Lehrstuhl für Data Science

Deep Domain Adaptation in Computer Vision

Masterarbeit von

AUTHOR

1. Prüfer	2. Prüfer
Prof. Dr. Michael Granitzer	Mr. Sahib Julka

October 16, 2020

Contents

1 Introduction

Artificial Intelligence plays a key role and is the driving force behind the rapid advances in automation and digitization that are changing & shaping our daily lives. Due to the rapid, exponential growth of technology, in the near future we may see the *utopian world* of superhuman intelligent robots or, as it is called, the *technological singularity*[1]. In machine learning, there is a theoretical limit to the error rate known as Bayes Optimal Error[2] or Human Error Rate and this limit is signified as the optimal error at which the model is able to perform its best. However, human intelligence is considered a competitive benchmark that machine intelligence seeks to emulate and eventually exceed. The fundamental characteristic of human intelligence is the capability of the human brain to adapt and transfer knowledge between different domains and learn the general features, and it has proven to be incredibly difficult to reproduce in silicon and software. But the question arises - Can the machine learn the most general characteristics like humans and apply this knowledge to solve different real-time tasks in different domains just like humans? This topic will be further examined in detail and the obstacles that stand in the way of accuracy at the human level will be discussed.

With the introduction of *neural networks* [LeC+89; RHW85] or *deep learning* in today's paradigm, triggered in 2012 by the spectacular success of AlexNet architecture [KSH12] via the ImageNet challenge [Den+09], we are now able to transform our thoughts & ideas into reality. Various types of tasks that constitute the basis for modern Computer Vision (CV) are object tracking, image classification, image captioning, instance and semantic segmentation, object and pose detection, and many more. Due to the effectiveness of Convolutional Neural Networks (CNNs) [LeC+98] over the last ten years, there has been enormous progress in this field. These deep networks can represent high-level abstractions through several layers of nonlinear transformations, thus simulating the perception of the human brain. CNN achieves state-of-the-art results for large-scale

[1] `https://en.wikipedia.org/wiki/Technological_singularity`
[2] `https://en.wikipedia.org/wiki/Bayes_error_rate`

supervised learning problems, and the reason for this is the availability of *large-scale labeled training data* [He+16; KSH12]. If we do not have huge manually annotated task-specific target dataset for our particular problem, we use *Transfer Learning*[3] to adapt the features of the CNN network trained on large source dataset and *fine-tune* some of the upper layers of this network using small annotated task-specific dataset [Yos+14]. In image recognition tasks, deep learning models have empirically outperformed humans as well [He+15]. However, the main assumption in such evaluations is that the test data comes from the same underlying distribution of training data.

1.1 Motivation

CNNs learn the representations which are generically useful across a variety of tasks and visual domains, which has led to remarkable success in many CV learning problems stated above. The concept of *Domain Adaptation* (DA) can be described with an example. An organization is building a new facial recognition authentication system that allows employees to log into their new application using their smartphones. The face recognition model is trained on the employee's existing passport-sized images, which are made available upon entry into the organization. The inference model takes the input image from the front camera of the smartphone and provides the authentication of the employee to log in to the application. This is a clear case of DA, as the trained model is applied to a similar but different data distribution during the recognition stage. The images captured by the forward-facing camera may differ from the images submitted during the joining time due to varying background and lighting, or due to different resolutions, or due to different face positioning, occlusion, and different camera qualities. Ultimately, this leads to unsatisfactory recognition performance, and the system's inability to produce a good performance at test images can be attributed to a *domain gap* that exists between the train and test images.

Let's take a look at some of the motivational factors and challenges we might encounter when solving the above problem or a transfer learning task.

[3] `https://en.wikipedia.org/wiki/Transfer_learning`

1.1.1 Expensive Data Acquisition and Annotation Problems

To fully exploit the potential of the *supervised learning* task, we need an extensive annotated training data set. However, in some domains, it is impossible to collect annotated data, and this is often time-consuming. It is also sometimes quite expensive and requires human experts in the loop. Also, collecting more labeled samples does not necessarily lead to better generalization performance [TE11].

1.1.2 Challenges of Dataset Bias or Domain Shift

The second major challenge that we have to deal with even after the availability of perfect annotations is the presence of *dataset-bias* [TE11]. In practice, the direct transfer of features from different domains usually does not work very well because the data distributions of both the domains can change. This phenomenon is known as *domain shift* (DS) [Gre+09] and is very common when *training* and *test* sets come from similar but different distributions, and also data can change over time and space [SS16]. Other category of DS is known as *Concept shift or Concept drift*, that requires labelled data from both the domains [WK96]. Various examples for domain shifts are: from simulation to real-world, from source dataset to target dataset, from RGB images to depth, and from Computer-aided design (CAD) models to real images.

1.1.3 Reverse-Engineer the Brain

The National Academy of Engineering (NAE) has identified 14 challenges for the 21st century[4], ranging from Sustainability, Cleaner Resources, and Medicine. Out of them, one of the biggest challenges is the reverse engineering of the human brain. It is about creating machines that can imitate human intelligence, which will go far beyond artificial intelligence, with applications in the healthcare sector, manufacturing, and communication. The ability to transfer already acquired knowledge and adaptation to a wide range of data inputs will be crucial for the success of this endeavor. A possible solution to this complex challenge must account for knowledge transfer and DA.

[4] `http://www.engineeringchallenges.org/challenges.aspx`

Our end goal is to generate the features that are (i) discriminative for the main learning task on the source domain and (ii) indiscriminate in terms of bias between source and target domain [Gan+16]. Various techniques of DA attempts to mitigate the harmful effects of DS either by finding a mapping from the source distribution to the target distribution or by mapping both domains into common shared domain where the distributions are aligned. There may be several such mappings, and they may be different from each other, and Deep Neural Networks (DNNs) allow these mappings to be learned.

1.2 Research Goals

This thesis's overall goal is to make progress in solving DS in visual object recognition through representation learning in the context of DA. This thesis discusses the development of a generalizable DA model that could work in different CV tasks. We will also perform pruning of our proposed model in order to develop a model with minimal complexity and the best possible domain transfer task accuracy. To achieve the overall goal, we define the following research objectives. (Refer to sec.- 3.1 for further details on the proposed methodology)

- How sophisticated is the target network compared to the source network in terms of complexity and to find the right threshold for pruning the target network so that there is no significant reduction in the performance of domain transfer task?
- How does selecting different CNN feature extractors (e.g., ResNet50, VGG16) in our proposed architecture change domain transfer task accuracy?
- How do pruning lower magnitude weights of the proposed model (technique-2) would impact domain transfer task accuracy? (Refer to fig.-3.3)

To design a generalized approach, we will perform several experiments with different datasets to answer the above questions.

