

Disclaimer:

This work has been accepted for publication in the Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems

link: <https://www.iros2018.org/>

Copyright:

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Kitting in the Wild through Online Domain Adaptation

Massimiliano Mancini^{*1,2}, Hakan Karaoguz^{*3}, Elisa Ricci^{2,4}, Patric Jensfelt³, Barbara Caputo^{1,5}

Abstract—Technological developments call for increasing perception and action capabilities of robots. Among other skills, vision systems that can adapt to any possible change in the working conditions are needed. Since these conditions are unpredictable, we need benchmarks which allow to assess the generalization and robustness capabilities of our visual recognition algorithms. In this work we focus on robotic kitting in unconstrained scenarios. As a first contribution, we present a new visual dataset for the kitting task. Differently from standard object recognition datasets, we provide images of the same objects acquired under various conditions where camera, illumination and background are changed. This novel dataset allows for testing the robustness of robot visual recognition algorithms to a series of different *domain shifts* both in isolation and unified. Our second contribution is a novel online adaptation algorithm for deep models, based on batch-normalization layers, which allows to continuously adapt a model to the current working conditions. Differently from standard domain adaptation algorithms, it does not require any image from the target domain at training time. We benchmark the performance of the algorithm on the proposed dataset, showing its capability to fill the gap between the performances of a standard architecture and its counterpart adapted offline to the given target domain.

I. INTRODUCTION

Robot technology is already an integral part of the manufacturing industry. Robots are currently being used in variety of tasks such as machine loading, part inspection, bin picking, kitting, and assembly. In this work, we consider the task of kitting which is the process of grouping related parts such as gathering components of a personal computer (PC) into one bin for assembly [1]. The kitting task requires the recognition of the parts in the environment, the ability to pick objects from the bins and placing them at the correct location [2]. All of these subtasks are very challenging on their own but the recognition of the parts is crucial for the robot to sequentially perform the other subtasks. Already in today’s factory settings, object recognition tasks possess challenges such as environmental effects (illumination, viewpoint, etc), varying object material properties and cluttered scenes [3]. In order to simplify the recognition task, some approaches use machine vision in rather isolated settings for decreasing

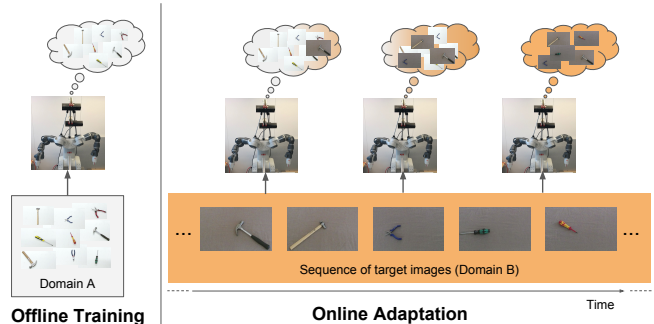


Fig. 1: Our approach for performing kitting in arbitrary conditions. Given a training set, we can train a robot vision model offline. As the robot performs the task, we gradually adapt the visual model to the current working conditions, in an online fashion and without requiring target data during the offline training phase.

the environmental variability [4]. Liu et. al [3] proposed a specially designed camera system and estimation based on 3D CAD models to robustly detect and verify the type and the pose of the object. Kaiba et. al. [5] proposed an interactive method where a remote human operator resolves ambiguities in the perception system. Unfortunately, none of the above methods are generic enough to be applied in a truly unconstrained setting. In this paper we are primarily concerned with solving the object recognition problem for kitting using vision in the wild, i.e. in non-isolated settings exhibiting large variations. Right now, most of the robots in manufacturing industry are operating in isolation, primarily because of safety concerns. However, many future scenarios have robots and humans working closer together, moving robots into new areas of applications, beyond mass production and preprogrammed behavior. For this to happen, not only safety, but perception will be a major challenge.

In the last years deep neural networks [6] have become the new dominant learning paradigm in visual recognition, establishing the new state of the art in various visual tasks such as object classification [7] and object detection [8]. Similarly deep architectures have been applied on real robots [9], [10], leading to significant improvements on a variety of robot vision tasks [11], [12], [13]. One known challenge with DNNs is that they are data hungry. This is particularly problematic for robotic scenarios where the data collection process can be costly or even unfeasible, thus the amount of training data is limited. Some authors proposed to overcome this issue by leveraging over synthetic data [14], but while this approach seems promising for depth data,

This work was partially supported by The Swedish Foundation for Strategic Research (SSF) and its Centre for Autonomous Systems and the project FACT (H.K., P. J.) and the ERC project RoboExNovo (M. M., B.C.).

