ConvNet option Faster R-CNN. Thus, the DINO-based class-agnostic object detector is a very good fit for our pipeline, such that we can operate with a more restricted number of object proposals for the subsequential re-identification stage.

4.1.6. Ablation Study: Other Factors Affecting Performance

Besides the choice of training dataset and model, the class-agnostic detection performance is also affected by some other factors, which we address in the following.

Model size: For two detectors, namely RTMDet and SAM, Figure 6 shows the performance for different model sizes. While for RTMDet all models, regardless of the size, achieve a similar recall, for SAM, we can see a significant drop in recall when the model becomes too small. Therefore, the influence of model size depends on the type of network. ConvNets (RTMDet) seem to depend less on model capacity than vision transformers (SAM), which is consistent with findings in vision transformer literature [47].

Input size: Table 2 shows the results of DINO and Faster R-CNN for different input sizes. For DINO, doubling the input size from 320 × 320 to 640 × 640 increases the recall by only 6.7 percentage points but slows down the processing by a factor of 4. Therefore, the input size for DINO can be chosen rather small to achieve a good tradeoff. In the case of Faster R-CNN, we observed a larger performance gap for input sizes below 450 × 450, which hinders the use of smaller inputs in order to speed up the processing pipeline. Therefore, the superior performance on smaller input sizes is another argument in favor of DINO over Faster R-CNN.

Table 2. Influence of input size for class-agnostic detection performance on the table benchmark. The training dataset is listed in parentheses. The number of permitted proposals was set to 1000.

Approach	Input Size	Recall [%]
DINO [70] (LVIS)	640 × 640	88.6
DINO [70] (LVIS)	450 × 450	85.1
DINO [70] (LVIS)	320 × 320	81.9
DINO [70] (LVIS)	270 × 270	75.1
DINO [70] (LVIS)	224 × 224	60.8
Faster R-CNN [10] (LVIS)	640 × 640	82.9
Faster R-CNN [10] (LVIS)	450 × 450	82.3
Faster R-CNN [10] (LVIS)	320 × 320	69.4
Faster R-CNN [10] (LVIS)	270 × 270	60.3
Faster R-CNN [10] (LVIS)	224 × 224	48.7

Batch size: The batch size also affects the inference speed. The HD image can be partitioned into several patches and then processed as a batch. We found that on an NVIDIA A100-PCIE-40GB GPU, we achieve the best speed with a batch size of 128. Batch sizes up to 4000 can be processed with similar speed. Above that, the processing significantly slows down.

Conclusion: The inference time of the overall processing pipeline is significantly influenced by the class-agnostic object detector, and can be accelerated by smaller models, lower input resolution, fewer proposals, and optimized batch size at the expense of novel-category-detection performance in terms of recall. Therefore, in consideration of the aforementioned findings, to obtain a good trade-off between inference speed and recall, the image size and number of proposals can be chosen rather low for the class-agnostic object detector in our pipeline (see Section 4.3).

4.2. Object ReID

In order to obtain a capable object ReID model, we conducted a series of experiments. First, we introduce our training dataset in Section 4.2.1. Next, we introduce several out-of-domain (OOD) test datasets in Section 4.2.2, which we use to evaluate our novel object ReID method. In Section 4.2.3, we describe our model and the training process in detail. Then, we perform experiments on our OOD test datasets in Section 4.2.4 and compare our model with the state-of-the-art content-based image retrieval (CBIR) model SuperGlobal [66]. Finally, we perform a series of ablation studies to better understand the behavior of object ReID compared to person ReID in Section 4.2.5.

4.2.1. Training Dataset

The main challenge in training a general object ReID model is the lack of existing object ReID datasets, forcing us to adapt datasets created for another task to our needs. A dataset suitable for ReID must necessarily fulfill two requirements:

1. Each object must be present in multiple different images.
2. Objects must be labeled in a way that allows for their unique identities to be identified, i.e., object-category-level labels used in object detection and classification are insufficient.

Additionally, as mentioned in Section 2.1, datasets with many diverse categories improve a trained model's ability to generalize to novel categories [16,17].

A good fit for our mandatory requirements are 3D reconstruction datasets. These datasets are usually composed of videos, each showcasing a single unique object instance from multiple perspectives. By sampling a number of frames from every video and creating crops only containing the recorded object, a ReID dataset can be constructed. We initially identified a number of potential dataset candidates: Common Objects in 3D (CO3D) [81], Redwood 3D Scan [82], Google scanned objects [83], Objectron [84] and AMT Objects [85]. Google scanned objects, Objectron, and AMT Objects were excluded as training datasets early on, since Google scanned objects only contains 1030 different object instances with

5 images each and the latter contain only 9 and 7 object categories, respectively. In contrast, CO3D contains 18,619 instances belonging to 50 categories included in COCO [14]. Therefore, the authors were able to apply a common object detector trained on COCO to create object segmentation masks. Bounding boxes and segmentation masks created with Pointrend [86] are provided. The Redwood 3D Scan dataset [82] contains 10,921 instances belonging to 320 categories, but, unlike CO3D, determining the correct image crops containing the recorded objects proved difficult in our experiments, since the object categories are unconstrained. Therefore, we decided in favor of CO3D [81] as our main dataset for training.

To create a ReID dataset from CO3D, we sampled at most 20 random frames from every video, thus creating a dataset with 329,846 images. We split the dataset similarly to the well-known Market-1501 person ReID dataset [87]: Instances of most object categories are split 50/50 into train and validation partitions. The only exceptions are the four categories banana, kite, parkingmeter and skateboard, which are included exclusively in the validation partition to better evaluate a trained model's ability to generalize to novel object categories. The images of each instance from the validation partition were further randomly split 80/20 into gallery and query partitions. The resulting ReID dataset contains 161,255 training images, 134,681 gallery images, and 33,910 query images. The prepared dataset is publicly available (https://www.tu-ilmenau.de/neurob/data-sets-code/object-reid).

4.2.2. Out-of-Domain ReID Test Datasets

In addition to our training and validation dataset, we also chose to create multiple out-of-domain (OOD) test datasets. We use Google scanned objects [83], which is insufficient for training, but provides a huge span of object categories not contained in the training dataset CO3D. Furthermore, we selected the tool datasets ATTACH [6], KTH-Handtools [88], Working-Hands [89] and OHO [90]. These datasets are introduced below.

Our main test dataset for ReID, called the OOD tool dataset, was created to evaluate our approach in the domain of human–robot collaboration in assembly. We utilized three object-detection datasets containing hand tools, e.g., hammers, screwdrivers, or pencils, in its creation: ATTACH, KTH-Handtools, and Working-Hands. The small size of these datasets allows for manual identification of object instances, which can be uniquely identified in multiple images and thus used for ReID. The ReID dataset was created by sampling images of each object instance and splitting them 80/20 into gallery and query partitions. Since the test datasets are only used to evaluate the cross-domain generalization abilities of trained object ReID models, no train partition is necessary. The OOD tool dataset contains 23 instances from 13 different tool-categories, 276 gallery images, and 69 query images. Unfortunately, this dataset is quite small. Therefore, we increased the difficulty by extending the gallery with 85,746 images from the Redwood 3D Scan dataset [82] serving as disturbance.

Additionally, we created ReID datasets from Google scanned objects [83] and OHO [90], which display objects in front of a white or empty background, respectively. These datasets were primarily used to evaluate the model's ability to ignore inconsequential background content. Google scanned objects contains 1030 object instances, 4120 gallery images, and 1030 query images. For OHO, we included all object categories that can be considered tools or work pieces, leading to a dataset containing 27 instances, 432 gallery images and 108 query images.