1.3 Contribution of the thesis

A hierarchical feature space procedure (with deep learning) will be developed. We will create a CNN named as *Deep Domain Adaptive Concatenation Network* (DDACN) to create cross-domain representation by mapping cross-domain features into a common space, by minimizing the domain discrepancy using Maximum Mean Discrepancy (MMD), or by aligning moments of the two distributions using Correlation Alignment (CORAL). The performance of this model is evaluated against other DA methods on a series of domain adaptive transfer tasks.

Theoretically, we propose the basic model technique to reduce the distributional difference between the source and a target domain and achieve good performance on the target domain's image classification task. The custom pruning layer could help to reduce the unwanted lower weights of the target feature extractor, reduce its complexity, and at the same time, achieve an acceptable image classification accuracy.

2 Background

The previous chapter gave a brief introduction to the topic and our motivation to solve various real-time domain-shifted CV tasks. In this chapter, we are taking a formal approach to the precise definition of DA in the landscape of transfer learning problems.

DA is ubiquitous in the field of machine learning, but we focus primarily on methods developed for CV.

2.1 Technical Background

2.1.1 Notations

All the transfer learning techniques can be classified using two terms: *domain* and *task* [PY09]. A domain $\mathcal{D}$ is defined as $\mathcal{D} = \{\mathcal{X}, P(X)\}$, where $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$, known as feature space, and $P(X)$ is the marginal probability distribution that reigns over that feature space. Given a specific domain $\mathcal{D}$, a task $\mathcal{T}$ under that domain is defined as $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$, where $\mathcal{Y}$ is the label space, and $f(\cdot)$ is known as mapping function $f : \mathcal{X} \rightarrow \mathcal{Y}$. In a traditional *supervised machine learning* approach, this function f is learned using a set of given training data pairs $\{x_i, y_i\}$ where $x_i \in \mathcal{X}$, and $y_i \in \mathcal{Y}$. This mapping function can also be seen as a posterior probability $P(Y \mid X)$, which predicts the label for a new data point x [WD18]. Venkateswara [Ven17] also purposed to define a domain $\mathcal{D}$ as joint space of features & the labels and also their joint probability distribution in some scenarios, such that $\mathcal{D} = \{(\mathcal{X} \times \mathcal{Y}), P(\mathcal{X}, \mathcal{Y})\}$.

We generally deal with two domains in domain adaptation problems: a *source* and a *target*. More formally, source dataset can be represented as a collection of data points $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, where $x_i^s \in \mathcal{X}^s$ and $y_i^s \in \mathcal{Y}^s$ or $\mathcal{D}^s = \{\mathcal{X}^s, P(\mathbf{X}^s)\}$ and a target dataset is given by $\mathcal{D}^t = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$, where $x_i^t \in \mathcal{X}^t$ and $y_i^t \in \mathcal{Y}^t$ or $\mathcal{D}^t = \{\mathcal{X}^t, P(\mathbf{X}^t)\}$.

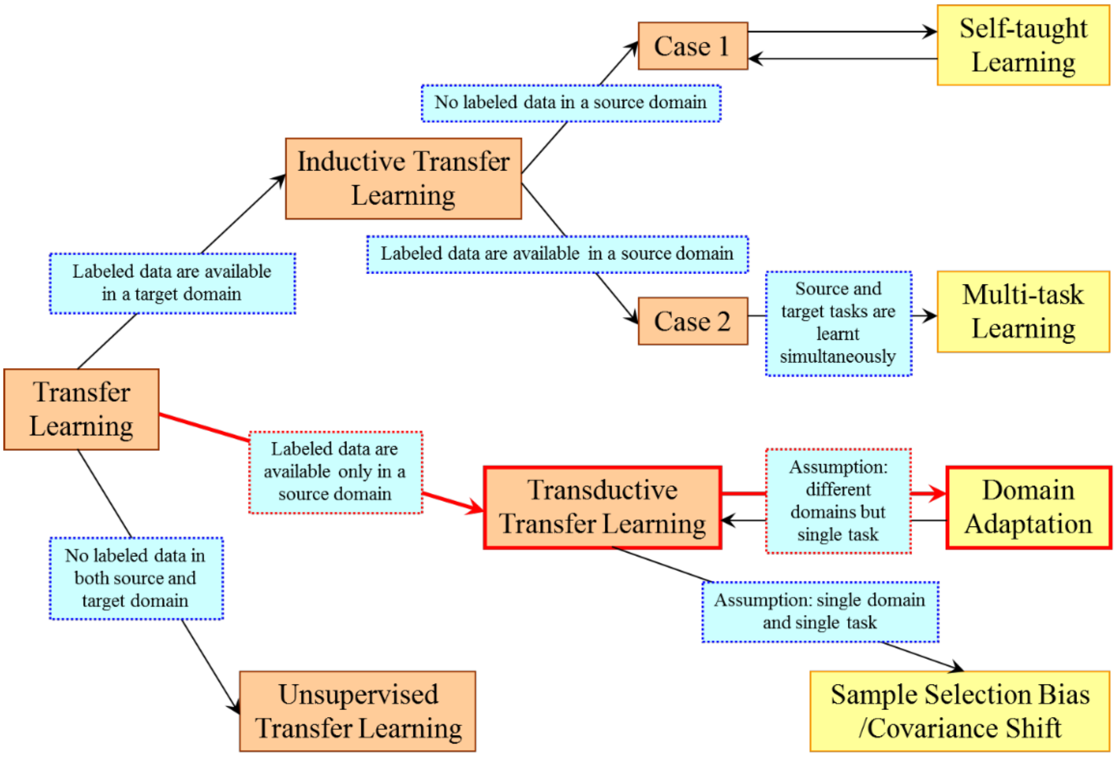

Figure 2.1: Different transfer learning approaches, Courtesy to Pan and Yang [PY09].

Similarly, source and target tasks can be represented as $T^s = \{\mathcal{Y}^s, P(Y^s \mid X^s)\}$ and $T^t = \{\mathcal{Y}^t, P(Y^t \mid X^t)\}$ respectively. If two domains are equal, i.e. $\mathcal{D}^s = \mathcal{D}^t$ and corresponding tasks are also equal, i.e. $\mathcal{T}^s = \mathcal{T}^t$, traditional ML techniques are used to solve these problems, and here $\mathcal{D}^s$ denotes the training set and $\mathcal{D}^t$ the test set.

