A Simple Domain Shifting Network for Generating Low Quality Images

Guruprasad Hegde*, Avinash Nittur Ramesh*, Kanchana Vaishnavi Gandikota*,
Roman Obermaisser, Michael Moeller
Department for Computer Science and Electrical Engineering
University of Siegen
{guruprasad.hegde, avinash.ramesh, kanchana.gandikota}@student.uni-siegen.de,
{roman.obermaisser, michael.moeller}@uni-siegen.de

***Abstract*—Deep Learning systems have proven to be extremely successful for image recognition tasks for which significant amounts of training data is available, e.g., on the famous ImageNet dataset. We demonstrate that for robotics applications with cheap camera equipment, the low image quality, however, influences the classification accuracy, and freely available data bases cannot be exploited in a straight forward way to train classifiers to be used on a robot. As a solution we propose to train a network on degrading the quality images in order to mimic specific low quality imaging systems. Numerical experiments demonstrate that classification networks trained by using images produced by our quality degrading network along with the high quality images outperform classification networks trained only on high quality data when used on a real robot system, while being significantly easier to use than competing zero-shot domain adaptation techniques.**

I. Introduction

On a closed set of images with predefined classes and controlled conditions, recent machine learning approaches match or even surpass human image classification abilities. Challenging situations, however, arise when transferring such computer vision systems into real world practical applications, in which the distribution and characteristics of the images differs from the distribution of the online training examples significantly.

As an example, we consider the problem of image classification in a video stream recorded with an Anki Cozmo© robot camera. As the robot is only about $4 \times 3 \times 2$ inches, and currently costs about € 100 only, the quality of recorded images is rather low compared to typical image classification data sets. As exemplified in Fig. 1 and detailed in Section V, the specific distortion in terms of the dynamic range, color reproduction, and noise causes the accuracy of image classification networks trained on usual online data sets to drop significantly. The latter implies that standard benchmark datasets cannot be harvested directly in order to train networks for such devices on solving various vision based tasks.

A large variety of different works have considered domain adaption methods (based on the availability of different types of examples from the source and target domain) for such situations. Their ideas include fine-tuning on labeled data of the new domain using adversarial training schemes to encourage a confusion between domains, or combining the classification with a reconstruction of data from the source domain. We refer to [1] and Section II for details.

Standard: 1.0 dog
Adapted: 1.0 dog

Standard: 0.007 dog
Adapted: 1.0 dog

Standard: 0.038 dog
Adapted: 1.0 dog

Fig. 1. Images from online databases as illustrated on the left are often of decent quality and standard networks (here: Standard [2]) trained on such good quality images achieve high classification accuracy. Low cost cameras, however, introduce distortions (middle image) in terms of the dynamic range and color reproduction, which can easily mislead standard networks. With the proposed technique of training classifiers on images that have been send through a quality degrading network to mimic low-cost cameras (right image), one obtains an adapted network that is significantly more robust with respect to low-quality images as illustrated in the classification scores below the respective images.

In this work, our objective is to classify low quality images captured by robot camera by leveraging high quality labeled data without any low quality training data specific to the classification task. However, we assume that unlabeled high quality-low quality image pairs are available which are unrelated to the final task of interest (classification). For this challenge, we propose a very simple, yet generic and powerful solution: We propose to train a simple convolutional regression network to degrade high quality images in such a way that their appearance mimics that of the same image recorded with a specific low quality camera. Subsequently, we use this domain shifting neural network to transfer any online classification dataset to a corresponding low-quality version of it and demonstrate in several numerical experiments that classifiers trained/fine-tuned on the low-quality versions yield better accuracy on images actually recorded with a low quality camera, see illustration in Fig. 1. To generate a training set for the domain shifting neural network, we record high quality images displayed on a screen with the given low-cost camera (see Fig. 4) and train the network to map the high quality images to their corresponding low quality counterparts. Since

*equal contribution

the images used to train the quality degrading network differ significantly from the images the classifier is trained on, the proposed approach can serve as a simple and generic scheme for adapting networks for various different tasks to the specific characteristics of a given low-cost camera.

II. Related Work

Within the last years deep classification network architectures have been adapted to work well on mobile systems with limited computation power, with MobileNet [2] and SqueezeNet [3] being among the most popular variants. Such networks achieve accuracies close to their significantly more computationally expensive relatives on popular computer vision benchmarks, but the architectural considerations do not specifically account for low quality input data.

