Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Self-supervised Vision Transformers for 3D pose estimation of novel objects

Stefan Thalhammer [a,*], Jean-Baptiste Weibel [a], Markus Vincze [a], Jose Garcia-Rodriguez [b]

[a] *Automation and Control Institute, TU Wien, Gußhausstraße 27-29, Vienna 1040, Vienna, Austria*
[b] *Department of Computer Technology, University of Alicante, Carr. de San Vicente del Raspeig, San Vicente del Raspeig 03690, Alicante, Spain*

ARTICLE INFO

Keywords:
Object pose estimation
Template matching
Vision transformer
Self-supervised learning

ABSTRACT

Object pose estimation is important for object manipulation and scene understanding. In order to improve the general applicability of pose estimators, recent research focuses on providing estimates for novel objects, that is, objects unseen during training. Such works use deep template matching strategies to retrieve the closest template connected to a query image, which implicitly provides object class and pose. Despite the recent success and improvements of Vision Transformers over CNNs for many vision tasks, the state of the art uses CNN-based approaches for novel object pose estimation. This work evaluates and demonstrates the differences between self-supervised CNNs and Vision Transformers for deep template matching. In detail, both types of approaches are trained using contrastive learning to match training images against rendered templates of isolated objects. At test time such templates are matched against query images of known and novel objects under challenging settings, such as clutter, occlusion and object symmetries, using masked cosine similarity. The presented results not only demonstrate that Vision Transformers improve matching accuracy over CNNs but also that for some cases pre-trained Vision Transformers do not need fine-tuning to achieve the improvement. Furthermore, we highlight the differences in optimization and network architecture when comparing these two types of networks for deep template matching.

1. Introduction

Object pose estimation is an important yet difficult vision problem. Many downstream tasks, such as robotic grasping [36], augmented reality [24] and reconstruction [34] benefit from the availability of object poses. Classical object pose estimation approaches encode latent representations of multiple object views per object during training. During run-time these are matched against an observation to retrieve a coarse object pose [20,23,12]. After retrieving the pose of the closest template, poses are refined using the Iterative-Closest-Points [17] algorithm or other algorithms to optimize the rigid transformation between two corresponding sets of points. In contrast, learning-based solutions using Convolutional Neural Networks (CNNs) learn a feature representation to infer object class and geometric correspondences during testing [33,37,46,25,47,49,1,42,9]. Yet, training pose estimators for each object instance [33,37], or each set of object instances [47,49] is insufficient to be usable in real world scenarios where object instances are manifold and constantly changing. As a consequence, research has shifted towards category-level [50,38] and novel object pose estimation [31,41,29]. These recent novel object pose estimation approaches are similar to classical approaches in that queries are matched against templates.

The approach of [31] employs a CNN backbone to learn occlusion-aware template matching for novel object pose estimation. Real observations are matched against rendered templates and tested for 3*D* pose estimation. While they show that such strategies are expedient for novel object pose estimation it has been shown that Vision Transformers (ViT) [11,48,5] learn more discriminative feature spaces than CNNs when trained in an unsupervised manner. This advantage of ViTs over CNNs, however, has primarily been empirically demonstrated by matching to distinct object classes and not by matching views of the same object class for more complex reasoning, such as 3*D* object pose estimation [5,8].

In this work we empirically demonstrate that ViTs excel over CNNs when used for novel object pose estimation. Modifying the approach of [31] for comparing two similarly sized feature extractors, ResNet50 [18] with 23*M* and ViT-s [48] with 21*M* parameters, we show that these improvements are manifold. Training self-supervised ViTs for 3*D* object pose estimation not only improves the template matching accuracy but

[*] Corresponding author.
E-mail addresses: thalhammer@acin.tuwien.ac.at (S. Thalhammer), weibel@acin.tuwien.ac.at (J.-B. Weibel), vincze@acin.tuwien.ac.at (M. Vincze), jgarcia@dtic.ua.es (J. Garcia-Rodriguez).

https://doi.org/10.1016/j.imavis.2023.104816
Received 6 June 2023; Received in revised form 4 August 2023; Accepted 6 September 2023
Available online 12 September 2023

also reduces the training time. Depending on the dataset and metric, template matching accuracy for seen objects improves by 1% on Linemod [20], over 4% on Linemod-Occlusion [4] and by 19% on T-LESS [21]. For unseen objects, the respective improvements are 3%, 5% and 18%. Achieving these improvements using ViT-s takes one quarter of the training time and iterations on LM and LM-O, and is 25 times faster on T-LESS. More remarkably, testing ViT-s on T-LESS in a zero-shot fashion, i.e., without fine-tuning, already improves over using fine-tuned ResNet50 by 7% and 9%, for seen and unseen objects respectively. Finally, works such as [5,8] train self-supervised ViTs to retrieve the object class of seen objects assuming the availability of templates in the same domain. These assumptions are impractical for novel object pose estimation. Uniform coverage of the pose space is crucial and thus rendering templates is expedient. Furthermore, handling unseen objects is desired to further generalize real-world deployment of pose estimators. As a consequence, this work provides ablations on the network architecture used for matching. While the aforementioned works [5,8] benefit from using high-dimensional, multi-layered projection heads, we empirically show that these increase the template matching error on unseen objects when matched against rendered templates. In summary we:

- Show that Vision Transformers not only exhibit reduced template matching errors compared to CNNs for matching synthetic templates to known objects but also to novel objects. The relative improvements for novel object pose estimation range from 3% to 18%, depending on the dataset and metric used.
- Demonstrate that pre-trained Vision Transformers exhibit excellent matching performance for zero-shot matching. On the T-LESS dataset, non fine-tuned Vision Transformers exhibit a relative improvement over fine-tuned CNNs of 7% and 9% on known and novel objects respectively. Fine-tuning further improves to 19% and 18% respectively.
- Highlight the differences in matching procedure and optimization of fine-tuning Vision Transformers for template matching. Our results indicate that Vision Transformers encode relevant features over a broad range of descriptor sizes for seen and novel objects as compared to CNNs where there is a trade-off when choosing the descriptor size for either seen or novel objects. Our results additionally indicate that high-dimensional, multi-layered projection heads increase the template matching error for the problem at hand.

The remainder of the manuscript is organized in the sections Related Work, Method, Experiments and Conclusion. The next section presents the state of the art for object pose estimation, focusing on deep template matching for deriving poses of novel objects, and self-supervised vision transformers.

2. Related work

This sections presents the state of the art for object pose estimation with the focus on novel object pose estimation. Subsequently, ViTs and methods for their self-supervised training are presented.

Learning-based object pose estimation research focuses on multi-staged pipelines [27,33,49,42] that often train separate networks for instance-level pose estimation [33,49] in order to improve the estimated pose's accuracy. Different streams of research improve on the scalability of instance-level pose estimation, presenting solutions for improved multi-object handling [1,46,55] and reducing the number of stages needed for providing reliable pose estimates [47,9,54]. Yet, re-training pose estimators every time novel objects or object sets are encountered is cumbersome and delays the deployment in the real world. As a consequence, recent works overcome these shortcomings by training for category-level pose estimation [50,38] or by training deep template matching for novel object pose estimation [31,41,29].

