Learning Object Localization and 6D Pose Estimation from Simulation and Weakly Labeled Real Images

Jean-Philippe Mercier[1], Chaitanya Mitash[2], Philippe Giguère[1] and Abdeslam Boularias[2]

***Abstract*— Accurate pose estimation is often a requirement for robust robotic grasping and manipulation of objects placed in cluttered, tight environments, such as a shelf with multiple objects. When deep learning approaches are employed to perform this task, they typically require a large amount of training data. However, obtaining precise 6 degrees of freedom for ground-truth can be prohibitively expensive. This work therefore proposes an architecture and a training process to solve this issue. More precisely, we present a weak object detector that enables localizing objects and estimating their 6D poses in cluttered and occluded scenes. To minimize the human labor required for annotations, the proposed detector is trained with a combination of synthetic and a few weakly annotated real images (as little as 10 images per object), for which a human provides only a list of objects present in each image (no time-consuming annotations, such as bounding boxes, segmentation masks and object poses). To close the gap between real and synthetic images, we use multiple domain classifiers trained adversarially. During the inference phase, the resulting class-specific heatmaps of the weak detector are used to guide the search of 6D poses of objects. Our proposed approach is evaluated on several publicly available datasets for pose estimation. We also evaluated our model on classification and localization in unsupervised and semi-supervised settings. The results clearly indicate that this approach could provide an efficient way toward fully automating the training process of computer vision models used in robotics.**

I. Introduction

Robotic manipulators are increasingly deployed in challenging situations that include significant occlusion and clutter. Prime examples are warehouse automation and logistics, where such manipulators are tasked with picking up specific items from dense piles of a large variety of objects, as illustrated in Fig. 1. The difficult nature of this task was highlighted during the recent Amazon Robotics Challenges [1]. These robotic manipulation systems are generally endowed with a perception pipeline that starts with object recognition, followed by the object's six degrees-of-freedom (6D) pose estimation. It is known to to be a computationally challenging problem, largely due to the combinatorial nature of the corresponding global search problem. A typical strategy for pose estimation methods [2]–[5] consists in generating a large number of candidate 6D poses for each object in the scene and refining hypotheses with the Iterative Closest Point (ICP) [6] method or its variants. The computational efficiency of this search problem is directly affected by the number of pose hypotheses. Reducing the number of candidate poses is thus an essential step towards real-time grasping of objects.

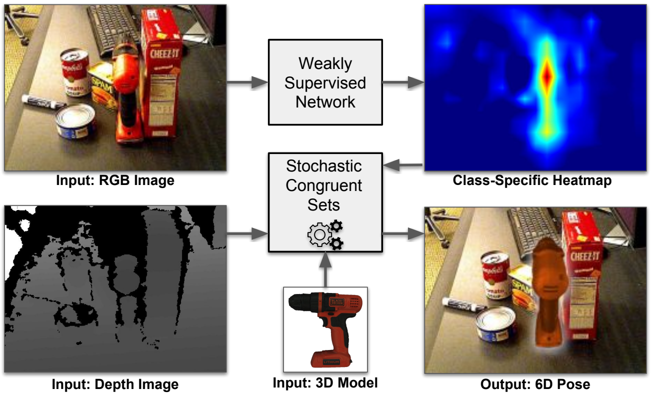


Fig. 1: Overview of our approach for 6D pose estimation at inference time. This figure shows the pipeline for the drill object of the YCB-video dataset [7]. A deep learning model is trained with *weakly annotated* images. Extracted class-specific heatmaps, along with 3D models and the depth image, guide the Stochastic Congruent Sets (StoCS) method [8] to estimate 6D object poses. Further details of the network are available in Section III.

Training Convolutional Neural Networks (CNN) for tasks such as object detection and segmentation [9]–[11] makes it possible to narrow down the regions that are used for searching for object poses in RGB-D images. However, CNNs typically require large amounts of annotated images to achieve a good performance. While such large datasets are publicly available for general-purpose computer vision, specialized datasets in certain areas such as robotics and medical image analysis tend to be significantly scarcer and time-consuming to obtain. In a warehouse context (our target context), new items are routinely added to inventories. It is thus impractical to collect and manually annotate a new dataset every time an inventory gets updated, particularly if it must cover all possible lighting and arrangement conditions that a robot may encounter during deployment. This is even more challenging if one wants this dataset to be collected by non-expert workers. The main goal of our approach is thus to reduce such a need for manual labeling, including completely eliminating bounding boxes, segmentation masks and 6D ground truth manual annotations.

