Knowledge Transfer in Vision Recognition: A Survey

Ying Lu, Lingkun Luo, Di Huang, Yunhong Wang, Liming Chen

Knowledge Transfer in Vision Recognition: A Survey

YING LU, Ecole Centrale de Lyon, France
LINGKUN LUO, Shanghai Jiao Tong University, China
DI HUANG, Beihang University, China
YUNHONG WANG, Beihang University, China
LIMING CHEN, Ecole Centrale de Lyon, France

In this survey, we propose to explore and discuss the common rules behind knowledge transfer works for vision recognition tasks. To achieve this, we firstly discuss the different kinds of reusable knowledge existing in a vision recognition task, and then we categorize different knowledge transfer approaches depending on where the knowledge comes from and where the knowledge goes. Compared to previous surveys on knowledge transfer that are from the problem-oriented perspective or from the technique-oriented perspective, our viewpoint is closer to the nature of knowledge transfer and reveals common rules behind different transfer learning settings and applications. Besides different knowledge transfer categories, we also show some research works that study the transferability between different vision recognition tasks. We further give a discussion about the introduced research works and show some potential research directions in this field.

CCS Concepts: • **Computing methodologies** → **Transfer learning**.

Additional Key Words and Phrases: knowledge transfer, transfer learning, vision recognition, computer vision, machine learning

ACM Reference Format:
Ying Lu, Lingkun Luo, Di Huang, Yunhong Wang, and Liming Chen. 2019. Knowledge Transfer in Vision Recognition: A Survey. 1, 1 (December 2019), 35 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Vision recognition is a core problem in computer vision. Its goal generally is to determine whether or not the input visual data contains some specific concept, object, or activity and to give corresponding predictive output about the recognized content. For example, in image classification tasks, the goal is to determine which pre-defined image class/concept (*e.g.* 'airport', 'castle', *etc.*) the input image belongs to; in object detection tasks, the goal is to determine whether the input image (or 3D data) contains some specific object (*e.g.* 'bicycle', 'dog', 'mug' *etc.*) and to output the corresponding bounding box (which defines the minimum rectangular/cubic area that contains the object) if an object exists; in semantic segmentation tasks, the goal is to predict semantic label for each pixel or super pixel in an input image. The common schema of this kind of tasks is that they all take some visual data as input and output some predictive information based on the input. Therefore the classical way to solve this kind of tasks is supervised learning-based.

Authors' addresses: Ying Lu, ying.lu@ec-lyon.fr, Ecole Centrale de Lyon, Ecully, France; Lingkun Luo, Shanghai Jiao Tong University, Shanghai, China, lolinkun1988@sjtu.edu.cn; Di Huang, Beihang University, Beijing, China, dhuang@buaa.edu.cn; Yunhong Wang, Beihang University, Beijing, China, yhwang@buaa.edu.cn; Liming Chen, Ecole Centrale de Lyon, Ecully, France, liming.chen@ec-lyon.fr.

Manuscript submitted to ACM

One first learns a predictive model (*e.g.* a Convolutional Neural Network (CNN)) with sufficient training data, and then applies the learned model for prediction on new data whose probabilistic distribution is assumed the same as that of the training data. This learning principle is known as the Empirical Risk Minimization in statistical learning theory [122]. In this way, each task is solved individually by learning a corresponding model from scratch. The disadvantage of this approach is obvious: the relatedness between different tasks is unexplored, thereby making the learning process inefficient. In many real-life applications, some tasks have abundant training data, but most others often have very few training data. When learning each task in an independent manner, it could be hard to solve a task that has limited training data since the available training data may not be enough for learning a reliable model.

It is thus of capital importance to be able to capitalize on previously learned knowledge. Indeed, taking into account the learned knowledge from previous tasks when learning a new task can be beneficial both in gaining extra training information and in saving training time (by avoiding training from scratch). For example, when training an image classification model for some rare categories, one may face the problem of having few training data, in this case, fine-tuning a CNN model pre-learned on some related image data as feature extractor could significantly improve the classification performance on target categories [138]. Knowledge transfer could also be applied for different kind of tasks. For example, since the ground-truth annotations for Object Detection tasks are usually harder to get than those for Image Classification tasks (the former includes not only class labels but also bounding box information), one could therefore borrow knowledge from a learned image classification task for training a new object detection model [114]. This knowledge transfer (also known as 'Transfer Learning (TL)') has been studied in previous works and has been attracting more and more attention from several research communities, *e.g.*, computer vision, machine learning, within the current big data era.

Within the research community of knowledge transfer, one usually defines a *target* task, to which the knowledge will be transferred, and one or several *source* tasks, from which the knowledge will be captured or learned. Depending on assumptions on target and source tasks, knowledge transfer setting can be categorized into different scenarios, *e.g.* Domain Adaptation (DA), self-taught learning, and few-shot learning.

Because of its importance, there exist an increasingly large amount of research work focused on TL and there have already been several surveys discussing state of the art research work on knowledge transfer at different time period as illustrated in Fig.1 and Fig.2. A first comprehensive survey on Transfer Learning(TL) was made in 2010 by Pan and Yang in [88], which discusses TL methods for a broad range of applications, including vision recognition and other types of applications. They have categorized different TL works according to their assumptions (settings) and the nature of content to transfer. Specifically, as shown in the top left diagram of Fig.1, they distinguish three settings of TL: (1) "*Inductive* TL", where the source task could be different from the target task, some labeled target training samples should be provided to induce the target predictive model with the help of the source data or the source model. (2) "*Transductive* TL", where the source and target share the same task, while having different data distributions. Therefore the source data should be adapted to the target data distribution in order to help learning an effective model for doing prediction on the target data; and (3) "*Unsupervised* TL", where the target task and source task are all unsupervised tasks, *e.g.* clustering, dimensionality reduction, density estimation, *etc.*. Each setting can further depict different TL scenarios with more detailed assumptions. Finally, as shown in the top left part of Figure 2, they come up with a synthesis of four different kinds of TL approaches: (1) "*Instance Transfer*", where re-weighted source instances are used directly for learning the target task; (2) "*Feature representation transfer*", which tries to find a "good" feature representation which reduces the discrepancy between the source and the target distributions and increase the performance of classification and regression models on target data; (3) "*Parameter transfer*", which transfers parameters or priors from the source

models to the target models; and (4) "*Relational knowledge transfer*", which builds mappings of relational knowledge from the source data to the target data. This survey is later enhanced as a book chapter[85], it builds the basic definition and taxonomy of TL, this methodology is adopted in various TL related works, including some survey papers, *e.g.* [131][40][24][144].

Another survey was made in 2014 by Shao *et al.* [105] who focus on TL works for vision categorization problems. As shown by the top right diagram of Figure 2, this survey categorizes TL techniques into "feature representation level knowledge transfer" and "classifier level knowledge transfer", respectively.

As [88] and [105] don't cover important development in TL since 2015, a more recent survey was made by Zhang *et al.* [144] who overview TL techniques for cross dataset recognition problems. Like in [88], they also distinguish different works according to the settings. Specifically, they show what kinds of methods can be used when the available source and target data are presented in different forms. Compared to [88], they give a more detailed categorization of different cross-dataset settings, as shown in the lower diagram of Fig.1. They also summarize different kinds of criteria which could be used in solving knowledge transfer problems, as shown in the middle diagram of Fig.2. Like in [88], the TL methods discussed in [144] also covers a broad range of applications. Another recent survey [145] on "Transfer Adaptation Learning" considers TL and DA as weakly supervised learning problems, and reviews five different kinds of related approaches, as shown in the bottom diagram of Fig.2.

As shown in the beginning of this section 1,the common way to solve vision recognition tasks follows the principle of Empirical Risk Minimization. Due to the specialty of visual data, this common solution could be further defined as a two-step framework: the feature extraction step and the prediction step. Following this two-step framework, [105] simply categorizes TL works for vision categorization problems into two categories (as shown in figure 2). Although this categorization shows its correspondence to the common way of solving visual problems, it does not fully reveal the characteristics of knowledge transfer within this scope. Furthermore, [105] only covers research works before 2014. While with the rapid growth of vision recognition techniques, a lot of new works, especially those based on deep neural networks, are published since 2015. These recent works are not discussed in [105]. On the contrary, [144] and [145] include more recent TL works, while they adopt commonly used methodologies to categorize TL settings and approaches.

In this survey, we propose to discuss knowledge transfer works for vision recognition tasks, and to explore the common rules behind these works. As the primary goal of TL methods is to harness learned knowledge for re-use in novel learning tasks, we overview in this survey knowledge transfer methods from the viewpoint of the knowledge being transferred. Specifically, We will firstly discuss the reusable knowledge in a vision recognition task, and then we will categorize different kinds of knowledge transfer approaches by where the knowledge comes from and where the knowledge goes. We aim at finding general rules across different TL settings instead of focusing on their particularities. This viewpoint is in clear contrast to previous surveys on TL, *i.e.*, [88] [105][144]. However, any existing method has its own scope of applicability, and we will indicate the applicable scenarios when introducing each method. In Table 5 we list some common knowledge transfer settings/problems and their corresponding possible solutions.

The contributions of this survey could be summarized in the following list:

(1) We give a survey about the knowledge transfer in vision recognition from the perspective of knowledge itself, *i.e.* we categorize different knowledge transfer approaches by where the knowledge comes from and where the knowledge goes. This is in clear contrast to existing surveys that are from the problem-oriented perspective [88] [85] [131] [24] [144], or from the technique-oriented perspective [145]. We believe that our methodology is

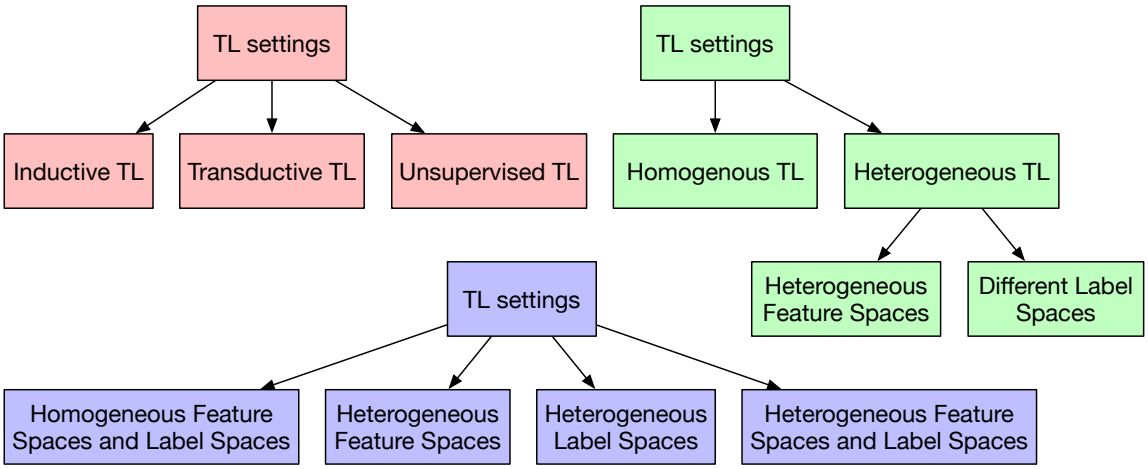

Fig. 1. Existing Categorizations of Transfer Learning Settings: The top left part is the categorization of TL settings from [88], the top right part is the categorization of TL settings from [85] and the bottom part is the categorization of TL settings from [144] (TL stands for Transfer Learning)

