Unsupervised Domain Adaptation Utilizing CycleGAN

Anonymous Authors[1]

Abstract

Conventional deep learning methods for in computer vision considers a scenario where the training data and test data are generated from the same distribution. But might not be the case in the real world. Machine Learning models failed to generalize across different domains due to this domain shift. Domain adaptation is a technique which helps model learn more transferable features and thus reducing this domain shift.

Image-to-image translation learns a mapping between domain-1 and domain-2 using a training set of aligned image pairs. This could solve the domain adaptation problem by translating test images to the domain of source images. However, getting this paired training data is not possible. CycleGAN solves this problem by learning $\mathcal{F}$ and $\mathcal{G}$ mappings such that $\mathcal{G} : \mathcal{X} \rightarrow \mathcal{Y}$ and $\mathcal{F} : \mathcal{Y} \rightarrow \mathcal{X}$. This mappings generates images from domain-1 which are indistinguishable from the distribution of the domain-2 using an adversarial and cyclic consistency loss.

We present an approach utilizing these cyclic consistency and adversarial loss to train a mapping from a target to a source domain. Here, the source domain is the domain on which our deep model is trained and the target domain is the domain on which we are making the predictions. We are using Office-31 dataset, a standard domain adaptation dataset consisting of three domains - Amazon, DSLR, Webcam, to show how CycleGAN translates between these domains and how it is useful for the classification task. Qualitative results and insights are presented for inter-domain changes after every few epochs to show how CycleGAN training is going on. Then testing it on the domain adaptation task to show how we can improve the target domain classification performance.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction

There are other literature focussing on image-to-image translation problem but it was for paired imaages, i.e. we needed to feed the corresponding style image to the content image to generate a new image with the given content and style. (Gatys et al., 2016) in fig. 1 showed the generated image correspond to content and style image.

The motivation of implementing CycleGAN is to illustrate how unpaired image-to-image translation works. We have reproduced the results mentioned in the paper *Unpaired image-to-image translation using cycle-consistent adversarial networks* (Zhu et al., 2017). After reproducing the experiments mentioned in the paper, we run this algorithm on a standard domain-adaptation datasets Office-31.

This paper demonstrates the following things:

1. Reproducing CycleGAN results
2. Implementing CycleGAN on a domain adaptation dataset, Office-31.
3. Training Resnet18 on Office-31 dataset and testing it on cross-domains
4. Explaining poor accuracy of source domain model on cross-domains using a TSNE plot
5. Showing translating images on the pixel space will not help adapting for a different domain

2. Substantive review and critique

2.1. Federated Adversarial Domain Adaptation (Peng et al., 2020)

Fig. 3 demostrates FADA algorithm used in (Peng et al., 2020) paper. FADA introduces a novel Unsupervised Federated Domain Adaptation(UFDA). Federated Learning is a learning paradigm where different clients come together to train a shared global model without sharing their personal data. Here, *data privacy* is of utmost important. Given their are multiple clients and they have their own way to collect data, we cannot expect their data distributions to match with other clients and thus we observed a domain shift there. This

Figure 1. A: Content image, B-F(small): Style images, B-F(large): Generated images. *Ref*: (Gatys et al., 2016).

UFDA method tries to solve it by aggregating the different gradients with dynamic attention mechanism.

The FADA algorithm mainly discusses the following key-points:

Target bound: Authors showed that the target domain is a convex combination of hypotheses fitted on multiple source domains by providing a novel generalization bound for the transfer learning error. (Mansour et al., 2009) have provided a multi-source domain bound by assuming target domain as mixture a mixture of N source domains. On the contrary, (Peng et al., 2020) has assumed target domain to be a source domain has included $\mathcal{H}\Delta\mathcal{H}$ discrepancy (Ben-David et al., 2010) and the VC-dimensional constraint (Vapnik, 2013) as well. Also, this bound considers no overlap Eachbetween source and target domains whereas (Zhu et al., 2017) considers a more basic case where multiple source and target domains shares the same hypothesis.

Dynamic attention model: The Dynamic attention model is used to aggregate the updated weights send to the server from different source domain clients. This attention is used to increases the weight of nodes whose gradients contributes more to learn the target domain features. They are using gap-statistics (Tibshirani et al., 2001) to evaluate how good the target features can be clustered.