Deep Template Matching: Matching observations against predefined templates is a long-standing concept of object pose estimation [20,23,12]. Recent learning-based solutions adopt this strategy since it has two major advantages [31,41,29]. First, training time is low since encoding templates does not require learning a representation of each object individually. Creating a latent representations for each relevant template only requires one network forward pass. Thus, template encoding is done in the magnitude of seconds for an object of interest as compared to training instance-level pose estimators, which takes hours to days, depending on the number of objects and the hardware used [33,49,47]. Second, training instance-level pose estimators encodes a latent representation of the object(s) of interest. This representation does not generalize to novel objects. This shortcoming is addressed by either category-level object pose estimation or by deep template matching.

The approach of [51] introduces deep descriptors for matching query objects against templates for retrieving the 3*D* pose using nearest neighbor search. In [3] the authors improve over [51] by guiding learning in pose space and also accounting for object symmetries in the process. Recently, [31] proposed further improvements. They replace triplet loss-based training with an InfoNCE-based procedure and improve occlusion handling by masking the feature embedding using the template's mask and an occlusion threshold. We adopt and improve over their approach for deep template matching by using ViTs for descriptor extraction, which have not yet been adopted by the community. As such, we demonstrate their advantage with respect to their generality as a deep template matcher and show empirical evaluations highlighting their advantages for the problem of novel object pose estimation.

Vision Transformers: It has recently been shown that ViTs [11,35] learn superior features when trained in a self-supervised fashion [8,5,48]. These mainstream works focus on training object classifiers from scratch and using large datasets with little domain shift between query images and templates. Such large datasets are difficult to obtain for object pose estimation due to the complexity of generating accurate 6*D* pose annotations. Additionally, it is relevant for pose estimation to effectively cover the viewing sphere around objects of interest [44]. This implies training on comparably small datasets and preferably using synthetic templates, i.e., using rendering for template creation [10]. As such, in this work, ViTs are assumed to be pre-trained and templates are rendered. We thus show the potential of self-supervised ViTs under this shifted perspective and also highlight the differences of network design as compared to the mainstream research direction.

3. Method

This section presents our self-supervised learning framework for matching real observations to synthetic templates for novel object 3*D* pose estimation. Fig. 1 provides an abstract visualization of the presented method.

Contrastive learning is used for Self-supervised training. This approach maximizes the similarity of semantically similar training samples, referred to as positive pairs, while minimizes the similarity for samples that are semantically dissimilar, that is negative pairs. More precisely, one training sample consists of a tuple of a query crop (I_q), a positive example (I_{pos}), and a negative example (I_{neg}). The positive and negative templates are rendered using physically-based rendering (pbr) [10]. Where the positive sample correlates with the object class and rotation in the query image. The negative sample deviates with respect to both properties. Crops are tokenized using random patch embeddings and a shared pre-trained ViT-s [48] is used for extracting features of the query and the template images. In contrast to self-supervised ViT frameworks for classification [5,8] we discard the class token and employ the positional tokens for similarity calculation. Using such spatial output enables dropping tokens based on the positive template's mask. Optimization is guided using InfoNCE-loss [32] with the positive and negative similarities as input. During testing, similarities are computed between real object observations and pbr-templates of seen

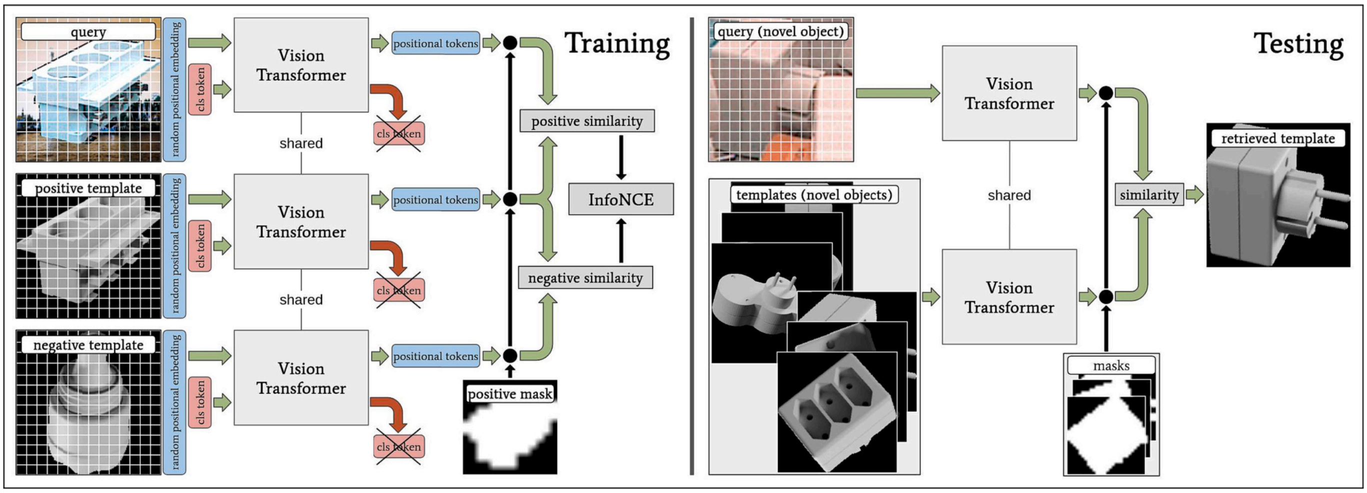

Fig. 1. Method overview During training, a query image, a positive template and a negative template are processed by a Vision Transformer to encode a feature embedding. The number of the positional tokens is retained for the feature map. InfoNCE [32] is used in a Triplet loss-like fashion with the input feature map being masked with the positive template. During testing, novel query objects are matched against templates to retrieve object class and 3*D* pose from the matched template. Template retrieval is guided using the masked cosine similarity.

and novel objects. Thus, in contrast to contemporary ViT research, similarities have to bridge the synthetic-to-real gap since templates are created using rendering [5,8]. The real observations are compared against templates that represent uniformly distributed object views of the potentially new objects. Ultimately, the class and the 3*D* rotation of the matched template are retrieved.

3.1. Feature embedding

The aim of this work is novel object pose estimation. Recent works shows that deep contrastive-learned template matching strategies are well suited for this task [31,41,29]. In order to exhibit high similarities between similar view points of the same object in different domains, the learned feature embedding has to represent the object view as accurately as possible. It has been shown that Vision Transformers [48,11,35] trained in an unsupervised way, learn to accurately model long-range image relationships, which improves over CNNs [5].