4.2.3. Implementation Details and Model Training

We trained our model for 12 epochs with the entire CO3D training dataset being processed once during each epoch. For the learning rate, we settled on 7×10^{-5} with a linear warmup of one epoch and a learning rate reduction by a factor of 10 after 6 and 9 epochs each. Our optimizer of choice is Adam [91] with default parameters. As loss functions, we utilized the weighted regularization triplet (WRT) loss [72] and softmax cross-entropy loss with label smoothing. The dimension of the output embedding is $D = 2048$ and the

metric used to calculate the distance between its feature vectors is the cosine similarity. Since object bounding boxes in CO3D are on average close to square shaped, we decided to resize all processed input images to a constant resolution of 128 × 128 pixels, instead of the 128 × 256 commonly used in person ReID. We used the data-augmentation methods of random horizontal flip and random cropping to resolution 128 × 128 after padding the image by 10 pixels in each direction. A higher resolution, e.g., 192 × 192, can lead to improvements of the model's ReID capabilities but will significantly impact its inference speed. Our default batch size during training was 64, and we used PK-sampling [92] as our batch sampling strategy with 4 images per object instance and, therefore, 16 different object instances.

For further implementation details we refer to the source code that is publicly available (https://www.tu-ilmenau.de/neurob/data-sets-code/object-reid).

4.2.4. Evaluation and Comparison

Since no object ReID models exist so far, we chose to compare our approach with the SuperGlobal [66] model from the field of contend-based image retrieval (CBIR). We used SuperGlobal as provided by the authors, who adopted their method for CVNet [67] with a ResNet-50 backbone trained on the clean subset [93] of the Google Landmarks dataset v2 [65] and tuned their pooling parameters on $\mathcal{R}$Oxford [64]. Our ReID model was adapted from [72] and trained on CO3D [81].

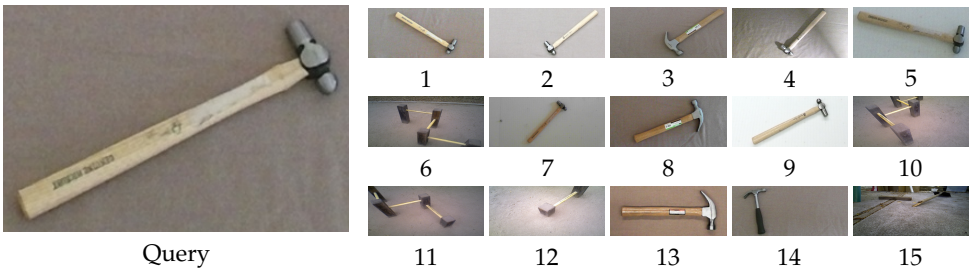
For both methods, we report the retrieval mean average precision (mAP), which is a common metric both in CBIR and ReID. The results are shown in Table 3.

Table 3. Tool and object image retrieval rankings compared. We list both the results on the respective in-domain validation datasets and three out-of-domain test datasets. The input resolution in this experiment was 192 × 192.

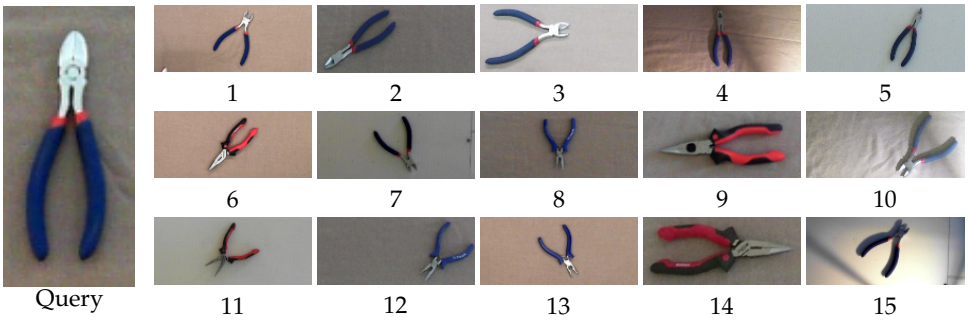
Validation/Test Dataset	mAP SuperGlobal [66] (CBIR)	mAP Proposed (ReID)
CO3D validation set [81] (in-domain ReID)	-	89.4
$\mathcal{R}$Oxford+1M (Medium) [64] (in-domain CBIR)	80.0	-
$\mathcal{R}$Paris+1M (Medium) [64] (in-domain CBIR)	83.4	-
OHO [90]	35.9	58.5
Google scanned objects [83]	55.4	64.0
OOD tool dataset (ours, composed of [6,82,88,89])	33.4	48.1

When evaluated on the validation set of CO3D, our model's performance is excellent. However, just like with person ReID, there exists a large performance gap between the training domain and other OOD datasets. Nevertheless, our model still performs significantly better than SuperGlobal, proving the efficacy of our training pipeline and dataset.

For qualitative results, two example rankings produced by our model on the OOD tool dataset are illustrated in Figure 7. In Figure 7a, the 15 images ranked highest in their similarity to the query image depicting a hammer out of a gallery of 85,746 images are displayed. Out of 12 possible correct matches present in the dataset, 6 are ranked in the top 15, specifically on ranks 1, 2, 4, 5, 7, and 9. The error cases display different hammers and a similar looking playground installation. In Figure 7b, the query image depicts a pair of diagonal cutting pliers. Out of 12 possible correct matches present in the dataset, 8 are ranked in the top 15, specifically on ranks 1, 2, 3, 4, 5, 7, 10, and 15. The error cases display different diagonal cutting pliers. For both queries, the gallery images that were not included in the top 15 were captured under conditions of extreme lighting, which significantly affects their appearance.



(**a**) Ranking for a query image of the OOD tool dataset depicting a hammer.



(**b**) Ranking for a query image of the OOD tool dataset depicting a pair of diagonal cutting pliers.

Figure 7. Exemplary rankings of the proposed object re-identification model applied to out-of-domain data.

4.2.5. Influence Factors on the Re-Identification Pipeline

We performed a series of additional experiments, which were inspired by the state of the art of person ReID, or aimed to further examine unique aspects of our new dataset or object ReID in general.

Data Augmentation

We examined a number of data-augmentation techniques during training to determine whether they are beneficial for object ReID. Random horizontal flipping and random cropping are two fundamental augmentation techniques of person ReID. Random erasing is also common in person ReID but has been proven to harm cross-domain performance [94,95]. Additionally, we examined random vertical flipping, since, unlike persons, many of our test categories still produce sensible images when flipped upside down. We trained on CO3D and evaluated on the proposed OOD tool dataset. The results are displayed in Table 4.

Table 4. Results of data-augmentation experiments for object re-identification on OOD tool dataset using an input resolution of 128 × 128 using CPK-sampling. The best result is highlighted in bold.

Horizontal Flip	Vertical Flip	Random Crop	Random Erasing	mAP
-	-	-	-	45.0
-	-	-	✓	41.0
✓	-	-	-	**47.3**
-	✓	-	-	46.9
-	-	✓	-	45.5
✓	✓	-	-	46.8
-	✓	✓	-	46.9
✓	-	✓	-	46.5
✓	✓	✓	-	46.9

We have found that all techniques lead to improvements, except for random erasing, which causes a large drop in performance. Note that combining multiple techniques might not always lead to improvements over applying them individually.

Input Resolution

We examined different input resolutions used for the ReID model, trained on CO3D and evaluated on the OOD tool dataset. The results are displayed in Table 5.

Table 5. Results of experiments regarding input resolution for object re-identification. Models were trained using PK-sampling. The best result is highlighted in bold.