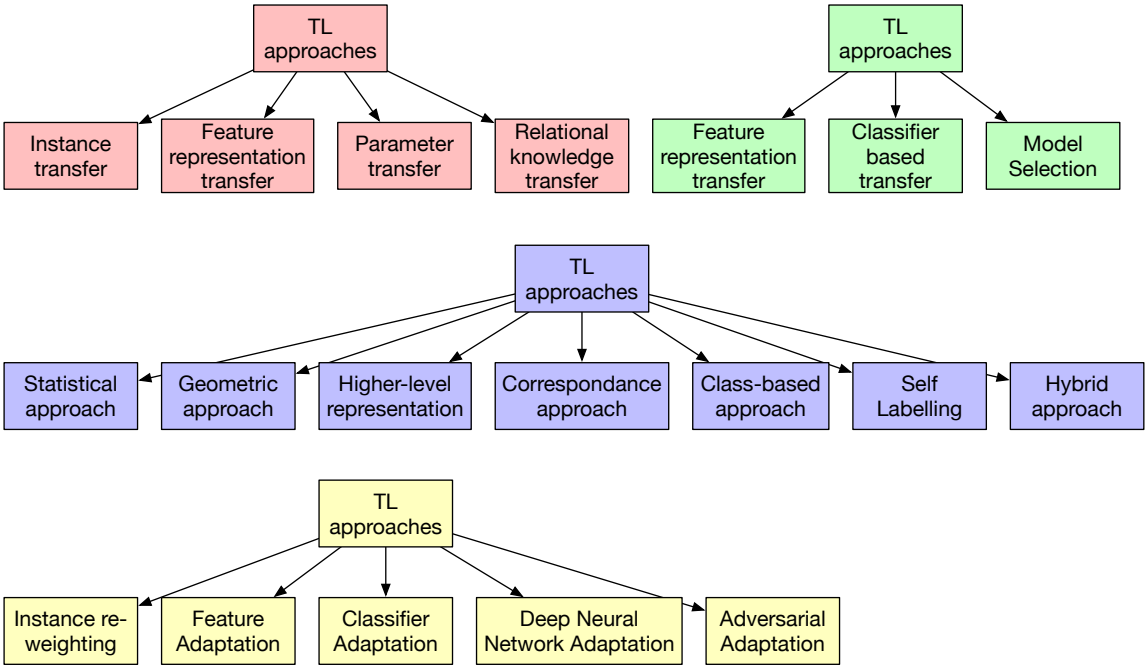

Fig. 2. Existing Categorization of Transfer Learning approaches/techniques: The top left part is categorization of TL approaches from [88], the top right part is categorization of TL approaches from [105], the middle part is categorization of TL techniques from [144], and the bottom part is categorization of TL techniques from [145](TL stands for Transfer Learning)

closer to the nature of knowledge transfer and could reveal the common rules behind different transfer learning settings by showing different kinds of knowledge flows from the source to the target (shown in Fig. 3).

(2) Unlike some existing surveys that only cover a sub-topic of knowledge transfer (for example [40] and [24] shows research works about heterogeneous transfer learning, [147] studies the multi-task learning, [77] studies the transfer metric learning), in our survey, we show knowledge transfer works that cover a wide range of transfer learning settings or problems, these related settings could be found in the Table 5.
(3) We cover very recent research trends including contrastive learning (in section 4.1.2), meta learning (in section 4.3.2), few shot learning (in section 4.2.2), *etc.*. These recent research trends are not all included in previous surveys [24] [40] [77] [88] [85] [105] [131] [144] [145] [147].
(4) We give a discussion and show some future research directions for knowledge transfer in vision recognition in the section 6, which we hope to inspire researchers to make further contributions to this domain.

The rest of this survey is organized as follows. In Section 2 we introduce knowledge types exist in a vision recognition pipeline and the possible knowledge transfer flows that we are going to introduce in this survey; In section 3 we show three categories of knowledge transfer works that transfer knowledge directly from the source data by data selection/re-weighting or adaptation; In section 4 we show three categories of knowledge transfer works that transfer knowledge from source model parameters by finding or learning generalizable knowledge in the source model; In section 5 we show research works that study the transferability between different tasks; In section 6 we give a discussion and some future directions; And finally in section 7 we conclude this paper.

2 KNOWLEDGE IN VISION RECOGNITION

In this section we firstly introduce the notations we adopt in this paper to describe a vision recognition task and its solution. Then we discuss different types of knowledge that can be reused (transferred) from a previous vision recognition task to a new one.

The following notations are adopted in the subsequent: calligraphic letters in upper cases, *e.g.*, $\mathcal{X}$, denote sets or data spaces; bold letters in upper cases, *e.g.*, $\mathbf{M}$, denote matrices; bold letters in lower cases, *e.g.*, $\mathbf{x}$, denote column vectors.

Table 1. Characteristics which define a vision recognition task and its solution

In/Out Data	$\mathbf{X}^{tr}, \mathbf{Y}^{tr}, \mathbf{X}^{te}, \mathbf{Y}^{te}$
In/Out data spaces	$\mathcal{X}$, $\mathcal{Y}$
In/Out data distributions	$P(X)$, $P(Y)$
Learned model	$f = f_K \circ \ldots \circ f_2 \circ f_1$
Feature data	$\mathbf{X}^{set,f_1}, \mathbf{X}^{set,f_2}, \ldots, \mathbf{X}^{set,f_{K-1}}$ $(set = \{tr, te\})$
Feature data spaces	$\mathcal{X}^{f_1}, \ldots, \mathcal{X}^{f_{K-1}}$
Feature data distributions	$P(X^{f_1}), P(X^{f_2}), \ldots, P(X^{f_{K-1}})$

Let's firstly define a vision recognition task $\mathcal{T}$ by two data spaces $\mathcal{X}, \mathcal{Y}$ and the corresponding data distributions $P(X), P(Y)$, where $\mathcal{X}$ is the input data space, $P(X)$ is the marginal probability distribution of the input data X, $\mathcal{Y}$ is the desired output label space, and $P(Y)$ is the probability distribution of the output data Y. Here $P(Y)$ can also be noted as $P(Y|X)$ and therefore be interpreted as the conditional distribution of Y knowing X. The goal of the vision recognition task is to find an optimal mapping $f(\cdot)$ which projects the input data X from the data space $\mathcal{X}$ to the label space $\mathcal{Y}$ so that the mapped labels Y best correspond to the ground-truth labels Y^{gt}. This mapping $f(\cdot)$ could be further decomposed into a series of projections $f = f_K \circ \ldots \circ f_2 \circ f_1$, where each f_k maps data from the space $\mathcal{X}^{f_{k-1}}$ to

the space $\mathcal{X}^{f_k}$. Specifically, $\mathcal{X}^{f_0} = \mathcal{X}$ is the input data space, $\mathcal{X}^{f_K} = \mathcal{Y}$ is the output data space, and $\{\mathcal{X}^{f_1}, \ldots, \mathcal{X}^{f_{K-1}}\}$ are intermediate feature spaces. In this way, a solution to task $\mathcal{T}$ could be defined as $\mathcal{S} = \{f_1, f_2, \ldots, f_K\}$, and these projections define new feature spaces $\mathcal{F} = \{\mathcal{X}^{f_1}, \ldots, \mathcal{X}^{f_{K-1}}\}$ and the output data space given the input data space.

For example, if we use a K-layer Neural Network model to solve the task $\mathcal{T}$, the solution could be denoted as $\mathcal{S}_{KlayersNN} = \{f_1, f_2, \ldots, f_K\}$, and the corresponding feature spaces are $\mathcal{F}_{KlayersNN} = \{\mathcal{X}^{f_1}, \ldots, \mathcal{X}^{f_{K-1}}\}$, where $\mathcal{X}^{f_k}$ ($k = \{1, 2, \ldots, K-1\}$) is the feature space which contains the output of the k-th layer of the Neural Network, and $\mathcal{X}^{f_K} = \mathcal{Y}$ contains the output of the last layer, which is the desired output label space. If we use a traditional way to solve $\mathcal{T}$, for example, a SIFT feature extraction step with an SVM classifier, the solution could then be denoted as $\mathcal{S}_{SIFT-SVM} = \{f_{SIFT}, f_{SVM}\}$ along with $\mathcal{F}_{SIFT-SVM} = \{\mathcal{X}^{f_{SIFT}}\}$, where $\mathcal{X}^{f_{SIFT}}$ is the feature space defined by the output of the SIFT feature extraction.

In reality, the probability distributions $P(X)$ and $P(Y)$ for $\mathcal{T}$ are usually not given in an analytical form. Normally they could be estimated through the given training data set $\{\mathbf{X}^{tr}, \mathbf{Y}^{tr}\}$. The test data $\{\mathbf{X}^{te}, \mathbf{Y}^{te}\}$ are supposed to follow the same distribution as the training data do, therefore allowing the model learned on training data to be applicable on the test data. In the training phase, a model $f(\cdot)$ is learned with the training set $\{\mathbf{X}^{tr}, \mathbf{Y}^{tr}\}$, and then in the testing phase, predictions could be made by applying the learned model $f(\cdot)$ on the test data $\mathbf{X}^{te}$, and the performance of the model could be evaluated by comparing the predictions with the ground-truth labels $\mathbf{Y}^{te}$. Following the decomposition of f described above, we could find out that there exist a series of feature data $\{\mathbf{X}^{set,f_1}, \mathbf{X}^{set,f_2}, \ldots, \mathbf{X}^{set,f_{K-1}}\}$ ($set = \{tr, te\}$), each belongs to their corresponding feature space, *i.e.* $\mathbf{X}^{tr,f_k}$ and $\mathbf{X}^{te,f_k}$ belong to $\mathcal{X}^{f_k}$. And in each feature space the data should follow a specific probability distribution, we denote these data distributions as $\{P(X^{f_1}), P(X^{f_2}), \ldots, P(X^{f_{K-1}})\}$.

Table 1 lists the main characteristics introduced above that describe a vision recognition task and its solution. At this level, We can observe that there exist two types of *knowledge* for a given vision recognition task: the first one is directly represented by raw data which includes the input and the output (ground-truth) data, the corresponding data spaces they belong to, and their data distributions (*i.e.* the first 3 lines in the Table 1); the other is learned knowledge which includes the learned model, feature data generated by this model, feature data spaces and feature data distributions (*i.e.* the last 4 lines in the Table 1). Both these two kinds of knowledge have the possibility to become the starting point of a knowledge transfer flow that transfers knowledge to a new vision recognition task. The knowledge directly represented by raw data is more flexible to be reused in a knowledge transfer flow since they could be adapted to the target task for learning a new model dedicated to the target; In contrast, the learned knowledge is more restricted to the source data. Nevertheless, when the source task is well chosen (*i.e.*, well related to target task), the reuse of learned knowledge could be both efficient and effective.

For example, it has been shown that deep Convolutional Neural Networks (CNNs) have the ability to produce transferable features [138] [27] [106]. Therefore, one can adopt a pre-learned CNN model, fix the feature extraction layers' parameters and only retrain the classification layer's parameters on new data, and the resulting new model is expected to have discriminative performance on the new data. In this way, we are actually reusing the *knowledge* from the parameters of the learned feature projections (*i.e.* $\{f_1, f_2, \ldots, f_{K-1}\}$) of a given pre-learned vision recognition task.

In some cases, the target training data is far from enough for learning a reliable classification model, some instance based TL approaches then choose to select source samples to enrich the target training data directly. For example, Dai *et al.* in [23] make use of AdaBoost to choose from the source labeled samples the ones which are close to the target probability distribution to help the learning of the target classification model. In this way, the *knowledge* transferred from the source task are source raw data (*i.e.* $\mathbf{X}$ and $\mathbf{Y}$), and they are reused for learning a prediction model for the target task.

Another example is unsupervised DA. In unsupervised DA, the target training data is unlabeled, therefore it is impossible to use the target training data solely for learning a reliable classification model. We thus need to seek for help in labeled source data, which is assumed to share the same label space with the target domain and to own similar but different marginal distribution *w.r.t* the target domain. In this case, one can attempt to align the source and target data distributions (either by projecting the two distribution into a shared feature space in which the two feature distributions are as close to each other as possible; or by projecting one distribution to fit the other one). Then as the two distributions are well aligned in a new feature space, a classification model learned on the source distributions is then considered as applicable for target data. In this way, the *knowledge* reused from source is actually the same as in the previous example, *i.e.*, source labeled data is reused and adapted to target data for learning a new classification model for target task.

