Source Sample Selection Strategy on Unsupervised Domain Adaptation and Segmentation Task

M.TECH PROJECT

Submitted in fulfillment of the requirements of
CSP 798 Project

By

Mr. Sujit Rai
ID No. 2017CSM1006

Under the supervision of:

Dr. Naraynan C Krishnan

INDIAN INSTITUTES OF TECHNOLOGY ROPAR, PUNJAB
July 2019

Declaration of Authorship

I, Mr. Sujit Rai, declare that this M.TECH PROJECT titled, 'Source Sample Selection Strategy on Unsupervised Domain Adaptation and Segmentation Task' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Master degree Project at this University.
- Where any part of this project/thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this project/thesis is work of mine and my supervisor.
- I have acknowledged all main sources of help.
- Where the Project/thesis is based on work done by myself jointly with supervisor, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

Certificate

This is to certify that the Project/thesis entitled, "*Source Sample Selection Strategy on Unsupervised Domain Adaptation and Segmentation Task*" and submitted by Mr. Sujit Rai ID No. 2017CSM1006 in fulfillment of the requirements of CSP 798 Project embodies the work done by him under my supervision.

Supervisor
Dr. Naraynan C Krishnan
Asst. Professor,
IIT Ropar, PUNJAB
Date:

INDIAN INSTITUTES OF TECHNOLOGY ROPAR, PUNJAB

Abstract

Master of Technology

Source Sample Selection Strategy on Unsupervised Domain Adaptation and Segmentation Task

by Mr. Sujit Rai

We propose a novel approach of reducing the shift between target and source domain using the source sample selection strategy. The sample selection strategy can be incorporated with any deep learning architecture consisting of a adversarial training for reducing the domain shift and a supervised task specific loss. Source sample selection works by assigning more importance to those source instances which consists of similar features as that of target instances. This similarity is quantified by the output of discriminator and importance is incorporated by assigning more weight to the task specific loss term of corresponding instance. We investigate our proposed sample selection strategy with both quantitative and qualitative experiments by incorporating it with various existing model available for domain adaptation and segmentation tasks.

KEYWORDS

Sample Selection, Domain Adaptation, Segmentation

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent advances in deep neural networks have produced state the of the art results in various machine learning tasks. To a considerable extent, the success of the neural network is appreciated by the the amount of labeled data available. One may note that the labeled data is not always available in enormous amount because it is a very expensive procedure. However, there are some cases when the labeled (source)data is available, but this data suffers from a shift when compared to actual target data distribution. This labeled data can be obtained synthetically via some automatic procedure or a program. Therefore, to infer the properties in target domain, we define the task of transferring the knowledge from the source domain (labeled data) to the target domain (unlabelled data) which is formally known as domain adaptation. One such use case of domain adaptation is to obtain a classifier on the images from street view house numbers dataset using hand-written digits images from MNIST dataset. Here MNIST dataset and SVHN dataset are of similar nature and instances in MNIST dataset is labelled whereas the instances of SVHN dataset are unlabelled.

Depending on the type of target domain, domain adaptation can be categorized as supervised and unsupervised. For supervised domain adaptation, the target domain consists of labeled data and in contrast, unsupervised domain adaptation considers only the data points in target domain. This work proposes a technique for unsupervised domain adaptation. Unsupervised domain adaptation can be further classified depending on the type of classes present in the source and target domains. Partial domain adaptation is defined when there may exist a partial overlap between the labels of the source and target domains, i.e., the source and target domain might consist of few or no common labels. In contrast to partial domain adaptation, the complete domain adaptation assumes a complete overlap in labels between the source and target domains.

In the problem setting discussed in the report, we assume that the inter-domain labels are completely overlapping and there exist a classifier such that it performs well on both the domains. Therefore in this work the focus is on solving the task of unsupervised and conservative domain

adaptation. The approaches which are extensively used in domain adaptation literature are Adversarial Methods [23] [22] [5] [11], Network Methods [6] [17], Optimal Transport [7].

The existing approaches mainly focus on learning a transformation function that would align the source and target domains by extracting common task specific features thus reducing the difference between the marginal probability distribution of both domains. However considering the domain shift, irrelevant source samples and high-dimensionality training settings of this approaches, the adaptation procedures are not capable of completely aligning the marginals and therefore, the KL-Divergence computed between the transformed source and target instances is greater than 0. Therefore, the problem is to sample source instances which are more relevant for the task of domain adaptation. This sampling strategy will result in better alignment by discarding the irrelevant samples.

The proposed method incorporates the source sample selection strategy on the existing adversarial methods for unsupervised domain adaptation. The main contributions of the proposed work are as follows

1. We propose a novel method using sample selection strategy to assign more weightage on source samples which are similar to the target instances.
2. We generalize this approach by incorporating it with models available for segmentation task
3. Both quantitative and qualitative experiments were performed to demonstrate the improvement in performance by incorporation of the proposed strategy.

The report is structured as follows. We start with first introducing the related work in chapter 2 which is followed by our proposed methodology in chapter 3. We discuss the experiments in chapter 4 where we show both our qualitative and quantitative results and then end with a conclusion and future work in chapter 5 and 6 respectively.

Chapter 2

Related Work

In this section, previous work for designing the deep architecture for domain adaptation and segmentation are discussed. The theory behind domain adaptation was proposed by ben-david etal [1] and ben-david etal [2]. ben-david etal [2] proposed that expected error on the target domain is bounded by the sum of expected error in the source domain, the discrepancy between two domains and the shared expected error of a single classifier on both source and target domain.

$$\forall h \in H, R_T(h) \leq R_S(h) + \frac{1}{2} d_H(S,T) + \lambda \tag{2.1}$$

Here, $\frac{1}{2} d_H(S,T)$ is the H-distance. In case, of conservative domain adaptation, the shared error λ is negligible as there exists a single classifier that can classify both source and target domains. While in case of non-conservative domain adaptation λ term is high and cannot be neglected.

Most of the work focus on reducing the discrepancy between source and target by minimizing the covariate shift between the two distributions using moment maching such as MMD, CMD [21] [15] [16] [14], adversarial loss [20] [8] [3] or batch normalization statistics [12]. One of the disadvantage of this approach is the requirement of the use of kernel that can be considered as a type of hyperparameter, As deep architectures are known to extract underlying features automatically using gradient descent approach, it is preferred to avoid the use of hand crafted features as much as possible. other approaches for domain adaptation focus on using adversarial loss in order to reduce the domain discrepancy.

One of the earliest work that used adversarial training for domain adaptation is Unsupervised domain adaptation using backpropagation [8]. In this work ganin etal used a single feature extractor which would extract discriminative and domain invariant features from the input. The output of this feature extractor was then passed to the class classifier. A domain discriminator was used in order to force the feature extractor to extract domain invariant features. The job of domain discriminator was to predict the domain of the output of the feature extractor, while the job of feature extractor was to fool the domain discriminator and minimize the classification loss.