Adversarial alignment and feature disentanglement: For aligning source and target domain, (Peng et al., 2020) used two steps: a local feature extractor and global discriminator as the source data is private to clients and cannot share globally. For feature disentanglement, they have trained a K-way classifier to predict the correct labels with domain invariant and domain specific features. Also, minimizing mutual information between these two features is further helping the disentanglement.

(Peng et al., 2020) also introduces Federated Adversarial Domain Adaptation(FADA) model which learn to extract domain-invariant features(corresponding to frontal object, without background, different illuminance, etc) by using "adversarial domain alignment and a feature disentangler".

2.2. Diverse image-to-image translation via disentangled representation(Lee et al., 2018)

This paper proposes a method to learn mapping between different visual domains. As images are not always in pairs and can be quite different from each other, it is quite tough to find some relation between them. (Lee et al., 2018) use disentanglement representation framework to learn to generate different outputs where images are not paired.

Authors have used two different semantic spaces to embed features in it. Domain-invariant content space captures the information common in both domains. Domain-specific attribute space models the variations (like different lightning conditions or different background) given the same main content(like frontal image of a dog).

Content adversarial loss is used to avoid domain specific cues. Latent regression loss for learning invertible mappings and cross-cycle consistency loss to handle unpaired dataset.

Comparison with other I2I methods: CycleGAN (Zhu et al., 2017) considers $\mathcal{X}$ and $\mathcal{Y}$ domains cannot map onto a same latent space and thus uses separate latent spaces. UNIT (Liu et al., 2017) assumes $\mathcal{X}$ and $\mathcal{Y}$ domains can be mapped onto a shared latent space. But this paper has made the reasonable assumption considering two different latent spaces for shared content space and different attribute spaces for different domains.

Experiments: (Lee et al., 2018) have shown extensive qualitative and quantitative results comparing with different existing image-to-image translation techniques.

Quantitative evaluation: Authors have conducted a "Realism vs. diversity" test of the generated images asking users which one is more realistic. It is very appreciable. Authors have also provided the data for diversity and reconstruct error. While discussing the reconstruction ability, they have conducted an experiment on the edge-to-shoes dataset for measuring the usefulness of disentangled coding.

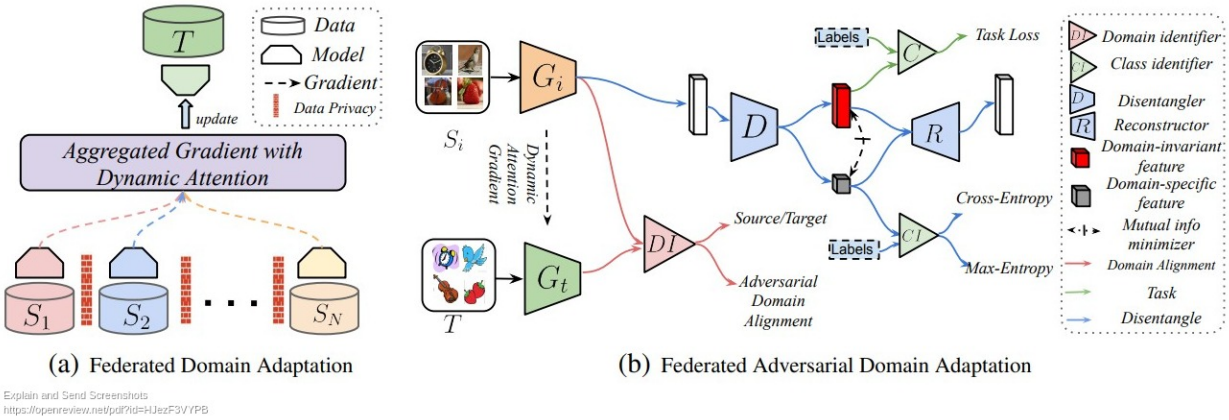


Figure 2. FADA algorithm. *Ref*: (Peng et al., 2020)

Domain adaptation: Authors used this method to train a unsupervised domain adaptation model. First translated source-labelled images to the target domain and then training a target-classifier on generated images helped to achieve a good accuracy on target domain data.