Our first solution to reduce manual annotations is to leverage synthetic images generated with a CAD model rendered on diverse backgrounds. However, the visual features difference between real and synthetic images can be

[1] Laval University, Quebec, Canada.
jean-philippe.mercier.2@ulaval.ca, philippe.giguere@ift.ulaval.ca
[2] Rutgers University, NJ, USA
{cm1074,ab1544}@rutgers.edu.

large to the point of leading to poor performance on real objects. The problem of learning from data sampled from non-identical distributions is known as *domain adaptation*. Domain adaptation has been increasingly seen as a solution to bridge the gap between domains [12], [13]. Roughly speaking, domain adaptation tries to generalize the learning from a *source domain* to a *target domain*, or in our case, from synthetic to real images. Since labeled data in the target domain is unavailable or limited, the standard way is to train on labeled source data, while trying to minimize the distribution discrepancy between source and target domains.

While having a small labeled dataset on a target domain allows to boost performances, it may still require significant human effort for the annotations. Our second solution is to use *weakly supervised learning*, which significantly decreases annotation efforts, albeit with a reduced performance compared to fully-annotated images. Some methods [14], [15] have been shown to be able to retrieve a high level representation of the input data (such as object localization) while only being trained for object classification. To the best of our knowledge, this promising kind of approach has not yet been applied within a robotic manipulation context.

In this paper, we propose a two-step approach for 6D pose estimation, as shown in Fig. 1. First, we train a network for classification through domain adaptation, by using a combination of weakly labeled synthetic and real color images. During the inference phase, the weakly supervised network generates class-specific heatmaps that are subsequently refined with an independent 6D pose estimation method called Stochastic Congruent Sets (StoCS) [8]. Our complete method achieves competitive results on the YCB-video object dataset [7] and Occluded Linemod [3] while using only synthetic images and few weakly labeled real images (as little as 10) per object in training. We also empirically demonstrate that for our test case, using domain adaptation in semi-supervised settings is preferable than training in unsupervised settings and fine-tuning on available weakly labeled real images, a commonly-accepted strategy when only a few images from the target domain are available.

II. Related Works

In this paper, we aim at performing object localization and 6D pose estimation with a deep network, with minimal human labeling efforts. Our approach is based on training from synthetic and weakly labeled real images, via domain adaptation. These various concepts are discussed below.

6D Pose Estimation Recent literature in pose estimation focuses on learning to predict 6D poses using deep learning techniques. For example, [7] predicts separately the object center in images for translation and regresses over the quaternion representation for predicting the rotation. Another approach is to first predict 3D object coordinates, followed by a RANSAC-based scheme to predict the object's pose [4], [5]. Similarly, [5] uses geometric consistency to refine the predictions from the learned model. These methods, however, need access to several images that are manually labeled with the full object poses, which is time-consuming to acquire. Some other approaches make use of the object segmentation output to guide a global search process for estimating object poses in the scene [8], [16], [17]. Although the search process could compensate for errors in prediction when the segmentation module is trained with synthetic data, the domain gap could be large, and a computationally expensive search process may be needed to bridge this gap.

Learning with Synthetic Data Training with synthetic data has recently gained significant traction, as shown by the multiple synthetic datasets recently available [18]–[23], with some focusing on optimizing the realism of the generated images. While the latter can decrease to a certain degree the gap between real and synthetic images, it somehow defeats the purpose of using simulation as a cost-effective way to create training data. To circumvent this issue, [24], [25] proposed instead to create images using segmented object instances copied on real images. This type of approach, akin to data augmentation, is however limited to the number of object views and illuminations that are available in the original dataset. Recently, [26], [27] showed promising results by training object detectors with 3D models rendered in simulation with randomized parameters, such as lighting, number of objects, object poses, and backgrounds. While in [26] they only uses synthetic images in training, [27] demonstrated the benefits of fine-tuning on a limited set of real labeled images. The last one also showed that using photorealistic synthetic images does not necessarily improve object detection, compared to training on a less realistic synthetic dataset generated with randomized parameters.

Domain Adaptation Domain adaptation techniques [12], [13] can serve to decrease the distribution discrepancy between different domains, such as real vs. synthetic. The popular DANN [28] approach relies on two classifiers: one for the desired task, trained on labeled data from a source domain, and another one (called *domain classifier*) that classifies whether the input data is from the source or target domain. Both classifiers share the first part of the network, which acts as a feature extractor. The network is trained in an adversarial manner: domain classifier parameters are optimized to minimize the domain classification loss, and shared parameters are optimized to maximize the domain classification loss. It is possible to achieve this minimax optimization in a single step by using a gradient reversal layer that reverses the sign of the gradient between shared and non-shared parameters of the domain classifier. To the best of our knowledge, the present work is the first use a DANN-like approach for point-wise object localization, a fundamental problem in robotic manipulation.