Despite the continuously growing amount of labeled training data from various sources, care has to be taken of adapting any image classification network trained on online data bases to the specific setting it is meant to operate in. While it is common practice to encourage certain invariances, e.g. with respect to small noise, rotations and scale using data augmentations, it has been observed that the image quality can have a significant impact on the classification result (see e.g. [4], [5]).

The problem of adapting a trained network to a new distribution of input images is a well-studied problem under the name of domain adaptation, see e.g. [1] for a comprehensive overview. Divergence-based domain adaptation approaches [6]–[8] obtain domain invariant data representations by minimizing some divergence measure between source and target distributions. Another approach is to employ adversarial training [9], [10] to encourage a confusion between source and target domains. Alternately, a common representation for source and target domains can be constructed by combining classification with an auxiliary reconstruction task [11].

In terms of the available training data, one can distinguish 4 categories of domain adaptation techniques. In the *supervised* case, one has a well-trained network on a set of source domain images with given labels, along with a reduced number of labeled images from a target domain, and when sufficiently many labeled images in target domain are available, simple fine-tuning/transfer learning techniques [12], [13] can be applied. In *semi-supervised domain adaptation*, weaker information is available in the target domain only, e.g. the works [14]–[16] utilize additional unlabeled data for transferring knowledge to target domain when few labeled examples are available in the target domain. *Unsupervised domain adaptation* techniques [17]–[19] address the scenario where a network pretrained on labeled source domain data is adapted to a target domain of unlabeled images that are related to the source domain, e.g. containing the same classes/objects. Finally, having no labeled images in the target domain for the task of interest refers to *zero-shot domain adaptation*, see e.g. [20]–[23].

Our work can be seen as a specific form of zero-shot domain adaptation, as we do not consider to have labeled target data and usually not even target images that show the same classes as the classifier we'd like to train. By recording images from a screen we do, however, ensure we have a one-to-one correspondence of images from the source and target domain. These corresponding images are, however, entirely unlabeled.

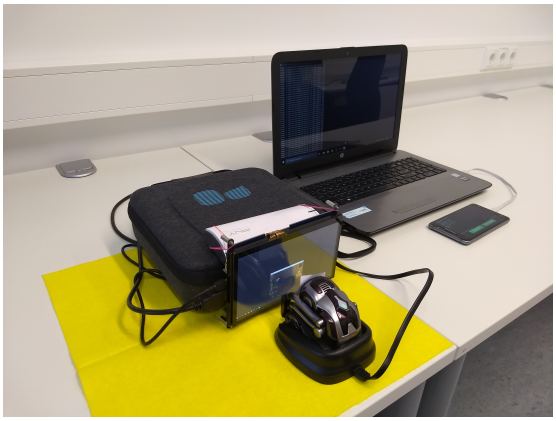

Fig. 2. Illustration of our setup for recording pairs of corresponding high and low resolution image: High resolution images are displayed on the small screen, which is captured by the the built-in camera of the robot.

In contrast to our assumptions, [22] and [21] do not have target domain data unrelated to the task of interest at all. Ishii *et al.* [22] assume good prior knowledge about attributes causing distribution shift between source and target domains, which is used in adapting to the target domain. Kumagai and Iwata [21] present the concept of latent domain vectors to represent multiple source domains, which are then used to find models for unseen target domains via bayesian inference. Similar to our assumptions, [20], [23] have target domain data unrelated to the task of interest. However, Wang and Jiang [23] do not assume correspondences between the source and target domain samples unrelated to the task of interest. They instead employ two generative adversarial networks (GANs) to learn the joint distribution of source and target domain data across two tasks. The work [20] is closest to the scenario we considered in this work. Akin to us, their approach assumes that paired data in source and target domains for an irrelevant task are available. Their approach involves two steps: first matching features of the target domain irrelevant images with the features of the source domain images from a pretrained source domain network. Second, training the source network on the relevant task, while maintaining feature similarity with the target network on the irrelevant task. The important difference, however, is that [20] assumes to have labeled paired data for classification of images (dissimilar to those considered in final classification).