Resolution	**128 × 128**	**192 × 192**	**256 × 256**
mAP (OOD tool dataset)	46.0	**48.1**	46.9

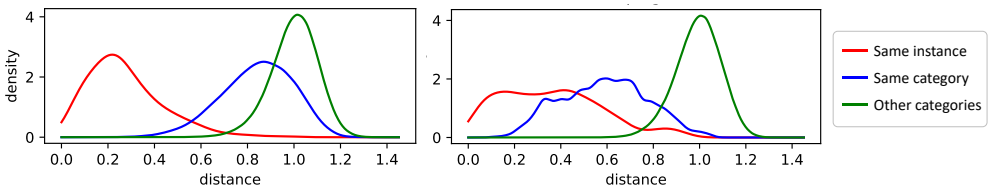
We have found that while increasing the input resolution can enhance ReID performance to a certain extent, it also results in a considerable increase in inference time. For higher resolutions (256 × 256 and above), we hypothesise that the blur due to up-scaling of small objects is harming the re-identification performance. We suggest using a resolution of 192 × 192 only if the objective is to achieve a very high ReID precision. In our pipeline, we utilize a resolution of 128 × 128, given that the ReID modules in the initial and final stages of our pipeline collectively account for 40–50% of the total processing time. This underscores the importance of inference speed.

Embedding Behavior and Dataset Quality

Since object ReID introduces object-category level labels in addition to instance level ones, we decided to examine the behaviour of different object categories within the embedding space. To that end, we trained a model on CO3D and used it to process the CO3D gallery and query partitions. We examined the distribution of cosine distances (Equation (9)) between query feature vectors and gallery feature vectors grouped according to their object category. We analyzed, how similar query images belonging to a certain category are to gallery images belonging to the same object instance, to the same category excluding that instance, and to all other categories, respectively. To estimate those distributions, we used kernel density estimation (KDE). Exemplary probability density functions for two categories are displayed in Figure 8.

Our investigations indicate that, in general, objects are perceived as most similar to images belonging to the same instance and as more similar to objects of the same category than to objects belonging to other categories. This aligns well with human intuition.

Additionally, we noticed that certain categories display a much larger overlap between the distribution of distances to samples of the same instance and the distribution of distances to samples of the same category. Figure 8b displays such an example. Instances of the category backpack (Figure 8a) have a much smaller overlap between distances to the same instance and distances to others than stop signs. This signifies that stop signs are much harder to distinguish from each other than backpacks, leading to mistakes during ReID.



(**a**) Category backpack (**b**) Category stop sign

Figure 8. Distribution of cosine distances of queries to gallery of CO3D validation split. A distance of 1.0 indicates an angle of 90°, which is the expected angle between two random vectors in high-dimensional space.

This discrepancy is in some cases, e.g., stop signs, explained by the fact that instances of these categories are naturally very difficult to distinguish from each other, even for humans. In other cases, it leads to the discovery of errors in the labels of our training

dataset. The most prominent case in the CO3D dataset for this is the category car. Images of cars recorded in parking lots usually contain additional cars present in the background. These cars are regularly picked up by the object detector used to create the bounding boxes for objects, instead of the car being recorded, leading to incorrect labels. This issue can be alleviated by implementing a sanity check for sampled images during dataset creation. Since all images of an object instance are sampled from a single video, errors can be detected by observing big jumps of bounding boxes in consecutive frames. Consequently, when creating the CO3D dataset, we sampled only from sequences of frames containing no such bounding box jumps. Furthermore, the sequence must consist of at least 20 consecutive frames to reduce the chance of sampling from a background object that was detected for multiple frames.

Embedding Dimension

Per default, the embedding space of our employed model possesses a dimensionality of 2048, which is appropriate when trained with 18,619 instances in the training dataset CO3D. However, in case of a smaller training dataset, employing an embedding with a smaller size might be preferable. If the number of instances in the dataset is smaller than the dimensionality of the embedding, the re-identification problem could theoretically be trivialized during training by assigning each instance one dimension of the embedding space, collapsing the learned embedding to the subspace made up of the coordinate axes, which represents an extreme case of overfitting. Although a full collapse of the embedding is unlikely in practice, in the case of person ReID a performance drop of multiple percentage points in mAP can be observed [73].

Since the utilized CO3D dataset contains 18,619 instances, we investigated this problem in the context of object ReID by creating a dataset from CO3D by removing 90% of all instances per category. The resulting dataset contains 1734 remaining instances, ca. half of which are present in the train split. We reduced the size of the embedding produced by the object ReID model by extending a single linear layer after the global pooling step to reduce the size of the output feature vector. We trained models with different embedding sizes on the reduced dataset and evaluated on the OOD tool dataset. The results are displayed in Table 6.

Table 6. Results of experiments regarding embedding size for object re-identification. Note that, for this experiment, the training dataset size was reduced to 10%. Therefore, the out-of-domain results are much lower than in Table 3. The best results are highlighted in bold.

Embedding Size	**2048**	**1024**	**256**
mAP validation (CO3D dataset)	**89.3**	88.0	79.1
mAP test (OOD tool dataset)	38.1	**40.7**	33.1

The largest embedding of 2048 has been found to produce the best results on the in-domain validation dataset. Nevertheless, reducing the number of dimensions to 1024 resulted in enhanced performance on the out-of-domain test dataset. Further reduction of the embedding size to 256 results in a significant decline in performance for both domains. Thus, reducing the dimensionality of the embedding to a value below the number of identities in the training dataset can enhance the generalization to other domains. However, this reduction also limits the representation capability of the ReID model. Therefore, an optimal trade-off must be identified.

Batch Sampling

The PK sampling traditionally used in person ReID during training, composes each training batch of size B by first sampling P identities and then sampling K images for each identity with $B = P \cdot K$. Since object ReID introduces object-category level labels in addition to instance level ones, we additionally investigated the influence of the object category membership of instances during sampling. To that end, we extend the PK-sampling scheme

to also consider object categories (referred to as CPK-sampling): First, sample C object categories, then P identities per category, and finally K images for each identity with $B = C \cdot P \cdot K$. For our experiments, we set $K = 4$ and $B = 64$, while varying C and P, with $C \cdot P = 16$. We trained on CO3D and evaluated on the OOD tool dataset. The results are displayed in Table 7.

Table 7. Results of sampling experiments for object re-identification. The input resolution was 128 × 128. The best result is highlighted in bold.

C, P	**1, 16**	**4, 4**	**8, 2**	**16, 1**
mAP (OOD tool dataset)	43.3	44.7	45.8	**46.0**

Maximizing the number of categories in a batch while decreasing the number of identities per category yields the best results using this strategy. Nevertheless, these results are still below those achieved with regular PK sampling (mAP = 47.3 for an input resolution of 128 × 128), during which category membership is random and C therefore varies.

In light of the results in Table 7, we hypothesize that the optimal number of categories C present in a batch is likely to be between 8 and 16. The expected value of the number of categories with regular PK sampling over CO3D is $C = 13$, which falls within the aforementioned interval. Some randomness seems to be beneficial.

Number of Categories in the Training Dataset

When creating our training dataset for object ReID as described in Section 4.2.1, we operated under the assumption that datasets with many diverse categories improve the generalization ability of a trained model to novel categories. To verify this claim, in the context of object ReID, we trained models on three different subsets of the CO3D training dataset named $CO3D_{100}$, $CO3D_{75}$, and $CO3D_{50}$. To create $CO3D_{100}$, we removed 50% of all instances for each category in CO3D, ending up with a dataset of half its size. To create $CO3D_{75}$ from $CO3D_{100}$, we removed 25% of its categories and restored instances of remaining categories until the datasets reached the same size. $CO3D_{50}$ was created analogously from $CO3D_{75}$. We ended up with three datasets of the same size containing 100%, 75%, and 50% of the categories of CO3D, respectively. All datasets contain the same number of instances. We trained on these datasets and evaluated on the OOD tool dataset. The results are displayed in Table 8.

