Adversarial Domain Adaptation for Screening Mammograms

TAO WANG

Master in Computer Science
Date: January 30, 2018
Supervisor: Hossein Azizpour
Examiner: Mårten Björkman
Swedish title: Detta är den svenska översättningen av titeln
School of Computer Science and Communication

Abstract

English abstract goes here.

Sammanfattning

Swedish abstract goes here.

Contents

1 Introduction **1**

2 Background **2**
2.1 Transfer Learning 2
2.2 Domain Adaptation 3
2.2.1 Definition 3
2.3 Generative Model 4
2.3.1 Variational Autoencoders 4
2.3.2 Generative Adversarial Network 6
2.4 Semantic Segmentation 7

3 Related work **9**
3.1 Adversarial Domain Adaptation 9
3.1.1 Feature-Level Adaptation 9
3.1.2 Pixel-Level Adaptation 10
3.2 Deep Semantic Segmentation 15

4 Methods **17**

Bibliography **18**

A Unnecessary Appended Material **22**

Chapter 1

Introduction

Chapter 2

Background

2.1 Transfer Learning

Machine learning has been quite successful in recent years. However, many machine learning methods are based on a common assumption: the training data and the testing data are sampled from a same distribution. When they come from different distributions, or the distribution changed, the trained model's performance will dramatically decrease. Transfer learning aims to solve this problem, which will transfer the knowledge learned from the original distribution to the new one. Here is how we define the transfer learning in a notation form[26]:

Firstly, let's define domain, a domain D contains two parts: a feature space $\mathcal{X}$, a marginal probability distribution $P(X)$. And $X = \{x_1, x_2, x_3, ..., x_n\}$, where $x_i \in \mathcal{X}$.

Then, when given a domain $D = \{\mathcal{X}, P(X)\}$, a task $\mathcal{T}$ is defined as $\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$, where $\mathcal{Y}$ is the label space, and $Y = \{y_1, y_2, y_3, ..., y_n\}$, where $y_i \in \mathcal{Y}$. And $P(Y|X)$ means the conditional probability, actually it is an objective predictive function. When given features X, our aim is to find the most possible labels Y. In the actually work, we need to find an approximate representation of $P(Y|X)$, which is also the goal of machine learning.

Finally, we can define the transfer learning: given source and target domains D_S and D_T, then source and target tasks $\mathcal{T}_S$ and $\mathcal{T}_T$. Transfer learning aims to help improve the performance of $P_T(Y|X)$ in $\mathcal{T}_T$ by using the knowledge learned from D_S and $\mathcal{T}_S$, where $D_S \neq D_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$.

As we have mentioned that in transfer learning, $D_S \neq D_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$, so the source and target conditions can vary in the following four ways:

- $\mathcal{X}_S \neq \mathcal{X}_T$. The target and source domain have different feature space, for example, two documents are written in different language, which will bring different feature space.
- $P_T(X) \neq P_S(X)$. The marginal probability distributions of source and target domain are different. The source and target features are sampled from different distribution. This scenario is called domain adaptation, which is the main focus in our paper.
- $\mathcal{Y}_T \neq \mathcal{Y}_S$. The label space are different. In the actually application of transfer learning, this scenario is rare.
- $P_T(Y|X) \neq P_S(Y|X)$. The conditional probability distributions of source and target are different. This scenario is also very common. But we do not focus on this problem in our project.

2.2 Domain Adaptation

2.2.1 Definition

Domain adaptation is one of the sub questions of transfer learning. Domain adaptation tried to build a model which is suitable on both source and target domain. The notation definition is: given source and target domains D_S and D_T, then source and target tasks $\mathcal{T}_S$ and $\mathcal{T}_T$. Domain adaptation aims to help improve the performance of $P_T(Y|X)$ in $\mathcal{T}_T$ by using the knowledge learned from D_S and $\mathcal{T}_S$, where $P_S(X) \neq P_T(X)$ and $P_S(Y|X) \approx P_T(Y|X)$.

