Deep Depth Domain Adaptation: A Case Study

Novi Patricia* Fabio M. Carlucci* Barbara Caputo*†
*Sapienza Rome University †IDIAP Research Institute Switzerland
novi@dis.uniroma1.it, fabiom.carlucci@dis.uniroma1.it, caputo@dis.uniroma1.it

Abstract

In the era of deep learning, many domain adaptation studies have been done on RGB images but not on depth. One of the reasons is that there are few databases available for researchers to explore domain shift on depth images. The contribution of this paper is to provide a benchmark to the community to study and evaluate deep domain adaptation methods on depth images, and compare the results with those obtained on the corresponding RGB data. We use two variants dataset that follow the settings from the first introduced RGB-D object dataset with 51 categories taken from multiple views. We also explore different colorization methods for depth images such as Colorjet and DE^{2}CO[3]. The experiments are conducted on several deep domain adaptation approaches on RGB and depth images. Our results show that current deep DA methods can work well for RGB images but how to tackle the domain shift problem on depth is still an open question.

1. Introduction

The ability to recognize objects across different visual domains is essential for the use of computer vision technology in a wide range of applications, from autonomous vehicles to intelligent security systems to service robotics. Traditionally, the visual information is considered in the form of RGB images, and indeed almost all of previous work has focused on this type of visual information[18, 10, 11, 6, 4]. Still, depth is an equally important source of perceptual information, especially in robotics and autonomous systems where artificial agents are expected to move and act in the environment. This topic is at the moment less investigated, due to the lack of a suitable database supporting a systematic study of the domain adaptation problem on depth.

This paper fills this gap, presenting a new testbed for depth domain adaptation. We started from the Washington Database[9], one of the most popular existing RGB-D databases supporting object categorization, and we acquired two new versions of it in two different domain, the Institute for Artificial Intelligence at Bremen University and the VANDAL Laboratory at the Sapienza Rome University, acquired in the city of Latina. The two new acquisitions preserved the exact same object categories contained in the Washington database, but the statistic of the objects per category, and the acquisition protocol changed for the two domains, hence leading to a tangible shift across domains. Upon acceptance of the paper, the new databases, together with all the scripts necessary to replicate our experimental setup, will be made available to the community.

To assess the data, we performed a benchmark evaluation, selecting five deep domain adaptation algorithms that can be considered as representative of the current research trends in the domain adaptation community, as well as legitimate 'off-the-shelf' state of the art choices [20]. This opens the question of how to use algorithms designed and trained so to work on RGB images (hence concretely building over deep networks pre-trained over ImageNet [5]) on depth images. To address this issue, we leveraged over recent trends in the robot vision community and mapped the depth images into 3 channel images, hence mimicking a sort of colorization [3].

Our results show that (a) algorithms obtaining the best performance on RGB data do not maintain their edge on the depth modality, and (b) a straightforward combination of results obtained over RGB and depth images does not achieve strong results. These two findings clearly call for a research effort specifically targeted at deep depth domain adaptation. The rest of the paper is organized as follows: section 2 introduces the databases, the colorization methods adopted in the paper, and the deep domain adaptation algorithms we chose for our benchmark. Section 3 reports our experimental results.

2. Methods

2.1. Dataset

We use the Washington dataset [9], the Bremen variant, and the Latina dataset. In figure 1 we show a few images from each.

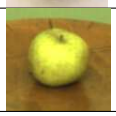
Dataset	Apple	Coffee Mug	Flashlight	Greens	Soda Can	Toothpaste
Washington						
Bremen						
Latina						

Figure 1. Images of six categories from each dataset.

Washington It contains 300 objects organized in 51 categories[1]. For each object, there are three turntable sequences captured from different camera elevation angles (30°, 45°, 60°). The sequences were captured with an ASUS Xtion Pro Live camera in both RGB and depth channels. The dataset provides an object segmentation based on depth and color. Since two consecutive views are extremely similar, only 1 frame out of 5 is used for evaluation purposes [2].