Since we assume that only unlabeled source-target image pairs are available, we train a simple regression network to map from high quality source images to low quality target images. However, our generic approach can also be directly applied in unsupervised domain adaptation and any-shot domain adaptation, by further augmenting the available samples in target domain.

The task of mapping an image from one representation to

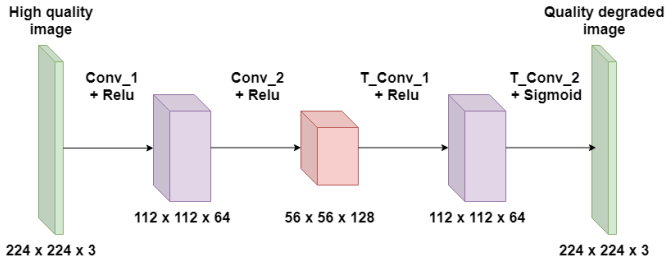

Fig. 3. Overview of the proposed domain shifting network

another is generally referred to as an image-to-image translation problem, which can include more complicated mappings such as mapping edges to natural images, colorizing a gray scale image, maps to photos etc. For such tasks, a simple convolutional regression network is not sufficient. Instead, it requires bigger and more powerful architectures employing GANs such as [24]. Similarly, GAN based approach [25] solves image to image translation when paired data is not available. While [24] or [25] have demonstrated impressive results in very complex image-to-image translation tasks, such generative adversarial approach are highly non-trivial to train. Murez *et al.* [26] employ an image-to-image translation network for domain adaptation with unpaired and unlabeled target domain data using a cycle GAN [25] with additional networks and losses.

As our recording setup does provide us with paired data, we can take the much simpler approach to train a regression network and avoid the (often cumbersome) training of a GAN.

III. Proposed Approach

A. Recording Training Data

Instead of mapping to domain-invariant features as done in many domain adaptation approaches, we create a simple image-to-image network to simulate a low quality image from high quality input image. Our idea is to simplify the domain adaptation problem by creating a dataset of *corresponding* high and low quality images and training a *simple domain shifting network* to map high to low quality images. For this purpose, we exploit the experimental setup shown in Fig. 2: A set of high resolution images is shown on a small screen which are recorded by the robot standing right in front of it. While this of course reduces the set of recorded training images to those in which the low quality image was taken of a screen, i.e., of a luminescent instead of a reflecting object, we will demonstrate that such a setting is sufficient to obtain improved classification accuracy even for real images taken with the low-cost camera. We'd like to point out that the pairs of corresponding images are fed into the domain shifting network (to be described in the next subsection) without any additional registration such that one has to expect small misalignment. This misalignment, however, can be reduced to a minimum by carefully positioning of the robot in front of the screen (as illustrated in Fig. 1), and the remaining difference does not seem to harm the accuracy of our approach.

a) Standard b) Cozmo recorded c) Network output

Fig. 4. Images from Pascal VOC [27] as illustrated in the left. Corresponding images recorded from Cozmo camera are in the middle, which introduce considerable distortion in terms of the dynamic range and color reproduction. The images in the right are outputs from the trained domain shifting network, with standard images as input.

B. A Domain Shifting Neural Network

To map high quality images to low quality images, a domain shifting network is proposed. This is a convolutional regression network whose input is a high quality image from standard dataset. The network is trained to mimic the corresponding low-quality camera image of a Cozmo robot by minimizing the reconstruction error (L_2 loss) between the network output and the corresponding low quality image. Once trained, this network provides a simple way to generate realistic low quality training samples even from previously unseen categories of high quality images.

Architecture: The network has a simple 2D convolutional network with 4 convolutional layers, as given below:

$$C^3_{3\to 64}\downarrow_2 \to C^3_{64\to 128}\downarrow_2 \to C^2_{128\to 64}\uparrow_2 \to C^2_{64\to 3}\uparrow_2$$

where $C^c_{a\to b}\ \downarrow_s$ represents convolution filter mapping from channel dimension of a to b and filter size of c and stride s. $C^c_{a\to b}\uparrow_s$ is a fractional strided convolution (transpose convolution) filter mapping from a channel dimension of a to b and using a filter size of c and stride s. This architecture is illustrated in Fig. 3, with the feature map dimension. The first 3 convolution layers are followed by a rectified linear units (ReLU) as a non-linearity, and we used a sigmoid after the last convolution layer. We do not use any batch normalization.