Weakly Supervised Learning We are interested in weakly supervised learning with inexact supervision, for which only coarse-grained labels are available [29]. In [14], a network was trained only with weak image-level labels (classes that are present in images, but not their position) and max-pooling was used to retrieve approximate location of objects. The proposed *WILDCAT* model [15] performs classification and weakly supervised point-wise detection and segmentation. This architecture learns multiple localized features for each

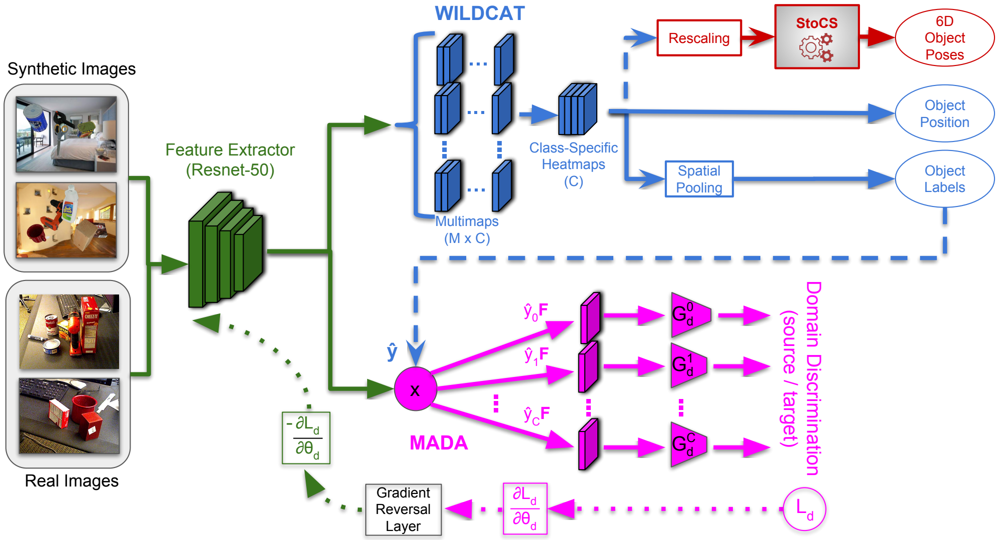


Fig. 2: Overview of the proposed approach for object localization and 6D pose estimation with domain adaptation, using a mix of synthetic images and weakly labeled real images.

class, and uses a spatial pooling strategy that generalizes to many ones (max pooling, global average pooling and negative evidence). In the present work, we push the paradigm of minimum human supervision *even further*. To this effect, we propose to train WILDCAT with synthetic images, in addition to weakly supervised real ones, and use MADA (a variant of DANN) for domain adaptation.

III. Proposed Approach

We present here our approach to object localization and 6D pose estimation. It is trained using a mix of synthetic and real images and only requires weak annotations (only class-presence) in both domains.

A. Overview

Figure 2 depicts an overview of our proposed system. It comprises *i*) a *ResNet-50* model pre-trained on *ImageNet* as a feature extractor (green), *ii*) a weak classifier inspired from the WILDCAT model [15] (blue), *iii*) the Stochastic Congruent Sets (*StoCS*) for 6D pose estimation (red) [8], and *iv*) the MADA domain adaptation network to bridge the gap between synthetic and real data. During the inference phase, the domain adaptation part of the network is discarded. Given a test image, class-specific heatmaps are generated by the network. These heatmaps indicate the most probable locations of each object in the image. This probability distribution is then fed to StoCS, a robust pose estimation algorithm that is specifically designed to deal with noisy localization. To force the feature extractor to extract similar features for both synthetic and real images, a MADA module (described below) is employed. MADA's purpose is to generate gradients during training (via a reversal layer) in order to improve the generalization capabilities of the feature extractor.

B. Synthetic Data Generation

For synthetic data generation, we used a modified version of the SIXD toolkit[1]. This toolkit generates color and depth images of 3D object models rendered on black backgrounds. Virtual camera viewpoints are sampled on spheres of different radii, following the approach described in [30]. We extended the toolkit with the functionality of rendering more than one object per image, and also used random backgrounds taken from the LSUN dataset [31]. Similarly to recent *domain randomization* techniques [32], we observed from our experiments that these simple modifications help transferring from simulation to real environments where there are multiple objects of interest, occlusions and diverse backgrounds. Figure 2 displays some examples of the generated synthetic images that we used to train our network.

