AUT Journal of Mathematics and Computing

AUT J. Math. Comput., 3(2) (2022) 217-235
DOI: 10.22060/AJMC.2022.21221.1079

Review Article

A survey on self-supervised learning methods for domain adaptation in deep neural networks focusing on the optimization problems

GholamHassan Shirdel*[a], Alireza Ghanbari[a]

[a]*Department of Mathematics, Faculty of Sciences, University of Qom, Qom, Iran*

ABSTRACT: Deep convolutional neural networks have been widely and successfully used for various computer vision tasks. The main bottleneck for developing these models has been the lack of large datasets labeled by human experts. Self-supervised learning approaches have been used to deal with this challenge and allow developing models for domains with small labeled datasets. Another challenge for developing deep learning models is that their performance decreases when deployed on a target domain different from the source domain used for model training. Given a model trained on a source domain, domain adaptation refers to the methods used for adjusting a model or its output such that when the model is applied to a target domain, it achieves higher performance. This paper reviews the most commonly used self-supervised learning approaches and highlights their utility for domain adaptation.

Review History:
Received:15 March 2022
Accepted:15 August 2022
Available Online:01 September 2022

Keywords:

Domain adaptation
Self-supervised learning
Convolutional neural networks
Deep learning

AMS Subject Classification (2010):

68T45; 68U10; 90-02

(Dedicated to Professor S. Mehdi Tashakkori Hashemi)

1. Introduction

Convolutional neural networks (CNNs) have been incorporated in numerous computer vision applications including object detection [21, 65, 22] and semantic segmentation tasks [8, 48, 99]. One factor contributing to the widespread use of CNNs in computer vision tasks is their ability to represent different levels of general visual features. CNN models trained on large-scale datasets such as ImageNet are often used as pretrained models to be further fine-tuned for a given task. This happens for two major reasons: 1) the model parameters learned from large-scale datasets provide a favorable onset for continuing the learning of new tasks resulting in faster convergence and improving model performance; 2) such pretrained models already learned hierarchy of features that can be useful in alleviating the over-fitting problem. This is more pronounced in scenarios where large-scale annotated datasets are not available.

The number of available samples for training CNN models is a determining factor in the performance of these models. Currently, considering the complexity of problems, different architectures with more capacities are developed and larger datasets are gathered. Different networks such as AlexNet [39], VGG [76], GoogLeNet [77], ResNet [25], and DenseNet [30], and large-scale datasets—such as ImageNet [15] and OpenImage [40]—have been

*Corresponding author.
E-mail addresses: g.h.shirdel@qom.ac.ir, shirdel81math@gmail.com, arghanbari@gmail.com

introduced for training very deep CNN models. CNNs have shown a high performance in numerous computer vision tasks due to their effective architectures and the use of large-scale datasets for model developments [22, 48, 78, 43, 80].

Nevertheless, the process of collecting and annotating large-scale datasets for some tasks is time-consuming, costly, and in some cases, impossible. ImageNet [15], as one of the most frequently used datasets for pretraining 2D CNNs, has roughly 1.3 million labeled images that cover 1000 classes while every single picture has been manually labeled with one class label.

Due to the difficulty of manual annotation, self-supervised learning (SSL) methods were introduced to learn visual features from large-scale unlabeled datasets without incorporating humans for the annotation processes. The main purpose of SSL methods is to avoid manual annotation, which is tedious, time-consuming, and costly. A common solution for learning visual features from unlabeled data is defining and using pretext tasks. The pretext tasks are designed in a way that a labeling function (without requiring manual labeling) can be defined for generating a labeled dataset. A deep network can then be trained using such an algorithmically labeled dataset. The learned features by the network, resulting from training a model for solving the pretext task, could facilitate and accelerate learning other tasks when such models are fine-tuned on a dataset for the new task.

A variety of pretext tasks have been introduced for self-supervised learning. Among them are colorizing grayscale images [93], image inpainting [61], and image jigsaw puzzle solving [57]. The process is based on two principles: 1) the visual features learned by convolutional networks to solve the pretext tasks facilitate learning features that are suitable for learning the main task; 2) the labels of pretext tasks can be generated automatically and without any need for manual annotation.

A predefined pretext task is designed to be solved by a model. First, the labels are generated automatically for the pretext task. Then, the model is trained to learn the objective functions of the pretext task. When the self-supervised training is completed, the parameters of the learned model can be used as a pretrained model for learning other tasks. In general, the initial layers of a deep CNN represent low-level features such as textures, edges, and corners. On the other hand, the deeper layers represent high-level features related to the task the model has been trained for. Hence, the features from the initial layers can be used during the training phase of another task, and it is not necessary to learn these parameters again from the beginning.

Considering the importance of self-supervised learning, and reviewing the literature pertaining to this topic, the lack of a thorough and comprehensive survey dealing specifically with this important part of deep learning from the objective functions perspective was perceived. Hence, the researchers decided to perform such a survey and, present the results to fill the observed gap.

2. Background

In this section, we briefly introduce the technical terms used in this paper and present the formulation of different types of learning. Moreover we focus on object detection and semantic segmentation as well as domain adaptation.

Human-annotated label: Human-annotated labels are those labels of data that are manually annotated by an expert human.

Pretext task: Pretext tasks are pre-designed tasks aiming to be solved by a model. Although these tasks are different from the primary task of interest that we are trying to learn, they help a model learn useful visual feature representations for performing the primary task.

Downstream task: These tasks are computer vision applications that we are trying to train a model for learning them. Most often, human-annotated labels are required for solving the downstream tasks.

Active learning:In active learning [1, 75, 68], a limited set of unlabeled samples (e.g., unlabeled images) is chosen to be manually labeled. To minimize the difficulty of labeling in active learning, unlabeled samples that could potentially have a higher contribution to improving model performance are chosen to be manually annotated. These samples are then used to train the model further. In an iterative process, an initial model is used for selecting some unlabeled samples that labeling and using them as the training data has the maximum effect on improving the model performance. These data are labeled by human workers and added to the training dataset. Then, the learning process is applied to the model again, and the resulting model is used for selecting a new set of unlabeled samples. This process is continued until the model reaches the desired performance or all data points are annotated. This contrasts with passive learning, in which training data is randomly selected and ultimately requires more labeling costs.

Self-labeling: In self-labeling [88, 89, 101], a model is trained on the labeled data. Then this model is used for the automatic labeling of unlabeled data. In the next stage, the automatically labeled data is used for further model training. The key requirement for the successful use of this method is that systematic errors in the self-labeling process should not happen. Also, the initial model used for self-labeling should have reasonably high performance.

Transfer learning: In transfer learning [59, 22], a model is pretrained with a large number of training samples. However, this trained model cannot be directly used for our main task, for which we often have a small dataset.

Rather the aim of this pretrained model is to use it as a starting point for training the model on our main task to improve the model performance. Since both tasks require low-level features such as finding the edges and image texture, such features that are learned during training the model for the first task can be reused for the main task. Therefore, the model is not required to learn them from scratch. This process usually leads to better learning the main task, often with fewer labeled samples.

Machine Learning: Machine learning is a field that occupies computational algorithms to transform empirical data into applicable models. The machine learning field was originated from traditional statistics and artificial intelligence fields.

Based on the definition of machine learning, a model is considered as follows:

$$\begin{aligned} f_\theta : X &\longrightarrow Y \\ f_\theta(x_i) &= \hat{y}_i \end{aligned} \tag{1}$$

Where f_θ is a function with parameters θ that map an input x_i to an output $\hat{y}_i$.
According to the availability of labels for the data used for model development, machine learning can be divided into five categories: supervised, semi-supervised, weakly supervised, self-supervised learning, and unsupervised. In this section, we explain these approaches. Moreover, the key terms are clarified in order to facilitate the understanding of these approaches.
Supervised learning: In supervised learning, for each data point x_i in a dataset D, a corresponding label y_i is known *a priori*. Given a model f_θ, each input x_i can be mapped to an observed output $\hat{y}_i = f_\theta(x_i)$. The observed value of $\hat{y}_i$ might not be the same as the desired output y_i, available from the dataset X. A loss function is used to measure the discrepancy between the observed outputs and the desired outputs. For a dataset X consisted of n labeled data data points $D = \{(x_i, y_i) \mid x_i \in X; y_i \in Y; 1 \leq i \leq n\}$ is defined as follows:

$$loss(D) = \min_\theta \frac{1}{n} \sum_{i=1}^{n} loss(\hat{y}_i, y_i) \tag{2}$$