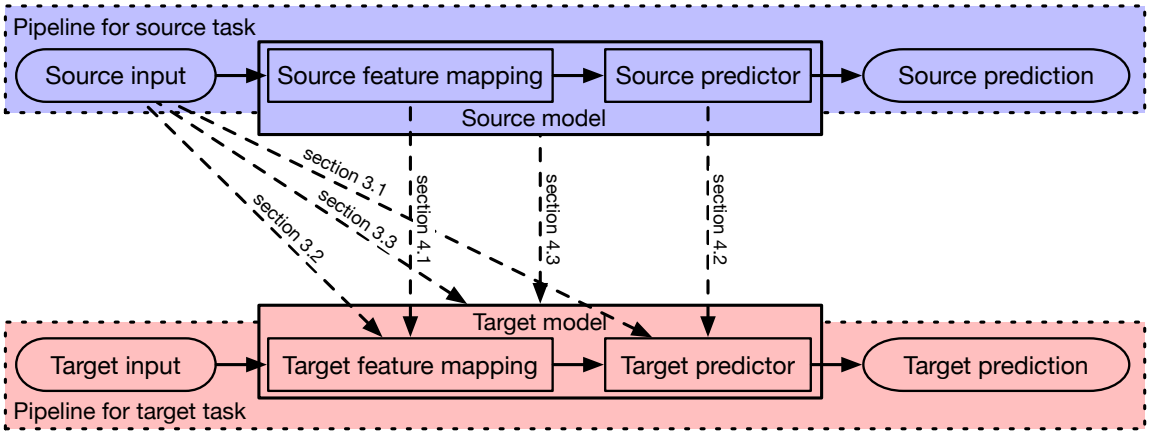

Fig. 3. Illustration of Knowledge Transfer: instead of learning a new task independently, knowledge transfer reuse existing knowledge in previous tasks for learning a new task. In this image the top part shows an example pipeline for a source task and the bottom part shows the pipeline for the target task. The dashed arrows that connect these two parts indicate the directions of knowledge flow from the source to the target. The text information on each arrow shows the corresponding section in this survey where this kind of knowledge transfer is introduced.

In the following two sections, we overview different knowledge transfer techniques/approaches for vision recognition tasks. They are categorized according to the origin of *knowledge* being transferred, *i.e. raw data* or *pre-learned model*), as well as the destination of the transferred knowledge, *e.g.* feature extractor parameters or predictor parameters, *etc.*). Figure 3 illustrates the different knowledge transfer types that are presented in the subsequent. In Table 2 we give a detailed comparison of these knowledge transfer categories. These categories are not mutually exclusive between one and another. This means that one knowledge transfer method may contain several different kinds of knowledge transfer types.

3 KNOWLEDGE TRANSFER FROM SOURCE DATA

As we have discussed in section 2, *raw data* is low level information compared to *pre-learned model*. Therefore there exist more possibilities to reuse *raw data* in target task. The main advantage of using *raw data* instead of *pre-learned model* is that *raw data* could be adapted to target task before or at the feature extraction stage, therefore the learned model after adaptation could suffer less from the discrepancy between source and target. There are various ways to adapt source data to target data. For example, when the source and target data distributions are not very far from each

Knowledge transfer types	Knowledge from	Knowledge to	Related settings/approaches
Data To Predictor Section 3.1	Re-weighted or selected source data	Target predictor	Inductive TL [23][135][93][74] Source domain selection [129] [16] [133] [75]
Data To Feat Section 3.2	Selected source data	Target feature extractor	Selective feature learning [45][78][150] Multi-source DA [10]
Data To Model Section 3.3	Adapted source data	Target model	Domain Adaptation [8] [107] [86] [87] [69] [76] Metric learning for DA [25] [77] Subspace Alignment [33] [73] [112] [143] Optimal Transport for DA [20] [21] [91] [19] DNNs for DA [130] [121] [72] [41] [119] [70] [71] Open set/Partial DA [12] [142] [13]
Feat Param Section 4.1	Pre-learned transferable feature extractor	Target feature extractor	Lifelong learning [116] Multi-task learning [3] [5] [2] [147] Supervised feature learning [94] [27] [106] Metric feature learning [34] [89] Self-taught learning [95] [126] [66] Unsupervised feature learning [31] [9] [124] [125] [64] [63] Self-supervised feature learning [28] [26] [90] [82] [146] [83]
Predictor Param Section 4.2	Pre-learned source predictor	Target predictor	SVM based TL [134] [6] [57] [118] [61] Hypothesis TL [60] [92] One-shot/Zero-shot learning [32] [139] [100]
Model Param Section 4.3	Pre-learned source model	Target model	Knowledge distillation [52] [97] [15] [136] [110] Meta learning [35] [81] [39] Lifelong learning [115] [17] [109]

Table 2. Knowledge transfer types introduced in each section and their corresponding related settings and approaches/techniques

other, a straight forward way is to re-weight (or select) source samples (or sets) so that the resulting data set could fit the target data distribution. An alternative way is to learn a shared feature extractor which projects both source and target data into a common feature space, in which the source and target feature distributions are well aligned to each other. When the source and target data distributions are not very close to each other, one could learn a projection which projects source data to the target data space (or the inverse) so that the resulting two data distributions would be close to each other. Once the two data/feature distributions are well aligned, the target task can then benefit from this shared data/feature space in different ways. For example, the target task could benefit from the discriminability of a learned shared feature space, or it could benefit from the source conditional distribution for learning a classifier in the shared data space if the target samples are not enough to support a reliable classification border.

We further categorize knowledge transfer approaches by two different kinds of knowledge destinations, *i.e.* the *feature extractor parameters* and the *predictor parameters* for the target task. As we have mentioned in section 2, these two kinds of knowledge destinations are not mutually exclusive, some works may focus on transferring source data knowledge to one particular kind of target model parameters, while some works reuse source data for learning all kinds of target model parameters.

In the following we introduce three groups of previous works that transfer source data knowledge for learning target model parameters. A brief introduction of these three categories and their corresponding related approaches could be found in table 2.

3.1 Knowledge transfer from reweighted source data samples/sets to target predictor

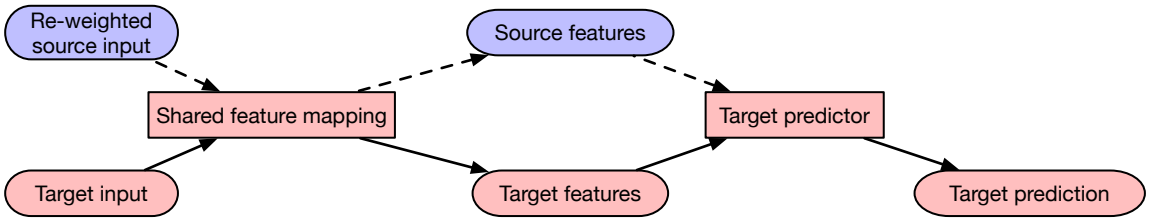

Fig. 4. Illustration of Knowledge Transfer from re-weighted source data to target predictor parameters

A natural way to adapt source data to target is by selecting most related data samples directly from source training data set, or selecting most related source sets when multiple source sets are presented. The selection is usually done by giving weights on source samples/sets. In this subsection we show a group of methods that transfers knowledge from the re-weighted source data samples/sets to the learning of target predictor parameters.

In early days before deep CNNs take over traditional feature extraction skills, research works on knowledge transfer mainly focus on borrowing knowledge from source to build the target predictor (*i.e.* knowledge transfer from source data to target predictor parameters). For feature extraction they use traditional methods (such as SIFT, HOG, *etc.*) on source and target training samples. And they use the extracted features of both source and target training samples as input data to learn a predictive model for target task. In the training process, source samples/sets that are more related to target samples will be given more important weights to enhance the knowledge transferred from them. Different works may use different ways to describe the relatedness between source and target samples, and use different strategies to select related source samples/sets.

In [23] the authors make use of AdaBoost for TL by choosing from the source labeled samples the useful ones for building a classifier for target data. Assume having a few target training samples and a large amount of source training samples, their aim is to select the source samples that follow the same probability distribution as the target samples. To achieve this goal, they build a Transfer AdaBoost framework, namely TrAdaBoost, for learning on target and source training samples at the same time. In each iteration, AdaBoost works normally on target samples, *i.e.* it increases the weights on misclassified target samples; on the other hand, for source training samples, the misclassified ones are considered as the outliers to the target distribution, therefore the weights on misclassified source samples are decreased. In this way, after several iterations, the source samples that fit the target distribution better will have larger weights, while the source samples that do not fit the target distribution will have lower weights. The source samples with large weights will then intent to help the learning a better classifier for target data.

Since this TrAdaBoost only borrows knowledge from one source task, in [135] the authors extend this method to MultiSource-TrAdaBoost which borrows knowledge from multiple source tasks. Assume having several different source training sample sets, each with abundant labeled samples, and one target training sample set with few labeled samples. In each iteration of AdaBoost, one weak learner is build on each source training set, and the one with the best performance on target set, *i.e.* the one appears to be the most closely related to the target, is chosen as the weak learner

for current iteration. In this way, the authors claim that the MultiSource-TrAdaBoost can better avoid negative transfer effect caused by brute-force knowledge transfer from the single source when this source is not closely related to the target.

Beware that, both TrAdaBoost and MultiSource-TrAdaBoost work for binary classification only, *i.e.* the source and target label spaces are the same, which could be defined as $\{+1, -1\}$ where $+1$ indicates positive sample and -1 indicates negative sample. Therefore these two works could make use of selected source samples simply as a part of target training data set for learning classification model. This strategy works when source set and target set are positively correlated, *i.e.* there exist source positive samples which are related to target positive samples and source negative samples which are related to target negative samples. Otherwise, when the two data distributions are not correlated, making use of source samples may harm the performance of the learned model for target task.

An alternative approach would solve this more complicated situation. In [93] the authors propose to use label propagation for knowledge transfer, *i.e.* they propagate labels from samples of selected source sets to each target sample. The resulting method is named Cross-Category Transfer Learning (CCTL). The coefficient for label propagation from a source sample to a target sample is defined by a transfer function, which combines both sample relatedness and domain relatedness between source and target. And the source set selection is also achieved by AdaBoost.

Since this CCTL takes into account both category correlations and sample correlations, it shows a better performance than the previously introduced TrAdaBoost and MultiSource-TrAdaBoost. However, when having L different source domains, in each iteration of CCTL one should solve $L + 1$ optimization problems. This makes this method not very efficient, especially when having a lot of source domains.

Although these methods make use of traditional feature extraction, we could easily replace their feature extractors with a state-of-the-art CNN model, which is pre-learned on some related large-scale databases (as shown in the section 4.1), to benefit from the better performance of deep features.

In [74] the authors propose a new method, namely Discriminative Transfer Learning (DTL), to reuse source selected data set for learning target classifier. They also show that by combining deep features and knowledge transfer in target classifier one can achieve better results than using traditional features. Unlike previous methods, the authors propose to build sparse reconstruction based discriminative classifiers for target task with selected source sample sets. They use positively correlated source sets as positive dictionaries and negatively correlated source sets as negative dictionaries, the difference between reconstruction errors of target samples on positive dictionary and those on negative dictionary is served as the discriminator. The source data sets are selected through two parallel AdaBoost processes. Therefore the resulting classification model is a combination of multiple selected dictionary pairs. Since this method makes use of both positively correlated source sets and negatively correlated source sets, it shows a better performance than the previously introduced CCTL, and it is also much more efficient than CCTL both on training time and on prediction time.