2.1.2 Transfer Learning

Under this learning, a model is trained on a source domain or task and evaluated on a different but related target domain or task, where either the tasks or domains (or both) differ [PY09; WKW16; Goo+16; DKC10]. If in this case, $\mathcal{D}^s \neq \mathcal{D}^t$, the model trained on $\mathcal{D}^s$ might result in the poor performance on $\mathcal{D}^t$, if $\mathcal{T}^s = \mathcal{T}^t$ [Csu17b]. And if $\mathcal{T}^s \neq \mathcal{T}^t$, trained model can't be applied directly to $\mathcal{D}^t$. We need a certain relationship between source and target domain to exploit the related information from $\{\mathcal{D}^s, \mathcal{T}^s\}$ to learn the posterior probability $P(Y^t \mid X^t)$. Fig.-2.1 shows three main types of TL techniques depending on the different situations concerning source and target domains and the corresponding tasks [PY09]. These are the transductive TL, inductive TL and unsupervised TL.

2.1.3 Domain Adaptation

Domain adaptation is a particular case of transfer learning and is a subset of *transductive* transfer learning where the tasks for both source and target datasets are the same ($\mathcal{T}^s = \mathcal{T}^t$), but the domains are different ($\mathcal{D}^s \neq \mathcal{D}^t$) [Gho+20]. Different feature spaces or different marginal probability distributions between source and target datasets lead to *domain shift*, also known as *domain divergence* or *dataset bias* [WD18]. This divergence can also be represented as divergence of joint probability distributions of both the domains, i.e. $\mathcal{P}^s(\mathcal{X}, \mathcal{Y}) \neq \mathcal{P}^t(\mathcal{X}, \mathcal{Y})$ [Dud19].

Homogeneous vs Heterogeneous Domain Adaptation

Based on different domain divergences, it is classified into two main categories, namely: *Homogeneous* DA and *Heterogeneous* DA. Besides, homogeneous DA is referred to cases when the source and target feature space is the same (($\mathcal{X}^s = \mathcal{X}^t$) and ($P(X)^s \neq P(X)^t$)), while in heterogeneous DA, features in source and target may have different representations spaces ($\mathcal{X}^s \neq \mathcal{X}^t$), as well as different modalities, and the dimensions may also generally differ ($d^s \neq d^t$). The end goal of DA is to estimate the $\hat{\mathcal{P}}^t(\mathcal{X}, \mathcal{Y})$ using the learned distribution $\hat{\mathcal{P}}^s(\mathcal{X}, \mathcal{Y})$.

Unsupervised and Semi-Supervised Domain Adaptation

Based on the availability of labels in the target domain, we can further divide the DA into two categories, namely *unsupervised* (UDA) and *semi-supervised* (SSDA). If only source label information is specified, it is UDA [Ben+07; Cha+19; KLB19; Nat+18; Kur+19; Lon+16; Ben+10; Tze+17]; otherwise, if a small group of examples is specified in the target domain in addition to the source examples, it is SSDA [Csu17a].

One-Step vs Multi-Step Domain Adaptation

All the above DA settings assume that both source and target domains directly relate to each other, and they are quite similar [Gho+20]. However, in minimal scenarios, we may come across situations where there is little overlap between two domains, and fortunately, we may find intermediate domains that can draw both the domains closer to each other as compared to their original distance. The former setting is known as *one-step* DA, as we can transfer the knowledge in one step only. Later is know as *multi-step or transitive* DA as we use to build an intermediate bridge or intermediate-steps to connect two seemingly unrelated domains and then perform one-step DA via this bridge [WD18].

Open/Partial-Set vs Closed-Set Domain Adaptation

In all the above cases of DA, it is assumed that the same source and target label space, i.e., $\mathcal{Y}^s = \mathcal{Y}^t$. This setting of DA falls under the category of *closed-set* DA. However, in the real-time scenarios, only a few categories of interest are shared between source and destination domains for recognition tasks. With an *open/partial-set* DA, the idea of the same label space is dropped, we have more flexibility to train the model with only categories of interest, and we discard the *unknown* classes. The only assumption that makes this setting is that a *shared* label space between the two domains is known and that it contains the interested categories [PG17].

Single-Source and Multi-Source Domain Adaptation

All the methods we discussed above comes under the category of *single-source* DA (SDA). In real-time scenarios, labeled or training data can be taken from multiple sources having different distributions [SSW15; BRR16]. Under this DA setting, we can redefine the source as: $S = \{(x_1, y_1, d_1), \ldots, (x_n, y_n, d_n)\} \subset \mathcal{X} \times \mathcal{Y} \times \mathcal{D}$ and it is known as *multi-source* DA (MDA). However, we can use an existing method of combining all the sources into single-source. However, there is a gap in the use of this approach, i.e., domain-shifting is done in parallel between source and a target domain and between different sources. And training our model on the combined source data from different sources can interfere with each other during the learning process [Rie+18] and lead to a poor generalization performance over target domain. Scientists and practitioners have shifted

DA Setting	***#Target Domains***	***Domain Labels***	***Open Classes***	***Open Domains***
Unsupervised Domain Adaptation (UDA)	Single	known	No	No
Multi-Target Domain Adaptation	Multiple	known	No	No
Open/Partial Set Domain Adaptation	Single	known	Yes	No
Open Compound Domain Adaptation (OCDA)	Multiple	unknown	No	Yes

Table 2.1: Different Domain Adaptation Settings, table from [Liu+19]

their focus from using shallow methods to deep-learning approaches to perform MDA and it is an active area of research. Some of the examples of multi-source DA can be seen in [Sun+11; Cha+12; MMR09].

Open Compound Domain Adaptation

All the previous work done by scientists and practitioners is based on the assumption that there always exists a clear distinction between the two domains, which is often not true in real-time scenarios. It is a type of DA setting under which the target domain is a compound of multiple homogeneous domains, and data is collected form mixed and novel situations in this way [Liu+19]. Table-2.1 shows some of the different possible settings of DA, where *Domain Labels* indicates the domain to which each instance belongs, *Open Classes* are the classes that appear during the testing time, but training and *Open Domains* are the domains whose data is unseen during the training time of the model.

Domain Generalization

Domain Generalization (DG) is one step ahead of DA, as we sometimes do not even have unlabeled target samples available during the training phase. Similar to DA, It also tries to mitigate the problem of *domain bias*, but with an additional requirement discussed before. In this setting, multiple sources can be used and the algorithm tries to learn the most general features from the source domain(s) and, in case of classification tasks, to build the most robust classifier. It is also an active research area, and much work has been done in this area in the case of object recognition problems [FXR13; Kho+12; Xu+14]. Therefore, DA algorithms that include unlabeled target samples are generally not applicable to the problem of DG.

A considerable amount of research has been done in the area of homogeneous and one-step domain adaptation. However, the potential of heterogeneous and multilevel domain adaptation approaches for visual understanding is relatively unexplored. In our thesis, we concentrate on homogeneous, single-source, closed-set and only one-step DA.

2.1.4 Pruning of Deep Neural Networks

It is well known that DNNs are likely to have increased performance but at a cost of increased fast memory requirements. One way to decrease the memory requirements is to introduce sparsity into a network's connections. Sparsity in traditional neural networks by pruning weights, generating random architectures or utilizing skip connections is being studied widely from the past few years and sparse structures in traditional neural networks have shown a promise of training potential. *Magnitude-based weight pruning* methodology will be used in our experiments to design our second proposed approach.