This works adopts the ViT-s network presented in [48] as feature extractor. The weights are pre-trained on ImageNet [28] in a self-supervised manner [5]. ViT-s is used by only retaining the class token for training and testing [5]. In this work, the class token is discarded and the positional tokens are retained in order to benefit from the spatial nature of the output. Diverse works indicate that augmenting feature extractors with deep multi-layered heads for projecting embeddings to higher dimensions improves performance when training on ImageNet [8,5,7,15]. The presented results in Section 4 indicate this finding does not apply to pose estimation. A single linearly-activated fully-connected layer projects the feature embedding, coming from the pre-trained backbone, to a lower dimensionality. It has to be noted that this different behavior is connected to the difference in problem: a) the backbone is initialized with pre-trained weights, b) the problem at hand matches real observations against rendered templates and c) testing is partially done on novel objects, thus data unseen during training. We hypothesize that using deeper heads causes overfitting to the training data characteristics.

The authors of [8] note that randomly initialized patch embedding stabilizes training on ImageNet and thus improves classification accuracy. Accordingly, the patch embedding layer is not updated during fine-tuning. Results are provided in Section 4.

3.2. Contrastive learning framework

The feature embeddings extracted using ViT-s are processed by a contrastive learning framework to learn a representation with increased similarity between object crops of the same class and a similar viewpoint. As similarity measure, the cosine similarity is employed:

$$sim\left(emb_{I_q,t}, emb_{*,t}\right) = \frac{emb_{I_q,t} \cdot emb_{*,t}}{\left\|emb_{I_q,t}\right\|_2, \left\|emb_{*,t}\right\|_2)} \tag{1}$$

where * is either I_{pos} or I_{neg}. The similarity is computed locally and aggregated for locations indicated by the mask image:

$$sim_{pos/neg} = \sum_{t=1}^{T} sim\left(I_q, *\right) \times M_t \begin{cases} sim & \text{if } M_t == 1, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where T refers to the number of feature map locations, i.e., the number of positional tokens. The negative similarity is summed over all embedded tokens inside the template's object mask, while the positive similarity is computed globally with $M = 1^{\mathrm{sizeof} I_q}$. Both similarities are used in a triplet loss fashion [6] using InfoNCE loss [16,32]. Each positive sample is compared against all negative samples in a batch, resulting in $B = (b \cdot b) - b$ negative samples per iteration. The final loss is:

$$L = -\sum_{i=1}^{b} \log \frac{\exp \frac{sim_{pos,i}}{\tau}}{\sum_{k=1}^{B} \frac{sim_{neg,k}}{\tau} \forall i \neq k} \tag{3}$$

where τ is a temperature parameter set to 0.1. For more details refer to [31].

3.3. Template matching

During testing, templates of seen and novel objects are matched against the query image. Embeddings are created for the query crop and all templates. The cosine similarity in (1) is reused with the modification:

$$sim_q = \sum_{t=1}^{T} sim\left(I_q, *\right) \times M \begin{cases} sim & \text{if } M_t == 1, \\ & \text{and } sim_t > \delta, \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where δ is a hyperparameter set to 0.2, which increases robustness against occluded image regions as introduced in [31]. The class and 3*D* rotation of the template leading to the highest cumulative cosine similarity are retrieved.

4. Experiments

This section presents results for 3*D* pose estimation. Firstly, we compare the performance of our approach that uses ViT-s as a feature extractor to CNN-based baseline methods such as [31]. Secondly, we evaluate the generality of the self-supervised pre-trained ViT-s without fine-tuning to show that even without fine-tuning, the template matching error is low and improves over the baseline method on T-LESS. Finally, we present diverse ablations that highlight the differences between ViT- and CNN-architectures for 3*D* pose estimation. The section is concluded by providing an ablation with respect to the projection head used for our approach, highlighting the fundamental difference that shallow heads are beneficial in comparison to classification approaches on ImageNet [5,8,7,15].

4.1. Experimental setup

In the following, we detail data retrieval and processing then explain template creation for matching is explained. In order to evaluate the proposed approach, standard metrics from concurrent, conceptually similar approaches are also presented.

4.1.1. Datasets

Results are provided on three standard datasets for object pose estimation: Linemod [20] (LM), Linemod-Occlusion [4] (LM-O), and T-LESS [21]. These datasets are processed to provide crop-level data in order to evaluate template matching accuracy and compare against the baseline method.

LM and LM-O These are two of the most used datasets for evaluating object pose estimation. LM features 13 objects and for each object a set of ≈ 1200 scene-level images is available. Annotations are only provided for the respective object although each set contains multiple objects of the dataset in the cluttered background. The main characteristics of the dataset are texture-poor objects of different geometry, sizes and colors. Annotated object views exhibit virtually no occlusion. As as consequence, the authors of [4] create annotations for all 8 dataset objects in the Benchvise's set, thus introducing LM-O as a test set specifically for strongly occluded object views.

With respect to training and testing, we follow [31] in order to provide a fair comparison. For evaluation on seen and unseen objects, the LM-objects are partitioned into three sets as shown in Table 1. Training data consists of 90% of LM images per object set and the remaining 10% are used for testing. As a consequence of this split, training images are without occlusion. The images of LM-O are exclusively used for testing. In order to evaluate on all objects, one split is used for testing on unseen objects, while the other two are used for training.

T-LESS For T-LESS we follow the protocol presented in [43]. Isolated object views of the objects 1 −18 are used for training and are pasted onto a randomly chosen image of SUN397 [52] using the cut-paste strategy [13]. These 18 objects are considered as seen objects, while the remaining objects, 19 −30, are considered novel. Test images are cropped from the primesense test set.

Table 1
LM/LM-O object splits. Two of the sets are used for training and testing on seen objects while the third is used for testing on unseen objects as done in [31].

Split	Objects
1	Ape, Benchvise, Camera and Can
2	Cat, Driller, Duck and Eggbox
3	Glue, Holepuncher, Iron, Lamp and Phone

4.1.2. Template generation

In contrast to works that train self-supervised ViTs for image classification [5,8], this work considers matching the closest template for viewpoint classification, thus for 3*D* pose estimation. The major difference is that templates uniformly distributed in the viewing sphere, respectively hemisphere, are required, which is not relevant for aforementioned works. Consequently, templates to match against are created using physically-based rendering [10].

LM and LM-O The training and test datasets for LM and LM-O are processed as done in [51,31]. These works crop the images from the real dataset by omitting in-plane rotations thus effectively only considering azimuth and elevation as degrees of freedom. Objects are cropped in a way that the image space at object distance projects 0.4 by 0.4 meters, which results in all objects appearing at the same distance to the camera, independent of their size. Furthermore, neither the LM nor LM-O training and test images show objects from the lower viewing hemisphere. Due to these constraints, 301 templates are sufficient for training and testing on LM and LM-O.

T-LESS For T-LESS, objects are cropped in a way to tightly encapsulate the objects. Additionally, objects appear in arbitrary views in the test set. As a consequence, 92, 232 templates are used for training and testing on T-LESS as done in [31,43].