2.3. Unpaired image-to-image translation using cycle-consistent adversarial networks (Zhu et al., 2017)

To understand the concept of cyclegan, (Zhu et al., 2017), let us first understand the concept of Genetative Adversarial Network or GAN(Goodfellow et al., 2014). A GAN essentially consists of two competing neural networks, a generator and a discriminator. The task of the generator is to produce images as close as possible to the real image, while the task of the discriminator is to differentiate between the fake generated image by the generator and real image. The two neural-networks compete against each other, until the generator gets trained enough to produce absolutely real like images to the extent that the discriminator fails to distinguish among the two.

The main goal of cyclegan is to translate an image from source domain $\mathcal{X}$ to target domain $\mathcal{Y}$, such that no pairwise training examples are presented during the time of training. To approach this problem, the authors first look forward to make a network learn the mapping from $\mathcal{X}$ to $\mathcal{Y}$, such that for all possible $\mathcal{X}$, that network can produce images $\mathcal{Y}$' which are quite indistinguishable from the original $\mathcal{Y}$. Let us call this network $\mathcal{G}$.

Again if we have another network, $\mathcal{F}$, that learns the opposite, i.e., takes in the distribution of $\mathcal{Y}$ and tries to obtain $\mathcal{X}$; we can easily say that $\mathcal{F}$ is an inverse of $\mathcal{G}$. As apparent from the above discussion, the two networks used here are GANs. However, to connect these two parallel networks, apart from the trivial losses, an important loss to discuss about is the "cycle consistency loss" that encourages $\mathcal{F}(\mathcal{G}(\mathcal{X}))= \mathcal{X}$ and $\mathcal{G}(\mathcal{F}(\mathcal{Y}))= \mathcal{Y}$

3. Implementation

This implementation is in three stages:

1. Implementation of CycleGAN
2. CycleGAN on Office-31 dataset
3. Training and Domain adaptation

3.1. Implementation of CycleGAN:

The motivation to implement CycleGAN is to show how unpaired image-to-image translation works. We have implemented two major loss functions, the adversarial loss and cyclic consistency loss.

Adversarial loss: Inspired from the GAN (Goodfellow et al., 2014) we use generator-discriminator pair where generator tries to generate a sample similar to the second domain from the first domain sample. The discriminator tries to find if the generated sample is real or fake. Precisely, we can put it in the following equation:

$\mathcal{L}_{\mathcal{GAN}}(\mathcal{G}, \mathcal{D}_{\mathcal{Y}}, \mathcal{X}, \mathcal{Y}) = \mathbb{E}_{y \sim P_{data(y)}}[log \mathcal{D}_{\mathcal{Y}}(y)] + \mathbb{E}_{y \sim P_{data(x)}}[log(1 - \mathcal{D}_{\mathcal{Y}}(\mathcal{G}(x)))]$.

Cycle consistency loss: As $\mathcal{G}$ is trying to map $\mathcal{X} \rightarrow \mathcal{Y}$ and $\mathcal{F}$ mapping $\mathcal{Y} \rightarrow \mathcal{X}$, we get a complete cycle $y \rightarrow \mathcal{F}(y) \rightarrow \mathcal{G}(\mathcal{F}(y)) \approx y$. We are promoting this behaviour by adding a cycle-consistency-loss:

$\mathcal{L}_{\mathcal{CYCLE}}(\mathcal{G}, \mathcal{F}) = \mathbb{E}_{x \sim P_{data(x)}}[||\mathcal{F}(\mathcal{G}(x)) - x||_1] + \mathbb{E}_{y \sim P_{data(y)}}[||\mathcal{G}(\mathcal{F}(y)) - y||_1]$.