C. Weakly Supervised Learning with WILDCAT

The images used for training our system are weakly labeled: only a list of object classes present in the image is provided. In order to recover localization from such weak labels, we leverage the WILDCAT architecture [15]. Indeed, WILDCAT is able to recover localization information through its high-level feature map, even though it is only trained with a classification loss. As a feature extractor, we employ a ResNet-50 (pretrained on ImageNet) for which the last layers (global average pooling and fully connected layers) are removed, as depicted in Figure 2. The WILDCAT architecture added on top of this ResNet-50 comprises three main modules: a *multimap transfer layer*, a *class pooling layer* and a *spatial pooling layer*. The *multimap transfer layer* consists of 1×1 convolutions that extracts M class-specific modalities per class C, with $M = 8$ as per the original paper [15]. The *class pooling* module is an average pooling layer that reduces the number of feature maps

[1] https://github.com/thodan/sixd_toolkit

from MC to C. Then, the *spatial pooling* module selects k regions with maximum/minimum activations to calculate scores for each class. The classification loss for this module is a multi-label one-versus-all loss based on max-entropy (*MultiLabelSoftMarginLoss* in PyTorch). The classification scores are then rescaled between 0 and 1 to cooperate with MADA.

D. Multi-Adversarial Domain Adaptation with MADA

We used the *Multi-Adversarial Domain Adaptation* (MADA) approach [33] to bridge the "reality gap". MADA extends the *Domain Adversarial Networks* (DANN) approach [28] by using one domain discriminator per class, instead of a single global discriminator as in the original version of DANN [28]. Having one discriminator per class has been found to help aligning class-specific features between domains. In MADA, the loss L_d for the K domain discriminators and input $\mathbf{x_i}$ is defined as:

$$L_d = \frac{1}{n}\sum_{k=1}^{K}\sum_{\mathbf{x_i}\in D_s\cup D_t} L_d^k\Big(G_d^k\big(\hat{y}_i^k G_f(\mathbf{x_i})\big), d_i\Big), \qquad (1)$$

wherein $i \in \{1,\ldots,n\}$, and $n = n_s + n_t$ is the total number of training images in source domain D_s (synthetic images) and the target domain D_t (real images). G_f is the feature extractor (the same for both domains), $\hat{y}_i^k$ is the probability of label k for image $\mathbf{x_i}$. This probability $\hat{y}_i^k$ is the output of the weak classifier WILDCAT. G_d^k is the k-th domain discriminator and L_d^k is its cross-entropy loss, given the ground truth domain $d_i \in \{\text{synthetic}, \text{real}\}$ of image $\mathbf{x_i}$. Our global objective function is:

$$C = \frac{1}{n}\sum_{\mathbf{x_i}\in D} L_y\Big(G_y\big(G_f(\mathbf{x_i})\big), y_i\Big) - \lambda L_d \quad , \qquad (2)$$

where L_y is the classification loss, L_d the domain loss and λ has been found to work well with a value of 0.5. The heat-map probability distribution extracted from WILDCAT is used to guide the StoCS algorithm in its search for 6D poses, as explained in the next section.

E. Pose Estimation with Stochastic Congruent Sets (StoCS)

The StoCS method [8] is a robust pose estimator that predicts the 6D pose of an object in a depth image from its 3D model and a probability heatmap. We employ a min-max normalization on the class-specific heatmaps of the Wildcat network, transforming them into a probability heatmaps w_{p_i}, using the per-class minimum (w_{min}) and maximum (w_{max}) values:

$$\pi_{p_i \to O_k} = \frac{w_{p_i} - w_{min}}{w_{max} - w_{min}}. \qquad (3)$$

This generates a heatmap providing the probability π of an object O_k being located at a given pixel p_i. The StoCS algorithm then follows the paradigm of a randomized alignment technique. It does so by iteratively sampling a set of four points, called a base B, on the point cloud S and finds corresponding set of points on the object model M. Each corresponding set of four points defines a rigid transformation T, for which an alignment score is computed between the transformed model cloud and the heatmap for that object. The optimization criteria is defined as

$$T_{opt} = arg\,max_T \sum_{m_i \in M_k} f(m_i, T, S_k), \qquad (4)$$

$$f(m_i, T, S_k) = \pi_k(s*), if \mid T(m_i) - s* \mid < \delta_s. \qquad (5)$$

The base sampling process in this algorithm considers the joint probability of all four points belonging to the object in question, given as

$$Pr(B \to O_k) = \frac{1}{Z}\prod_{i=1}^{4}\{\phi_{node}(b_i)\prod_{j=1}^{j<i}\phi_{edge}(b_i, b_j)\}. \qquad (6)$$

where ϕ_{node} is obtained from the probability heatmap and ϕ_{edge} is computed based on the point-pair features of the pre-processed object model. Thus, the method combines the normalized output of the Wildcat network with the geometric model of objects to obtain base samples which belong to the object with high probability.