2.2 Related Work

As it is shown in the fig. 2.4, we have categorized the methods of deep domain adaptation into five main groups based on the technology of these methods adopted.

Discrepancy-based deep adaptation techniques aim to minimize discrepancies between source and target domain in a latent space, and working concepts from early shallow methods inspire them. Discrepancy methods can be subdivided into *Maximum Mean Discrepancy* [Lon+16; Tze+14; Luo+18; Den+20], *Correlation Alignment* [SS16; PS18; Rah+20], *Entropy Minimization* [Soh+18; Zha+20; Man+19], *Moment Matching* and *Wasserstein Discrepancy* methods [Mad+20]. Figures-2.2 & 2.3 clearly explains the workflow of our Discrepancy-based techniques. They typically use Siamese networks, which use architecture with a shared weight between source and target feature extractors. The total loss of the network can be represented as the sum of task loss and discrepancy loss; thus $\mathcal{L}_{total} = \mathcal{L}_{task} + \mathcal{L}_{discrepancy}$.

Adversarial-based deep adaptation approaches are divided into *Discriminative* & *Generative* methods. The first method uses a discriminator between source and target feature

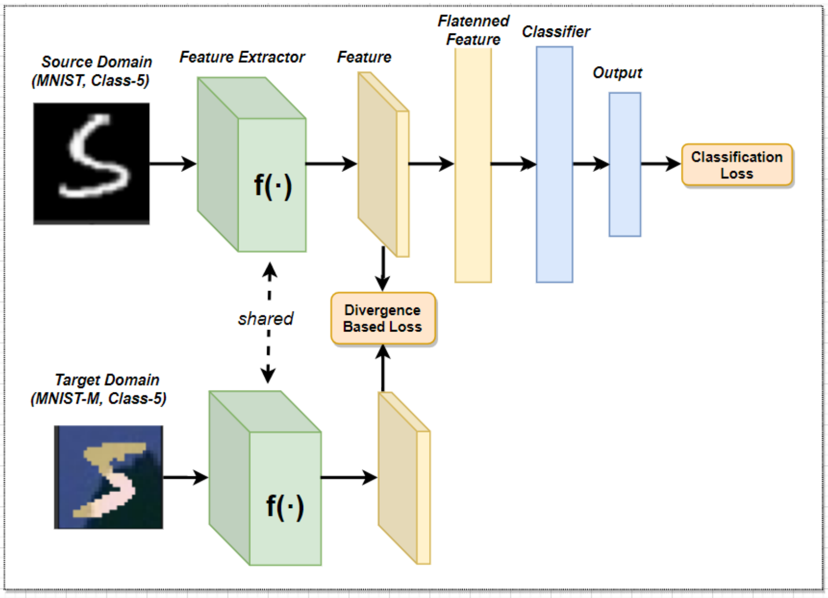

Figure 2.2: Discrepancy-Based technique, training phase

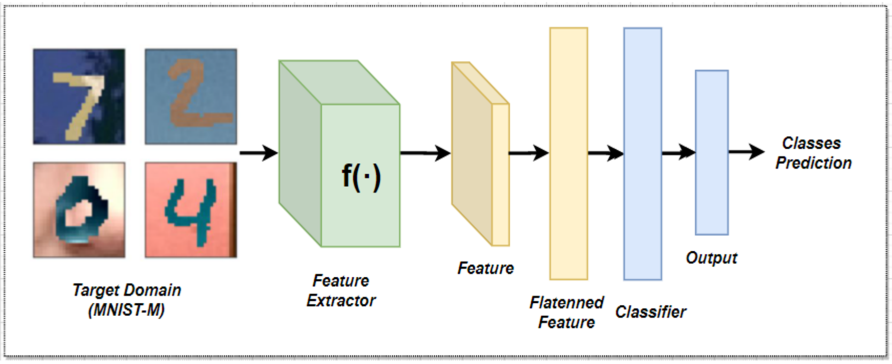

Figure 2.3: Discrepancy-Based technique, inference phase

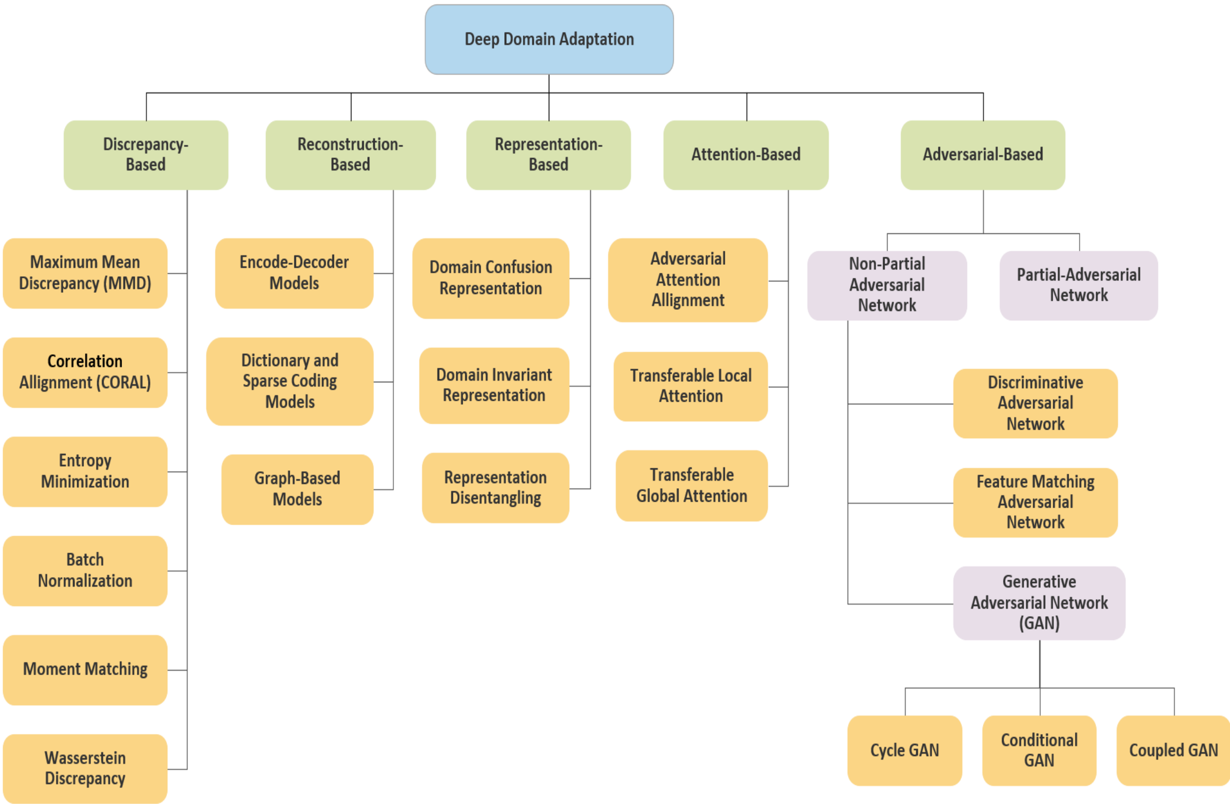

Figure 2.4: Taxonomy of Deep Visual UDA, Reconstructed from [Mad+20].

extractors to force the original classifier to produce domain invariant features. The second method uses a generative model to generate the target samples by imitating the source domain distribution.