The paper is using the following datasets: apple2orange,

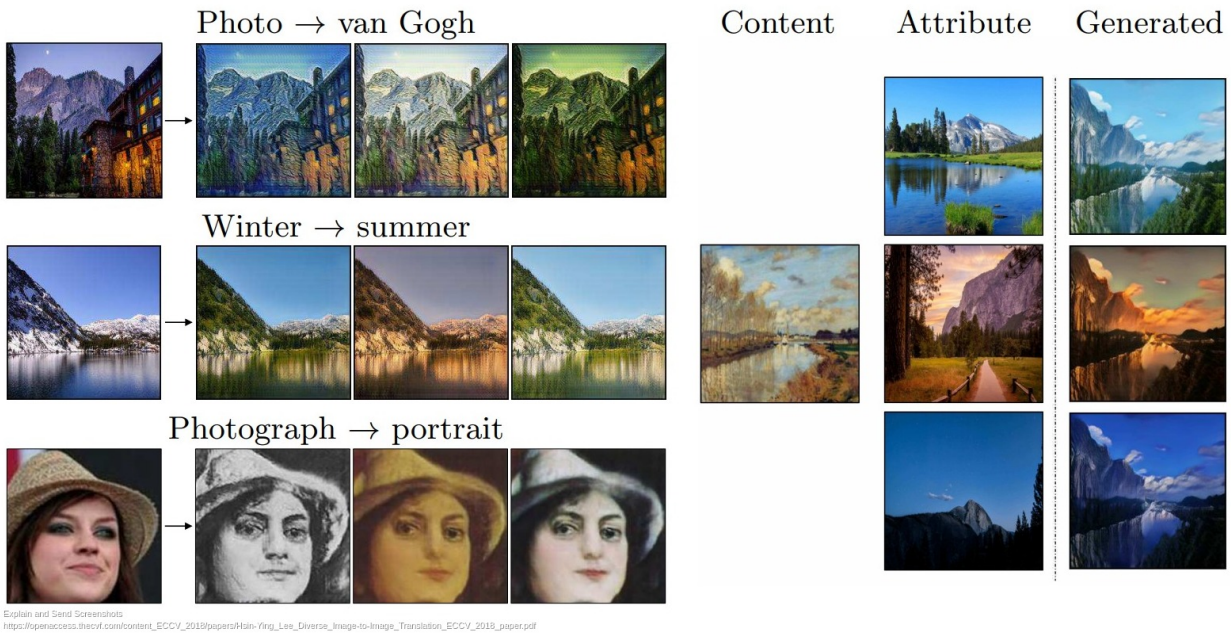


Figure 3. FADA algorithm. *Ref*: (Lee et al., 2018)

summer2winter, horse2zebra, monet2photo, cezanne2photo, ukiyoe2photo, vangogh2photo, maps, cityscapes, facades, iphone2dslr-flower. I have downloaded few (and not all as our aim is to do Domain Adaptation on Office-31) of them and have reproduced the results.

For reproducing the results, we have started with their official implementation and tweaked the data downloaders and loaders correspondingly. This code is also inspired by the Pytorch-DCGAN code (https://github.com/pytorch/examples/tree/master/dcgan). Thereafter changing the main function to accomodate different dataset training and utils visualizer for visualizing the results.

3.2. CycleGAN on Office-31 dataset

Office-31 dataset: "The Office dataset contains 31 object categories in three domains: Amazon, DSLR and Webcam. The 31 categories in the dataset consist of objects commonly encountered in office settings, such as bicycles, monitors, and calculators (see fig. 6). The Amazon domain contains on average 90 images per class and 2817 images in total. As these images were captured from a website of online merchants, they are captured against clean background and at a unified scale. The DSLR domain contains 498 low-noise high resolution images (4288×2848). There are 5 objects per category. Each object was captured from different viewpoints on average 3 times. For Webcam, the 795 images of low resolution (640×480) exhibit significant noise and color as well as white balance artifacts" - (Koniusz et al., 2017).

I will reuse the training part of the code. For this data, I need to build a dataset class to load the data in this format and then to integrate it with the main code. Then running all those experiments and visualise the results.

We have implemented CycleGAN on this Office-31 (https://www.cc.gatech.edu/~judy/domainadapt/) dataset to translate images from one domain to another. We have also shown how CycleGAN learns gradually (epoch by epoch) to consider different aspects and properties of the domain to translate properly. We have prepared a script to download the raw images of Office-31 dataset from the official web page of the dataset. This script then unpacks the dataset and prepares train and test split for the classification task training. Also, it copies the dataset in a format such that CycleGAN can use it to train. We have reused the CycleGAN training part here with custom dataset class to load the Office-31 data. Then I have integrated this loader with the main code and have created a visualizer for it. As, CycleGAN requires dataset to put it in a certain way while loading and training, and many arguments, we are maintaining a bash script to do all these things.