In the next two Sections, we demonstrate the usefulness of our approach. First in Section IV, we quantify the importance of each component (Wildcat, MADA) in order to train a network that generates *relevant* feature maps from weakly labeled images. In Section V, we then evaluate the performance of using these heatmaps with StoCS for rapid 6D pose estimation, which is the final goal of our paper.

IV. Weakly Supervised Learning Experiments for object detection and classification

In this first experimental section, we perform an ablation study to evaluate the impact of various components for classification and point-wise localization. We first tested our approach without any human labeling, as a baseline. We then evaluated the gain obtained by employing various numbers of weakly labeled images for four semi-supervised strategies.

We performed these evaluations on the YCB-video dataset [7]. This dataset contains 21 objects with available 3D models. It also has full annotations for detection and pose estimation on 113,198 training images and 20,531 test images. A subset of 2,949 test images (keyframes) is also available. Our results are reported for this more challenging subset, since most images in the bigger test set are video frames that are too similar and would report optimistic results.

For these experiments, we trained our network for 20 epochs (500 iterations per epoch) with a batch size of 4 images per domain. We used stochastic gradient descent with a learning rate of 0.001 (decay of 0.1 at epochs 10 and 16) and a Nesterov momentum of 0.9. The ResNet-50 was pre-trained on ImageNet and the weights of the first two blocks were frozen.

A. Unsupervised Domain Adaptation

For this experiment, we trained our model with weakly labeled synthetic images (WS) and unlabeled real images

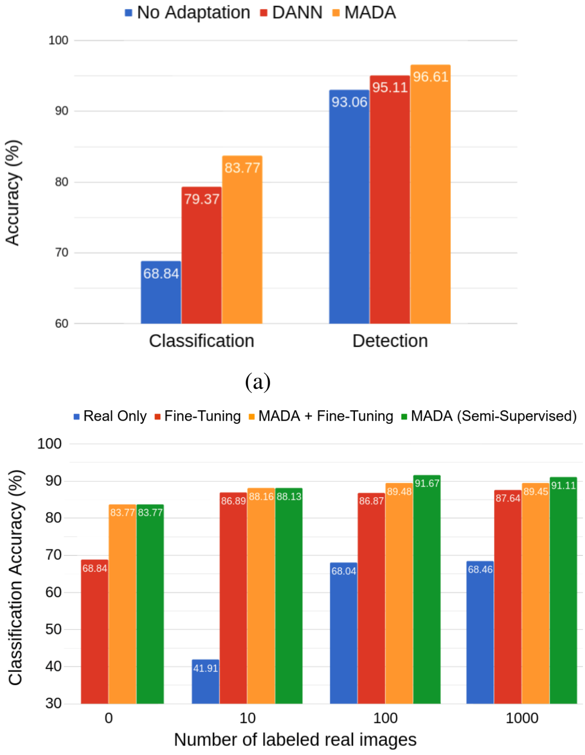
(UR). We tested three architecture configurations of domain adaptation: 1) without any domain adaptation module (WILDCAT model trained on WS), 2) with DANN (WS+UR) and 3) with MADA (WS+UR). We evaluated each of these configurations for both classification and detection. For classification, we used the accuracy metric to evaluate our model's capacity to discriminate which objects are in the image. We used a threshold of 0.5 on classification scores to predict the presence or absence of an object. For detection, we employed the point-wise localization metric [14], which is a standard metric to evaluate the ability of weakly supervised networks to localize objects. For each object in the image, the maximum value in their class-specific heatmap was used to retrieve the corresponding pixel in the original image. If this pixel is located inside the bounding box of the object of interest, it is counted as a good detection. Since the class-specific heatmap is a reduced scale of the input image due to pooling, a tolerance equal to the scale factor was added to the bounding box. In our case, a location in the class-specific heatmaps corresponds to a region of 32 pixels in the original image. In Figure 3a, we report the average scores of the last 5 epochs over 3 independent random runs for each network variation. These results *a)* confirm the importance of employing a domain adaptation strategy to bridge the reality gap, and *b)* the necessity of having one domain discriminator G_d^k for each of the X objects in the YCB database (MADA), instead of a single one (DANN). Next, we evaluate the gains obtained by employing weakly-annotated real images.