As can be seen, the methods introduced previously in this section are all based on boosting framework and all work for binary classification. It is possible to extend these kinds of methods to multi-class classification by one-vs-one, one-vs-all or EOOC (Error Correcting Output codes) based approaches, although this extension may increase the time for training and prediction. In table 3 we give a comparison of the detailed setting of these methods along with the original AdaBoost.

Apart from boosting based methods that re-weight source data based on their performances shown in the model, there exist also research works that explicitly define the criterion to select the best source domains for multi-source transfer learning. For example [129] and [16] propose source selection methods for Human Activity Recognition, [133] and [75] propose automatic source retrieval and selection methods for text classification.

Table 3. Comparison of boosting based knowledge transfer methods

Boosting based methods	In each Boosting iteration:	
	Update sample weights (↑: augment weight; ↓: decrease weight)	Choose weak learner
AdaBoost	Wrongly classified samples ↑ Correctly classified samples ↓	Learned with weighted samples
TrAdaBoost	Wrongly classified target samples ↑ Correctly classified target samples ↓ Wrongly classified source samples ↓ Correctly classified source samples ↑	Learned with weighted target and source samples
MultiSourceTrAdaBoost	Wrongly classified target samples ↑ Correctly classified target samples ↓ Wrongly classified source samples ↓ Correctly classified source samples ↑	The one with best performance on target from candidates learned with multiple sources
CCTL	Wrongly classified samples ↑ Correctly classified samples ↓	The one with best performance on target from candidate cross-category classifiers
DTL	Wrongly classified samples ↑ Correctly classified samples ↓	Multiple pairs of source sets which show best performance on target

3.2 Knowledge transfer from source data to target feature extractor

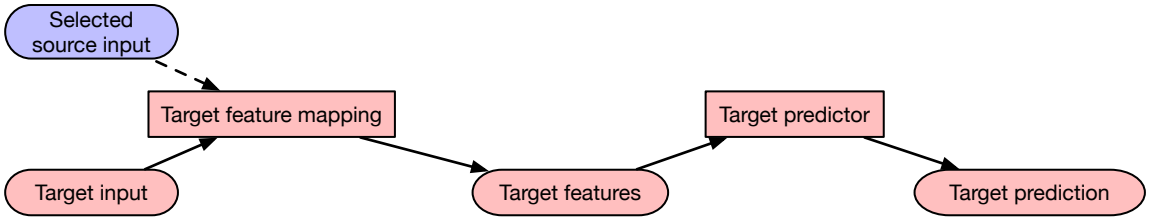

Fig. 5. Illustration of Knowledge Transfer from source data to target feature extractor

In the previous subsection we have introduced some TL methods that transfer knowledge from source data to target discriminator parameters. In this section, we show another group of methods, which also transfer knowledge from source data, while they mainly focus on using these knowledge to learn a feature extractor adapted for the target task.

As shown in the section 3.1, a straight forward way to adapt source data to target is by re-weighting source data samples/sets. For learning feature extractor parameters adapted for target, we could also use this kind of strategy.

For example, in [45] the authors propose a method which selects related source samples and then learn a deep CNN for feature extraction through joint fine-tuning with both selected source samples and target training samples. The proposed *Selective Joint Fine-Tuning* is done by two steps:

In step 1, they project both source and target training samples into a low-level feature space by applying either a Gabor filter bank or kernels in the convolutional layers of AlexNet (pre-trained on ImageNet). Then they select nearest source samples for each target training sample. The number of nearest samples is adaptive, *i.e.* hard target samples may get a larger number of source neighbors.

In step 2, they make use of the selected source samples with target training samples to optimize the source and target objective functions at the same time. A 152-layer residual network pre-trained on ImageNet or Places is shared by source and target predictive models as the feature extractor. In this way, the learned DNN for feature extraction benefits from the knowledge of the selected source data.

In a more recent work [78] the authors address a similar problem: how to select an optimal Subset of Classes (SOC) from the source data, subject to a budget constraint, for training a feature extractor which generates good features for the target task. To achieve this goal, they use a sub-modular set function to model the accuracy achievable on a new task when the features have been learned on a given subset of classes of the source dataset. An optimal subset is identified as the set that maximizes this sub-modular function. The maximization can be accomplished using a greedy algorithm that comes with guarantees on the optimality of the solution.

The source subset selection is also applicable to the multi-source DA and the "Partial Domain Adaptation" settings (see section 3.3.4 for details). For example in [10] the authors propose an iterative algorithm for multi-source DA that selects the best source domains and learns a feature extractor dedicated to the target task with the data from the selected source domains; in [150] the authors make use of the adversarial neural networks to achieve the subset selection for the Partial DA.

3.3 Domain Adaptation: knowledge transfer from adapted source data to target model

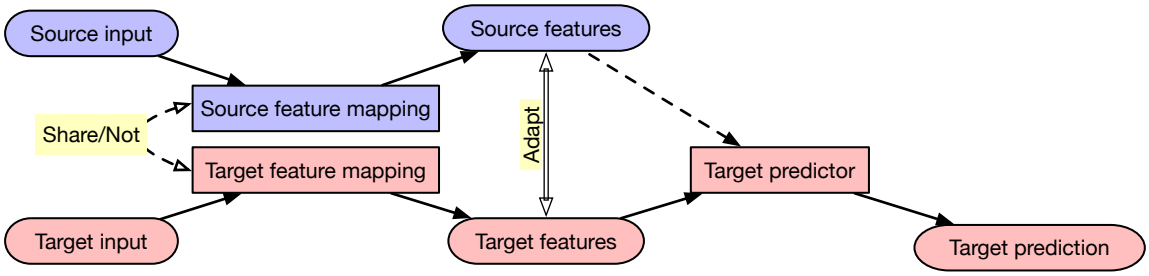

Fig. 6. Illustration of Knowledge Transfer in Domain Adaptation

A special research direction of knowledge transfer, which has been widely explored for a relatively long time, is the Domain Adaptation (DA) problem. In DA, we assume that the target data and source data share the same input and output spaces ($\mathcal{X}_T = \mathcal{X}_S$ and $\mathcal{Y}_T = \mathcal{Y}_S$), while having different but related data distributions ($P(X_T) \neq P(X_S)$ and $P(Y_T|X_T) \neq P(Y_S|X_S)$). Depending on whether labeled target training samples are available or not, DA problems could be further categorized into unsupervised DA (data available in training phase including: labeled source data and unlabeled target test data) and supervised DA (data available in training phase including: labeled source data, unlabeled target test data and a few labeled target training data). Whether supervised or unsupervised, the main goal of DA is to adapt source data distribution to target data distribution, so that the model parameters learned with both source and

target data could show good performance for target task. Therefore we consider DA as a kind of knowledge transfer from source raw data to target model parameters (instead of knowledge transfer from pre-learned model).

There exists various ways to achieve the distribution adaptation between source and target, in the following we show several groups of methods. The ultimate goal of adaptation is of course to adapt both marginal distributions ($P(X_S)$ and $P(X_T)$) and conditional distributions ($P(Y_S|X_S)$ and $P(Y_T|X_T)$), while some works may only focus on adapting one of the two, and others may focus on adapting both of them or on adapting the two joint distributions. In table 4 we show a comparison of the DA methods we introduce in this section.

3.3.1 Learning a shared feature mapping.

A natural way to achieve distribution adaptation is to learn a shared new feature space for source and target data, in which the distribution discrepancy between source and target feature data is minimized. A lot of DA works follow this way. Here we show a group of methods doing adaptation in this way. They make use of different methods for finding the new feature space (*e.g.* PCA, FLDA, metric learning, *etc.*), and use different methods for measuring the discrepancy between source and target distributions (*e.g.* Bregman divergence, MMD, *etc.*).

Adaptation with Bregman divergence based distance measure:

The first work is [107] where the authors proposed a Bregman Divergence based regularization schema for transfer subspace (representation) learning, which combines Bregman divergence with conventional dimensionality reduction algorithms. This regularized *subspace learning* learns a feature mapping and a classifier at the same time. The regularization term on the feature transformation parameters is based on a *bregman divergence* between the source marginal distribution and the target marginal distribution. Therefore the difference between the two marginal distributions will be explicitly reduced during optimization.

The authors show examples of this transfer subspace learning framework using different $F(\theta)$ (*i.e.* combining with different dimensionality reduction methods), such as transfered principal components analysis (TPCA), transfered Fisher's linear discriminant analysis (TFLDA), transfered locality preserving projections (TLPP) with supervised setting, *etc.* They also give experimental results on face image data sets and text data sets, which show the effectiveness of the proposed framework for TL problems.

Adaptation with Maximum Mean Discrepancy (MMD) as distance measure:

Similar to the previous approach, in [86] the authors proposed a TL algorithm which also combines conventional dimensionality reduction method and a distance measure for measuring the distance between marginal distributions of source data and target data. In this work the authors make use of the Maximum Mean Discrepancy as distribution distance measure, and PCA as the dimensionality reduction method.

The MMD between two sample sets could be considered as first mapping the two sample sets into a RKHS, then calculate the distance between the means of the two sets in the new space (in practice it is calculated with kernel trick which avoids the explicit mapping of the samples). By combining this MMD with common dimension reduction methods, the authors propose the Maximum Mean Discrepancy Embedding (MMDE), which learns a new latent feature space shared by source and target. A classifier is then learned in this latent space with source labeled data, and this learned classifier is directly used for target classification task (*i.e.*, they assume that in the latent space the conditional distributions of source data and target data are the same).

The authors perform experiments on indoor WiFi localization dataset and text classification dataset, the results showed that using knowledge transfer with the proposed MMDE can effectively improve the model performance compared to the same model learned without knowledge transferred from the source data.

However this method suffers from two limitations: it does not generalize to out-of-sample patterns, and the semi-definite program (SDP) it uses is computationally expensive. To get ride of these limitations, the authors further proposed in [87] a new approach, named *transfer component analysis* (TCA), which learns a set of common *transfer components* for the source and the target domains, at the same time minimizes the difference between the two data distributions in the new subspace and preserves the properties of the source and the target data.

Unlike MMDE, this proposed TCA avoids the use of SDP and does not have the problem for out-of-sample patterns. Furthermore, they propose to use an explicit low-rank representation in a unified kernel learning method, instead of using a two-step approach.

Both MMDE and TCA focus on minimizing the marginal distributions' discrepancy between source and target data, while assuming that the conditional distributions of source and target data in the learned novel feature space are equal so that a classifier learned on source data can be directly applied to target data. However, such equality assumption of conditional distributions is strong and cannot always be respected. In [69], Long *et al.* proposed a *Joint Distribution Adaptation* (JDA), which aims to jointly adapt both the marginal and conditional distributions in a principled dimensionality reduction procedure. Similar to previously introduced MMDE and TCA, JDA also makes use of *Maximum Mean Discrepancy* as the distance measurement between distributions.

Since they consider the problem of *unsupervised* DA and thereby assume that no labeled sample is provided in target training set. As a result, to reduce the mismatch of conditional distributions, the authors propose to make use of the *pseudo* labels of the target data, which are obtained by applying the classifier learned on source labeled data directly to the target data. Furthermore, Long *et al.* propose to explore the sufficient statistics of class-conditional distributions $P(\mathbf{x}^{(S)}|y^{(S)})$ and $P(\mathbf{x}^{(T)}|y^{(T)})$ instead of the posterior probabilities $P(y^{(S)}|\mathbf{x}^{(S)})$ and $P(y^{(T)}|\mathbf{x}^{(T)})$. With the true labels on the source data and pseudo labels on the target data, they match the distributions $P(\mathbf{x}^{(S)}|y^{(S)} = c)$ and $P(\mathbf{x}^{(T)}|y^{(T)} = c)$ for each class $c \in \{1, \ldots, C\}$ in the label set $\mathcal{Y}$.