Bremen The Bremen variant, captured in collaboration with and at the Bremen Institute for Artificial Intelligence, has the same 51 classes used in Washington, with one object instance per class. Data was collected using an Asus Xtion Pro kept at 1m distance from the objects. Items where positioned on a turntable and a full sequence, containing around 171 images, was captured at angles of 30°, 45° and 60°. Both RGB and Depth was recorded.

Latina The dataset has 51 categories with a total of 301 objects. The dataset is collected using a wearable device developed by the Cognitive Robotics Laboratory ALCOR at DIAG, Sapienza Rome University. The device used here is a Gaze Machine, a head mounted device composed of four cameras [12, 14]. The dataset is a collection of 35 images for each object with varies of 11 degrees angle per two consecutive images.

Description	Washington	Bremen	Latina
Number Of Class	51	51	51
Total Images	207920	26248	9874
Instances per Category	3-14	1	3-14

Table 1. Dataset summary

Latina is the smallest dataset among all databases and Bremen has only one instance for each category (Table 1).

2.2. Colorization Methods

RGB-D object recognition is a crucial element for real-world robotics application. RGB images provide information about appearance and texture, on the other hand depth data contains additional information about object shape and it is invariant to lighting or color variations. The approach currently consists of designing ad-hoc colorization method, where ColorJet has become the off-the-shelf state of the art colorization approach for depth-based object recognition [5]. We explore two different approaches used for depth colorization: shallow and deep depth colorization. Figure 2 shows depth images with three different colorization methods.

Shallow Depth Colorization: ColorJet The ColorJet technique [5] works by assigning different colors to different depth values. The original depth map is normalized between 0-255 values. The first step in colorization is to map the lowest value to the blue channel and the highest value to the red channel. The middle value is mapped to green and the intermediate values are arranged accordingly. The resulting image exploits the full RGB spectrum, with the intent of leveraging the filters learned by deep networks trained on very large scale RGB datasets like ImageNet. The method shows very strong results when tested on the Washington database but was not designed to create realistic looking RGB images for the objects depicted in the original depth data. This simple method outperformed more sophisticated approaches, for example a method proposed by Schwarz *et al*. [16]. Hence, we prefer to use ColorJet in our experiments.

Deep Depth Colorization: (DE)^{2}CO The method [3] tries to feed the depth maps, normalized into grayscale images, to a colorization network linked to a standard CNN architecture, pre-trained on ImageNet. The architecture uses 1×228×228 input depth map (*i.e*. grayscale image), reduced to 64×57×57 size by convolutional and pooling layer, passes through a sequence of 8 residual blocks, composed by 2 small convolutions with batch normalization layer and leackyReLu as non linearities. The last residual block output is convolved by a three features convolution to form the 3 channels image output. Its resolution is brought back to 228×228 by a deconvolution (upsampling) layer.