¹M. Mancini and B. Caputo are with Sapienza University of Rome, Rome, Italy. {mancini, caputo}@diag.uniroma1.it

²M. Mancini and E. Ricci are with Fondazione Bruno Kessler, Trento, Italy. eliricci@fbk.eu

³H. Karaoguz and P. Jensfelt are with KTH Royal Institute of Technology, Stockholm, Sweden. {hkarao, patric}@kth.se

⁴E. Ricci is with University of Trento, Trento, Italy.

⁵B. Caputo is with Italian Institute of Technology, Milan, Italy.

* denotes equal contribution.

it is questionable if it will work on RGB images. Domain Adaptation (DA, [15]) attempts to circumvent this issue by adapting a model from a given domain for which sufficient training data is available, denoted as *source* domain, to a domain for which few or no labeled data are available, called the *target* domain. Despite the remarkable performances achieved by DA algorithms in computer vision [16], [17], and their growing popularity in robot vision [18] they require the presence of images from the target domain in advance during training. This is a huge limitation due to the likely unpredictable conditions of the environment in which a robot is employed.

This paper attempts to advance the state of the art in kitting in realistic deployments with two contributions. First, we propose a novel kitting dataset which contains images of objects taken under varying illumination, viewpoint and background conditions from a robotic platform. This dataset, that upon acceptance of the paper will be made publicly available through a dedicated website, provides the community with a novel tool for studying the robustness of robot vision algorithms to drastic changes in the appearance of the input images, and assess progress in the field. We are not aware of existing, publicly available kitting databases covering this range of visual variability.

Second, we propose a novel approach for achieving *online* adaptation of a deep model. Differently from classical DA approaches, our algorithm can adapt a deep model to any target domain on the fly, without requiring any target domain data before-hand. We benchmark the performance of our algorithm on the presented dataset, showing how this model is able to produce large improvements on the target domain performances compared to the base architecture trained on the source domain, and matching what would be achieved by having all data from the target available beforehand.

The outline for the rest of the paper is as follows. In Section II-B we give an overview of related work. In Section III we present the new dataset, describing the collection process and the data contained. Section IV describes our online DA method and Section V presents the result of our evaluation. Finally, Section VI gives conclusions and outlines avenues for future research.

II. RELATED WORK

A. Kitting task

Robotic kitting and bin picking are similar and well known problems. Several methods have been proposed to solve these problems by either using specialized setups [3], high-level frameworks [2], [4] or using human-robot collaboration [1], [5]. Liu et al. in [3] use a customized camera for extracting edges of the objects in a bin and then using shape-matching to detect objects and estimating their poses. The proposed algorithm is computationally efficient so that it can be deployed in real robot scenario. Holz et al. in [2] propose a high-level framework that is composed of individual modules such as object detection, planning etc. to automatically perform kitting task in real world scenarios. Similarly in [4] a high-level framework that combines virtual and real setups

is proposed. Virtual setup helps to optimize the actual system without any risk of collisions. Therefore the real system can be setup in more economical way with less number of actual trials. Banerjee et al. in [1] employ human-robot interaction for performing the kitting task. They first present a common ontology for representing all the required subtasks for kitting. Then these subtasks are optimally partitioned between the robots and humans to complete the whole task faster and with less failures. Kaipa et al. in [5] present a method where a remote human operator assists the robot for selecting the object when the automated perception system fails.

B. Online Domain Adaptation

Recent years have witnessed great advances in domain adaptation both in computer [19], [17], [16] and robot vision [18], [12], [20]. Adapting a model from one domain to another requires to bridge the gap between the two different distributions generating the data of different domains. In deep learning architectures this is usually achieved by minimizing the difference between the features produced by images of different domains, *e.g.* through domain confusion losses or by minimizing discrepancy measures [19], [21], [22]. Recently, it has been shown how batch-normalization (BN) layers [23] can be employed to match the source and target data distributions by applying different BN layers for each domain [16], [24], [17]. Our method develops from this last research trend, with BN statistics adapted to the images of the *experienced* novel domain. Differently from [16], [24], [17] it does not require any target domain data during training.