Covariate shift
Shimodaira et al.[32] first proposed the concept of covariate shift, which is an important concept in domain adaptation. As we have already know that the source and target task's conditional probability distribution are same, then the difference between these two domains is called covariate shift. Shimodaira et al.[32] defined misspecified models to indicate the influence of covariate shift in model training. Although $P_S(Y|X) \approx P_T(Y|X)$ appears sweet, in the real world application, what we usually find is $P(Y|X, \theta)$. We optimize the parameter θ

to minimize the expected classification error. But it is hard to find a parameter θ^* which could fit $P(Y|X,\theta^*) = P(Y|X)$ for all $x \in \mathcal{X}$. So the model $P(Y|X,\theta^*)$ we find is called a misspecified model. The model's parameter θ^* is depend on $P(X)$. The difference between $P_S(X)$ and $P_T(X)$ will bring the difference between the model trained from source domain and the one from the target domain.

Single Good Hypothesis
We usually believe that there exists a model or hypothesis H^*, making the error on source domain $E_S(H^*)$ and the error on target domain $E_T(H^*)$ are both small.

Domain discrepancy and Error
Domain discrepancy is used to describe the difference between the source and target domain. Usually the smaller domain discrepancy is, the better performance we will get in the final model.

2.3 Generative Model

The goal of generative model is finding a function used to approximate fit the distribution of the original data. If we use $f(X;\Theta)$ represents such a function, so finding the parameter $P(\Theta)$ is a process of maximum likelihood estimation. The question is when the data distribution is complex, our f will also be complex. And deep neural network could be used to represent such a complex function. There are two successful frameworks used to build a generative model: variational autoencoder(VAE) and Generative Adversarial Network(GAN). In this project we mainly focus on the GAN.

2.3.1 Variational Autoencoders

Variational Autoencoders[19, 7](VAE) is based on encoder-decoder structure. Usually we will have a observed data x, for example an image and this observed data is generated by a latent code z. So training a encoder is trying to find a $q_\phi(z|x)$ and training a decoder is trying to find a $p_\theta(x|z)$. The training goal of the VAE is to maximize the following

likelihood function:

$$\log_\theta(x^{(1)}, x^{(2)}, ..., x^{(N)}) = \sum_{i=1}^{N} \log p_\theta(x^{(i)}) \tag{2.1}$$

Now we can use the $q_\phi(z|x^{(i)})$ to approach the posterior $p_\theta(z|x^{(i)})$. To estimate the similarity between these two distribution, we use Kullback-Leibler divergence to describe it.

$$\begin{aligned}
KL(q_\phi(z|x^{(i)})||p_\theta(z|x^{(i)})) &= \mathbb{E}_{q_\phi(z|x^{(i)})} \log \frac{q_\phi(z|x^{(i)})}{p_\theta(z|x^{(i)})} \\
&= \mathbb{E}_{q_\phi(z|x^{(i)})} \log \frac{q_\phi(z|x^{(i)})p_\theta(x^{(i)})}{p_\theta(z|x^{(i)})p_\theta(x^{(i)})} \\
&= \mathbb{E}_{q_\phi(z|x^{(i)})} \log \frac{q_\phi(z|x^{(i)})}{p_\theta(z, x^{(i)})} + \mathbb{E}_{q_\phi(z|x^{(i)})} \log p_\theta(x^{(i)}) \\
&= \mathbb{E}_{q_\phi(z|x^{(i)})} \log \frac{q_\phi(z|x^{(i)})}{p_\theta(z, x^{(i)})} + \log p_\theta(x^{(i)}) \\
&= \mathbb{E}_{q_\phi(z|x^{(i)})} \log \frac{q_\phi(z|x^{(i)})}{p_\theta(z)} - \mathbb{E}_{q_\phi(z|x^{(i)})} \log(p_\theta(x^{(i)}|z)) + \log p_\theta(x^{(i)}) \\
&= KL(q_\phi(z|x^{(i)})||p_\theta(z)) - \mathbb{E}_{q_\phi(z|x^{(i)})} \log(p_\theta(x^{(i)}|z)) + \log p_\theta(x^{(i)})
\end{aligned}$$