The intuition behind this paper formed the basis of lot of work on domain adaptation that used adversarial loss.

Tzeng etal [20] propose the Adversarial Discriminative Domain Adaptation framework using untied weights among source and target feature extractors, discriminative modeling and a GAN loss. Untying the weights leads to an independent source and target mapping thus providing more flexibility for learning domain specific features. The target feature extractor was pre-initialized by the learned source feature extractor as without proper initialization and training procedures the target feature extractor might learn a degenerate solution.

Shu etal [18], propose the Dirt-T approach for Domain Adaptation methodology which can be used with any existing model for domain adaptation. This work focused on overcoming two main issues present in most of the adversarial approaches for domain adaptation. 1.) If the feature extractor function has a very high capacity and if the source-target supports are disjoint then enforcing domain-invariance in the feature space is a very weak constraint. 2.) If the task of domain adaptation is non-conservative then there does not exist a single classifier that will perform better in case of both source and target domains. The critical component of this paper is the cluster assumption, which states that decision boundaries should not cross high-density regions. Conditional cross entropy was used for maintaining cluster assumption. Minimization of conditional entropy forces the classifier to be confident on the unlabeled target data. Inorder to enforce locally lipschitz constraint virtual adversarial training was used. The paper further extended this approach by proposing decision boundary iterative refinement training approach (DIRT-T) for non conservative domain adaptation. This approach first creates a decision boundary in the source domain using the VADA approach then iteratively refines the decision boundary on target domain by taking a seemingly small step that minimizes the conditional entropy subject to the constraint that the KL Divergence between previous and current predictions is small.

Partial domain adaptation is a scenario of domain adaptation in which label space of target domain is subset of label space of source domain. Zhang etal [24] propose Importance Weighted Adversarial Nets for Partial Domain Adaptation, an extension of Partial Transfer Learning with Selective Adversarial Networks [4], that uses k discriminators one for each of the k classes. The use of a single discriminator for every class is impractical in the presence of a large number of classes. The extension overcame the need for k discriminators by using only 2 discriminators. As the task was to perform partial domain adaptation, it was not possible to directly match the marginal distribution since the conditional distribution of labels were different. Therefore this approach matched the marginals of only the instances that belonged to the labels also present in target domain. In-order to identify the instances that belonged to the common class as that of target domain, a domain discriminator was used. The basic intuition behind this approach is that if an instance does not belong to the common class then the discriminator will be able to easily predict the domain of the instance. Thus if domain classifier is very confident for a instance then it means that the instance does not belong to the common class. This domain

classifier was then used to provide a lower weight to the instances which were not present in the common class and higher weight to the instances which were present in the common class. The resulting weighted instances were then used for training the feature extractor and another domain discriminator in order to align the weighted marginal distributions.

Hsu etal [10] proposed Learning To Cluster In Order To Transfer Across Domains And Tasks. This work is very different from the conventional approach towards domain adaptation. This approach can be used for domain adaptation tasks as well as for transferring knowledge across tasks. The approach is to transform the problem into problem of finding the pairwise similarity between instances of a particular domain and then learn a transferable similarity function that can be used for clustering in both domains. After similarity function is learned it can be used to form the clusters in both the target and source domains and then the class labels will be assigned to a cluster on basis of the majority of instances from a particular label present in the cluster of a particular domain.

Hoffman etal [9] proposed Cycle-Consistent Adversarial Domain Adaptation. Most of the previous work in domain adaptation focused only on aligning marginal distributions and it was assumed that the conditional distribution of labels will get aligned automatically however this assumption does not hold good in all cases. Also prior works did not enforce any constraint on maintaining semantic information while translating from source to target. Therefore this work focused on overcoming this issue. They used image to image translation in order translate an instance from source domain to target domain using cycle consistency. Thus maintaining one to one mapping and aligning marginals in the pixel space. Inorder to align conditional distribution of labels, Source classifier was used as a noisy labeller to ensure that the translated image in the target domain gets classified in the same way as that of the original image in the source domain. This work used adversarial loss for aligning marginals in the feature space as well as in the pixel space and source classifier as noisy labeller for aligning the conditional distributions. Using source classifier as the noisy labeller inorder to maintain the semantic consistency and aligning the conditional distributions is a weak constraint. Also the image to image translation performed in this method is a one to one translation while in many cases every single image from the source domain can get mapped to many different images in the target domain.

Most of the previous mentioned work focused only on aligning the marginal distributions while it was assumed that the conditional distribution will get aligned automatically once marginals are aligned. Also, there is little literature available that performs sample selection on the source samples in-order to select only those source sample that are relevant in aligning the marginals. This source sampling is done by assigning weights to the loss functions of the source samples.

Class importance weighting is one of the approaches of sample selection wherein every class is assigned a particular weight in-order to overcome class-imbalance problem. The conditional probability distributions are assumed to be equivalent once the marginals are matched in unsupervised domain adaptation. Exploiting this concept, the equation for target loss can be

written as follows

$$L(f) = \sum_{y \in Y} \int_X l(f(x), y) \frac{P_T(x|y)P_T(y)}{P_S(x|y)P_S(y)} P_S(x, y)dx \tag{2.2}$$

Since, the conditionals are equivalent therefore they can be cancelled.

$$L(f) = \sum_{y \in Y} \int_X l(f(x), y) \frac{P_T(y)}{P_S(y)} P_S(x, y)dx \tag{2.3}$$

Here, the class weights $w(y)$ is $\frac{P_T(y)}{P_S(y)}$ and is useful for correcting class imbalance problem. Similar results can be obtained by sampling more or less number of instances from a particular class. Lipton etal [13] proposed Black Box Shift Estimation. In this approach the inverse of the confusion matrix is computed on the source validation set and multiplied with the predicted target prior in order to obtain the class weights.

For the task of segmentation, We will focus specifically on retinal vessel segmentation. The retianl vessel segmentation task can be considered as an image translation task wherein an input image is translated to an output segmentation mask. The performance achieved by CNNs in retinal segmentation task has already surpassed the performance achieved by human experts in several studies. However, most of the earlier CNN based architecture produced blurry outputs and false positives because of the pixel-wise objective functions. Retinal vessel segmentation in fundoscopic images using generative adversarial networks [19] proposed a GAN framework in-order to overcome this issues. The intuition was to constrain the network to generate images that resemble the annotations produced by the human experts inorder to obtain sharp and clear output. Adversarial training forces the generator to generate outputs similar to gold standard.