We would like to remark that, despite its simplicity, our approach is the first deep domain adaptation method that operates in an online setting, without requiring any prior about the target domain.

Another related research trend is domain generalization (DG). DG frameworks [25], [20] aim at generalizing a model from multiple, given, source domains, to any target domain, with no data of the target domain available at training time. Differently from these techniques, we assume to have only one source domain during training, without the need of multiple data acquisition and labeling processes.

III. KTH HANDTOOL DATASET

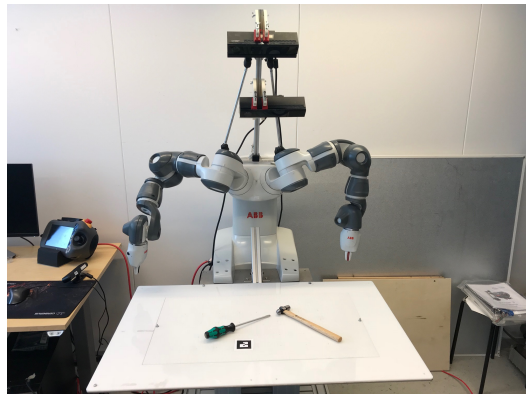


Fig. 2: The 2-arm stationary robot platform.

The KTH Handtool Dataset¹ is collected for evaluating the object recognition/detection performance of robot vision methods in varying viewpoint, illumination and background settings, all crucial abilities for robot kitting in unconstrained, real-world settings. Instead of having general household objects, the dataset consists of hand tools to represent a workshop setting in a factory. It consists of 9 different hand tools for 3 different categories; hammer, plier and screwdriver. The images are collected with a 2-arm stationary robot platform shown in Fig. 2. Dataset consists of 3 different illuminations, 2 different cameras (One Kinect camera and one webcam) with different viewpoints and 2 different background settings that correspond to 12 (3x3x2) domains in total. For each hand tool, approximately 40 images with different poses are collected for each camera and domain setting. Table I shows example images from different domains. In total, approximately 4500 RGB images are available in the dataset.

IV. ONLINE DOMAIN ADAPTATION

In this section we present our strategy for performing online domain adaptation by exploiting Batch Normalization (BN) [23]. After giving a formal definition of Domain Adaptation (section IV-A), we recall the basic principles of BN and discuss how it can be exploited into a neural architecture for reducing the domain shift (section IV-B). Then, we describe our approach for online adaptation of a deep model to novel domains (section IV-C).

A. Domain Adaptation

Suppose we collected a set of images using a robotic platform with the aim of training a robot vision model with it. Since the image collection has been acquired in the real world, the resulting model will be biased towards the particular conditions (*e.g.* illumination, environmental) under which the images have been acquired. Because of this, if we employ such a system and the current working conditions are different from those of the training set, the performances of the model will degrade due to the presence of a substantial *shift* between the training and test data. In this situation, to increase the generalization capabilities of the robot we can remove the acquisition bias either by collecting more training data in a large variety of conditions, which is extremely expensive, or by developing algorithms able to bridge the gap between the training and test data, aligning the original model to the novel scenario. The latter is the goal of domain adaptation.

Formally, we assume to have a source domain $\mathcal{S} = \{I_i^s, y_i^s\}_{i=1}^N$, where I_i^s is an image and $y_i^s \in \{1, \dots, C\}$ the associated semantic label. Together with the domain \mathcal{S} , at training time we assume to have collected a set of images, even unlabeled, of our target domain $\mathcal{T} = \{I_1^t, \dots, I_M^t\}$. The aim of DA algorithms is to build a deep model f_θ , with θ denoting the network parameters, able to correctly classify images of the target domain \mathcal{T} by exploiting the labeled data

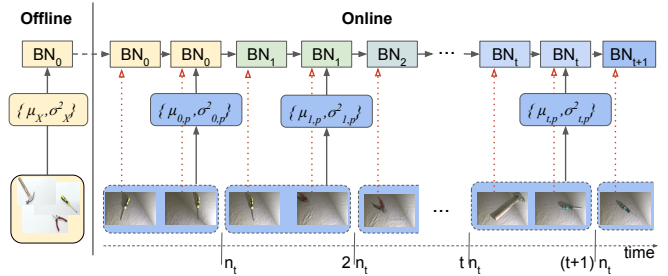


Fig. 3: The statistics of the BN layers are initialized offline, by training the network on the images of the source domain. At deployment time, the input frames are processed using the global estimate of the statistics (red lines). However the robot collects each n_t input frames to compute partial BN statistics, using these estimated values to gradually update the BN statistics for the current scenario.

provided for \mathcal{S} . In the standard scenario, the set of semantic labels is shared between \mathcal{S} and \mathcal{T} .