Semi-supervised learning: In semi-supervised learning we have two datasets $D = \{(x_i, y_i) \mid x_i \in X; y_i \in Y; 1 \leq i \leq n\}$ and $D' = \{z_i \mid z_i \in X; 1 \leq i \leq m\}$. The dataset D is often small, and for each data x_i in X, there is a corresponding (often human-annotated) label y_i. The dataset D' is often relatively larger, and for each data point z_i in X', there is no corresponding label available. The loss function used for model training is defined as follows:

$$loss(D, D') = \min_\theta \left(\frac{1}{n} \sum_{i=1}^{n} loss(f_\theta(x_i), y_i) + \frac{1}{m} \sum_{i=1}^{m} loss(f_\theta(z_i), R(z_i)) \right) \tag{3}$$

In which, $R(z_i)$ is a task-specific function. **Weakly-supervised learning**: In weakly-supervised visual feature learning, a dataset $D = \{(x_i, c_i) \mid x_i \in X; c_i \in Y; 1 \leq i \leq n\}$ is available. For each data point x_i in D, there is a noisy corresponding label c_i available. The loss function used for model training is defined as follows:

$$loss(D) = \min_\theta \frac{1}{n} \sum_{i=1}^{n} loss(f_\theta(x_i), c_i) \tag{4}$$

The supervisory signal c_i is noisy and weak; therefore, often large-scale datasets should be obtained. However, often developing such datasets is much easier. For example, different researches have suggested using hashtags as categories/labels for learning image features using images collected from the web [50, 44]. These approaches have shown promising performance [50]. **Unsupervised learning**: Unsupervised learning refers to methods aim at learning models without utilizing any human-annotated labels.

Self-supervised learning: Self-supervised learning [49, 35, 38] focuses on training a model without relying on manual labeling. More specifically, an auxiliary task, also known as pretext tasks, is used for model training. Image colorization [93], image rotation detection [38], and image inpainting [61], are among pretext tasks. These tasks are selected in a way that the desired output (label) corresponding to each image can be calculated computationally. The automatically labeled dataset is then used to train a model for the auxiliary task. After developing such models, the complete model or a part of it is used for developing a model for a downstream task, often with limited data. For example, the general feature extractors from a CNN model that is trained on an auxiliary task are used as feature extractors for another task. This compensates for fewer manually labeled samples for the main task.

Not relying on human-annotated labels in developing deep learning models for visual feature learning is currently of great importance. There are many instances of such researches [58, 74, 51, 66, 55, 5, 17, 84]. These approaches are considered a branch of unsupervised learning methods [38, 6, 27, 94, 4, 45, 16]. Unlike supervised learning methods that need a pair x_i and y_i where y_i is often annotated by human, in self-supervised learning for a data point x_i, a label p_i is generated algorithmically for a pre-defined pretext task with no human annotation. Using the features of the images including the context [93, 61, 57, 38] or incorporating the traditional methods which have been designed manually [73], we can generate the pseudo label p_i.

Having a set of n training data $D = \{(x_i, p_i) \mid x_i \in X; p_i \in P; 1 \leq i \leq n\}$, the training loss function is defined as follows:

$$loss(D) = \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} loss(f_\theta(x_i), p_i) \tag{5}$$

Object detection and semantic segmentation: Object detection and semantic segmentation are two examples of common optimization problems in computer vision. In object detection, the goal is finding a bounding box with a minimum area for each object in an image. In semantic segmentation, a label is allocated to each pixel of the image so that all pixels from one object have the same predetermined label. Here, the goal is to minimize the number of pixels with a wrong label. Different models have been proposed for these problems [21, 65, 24, 67, 64]. However, most of these methods use a large labeled datasets. Moreover, the resulting models have lower performance when applied to similar data from other domains. Many computer vision studies are focused on the problems in which the models for object detection and semantic segmentation are developed using data from a specific domain and used for that domain [21, 65, 24, 67, 64].

In object detection, a domain is defined as a pair (D, f) in which D is a set of data points and f is a labeled function as follows:

$$f : D \longrightarrow C \times R \tag{6}$$

In which C is the set $\{0, 1, \dots, k-1\}$, and k is the number of the considered objects, and R is a set of all ordered four tuples (x, y, w, h), which represent the coordinate of a bounding box for objects in an image. In semantic segmentation, R is a set of all matrices with the same size as the input images, and its elements belong to $\{0, 1, \dots, k-1\}$ in which k is the number of objects.

Domain adaptation: In domain adaptation [97, 82, 11, 29], a model is trained for performing a task in a source domain (e.g., semantic segmentation for simulated images). However, we need to apply the resulting model to perform the same task on a target domain, different from the source domain (e.g., semantic segmentation for real-world images).

As mentioned, the majority of the studies in the field of computer vision are based on the hypothesis that the domain distribution function is always the same. This hypothesis is rarely true for real-world problems. Therefore, the performance of the developed models in practice is often lower than the observed performance at their development time. In domain adaptation, this simplifying and mostly unrealistic hypothesis is not used. Instead, the realistic assumption that a model is developed by data from a source domain (D, f) and is used on a target domain (D', f'), is considered in which:

$$f' : D' \longrightarrow R' \tag{7}$$

Domain adaptation is performed by reusing the previous knowledge (as a model or labeled data) to maximize the model performance when applied to data from the target domain.

3. Common pretext tasks for self-supervised learning

Figure 1 illustrates the general schema of most of the available self-supervised learning approaches. In general, a pretext task should be defined to be learned by a deep learning model, and visual features can be learned through the process of performing the defined pretext task. Labels P for the pretext task are created automatically with no human annotation. By minimizing the error between the prediction of the model O and the labels P, the model can be optimized. In this way, the model can learn visual features for images when learning the pretext task.

In order to reduce the difficulty of annotating a large-scale dataset, generally, a pretext task is designed to be solved by the network, while labels for the pretext task are generated automatically. Many pretext tasks have been designed and used for self-supervised learning including motion-based [60], image inpainting [61], clustering [6], and image colorization [42]. Having an effective pretext task, semantic features can be learned through learning the pretext task.

Consider image colorization that is a task to colorize grayscale images into colorful ones. To accomplish this task, a model needs to learn the structural and contextual information of images to create realistic colorful images.

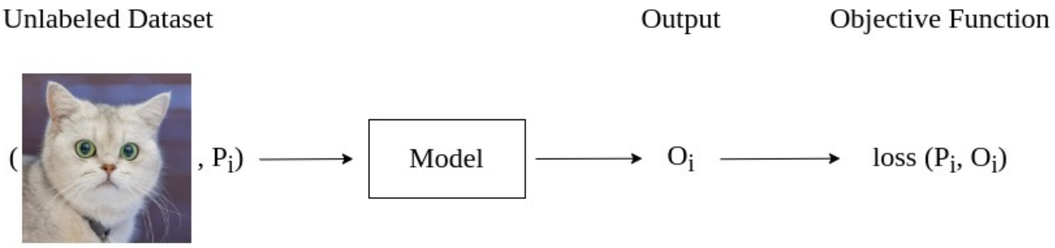

Figure 1: self-supervised visual feature learning schema. The convolutional network is trained through minimizing the error between the prediction of convolutional network O and the labels P. Labels P for the pretext task are created automatically with no human annotation.

In the mentioned pretext task, a data point x_i is a grayscale image that can be created from an RGB image through an automatic process, while the automatic label P_i is the RGB image itself. The training pair (x_i, P_i) can be generated quickly and with no cost.

According to the data attributes incorporated in pretext tasks, we summarize the most important pretext tasks into two main categories: methods based on image generation and methods based on image context. In methods based on image generation, visual features are learned through the process of image generation. These methods include image colorization [93], image resolution enhancement [43], image inpainting [61], and image generation using GAN networks [23, 100].

In methods based on image context, the design of the pretext mostly employ the contextual features of images including, context similarity, spatial structure, etc. Image clustering-based methods [58, 6] and spatial context-based methods are of this type. Spatial context-based methods are based on the spatial relations among image patches. These kinds of methods include, image jigsaw puzzle [57, 35, 85], context prediction, and geometric transformation prediction [38].

3.1. Generation-based feature learning

Self-supervised methods based on generation for learning image features include approaches such as image generation with GAN (for generating fake images), super-resolution (for generating high-resolution images), image inpainting (for predicting the missing parts of an image), and image colorization (for turning grayscale images into colorful ones). For these tasks, the training labels P are usually the images themselves, and there is no need for the human-annotated labels during the training process. As a result, these methods belong to self-supervised learning methods.

The pioneering work in image generation-based methods is the Autoencoder [26] that learns to encode an image and make a vector with low-dimension. Then this vector can be decoded and make the image again. To be able to represent the original image using the low-dimension vector, the vector should contain the original image information.

3.1.1. Image generation with inpainting

Image inpainting is a task involving the prediction of the missing parts of an image. Image inpainting is illustrated in figure 2. The image on the left is an image with a missing part, while the image on the right is a desirable prediction by a network. To predict the missing parts accurately, it is essential that the networks learn common knowledge, such as the structure and color of the common objects. By having this knowledge, inferring the missing parts based on other parts of the image is possible.