So

$$\log p_\theta(x^{(i)}) = \mathbb{E}_{q_\phi(z|x^{(i)})} \log(p_\theta(x^{(i)}|z)) + KL(q_\phi(z|x^{(i)})||p_\theta(z|x^{(i)})) - KL(q_\phi(z|x^{(i)})||p_\theta(z)) \tag{2.2}$$

This is a basic equation of VAE. And $KL(q_\phi(z|x^{(i)})||p_\theta(z|x^{(i)}))$ is non-negative. If $q_\phi(z|x^{(i)})$ and $p_\theta(z|x^{(i)})$ are totally equal, this term will be zero. So we could turn to optimize the

$$\sum_{i=1}^{N} \mathbb{E}_{q_\phi(z|x^{(i)})} \log(p_\theta(x^{(i)}|z)) - KL(q_\phi(z|x^{(i)})||p_\theta(z)) \tag{2.3}$$

instead of $\sum_{i=1}^{N} p_\theta(x^{(i)})$.

The first term could be optimized by stochastic gradient descent and using minibatch training samples. This term could be viewed as reconstruction error by using L2 loss. And in the second term similarly, $p(z)$ could be viewed as a normal distribution $N(0, 1)$, the encoder $q_\phi(z|x)$'s output is the mean and the variance of the normal distribution wanted to approximate. The training process could also be done by backpropagate algorithm.

Variational Autoencoders is quite efficient in generating samples, while VAE usually tend to result in blurry images[7] because of the pixel-wise reconstruction losses.

2.3.2 Generative Adversarial Network

Generative Adversarial Network(GAN) was proposed by Ian Goodfellow[12] inspired from two-players game. The roles of this two players in GAN are generative model and discriminative model. Following graph shows the structure of GAN.

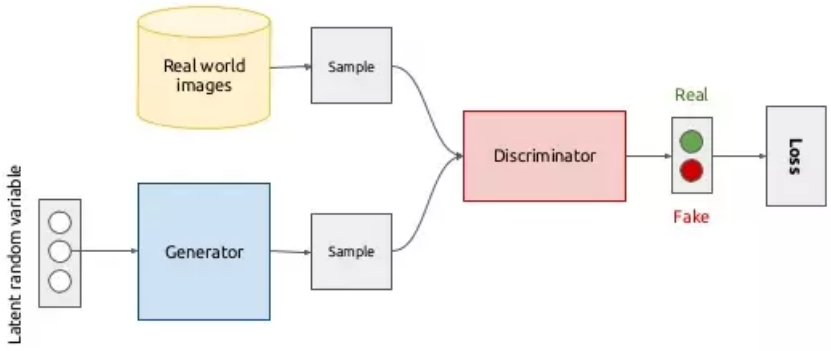

Figure 2.1: GAN's structure used in computer vision

Generative model is used to simulate the real world data's distribution, and the discriminative model is a binary classifier, used to classify the fake and the real data. A generator receive a random noise and generate a fake picture, and a discriminator tries to judge whether this picture is fake or not. The goal of the generator is to confuse the discriminator, while the discriminator aims to resist this confusion. Here is the representation of this minmax game:

$$\min_G \max_D V(D,G) = E_{x\sim P_{data}}[log(D(x)] + E_{z\sim P_z} log(1 - D(G(z)) \quad (2.4)$$

In this function, P_{data} is the real data's distribution, and the P_z is the fake distribution simulated by generator. z is the input noise. And usually the training method for GAN is mini-batch gradient descent.

So the loss function for each m size batch on discriminator D and generator G is:

$$L_D = \frac{1}{m}\sum_{i=1}^{m}[-log(D(x^{(i)}) - log[1 - (D(G(z^{(i)}))] \tag{2.5}$$

$$L_G = \frac{1}{m}\sum_{i=1}^{m} log[1 - (D(G(z^{(i)}))] \tag{2.6}$$

But the GAN method still faces some practical problem:

- Non-convergence. GAN has a good performance on the Nash equilibrium problem, while the gradient descent method only guarantee the Nash equilibrium on a convex problem. When both of the players is represented by neural network, the perfect equilibrium is hard to achieve, making the network continually update themselves, which brings the non-convergence problem.
- Collapse problem[31]. The GAN is a minmax game, so there does not exist a strict loss function. So it is hard to distinguish whether the generator get improved or not. Sometimes the generator will always generate the same point, making the training could not continue, then we call the GAN meets collapse problem.