Table 8. Results of experiments regarding the number of categories in training datasets for object re-identification. Note that, for this experiment, the training dataset size was reduced to 50%. Therefore, the out-of-domain results are somewhat lower than in Table 3. The best result is highlighted in bold.

Training Dataset	**$CO3D_{100}$**	**$CO3D_{75}$**	**$CO3D_{50}$**
mAP (OOD tool dataset)	**44.5**	44.4	43.0

Our initial assumption was proven correct, indicating that, under equivalent conditions, category variety plays a more important role in dataset choice than size alone. Nevertheless, none of the trained models achieved the same performance as one trained on the original CO3D dataset, which contains twice as many images. This shows that dataset size remains an important factor.

Image Backgrounds

Since all images belonging to a single object instance in CO3D originate from the same video, there can be some overlap in the background shown in these images. To check, whether the model incorrectly learns to use the image background during training, we created an alternative version of the CO3D dataset by using the segmentation masks included in the original dataset to remove the background from all images. We trained

models on both this modified dataset and its vanilla version. We evaluated these models on the OOD tool dataset, on the OHO dataset, and on the Google scanned objects dataset. The latter two datasets do not contain any image background. Our experiments in this regard proved inconclusive, and we were unable to determine the extent to which the models focus on the background during inference.

Qualitative examinations indicate that the model has a strong preference for color and can make mistakes with drab objects in front of colorful background.

Modern Loss Functions

Two of the most promising loss functions in the state of the art of person ReID are Additive Angular Margin Loss (AAML) [96] and Circle Loss [97]. We adapted both loss functions for the purpose of object ReID by using them during training on CO3D instead of softmax cross-entropy and, in the case of Circle Loss, triplet loss. We evaluated the trained models on the OOD tool dataset. We have found that the numerous hyperparameters of both loss functions have a large influence on the performance on the test dataset. However, we could not identify clear tendencies for hyperparameter choice. Even small changes led to large changes in performance, which were not reflected on the validation dataset during training. As these hyperparameters seem to depend highly on the training domain, we cannot recommend Circle Loss or AAML for practical application if a target domain dataset is not available for hyperparameter tuning.

Conclusion

We identified a variety of influence factors for object ReID, regarding dataset, model architecture, and training procedure: Creating a ReID dataset from a 3D reconstruction dataset requires care, since the processing steps using an object detector are prone to mistakes. Training datasets should contain many varied categories to aid the trained model's generalization capabilities, but dataset size still remains an important factor. The dimensionality of the model's output embedding should be chosen smaller than the number of instances in the training dataset to not hurt OOD performance. Increasing the model's input resolution to 192 $\times$ 192 will increase ReID precision, but we cannot afford the accompanying increase in inference time for our application and, therefore, keep using a resolution of 128 $\times$ 128. Data-augmentation techniques common in person ReID, like horizontal flipping and random cropping, are beneficial in object ReID, too. Specifically for our tool categories, vertical flipping is also useful. The batch sampling technique PK-sampling commonly employed in person ReID also proved well suited for object ReID.

4.3. Entire Processing Pipeline

After examining class-agnostic object detectors and object ReID models individually, we conducted experiments using our entire pipeline end-to-end. The purpose of the following experiments is to determine the overall effectiveness of our approach compared to the more simplistic sliding-window approach and a traditional state-of-the-art few-shot object detector. To that end, we performed a sweep over a large combination of hyperparameters with the goal of determining their influence on the overall performance of our processing pipeline and to find a suitable trade-off between detection quality and inference speed. All parameters considered during the experiments and their respective range of values are listed in Table 9.

Table 9. Examined hyperparameters for the evaluation of the three-stage pipeline.

Parameter	Description	Range
p	p-quantile determining the similarity threshold $\tau_s^{(c,k)}$, described in Section 3.2.5	$\left\{1 - \frac{i}{1000} \,\middle|\, i \in \{0, 1, \ldots, 10\}\right\}$
a	Test-time augmentation for queries in the form of horizontal flipping when creating similarity maps	{on, off}

Table 9. *Cont.*

Parameter	Description	Range
R_1	RoI box resolution after combining multiple boxes, described in Section 3.2.6	$\{256^2, 512^2\}$
R_2	RoI box resolution after expanding small boxes, described in Section 3.2.6	$\{256^2, 512^2\}$
τ_b	RoI box area threshold, described in Section 3.2.6	$\{0.8\}$
τ_o	Class-agnostic objectness threshold of object detector	$\left\{ \frac{i}{20} \,\middle|\, i \in \{1, 2, \ldots, 6\} \right\}$
r	Resizing RoIs to a constant size before applying the class-agnostic object detector	{on, off}

4.3.1. Evaluation Protocol

We evaluated our approach in a few-shot setting using the mean average precision (mAP) as metric while also taking into account the models' inference speed and compared it with DE-ViT [8], the currently best few-shot object detector that does not finetune on novel data. Whereas for traditional few-shot models, the confidence of the detector is used to create a ranking for all bounding boxes to calculate the mAP, for our approach, we use the matching distances of the ReID model to the same effect.

We again used the 20 images of the ATTACH table benchmark, described in Section 4.1, in their full resolution of 2560 × 1440 as input images.

We randomly sampled 20 shots of each object category from all ATTACH images not already in use for the table benchmark and ensured that they were reasonably diverse and possessed an area of at least 450 pixels, if possible.

We examined K-shot settings with $K \in \{1, 3, 5, 10, 20\}$ using K of the 20 shots of each object category, respectively. In total there are 1056 different parameter combinations, which we tested exhaustively for each K-shot setting, leading to 5280 data points, the best of which are shown in Figure 9 for our approach.

To evaluate the generalization ability on another assembly dataset, we additionally compared our approach with DE-ViT on the IKEA-ASM dataset. For this experiment, we did not tune any hyperparameters, but used the hyperparameters that achieved the best results on the ATTACH table benchmark.

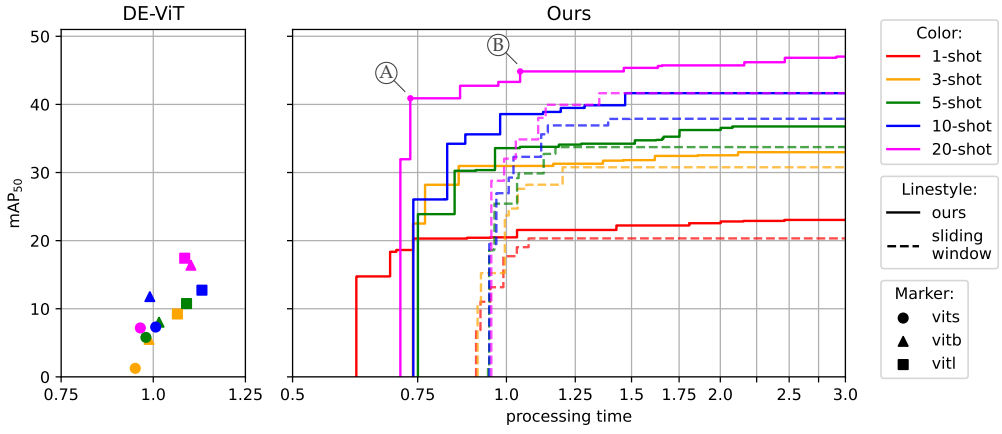


Figure 9. Results of hyperparameter sweeps and comparison with DE-ViT. The processing time represents the inference time in seconds per image on an A100 GPU. The continuous plots represent the Pareto front of data points from the hyperparameter sweeps when considering mAP and inference time. Specific hyperparameters that lead to the operating points marked with A and B are listed in the text.