Chapter 3

Proposed Work

3.1 Unsupervised Domain Adaptation

Notations : We use $P_s(x)$ and $P_t(x)$ for denoting the source and target distribution in some $\mathcal{X}$ space respectively. We use the notations $\mathbf{x}_s$ and $\mathbf{x}_t$ for denoting a data point in source and target domains respectively. Similarly, $\mathbf{y}_s$ and $\mathbf{y}_t$ are used for denoting the corresponding class labels. On input $\mathbf{x}$, we represent the output of the Feature Extractor and the discriminator networks as $\mathbf{z} = F(\mathbf{x})$ and $d = D(\mathbf{x})$ respectively. The output $d \in [0, 1]$ denotes the probability of $\mathbf{x}$ belonging to the target distribution ($d = 1$). Similarly on input $\mathbf{z}$, the output of classifier network is denoted by $c = C(\mathbf{x})$.

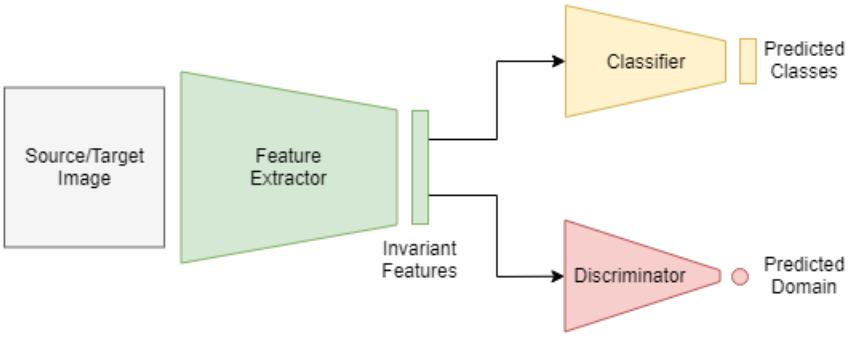

FIGURE 3.1: Adversarial Unsupervised Domain Adaptation Architecture.

The proposed approach of source sample selection is incorporated with the existing domain adaptation models based on adversarial methods. These adversarial methods mainly comprise of a Feature extractor, Domain Discriminator and a Classifier. The task of domain discriminator is to classify whether the input instance is from source or target domain. Similarly, the task of the classifier is to classify the class of the input instance. Input to both the Domain Discriminator

and Classifier is the output of the Feature Extractor. The task of feature extractor is to extract domain invariant and class specific features from the input instance. This feature extraction process is achieved by training the feature extractor to maximize the discriminator's loss L_D and minimize the classifier's loss L_C, whereas both the discriminator and classifier are trained to minimize the domain discrimination and classification losses respectively. The feature extractor may be shared among both the source and target domain or it can be separate. The corresponding loss functions for all the components are as follows

$$L_D = - \mathop{\mathbb{E}}_{x \in P_t(x)} \log(D(F(x)) - \mathop{\mathbb{E}}_{x \in P_s(x)} \log(1 - D(F(x)) \tag{3.1}$$

$$L_F = - \mathop{\mathbb{E}}_{x \in P_t(x)} \log(1 - D(F(x)) \tag{3.2}$$

$$L_C = - \mathop{\mathbb{E}}_{x \in P_s(x)} y_s \log C(F(x)) \tag{3.3}$$

Where, L_F is the loss functions for feature extractor. Therefore, the total objective function becomes.

$$\min_{F,C} L_C + \min_{F} L_F + \min_{D} L_D \tag{3.4}$$

The intuition behind the proposed sample selection strategy is to assign more importance to those source instances which consists of similar features as that of target instances. This similarity can be quantified in terms of the output of the discriminator. As, an optimal feature extractor will be able to extract domain invariant features from the source instances and fool the discriminator. Therefore, the incorrectness of discriminator for a particular source instance is a quantification of the similarity of source instance to the target domain. In the proposed approach the output of discriminator is thus used for assigning weights to corresponding classification loss term of the source instances. The weights are obtained as follows,

$$W = Q_s(x) = \frac{D(z_s)}{\sum D(z_s)} \tag{3.5}$$

Here, The weights are computed for all the instances of a batch and then normalized so that the weights sum up to 1 for a batch. Thus, forming a new weighted probability distribution Q_s, instances sampled according to this new distribution Q_s can be considered as being more closer to the target distribution. The classification loss for a particular instance is multiplied by its corresponding weight thus constraining the model to focus more on instances with high weights.

Therefore, the corresponding classification loss can be denoted as follows,

$$L_C^{'} = - \mathop{\mathbb{E}}_{x \in Q_s(x)} y_s \log C(F(x)) \tag{3.6}$$

An additional cross entropy term for source domain is added to the loss of discriminator.

$$L_D^{'} = - \mathop{\mathbb{E}}_{x \in P_t(x)} \log(D(F(x)) - \mathop{\mathbb{E}}_{x \in P_s(x)} \log(1 - D(F(x)) - \mathop{\mathbb{E}}_{x \in P_s(x)} D(F(x)) \log(D(F(x)) \tag{3.7}$$

This cross entropy term along with the binary cross entropy loss forces the discriminator to focus more on those source instances which are hard to classify and less on those source instances which are easy to classify thus giving the feature extractor an opportunity to generate instances similar to those source samples, which discriminator is able to classify easily. This helps in reducing the domain shift gradually. Thus the complete objective function after incorporating sample selection strategy is as follows

$$\min_{F,C} L_C^{'} + \min_{F} L_F + \min_{D} L_D^{'} \tag{3.8}$$

3.2 Segmentation

Various deep learning models have been proposed for performing segmentation on image. The goal of such models is to label each pixel of an image to the corresponding class depending on the semantics present in the image. As, the model is constrained to predict for every pixel present in the image, the segmentation task is known as a dense prediction task. Most of the early approaches proposed for segmentation task used pixel-level objective functions to compare the target segmented image with the generated images. These approaches leads to blurry results due to averaging of the loss function across all the target images. Existing approaches incorporated the use of a discriminator network to tackle this problem along with the previous pixel-level objective functions. The proposed work is focused on this type of architectures involving a task specific loss and an adversarial loss.

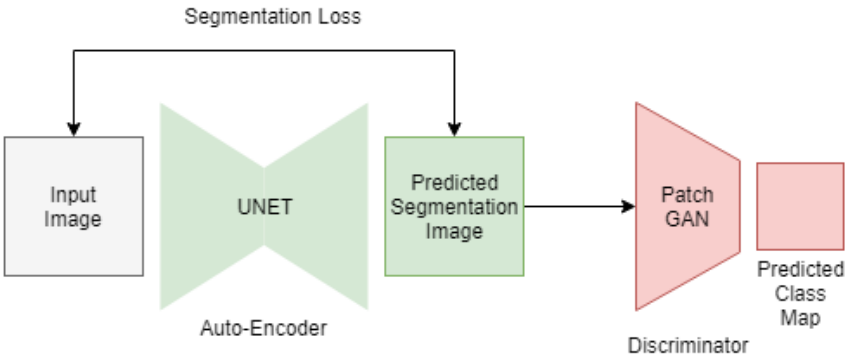

FIGURE 3.2: Architecture for segmentation task using adversarial approach and supervised segmentation loss.