The authors proposed an iterative approach where they optimize the feature mapping and pseudo labels alternatively until convergence of the pseudo labels. They performed experiments for image classification problems to evaluate the JDA approach. The results verified the effectiveness of JDA compared to other methods, including in particular TCA, on image classification problems.

Although the previous works adapt the marginal distribution and the conditional distribution at the same time, they do not consider that these two adaptation processes may have different importance for the final performance. In [127] the authors propose to balance the marginal distribution adaptation and the conditional distribution adaptation in a DA approach, in order to have the best adaptation result.

In 2010, Ben-David *et al.* proposed a theoretical work [8] to answer the important question in unsupervised DA: "under what conditions can a classifier trained from source data be expected to perform well on target data?" They address this question by giving a theoretical bound on a classifier's error on target data, which depends on its error on the source data and the divergence between source and target data. The bound is defined as follows (Theorem 2 in [8]): "Let $\mathcal{H}$ be a hypothesis space of VC dimension d. If $\mathcal{U}_S$, $\mathcal{U}_T$ are unlabeled samples of size m' each, drawn from source domain and target domain respectively, then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ (over the choice of the samples), for every $h \in \mathcal{H}$":

$$\epsilon_T(h) \leqslant \epsilon_S(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + 4\sqrt{\frac{2d\log(2m') + \log(\frac{2}{\delta})}{m'}} + \lambda \tag{1}$$

Where $\mathcal{H}\Delta\mathcal{H}$ is the "*symmetric difference hypothesis space*" for the hypothesis space $\mathcal{H}$. And $\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T)$ is the $\mathcal{H}\Delta\mathcal{H}$-divergence defined for the symmetric hypothesis class $\mathcal{H}\Delta\mathcal{H}$.

From the authors' analysis we could learn that: the performance of a hypothesis h on the target domain is mostly bounded by three terms: (1) the performance of this hypothesis on the source domain (the first term in Eq. 1), (2) the data divergence between the two domains (the second term in Eq. 1) and (3) the coherence of the hypothesis functions across the two domains (the last term in Eq. 1). In previous research works which we have introduced, *e.g.* TCA [87] and JDA [69], the authors only seek to minimize the second term of Eq.1, *i.e.* the difference between data distributions. In a recent work [76], the authors propose to also minimize the last term in Eq.1 by enhancing discriminativeness of the joint feature space and by performing geometric alignment of the underlying data manifold structures across source and target domains. They propose a novel method named *Discriminative and Geometry Aware Domain Adaptation (DGA-DA)*.

Based on the framework of JDA [69], DGA-DA [76] further add a *repulsive force term* to increase the distance of sub-domains with different labels, and two additional consistency constraints, *i.e. label smoothness consistency (LSC)* and *geometric structure consistency (GSC)*, in order to preserve the hidden data geometric structure across different domains.

The authors performed extensive experiments for 49 image classification DA tasks on 8 popular DA benchmarks to verify the effectiveness of the proposed DGA-DA method. They also carried out analysis of DGA-DA *w.r.t.* its hyper-parameters and the convergence speed. In addition, using both synthetic and real data, the authors provide some illustrations for visualizing the effect of data discriminativeness and geometry awareness.

Learning a shared feature space with metric learning:

An alternative way to learn a shared feature space is to cast the representation learning problem into the metric learning scenario. In metric learning, we learn a new metric that defines the distance between two samples in the input sample space. This distance could be used to measure the discrepancy between source and target distributions. For example, Ding and Fu develop a "robust transfer metric learning (RTML)" framework in [25] for unsupervised DA.

Suppose having a source training set $\mathbf{X}^{(S)} = \{\mathbf{x}_1^{(S)}, \ldots, \mathbf{x}_{n^{(S)}}^{(S)}\}$ of labeled samples from C categories, and a target set $\mathbf{X}^{(T)} = \{\mathbf{x}_1^{(T)}, \ldots, \mathbf{x}_{n^{(T)}}^{(T)}\}$ of unlabeled samples, where $\mathbf{x}_i^{(S)}, \mathbf{x}_i^{(T)} \in \mathbb{R}^d$.

The objective function of RTML is defined as follows:

$$\min_{\mathbf{M}\in\mathbb{S}_+^d} \sum_{c=0}^{C} \text{trace}(\mathbf{\Phi}^{(c)}\mathbf{M}) + \alpha\|\bar{\mathbf{X}} - \mathbf{M}\tilde{\mathbf{X}}\|_F^2 + \lambda \sum_{i=r+1}^{d} (\sigma_i(\mathbf{M}))^2 \tag{2}$$

where $\mathbf{\Phi}^{(c)}$ is the difference between the mean of the source samples labeled to the c-th category and the mean of the target samples with pseudo labels belonging to the c-th category. $\mathbf{M} \in \mathbb{R}^{d\times d}$ is the positive semi-definite ($\mathbf{M} \in \mathbb{S}_+^d$) distance metric to be learned. The second term in Eq.(2) is a denoising term for preserving energy of the two domains. The second term in Eq.(2) is a denoising term for preserving energy of the two domains. $\mathbf{X} = [\mathbf{X}^{(S)}, \mathbf{X}^{(T)}]$, $\bar{\mathbf{X}}$ is the m-times repeated version of $\mathbf{X}$, and $\tilde{\mathbf{X}}$ is the corrupted version of $\bar{\mathbf{X}}$. (See "marginalized Denoising Auto-Encoder (mDAE)" [14] and "Denoising Auto-Encoder (DAE)" [125] for details) The last term is a regularization term which controls the rank of $\mathbf{M}$ to not be larger than r. The optimization is performed as follows: $\mathbf{M}$ and the pseudo labels of the target data are refined alternatively in iterations, *i.e.* optimizing one while fixing the others, until the metric $\mathbf{M}$ converges.

Since $\mathbf{M}$ can be rewritten as $\mathbf{M} = \mathbf{P}\mathbf{P}^\top$, where $\mathbf{P} \in \mathbb{R}^{d\times r}$ and $r \leqslant d$ is the rank of the metric $\mathbf{M}$, the distance defined by this metric $d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)$ can be rewritten as $\|\mathbf{P}^\top(\mathbf{x}_i - \mathbf{x}_j)\|_2$. This shows that the metric learning could actually be considered as learning an underlying subspace, and the distance defined by this metric equals Euclidean distance in this new subspace. Therefore metric learning methods could be applied for various types of knowledge transfer problems, in the next section 4 we will meet some metric learning methods for transferable feature learning. A comprehensive survey about metric learning in knowledge transfer could be found in [77].

3.3.2 *Learning separate feature mappings .*

The way to learn a shared feature space for DA demands that the source and target data distributions be enough similar to each other. If not, there might not exist such a common feature transformation that projects two distinct data distributions into two nearby feature distributions. To relax this assumption, one could assume that the source and target data distributions lie in two different subspaces and then find a way to align these two subspaces to achieve adaptation. In the following we show some different ways to make this subspace alignment possible.

Subspace Alignment with dimensionality reduction methods:

Similar to the methods introduced in section 3.3.1, Fernando *et al.* also make use of dimensionality reduction methods for subspace learning [33]. The difference is that, they propose to use two PCAs as dimension reduction on both source and target domain, respectively. Following theoretical recommendations in [8] (see section 3.3.1), this method designs two different subspaces to represent the two different domains, rather than to drag different domains into a common shared subspace. They optimize a mapping function to transform the learned source subspace to the target subspace. In their proposed Subspace Alignment (SA) method, a novel similarity function $Sim(\mathbf{x}^{(S)}, \mathbf{x}^{(T)})$ is defined (with the optimized mapping function) for comparing a source sample $\mathbf{x}^{(S)}$ with a target sample $\mathbf{x}^{(T)}$, this $Sim(\mathbf{x}^{(S)}, \mathbf{x}^{(T)})$ could be used directly in a k-nearest neighbor classification model. An alternative solution is to firstly project the source data into the learned target subspace with the learned mapping between source and target, and project the target original data into the learned target subspace, then learn a SVM classifier in the target new subspace. This work is further improved to a tensor version in [73].

As in SA the source data and the target data are processed separately using their own corresponding feature transformations, and the resulting two different feature spaces are only aligned by their principle components, the variances of the source and target data in the two spaces are not aligned. In this case, the distribution mismatch between the source and target are actually not well minimized with SA. Another problem is that, SA could not handle the situations where the mapping between the two projected spaces is nonlinear. To solve these problems, the method "Subspace Distribution Alignment (SDA)" [112] improves SA by taking into account the variance of the principal components, and "Joint Geometrical and Statistical Alignment (JGSA)" [143] further improves the performance by reducing the mismatch between source and target both statistically and geometrically. To achieve its goal, JGSA also tries to find two feature projections for source data and target data. While instead of aligning the two new data spaces with a third mapping, it minimize the divergence between data distributions in the new spaces using the same way as in JDA [69] and it adapts Fischer discriminant criteria to maximize the variance of target data and preserve the source discriminative information. Similar to JDA, they also make use of pseudo-labels on target data and they also update alternatively the pseudo labels and the learned mappings to improve the final prediction performance until convergence.

Subspace Alignment with manifold learning:

Another way to align subspaces is to use the manifold learning methods [48] [46] [7] [128]. For example in [128], the authors propose to make use of the Geodesic Flow Kernel. They consider the source subspace and the target subspace as two points in a high dimensional space, and they use the Geodesic Flow as a path to connect these two points, therefore making the connection between the two suspaces.

Subspace Alignment with Optimal Transportation:

A different way to achieve subspace alignment is to learn the mapping between source and target subspaces as an optimal transportation [20] [21] [91] [19].

Optimal transportation (OT) [123] is a well explored mathematical problem which calculates the distance between two distributions. It considers the distance between two distributions as the minimum effort one should spend to "transport" the mass in one distribution to fit the other one. Recently, as some fast calculation methods for OT have been proposed (*e.g.* [22]), OT based distance (also called Wasserstein distance, Earth-Mover's distance, *etc.*) has been more and more applied in machine learning and vision problems, especially in solving DA problems. Most works (*e.g.* [20] [21] [91]) which make use of OT methods for solving DA problems consider finding the optimal transportation between source and target distributions, and then map the source data to the target distribution (or inversely) by explicitly doing a barycenter mapping. A recent work [19] considers on the contrary that the optimal transportation between source and target distributions is underlying and does not need to be found explicitly. They focus on estimating the target prediction function while at the same time learning the underlying OT based transformation between the source and target distributions.

3.3.3 Deep learning methods for Domain Adaptation.

The last years have seen breakthroughs enabled by deep learning in an increasing number of domains, in particular for various vision recognition tasks. Recent studies show that deep neural networks (DNN) can also learn transferable features, which can be well generalized to new tasks. Therefore, more and more works tend to rely upon Deep Neural Networks to solve DA problems. The basic idea is similar to those introduced previously in Sect.3.3.1 and Sect.3.3.2. They make use of DNN as the structure for feature learning, and they add regularization on the new feature space either by using conventional discrepancy measures [121] [72], or by adding an adversarial network as a discrepancy measure [41] [119] on one or multiple intermediate layers' outputs to match the source and target distributions. A comprehensive survey paper on Deep learning Methods for DA could be found in [130]. We will introduce in detail two representative groups of methods in the following part of this section.

Deep Adaptation with MMD criterion:

In [72], Long *et al.* proposed a "Deep Adaptation Network (DAN)" architecture. Similar to the previously introduced JDA in Sect.3.3.1, DAN also uses Maximum Mean Discrepancy (MMD) as distance measurement for adapting source and target distributions. Specifically, they used a multi-kernel version of MMD, named MK-MMD as proposed by [49], which gives better performance than MMD and also minimizes the error of rejecting a false null hypothesis.