IKEA-ASM contains images of people assembling IKEA furniture in different locations, including on a table and on the floor. The image resolution is 1920 × 1080. In contrast to

the ATTACH dataset, the furniture of the IKEA-ASM dataset does not contain any small components, like screws. The objects are different panels, boards, and legs. None of these specific objects are contained as categories in the COCO or LVIS dataset, but most of these object have high similarity with furniture categories in COCO and LVIS. Therefore, the class-agnostic object detection is easier than for the ATTACH dataset. However, the recognition of the correct object categories based on the few shots is much more challenging than in the ATTACH benchmark due to a larger variety of different scenes and backgrounds and a higher similarity of the different novel object categories. Manual object annotations as polygons are provided for the frontal camera view. We additionally used the appearance descriptions, such that 15 categories can be distinguished. The IKEA-ASM dataset contains 7710 annotated images in the train partition and 3635 annotated images in the test partition. We used the train partition to randomly select the K shots to describe each of the novel categories. Neither DE-ViT nor our approach were trained on IKEA-ASM data. The test partition is used to benchmark the few-shot object-detection abilities of our approach and DE-ViT. The IKEA-ASM results are shown in Table 11.

4.3.2. Influence of Individual Hyperparameters

We have found that most parameters are quite difficult to evaluate individually, due to the way they interact with each other.

The two most impactful parameters by far turned out to be the quantile p to specify the similarity threshold $\tau_s^{(c,k)}$ and the objectness threshold τ_o. Lower values for p result in more positions in the binary masks (Figure 3(I.B)). Lower values for τ_o result in more class-agnostic proposals being forwarded to the ReID module (Figure 3d). Therefore, lowering p or τ_o will generally increase the detection precision of the pipeline while decreasing its inference speed. These two parameters represent the main levers for determining a suitable trade-off between precision and inference speed.

Test-time augmentation a can increase the detection precision, especially for a low number of K shots provided, while impacting inference speed only slightly.

RoI box sizes R_1 and R_2 should primarily be chosen depending on the sizes of objects present in the input images, with small boxes being beneficial for small objects and vice versa.

Resizing RoIs (r) to a constant resolution of 600 $\times$ 600 after cropping can improve detection quality, especially for small objects, and will generally increase inference speed by a large amount as it allows for more efficient batch processing with the detector.

The operating points marked with A and B in Figure 9 are the result of the hyperparameter combination listed in Table 10.

Table 10. Results and hyperparameters for operating points marked with A and B in Figure 9.

	$\mathbf{mAP_{50}}$	**Inference Time**	p	a	R_1	R_2	τ_b	τ_o	r
Operating point A	40.9	0.73 s/image	1.0	on	512^2	512^2	0.8	0.25	on
Operating point B	44.8	1.05 s/image	1.0	on	512^2	512^2	0.8	0.15	on

The hyperparameters for these two operating points were identical to a large extent. The quantile p was chosen such that the threshold $\tau_s^{(c,k)}$ for similarity map binarization (Figure 3(I.B)) equals the highest similarity to any background feature in the reference set. Test time augmentation (a, Figure 3(I.A)) and input image resizing for the class-agnostic detector (r, Figure 3(II)) were enabled. The RoI box resolution for combining boxes (R_1) and for small box expansion (R_2) were both 512 $\times$ 512 (Figure 3(I.D)). The RoI box area threshold τ_b was 0.8 (Figure 3(I.D)). Only the class-agnostic objectness threshold τ_o (Figure 3(II)) controlled the tradeoff between inference speed and detection quality by influencing the number of object crops forwarded to the ReID module (Figure 3d).

4.3.3. Comparison to State-of-the-Art Few-Shot Object Detector

In Figure 9, we also compare our method with the few-shot object detector DE-ViT, which was trained on COCO. For DE-ViT, we report results for $K \in \{3, 5, 10, 20\}$ shots. In the case of a single shot, DE-ViT did not yield any detections at all on the table benchmark. Our entire processing pipeline, including RoI proposals to narrow the detection search space, outperforms both DE-ViT and the sliding-window-based baseline by a large margin in both mAP and speed. DE-ViT generally performs on par with our approach on larger objects but does not detect smaller ones very well, which make up more than half of the benchmark objects. The sliding-window-based baseline is composed of the same modules as the proposed pipeline, with the exception of the RoI proposal as first processing stage. A comparison of the related dashed lines (without RoI proposal) and solid lines (with RoI proposal) in Figure 9 reveals the benefits of this processing stage for both detection precision and inference speed. The example detection results in Figure 1 at the beginning of the paper were created with the 20-shot configuration, achieving an mAP of 46.2 at 2.15 s/image.

One notable difference when comparing our approach to traditional few-shot detectors is that the initial processing stage creates a different set of RoIs for each individual object instance present among the queries. While RoIs with a sufficiently large IoU overlap can be merged to speed up computation, even across different categories, varying the number of novel objects or categories will change the total number of RoIs and, therefore, impact the inference speed of our approach. This complicates comparisons with traditional few-shot detectors, whose inference speed tends to be fairly static. However, for any number of shots and for up to 12 categories, which is the upper limit in the benchmark, our approach significantly outperformed DE-ViT in terms of detection precision while being faster.

In the target scenario, when a robot collaborates with a human, in each assembly step only a few objects, if not only one, are of interest to identify the current assembly step. Therefore, in practice, an even smaller number of categories may be of interest. If our method is applied to only a few different categories instead of the 12 used in our experiments, a further speed-up can be observed.

4.3.4. Comparison of Generalization Abilities to Another Assembly Scenario

To evaluate the generalization abilities of our approach and DE-ViT [8], we conducted an additional experiment on the IKEA-ASM dataset [7]. We chose two operating points from Figure 9 marked as A and B. Operating point A achieves a good trade-off between detection quality and inference speed. Operating point B was chosen such that the resulting processing pipeline has an inference speed comparable to the best performing DE-ViT model, which uses the large vision transformer (vitl). The hyperparameters for these two operating points are listed in Table 10. For DE-ViT, we chose the two best performing models vitl and vitb. Table 11 shows the results of the two approaches on the IKEA-ASM dataset.

Table 11. Few-shot object-detection results on IKEA-ASM dataset [7] in terms of mean average precision (mAP) and inference time (sec/image). Operating points A and B are marked in Figure 9. The hyperparameters for these operating points are listed in Table 10. Our approach either achieves similar detection quality to DE-ViT while being significantly faster (A) or achieves significantly better detection quality than DE-ViT while being similarly fast (B). The best results are highlighted in bold.

	Mean Average Precision (mAP)			**Inference Time (sec/image)**		
	5 Shots	**10 Shots**	**20 Shots**	**5 Shots**	**10 Shots**	**20 Shots**
DE-ViT, vitl [8]	20.01	16.62	13.52	1.043	1.127	1.027
DE-ViT, vitb [8]	18.60	14.76	11.85	1.043	0.973	0.972
ours, operating point A	15.74	18.42	20.21	**0.551**	**0.553**	**0.537**
ours, operating point B	**26.78**	**32.93**	**35.45**	1.052	1.031	1.009

On the IKEA-ASM dataset, our approach with hyperparameters as in B achieves an inference speed comparable to both DE-ViT models. Regarding the detection quality it clearly outperforms DE-ViT for all numbers of shots. With hyperparameters as in A, our approach performs similar to DE-ViT, but is significantly faster than DE-ViT. However, it seems that the performance drops much more between the operating points A and B on the IKEA-ASM dataset than on the ATTACH dataset, for which the hyperparameters were tuned. Therefore, a more conservative choice of the operating point seems to be a good option to improve the generalization ability. Regarding inference speed, we observed that none of the approaches is significantly influenced by the number of shots. However, our approach achieves slightly better detection quality with more shots. Surprisingly, for DE-ViT this is not the case. We hypothesise that the greater diversity of shots is better exploited if the best fitting shot is utilized for matching detected objects to categories, as in our approach. DE-ViT seems to be confused if the appearance of shots for a category greatly varies.