C. Simple Domain Adaptation

In case a labeled target domain dataset for the desired task (here classification of specific classes) is not available,

we use the trained domain-shifting network to map the labeled source domain data to the target domain. Fig. 4 shows the mapping from source images to the (lower quality) target domain using our domain shifting network on three examples of our validation data set. We can observe that distortions specific to the target domain in terms of the dynamic range and color reproduction are captured in the network output. This synthetic data along with source domain data is subsequently used to train a network in target domain on the relevant task (we consider classification). In this work, we use MobileNet [2], a light weight network architecture based on inverted residual blocks containing depthwise separable convolutions, with thin bottleneck layers as inputs and outputs of these blocks. This network architecture is very effective in a variety of computer vision tasks including image classification, object detection and semantic segmentation.

For domain adaptation, we train MobileNet for image classification using clean data together with synthetic data generated by our domain-shifting regression network. This way, we obtain domain invariant features from the network without explicitly trying to minimize the divergence between the features of source and target domain data. We consider two settings:

- The first setting we consider is zero-shot domain adaptation, where our approach does not see any target domain data which is useful in the relevant task. In this case, the unlabeled data used for training the domain shifting network does not contain the categories of images from the relevant task.
- The second setting we consider is unsupervised domain adaptation, in which our regression network is provided with images, which also include a small subset of images containing objects of the final categories of interest.

In both cases, the regression network is trained without label information (purely on image-to-image mappings) and subsequently provides synthetic data used to train the classifier network.

IV. Experiment Setup

A. Datasets

For training the domain shifting network, we use images from Pascal VOC dataset [27]. Pascal VOC dataset has a total of $17,125$ images in 20 classes for training and testing. For training the classifier network, and evaluating domain adaptation we use images from the Asirra dataset [28] and a subset of the Imagenet [29] dataset. The Asirra dataset has $18,697$ images of cats and dogs, with a training-test split of $80:20$. From the Imagenet dataset, we collect images to obtain 5 classes {cat, dog, cow, horse and sheep}, with approximately 2500 images in each of this classes[1] We split this dataset into training, validation and test sets in the ratio $60:20:20$. The images from all these datasets are also recorded by Cozmo robot placed in front of the screen in a dark room as described in section III-A. Cozmo recording took 1.189 seconds per image. This required around 4 hours for capturing Pascal VOC dataset, 6 hours for capturing Asirra dataset and 2.5 hours for recording the subset of Imagenet.

[1]We form 5 classes by grouping images from Imagenet classes: {cat, alley cat, Burmese cat, domestic cat}, {cow, dairy cattle}, {dog, Australian terrier, golden retriever, hunting dog, Labrador retriever}, {draft horse, farm horse, horse, male horse, racehorse, wild horse}, {black sheep, domestic sheep, sheep, wild sheep}.

Fig. 5. Sample images from "Cozmo in wild".

We consider two tasks of interest: *i)* a two-way classification between 2 classes, cats and dogs, and *ii)* the classification into our 5 classes {cat, dog, cow, horse and sheep}. For two-way classification (zero-shot), the domain shifting network is trained on the 18 remaining classes of the Pascal VOC dataset along with the corresponding low quality images recorded by Cozmo robot. High quality cats and dogs images from the Asirra training set are then mapped to their corresponding low-quality versions by the domain shifting network, which is used to train the classifier network. The evaluation is performed on the low-quality Cozmo captured test images from Asirra dataset. Furthermore, we printed images of 4 different dogs and 6 different cats on a paper and captured 233 images of these cats and dogs under varying illuminations using the Cozmo camera placed at different distances and orientations. We call this setting "Cozmo in wild", which is used only in the evaluation. Some samples from this setting are shown in Fig. 5.

For the $5-$way classification, we again exclude images from the 5 classes of interest from the Pascal VOC dataset, and use the remaining ones for training the domain shifting network along with the corresponding low quality Cozmo recorded images. Subsequently, the domain shifting network is applied to the 5 classes we formed from the Imagenet training set to yield lower quality images, which are subsequently used to train the classification network.

Finally, we refer to the corresponding *unsupervised domain adaptation* approaches by training the regression network on the full Pascal VOC dataset and not leaving the classes of interest out. The training and evaluation of the classification network, however, remains identical to the zero-shot case.