Using a deep learning mode, Pathak et al. predicted the missing content of a region of an image considering the other parts of the image [61]. To do so, they used a CNN by an adversarial loss function to perform the inpainting. Most of the latest methods follow a similar process [33]. Normally, there are two types of networks in these methods: a generator network that generates the missing part with the pixel-wise reconstruction loss and a discriminator network that recognizes if the input image is real. The discriminator network aims at generating more accurate and more realistic content for the missing part of the image, using an adversarial loss function.

A generator network which is a completely convolutional network is consisted of two parts: encoder and decoder. An image that needs to be inpainted is considered as the input for the encoder, and the context encoder learns the semantic feature of the image. The context decoder can predict the missing part based on this feature. The generator should perceive the image content to generate plausible content for the missing piece. The discriminator is trained to recognize if the input image is the generator output. In order to perform the image inpainting task, it is required that both networks learn the semantic features of images.

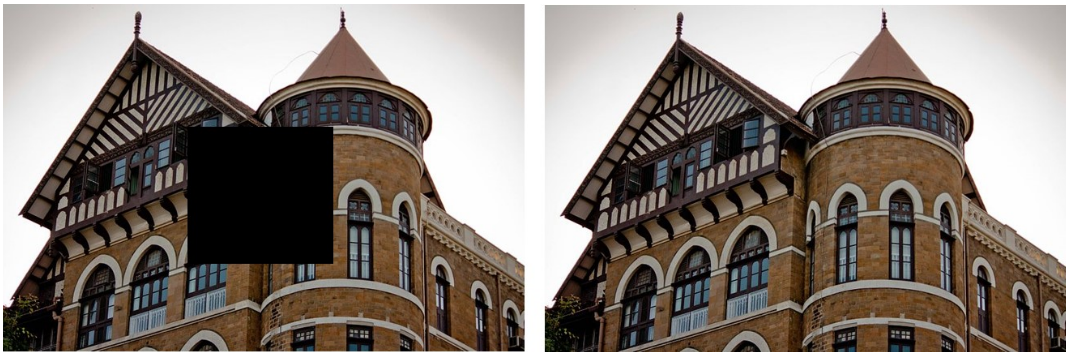

Figure 2: Image inpainting illustration. The right image is a sample image and the left image is an image that is obtained automatically by omitting a part of the image data. The left image is considered as the network input and the network tries to inpaint it so that the resulting image produced by the network is like the right image.

3.1.2. Image generation with super resolution

Image super-resolution is a task for increasing images resolution. Benefiting from the convolutional networks, better and more realistic images with high resolution can be produced from images with low resolutions [43]. Usually, two networks are involved in this process: a generator network that generates a high-resolution image using a low-resolution one. The other one is the discriminator network that distinguishes if the input image is the output of the generator or a real image with high resolution. The loss function for the generator can be L2 loss function which is calculated based on the pixel difference between the pixels of the real image and the image generated by the model. In addition, an auxiliary loss function can be used to compare the image content. This function approximately measures the similarity between the predicted high-resolution image and the real high-resolution image. A binary classification loss function is used for the discriminator network.

3.1.3. Image generation with colorization

Image colorization refers to a task in which a grayscale image is transformed into a colorful image. In order to colorize each pixel correctly, the networks need to recognize the objects and put the pixels of the same part in one category. Consequently, the visual features can be learned during the process of learning this task.

Many colorizing methods based on deep learning have been introduced [93, 32, 95]. These methods incorporate a fully convolutional neural network that includes an encoder for extracting the features and a decoder for retrieving the images and manifesting the color. L2 loss between the predicted color and the original one can help to optimize the network.

Some studies use the image colorization task specifically as the pretext for self-supervised image representation learning [93, 94, 42, 41]. The features learned using the colorization process are evaluated on downstream tasks using transfer learning.

3.2. Context-based image feature learning

The context-based pretext tasks mostly use image context features such as context similarity and spatial structure as the supervision signal. The convolutional network learns the features through solving the pretext tasks designed according to the attributes of the context images.

3.2.1. Learning with context similarity

Clustering is a method for grouping data points in which data points with similar characteristics are assigned to one cluster or group. Because of their potential in grouping data using the attributes of the data, clustering methods are extensively used in many fields, including machine learning, image processing, computer graphics, etc. [34].

In self-supervised methods, clustering methods are used as a means to cluster images. To do so, first, the images are clustered according to the features including HOG [14], SIFT [73], or Fisher Vector [73], so that the images from the same cluster have smaller distance, and images from different clusters have larger distance in feature space. The smaller the distance in feature space, the more similar the images are in the appearance in the RGB space [58, 6, 91, 87]. Then a model can be trained to categorize the data using the cluster assignment as the label. To do so, the model should learn the similarities between the images of a cluster and differentiate the images from different clusters [58, 6]. Hence, the model can learn the semantic meaning of images. These approaches often use KMeans algorithm to cluster features representing images.

3.2.2. Learning with spatial context structure

Images have a variety of information, including spatial context information such as the relative positions among different patches of an image that can be utilized to design pretext tasks for self-supervised learning. The pretext task can be predicting the relative positions of two patches from the same image [16] or recognizing the order of the shuffled sequence of patches from the same image [57, 35, 85]. Full image context can also be used as a supervision signal for designing the pretext tasks, including recognizing the rotating angles of the complete images [38]. Learning spatial context information such as objects shapes and the relative positions of different parts of them is necessary for a model to perform these pretext tasks.

Doersch et al. incorporated spatial context cues for self-supervised visual feature learning [16]. They extracted random pairs of image patches from each image; then a they trained a CNN for recognizing the relative positions of the two image patches. In order to solve this puzzle, the convolutional networks need to recognize objects in images and learn the relationship among different parts of objects. In order to avoid learning trivial solutions like using edges in the patches to perform the task, intense data augmentation is done during the training phase.

Many methods have been proposed for feature learning with more difficult spatial jigsaw puzzles [57, 35, 85]. As illustrated in figure refFig:3, an approach by Noroozi et al. aims at solving an image jigsaw puzzle using a convolutional network [57]. The left image is an image with 9 sampled image patches. The image in the middle is a sample of randomly placed image patches. The right image shows the correct order of the 9 image patches. The shuffled image patches are given to the network, which is trained to recognize the correct spatial positions of the input patches through learning spatial context structures of images, including object color, structure, and high-level semantic information.

Having 9 image patches from an image, there 362880 (9!) possible permutations, and due to the ambiguity of the task, it is very unlikely that a network recognizes all of them. In order to limit the number of permutations, normally, Hamming distance is used to select a subset of permutations with large relative Hamming distance. Only the chosen permutations are incorporated to train a convolutional network to be able to recognize the permutation of shuffled image patches [57, 35, 5, 85].

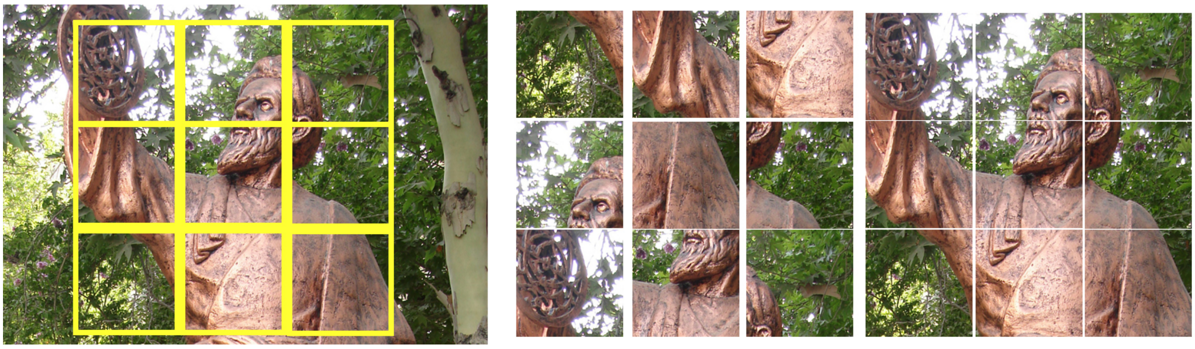

Figure 3: image jigsaw puzzle. The left image, is an image with 9 sampled image patches. The image in the middle, is an example of image patches with random order, and the right image illustrates the correct order of the sampled 9 image patches.