5. Limitations

Although we achieved promising results on the assembly datasets ATTACH and IKEA-ASM, it is important to acknowledge the limitations of the proposed method. In this paper, we have focused on assembly scenarios. In fact, our method is specifically designed to meet the requirements and conditions of assembly scenarios. It considers the importance of small objects and high-resolution images. It also exploits the fact that in the target scenario, objects of interest have been previously observed and will have a similar appearance in upcoming observations. Generalization to areas other than assembly is beyond the scope of this paper, but should be addressed in future work.

Furthermore, we have shown that our approach can outperform the best few-shot object detector DE-ViT under identical conditions, both in terms of detection quality and inference speed. However, the inference speed is still slow on high resolution images, such as those used in the ATTACH dataset. In fact, it is far from real-time. However, even if objects can only be detected at a low frequency, such as 1–2 Hz, these detections can be tracked to obtain bounding boxes for the missing frames. For action recognition with objects [2], where the objects represent only secondary information, object detections can be useful even if they occur at low frequency.

In our approach, the inference speed depends on the number of categories and shots. However, our experiments have shown that the influence of the number of shots is small. Furthermore, the inference speed is only significantly affected when the number of novel categories is large. In typical assembly scenarios the number of possible objects at a given assembly state is limited.

A more serious limitation of our approach is the relatively large number of hyperparameters that affect performance. However, in our experiments, we have identified the most influential parameters and provided hyperparameter sets that achieve a good trade-off between inference speed and detection quality. Some of the hyperparameters control the required thresholds based on a reference set of background images that must be available. In assembly scenarios, acquiring these images may be straightforward. In other use cases, it may be problematic.

The object re-identification (ReID) module appears to be constrained in its capacity to generalize. Our findings indicate that the module is highly susceptible to the visual characteristics of objects, such as their color, when the shots for a novel category exhibit significant visual similarity. It appears that the ReID module does not acquire knowledge about the shape of objects. Furthermore, discriminating the distributions of the distances of matching and mismatching objects compared to the shots becomes more challenging on out-of-domain data (workpieces, tools) than during training (common everyday objects). As suggested in [73], this can be improved, by using more advanced loss functions for ReID training.

Finally, it may be perceived as a limitation that our approach consists of three stages that are trained independently. It would be beneficial for future work to investigate the possibility of training these three stages end-to-end after pre-training them in the current form.

6. Conclusions

In the context of autonomous agents employed for collaborative assembly with humans, it is uncommon to have complete knowledge about the planned deployment during training. Moreover, when the desired outcome is a deterministic behavior, retraining during deployment is impractical. In light of these considerations, we proposed an object detector that is capable of adapting to novel objects without the need for fine-tuning on novel data. This is accomplished by initially detecting any objects within the scene through the use of a class-agnostic object detector, followed by the matching of these detected objects with the few shots provided to describe the novel categories. In order to obtain a robust class-agnostic object detector, we identified the training dataset to be the most impactful factor. It is essential that the training dataset is sufficiently large, is densely labeled, and includes many categories. These characteristics render LVIS an optimal choice, as evidenced by our experimental results. Moreover, the matching of detected objects with prototypes of novel categories is enabled by an object re-identification model that was trained on the 3D object reconstruction dataset CO3D. This dataset comprises objects from different viewpoints, thereby enabling the re-identification model to learn an embedding in which representations of the same object are in close proximity, regardless of the viewpoint, and representations of different objects are embedded further apart. The results of our experiments demonstrate that this embedding is capable of generalizing well to novel objects that were not present in the training data. Furthermore, we demonstrate that an embedding derived from a re-identification model can be utilized to predict regions of interest, thereby narrowing the search space of a class-agnostic detector based on the prototypes of novel categories. Our experiments revealed that in the assembly scenario, where the majority of objects belong to novel categories not included in the training datasets, our approach significantly outperforms the current state-of-the-art approach DE-ViT for few-shot object detection without fine-tuning, in terms of both inference speed and detection capabilities.

Future Work

Although the proposed processing pipeline demonstrated a clear advantage over the conventional few-shot object-detection pipeline, we identified potential opportunities for further improvements in future work. These include the following: (1) Further investigations into sampling strategies for batch composition and the elimination of the dependence on the background in certain corner cases could benefit the object re-identification module. (2) The pipeline could be accelerated by training a region of interest proposal model that can identify objects on multiple scales, rather than relying on multiple input resolutions. (3) An investigation into the efficacy of different aggregation functions for the matching distances of the K shots could reveal whether the multi-shot performance could be improved. We plan to investigate in these directions in future work.

Author Contributions: Conceptualization, M.E., H.F., E.F. and M.K.; methodology, M.E., H.F., E.F. and M.K.; software, H.F. and E.F.; validation, M.E., H.F., E.F., D.A. and D.S.; formal analysis, M.E. and D.S.; investigation, M.K., D.A. and D.S.; resources, M.K., H.F., E.F. and D.A.; data curation, M.E., H.F., E.F. and D.A.; writing—original draft preparation, M.E. and H.F.; writing—review and editing, M.E., M.K., D.A., D.S. and H.-M.G.; visualization, M.E., H.F., and E.F.; supervision, M.E., M.K., D.S. and H.-M.G.; project administration, M.E. and H.-M.G.; funding acquisition, M.E. and H.-M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Carl Zeiss Foundation as part of the project 'Engineering for Smart Manufacturing (E4SM) — Engineering of machine learning-based assistance systems for data-intensive industrial scenarios' (funding number: P2017-01-005).

Data Availability Statement: The data and the code are available at https://www.tu-ilmenau.de/neurob/data-sets-code/object-reid.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Eisenbach, M.; Aganian, D.; Köhler, M.; Stephan, B.; Schroeter, C.; Gross, H.M. Visual Scene Understanding for Enabling Situation-Aware Cobots. In Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE), Lyon, France, 23–27 August 2021.
2. Aganian, D.; Köhler, M.; Baake, S.; Eisenbach, M.; Groß, H.M. How object information improves skeleton-based human action recognition in assembly tasks. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023; pp. 1–9.
3. Li, W.; Wei, H.; Wu, Y.; Yang, J.; Ruan, Y.; Li, Y.; Tang, Y. TIDE: Test-Time Few-Shot Object Detection. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**. [CrossRef]
4. Antonelli, S.; Avola, D.; Cinque, L.; Crisostomi, D.; Foresti, G.L.; Galasso, F.; Marini, M.R.; Mecca, A.; Pannone, D. Few-shot object detection: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–37. [CrossRef]
5. Köhler, M.; Eisenbach, M.; Gross, H.M. Few-shot object detection: A comprehensive survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, 1–21. [CrossRef]
6. Aganian, D.; Stephan, B.; Eisenbach, M.; Stretz, C.; Gross, H.M. ATTACH dataset: Annotated two-handed assembly actions for human action understanding. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 11367–11373.
7. Ben-Shabat, Y.; Yu, X.; Saleh, F.; Campbell, D.; Rodriguez-Opazo, C.; Li, H.; Gould, S. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 847–859.
8. Zhang, X.; Wang, Y.; Boularias, A. Detect everything with few examples. *arXiv* **2023**, arXiv:2309.12969.
9. Liang, S.; Wang, W.; Chen, R.; Liu, A.; Wu, B.; Chang, E.C.; Cao, X.; Tao, D. Object Detectors in the Open Environment: Challenges, Solutions, and Outlook. *arXiv* **2024**, arXiv:2403.16271.
10. Dhamija, A.; Gunther, M.; Ventura, J.; Boult, T. The overlooked elephant of object detection: Open set. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Seattle, WA, USA, 13–19 June 2020; pp. 1021–1030.
11. Du, X.; Wang, Z.; Cai, M.; Li, Y. Vos: Learning what you don't know by virtual outlier synthesis. In Proceedings of the International Conference of Learning Representations (ICLR), Virtual Event, 25–29 April 2022.
12. Joseph, K.; Khan, S.; Khan, F.S.; Balasubramanian, V.N. Towards open world object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 5830–5840.
13. Zhao, X.; Ma, Y.; Wang, D.; Shen, Y.; Qiao, Y.; Liu, X. Revisiting open world object detection. *IEEE Trans. Circuits Syst. Video Technol.* **2023**, *34*, 3496–3509. [CrossRef]
14. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014 ; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
15. Gupta, A.; Dollar, P.; Girshick, R. LVIS: A dataset for large vocabulary instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5356–5364.
16. Singh, B.; Li, H.; Sharma, A.; Davis, L.S. R-FCN-3000 at 30fps: Decoupling detection and classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1081–1090.
17. Michaelis, C.; Bethge, M.; Ecker, A.S. A Broad Dataset is All You Need for One-Shot Object Detection. *arXiv* **2020**, arXiv:2011.04267.
18. Erhan, D.; Szegedy, C.; Toshev, A.; Anguelov, D. Scalable object detection using deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2147–2154.
19. Zhou, Z.; Yang, Y.; Wang, Y.; Xiong, R. Open-set object detection using classification-free object proposal and instance-level contrastive learning. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1691–1698. [CrossRef]
20. Jaiswal, A.; Wu, Y.; Natarajan, P.; Natarajan, P. Class-agnostic object detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Virtual, 5–9 January 2021; pp. 919–928.
21. Maaz, M.; Rasheed, H.; Khan, S.; Khan, F.S.; Anwer, R.M.; Yang, M.H. Class-agnostic object detection with multi-modal transformer. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 512–531.
22. He, Y.; Chen, W.; Tan, Y.; Wang, S. Usd: Unknown sensitive detector empowered by decoupled objectness and segment anything model. *arXiv* **2023**, arXiv:2306.02275.
23. Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.Y.; et al. Segment anything. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3October 2023; pp. 4015–4026.