3.3. Training and Domain Adaptation

We are using a pretrained ResNet-18 model trained on Imagenet dataset. We changed its last 1000-dimensional layer

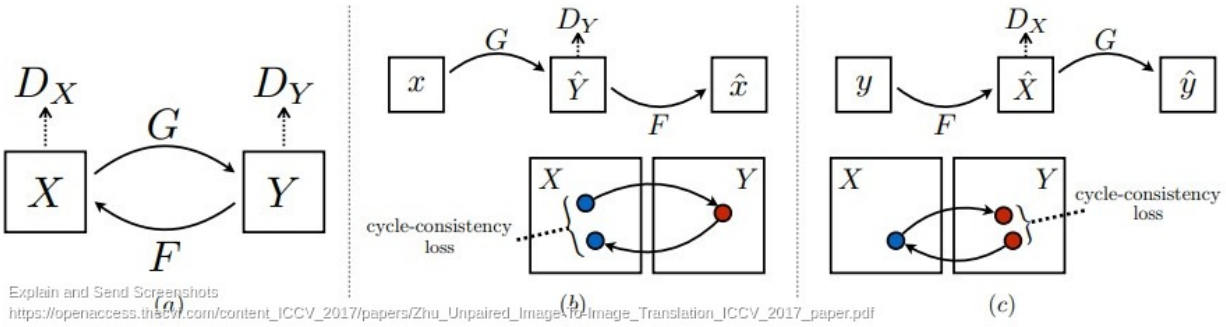


Figure 4. (a) CycleGAN model (b) Forward cycle consistency loss (c) Backward cycle consistency loss *Ref*: (Zhu et al., 2017).

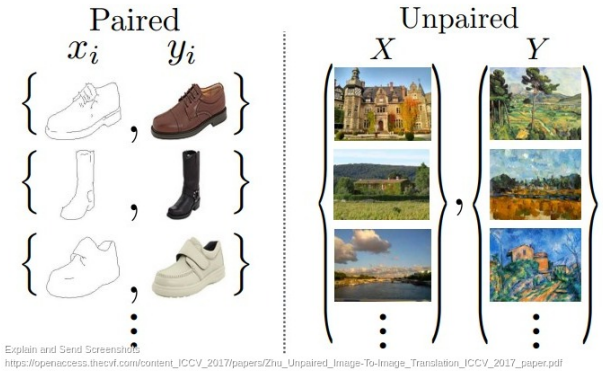


Figure 5. An example of paired and unpaired data. *Ref*: (Zhu et al., 2017).

Figure 6. Different domains in office-31 dataset.

	Accuracy			
	Train	Test		
Source Domain	Source	Amazon	DSLR	Webcam
Amazon	96.13	**87.94**	31	59.12
DSLR	90.2	46.63	**85**	81.97
Webcam	96.226	47.34	55	**98.11**

Table 1. Training and testing accuracy of Resnet18 on Office-31 dataset. Cross-domain test accuracy is also there.

to a 31-dimensional layer as Office-31 dataset has 31 classes. Then we finetune the model to fit on the data. The training script finetunes the model and saves weights for all the epochs. Also, it tests the model performance on the same domain as well as on cross-domains.

4. Evaluation

4.1. CycleGAN evaluation

We have reproduced the results for the Horse2zebra, Map2sat and Vangogh2photo. Here, in the dataset, for each section, there are two groups. For example, for Horse2zebra subset, we have two folders one for Horse and the other for the Zebra class. The algorithm is treating one folder as one domain and another one as a different domain. This CycleGAN algorithm is cyclically translating one image from one domain to another and then translating it back to the parent domain. In this way, the algorithm is enforcing cyclic consistency between the two domains and adding a robust objective function for the feature learning purpose. **Cycle consistency loss** is encouraging $\mathcal{F}(\mathcal{G}(\mathcal{X}))= \mathcal{X}$ and $\mathcal{G}(\mathcal{F}(\mathcal{Y}))= \mathcal{Y}$.

Fig. 3 is *Photo to Van Gogh* art translation. Here, domain-1 contains the real life photos. Domain-2 contains images of Van Gogh paintings. Note that in the dataset, the images are not paired but random.

Similarly, for the following samples, *Horse to Zebra* (fig. 4) and *Maps to Satellite* (fig. 5) is there.