4. Self-supervised Learning for Domain adaptation

Domain adaptation methods for object detection were first proposed in [69]. These methods for performing semantic segmentation have attracted the attention of many researchers in this field [97, 29]. Among the available domain adaptation methods, some try to adapt the domains at the input level, including GAN-based methods [7, 46, 28] as well as image stylization methods [86, 18, 98]. Some other methods focus on feature-level adaptation [29, 71, 12, 31, 62], and others focus on output space adaptation [81, 52, 79, 72]. According to recent studies [82, 96], most methods are designed based on the principles of adversarial domain training [20]. In [101], iterative self-labeling is used to perform domain adaptation. In [97], a curriculum learning approach is used to match the distribution of super-pixels in the source and target domains. Matching these distributions occurs as an auxiliary task in the semantic segmentation training process. Using such an auxiliary task is inherently similar to the multitask learning approach in which the pretext tasks act as a connector between the source and target domains. In this study, we focus on using self-supervision learning methods for the domain adaptation problem.

After a brief overview of several studies related to Self-supervised Learning for Domain adaptation, we present a more comprehensive discussion of some of the most recent studies in this field. Here, our focus is mostly on the objective functions. Note that, like other machine learning problems, these studies exploited optimization problems. In some of them they explicitly mentioned their loss functions.

In the study [90] the authors proposed a generic method for self-supervised domain adaptation. In their method, they used object detection and semantic segmentation of urban scenes. They specifically worked on simple/auxiliary tasks and assessed a variety of learning strategies in order to enhance domain adaptation effectiveness. Moreover, they proposed two strategies to make domain adaptation more accurate including prediction layer alignment and batch normalization calibration. The favorable results showed that self-supervision could be a new alternative for acquiring domain adaptation. Putting forth an optimization problem, they defined the following loss function:

$$\mathscr{L} = \frac{1}{N_s}\sum_{i=1}^{N_s}\mathscr{L}_{seg}\left(x_i^s, \theta_e, \theta_s\right) + \frac{\lambda_p}{N_t}\sum_{i=1}^{N_t}\mathscr{L}_p\left(x_i^t, \theta_e, \theta_p\right) + \frac{\lambda_{adv}}{N_t}\sum_{i=1}^{N_t}\mathscr{L}_{adv}\left(x_i, \theta_e\right) + \frac{\lambda_d}{N_t+N_s}\sum_{i=1}^{N_t+N_s}\mathscr{L}_d\left(x_i, \theta_d\right) \tag{8}$$

The authors minimized this loss function as

$$\min_{\theta_e,\theta_p,\theta_s,\theta_d} \mathscr{L} \tag{9}$$

Besides using object detection and semantic segmentation, they offered a general method for self-supervised domain adaptation. The empirical results of this method show better performance compared to the other domain adaptation methods.

The loss function $\mathscr{L}$ is a combination of 4 loss functions $\mathscr{L}_{seg}, \mathscr{L}_p, \mathscr{L}_{adv}, \mathscr{L}_d$ in which $\mathscr{L}_{seg}$ is the semantic segmentation loss function which tries to penalize the results which are different from the desired segmentation. $\mathscr{L}_p$ is the loss function for the image rotation pretext task. In the minimization process, $\mathscr{L}_p$ learns the representation from the images based on recognizing image rotations. Although this representation is for detecting the image rotation, using it results in facilitating learning others tasks. $\mathscr{L}_{adv}$ is the adversarial loss function and is used for penalizing the model when the generated images using the decoder are different from the target domain. $\mathscr{L}_d$ is the domain discriminator loss function. In an ideal domain adaptation, there should be no difference between the source domain and the adapted target domain. $\mathscr{L}_d$ penalizes the model for having such differences. In equation 8, N_s is the number of labeled training samples from the source domain; N_t is the number of unlabeled samples of the target domain; x_i^s is the images of the source domain; x_i^t is the images of the target domain; x_i is an input image of the discriminator D; θ_e represents the parameters of the encoder E, θ_p represents the parameters of the pretext network P; θ_s represents the parameters of decoder S, and θ_d represents the parameters of the discriminator D. In addition, the importance weights of the loss functions $\mathscr{L}_{seg}$, $\mathscr{L}_p$, $\mathscr{L}_{adv}$, and $\mathscr{L}_d$ are model hyperparameters. Figure 4 illustrate various components of the model. In the article [53], it is shown that having a few target labels, self-supervision and consistency regularization which are considered as simple techniques, could be fruitful

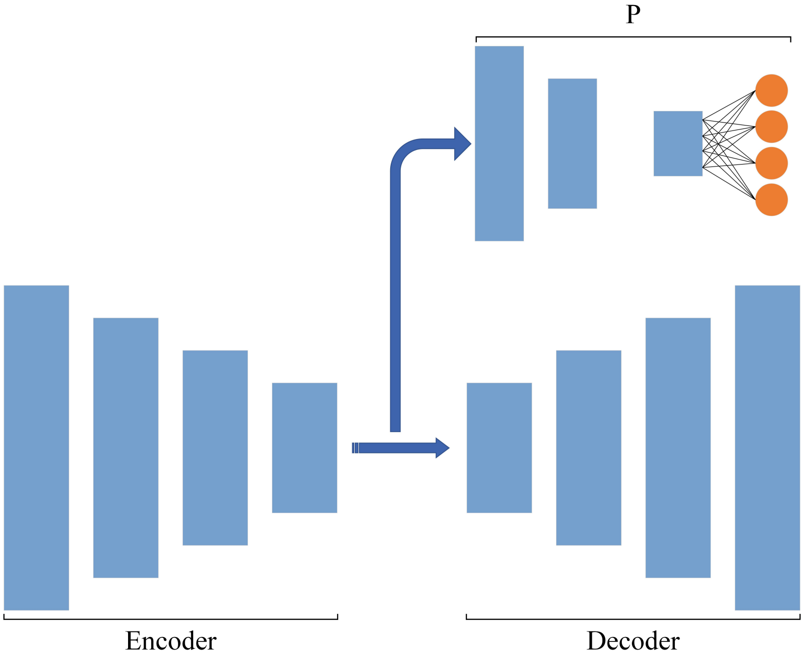

Figure 4: the schema of the proposed method in [90]

even without any adversarial alignment to learn an appropriate target classifier. Their Pretraining and Consistency (PAC) approach showed state of the art accuracy on semi-supervised domain adaptation task. In this article the domain adaptation has been defined as an optimization problem with the following loss function:

$$\mathscr{L}=\frac{1}{|M_s|}\sum_{(x,y)\in M_s}H\left(\overline{y},p\left(x\right)\right)+\frac{1}{|M_t|}\sum_{(x,y)\in M_t}H\left(\overline{y},p\left(x\right)\right)+\frac{1}{|M_u|}\sum_{x\in M_u}\mathscr{L}_{CR}\left(x\right) \tag{10}$$

This target function includes 3 elements in which, $\frac{1}{|M_s|}\sum_{(x,y)\in M_s}H\left(\overline{y},p\left(x\right)\right)$ is the cross-entropy function and calculated using the labeled data in the source domain. $\frac{1}{|M_t|}\sum_{(x,y)\in M_t}H\left(\overline{y},p\left(x\right)\right)$ is the cross-entropy function in which the ordered pairs (x,y) are obtained based on the image rotation pretext problem for unlabeled images of target domain. In addition, $\frac{1}{|M_u|}\sum_{x\in M_u}\mathscr{L}_{CR}\left(x\right)$ is the consistency regularization function and is used for data representations in a way that the created representation is robust against the small changes of the images. The main idea behind consistency regularization is that the representation of an image and its transformed version (like the addition of some noise) should be very similar.

M_s, M_t, and M_u represents the number of data points from D_s, D_t, and D_u, respectively.The used terms in the above-mentioned loss function include: M_s, M_t, and M_u which are data corresponding to D_s that is the source domain labeled data, D_t that is target domain labeled data, and D_u that is target unlabeled data, respectively. Moreover, H is the cross entropy loss function.

Saito et. al. [70], presented Domain Adaptive Neighborhood clustering via Entropy optimization (DANCE). Their suggested optimization yields promising results on universal domain adaptation. They introduced two supervision based parts: neighborhood clustering and entropy separation. DANCE was the only model that performed better than the source-only model. To fulfill the purposes of the study, the domain adaptation has been defined as an optimization problem with the following loss function:

$$\mathscr{L}=\mathscr{L}_{cls}+\lambda\left(\mathscr{L}_{nc}+\mathscr{L}_{es}\right) \tag{11}$$

In which a set of labeled data points from a source domain as well as labeled and unlabeled data from a target domain have been used. Two ideas of neighborhood clustering and entropy separation were used by the authors. The intermediate representations of the labeled and unlabeled data are saved in a memory bank. The target domain of the neighborhood clustering loss function is defined as follows:

$$\mathscr{L}_{nc}=-\frac{1}{|B_t|}\sum_{i\in B_t}\sum_{j=1,j\neq i}^{N_t+k}p_{i,j}log\left(p_{i,j}\right) \tag{12}$$

In which B_t is a batch of data points from the target domain. $p_{i,j}$ is the probability that the i^{th} data point of B_t be neighbor with the jth data point. The neighborhood clustering tries to make the representation of an input close to the other inputs of the same class.