[1] https://rgbd-dataset.cs.washington.edu/

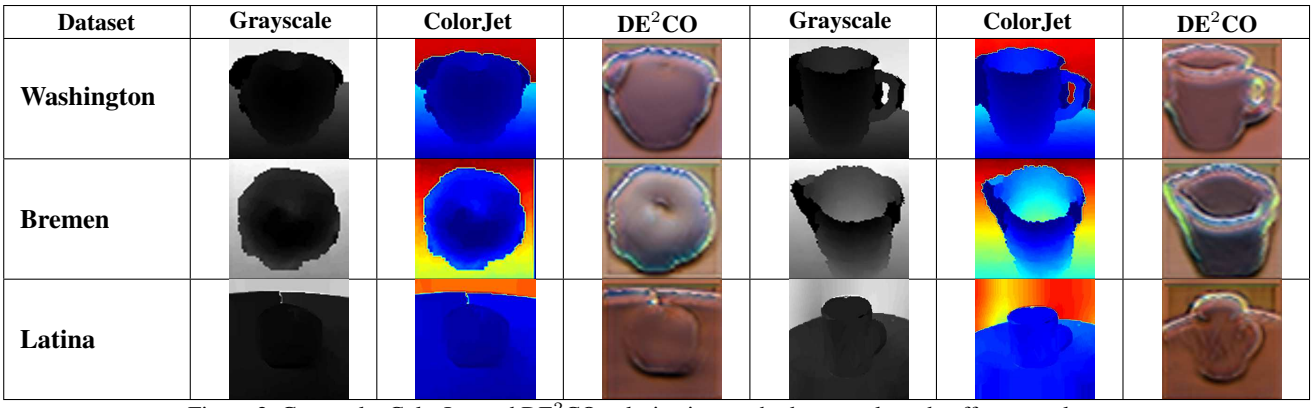


Figure 2. Grayscale, ColorJet, and DE^{2}CO colorization methods on apple and coffee mug classes.

2.3. Algorithms

We use recently published deep domain adaptation methods to assess the performance of the algorithms on depth and RGB images.

DDC The method [18] was designed to minimize the distance between source and target distributions, it trained a classifier on the source labeled data and applied them directly to the target domain with minimal loss in accuracy. To minimize the distance, the authors use the standard distribution distance metric, Maximum Mean Discrepancy (MMD). This distance is computed with respect to the source and target data points. Minimizing the distance between domains (or maximizing the domain confusion), the method has a strong classification representation and uses MMD to decide which layer to use activations from to minimize the domain distribution distance. This representation can be used to train another classifier for the classes that we are interested in recognizing.

DAN Long *et al*. [10] proposed a method to bound the target error by the source error plus a discrepancy metric between the source and the target. The method uses the multiple kernel variant of MMD (MK-MMD) [7], which is formalized to jointly maximize the two-sample test power and minimize the failure of rejecting a false null hypothesis. The method fine-tunes CNN on the source labeled examples and requires the distributions of the source and target to become similar under the hidden representations of fully connected layers fc6-fc8. This can be established by adding an MK-MMD based multi-layer adaptation regularizer to CNN risk. With DAN optimization framework, we learn transferable features from a source domain to a related target domain. The learned representation can both be salient benefiting from CNN and unbiased thanks to MK-MMD.

RTN The focus of RTN [11] is to reduce the mismatches in both features and classifiers by fixing the joint adaptation of features and classifiers. Classifier adaptation is more difficult than feature adaptation because it is directly related to the labels but the target domain is fully unlabeled. Deep features must eventually through transition from general to specific along the network and the transferability of features and classifiers will decrease when the cross-domain discrepancy increases [19]. In other words, the shifts in the data distributions linger even after multilayer feature abstractions, therefore the feature distributions need to be adapted on multiple layers across domains. As feature adaptation cannot remove the mismatch in classification models, RTN performs classifier adaptation using a residual function. The residual networks are used to bridge the inputs and outputs of the residual layers by the identity mapping which is similar to the perturbation function across the source and target classifiers.

GRL The method proposed by Ganin and Lempitsky [6] focuses on optimizing the features as well as two discriminative classifiers operating on: the label predictor and the domain classifier. The label predictor predicts class labels and is used both during training and at test time. The domain classifier discriminates between the source and the target domains during training. While the parameters of the classifiers are optimized in order to minimize their error on the training data, the parameters of the underlying deep feature mapping are optimized in order to minimize the loss of the label classifier and to maximize the loss of the domain classifier. During the forward propagation, GRL behaves as an identity transform. However during the backpropagation, GRL takes the gradient from the subsequent level, multiplies with a constant and passes it to the preceding layer. Implementing the layer using existing object-oriented packages for deep learning is simple, as defining procedures for forward-propagation, back-propagation, and parameter update is trivial.