B. Baselines

For comparisons, we obtain the performance references for the fully supervised classification in the source (clean image)

Approach	Standard	Cozmo	Cozmo in wild
Source Supervised	97.86%	86.97%	90.13%
Ours Unsupervised	98.76%	94.67%	91.27%
Ours zero-shot	98.60%	94.24%	95.28 %
Cozmo Supervised	97.40%	95.00%	92.27%

TABLE I
PERFORMANCE COMPARISON FOR $2-$WAY CLASSIFICATION. THE REPORTED NUMBERS ARE CLASSIFICATION ACCURACY

Approach	Standard	Cozmo
Source Supervised	92.87%	73.49%
Ours Unsupervised	91.66%	77.56%
Ours zero-shot	92.09%	76.39%
Cozmo Supervised	84.88%	80.15%

TABLE II
PERFORMANCE COMPARISON FOR $5-$WAY CLASSIFICATION. THE REPORTED NUMBERS ARE CLASSIFICATION ACCURACY

domain, and the target (Cozmo captured) domain. We train fully supervised classifiers for both domains by retraining an Imagenet-pretrained MobileNet V2 for the 2 and 5 classes respectively.

Additionally, we compared to the zero-shot domain of [20], but unfortunately were not able to even beat the naive supervised training on the source images (even not when pretraining on the latter). Thus, we decided to leave out the specific accuracies of this approach in our numerical results below. Moreover, we tried to compare to the adversarial domain adaption approach in [30], for which code is provided at https://github.com/jvanvugt/pytorch-domain-adaptation for an MNIST classification example. However, to show a fair comparison we needed to adapt the classifier to the same network architecture we used, i.e., MobileNet-v2, which subsequently required an adaptation of the discriminator (which otherwise was too weak). Unfortunately, we, again, were unable to find a suitable architecture that improved the results of the naive supervised training on the source domain. While we do believe that adversarial domain adaptation techniques can be very powerful, our experiments demonstrate that balancing the two players in the adversarial training can be a difficult task. More specifically, we did not manage to reach a Nash-equilibrium our adapted discriminator did not manage to decrease the loss.

C. Training Details

We used Pytorch 1.1.0 and Python 3.6.9 for all the experiments. We have made our code available at https://github.com/Guru-Uni-siegen/Domain-Shifting-Network. We now describe the details of training for both the domain shifting network and the classification network.

1) Domain Shifting Network: For training the domain shifting network, we use the Adam optimizer [31] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, with an initial learning rate of 0.01, which is decreased by a factor of 0.5 every 30 epochs, and train for a total of 100 epochs using a batch size of 32.

2) Classification Networks: We train all classification networks using stochastic gradient descent with cyclical learning rate scheduling [32] with learning rate increasing exponentially from $1e-5$ to $1e-3$ in 20 steps. We use a batch size of 32 and train for 100 epochs and select the model with the best validation error for testing. We start our training with a MobileNet-v2 pretrained on Imagenet and freeze the weights of the first 100 out of the total number of 157 layers.

V. RESULTS

A numerical comparison of both the 2- and 5-classification problems with the baseline of training the same network on the source data only is given in Tables I and II, respectively. For comparison purposes, we also include the oracle network that is trained on the cozmo-recorded data directly (which we assume to be unavailable).

In both, $2-$way and $5-$ way classification, we can observe that networks trained on the source domain do not yield a high classification performance on the target domain. The proposed unsupervised as well as zero-shot domain adaptation techniques improve upon the source supervised network, with our zero-shot approach yielding improvements of 7.27% and 2.9% for the two tasks, respectively. According to the oracle network our performance is nearly optimal for the 2-class classification and about half-way in between the oracle and naive approach for the 5-class classification. It is interesting to see that the gain in accuracy on the cozmo images came at no price in accuracy on the higher quality source domain images.

Comparing the unsupervised and zero-shot approaches, we can see that having paired source and target images of the categories relevant to the classification task does help (showing improvements of 0.43% and 1.17%), but does not yield a large margin.

The general trend of improvement in classification accuracy over standard supervised network baseline is also observed on the 'Cozmo in wild' dataset in Table I. However, we also note that this test dataset is small and is not diverse enough to infer the average improvement in performance in this domain.