The entropy separation is defined as follows:

$$\mathscr{L}_{es}=-\frac{1}{|B_t|}\sum_{i\in B_t}\mathscr{L}_{es}\left(p_i\right),\quad \mathscr{L}_{es}\left(p_i\right)=\begin{cases}-|H\left(p_i\right)-\rho| & ,|H\left(p_i\right)-\rho|>m\\ 0 & ,\text{otherwise}\end{cases} \tag{13}$$

In which, the constant $\rho=\frac{logK}{2}$ where K is the number of source classes. Moreover, H is the entropy function and p_i is the probabilities corresponding to the i^{th} input of B_t. $\mathscr{L}_{cls}$ indicates the cross-entropy loss on the source samples. Also, λ and m are hyperparameters for the model.

The main challenge in the domain adaptation problem is the need for large datasets. This problem is more significant in the fields where data labeling requires rare and expensive experts. Here, the focus is on proposing methods that perform domain adaptation using a limited set of labeled data.

In a novel study, Achituve et. al. [2] described self-supervised learning for domain adaptation on point clouds which was based on a multi-task architecture with a multi-head network. They presented a new family of pretext tasks called Deformation Reconstruction (DefRec). Furthermore, they introduce a new training procedure for labeled point cloud data inspired by the MixUp method called Point cloud Mixup (PCM). The analysis of domain adaptation datasets showed a significant improvement comparing to the common methods for classification and segmentation. To conduct their experiment, they used the following loss function:

$$L(S,T;\Phi,h_{\text{sup}},h_{SSL})=L_{ce}(S;\Phi,h_{\text{sup}})+\lambda L_{SSL}(\widehat{S\cup T};\Phi,h_{SSL}) \tag{14}$$

In the above equation, λ shows a hyperparameter that controls the importance of the self-supervised term, L_{ce} represents cross entropy-loss applied to the output of h_{sup} and the new label, and L_{SSL} is a self-supervised learning loss computed by Chamfer distance. Moreover, S represents labeled data from the source domain, T represents unlabeled data from the target domain, Φ is feature encoder, h_{sup} denotes fully connected sub-network (head) for supervised task, h_{SSL} shows another head for self-supervised task, and $\widehat{S \cup T} \subset X \times X$ in which X denotes input space.

In their study, Akada et. al. [3] offered a training strategy to force the task network to learn domain invariant representations in a self-supervised way. They concentrated on domain invariant representation learning that utilized images from two different domains using image-to-image translation network. Their proposed method outperformed other related methods. They utilized the loss function that follows:

$$L_{cossim} = \frac{1}{2} cossim\left(p_S, sg\left(z_{S\to T}\right)\right) + \frac{1}{2} cossim\left(p_{S\to T}, sg\left(z_S\right)\right) \tag{15}$$

In this equation, $sg(\cdot)$ denotes a stop gradient operation, $cossim(\cdot,\cdot)$ is a pixel-wise negative cosine similarity function, S represents labeled data from the source domain, T represents unlabeled data from the target domain, z_S is the output of the projector on the feature map of discriminator, $z_{S\to T}$ is the output of the projector on the generator, p_S is the output of the predictor on z_S , and $p_{S\to T}$ is the output of the predictor on $z_{S\to T}$.

Wang et. al. [83] they presented a new domain adaptation framework for semantic segmentation capable of leveraging the guidance from self-supervision of auxiliary task to bridge domain gaps, effectively. In order to be able to transfer the domain-shared knowledge to the target domain better, their proposed method learns the correlation between semantics and auxiliary tasks. Hence, they utilized a domain-shared task feature correlation module. Moreover, for refining their segmentation predictions, they used the adaptation difficulty. They believed that their proposed method, Correlation-Aware Domain Adaptation (CorDA), could be easily implemented into current segmentation frameworks. Combining their approach with an available self-training framework, they obtained remarkable performance. The loss function they used is as follows:

$$\mathcal{L} = \tilde{\mathcal{L}}^S_{seg} + \tilde{\mathcal{L}}^T_{seg} + \alpha^S \tilde{\mathcal{L}}^S_{depth} + \alpha^T \tilde{\mathcal{L}}^T_{depth} + \mathcal{L}^S_{seg} + \mathcal{L}^T_{seg} + \alpha^S \mathcal{L}^S_{depth} + \alpha^T \mathcal{L}^T_{depth} \tag{16}$$

In this function, α^S and α^T are the hyperparameters for the depth loss, $\tilde{\mathcal{L}}^S_{seg}$, $\tilde{\mathcal{L}}^T_{seg}$, $\tilde{\mathcal{L}}^S_{depth}$, $\tilde{\mathcal{L}}^T_{depth}$, $\mathcal{L}^S_{seg}$, $\mathcal{L}^T_{seg}$, $\mathcal{L}^S_{depth}$, and $\mathcal{L}^T_{depth}$ represent loss functions for the decoders of source initial semantic, target initial semantic, source initial depth, target initial depth, source shared semantic, target shared semantic, source depth, and target depth, respectively.

Investigating a new domain adaptation task with only few source labels available and numerous unlabeled source data, Kim et. al. [36] proposed a new Cross-Domain Self-supervised (CDS) learning which learns discriminative and domain-invariant features for domain adaptation. Their self-supervised learning method recognizes obvious visual similarity with in-domain self-supervision in a domain adaptive way and performs cross-domain feature matching with across-domain self-supervision. This method, drastically improved the performance of target accuracy in the new target domain having few source labels and proved to be useful on classical domain adaptation problem. In their study, they first conducted the pre-training stage with the overall objective of:

$$\mathcal{L}_{CDS} = \mathcal{L}_{W-INS} + \mathcal{L}_{CDM} \tag{17}$$

Here, $\mathcal{L}_{CDS}$ is loss function for Cross-domain Self-supervision, $\mathcal{L}_{W-INS}$ is the average of cross entropy losses of source and target domain in a batch, and $\mathcal{L}_{CDM}$ is the averaged entropy of the similarity distributions in a batch, which clusters source and target features and aligns their distributions. Later, they optimized the loss function:

$$\mathcal{L} = \mathcal{L}_{DA}\left(D_s, D_{tu}\right) + \lambda \mathcal{L}_{su}\left(D_{su}\right) \tag{18}$$

Where, D_s is (sparsely) source domain labeled data, D_su is source domain unlabeled data, D_tu is unlabeled target domain data, $\mathcal{L}_{DA}(\cdot,\cdot)$ is learning objective function of a domain adaptation method with labeled source and unlabeled target examples, $\mathcal{L}_{su}(\cdot)$ is a semi-supervised learning method (e.g., entropy minimization), and λ is a hyperparameter for controlling the importance of $\mathcal{L}_{DA}$ and $\mathcal{L}_{su}$.

Yue et. al. [92] studied Few-shot Unsupervised Domain Adaptation with only few labeled instances available in the source domain and no labeled instances in the target domain. They proposed an end-to-end Prototypical Cross-domain Self-Supervised Learning framework. Their offered approach improved the mean classification accuracy over divers domain pairs on Few-shot Unsupervised Domain Adaptation. They defined the loss function of Prototypical Cross-domain Self-supervised Learning (PCS) as follows:

$$\mathcal{L}_{PCS} = \mathcal{L}_{\text{cls}} + \lambda_{\text{in}} \mathcal{L}_{\text{InSelf}} + \lambda_{\text{cross}} \mathcal{L}_{\text{CrossSelf}} + \lambda_{\text{mim}} \mathcal{L}_{\text{MIM}} \tag{19}$$

In this loss function, $\mathcal{L}_{\text{cls}}$, $\mathcal{L}_{\text{InSelf}}$, $\mathcal{L}_{\text{CrossSelf}}$, and $\mathcal{L}_{\text{MIM}}$ denote standard cross-entropy loss for classification used to train discriminative feature encoder and classifier, in-domain self-supervision loss function, cross-domain instance-prototype SSL loss, and negative of mutual information loss between input and output, respectively. Also, λ_{in}, λ_{cross}, and λ_{mim} are the hyperparameters that control the importance of the mentioned loss functions.

In another study, Rao and Ni [63], proposed a self-supervised domain adaptation network consisting of a backbone network with Siamese architecture and a compression approximation network (ComNet) for the localization of JPEG-resistant image forgery. They generated JPEG-agent images using ComNet trained by self-supervised learning for the purpose of approximating the JPEG compression operation. JPEG-agent images generalizable attributes of JPEG compressions. These images help in reducing the domain shift between uncompressed and JPEG-agent images which results in a better robustness performance. They deployed the loss function which follows:

$$L = L_e + L_r + \alpha L_d \tag{20}$$

Where, L_e denotes edge loss for generating the attention map of forged boundary, L_r shows region loss for performing pixelwise classification, and L_d represents domain loss for reducing domain shift and transferring effective knowledge from source to target domain. Here, is a hyperparameter that maintains the balance between various tasks. In this equation, L_e and L_r are only computed within source domain.