Ganin et al. proposed the *Domain Adversarial Neural Network (DANN)*, in which the authors trained a deep neural network in a domain-adversarial manner for domain adaptation based on image classification. The feature extractors (lower layers) of the model are fed into two branches. The first branch is a typical softmax classifier trained on source-labeled data, and the second branch is a domain classifier learning to distinguish between features of the source and the target domain. They also proposed introducing the gradient reversal layer before the second branch, which increased the domain confusion loss while reducing the classification loss to extract the domain invariant features [Gan+16].

3 Methods

In this chapter, we will talk about the implementation approach we would be using when building our DA model. We will cover the entire workflow of the adaptation process and give a brief overview of the underlying model architecture and its basic building blocks. As discussed in the last chapter, there is already some research done on this topic, and many DA-based model architectures have been proposed in the past. We will discuss our purposed techniques in detail and emphasize the importance of their usage.

3.1 Approaches for Model creation

3.1.1 Deep Domain Adaptive Concatenation Network (DDACN)

Fig.-3.1 & 3.2 represent our proposed base-model's flowchart in the training and inference phase, respectively. Different CNN model architectures are used as feature extractors, and the model will be trained on the provided source labels. Our proposed technique is inspired from the Discrepancy-based DA techniques as discussed in sec.-2.2. Below are some of the highlights of proposed model architecture.

- Concatenation of Source and Target CNN Feature Extractors: Here, we concatenate both source and target feature extractors' outputs.
- Image classification on concatenated features: The outputs from above step are passed to the classifier to predict the output class.
- Domain invariance minimization: Also, individual outputs from source & target feature extractors are used to compute the domain invariance loss.