DIAL The approach tries to couple the training process and the domain adaptation step within deep neural networks, but ignoring the assumption that the domain alignment satisfies by applying the same predictor to the source and target domains [4]. Motivated by [1], the authors assume that the source and target predictors are in general different functions. The common hypothesis couples explicitly

two predictors, but it is not directly involved in the alignment of the source and target domains. The source and target predictors are implemented as two deep neural networks being almost identical, however the two networks contain also a number of special layers, called *Domain-Alignment* layers, which implement a domain-specific operation. In general, when considering channel-wise linear transformations and a standard normal distribution as reference, DA-layer can be built from Batch Normalization, concatenate and split the layers. The outputs of the BNs are then concatenated again and fed to the following layer.

3. Experiments

3.1. Setup

We perform evaluations on five different deep domain adaptation networks: DDC[2], DAN, RTN[3], GRL[4], and DIAL[5], with three colorization methods. We use *SourceOnly* as the lower-bound for testing the target-domain data (*i.e*. no domain classifier branch included into the network). All methods are assessed on all six transfer tasks B→W, W→B, W→L, L→W, B→L, and L→B.

Depth Colorization We colorize all depth images using three different methods: Grayscale, ColorJet, and DE^{2}CO. We include grayscale colorization for our experiments, which has one color channel, as a baseline for DE^{2}CO method. The colorized depth images become an input to the deep domain adaptation networks.

Training The model is trained on 128-sized batches for both source and target samples. We follow the default parameters proposed by each algorithm, using total of 60 epochs for all scenarios. We use grid-search for having the best learning rate $\{1^{-5}, \cdots, 1^{-2}\}$ on DE^{2}CO. We use ImageNet pre-trained model for ColorJet and Grayscale colorization methods and JHUIT-50 pre-trained model for DE^{2}CO [3]. For Washington dataset, we left out one instance from each object for testing and training [2] but use all samples for Bremen and Latina. We follow the settings for the unsupervised deep domain adaptation in our experiments [6].

3.2. Results: Depth Only

First, we show the classification accuracies for depth channels on different colorization methods. We want to understand how good the depth images only on the deep domain adaptation networks, then we present the results for the RGB images (Table 5).

[2]https://gist.github.com/jhoffman/9a28bcaf354f21ad3169f0679d73f647

[3]https://github.com/thuml/transfer-caffe

[4]https://github.com/ddtm/caffe/tree/grl

[5]https://github.com/ducksoup/autodial

Method	B→W	W→B	W→L	L→W	B→L	L→B
SourceOnly	24.74	35.2	12.45	12.13	4.87	7.25
DDC	29.86	38.4	9.97	15.23	5.19	**9.19**
DAN	26.73	**42.27**	11.9	13.21	5.94	8.84
RTN	29.56	41.94	9.23	11.74	5.65	8.83
GRL	**32.58**	38.82	**14.35**	14.92	**8.97**	7.74
DIAL	29.09	39.31	12.2	**16.61**	7.79	8.57

Table 2. Accuracy on Grayscale using standard unsupervised domain adaptation protocol.

Method	B→W	W→B	W→L	L→W	B→L	L→B
SourceOnly	28.02	31.26	10.54	10.45	4.96	7.18
DDC	31.88	**42.92**	13.28	14.1	5.17	9.54
DAN	31.14	42.81	14.82	14.06	8.07	10.6
RTN	32.67	39.61	15.16	12.16	7.4	10.33
GRL	**33.24**	42.77	12.22	**17.35**	7.63	**13.77**
DIAL	29.45	36.82	**15.9**	15.47	**8.65**	11.58

Table 3. Accuracy on ColorJet using standard unsupervised domain adaptation protocol.

Method	B→W	W→B	W→L	L→W	B→L	L→B
SourceOnly	29.5	34.4	7.37	8.52	4.45	4.77
DDC	32.21	40.4	8.25	12.07	5.75	10.35
DAN	32.03	42.02	12.82	15.46	5.92	8.06
RTN	33.46	39.03	10.52	14.22	6.88	7.52
GRL	34.08	44.45	14.54	**19.28**	8.7	11.06
DIAL	**35.23**	**45.04**	**15.15**	13.18	**9.74**	**12.03**

Table 4. Accuracy on DE^{2}CO using standard unsupervised domain adaptation protocol.