Self-supervised contrastive regulations (SelfReg), is a new regulation method proposed by Kim et. al. [37] for domain generalization based on contrastive learning. It uses only positive data pairs, therefore, it can tackle different problems resulted from negative pair sampling. Their method showed promising results. Offering the following loss function, they conducted their study:

$$\mathcal{L} = \mathcal{L}_{\mathrm{c}} + \mathcal{L}_{\mathrm{SelfReg}} \tag{21}$$

In the presented equation, $\mathcal{L}_{\mathrm{c}}$ and $\mathcal{L}_{\mathrm{SelfReg}}$ denote classification loss and self-supervised contrastive loss, respectively.

A framework offered by Cheng et. al. [13] is a dual path learning (DPL), which helped to reduce visual inconsistency exploiting two complementary and interactive paths for domain adaptation of segmentation. In order to make two paths interactive, dual path image translation and dual path adaptive segmentation were presented. Their experiments showed that their dual path learning framework was superior comparing to the latest methods. To achieve their objectives, they defined two loss functions for Dual Path Image Translation and Dual Path Adaptive Segmentation. For the former one they defied the loss function as follows:

$$\mathcal{L}_{DualPer}\left(S, S', T, T'\right) = \mathcal{L}_{Per}\left(F_T\left(S'\right), F_S\left(S\right)\right) + \mathcal{L}_{Per}\left(F_T\left(T\right), F_S\left(T'\right)\right) \tag{22}$$

In the formulated equation, $\mathcal{L}_{Per}$ is perceptual loss while $F_T\left(\cdot\right)$ and $F_S\left(\cdot\right)$ are perceptual feature extracted by M_S and M_T, respectively. M_S is a semantic segmentation model in domain-S and M_T is a semantic segmentation model in domain-T. Moreover, S denotes source dataset (synthetic data), T target dataset (real data) with no labels, S' shows translated image of S, and T' represents translated image of T. The latter one is defined as:

$$\mathcal{L}_{DualSeg}\left(S, S', T, T'\right) = \mathcal{L}_{seg}^{T}\left(S', Y_S\right) + \mathcal{L}_{seg}^{T}\left(T, \hat{Y}_*\right) + \mathcal{L}_{seg}^{S}\left(S, Y_S\right) + \mathcal{L}_{seg}^{S}\left(T', \hat{Y}_*\right) + \lambda_{adv}\left(\mathcal{L}_{adv}^{T}\left(S', T\right) + \mathcal{L}_{adv}^{S}\left(S, T'\right)\right) \tag{23}$$

In this equation, $\mathcal{L}_{adv}^{S}$ and $\mathcal{L}_{adv}^{T}$ represent typical adversarial loss, $\mathcal{L}_{seg}^{S}$ and $\mathcal{L}_{seg}^{T}$ denote per-pixel segmentation loss, and λ_{adv} is in charge of controlling the contribution of adversarial loss. In addition, Y_S is pixel-level segmentation labels of S, and $\hat{Y}_*$ represents pseudo labels of target images.

In their study, Mitsuzumi et. al. [54] proposed Generalized Domain Adaptation (GDA) which is a general representation of unsupervised domain adaptation (UDA) problems. These problems include all types of UDA and the new settings where current UDA methods fail. They also proposed a novel self-supervised class-destructive learning approach that estimates domain labels independently. Their method outperformed the existing methods. Their network was consisted of three main components: a shared feature extractor G_f, a class label predictor F_y, and a domain classifier F_d. Solving the following problems, they conducted their study:

$$\begin{aligned} &\min_{G_f, F_y} \mathcal{L}_y - \lambda \mathcal{L}_d, \\ &\min_{F_d} \mathcal{L}_d \end{aligned} \tag{24}$$

In the equation, $\mathcal{L}_y$ is a class classification loss and $\mathcal{L}_d$ is a domain classification loss. The two problems can be properly minimized at the same time.

Chen et. al. [9] addressed the problem of spatio-temporal variation in fully-supervised action segmentation techniques. Performing this task, they utilized unlabeled videos and reformulated the action segmentation task as a cross-domain problem. This was done with domain discrepancy resulted from spatio-temporal variations. They proposed Self-Supervised Temporal Domain Adaptation (SSTDA) consisted of two Self-supervised auxiliary tasks to align cross-domain feature spaces with local and global temporal dynamics. SSTDA showed significant improvements over current methods.

$$\mathcal{L} = \sum^{N_s} \mathcal{L}_y - \sum^{\widetilde{N_s}} \left(\beta_1 \mathcal{L}_{ld} + \beta_g \mathcal{L}_{gd} + \mu \mathcal{L}_{ae}\right) \tag{25}$$

They formulated the above equation where, $\mathcal{L}_y$ denotes baseline prediction loss, $\mathcal{L}_{ld}$ is a binary cross-entropy loss function for local domain, $\mathcal{L}_{gd}$ shows the global domain loss, $\mathcal{L}_{ae}$ is domain attentive entropy loss, N_s is the total stage number in multi-stage temporal convolution networks, and $\widetilde{N_s}$ is the number of stages integrated with shallow binary domain classifier which is equipped with a gradient reversal layer. Moreover, β_1, β_g, and μ are hyperparameters for controlling the importance of the mentioned loss functions.

In their paper, Lin et. al. [47], used self-supervised learning to learn a more appropriate embedding space where the subjects in target domain could be distinguished better. They aimed at enhancing the similarity between the embeddings of each image and its mirror. By adding an adapting ratio between the self-similarity losses on source and target domain, Self-Supervised Adapting (SSA) loss was proposed. In this regard, they achieved promising results. The following loss function was introduced:

$$L = L_c + L_a \tag{26}$$

In the above loss function, L_c is a the cross-entropy between source domain dataset labels and the labels of embedding function that maps images to a lower dimension space, L_a is the convex combination of SimSiam [10] self-supervised learning loss on source and target domain images.

Fujii et. al. [19] proposed a generative and self-supervised domain adaptation method for one-stage detectors performing object detection consisted of an adversarial generative method and a method based on self-supervision. Their method showed improvements in domain adaptation performance. The following equation is the loss function that they used in their study:

$$L(b,c,l,g) = \frac{1}{N}\left(L_{conf}(b,c) + \alpha L_{loc}(b,l,g)\right) \tag{27}$$

Where, b denotes a matched default box, c shows the confidence of multiple classes, l denotes a predicted box, g is a ground-truth box, N is the number of matched default boxes, α shows the weight, L_{conf} denotes the confidence loss, and L_{loc} denotes the localization loss.

In [56], Najafian et. al. proposed a semi-self-supervised learning approach for the wheat head detection problem. For training the deep convolutional neural network, the authors used video files taken from wheat fields. The authors trained a deep CNN using simulated datasets. Figure 5 illustrates the procedure for data simulation, where wheat heads were extracted from an image and automatically overlaid on background images. Then they applied two domain adaptation steps for improving model performance. They used an implementation of YOLOV4 [64] for object detection. The trained model achieved promising results using very little human annotation.

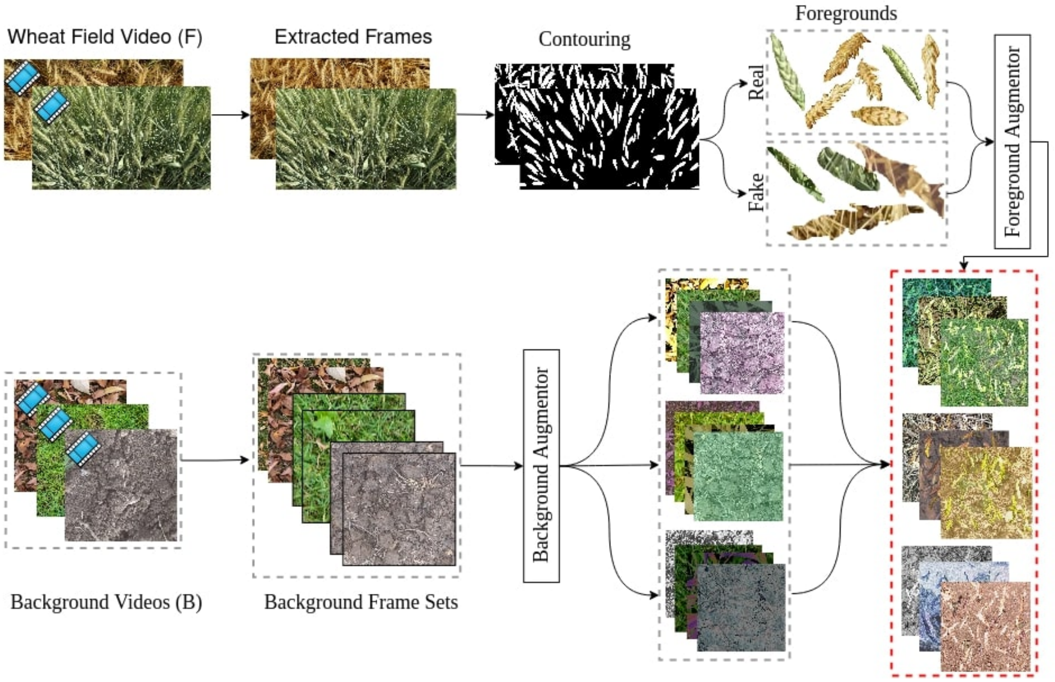

Figure 5: The simulation of the training samples in reference [56].