These methods mainly comprise of a generator network mostly an UNET network and a discriminator network. The task of generator network is to predict the class of every pixel in the input image. While task of discriminator is to consider a patch of the generated image and classify whether its a patch from target image or generated image. Both the generator and discriminator are trained adversarialy.

Notations : We use G and D for denoting the generator network and Discriminator Network. The generator network maps the input image x to the corresponding segmentated image y or $G : x \rightarrow y$ and the Discriminator network maps a pair of x, y to binary classification $\{0, 1\}^N$ where 0 and 1 mean that y is an generated image or target image and N is the number of patches present in an image.

The generator is trained to minimize the segmentation loss which in our case is the binary cross entropy loss between target image and generated image.

$$L_{SEG}(G) = \mathop{\mathbb{E}}_{x,y \in P_{data}(x,y)} -y \log G(x) - (1-y) \log(1 - G(x)) \tag{3.9}$$

Apart from the pixel level segmentation loss, The generator is also constrained with the following adversarial loss.

$$L_G = - \mathop{\mathbb{E}}_{x \in P_{data}(x)} \log D(x, G(x)) \tag{3.10}$$

Similarly, The adversarial loss function for training the discriminator network is as follows.

$$L_D = - \mathop{\mathbb{E}}_{x,y \in P_{data}(x,y)} \log D(x, y) - \mathop{\mathbb{E}}_{x \in P_{data}(x)} \log(1 - D(x, G(x))) \tag{3.11}$$

Therefore, the total objective function is as follows.

$$\min_G L_{SEG} + \min_G L_G + \min_D L_D \tag{3.12}$$

The discriminator network present in the above existing approach considers each patch as an instance and assigns a probabilistic value to it. Therefore we applied our proposed sample selection startegy considering each patch as a sample. The discriminator network was trained to classify the patches present in an image thus forcing the generator to generate realistic images and the task of segmentation loss to force the generator to generate images similar to the ground truth. Therefore, the intuition behind our sample selection strategy was to assign more importance to segmentation loss of those generated samples(patches) which were being

easily classified by the discriminator so that the generator is forced to focus on these important samples. Thus, the output of discriminator is used as weights for assigning importance to the samples. The weights are obtained as follows,

$$W = Q(x, y) = \frac{D(x, G(x))}{\sum D(x, G(x))} \tag{3.13}$$

Therefore, the corresponding segmentation loss can be denoted as follows,

$$L^{'}_{SEG}(G) = \mathop{\mathbb{E}}_{x,y \in Q(x,y)} -y \log G(x) - (1-y) \log(1 - G(x)) \tag{3.14}$$

Similarly, the corresponding adversarial loss functions for discriminator network after incorporation of sample selection is as follows,

$$L^{'}_{D} = - \mathop{\mathbb{E}}_{x,y \in P_{data}(x,y)} \log D(x, y) - \mathop{\mathbb{E}}_{x \in P_{data}(x)} \log(1 - D(x, G(x))) - \mathop{\mathbb{E}}_{x \in P_{data}(x)} D(x, G(x)) \log(1 - D(x, G(x))) \tag{3.15}$$

The complete objective function after is as follows,

$$\min_{G} L^{'}_{SEG} + \min_{G} L_G + \min_{D} L^{'}_{D} \tag{3.16}$$

Chapter 4

Experiments

4.1 Unsupervised Domain Adaptation

In this section, we evaluate our proposed methodology by comparing the improvement in accuracy achieved after incorporating sample selection strategy in various existing approaches for unsupervised domain adaptation. We first introduce the datasets used in experiments and finally the implementation specifications.

4.1.1 Datasets

The datasets used for our experiments are MNIST, SVHN and USPS. All the three datasets consist of handwritten digits images with different styles. Each image is associated with one of the 10 classes (0-9).

MNIST : A dataset consisting of binary images of handwritten digits. The dimensions of images are 28×28. It consists of 50000 training images, 10000 validation images and 10000 test images.

FIGURE 4.1: Images from MNIST Dataset.

SVHN : A dataset consisting of real-world images of digits. The dimensions of images are $32 \times 32 \times 3$. It consists of 600000 images with 73257 images for training, 26032 images for testing and 531131 less difficult extra images.

FIGURE 4.2: Images from SVHN Dataset.

USPS : A dataset consisting of binary hand-written images of digits. The dimensions of images are 16×16. It consists of 9298 images with 7291 images for training and 2007 images for testing.

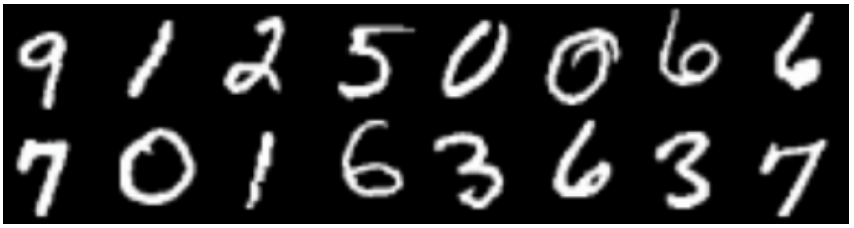

FIGURE 4.3: Images from USPS Dataset.

4.1.2 Implementation Specification

Parameters	DANN	ADDA	VADA, DIRT-T
Batch Size	64	64	100
Epochs	300	50	80
Learning Rate	0.01	1e-3	1e-3
Optimizer	Momentum SGD	Adam Optimizer	Adam Optimizer
Feature Extractor	2 × (Convolution + Max Pool)	3 × (Convolution + Max Pool) + 1 Fully Connected	3 × (3 × Convolution + 1 Max Pool)
Discriminator	2 × Fully Connected	3 × Fully Connected	2 × Fully Connected
Classifier	1 × Fully Connected	1 × Fully Connected	Argmax
Activation Function	Relu	Leaky Relu	Leaky Relu

TABLE 4.1: Implementation Specification.

All the above mentioned datasets are quite different with respect to each other therefore we considered 3 settings as MNIST(source) to SVHN(target), SVHN(source) to MNIST(target) and MNIST(source) to USPS(target) for evaluating proposed approach. DANN, ADDA, VADA and DIRT-T were the baseline architectures considered. All the baseline architectures had a classifier, feature extractor and discriminator with similar implementation settings mentioned by in the respective papers. Firstly, all the baseline architectures were trained and evaluated

on the above mentioned settings. Similarly, all the baseline architectures were then evaluated after incorporating the sample selection strategy. The evaluation metric used for measuring the performance of architectures was accuracy obtained on target test dataset by the classifier.