Figure 7. TSNE plot for visualizing different domains (plotted only ten classes for clearity)

4.2. CycleGAN on Office-31 dataset

We have trained our CycleGAN model on Amazon and DSLR domains of the Office-31 dataset. In fig. 16 we have shown how CycleGAN model is performing in the initial epochs and in final epochs. Amazon images are captured against clean background (complete white) and at a uniform scale, which are taken from the a website of online merchants. The DSLR images are low-noise, high resolution images captured from different view points. The objects are mostly on a table or floor having an uniform background.

In fig. 16, we can see that initially, the model is confused between the object and the background. It is trying to add the background but along with it, model is also changing the object's colour creating some artifacts. But in the next image, we can see that the model is able to distinguish between the object and the background and thus changing the background only. Though the difference between Amazon and DSLR is the high-resolution, low noise and the background, but the model is capturing only the background differences. It is also obvious as we are preprocessing the images and resizing it to (224,224) and thus loosing on the high-resolution. Also, generated images are not that natural looking and sometimes can be quite noisy (though recent literature is generating quite natural images).

In ig. 19, CycleGAN is translating a DSLR domain image to Amazon domain perfectly removing the table background and making it completely white. Here, the model has learnt the difference between foreground object and the background and thus changing the background only.

In fig. 22, the Amazon domain image contains illustration of the signal showing how a laptop can connect to a modem. There isn't any real significance of this signal and thus model is not able to capture it, thus blurring the generated image.

In fig. 27, we can see the difference in lighting condition. As Webcam images are low quality, noisy images with poor lighting conditions, model learn to generate images with bright light and more noise. Here, unlike Amazon to DSLR, CycleGAN model is learning to change the foreground ob-

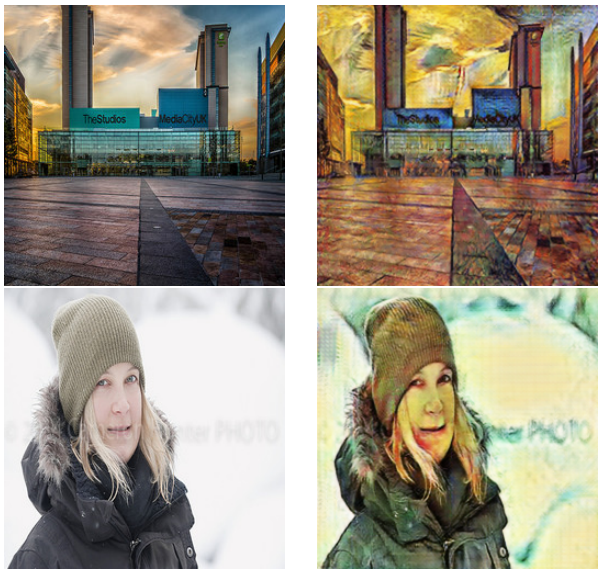


Figure 8. Photo to Van Gogh style

Figure 9. Horse to Zebra style

ject as well and not only the background.

Also, one important point to note in this Amazon to DSLR translation is that it is keeping the background objects as it is. On contrary to this, in Webcam to Amazon (fig. 32) translation, model is removing the background keeping the monitor and wires as it is but also removing the speaker in the top right corner. For the laptop sample in 32, model is able to remove the table but not the bag in the background as the bagpack is also a class in the dataset.

Figure 10. Map to satellite

. Real Amazon . Fake DSLR

. Earlier epochs of Amazon-DSLR CycleGAN training

. Real Amazon . Fake DSLR

Figure 16. Amazon to DSLR (initial and latter epochs)

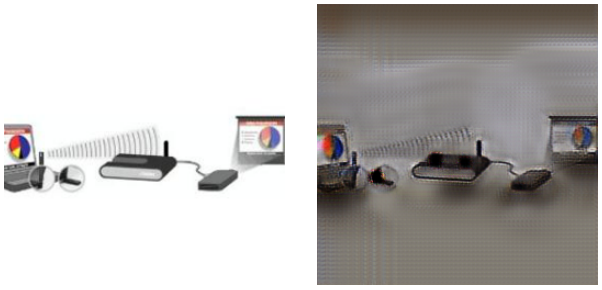
4.3. Training and Domain adaptation

In table 1, we can see that the testing accuracy is high if the training domain and testing domain is same. Also, as background makes a big difference, we can see less correlation between Amazon and DSLR, and Amazon and Webcam. But DSLR and Webcam domains are more correlated to each other as the difference between those two is of quality and noise, resizing a high quality DSLR image to a low-quality image will itself look like an image from the Webcam domain and thus there isn't much difference between them.