We see that DE^{2}CO (Table 4) shows competitive accuracies with ColorJet (Table 3), with B→W, W→B, L→W, and B→L work best with DE^{2}CO. DIAL outperforms all comparison methods for DE^{2}CO, while GRL performs slightly better for ColorJet and Grayscale. Grayscale gives the lowest accuracies compared to ColorJet and DE^{2}CO, as we expect to happen to the images with one color channel (Table 2). B→L and L→B settings have the lowest accuracy on depth images for all colorization methods, but only B→L task on RGB images shows the lowest result comparing to all tasks (Table 5). For RGB images, DAN and DIAL methods perform better than other deep networks. Overall, we see that domain adaptation strategies are an important factor to reduce the distribution shift between domains, where the experiments show that deep domain adaptations methods (*i.e*. DDC, DAN, RTN, GRL, and DIAL) work better than *SourceOnly* even on depth only images.

Method	B→W	W→B	W→L	L→W	B→L	L→B
SourceOnly	35.43	50.92	31.25	24.29	19.5	17.39
DDC	31.58	55.12	30.85	22.05	23.56	25.46
DAN	**38.73**	59.44	**43.03**	**40.59**	33.28	44.02
RTN	35.13	56.19	36.88	37.64	30.3	38.83
GRL	35.39	56.91	34.35	30.36	28.62	38.58
DIAL	34.81	**76.17**	41.78	35.4	**34.56**	**55.72**

Table 5. Accuracy on RGB using standard unsupervised domain adaptation protocol.

3.3. Results: Depth+RGB

We extend the experiments by combining the depth and RGB channels. It is normal to have a better classification result on RGB channels, as we have clearer RGB images (Figure 1) than depth images (Figure 2). The idea of the following experiments is to understand how much improvement we can get when we use both channels. For example, adding information about object shape to RGB images can help home robot to differentiate better between soda can and coffee mug.

We combine both depth and RGB channels by finding the max value over 51 sized vectors to determine the object class. This approach can be considered as the high-level integration scheme [15, 13]. We use this simple approach as a step stone to understand how to combine different image channels on deep domain adaptation architectures.

Method	B→W	W→B	W→L	L→W	B→L	L→B
SourceOnly	1.75	2.04	1.64	1.75	1.64	2.04
DDC	32.03	56.06	19.41	27.46	12.34	25.3
DAN	36.81	59.16	35.8	**35.79**	20.9	38.07
RTN	30.88	56.17	31.05	34.51	20.02	36.64
GRL	34.34	53.73	**41.86**	32.06	**24.77**	40.9
DIAL	**41.93**	**75.96**	24.52	27.14	24.69	**53.39**

Table 6. Accuracy on Grayscale+RGB using standard unsupervised domain adaptation protocol.

Method	B→W	W→B	W→L	L→W	B→L	L→B
SourceOnly	1.75	2.04	1.64	1.75	1.64	2.04
DDC	38.92	60.23	25.98	24.27	13.57	28.02
DAN	39.8	62.23	35.91	**37.21**	22.3	36.08
RTN	37.68	57.62	30.68	35.46	22.36	38.64
GRL	39.17	54.96	**40.92**	29.29	**24.98**	40.88
DIAL	**47.38**	**72.25**	24.5	26.36	24.08	**51.17**

Table 7. Accuracy on ColorJet+RGB using standard unsupervised domain adaptation protocol.

Method	B→W	W→B	W→L	L→W	B→L	L→B
SourceOnly	1.75	2.04	1.64	1.75	1.64	2.04
DDC	45.97	60.9	20.22	28.24	11.84	25.7
DAN	42.33	60.75	33.7	**35.75**	24.43	36.41
RTN	37.62	56.61	30.51	31.78	21.69	33.39
GRL	41.29	52.87	**37.95**	32.03	24.29	35.42
DIAL	**48.73**	**77.83**	19.25	20.87	**24.74**	**55.89**

Table 8. Accuracy on DE^{2}CO+RGB using standard unsupervised domain adaptation protocol.