KL-divergence is a measure of dissimilarity between two probability distributions. It is non negative and asymmetric thus can be interpreted as measuring how accurately one probability distributions approximately another reference probability distribution. Therefore, KL-divergence was used as measure of dissimilarity between transformed probability distributions of source and target domain.

4.2 Segmentation

In this section, we evaluate our proposed methodology by comparing the improvement in accuracy achieved after incorporating sample selection strategy in an existing approach for retinal vessel segmentation. We first introduce the dataset used in experiments and finally the implementation specifications.

4.2.1 Datasets

The datasets used for our experiments are DRIVE and STARE dataset. These are publicly available dataset and consists of 20 annotated images with 10 images for training and 10 images for testing in STARE dataset. Similarly DRIVE dataset consists of 40 annotated images with 22 images for training and 18 images for testing. The dimensions of the images are 720×720 for STARE and 640×640 for DRIVE.

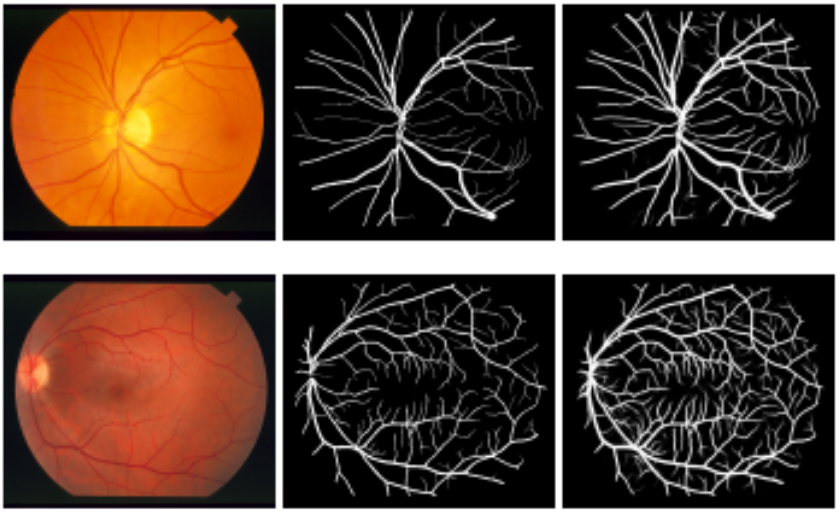

FIGURE 4.4: Images from STARE Dataset. Leftmost image is the fundoscopic image, center image is the gold standard probability and right most is the probability map from best existing approach.

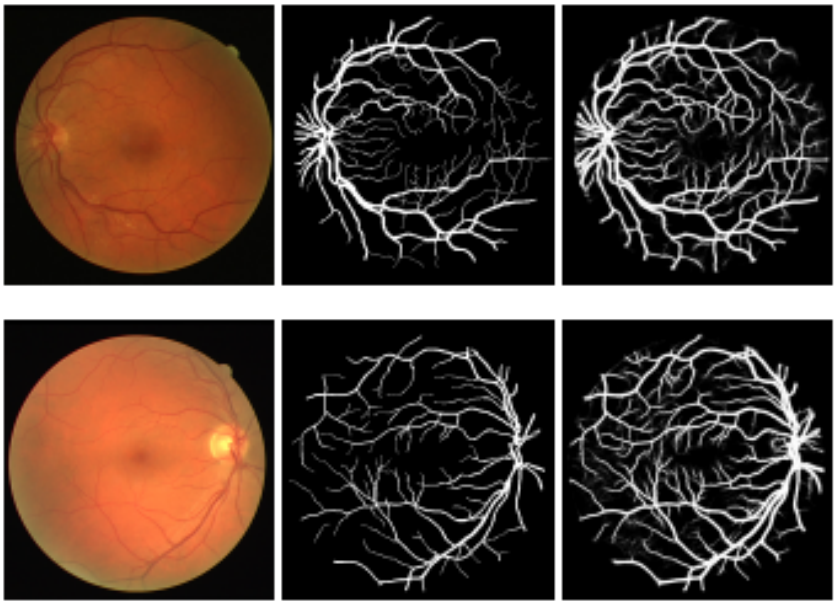

FIGURE 4.5: Images from DRIVE Dataset. Leftmost image is the fundoscopic image, center image is the gold standard probability and right most is the probability map from best existing approach.

4.2.2 Implementation Specification

The number of images for training in STARE and DRIVE dataset is very less, Therefore, various types of augmentations such as left-right flip and rotations with 3-degree interval were applied, resulting in 240 images of STARE dataset, which was further split into training and validation with ratio of 9:1. Before augmenting the dataset, each image was normalized with z-score for each channel.

The generator is an U-NET architecture with 5 convolution layers in the encoder and the decoder consists of 5 upsampling (2×2) layer each followed by a convolutional layer. The discriminator follows a patch GAN architecture with 3 convolution layers resulting in generation of probability maps of dimentions $80 \times 80 \times 3$. Each value in the probability map describes the authencity of 8×8 patch in the generated image. The batch size was set to 1 because of memory constraints. For evaluating the proposed approach, Area under curve for precision and recall curve, Area under curve for receiver operating characteristics and dice coefficient were used as a metric for quantifying the performance and comparison with the baseline architecture. For computing the dice coefficient, the probability maps were thresholded by Otsu threshold.

Chapter 5

Results

5.1 Unsupervised Domain Adaptation

The proposed approach was evaluated using the accuracy obtained by the classifier on the target dataset. Table 5.1 displays the accuracy obtained on the target dataset by baseline architecture and same baseline architectures equipped with sample selection strategy.

Approaches	SVHN → MNIST	SVHN → MNIST (Sampling)	MNIST → SVHN	MNIST → SVHN (Sampling)	MNIST → USPS	MNIST → USPS (Sampling)
DANN	76.3	78.2	12.4	13.7	60.8	44.5
ADDA	76	81	60.4	63.1	89.4	92.6
VADA	94.5	97.1	73.3	74.8	90.6	95.7
DIRT-T	99.4	99.5	76.5	76.9	NA	NA

TABLE 5.1: Accuracy obtained by the source classifier on target dataset. Rows represent the various baseline architectures, columns represent the different training settings.

The proposed approach led to an improvement in accuracy across almost all the training settings and the baseline architectures. Further experimentation was conducted in order to check the KL-divergence between the features generated for source and target dataset.Table 5.2 displays the KL-Divergence obtained between source and target features for the baseline architectures as well as the architectures equipped with proposed methodology.