B. Semi-Supervised Domain Adaptation

A significant challenge for agile deployment of robots in industrial environments is that they ideally should be trained with limited annotated data, both in terms of numbers of images and of their extensiveness of labeling (no pose information, just class). We thus evaluated the performance of four different strategies as a function of the number of such weakly-labeled real images:

1) Without domain adaptation:
   a) Real Only: Trained only on weakly labeled real images,
   b) Fine-Tuning: Trained on synthetic images and then fine-tuned on weakly labeled real images,
2) With domain adaptation:
   a) Fine-Tuning: Trained on synthetic images and then fine-tuned on weakly labeled real images,
   b) Semi-Supervised: Trained with synthetic images and weakly labeled real images simultaneously.

For 1.a and 1.b, we validate that using fine-tuning on a network pre-trained with synthetic data is preferable to training directly on real images. For 2.a and 2.b, we compare the performance of our approach trained with fine-tuning, and in a semi-supervised way (using images from both domains at the same time). We are particularly interested in comparing the two approaches 2.a and 2.b, since [36] achieved the lowest error rate compared to any other semi-supervised approach by only using fine-tuning.



(a)

(b)

Fig. 3: Performance analysis. In (a), we compare classification accuracy and point-wise detection when no label on real images are available. In (b), we compare the performance of different training processes when different numbers of real images are weakly labeled.

Our results are summarized in Figure 3b. From them, we conclude that training with synthetic images improves classification accuracy drastically, especially when few labels are available. Also, our approach performs slightly better when trained in a semi-supervised setting (2.b) than with a fine-tuning approach (2.a), which is contrary to [36].

In this Section, we justified our architecture, as well as the training technique employed, to create a network capable of performing object identification and localisation through weak learning. In the next Section, we demonstrate how the feature maps extracted by our network can be employed to perform precise 6 DoF object pose estimation via StoCS.

V. 6D Pose Estimation Experiments

We evaluated our full approach for 6D pose estimation on YCB-video [7] and Occluded Linemod [3] datasets. We used the most common metrics to compare with similar methods. The average distance (ADD) metric [37] measures the average distance between the pairwise 3D model points transformed by the ground truth and predicted pose. For symmetric objects, the ADD-S metric measures the average distance using the closest point distance. Also, the visible surface discrepancy [38] compares the distance maps of rendered models for estimated and ground-truth poses.

Method	Modality	Supervision	Full Dataset	Accuracy (%) YCB-Video	Accuracy (%) Occluded Linemod
PoseCNN [7]	RGB	Pixelwise labels + 6D poses	Yes	75.9	24.9
PoseCNN+ICP [7]	RGBD	Pixelwise labels + 6D poses	Yes	93.0	**78.0**
DeepHeatmaps [34]	RGB	Pixelwise labels + 6D poses	Yes	81.1	28.7
FCN + Drost et. al. [35]	RGBD	Pixelwise labels	Yes	84.0	-
FCN + StoCS [8]	RGBD	Pixelwise labels	Yes	90.1	-
Brachmann et al. [4]	RGBD	Pixelwise labels + 6D poses	Yes	-	56.6
Michel et. al. [5]	RGBD	Pixelwise labels + 6D poses	Yes	-	76.7
OURS	RGBD	Object classes	No (10 weakly labeled images)	88.7	68.8
OURS	RGBD	Object classes	Yes	90.2	-
OURS (multiscale inference)	RGBD	Object classes	No (10 weakly labeled images)	-	76.6
OURS (multiscale inference)	RGBD	Object classes	Yes	**93.6**	-

TABLE I: Area under the accuracy-threshold curve for 6D Pose estimation on YCB-Video dataset and Occluded Linemod.

We used the same training details mentionned in section IV. Since the network architecture is fully convolutional, we also added an experiment for which we combined the output of the network for 3 different scales of the input image (at test time only).