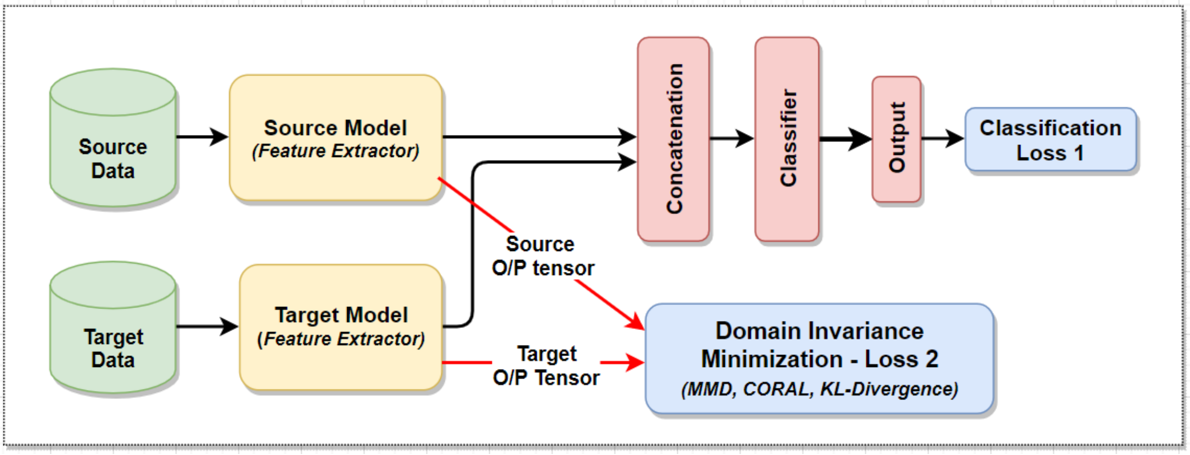

Figure 3.1: Base-Model (DDACN) flowchart, Training time

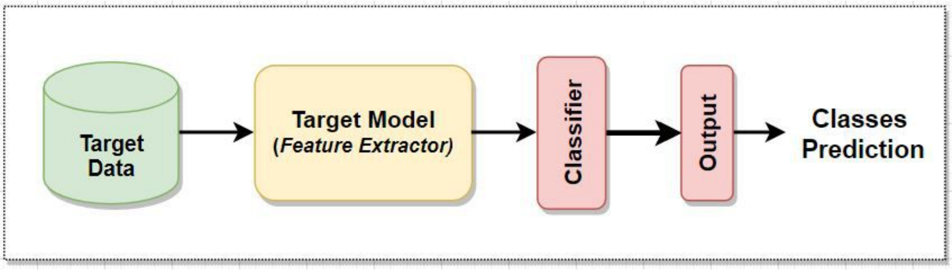

Figure 3.2: Base-Model (DDACN) flowchart, Inference time

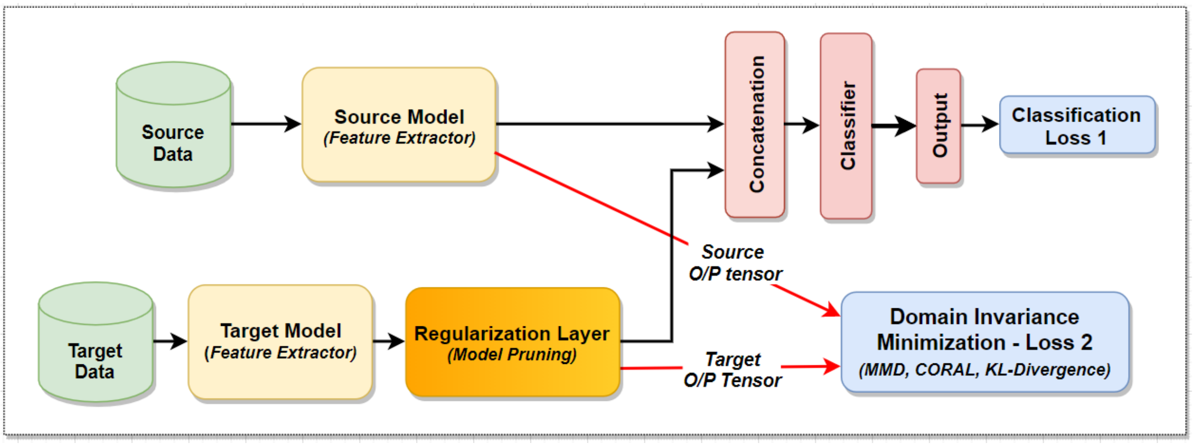

Figure 3.3: Pruned Base-Model, Training time

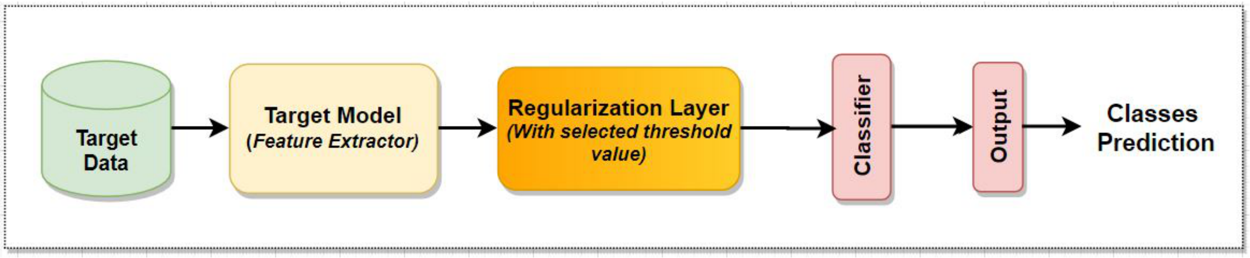

Figure 3.4: Pruned Base-Model, Inference time

- **Total Loss:** The model will learn the most general features by minimizing over the combination of both the losses, where λ is a parameter.

$$\mathcal{L}_{total} = \mathcal{L}_{class} + \lambda \mathcal{L}_{domain\,invariance} \tag{3.1}$$

- **Inference Model:** The features learned by the target feature extractor during the training time will be passed to the classifier to predict the output class.

3.1.2 Pruned DDACN model

Fig.-3.3 & 3.4 represent our pruned base-model flowchart in the training and inference phase, respectively. The following steps will be performed to create a pruned-target feature extractor architecture:

- Construct a custom prunable layer with a parameterized threshold (to drop the lower weights of the target feature extractor).
- Train our pruned model repeatedly for a given number of epochs, each with a given threshold.
- Retrieve the Weight matrix of target feature extractor (at a specific threshold), prune the weights below a certain threshold and analyze its effect on the overall performance.
- Compare the performance of our pruned model (at different thresholds) with the base-model.
- Examine and interpret the effects of our base-model pruning and find the most appropriate threshold with minimal loss of accuracy.

- **Inference Model:** The features learned by the target feature extractor is fed to the pruning layer with the selected best threshold value (after observation) and passed to the classifier to predict the output class.

3.2 Datasets

In this thesis, the proposed techniques are evaluated using a following benchmark datasets for visual objects.

- **MNIST:** MNIST dataset is famous handwritten digit dataset which is derived from NIST dataset. It has a training set of 60k examples, and a test set of 10k examples [LeC98].
- **MNIST-M:** [Gan+16] is coloured variant of standard MNIST dataset where the background is substituted by a randomly extracted patch obtained from color photos of BSDS500 [Arb+10].
- **USPS:** It is also a handwritten digit dataset automatically scanned from envelopes by the U.S. Postal Service containing. The training set and the test set have 7291 and 2007 samples, respectively.
- **Synthetic-Signs:** It contains 100,000 samples of common street signs from Wikipedia, which have been artificially transformed to simulate different imaging conditions. [Moi+13].
- **GTSRB:** The German Traffic Signs Recognition Benchmark consists of 51,839 cropped images of German traffic signs [Sta+11].

Once the stable implementation is ready, it can be generalized to make it flexible for working with other real-world challenging problems like Real-to-Synthetic images domain adaptation[1].

[1] https://ai.bu.edu/visda-2018/

4 Experiments

In our thesis, we concentrate on homogeneous, single-source, closed-set and only one-step DA. We will solve a much simpler problem than existing complex real-time problems involving multiple source domains, sometimes requiring multi-step DA, and dealing with unseen classes or even unseen domains during the evaluation period (open/partial set or open-compound DA). Our experiments assume that we have a single source domain as input, the same number of classes are present in both the datasets (hence closed-set DA) and the same feature space (hence homogeneous DA). This kind of problem can be solved in one step, so we use a single-step DA. As discussed in sec.-3.1, using our proposed methodology, we will be performing the domain transfer tasks like MNIST $\longrightarrow$ MNIST-M digits transfer, MNIST $\longrightarrow$ USPS digits transfer, USPS $\longrightarrow$ MNIST digits transfer, and Synthetic-Signs $\longrightarrow$ GTSRB street signs transfer.

4.1 Evaluation Metrics

As we perform an image classification task, the use of classification accuracy is a simple and effective way to compare the results with the most advanced techniques currently available. Confusion matrices will be generated to measure the performance of our model and find the misclassified samples. We can use t-SNE embeddings plots to visualize the learned feature space.

5 Preliminary Results

The network architecture and training procedures of DDACN were implemented in TensorFlow. The source feature extractor is consists of a *Resnet50* model with weights pre-trained on the ImageNet dataset. The target feature extractor consists of *VGG16* model initializated with *Xavier and He Normal* distribution. Refer to tab.-5.1 for more details about the complexity of both feature extractors. Compared to the proposed methodology in sec.-3.1, different experimental setup is used here for the inference phase. Fig.-5.1 shows the inference phase flowchart, which takes two input images for the class prediction. Below experiments were tested using the discussed setup. However, the latest experimental setup for the inference phase is shown in fig.-3.2. Only a single test image is needed during the evaluation time, which is theoretically the more appropriate way of evaluation since our final prediction depends only on the features learned by the target feature extractor during the training phase.

Source Model	***#Source Params.***	***Target Model***	***#Target Params.***	***Optimizer***	***Learning Rate***
ResNet50	24,059,136	VGG-16	2,353,888	Adam	0.001

Table 5.1: Configuration of the conducted experiments

Following the proposed methodology, we conducted several experiments. In our first experiment, we performed DA from MNIST to MNIST-M dataset, where domain transfer took place on 10 digit classes. Here, Correlation alignment (CORAL) technique was used as Domain Invariance loss to bring the distributions of both the domains together. Different λ values were selected during the experiment. Kindly refer to tab.-5.2 for details of the experiment. We can observe that though the classification accuracy of the target domain as 62.2%, but by increasing the λ parameter, we were able to get better results.

In our second experimental setup, we transferred the domain knowledge from MNIST-M to MNIST dataset. Tab.-5.3 shows us that our test accuracy of 97.5% at $\lambda = 1$ was very high.