As discussed in section II-B, one of the most successful stream of recent works has addressed DA through BN. Next section summarizes this approach, that will give us the fundamental tools for our ONline DA (ONDA) algorithm.

B. Domain Adaptation with Batch Normalization

BN [23] is a common strategy for avoiding internal covariate shift within deep learning architectures. It works by normalizing the input features to a fixed, target distribution, *i.e.* a standard Gaussian distribution. Formally, let us denote with $x^{l,k}$ the activations of the k_{th} channel of a layer l . In order to perform the normalization, the BN layer requires to compute the mean $\mu_{\mathcal{X}}^{l,k}$ and standard deviation $\sigma_{\mathcal{X}}^{l,k}$ over the training set \mathcal{X} for the activations $x^{l,k}$. Since the formulation is layer and channel independent, in the following we will remove the superscript l, k for sake of clarity. The normalization is performed as follows:

$$\hat{x} = \gamma \frac{x - \mu_{\mathcal{X}}}{\sqrt{\sigma_{\mathcal{X}}^2 + \epsilon}} + \beta, \quad (1)$$

where γ is a scale factor and β is a bias term, while ϵ is a constant introduced for numerical stability.

Since the optimization of the network is usually performed using mini-batches, the statistics of a mini-batch are used as a local estimate of the true BN statistics $\{\mu_{\mathcal{X}}, \sigma_{\mathcal{X}}^2\}$. Given a batch \mathcal{B} with n_b samples, the approximate statistics are computed as:





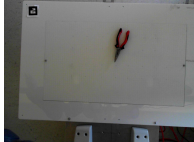

$$\mu_{\mathcal{B}} = \frac{1}{n_b} \sum_{i=1}^{n_b} x_i \quad \sigma_{\mathcal{B}}^2 = \frac{1}{n_b} \sum_{i=1}^{n_b} (x_i - \mu_{\mathcal{B}})^2$$

where x_i denotes the activations of the i_{th} sample in the mini-batch. The above statistics are exploited to progressively update the global estimate $\{\mu_{\mathcal{X}}, \sigma_{\mathcal{X}}^2\}$.

Recent works [16], [24], [17] have shown how BN layers can be exploited to perform domain adaptation in a traditional batch setting. The main idea behind these works is to create a deep architecture with two parallel branches,

¹<https://www.nada.kth.se/cas/data/handtool/>

TABLE I: Example Images from KTH Handtool Dataset

Camera Type	Illumination		
	Artificial	Cloudy	Directed
Kinect			
Webcam			

one for the source and the other for the target domain. The two branches share the same parameters but embed different domain-specific BN layers. These layers compute different statistics for the source and the target domains, resulting into domain-specific normalizations. In other words, the domain-specific BN layers allow the distributions of features of different domains to be aligned to the same reference distribution, achieving the desired domain adaptation effect.

C. ONDA: ONline Domain Adaptation

In this paper we adopt the same idea proposed in [16], [24], [17] but we consider an online setting. Instead of having a fixed target set available during training, we propose to exploit the stream of data acquired while the robot is acting in the environment and continuously update the BN statistics. In this way, we can gradually adapt the deep network to a novel scenario.

Formally, we consider a different scenario with respect to standard DA algorithms. Opposite to traditional domain adaptation in a batch setting, during training we have only access to the source domain \mathcal{S} and we do not have any data or prior information about the target domain \mathcal{T} , apart from the set of semantic labels which is assumed to be shared. When the robot is active, the current working conditions will compose the target domain and we will have access to the automatically acquired sequence of images $\mathcal{T} = \{I_1^t, \dots, I_T^t\}$. In this scenario, in order to adapt the network parameters θ to this novel domain, we must exploit the incoming test images collected by the robot on the fly.