In order to facilitate the comparison between the above-mentioned articles, we provided Table 1 in the Supplemental Material.

5. Concluding Remarks

In this paper, we provided a general overview of self-supervised learning methods and highlighted their utility for domain adaptation. Considering the bottleneck of manual labeling, these approaches have shown to be effective in learning visual feature representations for domains with no manual annotations.

Deep neural networks can be applied to a wide range of tasks and have shown promising performance for domains where large-scale annotated datasets are available. In addition, utilizing models trained on a large-scale dataset like ImageNet has been shown to reduce the negative impact of the lack of large-scale annotated datasets. Using such pre-trained models to learn a new task, the model requires fewer annotated training samples from the new task to achieve desirable performance. This is due to the fact that the pre-trained model has already learned a hierarchy of features that could be reused for learning the new task instead of learning them from scratch.

However, the need for a highly accurate and large-scale annotated dataset is still a critical bottleneck in developing deep neural networks. Providing such datasets is expensive, time-consuming, and in some areas impossible. Self-supervised learning methods make it possible to learn visual features without the need for a manual annotation.

In self-supervised learning methods, a pretext task is designed so that data annotation can be derived algorithmically without a manual process. Using this algorithmically annotated dataset, a model is trained to learn the pretext task. The trained model using a self-supervised approach can be used as a pre-trained model for learning other tasks.

Domain adaptation is another discussed topic that helps us to apply the pre-trained model on the source dataset domain to the target domain that is different from the source data distribution. The main goal of this field is to develop a model that has high functionality and performance on the target domains.

Besides being used for learning visual feature representations, self-supervised learning has shown potential for domain adaptation. We highlighted several approaches that utilized self-supervision for domain adaptation. Considering the increase in the amount of unlabeled data and the challenge of manual labeling, self-supervised learning could be an effective tool for domain adaptation. Despite the early works in this area, the optimal use of these approaches for domain adaptation still remains an active research area. For instance, other researchers can focus on conducting surveys on unsupervised and semi-supervised learning for domain adaptation focusing on the optimization problems.

References

[1] Y. Abramson and Y. Freund, *Active learning for visual object recognition*, in Technical report, UCSD, 2004.

[2] I. Achituve, H. Maron, and G. Chechik, *Self-supervised learning for domain adaptation on point clouds*, in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), January 2021, pp. 123–133.

[3] H. Akada, S. F. Bhat, I. Alhashim, and P. Wonka, *Self-supervised learning of domain invariant features for depth estimation*, in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), January 2022, pp. 3377–3387.

[4] P. Bojanowski and A. Joulin, *Unsupervised learning by predicting noise*, in Proceedings of the 34th International Conference on Machine Learning, D. Precup and Y. W. Teh, eds., vol. 70 of Proceedings of Machine Learning Research, PMLR, 06–11 Aug 2017, pp. 517–526.

[5] U. Buchler, B. Brattoli, and B. Ommer, *Improving spatiotemporal self-supervision by deep reinforcement learning*, in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 770–786.

[6] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, *Deep clustering for unsupervised learning of visual features*, in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 132–149.

[7] W.-L. Chang, H.-P. Wang, W.-H. Peng, and W.-C. Chiu, *All about structure: Adapting structural information across domains for boosting semantic segmentation*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1900–1909.

[8] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 40 (2018), pp. 834–848.

[9] M.-H. Chen, B. Li, Y. Bao, G. AlRegib, and Z. Kira, *Action segmentation with joint self-supervised temporal domain adaptation*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[10] X. Chen and K. He, *Exploring simple siamese representation learning*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021, pp. 15750–15758.

[11] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, *Domain adaptive faster r-cnn for object detection in the wild*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3339–3348.

[12] Y.-C. Chen, Y.-Y. Lin, M.-H. Yang, and J.-B. Huang, *Crdoco: Pixel-level domain transfer with cross-domain consistency*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1791–1800.

[13] Y. Cheng, F. Wei, J. Bao, D. Chen, F. Wen, and W. Zhang, *Dual path learning for domain adaptation of semantic segmentation*, in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 9082–9091.

[14] N. Dalal and B. Triggs, *Histograms of oriented gradients for human detection*, in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, 2005, pp. 886–893 vol. 1.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, *Imagenet: A large-scale hierarchical image database*, in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[16] C. Doersch, A. Gupta, and A. A. Efros, *Unsupervised visual representation learning by context prediction*, in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1422–1430.

[17] C. Doersch and A. Zisserman, *Multi-task self-supervised visual learning*, in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2051–2060.

[18] A. Dundar, M.-Y. Liu, T.-C. Wang, J. Zedlewski, and J. Kautz, *Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation*, arXiv preprint arXiv:1807.09384, (2018).

[19] K. Fujii and K. Kawamoto, *Generative and self-supervised domain adaptation for one-stage object detection*, Array, 11 (2021), p. 100071.

[20] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, *Domain-adversarial training of neural networks*, The journal of machine learning research, 17 (2016), pp. 2096–2030.

[21] R. Girshick, *Fast r-cnn*, in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial nets*, in Advances in Neural Information Processing Systems, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, eds., vol. 27, Curran Associates, Inc., 2014.

[24] K. He, G. Gkioxari, P. Dollar, and R. Girshick, *Mask r-cnn*, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[25] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[26] G. E. Hinton and R. R. Salakhutdinov, *Reducing the dimensionality of data with neural networks*, Science, 313 (2006), pp. 504–507.

[27] E. Hoffer, I. Hubara, and N. Ailon, *Deep unsupervised learning through spatial contrasting*, arXiv preprint arXiv:1610.00243, (2016).

[28] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, *Cycada: Cycle-consistent adversarial domain adaptation*, in International conference on machine learning, PMLR, 2018, pp. 1989–1998.

[29] J. Hoffman, D. Wang, F. Yu, and T. Darrell, *Fcns in the wild: Pixel-level adversarial and constraint-based adaptation*, arXiv preprint arXiv:1612.02649, (2016).

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, *Densely connected convolutional networks*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

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

[32] S. Iizuka, E. Simo-Serra, and H. Ishikawa, *Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification*, 35 (2016).

[33] ——, *Globally and locally consistent image completion*, ACM Transactions on Graphics (ToG), 36 (2017), pp. 1–14.

[34] A. K. Jain, M. N. Murty, and P. J. Flynn, *Data clustering: A review*, ACM Comput. Surv., 31 (1999), p. 264–323.

[35] D. Kim, D. Cho, D. Yoo, and I. S. Kweon, *Learning image representations by completing damaged jigsaw puzzles*, in 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 793–802.

[36] D. Kim, K. Saito, T.-H. Oh, B. A. Plummer, S. Sclaroff, and K. Saenko, *Cross-domain self-supervised learning for domain adaptation with few source labels*, arXiv preprint arXiv:2003.08264, (2020).

[37] D. Kim, Y. Yoo, S. Park, J. Kim, and J. Lee, *Selfreg: Self-supervised contrastive regularization for domain generalization*, in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 9619–9628.

[38] N. Komodakis and S. Gidaris, *Unsupervised representation learning by predicting image rotations*, in International Conference on Learning Representations (ICLR), 2018.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, Advances in neural information processing systems, 25 (2012).

[40] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, et al., *The open images dataset v4*, International Journal of Computer Vision, 128 (2020), pp. 1956–1981.

[41] G. Larsson, M. Maire, and G. Shakhnarovich, *Learning representations for automatic colorization*, in Computer Vision – ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, eds., Cham, 2016, Springer International Publishing, pp. 577–593.

[42] ——, *Colorization as a proxy task for visual understanding*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 6874–6883.

[43] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., *Photo-realistic single image super-resolution using a generative adversarial network*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.

[44] W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool, *Webvision database: Visual learning and understanding from web data*, arXiv preprint arXiv:1708.02862, (2017).

[45] Y. Li, M. Paluri, J. M. Rehg, and P. Dollár, *Unsupervised learning of edges*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1619–1627.