2.4 Semantic Segmentation

Semantic segmentation is understanding an image at pixel level, for example we want to assign each pixel in the image an object class. For example, in the following picture

Figure 2.2: An example of semantic segmentation

Apart from recognizing the man and the background, we also have to delineate the boundaries of each object. Therefore, unlike classification, we need dense pixel-wise predictions from our models.

Chapter 3

Related work

This chapter reviews recent works in adversarial domain adaptation and also introduces some recent works in deep semantic segmentation.

3.1 Adversarial Domain Adaptation

The proposed of GAN[12] has provided domain adaptation a new direction, adversarial domain adaptation. Actually, Hu et al.'s [15] work indicated that GAN is a special case of adversarial domain adaptation with a degenerated source domain. In the following part, we will show some recent study of adversarial domain adaptation.

3.1.1 Feature-Level Adaptation

DANN

At 2015, [8, 9] proposed a model in feature-leveled domain adaptation by using the adversarial method [12]: Domain-adversarial Neural Network (DANN), which could be viewed as the birth of adversarial domain adaptation. This model contains three parts: a feature extractor used to extract the feature from the input images, a label classifier used to predict the class label and a domain classifier used to classify the domain label. On one hand, the loss from the domain classifier needs to be minimized in order to correctly distinguish the source and target domain. On the other hand, to find the domain invariant feature, the loss function of the domain classifier needs to be maximized.

By formulating such a min-max game, DANN find a common feature space between the source and target.
Then, there appear some variants based on DANN. [36] considered the similarity between the classes, and proposed the soft label loss. [35] used untied weight mapping in feature extractor and reduce the difficult in optimization.

DSN

Finding a mapping function or a function or a domain invariant representation, which in principle, is a good idea. [3] thinks that these methods will be affected by the different distribution in low level. So Domain Separation Network(DSN) was proposed. DSN not only extract the domain invariant feature but also extract the identity feature of the source and target domain, which could be viewed as the low level features. And from the authors point of view, the invariant feature subspace show be orthogonal with the identity subspace. Also this paper adopted MMD[13] method and gradient reversal layer[8, 9] to train the invariant feature extractor.
However, all these approaches are based on feature level, do not enforce any semantic consistency[14].

3.1.2 Pixel-Level Adaptation

Pixel-Level domain adaptation is usually combined with image generation or style transfer. The task of pixel-level adaptation is not only focus on a correct classification but also on generating a picture with a clear semantic meaning. Also, There are several works about the pixel-level adaptation based on GAN in recent years.

PixelDT

Yoo et al.[38] studied the domain transfer problem. They tried to achieve a domain transfer of clothes on semantic level. They build a three levels network. The first level is called convert, which is build by an encoder and a decoder. Encoder firstly extracts the low-level semantic information of a source picture, and the decoder decodes this information into another picture. Actually, the encoder is a convolutional neural network, and we will use the second and third level network to

train the decoder. The second level is a real/fake discriminator and the third level is a domain discriminator. The decoder tried to confuse the real/fake discriminator, making the decoder result look natural and it tried to confuse the domain discriminator, making the decoder result come from the target domain.

PixelDA

Bousmalis et al.[4] did a study about the domain adaptation on the pixel-level. They got inspiration from the style transfer [10, 17] using GAN. Except from using domain loss and task loss like some feature-level domain adaptation[8, 9, 35], they also proposed a content-similarity loss on the pixel-level. Different from the style transfer, this model PixelDA aims to learn the style of the whole source domain. By learning the style of the source on pixel-level, the model could decouple from the task-specific architecture and has a higher training stability.