4.1.3. Evaluation

This section presents the metrics used in this work. The approach of [31] introduces *Acc15* for evaluating template matching accuracy and classification. The *VSD*-score, as proposed in [22], is a standard metric for evaluating 6*D* object pose estimation accuracy.

Acc15 This metric is introduced in [31]. It represents the accumulated true positive rate for matched templates that are below 15deg rotational error with respect to the object class and ground truth rotation of the query crop:

$$Acc15 = \sum_{n=1}^{n} \begin{cases} 1 & \text{if } \arccos \dfrac{R_q \times R_t}{\|R_q\|_2 \cdot \|R_t\|_2} < 15\text{deg} \\ & \text{and } C_q == C_t, \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where n refers to the number of query crops, R_q and R_t are the three-dimensional rotation vectors, and C_q and C_t are the object class of the queries' ground truth and the template, respectively. Consequently, matched templates with a rotation deviation of more than 15deg from the ground truth, or those having a different class than the query image, are considered false positives.

VSD This metric has been proposed in [22]. For each query object crop, the deviation of the estimated pose $\widehat{P}$ to the ground truth P is projected to a scalar value using:

$$e_{VSD} = \mathop{avg}_{p \in \widehat{V} \cup V} \begin{cases} 0 & \text{if } p \in \widehat{V} \cap V \wedge |\widehat{D}(p) - D(p)| < \tau, \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

where $\widehat{V}$ and V are sets of image pixels, $\widehat{D}$ and D are distance maps and τ is a misalignment tolerance with the standard value of 20*mm*. Distance maps are rendered and compared to the distance map of the test image to derive $\widehat{V}$ and V. Since $\widehat{P}$ and P need to represent 6*D* poses, including the 3*D* translation, we need to raise estimates to 6*D* by adpoting the strategy of [45,31]. Using the bounding box of the observation box_{obs}, and that of the template box_{tmp}, the corresponding intrinsics f_{obs} and f_{tmp} and the template distance to the camera z_{tmp}, derives the observed object's distance $\widehat{z}_{obs}$:

$$\widehat{z}_{obs} = z_{tmp} \cdot \frac{\left\|box^2_{tmp,x} \cdot box^2_{tmp,y}\right\|_2}{\left\|box^2_{obs,x} \cdot box^2_{obs,y}\right\|_2} \frac{f_{obs}}{f_{tmp}} \tag{7}$$

Using $\widehat{z}_{obs}$, the relative translation between the observation and template of the other two translation parameters are derived where $\bullet$ is a placeholder for x and y:

$$\Delta\bullet_{obs} = \frac{\left(box_{obs,\bullet} - c_{obs,\bullet}\right)\cdot\widehat{z}_{obs}}{f_{obs,\bullet}} - \frac{\left(box_{tmp,\bullet} - c_{tmp,\bullet}\right)\cdot\widehat{z}_{tmp}}{f_{tmp,\bullet}} \tag{8}$$

The 3D translation vector is ultimately composed as $t_{obs} = \{x_{tmp} + \Delta x_{obs}, y_{tmp} + \Delta y_{obs}, \widehat{z}_{obs}\}$.

The *VSD*-score is then defined as:

$$VSD = \sum_{n=1}^{n} \frac{1}{n} \begin{cases} 1 & e_{VSD,n} < 0.3, \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where n again refers to the number of the query sample in an evaluated test set.

4.2. Implementation details

This sections outlines the baseline method for comparing ViT to CNN-based template matching. Following that the training procedure and the network architecture are detailed.

Baseline method For demonstrating the difference of CNNs and ViTs for self-supervised matching of real query crops to synthetic templates, the baseline method of [31] is modified. In order to provide a fair comparison, all results are generated comparing backbones with a similar number of trainable parameters: ResNet50 [18] with 23M and ViT-s [48] with 21M parameters, pre-trained in a self-supervised manner [5] on [39].

Optimizer Setting AdamW [53] is used as the optimizer. The batch size is set to 16, which is also the case for the reference method [31]. The ViT networks are only trained for five epochs, as compared to the baseline, which is trained for 20 epochs. The linear scaling rule $lr = lr_b \cdot batch\_size/256$[14] is adopted for choosing the learning rate. A grid search was used to determine the base learning rate (lr_b) of $2.5\cdot10^{-5}$. No learning rate scheduling is used. Cosine weight decay scheduling, starting at 0.04 and ending at 0.4 after two epochs, is employed.

The input image size is 224 × 224 and the template's mask size 14 × 14. A patch size of 16 is used for input image tokenization. A single linear layer is used to project the backbone feature size of 384 to 32. This stands in contrast to works like [5,8,7], where multi-layered high-dimensional projectors are used. The input to the projection head is normalized using batch normalization [26]. The output of the projector is normalized using [2]. Section 4.5 ablates mask and descriptor size as well as the choice for the projection head.

4.3. Main results

This section presents experiments comparing ResNet50 [18] as feature extractor to ViT-s [48]. Evaluations are provided comparing our approach to the state of the art for 3D template matching.

4.3.1. Results on LM/LM-O

Table 2 compares the results of the presented approach to those of [51,3,31] for template matching on LM and LM-O. The true positive rates of matched templates with respect to object class and rotational error below 15deg (*Acc15*) as defined in [51] are shown. We follow the paradigm of [31] and report the results of the best-performing epoch during fine-tuning. The results show that using ViTs as feature extractor consistently outperforms the CNN approach for objects seen and unseen during training. Both, conceptually similar approaches, use backbones with a comparable amount of parameters, ResNet50 [18] with 23M and ViT-s [48] with 21M. It has to be mentioned that the method of [31] is fine-tuned for 20 epochs while the ViTs are fine-tuned for only 5.

Fig. 2 shows a detailed comparison for the individual data splits of LM and LM-O using ResNet50 [18] and ViT-small [48] as feature extractors for template matching. Generally, ViT-s improves in pose estimation with respect to all rotational error thresholds on all splits except the seen LM split 3, unseen LM-O split 2 and seen LM-O split 3.

Table 2
Comparison on LM/LM-O. Amount of true poses for a rotational error threshold of 15deg (*Acc15*[51]) for objects seen and unseen during training, see Table 1. The compared backbones have similar parameters, 23M for ResNet50 [18] and 21M for ViT-s [48]. Results for the methods indicated with † are taken from [31].

		seen		unseen	
Method	Backbone	LM	LM-O	LM	LM-O
[51]†	RN50[18]	98.1	67.5	45.1	29.9
[3]†	RN50[18]	96.1	64.7	44.3	29.1
[31]	RN50[18]	99.1	79.4	93.5	76.3
Ours	ViT-s[48]	**99.8**	**82.2**	**96.4**	**80.2**

4.3.2. Results on T-LESS

Table 3 compares the proposed approach to the approaches of [31,43,45]. We follow the evaluation paradigm of [43] and report the VSD-score [22] using the standard thresholds and the ground truth bounding box as basis for translation estimation. We report the performance after one epoch of fine-tuning as compared to the 25 epochs for [31]. The results show that our approach, using ViT-small [48] as feature extractor, consistently outperforms the competing approaches for objects seen and unseen during training. Especially relevant is the comparison to the conceptually similar approach of [31], which again uses ResNet50 [18] as backbone. These results show that ViTs work well for industrial objects of T-LESS, resulting in similar pose estimation accuracy for seen and unseen objects.