Source DS	*Target DS*	*Test DS*	*Test Accuracy*	*Training Accuracy*	*Additional Loss*
60K MNIST	60K MNIST-M	MNISTM 10K-10K	51%	99.5%	CORRELATION, $\lambda = 0.5$
60K MNIST	60K MNIST-M	MNISTM 10K-10K	62.2%	99.5%	CORRELATION, $\lambda = 1$

Table 5.2: Classification accuracy on transfer tasks from MNIST $\longrightarrow$ MNIST-M dataset

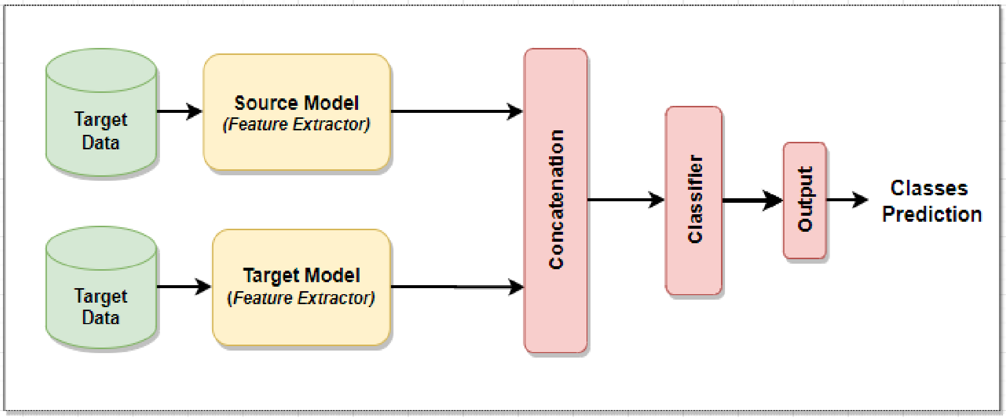

Figure 5.1: DDACN flowchart (old), Inference time

The results show that our experiments require hyperparameter tuning, which can give us better results in the future. In addition, the latest experimental setup will be used for evaluation phase, which can significantly improve the current results.

Source DS	*Target DS*	*Test DS*	*Test Accuracy*	*Training Accuracy*	*Additional Loss*
60K MNIST-M	60K MNIST	MNIST 10K-10K	97.5%	99.5%	CORRELATION, $\lambda = 1$

Table 5.3: Classification accuracy on transfer tasks from MNIST-M $\longrightarrow$ MNIST dataset

6 Schedule

The schedule is divided into four significant tasks, i.e., Research, Implementation (will be done using TensorFlow), Evaluation, and Documentation. Fig.-6.1 gives information about the amount of time required to finish each task.

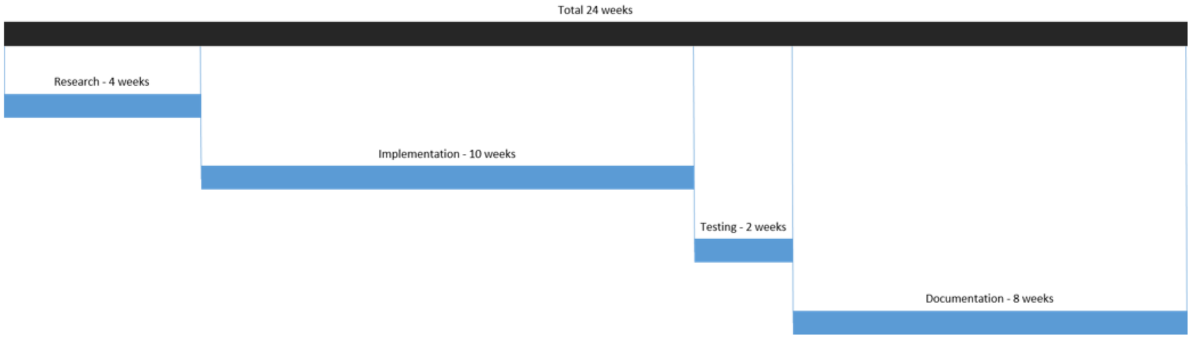

Figure 6.1: Thesis schedule

Bibliography

[Arb+10] Pablo Arbelaez et al. "Contour detection and hierarchical image segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 33.5 (2010), pp. 898–916 (cit. on p. 17).

[Ben+07] Shai Ben-David et al. "Analysis of representations for domain adaptation". In: *Advances in neural information processing systems*. 2007, pp. 137–144 (cit. on p. 8).

[Ben+10] Shai Ben-David et al. "A theory of learning from different domains". In: *Machine learning* 79.1-2 (2010), pp. 151–175 (cit. on p. 8).

[BRR16] Himanshu S Bhatt, Arun Rajkumar, and Shourya Roy. "Multi-Source Iterative Adaptation for Cross-Domain Classification." In: *IJCAI*. 2016, pp. 3691–3697 (cit. on p. 9).

[Cha+19] Woong-Gi Chang et al. "Domain-specific batch normalization for unsupervised domain adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7354–7362 (cit. on p. 8).

[Cha+12] Rita Chattopadhyay et al. "Multisource domain adaptation and its application to early detection of fatigue". In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 6.4 (2012), pp. 1–26 (cit. on p. 10).

[Csu17a] Gabriela Csurka. "Domain adaptation for visual applications: A comprehensive survey". In: *arXiv preprint arXiv:1702.05374* (2017) (cit. on p. 8).

[Csu17b] Gabriela Csurka. *Domain adaptation in computer vision applications*. Vol. 8. Springer, 2017 (cit. on p. 7).

[Den+09] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255 (cit. on p. 1).

[Den+20] Weijian Deng et al. “Rethinking triplet loss for domain adaptation”. In: *IEEE Transactions on Circuits and Systems for Video Technology* (2020) (cit. on p. 11).

[DKC10] Mark Dredze, Alex Kulesza, and Koby Crammer. “Multi-domain learning by confidence-weighted parameter combination”. In: *Machine Learning* 79.1-2 (2010), pp. 123–149 (cit. on p. 7).

[Dud19] Andrew Dudley. “Deep Domain Fusion for Adaptive Image Classification”. PhD thesis. Arizona State University, 2019 (cit. on p. 8).

[FXR13] Chen Fang, Ye Xu, and Daniel N Rockmore. “Unbiased metric learning: On the utilization of multiple datasets and web images for softening bias”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 1657–1664 (cit. on p. 10).

[Gan+16] Yaroslav Ganin et al. “Domain-adversarial training of neural networks”. In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 2096–2030 (cit. on pp. 4, 13, 17).

[Gho+20] Soumyadeep Ghosh et al. “Domain Adaptation for Visual Understanding”. In: *Domain Adaptation for Visual Understanding*. Springer, 2020, pp. 1–15 (cit. on pp. 8, 9).

[Goo+16] Ian Goodfellow et al. *Deep learning*. Vol. 1. MIT press Cambridge, 2016 (cit. on p. 7).