CoGAN

Coupled Generative Adversarial Networks[22] was used to learn the joint distribution of multi-domain images. The method of CoGAN is simple, by training two GANs at the same time, and there are some layers shared between these GANs. In the generator, the first several layers shared the weight, and in the discriminator the last several layers shared the weight. From this paper's view, we could extract the invariant features, for example the object contour, and by the first several shared layers. And in these private layers, we could get some detail information, such as texture. So the final loss function is the combination of these identical GAN's loss function, and the training method is as same as the DCGAN's [27] training method.

pix2pix

Isola et al.[16] studied the image-translation problem, they build a image-to-image translation fromework(pix2pix) based on conditional adversarial networks[24]. They add a L1 term into loss function, making the generator's task is not only fooling the discriminator but also being near the ground truth output in an L1 sense. The authors thinks

comparing with L2 term, L1 could encourage less blurring. Also different from the traditional encoder-decoder structure, the structure is based on U-net[29], which is an skip encoder-decoder with skip connections between mirrored layers in the encoder and decoder stacks. While in discriminator, we need to concern about both high and low level correctness. So the author proposed a PatchGAN method, by splitting the image into several N×N patches.The final discriminator result is the average of these patches results.

CycleGAN

No matter how impressive the pix2pix structure's result is, this framework is based on cGAN, which means the input images are paired. Zhu et al.[40] used cycle-consistent adversarial network(CycleGAN) to finish an unpaired-image translation on pixel-level. When doing adversarial domain adaptation, usually we can not find the low level corresponding relationship. While CycleGAN adopts another generator, which tried to convert the generated picture back into the source picture and add such cycle consistent loss into the loss function. Similarly, DiscoGAN[18], DualGAN[37] were based on the same method. The only difference is DualGan change traditional GAN into Wasserstein GAN[1] to improve the stability and DiscoGAN chosse L2 for cycle consistency, while DualGan and CycleGAN choose L1 for cycle consistency.

UNIT

Liu et al.[21] combine the VAE[19, 28], coGAN and CycleGAN together and get a network called Unsupervised Image-to-Image Translation Networks(UNIT). Such framework also based on a shared latent space assumption. So UNIT firstly use a VAE to map the images from different source into a same latent space. KL divergence terms is used to penalize deviation of the distribution of the latent code from the prior distribution. Then, similar with what we did in coGAN, for each domain, we used a partly weight shared generator and a totally identical discriminator. To make the generator learn the domain specific feature, we use a cycle-consistent loss function, by comparing the generated pictures and the original pictures. So the final method is using VAE get the latent code and using coGAN to re-generate the

pictures from the latent code. We can just simply change the generate and discriminator to achieve a domain transfer. The similar structure and method are also used in XGAN.[30]

CyCADA

Hoffman et al. [14] make a improvement named cycle-consistent adversarial domain adaptation(CyCADA) based on [40]. They proposed that there are three kind of loss during domain adaptation: pixel loss, feature loss, semantic loss. Also they also adopted cycle consistent like[40] to learn an invertible mapping function. Different from the style-transfer[10, 17], they defined a semantic loss instead of content loss. Content loss is calculated from pixel to pixel, while semantic loss in this paper is represented by the sum of the classification loss on both fake picture and the inverted picture. And the author insist that these two representation are analogous.

StarGAN

While their is still a big problem in CycleGAN and its variants: CycleGAN is only suitable for the two domain problem, when we are facing multi-domains question, we need to build a model for each two domain. Choi et al.[6] find an approach to solve the multi-domains problem: StarGAN. The method behind StarGAN is simple, instead of finding a mapping function between each two domains, StarGAN tried to find a mapping function between each domain and a common domain. Also they use mask vector to build a uniform domain label space. To stabilize the training process and generate higher quality images, the authors chose Wasserstein GAN[1] instead of GAN.

DTN

Domain Transfer Network[34] is another important work in adversarial domain adaptation. The authors think that when GAN reaches a balance, the generator gets knowledge from the source domain, and its generated result is indistinguishable for any discriminator, which means the GAN has learned a invariant representation of the source domain. The author creatively add a new loss called identity loss. When we input the generated image into the generator, identity loss

will estimate the similarity between such fake input and its output, to guarantee that for a fake input, the output tend to be this original fake input. The generator's structure is similar with Yoo's[38] method, by encoding the picture into a low level information and then decode it into another picture.