If the network contains BN layers, following the idea of previous works [16], [24], [17], we can perform the adaptation by simply updating the BN statistics with the incoming images of the novel domain. Specifically, we start by training the network on the source domain \mathcal{S} , initializing the BN statistics at time $t = 0$ as $\{\mu_0, \sigma_0^2\} = \{\mu_S, \sigma_S^2\}$. Assuming that the set of network parameters θ are shared between the source and target domain except for the BN statistics, we can adapt the network classifier f_θ by updating the BN statistics with the estimates computed from the sequence \mathcal{T} . Defining as n_t the number of target images to use for updating online the BN statistics, we can compute

a partial estimate $\{\hat{\mu}_t, \hat{\sigma}_t^2\}$ of the BN statistics as:

$$\hat{\mu}_t = \frac{1}{n_t} \sum_{i=1}^{n_t} x_i \quad \hat{\sigma}_t^2 = \frac{1}{n_t} \sum_{i=1}^{n_t} (x_i - \hat{\mu}_t)^2$$

The global statistics at time t can be updated as follows:

$$\sigma_t^2 = (1 - \alpha)\sigma_{t-1}^2 + \alpha \frac{n_t}{n_t - 1} \hat{\sigma}_t^2$$

$$\mu_t = (1 - \alpha)\mu_{t-1} + \alpha \hat{\mu}_t$$

where α is the hyper-parameter regulating the decay of the moving average.

The above formulation achieves a similar adaptation effect of the methods [16], [24], [17] but with three main advantages. First, no samples of the target domain, neither labeled nor unlabeled, are used during training. Thus, no further data acquisition and annotation efforts are required. Second, since we do not exploit target data for training, contrary to standard DA algorithms, we have no bias towards a particular target domain. Third, since the adaptation process is online, the model can adapt itself to multiple sequential changes of the working conditions, being able to tackle unexpected environmental variations (*e.g.* sudden illumination changes).

The reader might wonder if other possible choices may be considered for initializing $\{\mu_0, \sigma_0^2\}$, such as exploiting a first calibration phase where the robot collects images of the target domain in order to produce a first estimate of the BN statistics. Here we choose to use the statistics estimated on the source domain because 1) we want a model ready to be employed, without requiring any additional preparation at test time; 2) the robot may occur in multiple domains during employment and if a shift occurs (*e.g.* illumination condition changes) our method will automatically adapt the visual model to the novel domain starting from the current estimated statistics: initializing $\{\mu_0, \sigma_0^2\} = \{\mu_S, \sigma_S^2\}$ allows to check the performance of the algorithm even for multiple sequential shifts and long-term applications. Obviously our method can benefit from a calibration phase or initializations of the statistics closer to the target working conditions: we plan to analyze these aspects in future works.

V. EXPERIMENTS

A. Networks and training protocols

We perform our experiments with the AlexNet [6] architecture pre-trained on ImageNet [26]. We train 3 additional models: a variant of AlexNet with BN, the DA architecture DIAL from [24] and our ONline DA model (ONDA). Following [24], we add BN layers or its variants after each fully-connected layer. Both the standard AlexNet, AlexNet with BN and DIAL are trained with a batch-size of 128. We implemented [24] by splitting the batch-size between images of source and target domain proportionally to the number of images for each set, as in [24], without employing the entropy-loss for target images [24], [17]. We highlight that DIAL is our upper-bound in this case, since it shares the same philosophy of ONDA but with the assumption that images of the target domain are available at training time.

As preprocessing, we rescale all the images in order to ensure a shortest side of 256 pixels, preserving the aspect ratio and subtracting the mean value per channel computed over the ImageNet database. As input to the network we use a random crop of 227×227 at training time, employing a central crop with the same dimensions during test. No additional data augmentation is performed. For all the variants of the architecture, we fine-tune the last layers for 30 epochs with an initial learning rate of 0.001 for f_{c6} , f_{c7} and of 0.01 for the classifier, with a weight decay of 0.0005 and momentum 0.9. The initial learning rates are scaled by a factor of 0.1 after 25 epochs.

In order to apply our method, we start from the weights of AlexNet with BN, training on the given source domain. Then, we perform one iteration over the target domain, without updating any parameter other than the BN statistics. As a trade-off between stability of the statistics and fastness of adaptation we set $n_t = 10$ and $\alpha = 0.1$. We will detail the impact of these choices in the following sections.

In all the experiments, we consider the task of object recognition in the *fine-grained* setting, with all the 9 classes considered as classification objective. We report the average accuracy between 5 runs, shuffling the order of the input images in each run of our model.