[Gre+09] Arthur Gretton et al. “Covariate shift by kernel mean matching”. In: *Dataset shift in machine learning* 3.4 (2009), p. 5 (cit. on p. 3).

[He+15] Kaiming He et al. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. 2015. arXiv: `1502.01852 [cs.CV]` (cit. on p. 2).

[He+16] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778 (cit. on p. 2).

[Kho+12] Aditya Khosla et al. “Undoing the damage of dataset bias”. In: *European Conference on Computer Vision*. Springer. 2012, pp. 158–171 (cit. on p. 10).

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*. 2012, pp. 1097–1105 (cit. on pp. 1, 2).

[KLB19] Jogendra Nath Kundu, Nishank Lakkakula, and R Venkatesh Babu. "Um-adapt: Unsupervised multi-task adaptation using adversarial cross-task distillation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 1436–1445 (cit. on p. 8).

[Kur+19] Seiichi Kuroki et al. "Unsupervised domain adaptation based on source-guided discrepancy". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 4122–4129 (cit. on p. 8).

[LeC98] Yann LeCun. "The MNIST database of handwritten digits". In: *http://yann. lecun. com/exdb/mnist/* (1998) (cit. on p. 17).

[LeC+89] Yann LeCun et al. "Backpropagation applied to handwritten zip code recognition". In: *Neural computation* 1.4 (1989), pp. 541–551 (cit. on p. 1).

[LeC+98] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cit. on p. 1).

[Liu+19] Ziwei Liu et al. "Compound domain adaptation in an open world". In: *arXiv preprint arXiv:1909.03403* (2019) (cit. on p. 10).

[Lon+16] Mingsheng Long et al. "Unsupervised domain adaptation with residual transfer networks". In: *Advances in neural information processing systems*. 2016, pp. 136–144 (cit. on pp. 8, 11).

[Luo+18] Zimeng Luo et al. "Deep unsupervised domain adaptation for face recognition". In: *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. IEEE. 2018, pp. 453–457 (cit. on p. 11).

[Mad+20] Yeganeh Madadi et al. "Deep Visual Unsupervised Domain Adaptation for Classification Tasks: A Survey". In: *Iet Image Processing* (2020) (cit. on pp. 11, 13).

[Man+19] Massimiliano Mancini et al. "Discovering Latent Domains for Unsupervised Domain Adaptation Through Consistency". In: *International Conference on Image Analysis and Processing*. Springer. 2019, pp. 390–401 (cit. on p. 11).

[MMR09] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. "Domain adaptation with multiple sources". In: *Advances in neural information processing systems*. 2009, pp. 1041–1048 (cit. on p. 10).

[Moi+13] Boris Moiseev et al. “Evaluation of traffic sign recognition methods trained on synthetically generated data”. In: *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer. 2013, pp. 576–583 (cit. on p. 17).

[Nat+18] Jogendra Nath Kundu et al. “Adadepth: Unsupervised content congruent adaptation for depth estimation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2656–2665 (cit. on p. 8).

[PY09] Sinno Jialin Pan and Qiang Yang. “A survey on transfer learning”. In: *IEEE Transactions on knowledge and data engineering* 22.10 (2009), pp. 1345–1359 (cit. on pp. 6–8).

[PG17] Pau Panareda Busto and Juergen Gall. “Open set domain adaptation”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 754–763 (cit. on p. 9).

[PS18] Xingchao Peng and Kate Saenko. “Synthetic to real adaptation with generative correlation alignment networks”. In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 1982–1991 (cit. on p. 11).

[Rah+20] Mohammad Mahfujur Rahman et al. “On minimum discrepancy estimation for deep domain adaptation”. In: *Domain Adaptation for Visual Understanding*. Springer, 2020, pp. 81–94 (cit. on p. 11).

[Rie+18] Matthew Riemer et al. “Learning to learn without forgetting by maximizing transfer and minimizing interference”. In: *arXiv preprint arXiv:1810.11910* (2018) (cit. on p. 9).

[RHW85] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985 (cit. on p. 1).

[Soh+18] Kihyuk Sohn et al. “Unsupervised domain adaptation for distance metric learning”. In: *International Conference on Learning Representations*. 2018 (cit. on p. 11).

[Sta+11] Johannes Stallkamp et al. “The German traffic sign recognition benchmark: a multi-class classification competition”. In: *The 2011 international joint conference on neural networks*. IEEE. 2011, pp. 1453–1460 (cit. on p. 17).

[SS16] Baochen Sun and Kate Saenko. "Deep coral: Correlation alignment for deep domain adaptation". In: *European conference on computer vision*. Springer. 2016, pp. 443–450 (cit. on pp. 3, 11).

[Sun+11] Qian Sun et al. "A two-stage weighting framework for multi-source domain adaptation". In: *Advances in neural information processing systems*. 2011, pp. 505–513 (cit. on p. 10).

[SSW15] Shiliang Sun, Honglei Shi, and Yuanbin Wu. "A survey of multi-source domain adaptation". In: *Information Fusion* 24 (2015), pp. 84–92 (cit. on p. 9).

[TE11] Antonio Torralba and Alexei A Efros. "Unbiased look at dataset bias". In: *CVPR 2011*. IEEE. 2011, pp. 1521–1528 (cit. on p. 3).

[Tze+14] Eric Tzeng et al. "Deep domain confusion: Maximizing for domain invariance". In: *arXiv preprint arXiv:1412.3474* (2014) (cit. on p. 11).

[Tze+17] Eric Tzeng et al. "Adversarial discriminative domain adaptation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7167–7176 (cit. on p. 8).

[Ven17] Hemanth Kumar Demakethepalli Venkateswara. "Domain Adaptive Computational Models for Computer Vision". PhD thesis. Arizona State University, 2017 (cit. on p. 6).

[WD18] Mei Wang and Weihong Deng. "Deep visual domain adaptation: A survey". In: *Neurocomputing* 312 (2018), pp. 135–153 (cit. on pp. 6, 8, 9).

[WKW16] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big data* 3.1 (2016), p. 9 (cit. on p. 7).

[WK96] Gerhard Widmer and Miroslav Kubat. "Learning in the presence of concept drift and hidden contexts". In: *Machine learning* 23.1 (1996), pp. 69–101 (cit. on p. 3).

[Xu+14] Zheng Xu et al. "Exploiting low-rank structure from latent domains for domain generalization". In: *European Conference on Computer Vision*. Springer. 2014, pp. 628–643 (cit. on p. 10).

[Yos+14] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems*. 2014, pp. 3320–3328 (cit. on p. 2).

[Zha+20] Yabin Zhang et al. “Unsupervised multi-class domain adaptation: Theory, algorithms, and practice”. In: *arXiv preprint arXiv:2002.08681* (2020) (cit. on p. 11).