DRCN

Deep Reconstruction-Classification Networks[11](DRCN) is another network structure used in domain adaptation. DRCN is made by one traditional classification pipeline(an feature extractor and a feature classifier) and another reconstruction pipeline(from extracted feature back to image). DRCN is trained by minimizing the reconstruction error and the classification error on source domain. The author thinks that once we minimize these two errors, the extracted feature would find a domain adaptive representation for both source and target domain. Also to improve the performance of DRCN, the author adopted data augmentation and denoising autoencoders to improve the generalization ability.

FADA

In supervised domain adaptation, usually we will face a problem that there are only few labelled target example, Few-Shot Adversarial Domain Adaptation(FADA)[25] provides us a solution. The main method of this paper is to learn an embedded subspace which maximizes the confusion between source and target domains while semantically aligning their embedding. The author thinks that in unsupervised domain adaptation, especially the domain adaptation using adversarial method, we need to align the distributions, but the distributions alignment can not always guarantee to achieve a semantic alignment, which the author thinks is a major source of performance reduction. So the main innovation of this paper is introducing a domain-class discriminator(DCD), which is a two fully connected layer with a softmax activation in the last layer, used to align the semantic level meanings in source and target domain.

3.2 Deep Semantic Segmentation

Semantic Segmentation is a classical computer vision problem. With the development of deep learning technology, semantic segmentation could achieve a more accurate result. In the following part, we will show some recent study of semantic segmentation.

FCN
Fully connected network(FCN) was proposed by Long et al.[23]. The structure of FCN is based on traditional CNN. By using FCN we could classify each pixel's class. Usually a CNN will add several fully connected layers after the convolution layers. While in FCN, we will use the convolution layers to instead these fully connected layers. And FCN will use the upsampling method to upsample these convolution layers back to the original size. And also skip architecture is used in FCN to optimize the result.

SegNet
SegNet[2] is another network based on FCN, which is used for autopilot. The structure of SegNet is also encoder-decoder structure, the encoder part is a pre-trained VGG-16[33] network. The difference with FCN is the way how the SegNet do the up-sampling process. In SegNet, when doing the pooling process, the network will store the pooling indices and in decoder the up-sampled is based on these pooling indices. While in FCN, the up-sampling method is tried to find a deconvolution function and sum the up-sampling result with the encoded feature map.

U-Net
However, the FCN is based on millions of training images, which is hard to achieve in a biomedical task. So here comes U-net[29]. The structure of U-net is similar with FCN, but different from FCN, U-Net does not use pre-trained CNN model[33], because U-net is usually used in binary classification. And in lower level feature fusion part, U-net concat the original feature with the up-sampled feature. Also the author mentioned that they also use data augmentation to solve the limited training data problem.

DeepLab

DeepLab[5] is a popular semantic segmentation framework, which made some improvements based on FCN. First of all, while doing up-sampling in FCN, the existence of stride will bring the decrease of the resolution in the result. So in deeplab framework, we will set stride to one and also adopted dilated convolutions[39] strategy, to avoid the shrink of the result size. Secondly, the FCN is basically a encode-decode process, which will absolutely bring the loss of the information. So deeplab used a fully connected conditional random field[20] together with a bilinear interpolation to refine the segmentation result.

Chapter 4

Methods

Bibliography

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein gan". In: *arXiv preprint arXiv:1701.07875* (2017).

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 39.12 (2017), pp. 2481–2495.

[3] Konstantinos Bousmalis et al. "Domain separation networks". In: *Advances in Neural Information Processing Systems*. 2016, pp. 343–351.

[4] Konstantinos Bousmalis et al. "Unsupervised pixel-level domain adaptation with generative adversarial networks". In: *arXiv preprint arXiv:1612.05424* (2016).

[5] Liang-Chieh Chen et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs". In: *arXiv preprint arXiv:1606.00915* (2016).

[6] Yunjey Choi et al. "StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation". In: *arXiv preprint arXiv:1711.09020* (2017).