A. YCB-Video Dataset

This dataset comprises several frames from 92 video sequences of cluttered scenes created with 21 YCB objects. The training for competing methods [7], [34], [35] is performed using 113,199 frames from 80 video sequences with semantic (pixelwise) and pose labels. For our proposed approach, we used only 10 randomly sampled weakly annotated (class labels only) real images per object class combined with synthetic images. As in [7], we report the area under the curve (AUC) of the accuracy-threshold curve, using the ADD-S metric. Results are reported in Table I. Our proposed method achieves 88.67% accuracy with a limited number of weakly labeled images and up to 93.60% when using the full dataset with multiscale inference. It outperforms competing approaches, with the exception of PoseCNN+ICP, which performs similarly. However, our approach has a large computational advantage with an average runtime of 0.6 seconds per object as opposed to approximately 10 seconds per object for the modified-ICP refinement for PoseCNN. It also uses *a)* nearly a hundredfold less real data, and *b)* also only using the class labels. This results thus demonstrate that we can reach *fast* and *competitive* results without the need of 6D fully-annotated real datasets.

B. Occluded Linemod Dataset

This dataset contains 1215 frames from a single video sequence with pose labels for 9 objects from the LINEMOD dataset with high level of occlusion. Competing methods are trained using the standard LINEMOD dataset, which consists in average of 1220 images per object. In our case, we used 10 real random images per object (manually labelled) on top of the generated synthetic images, using the weak (class) labels only. As reported in Table I, our method achieved scores of 68.8% and 76.6% (multiscale) for the ADD evaluation metric and using a threshold of 10% of the 3D model diameter. These results compare with state-of-the-art methods while using less supervision and a fraction of training data. The multiscale variant (input image at 3 different resolutions) made our approach more robust to occlusions. We did not train with the full Linemod training dataset, since the dataset only has annotations for 1 object per image and our method requires the full list of objects that are in the image. Furthermore, we evaluated our approach on the 6D pose estimation benchmark [38] using the visual discrepency metric. We evaluated our network with multiscale inference and we can see in Table II that we are among the top 3 for the recall score while being the fastest. We also tested the effect of combining ICP with StoCS. At the cost of more processing time, we obtain the best performance among the methods that were evaluated on the benchmark.

Method	Recall Score (%)	Time (s)
Vidal-18 [39]	59.3	4.7
Drost-10 [35]	55.4	2.3
Brachmann-16 [40]	52.0	4.4
Hodan-15 [41]	51.4	13.5
Brachmann-14 [4]	41.5	1.4
Buch-17-ppfh [42]	37.0	14.2
Kehl-16 [43]	33.9	1.8
OURS (MS)	55.2	0.6
OURS (MS) + ICP	**62.1**	6.4

TABLE II: Visual discrepency recall scores (%) (correct pose estimation) for $\tau = 20\text{mm}$ and $\theta = 0.3$ on Occluded Linemod, based on the 6D pose estimation benchmark [38]. MS means multiscale.

VI. Conclusion

In this paper, we explored the problem of 6D pose estimation in the context of limited annotated training datasets. To this effect, we demonstrated that the output of a weakly-trained network is sufficiently rich to perform full 6D pose estimation. Pose estimation experiments on two datasets showed that our approach is competitive with recent approaches (such as PoseCNN), despite requiring *significantly less annotated images*. Most importantly, our annotation level requirement for real images is *much weaker*, as we only need a class label without any spatial information (either bounding box or full 6D ground truth). In this end, this makes our approach compatible with an agile automated warehouse, where new objects to be manipulated are constantly introduced in a training database by non-expert employees.

References

[1] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Osada, A. Rodriguez, J. Romano, and P. Wurman, "Analysis and Observations From the First Amazon Picking Challenge," *Transactions on Automation Science and Engineering*, 2016.

[2] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige, "Going further with point pair features," in *European Conference on Computer Vision*. Springer, 2016, pp. 834–848.

[3] A. Krull, E. Brachmann, F. Michel, M. Ying Yang, S. Gumhold, and C. Rother, "Learning analysis-by-synthesis for 6d pose estimation in rgb-d images," in *International Conference on Computer Vision*, 2015, pp. 954–962.

[4] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6d object pose estimation using 3d object coordinates," in *European Conference on Computer Vision*. Springer, 2014, pp. 536–551.

[5] F. Michel, A. Kirillov, E. Brachmann, A. Krull, S. Gumhold, B. Savchynskyy, and C. Rother, "Global hypothesis generation for 6d object pose estimation," in *Conference on Computer Vision and Pattern Recognition*, 2017, pp. 462–471.

[6] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–607.

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

[8] C. Mitash, A. Boularias, and K. Bekris, "Robust 6d object pose estimation with stochastic congruent sets," *arXiv preprint arXiv:1805.06324*, 2018.