4.4. Feature extractor fine-tuning

This section presents results using only ImageNet- pretrained ViTs as feature extractor. In order to use the pre-trained backbone without fine-tuning, the last linear projection layer is discarded. The output dimensionality per feature map location is 384. Table 4 compares the presented approach with and without fine-tuning on LM, LM-O and T-LESS. The pre-trained ViT-s demonstrate tremendous generality with respect to feature embedding. On the LM and LM-O datasets, the matching accuracy using *Acc15* is higher than that of [51,3] as shown in Table 2. Yet, fine-tuning improves for all test cases. The matching accuracy on both seen and unseen T-LESS sets, evaluated using the *VSD*-metric, is higher than for all methods compared against in Table 3 even without fine-tuning. The presented evaluation shows that ViTs pre-trained in a self-supervised fashion learn features that translate well to new tasks with a large shift in object categories even without fine-tuning.

4.5. Ablation study

This sections discusses the difference in output space size and descriptor size for CNNs and ViTs. ViT and CNN approaches benefit from multi-layered, high-dimensional projection heads [8,5,7,15]. Ultimately, we present experiments on the influence of the projection head on our approach and additional architectural choices.

4.5.1. Descriptor size

The left plot of Fig. 3 evaluates the influence of the descriptor size on the presented approach and the ResNet50 baseline on the seen and unseen sets of LM. The cumulative rotational error on LM decreases steadily with increasing descriptor size when using ResNet50. The optimal dimensionality is 16 for minimizing the rotational error for the unseen LM objects. While the descriptor size has a large influence on the seen LM set and even more on the unseen set, the behavior using ViT-s is

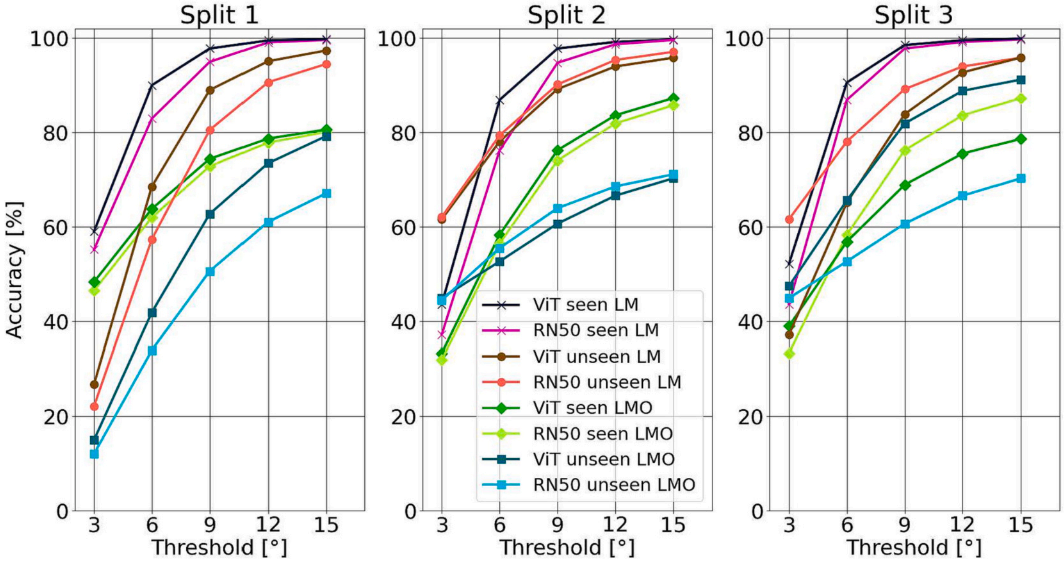

Fig. 2. Results on LM and LM-O splits in detail. Reported is the percentage of true poses for different rotational error thresholds of the CNN- and ViT-backbone for the seen and unseen object splits.

Table 3
Comparison on T-LESS. Results are presented using the *VSD*-score with the standard thresholds presented in [22].

Method	**seen:** Objects 1–18	**unseen:** Objects 19–30	Average
[45]	35.60	42.45	38.34
[43]	35.25	33.17	34.42
[31]	59.62	57.75	58.87
Ours	**70.65**	**68.03**	**69.71**

Table 4
Influence of fine-tuning. Result comparison for fine-tuning the ViT-s backbone versus only using the pre-trained feature extractor without fine-tuning. For LM and LM-O the *Acc*15-score [51] and for T-LESS the *VSD*-score [22] is reported.

Dataset	fine-tuning	**seen**	**unseen**
LM	×	81.3	85.1
	✓	**99.8**	**96.4**
LM-O	×	56.3	63.6
	✓	**82.2**	**80.2**
T-LESS	×	63.93	62.93
	✓	**70.65**	**68.03**

vastly different. For ViT-s, the descriptor dimensionality has little influence and leads to low errors over a broad range of dimensions for seen and unseen objects. While for ResNet50 the error progression is different for both sets, the dimensionality that minimizes the error on both sets is 32 when using ViT-s.

4.5.2. *Mask size*

The matching accuracy of the baseline method [31] increases when using spatially higher-dimensional feature maps since occlusion handling improves. In order to use larger feature maps for computing the template similarities, we adopt the projection head of the baseline. Instead of using two convolutional layers for downsampling, we employ two transposed convolutional layers for upsampling. Both are ReLU [30]-activated. The first layer projects the 384 dimensional feature vectors output by the backbone to 256 and the second layer projects to 32. Both convolutions apply no feature map padding, slide with a stride of one over the feature map and use the same kernel size, which is set depending on the desired mask size to either 3, 5, 7, 9, or 11. This projector replaces the projection head detailed in Section 4.2.

The right plot of Fig. 3 evaluates the influence of the mask size on the rotational error of the matched templates. For the presented comparison, the ResNet50 baseline approach [31] uses a descriptor size of 16 and ViT-s is used with a descriptor size of 32. With the ResNet50 backbone, the rotational error reduces with increasing mask size for both the the seen and unseen objects. However, using the proposed ViT-s approach again results in vastly different behavior. While the influence of the mask size is negligible for the seen objects, the rotational error for the novel objects increases significantly when increasing the mask size. This indicates that ViT-s learns relevant features for the seen objects during fine-tuning with projection heads with larger spatial output. As such, the template matching accuracy remains constant. However, increasing the feature map size used for matching is detrimental for novel objects. This correlates with the results presented in Section 4.4, which indicate that ViTs already generalize well without fine-tuning. The feature projection learned by a projection head with increased spatial output is less general and thus increases the template matching error for novel objects.