VI. DISCUSSION

Domain adaptation addresses the problem of using machine learning algorithms, when there is a shift in the data distribution. The proposed approach presents a practical solution in several real-world applications where the tasks of a robot, or the classes it has to determine, change over time: By recording paired images of high quality and ones recorded by the on-board camera, one can train a domain-shifting network once, and subsequently exploit any online data-base fed through the domain-shifting network to train the robot on new tasks or classes without the need to record new data with the robot.

Beyond the possibility to avoid the cumbersome labeling of data this way, we'd like to point out that even recording separate images with Cozmo took about 1.19 second per image, rendering the acquisition of gigantic datasets impossible. On

the contrary, the forward pass of the domain-shifiting network is in the order of milliseconds.

Instead of explicitly minimizing divergence between source and target domain data distributions [6]–[8] or minimizing the distance between features of images from both domains [20], our approach implicitly learns invariant representation by training on both source and target domain for the classification task. This has the advantage that the performance in the target domain improves significantly, while maintaining good performance in the source domain. A small caveat, however, is the increase in computational load during training due to the increase in training data (considering the union of source and simulated target domain data).

VII. Conclusion

We proposed a framework to map high quality images to corresponding low quality version using a simple convolutional regression network. This can be used to efficiently generate low quality images of previously unseen categories. We propose a simple domain adaptation approach where we utilize such synthetic data when real labeled data in target domain is not available. Our experiments demonstrate the merit of our simple approach, showing an improved accuracy on the target domain at no sacrifice of source domain accuracy.

References

[1] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[2] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[3] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size," *CoRR*, vol. abs/1602.07360, 2016.

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2014, pp. 647–655.

[5] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *Proceedings of the International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2016, pp. 1–6.

[6] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 443–450.

[7] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 41, no. 4, pp. 801–814, 2018.

[8] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 285–10 295.

[9] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 469–477.

[10] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon, "Pixel-level domain transfer," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 517–532.

[11] M. Ghifary, W. B. Kleijn, M. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 597–613.

[12] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3320–3328.

[13] B. Chu, V. Madhavan, O. Beijbom, J. Hoffman, and T. Darrell, "Best practices for fine-tuning visual classifiers to new domains," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 435–442.

[14] T. Yao, Y. Pan, C.-W. Ngo, H. Li, and T. Mei, "Semi-supervised domain adaptation with subspace learning for visual recognition," June 2015.

[15] S. Ao, X. Li, and C. X. Ling, "Fast generalized distillation for semi-supervised domain adaptation," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[16] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, "Semi-supervised domain adaptation via minimax entropy," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[17] H. Huang, Q. Huang, and P. Krahenbuhl, "Domain transfer through deep activation matching," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 590–605.

[18] S. Roy, A. Siarohin, E. Sangineto, S. R. Bulo, N. Sebe, and E. Ricci, "Unsupervised domain adaptation using feature-whitening and consensus loss," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[19] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei, "Transferrable prototypical networks for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 2239–2247.

[20] K.-C. Peng, Z. Wu, and J. Ernst, "Zero-shot deep domain adaptation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 764–781.

[21] A. Kumagai and T. Iwata, "Zero-shot domain adaptation without domain semantic descriptors," *CoRR*, 2018.

[22] M. Ishii, T. Takenouchi, and M. Sugiyama, "Zero-shot domain adaptation based on attribute information," in *Proceedings of the Eleventh Asian Conference on Machine Learning (ACML)*, ser. Proceedings of Machine Learning Research, vol. 101. PMLR, 17–19 Nov 2019, pp. 473–488.

[23] J. Wang and J. Jiang, "Conditional coupled generative adversarial networks for zero-shot domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1125–1134.

[25] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision (ICCV),*, 2017.

[26] Z. Murez, S. Kolouri, D. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4500–4509.

[27] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision (IJCV)*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[28] J. Elson, J. J. Douceur, J. Howell, and J. Saul, "Asirra: A captcha that exploits interest-aligned manual image categorization," in *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, Inc., October 2007.

[29] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[30] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of the International Conference on International Conference on Machine Learning (ICML)*, 2015, pp. 1180–1189.

[31] D. P. Kingma and J. Ba, "Adam: A method for sstochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] L. N. Smith, "Cyclical learning-rates for training neural networks," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 464–472.