From Table 8, we see that B→W achieves more than 10% improvement on DE^{2}CO+RGB comparing the classification on RGB only (Table 5). However, W→B setting does not show significant improvement, even ColorJet+RGB (Table 7) and Grayscale+RGB (Table 6) have lower accuracy than RGB. The combination of depth and RGB channels on W→L, L→W, and B→L settings get accuracies below the RGB classification and it occurs on all

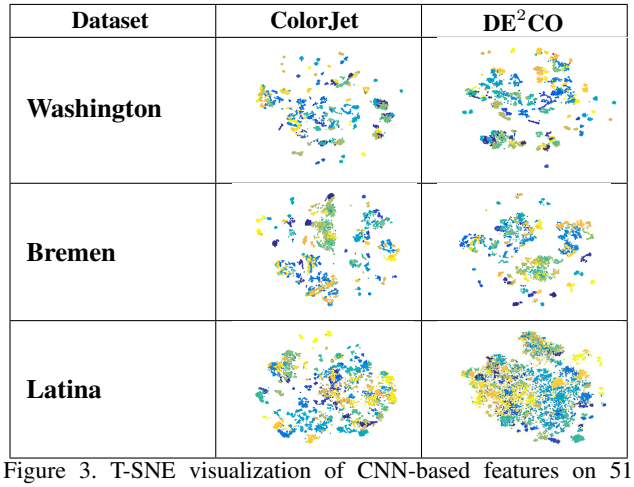


Figure 3. T-SNE visualization of CNN-based features on 51 classes.

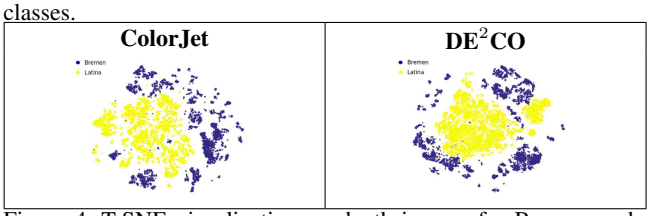


Figure 4. T-SNE visualization on depth images for Bremen and Latina dataset

colorization methods. The Grayscale+RGB performs better on W→B, W→L and L→B than ColorJet+RGB. The L→B task on Grayscale+RGB and DE^{2}CO+RGB show relatively same behavior with RGB channel. Overall, we see at Table 9 that DIAL performs the best on all colorization methods. In average, combining depth and RGB channels here does not give the best classification result comparing to RGB channel. This might happen, where we use the simplest way to combine the depth and RGB channels, by averaging the-fc7 activation of depth and RGB domain adaptation networks. There are different ways to combine the depth and RGB channels where we do not explore further in our experiments, such as plug-in depth and RGB images directly to the CNN networks [5, 8] while at the same time considering the domain shift between domains [17].

We visualize the depth images based on the CNN-features, where the Latina dataset seems to be denser than others (Figure 3). When visualizing Bremen and Latina together, we see that Latina clustered in the center where Bremen surrounding it (Figure 4). This might become one of the reason that the domain adaptation does not work well for this case. Meanwhile, Washington and Bremen are well clustered and overlapped with each other (Figure 5). We see this as a good challenge for further research on deep domain adaptation problem, specifically to have more complex deep domain adaptation networks.