Approaches	SVHN → MNIST	SVHN → MNIST (Sampling)	MNIST → SVHN	MNIST → SVHN (Sampling)	MNIST → USPS	MNIST → USPS (Sampling)
DANN	0.33	0.19	0.54	0.21	0.371	0.28
ADDA	14.019	11.35	11.39	9.47	5.489	4.83
VADA	0.3	0.27	0.612	0.47	0.359	0.313
DIRT-T	0.36	0.338	0.487	0.461	NA	NA

TABLE 5.2: KL-Divergence between features of source and target dataset. Rows represent the various baseline architectures, columns represent the different training settings.

In all the case the KL-Divergence was reduced compared to the baseline architecture. Thus, the proposed approach performed better by reducing the domain shift as well as increasing the accuracy.

T-SNE visualizations were performed on the output of the feature extractor for source and target dataset from ADDA(SVHN-MNIST).Figure 5.1 and 5.2 represent the T-SNE visualization of the features obtained from baseline architecture and same architecture equipped with sampling strategy.

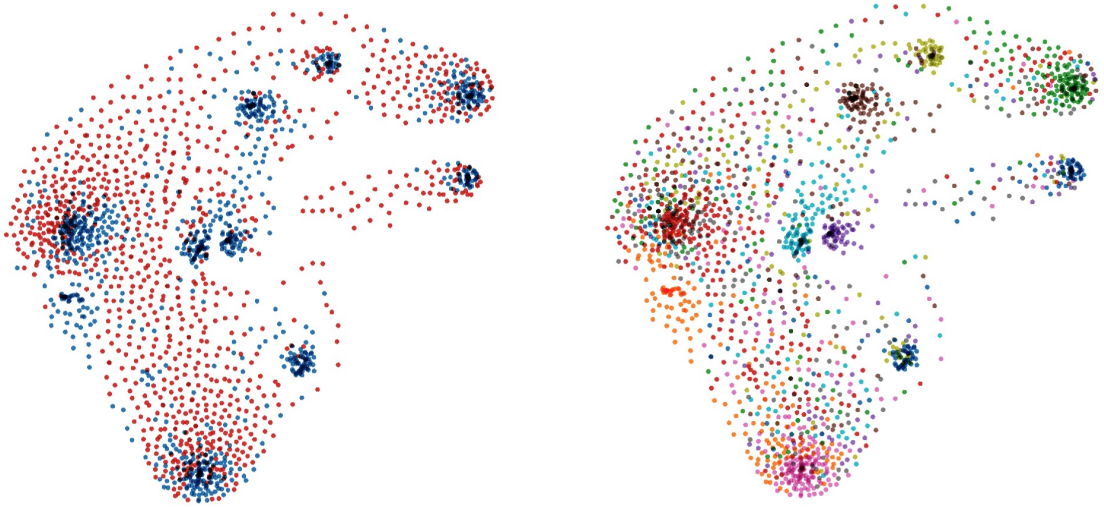

FIGURE 5.1: T-SNE representation of the feature vectors obtained from ADDA(SVHN-MNIST). [Left Image] red color denotes features belonging to target dataset and blue color denotes features belonging to source dataset.[Right Image] This consists of 10 colors denoting the class label of the corresponding feature vector

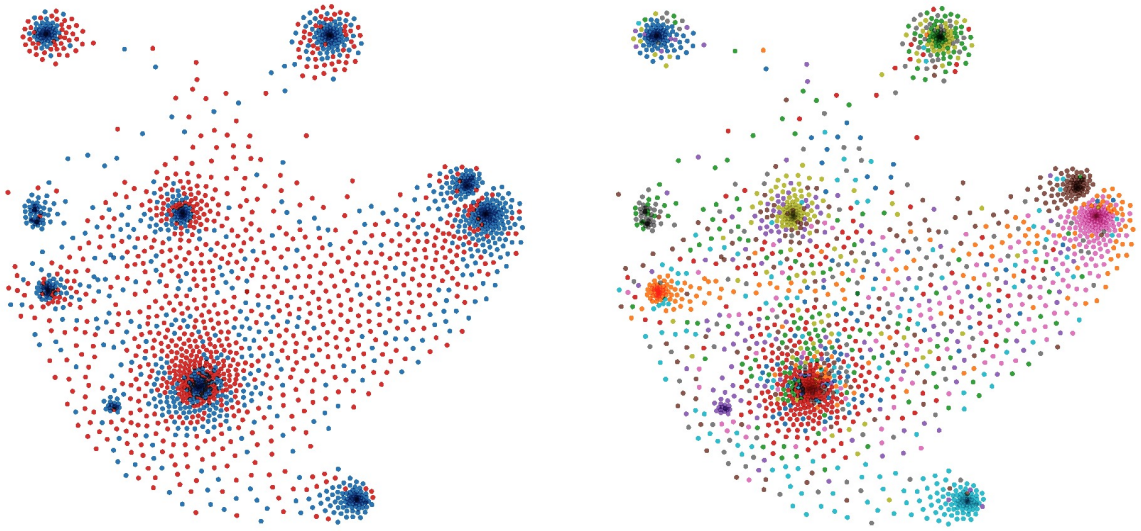

FIGURE 5.2: T-SNE representation of the feature vectors obtained from ADDA(SVHN-MNIST) with sample selection strategy.[Left Image] red color denotes features belonging to target dataset and blue color denotes features belonging to source dataset.[Right Image] This consists of 10 colors denoting the class label of the corresponding feature vector

It is evident from Figure 5.1 and 5.2 that both the approaches were successful in reducing the domain shift resulting in the marginals being matched. However, the conditional probability distributions were not aligned successfully. The proposed sample selection approach led to formation of dense clusters corresponding to each class label thus helping in aligning the conditional probability distributions.

Table 5.3 represents the class wise accuracy on target dataset of baseline architecture ADDA(SVHN-MNIST) and same architecture with the sample selection strategy. The baseline architecture resulted in a mode collapse for instances of class label 9 thus resulting in class accuracy of 0.02 whereas the proposed approach prevented the mode collapse thus improving the class accuracy to 0.419.

Classes	Baseline	Sampling
0	0.546	0.676
1	0.974	0.991
2	0.975	0.976
3	0.946	0.916
4	0.920	0.573
5	0.854	0.918
6	0.7	0.718
7	0.675	0.762
8	0.735	0.712
9	0.02	0.419

TABLE 5.3: Class wise accuracy on ADDA(SVHN-MNIST).