4.5.3. *Network architecture design*

This section ablates different aspects of network design choices when using self-supervised learning frameworks. We investigate patch embedding and projection head design.

Projection Head The works of [7,15,8] use high-dimensional, multi-layered projection heads to project the feature output of the backbone to the desired dimensionality. The work of [7] uses a two-layered MLP with the first ReLU [30] and the second layer linearly activated. The work of [15,5] both use three-layered MLPs but with different versions. The latter uses GELU [19]-activated hidden layers and weight normalization [40]. In [8], the projection head of [15] and the prediction head of [7] are combined. Features are normalized using batch normalization [26] and hidden layers are ReLU-activated. We compare using these projection heads to using no head or a single linear layer as head. Since using no head requires using the backbone's output as it is, the descriptor dimensionality per feature map location is 384. For all the evaluated projection heads, a hidden dimension of four times the output dimension of the previous stage and batch normalization are used. We have tested

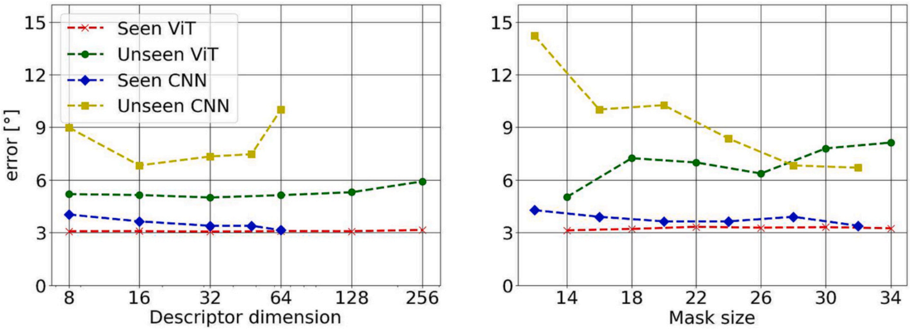

Fig. 3. Influence of the descriptor and mask size on LM seen and unseen The left plot shows the influence on the rotational error of the retrieved templates when using ResNet50 and ViT-s with different descriptor sizes. The mask size is set to 32 for ResNet50 and to 14 for ViT-s. The right plot shows the same comparison for different mask sizes. The descriptor size is set to 16 for ResNet50 and to 32 for ViT-s.

with and without using weight normalization as used by [5], both of which consistently lead to increased rotational error of the matched templates compared to batch normalization.

Table 5 compares the average rotational errors of different projection heads on LM and LM-O. The lowest error per set is indicated in bold and the highest is indicated with an underline. The lowest errors on seen and unseen LM, and unseen LM-O occur with heads with less layers while using no head leads to comparably high errors. When using projection heads, the highest errors over all sets occur using higher-dimensional heads. In general, for the seen objects the results are similar for all heads, however for unseen objects, projection heads with a smaller number of layers lead to less rotational error. This evaluation stands in contrast to self-supervised ViTs for classification that use projection heads with >= 3 layers and high dimensional hidden and last layers [8,5]. The choice of activation appears to have little influence, however, heads with a lower number of layers shows reduced error on unseen objects when using no activation function.

Patch embedding The authors of [8] propose to use random patch embedding to increase stability during training. We experiment with the initialization of the convolution layer used for patch embedding. The second column in Table 5 (p.e.) ablates the influence. Updating the pre-trained patch embedding layer during fine-tuning is referred to as learned (l). With a slight abuse of notation, we refer to not updating the patch embedding layer during fine-tuning as random (r). We observe a similar effect as in [8]: while the error difference for the seen objects is insignificant, using random patch embedding leads to significantly less error on the unseen objects.

Table 5
Network architecture. Reported is the average rotational error on LM and LM-O. The projection heads output a feature dimensionality of 32. When no head is used the standard ViT-s dimensionality of 384 is output. The column patch embedding (p.e.) indicates if the patch embedding layer is updated (l) or frozen (r) during fine-tuning.

			seen		unseen	
Head	p.e.	act.	LM	LM-O	LM	LM-O
none	l		3.14	10.95	7.80	15.44
	r		3.27	10.96	5.87	13.05
linear	l		3.07	11.05	5.39	12.83
	r		3.14	10.69	5.02	**11.78**
	r	ReLU	3.04	10.56	5.37	12.85
	r	GELU	3.12	10.28	4.98	12.75
[7]	r		3.11	10.47	**4.67**	12.20
[7]	r	ReLU	3.04	10.53	4.92	12.06
[7]	r	GELU	**3.02**	10.69	5.52	13.59
[15]	r		3.12	10.66	5.11	12.56
[15]	r	ReLU	3.17	**10.20**	5.14	14.49
[15]	r	GELU	3.04	10.70	5.17	13.56
[8]	r		3.07	10.92	5.28	12.67
[8]	r	ReLU	3.05	11.22	5.69	14.45
[8]	r	GELU	3.14	10.87	5.01	12.98

4.6. Self-attention

Figs. 4 and 5 visualize self-attention maps on the training and test sets of LM/LM-O and T-LESS, respectively. The same projection mechanism as in [5] is used.

On LM/LM-O (Fig. 4), ViT-s effectively learns to encode relevant features of the seen objects. The unseen test case shows that the learned self-attention not only transfers the concept of objectness to unseen objects but also manages to distinguish relevant from irrelevant feature map locations.

On T-Less (Fig. 5), object crops often show dataset objects in front or behind the query object, as is visualized in the seen and unseen test images. Cropping the feature map using the template's mask is important in order to improve matching accuracy.

5. Conclusion

This work presents diverse empirical analyses of ViTs for self-supervised template matching for 3*D* pose estimation. The presented findings are threefold. Firstly, using ViTs for deep template matching improves matching accuracy for seen and novel objects in comparison to CNNs. Secondly, using pre-trained ViTs in a zero-shot fashion, that is, without fine-tuning, already exhibits strong matching accuracy depending on the object set and metric used for evaluation. This even improves over using a similar, fine-tuned CNN-based approach. Thirdly, for the problem of self-supervised synthetic template to real query object matching, the network architecture is different to a comparable CNN approach and to self-supervised ViTs for image classification. In comparison to CNNs, ViTs benefit more from pre-training due to their feature extraction being more general. In comparison to self-supervised ViTs for image classification, large, multi-layered projector heads are detrimental to the matching accuracy on novel objects. We hypothesize that this occurs due to the stronger overfitting of deeper heads on the seen examples during fine-tuning, which in turn harms the generality of the features learned during pre-training. Future work will investigate how to effectively exploit the features learned during ViT pre-training.