B. Domain Adaptation results

In this subsection, we will present the results of our algorithm. In order to analyze the particular effect that each possible change may have to the adaptation capabilities of our model, we isolate the sources of shift. To this extent, we consider two sample starting source domains: in the first case (Figure 4a), the acquisition conditions are artificial light, Kinect camera and white background; in the second case we consider cloudy illumination, webcam and brown background (Figure 4b). From these source domains we start by changing only one of the acquisition conditions (left part of the figures) and gradually increasing the number of changes to 2 and 3 conditions (middle and right parts respectively). We report the results for our model after 25%, 50% and 90% of the target data processed.

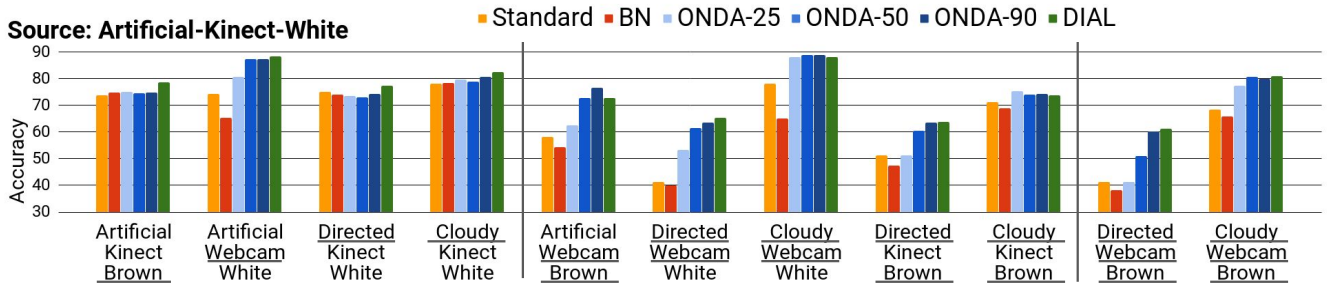
As the figures show, our model is able to fill the gap between the BN baseline (red bars) and the DA upper bound DIAL (green bars) in almost all settings. Only in few cases, where the shift between the performances of BN and DIAL is lower, this does not happen (*i.e.* Figure 4a, target artificial-Kinect-brown and directed-Kinect-White). In all the other settings the gains are remarkable: considering both figures, the average difference between the performance of BN and ONDA-90 are of 15%, 18% and 20% for the single, double and triple shift cases respectively. We stress that the gain increases with the amount of shift between the source and target domains, underlying the importance of applying DA adaptation methods in changing environments. As expected, the statistics computed in the first stages (*i.e.* ONDA-25) are not always sufficiently representative of the true estimate since they may be still biased by the statistics computed over the source domain. However the estimate becomes more precise as more images of the target domain are processed (*i.e.* ONDA-50 and ONDA-90), gradually covering the gap with the estimate computed by DIAL. The fastness of adaptation and the quality of the estimates depend on the two hyper-parameters α and n_t . In the next subsection we will analyze their impact to the final performances of the algorithm.

C. Ablation study

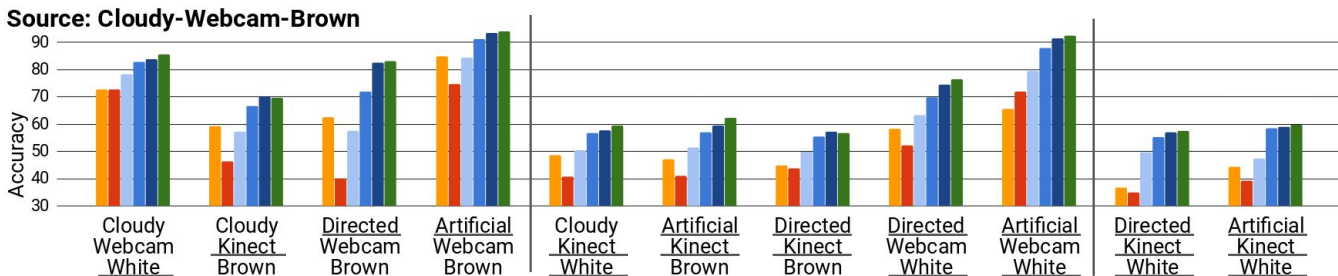
In this subsection we analyze the impact of the two hyper-parameters, the update frequency n_t and the decay α , on the number of images needed by ONDA to estimate the statistics for the target domain. We use a sample scenario of Figure 4b, where cloudy illumination, webcam camera and brown background are the source domain conditions and artificial light, Kinect camera and white background are the target domain ones. In the first experiment, we fix n_t to 10, varying the value of α . We start by a single pre-trained model of AlexNet with BN repeating the experiments for 5 runs, shuffling the order of the input data, and reporting the average accuracy for each update step.