5.2 Segmentation

Table 5.4 and 5.5 display the results on STARE and DRIVE dataset obtained by the baseline architecture(V-GAN) and V-GAN equipped with proposed sample selection strategy for segmentation. Various metrics such as AUC_PR, AUC_ROC, AUC_SUM and DICE_COEFF were computed and used as the evaluation metrics. The proposed strategy has resulted in substantial improvement in performance.

Metrics	V-GAN	V-GAN (Sampling)
Iteration	49500	1500
AUC_PR	0.806	0.90
AUC_ROC	0.912	0.967
DICE_COEFF	0.765	0.826
ACC	0.948	0.964
Sensitivity	0.764	0.77
Specificity	0.971	0.987
Score	5.168	5.442
AUC_SUM	1.718	1.868
AVG_PT	62.679	76.235

TABLE 5.4: Comparison of baseline and proposed approach on STARE dataset.

Metrics	V-GAN	V-GAN (Sampling)
Iteration	17500	1000
AUC_PR	0.8022	0.86
AUC_ROC	0.936	0.97
DICE_COEFF	0.7422	0.788
ACC	0.935	0.947
Sensitivity	0.794	0.841
Specificity	0.954	0.961
Score	5.165	5.36
AUC_SUM	1.739	1.830
AVG_PT	44.11	65.761

TABLE 5.5: Comparison of baseline and proposed approach on DRIVE dataset.

Figure 5.3 and 5.5 represent the segmented output from base architecture(V-GAN) on DRIVE and STARE Dataset. Figure 5.4 and 5.6 represent the segmented output from V-GAN equipped with proposed approach. Since, the segmentation with respect to the output patches were weighted therefore, during training more attention was given to incorrect patches thus providing the model a more guided path to optimal solution resulting in a more sharper segment generation.

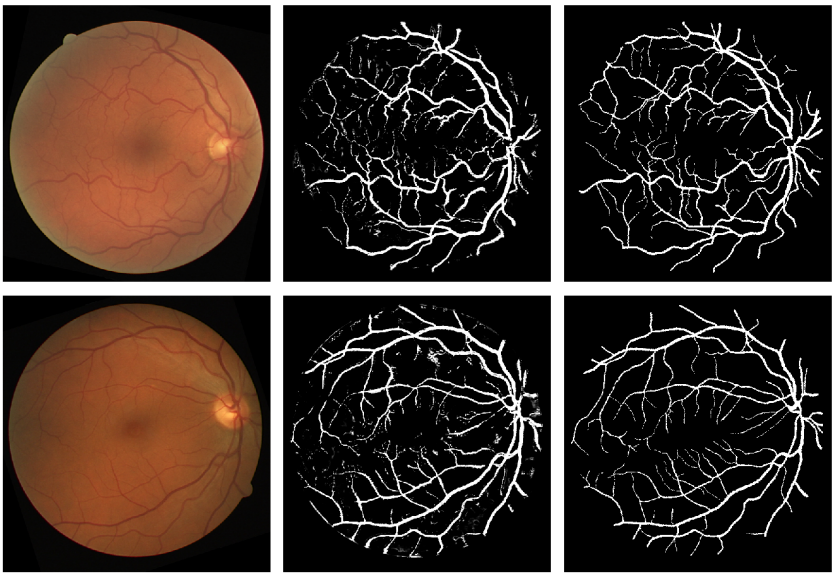

FIGURE 5.3: Results of baseline on drive dataset.

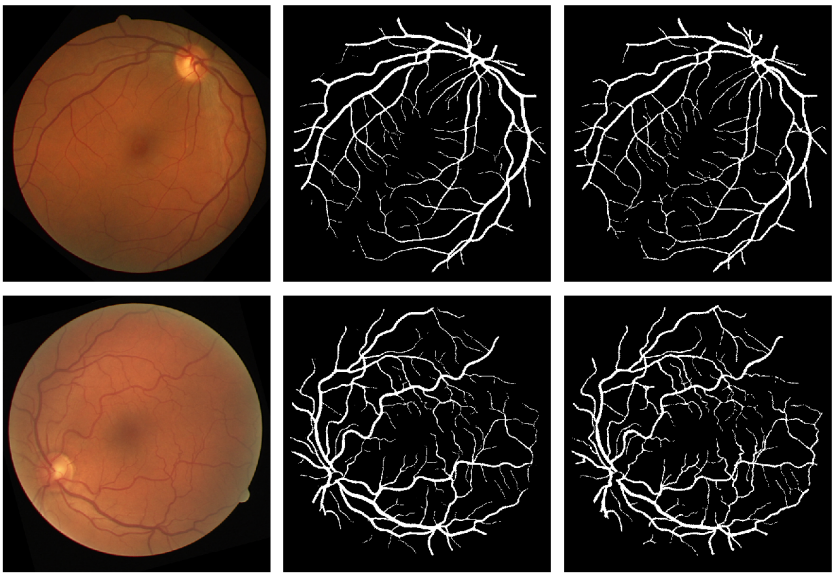

FIGURE 5.4: Results of sampling on drive dataset.

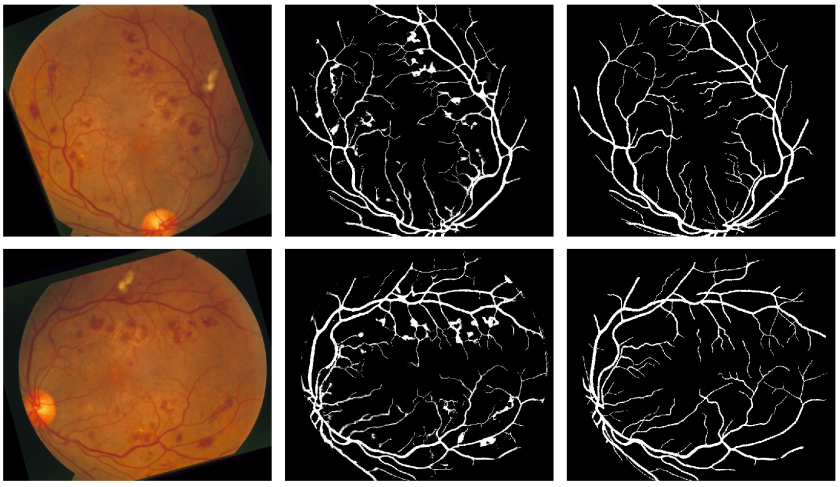

FIGURE 5.5: Results of baseline on STARE dataset.