CRediT authorship contribution statement

Stefan Thalhammer: Conceptualization, Investigation, Writing-original-draft. **Jean-Baptiste Weibel:** Writing-review-editing. **Markus Vincze:** Writing-review-editing, Supervision, Project-administration,

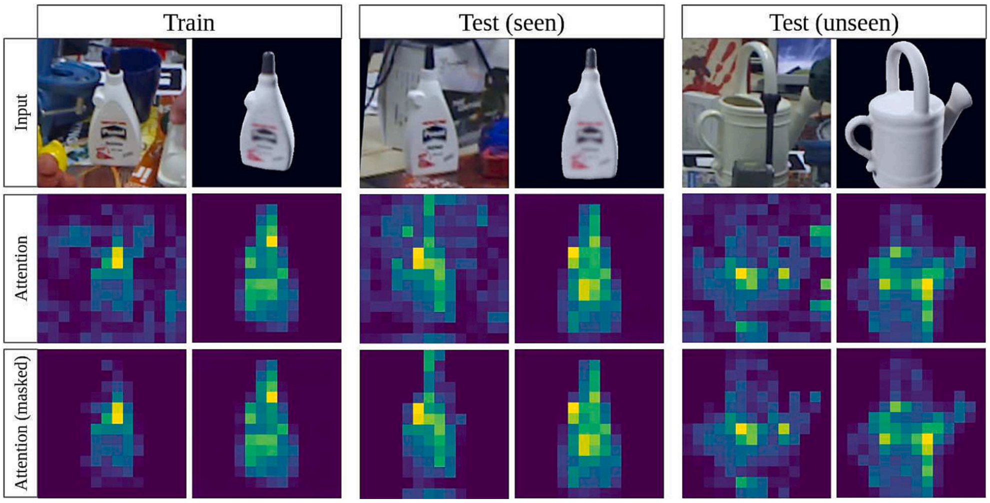

Fig. 4. Self-Attention on LM/LM-O Visualized is the self-attention of the first head of the last self-attention layer using the positional tokens as input.

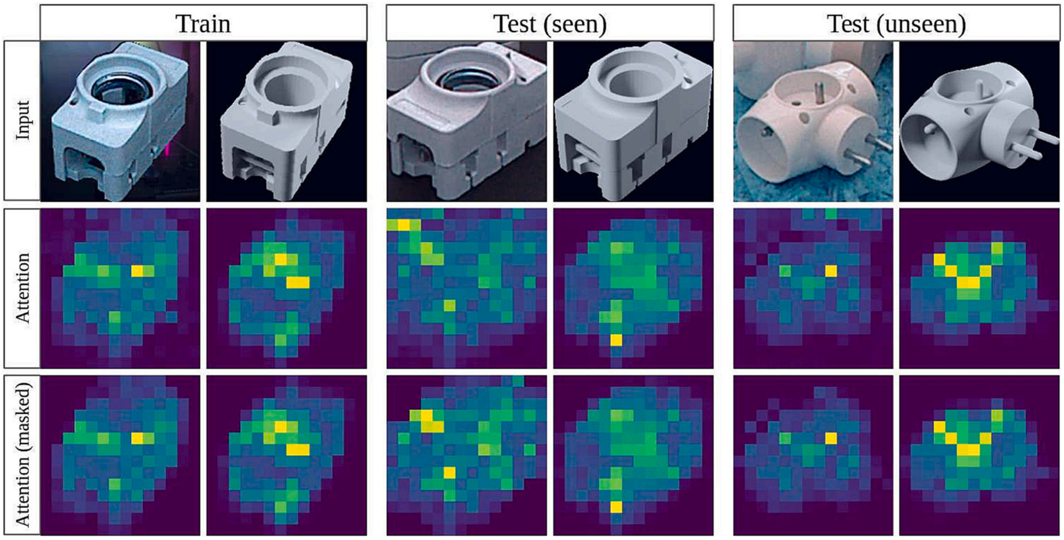

Fig. 5. Self-Attention on T-LESS Visualized is the self-attention of the first head of the last self-attention layer using the positional tokens as input.

Funding-acquisition. **Jose Garcia-Rodriguez:** Writing-review-editing, Supervision, Funding-acquisition.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: All authors are either employed by the TU Wien or the University of Alicante. Stefan Thalhammer reports financial support was provided by European Commission and equipment was provided by NVIDIA Corp. Jean-Baptiste Weibel reports financial support was provided by European Commission.

Data availability

I have shared the links to the data and code used in the Attach File step.

Acknowledgements

We gratefully acknowledge the support of the EU-program EC Horizon 2020 for Research and Innovation under grant agreement No. 101017089, project TraceBot and the NVIDIA Corporation for supporting this research by providing hardware resources.

References

[1] L. Aing, W.N. Lie, G.S. Lin, Faster and finer pose estimation for multiple instance objects in a single rgb image, Image Vis. Comput. 130 (2023), 104618.

[2] J.L. Ba, J.R. Kiros, G.E. Hinton, Layer normalization, arXiv preprint arXiv: 1607.06450 (2016).

[3] V. Balntas, A. Doumanoglou, C. Sahin, J. Sock, R. Kouskouridas, T.K. Kim, Pose guided rgbd feature learning for 3d object pose estimation, in: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 3876–3884.

[4] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, C. Rother, Learning 6d object pose estimation using 3d object coordinates, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Springer International Publishing, Cham, 2014, pp. 536–551.

[5] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, A. Joulin, Emerging properties in self-supervised vision transformers, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, 2021, pp. 9630–9640.

[6] G. Chechik, V. Sharma, U. Shalit, S. Bengio, Large scale online learning of image similarity through ranking, J. Mach. Learn. Res. 11 (2010) 1109–1135.

[7] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: Proceedings of the 37th International Conference on Machine Learning, ICML'20, JMLR.org, 2020.

[8] X. Chen, S. Xie, K. He, An empirical study of training self-supervised vision transformers, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, 2021, pp. 9620–9629.

[9] M.A. Dede, Y. Genc, Object aspect classification and 6dof pose estimation, Image Vis. Comput. 124 (2022), 104495.

[10] M. Denninger, M. Sundermeyer, D. Winkelbauer, D. Olefir, T. Hodan, Y. Zidan, M. Elbadrawy, M. Knauer, H. Katam, A. Lodhi, Blenderproc: Reducing the reality gap with photorealistic rendering, in: International Conference on Robotics: Sciene and Systems, (RSS 2020).

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16 × 16 words: Transformers for image recognition at scale (2021).

[12] B. Drost, M. Ulrich, N. Navab, S. Ilic, Model globally, match locally: Efficient and robust 3d object recognition, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 998–1005.

[13] D. Dwibedi, I. Misra, M. Hebert, Cut, paste and learn: Surprisingly easy synthesis for instance detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 1310–1319.

[14] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, K. He, Accurate, large minibatch sgd: Training imagenet in 1 hour, arXiv preprint arXiv: 1706.02677 (2017).

[15] J.B. Grill, F. Strub, F. Altché, C. Tallec, P.H. Richemond, E. Buchatskaya, C. Doersch, B.A. Pires, Z.D. Guo, M.G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, M. Valko, Bootstrap your own latent a new approach to self-supervised learning (2020).