24. Han, J.; Ren, Y.; Ding, J.; Pan, X.; Yan, K.; Xia, G.S. Expanding low-density latent regions for open-set object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9591–9600.
25. Bansal, A.; Sikka, K.; Sharma, G.; Chellappa, R.; Divakaran, A. Zero-shot object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 384–400.
26. Zhu, P.; Wang, H.; Saligrama, V. Zero shot detection. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 998–1010. [CrossRef]
27. Rahman, S.; Khan, S.H.; Porikli, F. Zero-shot object detection: Joint recognition and localization of novel concepts. *Int. J. Comput. Vis.* **2020**, *128*, 2979–2999. [CrossRef]
28. Tan, C.; Xu, X.; Shen, F. A survey of zero shot detection: Methods and applications. *Cogn. Robot.* **2021**, *1*, 159–167. [CrossRef]
29. Zareian, A.; Rosa, K.D.; Hu, D.H.; Chang, S.F. Open-vocabulary object detection using captions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 14393–14402.
30. Zhu, C.; Chen, L. A Survey on Open-Vocabulary Detection and Segmentation: Past, Present, and Future. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, 1–20. [CrossRef]
31. Cheng, T.; Song, L.; Ge, Y.; Liu, W.; Wang, X.; Shan, Y. YOLO-World: Real-Time Open-Vocabulary Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle WA, USA, 17–21 June 2024; pp. 16901–16911.
32. Zhang, J.; Huang, J.; Jin, S.; Lu, S. Vision-language models for vision tasks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2024**, *46*, 5625–5644. [CrossRef]
33. Huang, Q.; Zhang, H.; Xue, M.; Song, J.; Song, M. A survey of deep learning for low-shot object detection. *ACM Comput. Surv.* **2023**, *56*, 1–37. [CrossRef]
34. Fan, Q.; Zhuo, W.; Tang, C.K.; Tai, Y.W. Few-shot object detection with attention-RPN and multi-relation detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 4013–4022.
35. Li, X.; Zhang, L.; Chen, Y.P.; Tai, Y.W.; Tang, C.K. One-shot object detection without fine-tuning. *arXiv* **2020**, arXiv:2005.03819.
36. Li, Y.; Feng, W.; Lyu, S.; Zhao, Q.; Li, X. MM-FSOD: Meta and metric integrated few-shot object detection. *arXiv* **2020**, arXiv:2012.15159.
37. Perez-Rua, J.M.; Zhu, X.; Hospedales, T.M.; Xiang, T. Incremental few-shot object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 13846–13855.
38. Yang, Y.; Wei, F.; Shi, M.; Li, G. Restoring negative information in few-shot object detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 3521–3532.
39. Chen, T.I.; Liu, Y.C.; Su, H.T.; Chang, Y.C.; Lin, Y.H.; Yeh, J.F.; Chen, W.C.; Hsu, W.H. Dual-awareness attention for few-shot object detection. *IEEE Trans. Multimed.* **2021**, *25*, 291–301. [CrossRef]
40. Han, G.; He, Y.; Huang, S.; Ma, J.; Chang, S.F. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 3263–3272.
41. Zhang, L.; Zhou, S.; Guan, J.; Zhang, J. Accurate few-shot object detection with support-query mutual guidance and hybrid loss. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 14424–14432.
42. Han, G.; Huang, S.; Ma, J.; He, Y.; Chang, S.F. Meta faster r-cnn: Towards accurate few-shot object detection with attentive feature alignment. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022; Volume 36, pp. 780–789.
43. Kobayashi, D. Self-supervised prototype conditional few-shot object detection. In Proceedings of the International Conference on Image Analysis and Processing, Lecce, Italy, 23–27 May 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 681–692.
44. Li, B.; Wang, C.; Reddy, P.; Kim, S.; Scherer, S. Airdet: Few-shot detection without fine-tuning for autonomous exploration. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 427–444.
45. Bulat, A.; Guerrero, R.; Martinez, B.; Tzimiropoulos, G. FS-DETR: Few-Shot DEtection TRansformer with prompting and without re-training. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 11793–11802.
46. Yang, H.; Cai, S.; Deng, B.; Ye, J.; Lin, G.; Zhang, Y. Context-aware and Semantic-consistent Spatial Interactions for One-shot Object Detection without Fine-tuning. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 5424–5439. [CrossRef]
47. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 $\times$ 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
48. Gao, C.; Hao, J.; Guo, Y. OSDet: Towards Open-Set Object Detection. In Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN), Gold Coast, Australia, 18–23 June 2023; pp. 1–8.
49. Mallick, P.; Dayoub, F.; Sherrah, J. Wasserstein Distance-based Expansion of Low-Density Latent Regions for Unknown Class Detection. *arXiv* **2024**, arXiv:2401.05594.
50. Sarkar, H.; Chudasama, V.; Onoe, N.; Wasnik, P.; Balasubramanian, V.N. Open-Set Object Detection by Aligning Known Class Representations. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 1–6 January 2024; pp. 219–228.