To learn an optimal feature transformation, the authors propose to extend the AlexNet architecture [59]. According to [138], the convolutional layers in a CNN learn transferable features in the first layers and less transferable features in the middle layers, while the last layers are more domain specific. The authors therefore propose to fix the parameters in the pre-learned *conv1-conv3* layers, fine-tune these in *conv4 – conv5* layers, and retrain the parameters in *fc6 – fc8* layers. At the same time they add an MK-MMD based adaptation regularizer on multiple layers' outputs to require the source and target distributions be as close to each other as possible in the hidden feature space. The authors propose to

initialize the parameters in DAN with those of the AlexNet model pre-trained on ImageNet dataset [99]. The training process is therefore a fine-tuning of this pre-trained model on source and target training data.

This work is further improved in [70] and [71]. Since in DAN only the marginal distributions of source and target features are adapted, directly sharing the source classifier to target may suffer from the conditional distribution discrepancy between the two domains. In [70] the authors proposed a Residual Transfer Network which learns a classifier dedicated to target. They add some extra residual layers on the target classifier to get the source classifier, in this way they avoid brute-force application of source predictor to the target. In [71] the authors further propose a joint maximum mean discrepancy (JMMD) criterion in order to adapt the joint distributions (feature+label) of source and target data, the resulting model is named Joint Adaptation Network (JAN). A comparison of these methods to other DA methods is shown in Table 4.

Deep Adaptation with Adversarial Networks:

Another group of methods are those who make use of adversarial networks, instead of MMD used in previously introduced works, to reduce the distribution discrepancy between source and target feature data (as shown in figure 6). These methods are inspired by the recent research trend on generative adversarial networks (GANs) for unsupervised learning [47]. A generative adversarial network is usually build by two parts: one generative part (*i.e.* the generator) and one discriminative part (*i.e.* the discriminator). The generator's objective is to generate samples that are as close as possible to the training samples, while the discriminator's objective is to distinguish between the generated samples and the real training samples. The two parts are adversarial. When the two parts are trained alternatively, the network can get to a balanced situation where both the two parts are well trained. In GANs, the discriminator plays the role as a distance measure, it evaluates the distance between the distribution of the generated samples and that of the real samples. That's why the idea of adversarial networks could be applied to solve DA problems, simply by adopting the discriminator as a replacement to traditional distance measures (*e.g.* MMD). In [120] the authors propose a general framework to describe these kind of methods, in the following we show that this general framework corresponds to our categorization method for traditional DA methods.

The main goal of adversarial adaptive methods is to regularize the optimization of the source and target feature mappings, $f^{(S)}$ and $f^{(T)}$, so as to minimize the discrepancy between the source and target feature distributions: $\mathbf{X}^{(S)(f^{(S)})} = f^{(S)}(\mathbf{X}^{(S)})$ and $\mathbf{X}^{(T)(f^{(T)})} = f^{(T)}(\mathbf{X}^{(T)})$. When the distance between two distributions is minimized, a classifier $f^{c(S)}$ learned on source data could then be applied for classification on target data.

An adversarial adaptation approach could then be determined by answering the following questions: (1) Are the source mapping and target mapping generative or discriminative model? (2) Are the weights of source and target mappings shared or not? (3) Which kind of adversarial objective is used?

The first question asks about the parameterization of the source and target mappings, it equals the choice of feature extractions methods in traditional DA methods (*e.g.* PCA or FLDA, *etc.*), except for that most DA methods use discriminative mappings since DA problems generally consider discriminative tasks. While generative mappings are also possible to be used for solving DA problems and they are explored in some recent works (*e.g.* [68]), in these works they use random noise as input to the generative network mapping to get output samples, an intermediate output of the mapping is used as feature for training task-specific classifier. The second question equals the choice of shared feature space or separate feature spaces with subspace/distribution alignment in DA methods. The third question equals the choice of distance measure for evaluating the discrepancy between source and target distributions. By answering these three questions, we could then categorize these adversarial DA methods as in the bottom part of Table 4.

Table 4. Comparison of Domain Adaptation methods (In the last column, 'M' stand for Marginal, 'C' stand for Conditional, 'J' stand for Joint, 'S' stand for Source and 'T' stand for target)

Method	Mapping method	Shared mapping	Discrepancy measure	Minimize Distance between
TSL [107]	dimension reduction	yes	Bregman divergence	M distributions
MMDE [86]	PCA	yes	MMD	M distributions
TCA [87]	PCA	yes	MMD	M distributions
JDA [69]	PCA	yes	MMD	M and C distributions
BDA [127]	PCA	yes	MMD	M and C distributions
MEDA [128]	PCA	no	GFK + MMD	M and C distributions
DGA-DA [76]	PCA	yes	MMD	M and C distributions
RTML [25]	underlying	yes	learned metric	M and C distributions
SA [33]	PCAs	no	distance between Principle components	S and T subspaces
TSA [73]	Tucker Decomposition	no	distance between Principle components	S and T subspaces
SDA [112]	PCAs	no	distance between Principle components	S and T subspaces
JGSA [143]	PCAs	no	distance between Sample means	M and C distributions
DA-ROT [20]	Barycenter mapping	no	Optimal transport	M distributions
OT-DA [21]	Barycenter mapping	no	Optimal transport	M and C distributions
JDOT [19]	underlying	no	Optimal transport	J distributions
ME-DOT [91]	Barycenter mapping	no	Optimal transport	M distributions
DAN [72]	Discriminative NN	yes	MK-MMD	M distributions
RTN [70]	Discriminative NN	yes	MK-MMD	M distributions
JAN [71]	Discriminative NN	yes	MK-MMD	J distributions
Gradient reversal [42]	Discriminative NN	yes	Adversarial NN (Mini-max)	M distributions
Domain confusion [119]	Discriminative NN	yes	Adversarial NN (Confusion)	M distributions
CoGAN [68]	Generative NN	no	Adversarial NN (GAN)	M distributions
ADDA [120]	Discriminative NN	no	Adversarial NN (GAN)	M distributions
CyCADA [55]	Discriminative NN	no	Adversarial NN (GAN)	M distributions

3.3.4 *Relaxation on shared label space assumption.*

One of the assumptions of DA, that source and target task share a same label space, restricts its application in reality. When outlier classes appear in source or target data, it will make the equal class number adaptation difficult. Therefore, recently several works start to study the case without this shared label space assumption. The main idea of these works is to identify the outliers and only do adaptation on classes shared by source and target.

For example, in [12] the authors proposed the 'open set Domain Adaptation' problem, where only a few categories of interest are shared between source and target (outliers exist in both source and target). To deal with this problem, they learn a mapping from source to target, which maps source samples to be close to target distribution. This is done iteratively: first assign pseudo class labels to a part of the target samples, then minimize the distance between the target and source samples which have the same label. The assignment problem is defined by a binary linear program that also includes an implicit outlier handling, which will not assign labels to images that are not related to any source domain images.

In [142] the authors deal with a similar problem *partial Domain Adaptation*, where the source domain has more classes than the target domain. They extend the adversarial neural networks based DA methods for finding the source samples which are from the outlier classes and at the same time reduce the discrepancy between the source and target distributions for the shared classes. In [13] the authors also deal with the partial DA problem. They propose Selective Adversarial Network (SAN), which selects out the outlier source classes and maximally matches the data distributions in the shared label space.

4 KNOWLEDGE TRANSFER FROM PRE-LEARNED MODEL

In this section we are going to introduce TL works that reuse knowledge from previously learned source tasks. Compared to those introduced in section 3, these methods reuse knowledge from parameters (or outputs) of a model pre-learned for source task. Since they avoid learning from scratch with source raw data, these methods are usually more efficient in training. While as the pre-learned model is more adapted to source task, it makes the result of knowledge transfer more sensitive to distance between target and source.

4.1 Knowledge transfer from feature extractor parameters

The very first group of methods is those who reuse a pre-learned feature extractor for new target tasks. We categorize these approaches into supervised feature learning methods and unsupervised feature learning methods.

4.1.1 Supervised transferable feature learning.

One of the pioneer works on knowledge transfer is [116], in which Thrun proposed the concept of *lifelong learning*. He proposed an algorithm, which makes use of source data to learn a feature mapping, denoted by $g : I \rightarrow I'$, and then apply this learned mapping for feature extraction on target data to help the classification of the target task. The learning objective of the feature mapping is to make every pair of positive samples stay close to each other and every pair of positive and negative samples stay far from each other in the new feature space. In this way, the learned new feature representation is considered discriminative for source data, and hopefully also be discriminative for target data. Thrun has perform experiments on an object recognition task to show that the use of the transferable feature mapping on target task could improve the performance of the target task when there exist only a small number of labeled target training samples.

This work is further generalized by several authors [3] [5] [2]. The three works can all be considered as special cases of the framework of '*structural learning*' proposed in [3]. And [3] is further applied to image classification in [94].

As shown in section 3.3.1, metric learning could provide same effect as subspace learning. Therefore transferable/shared feature learning could also be done with metric learning methods.

One of the first works is [34], which tries to learn a shared feature representation using metric learning disciplines. Similar to the very first TL method [116], this algorithm learns a feature transformation which is later taken as input by a nearest neighbor classifier for the target task. Unlike Thrun's transfer algorithm which deploys a neural network to learn the feature transformation, Fink's transfer algorithm make use of a max-margin approach to directly learn a distance metric.

Like the early works introduced previously in this sub-section, the strong assumption which requests the source data to be very close to the target restricts the effectiveness of this method for more general situations. In [89] a new method is introduced which combines the large margin nearest neighbor classification with the multi-task learning paradigm. Unlike the previously introduced method, this method learns a specific metric $d_t(\cdot,\cdot)$ for each of the T tasks. They then model the commonalities between various tasks through a shared Mahalanobis metric with $\mathbf{M}_0 \succeq 0$ and the task-specific characteristics with additional matrices $\mathbf{M}_1, \ldots, \mathbf{M}_T \succeq 0$. The distance for task t is defined as follows:

$$d_t(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^\top (\mathbf{M}_0 + \mathbf{M}_t)(\mathbf{x}_t - \mathbf{x}_j)} \tag{3}$$

Although there is not a specific projection as θ defined in [34], this distance defined in Eq. (3) could still be considered as a distance in an underlying new feature space. The metric defined by $\mathbf{M}_0$ picks up general trends across multiple data sets and $\mathbf{M}_{t>0}$ specialize the metric further for each particular task.

With the growth of deep learning techniques, deep neural networks have shown great success on learning transferable features, especially for visual data like images. Actually in some of the knowledge transfer methods we have introduced above (*e.g.* the methods introduced in section 3.1) where the focus is on transferring knowledge to classifier parameters, if we make use of a pre-trained deep neural network for feature extraction, the performance will get a significant improvement comparing to the same method with traditional feature extraction technique (*e.g.* the results in [74] confirm this performance gain with deep features). Nowadays, fine-tuning a deep neural network model pre-trained on some large-scale source dataset for a small customized target dataset has already become a popular way to do knowledge transfer. In [27] and [106] the authors show the transferability of pre-learned CNN features. In [138] the authors study the transferability of features from different layers of a pre-learned CNN. They show that the features from the first layers of a CNN capture general visual information and are more transferable to different tasks, while the features from the last layers of a CNN are more task-specific. This kind of generic feature learning is not only studied on 2D images, but also explored for 3D contents. In [141] the authors propose a method to learn transferable 3D representations by solving some basic 3D tasks (*e.g.* camera pose estimation, feature matching, *etc.*).

4.1.2 *Unsupervised transferable feature learning.*

The methods shown in previous section 4.1.1 all make use of labeled source data for training feature extractor. In reality, it is usually more expensive to get labeled data than unlabeled data, therefore transferring knowledge from unlabeled data would be a good choice when it is not easy to get source labels. In this subsection, we show some works that make use of unlabeled source data for learning a feature extractor purposed for target task.