[9] A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker Jr, A. Rodriguez, and J. Xiao, "Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge," in *International Conference on Robotics and Automation*, 2017.

[10] C. Hernandez, M. Bharatheesha, W. Ko, H. Gaiser, J. Tan, K. van Deurzen, M. de Vries, B. Van Mil, J. van Egmond, R. Burger *et al.*, "Team delft's robot winner of the amazon picking challenge 2016," in *Robot World Cup*. Springer, 2016, pp. 613–624.

[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[12] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *arXiv preprint arXiv:1802.03601*, 2018.

[13] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," *arXiv preprint arXiv:1702.05374*, 2017.

[14] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?-weakly-supervised learning with convolutional neural networks," in *Conference on Computer Vision and Pattern Recognition*, 2015, pp. 685–694.

[15] T. Durand, T. Mordan, N. Thome, and M. Cord, "Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2017.

[16] V. Narayanan and M. Likhachev, "Discriminatively-guided deliberative perception for pose estimation of multiple 3d object instances." in *Robotics: Science and Systems*, 2016.

[17] C. Mitash, A. Boularias, and K. E. Bekris, "Improving 6d pose estimation of objects in clutter via physics-aware monte carlo tree search," *arXiv preprint arXiv:1710.08577*, 2017.

[18] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," *arXiv preprint arXiv:1605.06457*, 2016.

[19] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040–4048.

[20] W. Qiu and A. Yuille, "Unrealcv: Connecting computer vision to unreal engine," in *European Conference on Computer Vision*. Springer, 2016, pp. 909–916.

[21] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3234–3243.

[22] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" in *International Conference on Robotics and Automation*. IEEE, 2017, pp. 746–753.

[23] S. R. Richter, Z. Hayder, and V. Koltun, "Playing for benchmarks," in *International Conference on Computer Vision*, 2017.

[24] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," *ArXiv*, vol. 1, no. 2, p. 3, 2017.

[25] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka, "Synthesizing training data for object detection in indoor scenes," *arXiv preprint arXiv:1702.07836*, 2017.

[26] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige, "On pre-trained image features and synthetic images for deep learning," *arXiv preprint arXiv:1710.10710*, 2017.

[27] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," *arXiv preprint arXiv:1804.06516*, 2018.

[28] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[29] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, 2017.

[30] S. Hinterstoisser, S. Benhimane, V. Lepetit, P. Fua, and N. Navab, "Simultaneous recognition and homography extraction of local patches with a simple linear classifier." in *BMVC*, 2008, pp. 1–10.

[31] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, "LSUN: construction of a large-scale image dataset using deep learning with humans in the loop," *CoRR*, vol. abs/1506.03365, 2015.

[32] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Intelligent Robots and Systems*. IEEE, 2017, pp. 23–30.

[33] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *AAAI Conference on Artificial Intelligence*, 2018.

[34] M. Oberweger, M. Rad, and V. Lepetit, "Making deep heatmaps robust to partial occlusions for 3d object pose estimation," *arXiv preprint arXiv:1804.03959*, 2018.

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

[36] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, and I. J. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," *arXiv preprint arXiv:1804.09170*, 2018.

[37] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Asian conference on computer vision*. Springer, 2012, pp. 548–562.

[38] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis *et al.*, "Bop: benchmark for 6d object pose estimation," in *European Conference on Computer Vision*, 2018, pp. 19–34.

[39] J. Vidal, C.-Y. Lin, and R. Martí, "6d pose estimation using an improved method based on point pair features," in *International Conference on Control, Automation and Robotics*. IEEE, 2018, pp. 405–409.

[40] E. Brachmann, F. Michel, A. Krull, M. Ying Yang, S. Gumhold *et al.*, "Uncertainty-driven 6d pose estimation of objects and scenes from a single rgb image," in *Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3364–3372.

[41] T. Hodaň, X. Zabulis, M. Lourakis, Š. Obdržálek, and J. Matas, "Detection and fine 3d pose estimation of texture-less objects in rgb-d images," in *Conference on Intelligent Robots and Systems*. IEEE, 2015, pp. 4421–4428.

[42] A. G. Buch, L. Kiforenko, and D. Kraft, "Rotational subgroup voting and pose clustering for robust 3d object recognition," in *International Conference on Computer Vision*. IEEE, 2017, pp. 4137–4145.

[43] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab, "Deep learning of local rgb-d patches for 3d object detection and 6d pose estimation," in *European Conference on Computer Vision*. Springer, 2016, pp. 205–220.