[16] M. Gutmann, A. Hyvärinen, Noise-contrastive estimation: A new estimation principle for unnormalized statistical models, in: Y.W. Teh, M. Titterington (Eds.), Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, volume 9 of Proceedings of Machine Learning Research, PMLR, Chia Laguna Resort, Sardinia, Italy, 2010, pp. 297–304.

[17] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2 edition,, Cambridge University Press, 2004.

[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 770–778.

[19] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), arXiv preprint arXiv: 1606.08415 (2016).

[20] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, N. Navab, Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes, in: K.M. Lee, Y. Matsushita, J.M. Rehg, Z. Hu (Eds.), Computer Vision – ACCV 2012, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 548–562.

[21] T. Hodan, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, X. Zabulis, T-less: An rgb-d dataset for 6d pose estimation of texture-less objects, in: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 880–888.

[22] T. Hodaň, F. Michel, E. Brachmann, W. Kehl, A.G. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T.K. Kim, J. Matas, C. Rother, Bop: Benchmark for 6d object pose estimation, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision – ECCV 2018, Springer International Publishing, Cham, 2018, pp. 19–35.

[23] T. Hodaň, X. Zabulis, M. Lourakis, Š. Obdržálek, J. Matas, Detection and fine 3d pose estimation of texture-less objects in rgb-d images, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 4421–4428.

[24] T. Hou, A. Ahmadyan, L. Zhang, J. Wei, M. Grundmann, Mobilepose: Real-time pose estimation for unseen objects with weak shape supervision, arXiv preprint arXiv: 2003.03522 (2020).

[25] L. Huang, T. Hodan, L. Ma, L. Zhang, L. Tran, C. Twigg, P.C. Wu, J. Yuan, C. Keskin, R. Wang, Neural correspondence field for object pose estimation, in: S. Avidan, G. Brostow, M. Cissé, G.M. Farinella, T. Hassner (Eds.), Computer Vision – ECCV 2022, Springer Nature Switzerland, Cham, 2022, pp. 585–603.

[26] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: F. Bach, D. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 37, PMLR, Lille, France, 2015, pp. 448–456.

[27] Z. Jiang, X. Wang, X. Huang, H. Li, Triangulate geometric constraint combined with visual-flow fusion network for accurate 6dof pose estimation, Image Vis. Comput. 108 (2021), 104127.

[28] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (2017) 84–90.

[29] Y. Labbé, L. Manuelli, A. Mousavian, S. Tyree, S. Birchfield, J. Tremblay, J. Carpentier, M. Aubry, D. Fox, J. Sivic, Megapose: 6d pose estimation of novel objects via render & compare, in: K. Liu, D. Kulic, J. Ichnowski (Eds.), Proceedings of The 6th Conference on Robot Learning, Proceedings of Machine Learning Research, vol. 205, PMLR, 2023, pp. 715–725.

[30] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10, Omnipress, Madison, WI, USA, 2010, pp. 807–814.

[31] V.N. Nguyen, Y. Hu, Y. Xiao, M. Salzmann, V. Lepetit, Templates for 3d object pose estimation revisited: Generalization to new objects and robustness to occlusions, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2022, pp. 6761–6770.

[32] A.v.d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, arXiv preprint arXiv: 1807.03748 (2018).

[33] K. Park, T. Patten, M. Vincze, Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7667–7676.

[34] K. Park, T. Patten, M. Vincze, Neural object learning for 6d pose estimation using a few cluttered images, in: A. Vedaldi, H. Bischof, T. Brox, J.M. Frahm (Eds.), Computer Vision – ECCV 2020, Springer International Publishing, Cham, 2020, pp. 656–673.

[35] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, D. Tran, Image transformer, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 4055–4064.

[36] T. Patten, K. Park, M. Vincze, Dgcm-net: Dense geometrical correspondence matching network for incremental experience-based robotic grasping, Front. Robot. AI 7 (2020).

[37] S. Peng, Y. Liu, Q. Huang, X. Zhou, H. Bao, Pvnet: Pixel-wise voting network for 6dof pose estimation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019, pp. 4556–4565.

[38] A. Remus, S. D'Avella, F.D. Felice, P. Tripicchio, C.A. Avizzano, i2c-net: Using instance-level neural networks for monocular category-level 6d pose estimation, IEEE Robot. Autom. Lett. 8 (2023) 1515–1522.

[39] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (2015) 211–252.

[40] T. Salimans, D.P. Kingma, Weight normalization: A simple reparameterization to accelerate training of deep neural networks (2016) 901–909.

[41] I. Shugurov, F. Li, B. Busam, S. Ilic, Osop: A multi-stage one shot object pose estimation framework, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2022, pp. 6825–6834.

[42] H. Sun, T. Wang, E. Yu, A dynamic keypoint selection network for 6dof pose estimation, Image Vis. Comput. 118 (2022), 104372.

[43] M. Sundermeyer, M. Durner, E.Y. Puang, Z.C. Marton, N. Vaskevicius, K.O. Arras, R. Triebel, Multi-path learning for object pose estimation across domains, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 13913–13922.

[44] M. Sundermeyer, T. Hodaň, Y. Labbe, G. Wang, E. Brachmann, B. Drost, C. Rother, J. Matas, Bop challenge 2022 on detection, segmentation and pose estimation of specific rigid objects, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2023, pp. 2784–2793.

[45] M. Sundermeyer, Z.C. Marton, M. Durner, M. Brucker, R. Triebel, Implicit 3d orientation learning for 6d object detection from rgb images, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision – ECCV 2018, Springer International Publishing, Cham, 2018, pp. 712–729.

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

[47] S. Thalhammer, T. Patten, M. Vincze, Cope: End-to-end trainable constant runtime object pose estimation, in: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), IEEE, 2023, pp. 2860–2870.

[48] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jegou, Training data-efficient image transformers & distillation through attention, in: M. Meila, T. Zhang (Eds.), Proceedings of the 38th International Conference on Machine Learning, Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 10347–10357.

[49] G. Wang, F. Manhardt, F. Tombari, X. Ji, Gdr-net: Geometry-guided direct regression network for monocular 6d object pose estimation, in: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2021, pp. 16606–16616.

[50] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, L.J. Guibas, Normalized object coordinate space for category-level 6d object pose and size estimation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019, pp. 2637–2646.

[51] P. Wohlhart, V. Lepetit, Learning descriptors for object recognition and 3d pose estimation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 3109–3118.

[52] J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, A. Torralba, Sun database: Large-scale scene recognition from abbey to zoo, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 3485–3492.

[53] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, C.J. Hsieh, Large batch optimization for deep learning: Training bert in 76 minutes (2020).

[54] X. Zhang, Z. Jiang, H. Zhang, Real-time 6d pose estimation from a single rgb image, Image Vis. Comput. 89 (2019) 1–11.

[55] X. Zhang, Z. Jiang, H. Zhang, Out-of-region keypoint localization for 6d pose estimation, Image Vis. Comput. 93 (2020), 103854.