A first group of methods is the so called *self-taught learning* methods. These methods aim to learn a re-constructive dictionary from unlabeled source data, this learned dictionary could then be used for feature extraction by sparse coding for target task.

[95] is the first work that proposed the *self-taught learning* problem. They proposed a sparse coding based approach, which learns high-level feature expressions with unlabeled data using sparse coding. By applying the learned model for feature extraction on target data, the target classification performance could be improved.

This sparse coding based approach is widely adopted for self-taught learning scenarios, and is also improved from different aspects by different researchers. For example, in [126] the authors propose to learn the sparse coding basis (*i.e.*, the redundant dictionary) using not only unlabeled samples, but also labeled samples. They also proposed a principled method to seek the optimal dictionary basis vectors for a smaller dictionary that demands less computational cost.

In a recent work [66], the authors propose a new sparse coding based self-taught learning framework for visual learning, which is named "self-taught low-rank (S-Low) coding". In addition to sparse coding, they also add a low-rank constraint into the reconstruction objective function to preserve the subspace structures contained in target data space. This problem is formulated as a dictionary-learning problem with rank-minimization constraint, which is a non-convex problem. The authors propose a "majorization-minimization augmented Lagrange multiplier (MM-ALM) algorithm" to solve it.

Another group of the methods are recent deep learning methods. The rapid growth of deep learning techniques shows new ways for unsupervised feature learning [31] [9]. Unsupervised deep learning models, such as auto-encoders (AEs) [124] [125] and deep belief networks (DBNs) [64] [63], could be used for feature learning with unlabeled data. Like the sparse coding approaches introduced previously, these deep models also learn the transferable feature extractor by reconstructing source samples. Once the models are learned on source data, the resulting feature extraction networks could then be applied for new target tasks.

Instead of learning transferable features by reconstructing source data, one could also define some other kinds of tasks for self-taught learning with unlabeled data, these kinds of tasks are often called "pretext tasks". For example, the pretext task of Generative Adversarial Networks (GANs) [28] is to generate new samples which are indistinguishable with source samples. In [26] the authors propose to learn features with deep CNNs by predicting context information in images. Specifically, they randomly select two patches in an image, and let the CNNs to learn to predict the spatial relationship between these two patches. In [90] the authors propose another context aware pre-text task: in-painting. They randomly mask a region in an input image, and let the CNN model to predict the missing pixels in this region with the context information around this masked region. The resulting model could be applied for in-painting tasks, and is also possible to be used as feature extractor for classification, detection and segmentation. In [82] the proposed pre-text task is to solve jigsaw puzzles. The authors propose a siamese-ennead CNN, which is named the "*context-free* network (CFN)". The learned model could then be applied for feature extraction in object classification and object detection. In [146] the authors propose to use image colorization as the pretext task. They take colorful images as input, convert the images into gray-scale images, and then train a CNN model to re-colorize these images. The learned CNN model could then be applied as a transferable feature extractor in new tasks. In [83] the proposed pre-text task is to count visual primitives in images. They either cut input image into pieces or scale the image, then let the CNN model to count visual primitives in the input and the transformed images. The model should follow two supervision rules as the objective: one is that the sum of detected visual primitives in all pieces belonging to one image should be the same as the number of visual primitives detected in the original image; the second is that the number of detected visual primitives should be the same for one image before and after scaling.

A specific group of unsupervised feature learning methods aim at learning features that are discriminative for individual images. These methods are considered as "contrastive learning" [132] [53] [149] [51]. For example, in [132] the authors propose to make use of the DNN architectures for learning an "instance-level discrimination" model. They

replace the softmax part of DNNs with a "noise-contrastive estimation (NCE)" to approximate the softmax supervision on the output distribution of DNNs. Once the model is learned, it could then be applied as a feature extractor on new tasks combining with a simple k Nearest Neighbor classifier. In [51] the authors propose the "Momentum Contrast(MoCo)", which learns a large and consistent dynamic dictionary for unsupervised feature learning with the contrastive loss. The performance of the learned features with MoCo is even comparable to that of the features learned in a supervised way.

4.2 Knowledge transfer from predictor parameters

Apart from feature extractor parameters, predictor parameters of a pre-learned model could also be reused for learning a new target task. In this section we show two groups of methods, one is based on discriminative models (*e.g.* SVMs), the second one is based on generative models(*e.g.* Bayesian models).

4.2.1 Knowledge transfer from discriminative models.

Support Vector Machine (SVM) is a supervised discriminative learning method, which learns the conditional distribution of labels on knowing input features. Several early works on knowledge transfer from model parameters are constructed based on the SVM classifier [134] [6] [67]) [57] [118] [61]. A common form of the objective function of these SVM based TL models could be expressed as follows:

$$\min_{\mathbf{w}^{(T)},b} \Phi(\mathbf{w}^{(T)}) + C \sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}_T} \varepsilon(\mathbf{x}_i, y_i; \mathbf{w}^{(T)}, b) \tag{4}$$

where $\sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}_T} \varepsilon(\mathbf{x}_i, y_i; \mathbf{w}^{(T)}, b)$ is the loss on labeled samples in the target data set $\mathcal{D}_T$, and $\Phi(\mathbf{w}^{(T)})$ is the regularization on model parameter $\mathbf{w}^{(T)}$ which enforces the margin maximization and the knowledge transfer. The knowledge transfer regularization is usually expressed as a minimization of the distance between the pre-learned source parameter $\mathbf{w}^{(S)}$ and the target parameter $\mathbf{w}^{(T)}$.

As can be seen, SVM based TL methods enforce knowledge transfer simply by adding a regularization term to minimize the distance between the target model parameters and the source model parameters. This brute-force regularization work well for binary-classification when target positive category and source positive category are as close as possible. The extension to multi-class classification could be done in a one-vs-all manner as shown in [61], and the negative transfer could be prevented by tuning a model selection parameter as shown in [61].

Another group of methods which transfers knowledge from source models is the so-called *hypothesis transfer learning* [60] [92]. In these methods they assume not having access to source data but only having access to hypothesis induced from source domain. For example in [92] the authors have explored the setting *Metric Hypothesis Transfer Learning*, in which they assume that the source training samples are not accessible so one can only make use of the pre-learned source metric $\mathbf{M}_S$ to help learning the target metric $\mathbf{M}$. They have mainly provided some theoretical analysis and guarantees for transfer metric learning with a biased regularization term. They propose a new stability notion called *on-average-replace-two-stability*, which measures the stability of an algorithm when suffering from a little change in its input. Then based on this, they prove that the metric hypothesis transfer learning can achieve a fast converge rate with a high probability generalization bound under certain conditions. For the weighted biased regularization term they use, *i.e.* $\|\mathbf{M} - \beta\mathbf{M}_S\|$, they propose an approach to set the parameter β rather than tune it with brute-force search. As can be seen, this metric hypothesis transfer learning uses a similar regularization term as what is used in SVM-based TL methods.

4.2.2 *Knowledge transfer from generative models.*

Another kind of classification methods are generative models, which learn the joint distribution of the labels and input features. Generative models are also adopted for knowledge transfer, especially in the case of zeros-shot or one-shot learning for object recognition, where no target sample or only one target sample is given for training an object recognition model.

A representative work is proposed in [32], which is a Bayesian-based unsupervised one-shot learning framework for object categorization. This work is based on the *constellation model* [11], where an object model consists of several parts and each part is described by its appearance. The shape of the object is described by the relative positions between each part. The appearances and relative positions are modeled by probability density functions (*e.g.* Gaussians). The objective in training step is to estimate the model distribution parameters conditioned on training samples. This could be done by using the Variational Bayes Procedure, which approximates the desired distribution using an EM like iterative updating strategy. This procedure allows incremental learning, therefore one could take an object model pre-learned on source object samples, and update its parameters with new training samples from target object for learning a new target model.

In [100] the authors propose a hierarchical non-parametric Bayesian model for one-shot learning or unsupervised few shots learning. They borrow knowledge from model priors on means and variances previously learned on source categories. When a sample from a new category is given, the model firstly find some related super-categories to this new sample, and then use the model parameters of these super categories to estimate the priors of the model for the new category.

Another work is [139], where the authors propose a generative attribute model for zero-shot and one-shot learning. In their proposed framework, one category is associated with a list of attributes. They build generative models, which are considered as attribute priors, to describe the probabilistic distributions of image features for all the attributes. Then for zero-shot or one-shot learning, one could classify images from unseen categories just by using their corresponding attribute lists and the pre-learned attribute priors. Recently a large-scale attribute data-set [148] is released specially for attribute based zero-shot learning.

As can be seen, knowledge transfer with generative models reuse model parameters pre-learned on source data and usually demand some prior knowledge on target data. For example, in [32] they assume that the target objects could be expressed by a pre-defined constellation model with fixed number of object parts, in [139] they demand the attribute list information is available for target categories. From a general point of view these methods could be seen as also transferring knowledge from some non-visual information (*e.g.* prior knowledge/semantic information).

4.3 Knowledge transfer from source model to target model

As we have mentioned previously, the knowledge transfer categories we show in this survey (see Table 2) are not mutually exclusive between one and another. In the previous two sub-sections 4.1 and 4.2 we have shown two knowledge transfer categories, which either focus on transferring feature extractor parameters, or focus on transferring predictor parameters. In this sub-section, we are going to introduce some groups of research works that consider these two parts as a whole model and transfer knowledge from this pre-learned source model to the target task.

4.3.1 *Knowledge distillation.*

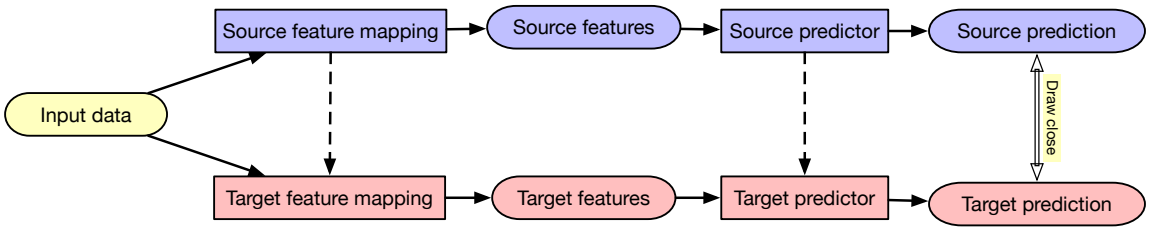

Fig. 7. Illustration of Knowledge Distillation

A special group of works is the knowledge distillation methods for Deep Neural Networks [52] [97] [15] [136] [110]. As its name 'distillation' suggests, these works try to learn a small student network that could give equal performance as a big teacher network. These works not only benefit knowledge transfer from a big pre-learned DNN, but also try to make knowledge more compact by putting the transferred knowledge into a new DNN with smaller amount of parameters.

The concept "knowledge distillation (KD)" is firstly introduced in [52], in which the authors propose the "teacher-student" framework. They make use of a softened version of the output of a big neural network, which is considered as the "teacher network", to teach a smaller neural network, which is considered as the "student network", to give the same output. In this way, the knowledge saved in the big "teacher network" could be compressed/distilled into the smaller "student network". In [97] the authors further make use of the intermediate output of the "teacher network" for training a "deeper and thinner student network". The learned student network can even give better performance than the teacher network and at the same time run faster than the teacher network thanks to the smaller number of parameters. Net2Net [15] make use of a similar "teacher-student" system, while instead of transferring knowledge from a bigger teacher network to a smaller student network, Net2Net focus on transferring knowledge to a "deeper and wider" student network in order to make the initialization and training of the big student network more efficient. Inspired by image style transfer with neural networks [43], which make use of Gram matrix to represent the global style of an image, [136] also tries to make use of Gram matrix to represent the knowledge of how neural nets solve a problem. The Gram matrix is calculated by doing inner products between outputs of two layers, and the student network is trained to generate similar Gram matrices as the teacher network in order to learn the knowledge from the teacher network.