[7] Carl Doersch. "Tutorial on variational autoencoders". In: *arXiv preprint arXiv:1606.05908* (2016).

[8] Yaroslav Ganin and Victor Lempitsky. "Unsupervised domain adaptation by backpropagation". In: *International Conference on Machine Learning*. 2015, pp. 1180–1189.

[9] Yaroslav Ganin et al. "Domain-adversarial training of neural networks". In: *Journal of Machine Learning Research* 17.59 (2016), pp. 1–35.

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 2414–2423.

[11] Muhammad Ghifary et al. "Deep reconstruction-classification networks for unsupervised domain adaptation". In: *European Conference on Computer Vision*. Springer. 2016, pp. 597–613.

[12] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[13] Arthur Gretton et al. "A kernel two-sample test". In: *Journal of Machine Learning Research* 13.Mar (2012), pp. 723–773.

[14] Judy Hoffman et al. "CyCADA: Cycle-Consistent Adversarial Domain Adaptation". In: *arXiv preprint arXiv:1711.03213* (2017).

[15] Zhiting Hu et al. "On Unifying Deep Generative Models". In: *arXiv preprint arXiv:1706.00550* (2017).

[16] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: *arXiv preprint arXiv:1611.07004* (2016).

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution". In: *European Conference on Computer Vision*. Springer. 2016, pp. 694–711.

[18] Taeksoo Kim et al. "Learning to discover cross-domain relations with generative adversarial networks". In: *arXiv preprint arXiv:1703.05192* (2017).

[19] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[20] Philipp Krähenbühl and Vladlen Koltun. "Efficient inference in fully connected crfs with gaussian edge potentials". In: *Advances in neural information processing systems*. 2011, pp. 109–117.

[21] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. "Unsupervised Image-to-Image Translation Networks". In: *arXiv preprint arXiv:1703.00848* (2017).

[22] *Coupled generative adversarial networks*. 2016, pp. 469–477.

[23] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3431–3440.

[24] Mehdi Mirza and Simon Osindero. "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784* (2014).

[25] Saeid Motiian et al. "Few-Shot Adversarial Domain Adaptation". In: *Advances in Neural Information Processing Systems*. 2017, pp. 6673–6683.

[26] Sinno Jialin Pan and Qiang Yang. "A survey on transfer learning". In: *IEEE Transactions on knowledge and data engineering* 22.10 (2010), pp. 1345–1359.

[27] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434* (2015).

[28] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and variational inference in deep latent Gaussian models". In: *International Conference on Machine Learning*. 2014.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2015, pp. 234–241.

[30] Amélie Royer et al. "XGAN: Unsupervised Image-to-Image Translation for many-to-many Mappings". In: *arXiv preprint arXiv:1711.05139* (2017).

[31] Tim Salimans et al. "Improved techniques for training gans". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2234–2242.

[32] Hidetoshi Shimodaira. "Improving predictive inference under covariate shift by weighting the log-likelihood function". In: *Journal of statistical planning and inference* 90.2 (2000), pp. 227–244.

[33] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[34] Yaniv Taigman, Adam Polyak, and Lior Wolf. "Unsupervised cross-domain image generation". In: *arXiv preprint arXiv:1611.02200* (2016).

[35] Eric Tzeng et al. "Adversarial discriminative domain adaptation". In: *arXiv preprint arXiv:1702.05464* (2017).

[36] Eric Tzeng et al. “Simultaneous deep transfer across domains and tasks”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 4068–4076.

[37] Zili Yi, Hao Zhang, Ping Tan Gong, et al. “DualGAN: Unsupervised Dual Learning for Image-to-Image Translation”. In: *arXiv preprint arXiv:1704.02510* (2017).

[38] Donggeun Yoo et al. “Pixel-level domain transfer”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 517–532.

[39] Fisher Yu and Vladlen Koltun. “Multi-scale context aggregation by dilated convolutions”. In: *arXiv preprint arXiv:1511.07122* (2015).

[40] Jun-Yan Zhu et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *arXiv preprint arXiv:1703.10593* (2017).

Appendix A

Unnecessary Appended Material