[46] Y. Li, L. Yuan, and N. Vasconcelos, *Bidirectional learning for domain adaptation of semantic segmentation*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6936–6945.

[47] C.-H. Lin and B.-F. Wu, *Domain adapting ability of self-supervised learning for face recognition*, in 2021 IEEE International Conference on Image Processing (ICIP), 2021, pp. 479–483.

[48] J. Long, E. Shelhamer, and T. Darrell, *Fully convolutional networks for semantic segmentation*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015.

[49] M. Long, H. Zhu, J. Wang, and M. I. Jordan, *Deep transfer learning with joint adaptation networks*, in International conference on machine learning, PMLR, 2017, pp. 2208–2217.

[50] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. Van Der Maaten, *Exploring the limits of weakly supervised pretraining*, in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 181–196.

[51] A. Mahendran, J. Thewlis, and A. Vedaldi, *Cross pixel optical-flow similarity for self-supervised learning*, in Asian Conference on Computer Vision, Springer, 2018, pp. 99–116.

[52] J. Manders, T. van Laarhoven, and E. Marchiori, *Adversarial alignment of class prediction uncertainties for domain adaptation*, arXiv preprint arXiv:1804.04448, (2018).

[53] S. Mishra, K. Saenko, and V. Saligrama, *Surprisingly simple semi-supervised domain adaptation with pretraining and consistency*, arXiv e-prints, (2021), pp. arXiv–2101.

[54] Y. Mitsuzumi, G. Irie, D. Ikami, and T. Shibata, *Generalized domain adaptation*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021, pp. 1084–1093.

[55] T. N. Mundhenk, D. Ho, and B. Y. Chen, *Improvements to context based self-supervised learning*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9339–9348.

[56] K. Najafian, A. Ghanbari, I. Stavness, L. Jin, G. Hassan Shirdel, and F. Maleki, *A semi-self-supervised learning approach for wheat head detection using extremely small number of labeled samples*, in 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), 2021, pp. 1342–1351.

[57] M. Noroozi and P. Favaro, *Unsupervised learning of visual representations by solving jigsaw puzzles*, in European conference on computer vision, Springer, 2016, pp. 69–84.

[58] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, *Boosting self-supervised learning via knowledge transfer*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 9359–9367.

[59] S. J. Pan and Q. Yang, *A survey on transfer learning. ieee transactions on knowledge and data engineering*, 22 (10): 1345, 1359 (2010).

[60] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan, *Learning features by watching objects move*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2701–2710.

[61] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, *Context encoders: Feature learning by inpainting*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2536–2544.

[62] Z. Pei, Z. Cao, M. Long, and J. Wang, *Multi-adversarial domain adaptation*, in Thirty-second AAAI conference on artificial intelligence, 2018.

[63] Y. Rao and J. Ni, *Self-supervised domain adaptation for forgery localization of jpeg compressed images*, in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 15034–15043.

[64] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[65] S. Ren, K. He, R. Girshick, and J. Sun, *Faster r-cnn: Towards real-time object detection with region proposal networks*, Advances in neural information processing systems, 28 (2015), pp. 91–99.

[66] Z. Ren and Y. J. Lee, *Cross-domain self-supervised multi-task feature learning using synthetic imagery*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 762–771.

[67] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, eds., Cham, 2015, Springer International Publishing, pp. 234–241.

[68] S. Roy, A. Unmesh, and V. P. Namboodiri, *Deep active learning for object detection.*

[69] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, *Adapting visual category models to new domains*, in European conference on computer vision, Springer, 2010, pp. 213–226.

[70] K. Saito, D. Kim, S. Sclaroff, and K. Saenko, *Universal domain adaptation through self supervision*, Advances in Neural Information Processing Systems, 33 (2020).

[71] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, *Adversarial dropout regularization*, arXiv preprint arXiv:1711.01575, (2017).

[72] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, *Maximum classifier discrepancy for unsupervised domain adaptation*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3723–3732.

[73] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, *Image classification with the fisher vector: Theory and practice*, International journal of computer vision, 105 (2013), pp. 222–245.

[74] N. Sayed, B. Brattoli, and B. Ommer, *Cross and learn: Cross-modal self-supervision*, in German Conference on Pattern Recognition, Springer, 2018, pp. 228–243.

[75] B. Settles, *Active learning: Synthesis lectures on artificial intelligence and machine learning*, Long Island, NY: Morgan & Clay Pool, (2012).

[76] K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, arXiv preprint arXiv:1409.1556, (2014).

[77] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, *Going deeper with convolutions*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[78] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, *Learning spatiotemporal features with 3d convolutional networks*, in Proceedings of the IEEE international conference on computer vision, 2015, pp. 4489–4497.

[79] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, *Learning to adapt structured output space for semantic segmentation*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7472–7481.

[80] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, *Show and tell: A neural image caption generator*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3156–3164.

[81] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, *Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2517–2526.

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

[83] Q. Wang, D. Dai, L. Hoyer, L. Van Gool, and O. Fink, *Domain adaptive semantic segmentation with self-supervised depth estimation*, in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 8515–8525.

[84] X. Wang, K. He, and A. Gupta, *Transitive invariance for self-supervised visual representation learning*, in Proceedings of the IEEE international conference on computer vision, 2017, pp. 1329–1338.

[85] C. Wei, L. Xie, X. Ren, Y. Xia, C. Su, J. Liu, Q. Tian, and A. L. Yuille, *Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1910–1919.

[86] Z. Wu, X. Han, Y.-L. Lin, M. G. Uzunbas, T. Goldstein, S. N. Lim, and L. S. Davis, *Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation*, in European Conference on Computer Vision, Springer, 2018, pp. 535–552.

[87] J. Xie, R. Girshick, and A. Farhadi, *Unsupervised deep embedding for clustering analysis*, in Proceedings of The 33rd International Conference on Machine Learning, M. F. Balcan and K. Q. Weinberger, eds., vol. 48 of Proceedings of Machine Learning Research, New York, New York, USA, 20–22 Jun 2016, PMLR, pp. 478–487.

[88] J. Xu, S. Ramos, D. Vázquez, and A. M. López, *Domain adaptation of deformable part-based models*, IEEE transactions on pattern analysis and machine intelligence, 36 (2014), pp. 2367–2380.

[89] J. Xu, D. Vázquez, K. Mikolajczyk, and A. M. López, *Hierarchical online domain adaptation of deformable part-based models*, in 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2016, pp. 5536–5541.

[90] J. Xu, L. Xiao, and A. M. López, *Self-supervised domain adaptation for computer vision tasks*, IEEE Access, 7 (2019), pp. 156694–156706.

[91] J. Yang, D. Parikh, and D. Batra, *Joint unsupervised learning of deep representations and image clusters*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5147–5156.

[92] X. Yue, Z. Zheng, S. Zhang, Y. Gao, T. Darrell, K. Keutzer, and A. S. Vincentelli, *Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation*, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021, pp. 13834–13844.

[93] R. Zhang, P. Isola, and A. A. Efros, *Colorful image colorization*, in Computer Vision – ECCV 2016, B. Leibe, J. Matas, N. Sebe, and M. Welling, eds., Cham, 2016, Springer International Publishing, pp. 649–666.

[94] R. Zhang, P. Isola, and A. A. Efros, *Split-brain autoencoders: Unsupervised learning by cross-channel prediction*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1058–1067.

[95] R. Zhang, J.-Y. Zhu, P. Isola, X. Geng, A. S. Lin, T. Yu, and A. A. Efros, *Real-time user-guided image colorization with learned deep priors*, ACM Trans. Graph., 36 (2017).

[96] Y. Zhang, P. David, H. Foroosh, and B. Gong, *A curriculum domain adaptation approach to the semantic segmentation of urban scenes*, IEEE transactions on pattern analysis and machine intelligence, 42 (2019), pp. 1823–1841.

[97] Y. Zhang, P. David, and B. Gong, *Curriculum domain adaptation for semantic segmentation of urban scenes*, in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2020–2030.

[98] Y. Zhang, Z. Qiu, T. Yao, D. Liu, and T. Mei, *Fully convolutional adaptation networks for semantic segmentation*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6810–6818.

[99] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, *Pyramid scene parsing network*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[100] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

[101] Y. Zou, Z. Yu, B. V. K. Vijaya Kumar, and J. Wang, *Unsupervised domain adaptation for semantic segmentation via class-balanced self-training*, in Computer Vision – ECCV 2018, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds., Cham, 2018, Springer International Publishing, pp. 297–313.

Please cite this article using:

GholamHassan Shirdel, Alireza Ghanbari, A survey on self-supervised learning methods for domain adaptation in deep neural networks focusing on the optimization problems, AUT J. Math. Comput., 3(2) (2022) 217-235
DOI: 10.22060/AJMC.2022.21221.1079