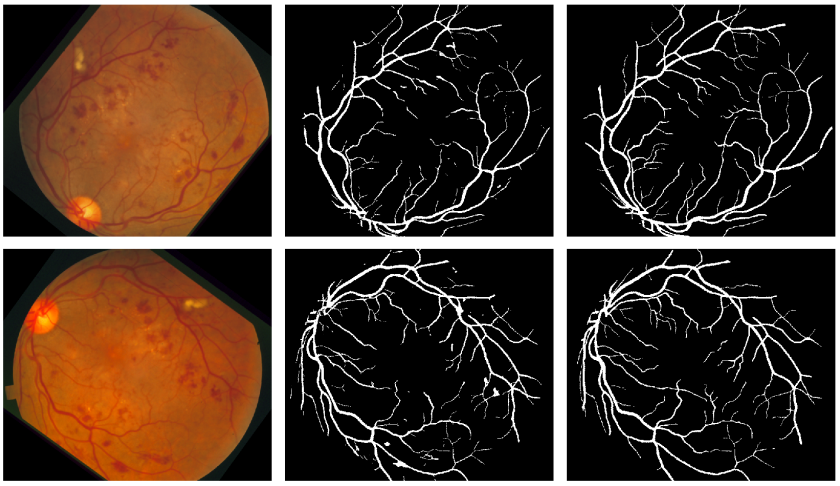

FIGURE 5.6: Results of sampling on STARE dataset.

Chapter 6

Conclusion

We have proposed a new sample selection strategy which can be incorporated in various supervised deep learning framework involving a discriminator for adversarial training. The proposed strategy uses the discriminator's output as measure for relevant samples. Thus the sampling strategy resulted in improved performance compared to baselines across various tasks such as unsupervised domain adaptation and segmentation. For the task of unsupervised domain adaptation the proposed approach resulted in an improvement in accuracy on target dataset as compared to baselines. Also, further analysis suggested an improvement in alignment of feature distributions of both the source and target domains and a reduction in mode collapse. Since, most of approaches follows standard backpropagation training therefore, The proposed approach is scalable and can be easily incorporated in most of the existing deep learning frameworks designed for unsupervised domain adaptation. For the task of segmentation, we considered the problem of retinal vessel segmentation using generative adversarial networks. After incorporating the proposed sampling strategy, The discriminator worked as an attention mechanism by providing the importance weights to the segmentation loss of individual patches thus constraining the generator to focus more on less realistic patches. Thus leading to an improvement in performance as well as reduction in training time. Compared to the baseline, The proposed approach led to reduction in false positives as well as generation of more clear and detailed edges similar to the gold standard. Overall, the proposed approach was capable to sampling relevant samples dynamically during training and can be easily implemented or incorporated in most of the deep learning frameworks using any existing deep learning packages.

Chapter 7

Future Work

Future work constitutes of evaluation and analysis of proposed work across various other tasks apart from domain adaptation and segmentation. Also, evaluation and analysis on large-scale domain adaptation datasets such as Office-31 and complex segmentation datasets. Since, the proposed architecture assigns weights to an instance irrespective of its class label. This might result in low weight assignment for all instances of a particular class. For our future work we would like to also focus on sample selection strategies involving class label information.

Bibliography

[1] Shai Ben-David et al. "A theory of learning from different domains". In: *Machine learning* 79.1-2 (2010), pp. 151–175.

[2] Shai Ben-David et al. "Analysis of representations for domain adaptation". In: *Advances in neural information processing systems*. 2007, pp. 137–144.

[3] Konstantinos Bousmalis et al. "Unsupervised pixel-level domain adaptation with generative adversarial networks". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 2. 2017, p. 7.

[4] Zhangjie Cao et al. "Partial transfer learning with selective adversarial networks". In: *arXiv preprint arXiv:1707.07901* (2017).

[5] Aaron Chadha and Yiannis Andreopoulos. "Improving Adversarial Discriminative Domain Adaptation". In: *arXiv preprint arXiv:1809.03625* (2018).

[6] Chao Chen et al. "Joint Domain Alignment and Discriminative Feature Learning for Unsupervised Deep Domain Adaptation". In: *arXiv preprint arXiv:1808.09347* (2018).

[7] Bharath Bhushan Damodaran et al. "DeepJDOT: Deep Joint distribution optimal transport for unsupervised domain adaptation". In: *arXiv preprint arXiv:1803.10081* (2018).

[8] Yaroslav Ganin and Victor Lempitsky. "Unsupervised domain adaptation by backpropagation". In: *arXiv preprint arXiv:1409.7495* (2014).

[9] Judy Hoffman et al. "Cycada: Cycle-consistent adversarial domain adaptation". In: *arXiv preprint arXiv:1711.03213* (2017).

[10] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. "Learning to cluster in order to Transfer across domains and tasks". In: *arXiv preprint arXiv:1711.10125* (2017).

[11] Issam Laradji and Reza Babanezhad. "M-ADDA: Unsupervised Domain Adaptation with Deep Metric Learning". In: *arXiv preprint arXiv:1807.02552* (2018).

[12] Yanghao Li et al. "Revisiting batch normalization for practical domain adaptation". In: *arXiv preprint arXiv:1603.04779* (2016).

[13] Zachary C Lipton, Yu-Xiang Wang, and Alex Smola. "Detecting and correcting for label shift with black box predictors". In: *arXiv preprint arXiv:1802.03916* (2018).

[14] Mingsheng Long et al. "Deep transfer learning with joint adaptation networks". In: *arXiv preprint arXiv:1605.06636* (2016).

[15] Mingsheng Long et al. "Learning transferable features with deep adaptation networks". In: *arXiv preprint arXiv:1502.02791* (2015).

[16] Mingsheng Long et al. "Unsupervised domain adaptation with residual transfer networks". In: *Advances in Neural Information Processing Systems*. 2016, pp. 136–144.

[17] Massimiliano Mancini et al. "Boosting Domain Adaptation by Discovering Latent Domains". In: *arXiv preprint arXiv:1805.01386* (2018).

[18] Rui Shu et al. "A DIRT-T Approach to Unsupervised Domain Adaptation". In: *arXiv preprint arXiv:1802.08735* (2018).

[19] Jaemin Son, Sang Jun Park, and Kyu-Hwan Jung. "Retinal vessel segmentation in fundoscopic images with generative adversarial networks". In: *arXiv preprint arXiv:1706.09318* (2017).

[20] Eric Tzeng et al. "Adversarial discriminative domain adaptation". In: *Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 2. 2017, p. 4.

[21] Eric Tzeng et al. "Deep domain confusion: Maximizing for domain invariance". In: *arXiv preprint arXiv:1412.3474* (2014).

[22] Jiawei Wang et al. "Domain Confusion with Self Ensembling for Unsupervised Adaptation". In: *arXiv preprint arXiv:1810.04472* (2018).

[23] Jun Wen et al. "Exploiting Local Feature Patterns for Unsupervised Domain Adaptation". In: *arXiv preprint arXiv:1811.05042* (2018).

[24] Jing Zhang et al. "Importance Weighted Adversarial Nets for Partial Domain Adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8156–8164.