4. Conclusions

This paper presents a new benchmark for studying domain adaptation on depth images. We collected two

Method	RGB	Grayscale	ColorJet	DE^{2}CO	RGB+Graycale	RGB+ColorJet	RGB+DE^{2}CO
SourceOnly	29.8	16.11	15.4	14.84	1.81	1.81	1.81
DDC	31.43	17.97	19.48	18.17	28.77	31.83	32.15
DAN	43.18	18.15	20.25	19.39	37.76	38.92	38.9
RTN	39.16	17.83	19.55	18.61	34.88	37.07	35.27
GRL	37.37	**19.56**	**21.16**	**22.02**	37.94	38.37	37.31
DIAL	**46.41**	18.93	19.65	21.73	**41.27**	**40.96**	**41.22**

Table 9. Summary of mean classification accuracy on all domain adaptation scenarios.

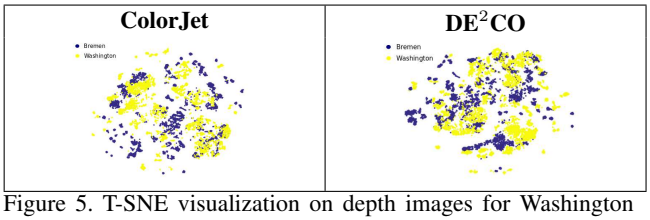


Figure 5. T-SNE visualization on depth images for Washington and Bremen dataset

databases that replicate the structure of the Washington database, and we assess this new domain adaptation testbed using three different colorization methods and five different deep domain adaptation approaches. Our results show that domain alignment in the depth space is a challenging and open problem, calling for specific research efforts by the community.

References

[1] S. Ben-David, T. Lu, T. Luu, and D. Pl. Impossibility theorems for domain adaptation. In *AISTATS*, volume 9 of *JMLR Proceedings*. JMLR.org, 2010.

[2] L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for rgb-d based object recognition. In *In International Symposium on Experimental Robotics (ISER*, 2012.

[3] F. Carlucci, P. Russo, S. Baharlou, and B. Caputo. De2co: Deep depth colorization. *arXiv preprint arXiv:1703.10881*, 2017.

[4] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulo. Autodial: Automatic domain alignment layers. In *International Conference on Computer Vision, ICCV*, 2017.

[5] A. Eitel, J. T. Springenberg, L. Spinello, M. A. Riedmiller, and W. Burgard. Multimodal deep learning for robust RGB-D object recognition. *CoRR*, abs/1507.06821, 2015.

[6] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. JMLR Workshop and Conference Proceedings, 2015.

[7] A. Gretton, B. Sriperumbudur, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, and K. Fukumizu. Optimal kernel choice for large-scale two-sample tests. In *Advances in Neural Information Processing Systems 25*, pages 1214–1222, 2012.

[8] J. Hoffman, S. Gupta, J. Leong, S. Guadarrama, and T. Darrell. Cross-modal adaptation for rgb-d detection. In *International Conference in Robotics and Automation (ICRA)*, 2016.

[9] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, pages 1817–1824. IEEE, 2011.

[10] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 97–105, 2015.

[11] M. Long, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. *CoRR*, abs/1602.04433, 2016.

[12] V. Ntouskos, F. Pirri, M. Pizzoli, A. Sinha, and B. Cafaro. Saliency prediction in the coherence theory of attention. *Biologically Inspired Cognitive Architectures*, 5(0):10 – 28, 2013.

[13] N. Patricia and B. Caputo. Learning to learn, from transfer learning to domain adaptation: A unifying perspective. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1442–1449, 2014.

[14] F. Pirri, M. Pizzoli, and A. Rudi. A general method for the point of regard estimation in 3d space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 921–928, 2011.

[15] A. Pronobis, O. M. Mozos, and B. Caputo. SVM-based discriminative accumulation scheme for place recognition. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA'08)*, 2008.

[16] M. Schwarz, H. Schulz, and S. Behnke. RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. In *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*, pages 1329–1335, 2015.

[17] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014.

[19] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 3320–3328, 2014.

[20] H. F. M. Zaki, F. Shafait, and A. S. Mian. Convolutional hypercube pyramid for accurate RGB-D object category and instance recognition. In *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, pages 1685–1692, 2016.