Results are shown in Figure 5: increasing the value of α to 0.2 (green line) or 0.5 (black line) allows the model to achieve a faster adaptation to the target conditions, with the drawback of a noisier estimation of the statistics. Thus, increasing α leads to an unstable convergence of the performance. On the other hand, choosing too low values of α (*e.g.* 0.05 or 0.01, purple and gold lines respectively) allows a more stable convergence of the model, but with the drawback of slower adaptation to the novel conditions.

Regarding the hyper-parameter n_t , we follow the same protocol of the first experiment, fixing α to 0.1 and varying the number of images collected before updating the statistics, n_t , reporting how the accuracy changes with respect to the number of frames processed. As Figure 6 shows, low values of n_t (*e.g.* $n_t = 2$) allows a faster adaptation, due to the higher update frequency, but at the price of a noisier estimation of the statistics, which is harmful to the final accuracy achieved by the model. At the same time, high values of n_t (*e.g.* 20, 30) allow for a more precise estimate of



(a) Source Domain: Artificial light, Kinect camera and White background



(b) Source Domain: Cloudy light, Webcam camera and Brown background

Fig. 4: Experiments on isolated shifts. The labels of the x-axes denote the conditions of target domain, with the first line indicating the light condition, the second the camera and the third the background. We underlined the changes between the source and target domains.

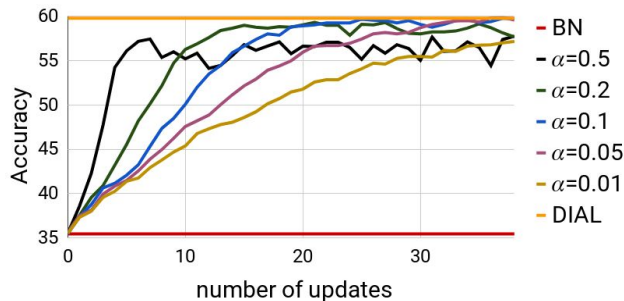


Fig. 5: Accuracy vs number of updates of ONDA for different values of α fixing $n_t = 10$ in a sample scenario. The red line denotes the BN lower bound of the starting model, while the yellow line the DIAL upper bound.

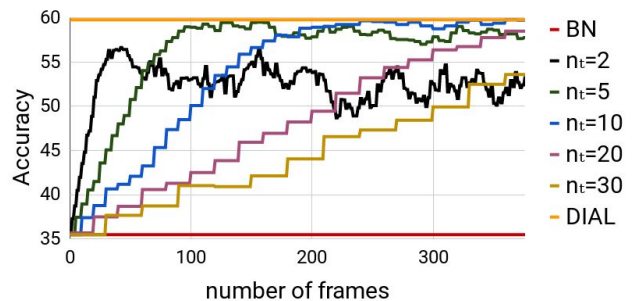


Fig. 6: Accuracy vs number of frames processed of ONDA for different values of n_t fixing $\alpha = 0.1$ in a sample scenario. The red line denotes the BN lower bound of the starting model, while the yellow line the DIAL upper bound.

the statistics, highlighted by the smoothness of the respective lines in the graph, with the drawback of a lower speed of adaptation to the novel domain, caused by the lower update frequency.

The speed of adaptation and the final quality of the BN statistics is obviously a consequence of the values chosen for both hyper-parameters. Obviously α and n_t are not independent from each other: for a lower n_t a lower α should be selected in order to preserve the final performance of the algorithm and conversely for a higher n_t , a higher α will allow a faster adaptation of the model. As a trade-off between fast adaptation and good results, we found experimentally that choosing $n_t = \{5, 10, 20\}$ and $\alpha = \{0.05, 0.1\}$ worked well for both short and long term experiments.

VI. CONCLUSIONS

In this work, we presented a novel dataset for addressing the kitting task in robotics. The dataset takes into account multiple variations of acquisition conditions such as camera, illumination and background changes which may occur during the robot employment. This dataset is intended for testing the robustness of robot vision algorithms to changing environments, providing a novel benchmark for assessing the robustness of robot vision systems.