To show this difference and correlation between domains, we have plotted the TSNE visualizations for the above. This TSNE is plotted using the pre-softmax features of the

. Real DSLR . Fake Amazon

Figure 19. DSLR to Amazon



. Real DSLR . Fake Amazon

Figure 22. DSLR to Amazon

. Real DSLR . Fake Webcam

. Real DSLR . Fake Webcam

Figure 27. DSLR to Webcam

Resnet50 trained on Imagenet to keep it unbiased. In fig. 7, we can see for the class bike that DLSR and Webcam samples are overlapping and the Amazon samples are quite away from them. Similarly, we can verify this correlation from other classes as well (backpack, calculator, dsek chair, etc.)

Then we checked the misclassified images for cross-domain, translated it to the source domain and again classify it. This way, we are able to classify it correctly. Fig. 38 and 35 shows the misclassified and translated reclassified object. Though CycleGAN generates images visually similar to the other domains , it is quite tough to get an image without any artifacts. And thus it cannot directly utilize for the domain adaptation.

5. Conclusion

CycleGAN translates images from domain-1 to domain-2 without having paired image-to-image relation between them. Also, we don't need any class-wise annotated data for training a CycleGAN model. This makes this very helpful for Unsupervised training scenarios. In our case, CycleGAN is the most appropriate candidate for Unsupervised Domain Adaptation as we don't have annotated samples for the target domains (or else we would have gone for Supervised Domain Adaptation techniques). But the quality of the generated images is not good which is making classifier difficult to classify it.

From here, we can conclude that adapting on pixel level may not solve the Domain shift problem and we need to work at feature (semantic) level as well. For the future work, we can try aligning feature extractor's projection in the feature space for samples from the different domains. This will help reduce domain gap between them.

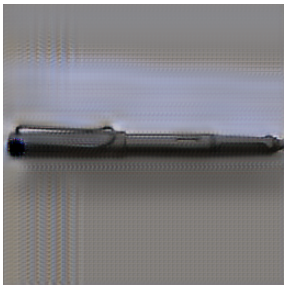

. Real Webcam . Fake Amazon


. Real Webcam . Fake Amazon

Figure 32. Webcam to Amazon


. DSLR - prediction: file cabinet . Fake Amazon - prediction: Stapler

Figure 35. ”Stapler” tested on model trained on Amazon

. Amazon - prediction: paper notebook . Fake Webcam - prediction: pen

Figure 38. ”Pen” tested on model trained on Amazon

References

Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. URL https://link.springer.com/article/10.1007/s10994-009-5152-4.

Gatys, L. A., Ecker, A. S., and Bethge, M. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. URL https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Koniusz, P., Tas, Y., and Porikli, F. Domain adaptation by mixture of alignments of second-or higher-order scatter tensors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4478–4487, 2017. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8100238.

Lee, H.-Y., Tseng, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. URL https://openaccess.thecvf.com/content_ECCV_2018/papers/Hsin-Ying_Lee_Diverse_Image-to-Image_Translation_ECCV_2018_paper.pdf.

Liu, M.-Y., Breuel, T., and Kautz, J. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pp. 700–708, 2017. URL https://proceedings.neurips.cc/paper/2017/file/dc6a6489640ca02b0d42dabeb8e46bb7-Paper.pdf.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation with multiple sources. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2008/file/0e65972dce68dad4d52d063967f0a705-Paper.pdf.

Peng, X., Huang, Z., Zhu, Y., and Saenko, K. Federated adversarial domain adaptation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJezF3VYPB.

Tibshirani, R., Walther, G., and Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00293.

Vapnik, V. *The nature of statistical learning theory*. Springer science & business media, 2013. URL https://www.wiley.com/en-us/Statistical+Learning+Theory-p-9780471030034.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. URL https://openaccess.thecvf.com/content_ICCV_2017/papers/Zhu_Unpaired_Image-To-Image_Translation_ICCV_2017_paper.pdf.