51. Wu, A.; Deng, C. TIB: Detecting Unknown Objects via Two-Stream Information Bottleneck. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *46*, 611–625. [CrossRef]
52. Wu, A.; Chen, D.; Deng, C. Deep feature deblurring diffusion for detecting out-of-distribution objects. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 13381–13391.
53. Wan, Q.; Wang, S.; Xiang, X. A Simple Unknown-Instance-Aware Framework for Open-Set Object Detection. In Proceedings of the 2023 13th International Conference on Information Science and Technology (ICIST), Cairo, Egypt, 8–14 December 2023; pp. 586–593.
54. Yang, H.M.; Zhang, X.Y.; Yin, F.; Yang, Q.; Liu, C.L. Convolutional prototype network for open set recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 2358–2370. [CrossRef] [PubMed]
55. Zheng, J.; Li, W.; Hong, J.; Petersson, L.; Barnes, N. Towards open-set object detection and discovery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 3961–3970.
56. Hayes, T.L.; de Souza, C.R.; Kim, N.; Kim, J.; Volpi, R.; Larlus, D. PANDAS: Prototype-based Novel Class Discovery and Detection. *arXiv* **2024**, arXiv:2402.17420.
57. Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; et al. Dinov2: Learning robust visual features without supervision. *arXiv* **2023**, arXiv:2304.07193.
58. Gorlo, N.; Blomqvist, K.; Milano, F.; Siegwart, R. ISAR: A Benchmark for Single-and Few-Shot Object Instance Segmentation and Re-Identification. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 1–6 January 2024; pp. 4384–4396.
59. Jiang, S.; Liang, S.; Chen, C.; Zhu, Y.; Li, X. Class agnostic image common object detection. *IEEE Trans. Image Process.* **2019**, *28*, 2836–2846. [CrossRef]
60. Nguyen, C.H.; Nguyen, T.C.; Vo, A.H.; Masayuki, Y. Single Stage Class Agnostic Common Object Detection: A Simple Baseline. *arXiv* **2021**, arXiv:2104.12245.
61. Guo, X.; Li, X.; Wang, Y.; Jiang, S. TransWeaver: Weave Image Pairs for Class Agnostic Common Object Detection. *IEEE Trans. Image Process.* **2023**, *32*, 2947–2959. [CrossRef]
62. Dümmel, J.; Gao, X. Object Re-Identification with Synthetic Training Data in Industrial Environments. In Proceedings of the 2021 27th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Shanghai, China, 26–28 November 2021; pp. 504–508.
63. Chen, W.; Liu, Y.; Wang, W.; Bakker, E.M.; Georgiou, T.; Fieguth, P.; Liu, L.; Lew, M.S. Deep Learning for Instance Retrieval: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 7270–7292. [CrossRef]
64. Radenović, F.; Iscen, A.; Tolias, G.; Avrithis, Y.; Chum, O. Revisiting Oxford and Paris: Large-Scale Image Retrieval Benchmarking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
65. Weyand, T.; Araujo, A.; Cao, B.; Sim, J. Google Landmarks Dataset v2 - A Large-Scale Benchmark for Instance-Level Recognition and Retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
66. Shao, S.; Chen, K.; Karpur, A.; Cui, Q.; Araujo, A.; Cao, B. Global features are all you need for image retrieval and reranking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–3 October 2023; pp. 11036–11046.
67. Lee, S.; Seong, H.; Lee, S.; Kim, E. Correlation Verification for Image Retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5374–5384.
68. Radenović, F.; Tolias, G.; Chum, O. Fine-Tuning CNN Image Retrieval with No Human Annotation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1655–1668. [CrossRef]
69. Luo, H.; Gu, Y.; Liao, X.; Lai, S.; Jiang, W. Bag of tricks and a strong baseline for deep person re-identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 16–17 June 2019.
70. Zhang, H.; Li, F.; Liu, S.; Zhang, L.; Su, H.; Zhu, J.; Ni, L.M.; Shum, H.Y. Dino: DETR with improved denoising anchor boxes for end-to-end object detection. *arXiv* **2022**, arXiv:2203.03605.
71. Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv* **2019**, arXiv:1906.07155.
72. Ye, M.; Shen, J.; Lin, G.; Xiang, T.; Shao, L.; Hoi, S.C. Deep learning for person re-identification: A survey and outlook. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 2872–2893. [CrossRef]
73. Aganian, D.; Eisenbach, M.; Wagner, J.; Seichter, D.; Gross, H.M. Revisiting Loss Functions for Person Re-identification. In Proceedings of the Artificial Neural Networks and Machine Learning–ICANN, Bratislava, Slovakia, 14–17 September 2021; Farkaš, I., Masulli, P., Otte, S., Wermter, S., Eds.; Springer: Cham, Switzerland, 2021; pp. 30–42.
74. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-Local Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
75. Seichter, D.; Fischedick, S.B.; Köhler, M.; Groß, H.M. Efficient multi-task rgb-d scene analysis for indoor environments. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–10.
76. Lyu, C.; Zhang, W.; Huang, H.; Zhou, Y.; Wang, Y.; Liu, Y.; Zhang, S.; Chen, K. Rtmdet: An empirical study of designing real-time object detectors. *arXiv* **2022**, arXiv:2212.07784.

77. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020. Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
78. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
79. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
80. Wu, Y.; Kirillov, A.; Massa, F.; Lo, W.Y.; Girshick, R. Detectron2. 2019. Available online: https://github.com/facebookresearch/detectron2 (accessed on 1 August 2024).
81. Reizenstein, J.; Shapovalov, R.; Henzler, P.; Sbordone, L.; Labatut, P.; Novotny, D. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10901–10911.
82. Choi, S.; Zhou, Q.Y.; Miller, S.; Koltun, V. A large dataset of object scans. *arXiv* **2016**, arXiv:1602.02481.
83. Downs, L.; Francis, A.; Koenig, N.; Kinman, B.; Hickman, R.; Reymann, K.; McHugh, T.B.; Vanhoucke, V. Google scanned objects: A high-quality dataset of 3d scanned household items. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 2553–2560.
84. Ahmadyan, A.; Zhang, L.; Ablavatski, A.; Wei, J.; Grundmann, M. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 7822–7831.
85. Henzler, P.; Reizenstein, J.; Labatut, P.; Shapovalov, R.; Ritschel, T.; Vedaldi, A.; Novotny, D. Unsupervised learning of 3d object categories from videos in the wild. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 4700–4709.
86. Kirillov, A.; Wu, Y.; He, K.; Girshick, R. PointRend: Image Segmentation As Rendering. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
87. Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; Tian, Q. Scalable Person Re-Identification: A Benchmark. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
88. Karaoguz, H.; Jensfelt, P. Fusing saliency maps with region proposals for unsupervised object localization. *arXiv* **2018**, arXiv:1804.03905.
89. Shilkrot, R.; Narasimhaswamy, S.; Vazir, S.; Hoai, M. WorkingHands: A Hand-Tool Assembly Dataset for Image Segmentation and Activity Mining. In Proceedings of the British Machine Vision Conference, Cardiff, UK, 9–12 September 2019.
90. Stephan, B.; Köhler, M.; Müller, S.; Zhang, Y.; Gross, H.M.; Notni, G. OHO: A Multi-Modal, Multi-Purpose Dataset for Human-Robot Object Hand-Over. *Sensors* **2023**, *23*, 7807. [CrossRef]
91. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
92. Hermans, A.; Beyer, L.; Leibe, B. In defense of the triplet loss for person re-identification. *arXiv* **2017**, arXiv:1703.07737.
93. Yokoo, S.; Ozaki, K.; Simo-Serra, E.; Iizuka, S. Two-Stage Discriminative Re-Ranking for Large-Scale Landmark Retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Seattle, WA, USA, 13–19 June 2020.
94. Gong, Y.; Zeng, Z.; Chen, L.; Luo, Y.; Weng, B.; Ye, F. A person re-identification data augmentation method with adversarial defense effect. *arXiv* **2021**, arXiv:2101.08783.
95. Zhu, S.; Zhang, Y.; Feng, Y. GW-net: An efficient grad-CAM consistency neural network with weakening of random erasing features for semi-supervised person re-identification. *Image Vis. Comput.* **2023**, *137*, 104790. [CrossRef]
96. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
97. Sun, Y.; Cheng, C.; Zhang, Y.; Zhang, C.; Zheng, L.; Wang, Z.; Wei, Y. Circle Loss: A Unified Perspective of Pair Similarity Optimization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.