Together with the dataset, we proposed an algorithm which is able to perform online adaptation of deep models to unseen scenario. The algorithm, based on the update of the statistics of batch-normalization layers, is able to continuously adapt the model to the current environmental conditions of the

robot, providing more robustness to unexpected working conditions. Experiments on the newly proposed dataset, confirm the ability of our algorithm to fill the gap between a standard architecture and its domain adapted counterpart without requiring any additional target data during training.

As future works, we plan to enlarge the dataset, including more source of variations and more objects. We further plan to provide a deeper analysis of our algorithm with more architectures, as well as exploring possible extensions which could exploit knowledge coming from previously met scenarios.

REFERENCES

- [1] A. G. Banerjee, A. Barnes, K. N. Kaipa, J. Liu, S. Shriyam, N. Shah, and S. K. Gupta, "An ontology to enable optimized task partitioning in human-robot collaboration for warehouse kitting operations," in *Next-Generation Robotics II; and Machine Intelligence and Bio-inspired Computation: Theory and Applications IX*, vol. 9494. International Society for Optics and Photonics, 2015, p. 94940H.
- [2] D. Holz, A. Topalidou-Kyniazopoulou, F. Rovida, M. R. Pedersen, V. Krüger, and S. Behnke, "A skill-based system for object perception and manipulation for automating kitting tasks," in *IEEE Conference on Emerging Technologies & Factory Automation (ETFA)*, 2015.
- [3] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa, "Fast object localization and pose estimation in heavy clutter for robotic bin picking," *IJRR*, vol. 31, no. 8, pp. 951–973, 2012.
- [4] A. Schyja, A. Hypki, and B. Kuhlenkter, "A modular and extensible framework for real and virtual bin-picking environments," in *ICRA*, 2012.
- [5] K. N. Kaipa, S. S. Thevendria-Karthic, S. Shriyam, A. M. Kabir, J. D. Langsfeld, and S. K. Gupta, "Resolving automated perception system failures in bin-picking tasks using assistance from remote human operators," in *CASE*, 2015.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.
- [7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
- [8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE T-PAMI*, vol. 39, no. 6, pp. 1137–1149, 2017.
- [9] H. Karaoguz, N. Bore, J. Folkesson, and P. Jensfelt, "Human-centric partitioning of the environment," in *IEEE International Symposium on Robot and Human Interactive Communication, RO-MAN*, 2017.
- [10] J. Young, L. Kunze, V. Basile, E. Cabrio, N. Hawes, and B. Caputo, "Semantic web-mining and deep vision for lifelong object discovery," in *ICRA*, 2017.
- [11] G. Pasquale, C. Ciliberto, L. Rosasco, and L. Natale, "Object identification from few examples by improving the invariance of a deep convolutional neural network," in *IROS*, 2016.
- [12] J. Hoffman, S. Gupta, J. Leong, S. Guadarrama, and T. Darrell, "Cross-modal adaptation for rgb-d detection," in *ICRA*, 2016.
- [13] H. F. Zaki, F. Shafait, and A. Mian, "Convolutional hypercube pyramid for accurate rgb-d object category and instance recognition," in *ICRA*, 2016.
- [14] F. M. Carlucci, P. Russo, and B. Caputo, "A deep representation for depth images from synthetic data," in *ICRA*, 2017.
- [15] G. Csurka, *A Comprehensive Survey on Domain Adaptation for Visual Applications*. Springer, 2017, pp. 1–35.
- [16] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," in *ICLR WS*, 2017.
- [17] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, "Autodial: Automatic domain alignment layers," in *ICCV*, 2017.
- [18] G. Angeletti, B. Caputo, and T. Tommasi, "Adaptive deep learning through visual domain localization," in *ICRA*, 2017.
- [19] M. Long and J. Wang, "Learning transferable features with deep adaptation networks," in *ICML*, 2015.
- [20] M. Mancini, S. R. Bulo, B. Caputo, and E. Ricci, "Robust place categorization with deep domain generalization," *IEEE RA-L*, 2018.
- [21] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *ICCV*, 2015.
- [22] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," *ICML*, 2015.
- [23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015.
- [24] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò, "Just dial: Domain alignment layers for unsupervised domain adaptation," in *ICIAP*, 2017.
- [25] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *ICCV*, 2017.
- [26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009.