Unlike other methods introduced above, these methods do not focus on adapting a model learned on source dataset to a new model for a different target dataset, they rather consider two different models for a same data distribution. However we still consider these methods as a kind of knowledge transfer since they consider transferring knowledge from one model to another one, and they have the potential to be extended to a normal transfer learning scenario with different source and target data distributions.

4.3.2 *Meta learning.*

Another important research direction is the so-called "meta learning" [102] [80], or "learning to learn" [117]. The objective of learning to learn is to let the algorithm benefit from the previously learned tasks so that its performance on a new task would be better than learning the new task from scratch. To achieve this the algorithm needs to encode

knowledge from previously learned source models, *i.e.* to find the meta knowledge of different tasks which could be well generalized to help the learning of a new task.

There are various ways to achieve this meta knowledge transfer. One direction is to find a good weight initialization from source models, so that the target model initialized this way could converge fast with only a few labeled samples. For example in [35] the authors propose a model-agnostic method for meta learning (MAML), which is applicable to any models that use gradient descent optimization. By defining the meta-loss as a sum of the updated source task losses, the algorithm aims to find a set of parameters which does not need to be good for every task at the current step, but should be potentially good for all tasks, *i.e.* could yield good results after updating one step further on each task. In [81] the authors propose Reptile, which is related to First Order MAML. Instead of updating the meta model with the sum of losses of the one-step-updated source models, Reptile update the meta model with the direction suggested by a model updated multiple steps on a randomly sampled source task. The authors also give theoretical analysis showing that both First Order MAML and Reptile optimize for "within task generalization".

Another way to achieve meta knowledge transfer is to learn a meta-learner, which could be used to update model parameters. For example in [103] the authors use one neural network to predict parameter changes of the second neural network. In [4] the authors propose to learn an optimizer for deep neural networks instead of using a standard optimizer such as the vanilla gradient descent. (Which is based on an earlier work [54] for shallow neural network.) They show that an optimizer should be specially designed for a particular kind of tasks to yield the best performance on this kind of task. Instead of handcrafting an optimizer for each task, one could cast the optimizer design as a learning problem and learn the best optimizer for one type of tasks. They propose a LSTM (Long Short Term Memory) network to achieve this optimizer learning. In [65] the authors tackle the same problem. They consider one optimizer as a learning policy and cast the optimizer learning problem to a reinforcement learning problem. They solve this problem by a guided policy search. In [96] the authors propose a LSTM network which not only learns the updates of parameters of the target deep neural network, but also learns a set of good weight initialization for the target model. Instead of learning the optimizer, in [50] the authors propose to use a deep neural network, which is named "HyperNetworks" to directly learn the parameters of another deep neural network.

As can be seen, a lot of methods introduced above make use of methods with a natural internal memory as the meta-learner, such as Recurrent Neural Networks (RNNs) or LSTMs. In some extreme situations, for example in one-shot learning, where the available information for learning a new task is limited, the internal memory of deep models may not be enough to store the meta knowledge learned from the source. Therefore some recent approaches make use of the Memory Augmented Neural Networks(MANNs), where external memory are used to extend the memory size and flexibility of Neural Networks. For example in [101] the authors propose to use the Neural Turing Machines (NTMs) to encode meta knowledge for one-shot learning. In [79] the authors propose the "MetaNet" where a meta-learner network and an external memory is used to learn and store meta knowledge for one-shot and few-shot learning problems.

Instead of learning initialization or meta-learner, another group of methods learn a meta metric, *i.e.* an embedding space where the query samples could be very easily recognized through some linear models or nearest neighbor classifier. For example in [113] the authors propose a "Relation Network (RN)", which is an end-to-end neural network, for few-shot learning and zero-shot learning problems. The RN learns a deep metric with source data, which describes relations between a small group of input images. Therefore when a few target samples and a query image is provided, the learned model could make prediction based on relations between the query image and the target samples.

In [137] the authors propose a special meta learning method especially designed for transfer learning. They define a "reflection function" that encodes the knowledge about how and what to transfer from a source task to a target task.

This function is trained with a series of source-target task pairs and then is finally applied to a new pair of source and target tasks to help it find the best way of knowledge transfer.

Another special group of meta learning methods aim at revealing and making use of the relationships between different tasks, we will introduce these methods in section 5.

4.3.3 Lifelong learning and Reinforcement learning.

The concept of "lifelong learning" is firstly introduced in [116]. The goal is to retain knowledge in previously learned tasks in order to make the learning of the new tasks easier. Some existing works have reviewed the development of research works on this topic [109] [17]. In [115] the authors show a real situation of lifelong learning in the computer game "minecraft", where the players should learn an endless series of different tasks. Learning a dedicated model for every task would be impossible since existing models will take bigger and bigger storage space when the number of tasks increases. Therefore the authors use the "divide and conquer" strategy, where one task is divided into a group of sub-tasks. The sub-tasks are essential skills to different kinds of tasks, therefore the sub-task models could be considered as the general knowledge in this game which could be re-used in future tasks.

Another research direction in machine learning that is quite related to lifelong learning is the "Reinforcement learning (RL)". In RL, the tasks are no longer static, they are rather "continuous". This means that a RL algorithm should make a series of decisions/actions in the RL environment in order to get positive feedbacks for its actions. RL is quite explored for vision related problems, for example for vision based robot control [37] [36] [38] and for playing graphical computer games [29] [56] [58].

5 TRANSFERABILITY: RELATIONS BETWEEN TASKS

When performing knowledge transfer from source tasks to the target task, one should always assume that the source tasks are related to the target task. Therefore, it is of grate importance to discover the relations between tasks, in order to avoid using unrelated source tasks which would cause the "negative transfer" effect [98] and to find these closely related source tasks which may yield best knowledge transfer performance [140]. In this section we show some research works that study the relatedness between different tasks.

A straight forward way to find the relatedness between tasks is to measure the distance between data distributions from different tasks. For example in [104] the authors define a "Predictive Distribution Matching (PDM)" framework, which measures the relatedness between source and target data distributions in order to encourage the knowledge transfer from source data which are "positively transferable". The data selection transfer learning methods we have introduced in section 3.2, *e.g.* [45] [78] [150], also avoids "negative transfer" by selecting source data that are most similar to target data.

Another way to measure the transferability of source tasks is to find their model performance on the target data. For example in [44], the authors propose a "Supervised Local Weight(SLW)" schema to assign a weight to each source model based on their performance on the target data. Some boosting based transfer learning methods we have introduced in section 3.1, *e.g.* [23] [135] [93] [74], implicitly assign weights to sources through the calculation of sub-model weights based on their corresponding performances in the boosting framework. A special work is [18], where the authors propose a metric to measure the transferability of each layer of the source neural network model to the target task, instead of measuring the transferability of the whole source model.

As can be seen, these methods introduced above study the transferability of individual tasks, therefore the relations between tasks should be measured every time when knowledge transfer is performed. In computer vision, the existing types of vision tasks, *e.g.* colorization, image in-painting, segmentation, *etc.*, are countable, hence knowing the relatedness between these types of vision tasks would be certainly helpful for performing knowledge transfer between these vision tasks. In [140] the authors give a thorough study on the transferability between different kinds of vision tasks. They have built a database with four million images and 26 different kinds of annotations corresponding to 26 vision tasks for each image. They firstly learn a deep encoder-decoder model for each of the 26 vision tasks on this database. Then they perform knowledge transfers on all possible first-order task pairs (one source to one target) and some high-order task pairs (multiples sources to one target). The knowledge transfer is achieved by directly applying the feature extractor (encoder) learned on source to the target task without further fine-tune. A shallow neural network is then learned on target features (output of the source encoder with target data as input) as the predictor. The authors show the performances of each source-target pairs in an affinity matrix, and then further optimize this matrix to reveal the transferability between tasks in some directed graphs, which are named "Taskonomies". Each "Taskonomy" graph shows the transferability between tasks for chosen transfer order and supervision budget and could be served as a guide for performing knowledge transfer.

Instead of evaluating transfer learning performances on thousands of source-target pairs in order to get the Taskonomy graphs, in [30] the authors propose a less time-consuming approach for learning task taxonomy. Specifically, they propose to use the "Representation Similarity Analysis (RSA)" to evaluate similarities between pre-learned source models. To calculate RSA they only need the pre-learned models and some randomly selected images as input. Therefore there is no need to finetune source models on the target data and evaluate their performances. Their experimental results show that the RSA similarity between two tasks corresponds well to the transfer learning performance from one task to the other one.

In [1] the authors propose another way to measure task relationships, especially for visual classification tasks. They propose to use a probe network to encode one classification task into a vector of fixed length. This vector is then considered as the embedding of the corresponding classification task. Therefore the relationships between different tasks could be measured in the resulting embedding space of these tasks. The authors propose a meta learning framework, which learns a metric on this embedding space and use it to predict the best source models for knowledge transfer to the target task.

The methods shown in [140] and [30] are good examples of how to measure the relatedness between different vision tasks, these relatedness could also be collected in other ways, *e.g.* by crowd-sourcing [84]. Once these relatedness between tasks are known, one can therefore make use of them for solving knowledge transfer problems. For example, in [84] the authors propose to solve the zero-shot learning problem using relationships between source tasks and the target task. Since there is no training data provided for the target task, they directly calculate the target model parameters from source model parameters based on the relatedness between source tasks to target task. This method is also considered as a meta learning method, since it could be seen as learning a meta-learner which produces parameters for the target model.

Instead of studying transferability from source to target in a two-phases manner (*i.e.* pre-learn the source model and then apply it to the target) as in [140] or [30], multi-task learning aims at knowledge sharing between multiple tasks at the same time. In [111] the authors investigate systematically this knowledge sharing between tasks to answer the question: "which tasks should and should not be learned together in one network when employing multi-task learning?". They propose a framework which could divide input tasks into groups so that the tasks in a same group

share a same network structure in the multi-task learning manner. In this way, the could avoid the situation where two unrelated tasks harm each other when they are learned in a same network.

Table 5. Some common Transfer Learning settings and their corresponding possible knowledge transfer approaches(All possible knowledge transfer flows could be found in table 2)

Name	Source v.s. Target label space	Source labels	Target labels	Objective	Possible knowledge flow
Inductive TL[108]	Different	yes	yes	single	*All*
Domain Adaptation[8]	Same/Different	yes	no	single	*Data To Model*
Self-taught Learning[95]	Different	no	yes	single	*Data To Feat* *Feat Param*
Unsupervised/self-supervised feature learning[28][83]	Different	no	no	single	*Data To Feat* *Feat Param*
Multi-task Learning[147]	Different	yes	yes	multiple	*Feat Param* *Model Param*
Zero shot Learning[62]	Different	yes	no	single	*Predictor Param* *Model Param*
One shot Learning[32]	Different	yes	yes	single	*Predictor Param* *Model Param*
Meta learning[35]	Different	yes	yes/no	multiple	*Model Param*
Life long learning[109]	Different	yes	yes	multiple	*Model Param*
Reinforcement learning [29]	Same	yes	yes	single	*Model Param*

6 DISCUSSION AND FUTURE DIRECTIONS

In the previous three sections we have introduced two groups of knowledge transfer methods, *i.e.* knowledge transfer from data (section 3) and from pre-learned models (section 4), and some research works that study the transferabilities between different tasks (section 5).

As can be seen, the methods introduced in section 3 mostly focus on data selection, re-weighting or adaptation. These methods select and extract knowledge from the source data directly and learn a model for the target task at the same time. On the contrary, the methods introduced in section 4 mainly performs knowledge transfer in a consecutive manner: they firstly pre-learn the source models and then they transfer knowledge from the pre-learned parameters to the target model. (An exception is the multi-task learning, where there is no explicit distinction of source or target, and all tasks share the learned knowledge from each other at the same time.) As a result, these two groups of methods show their own advantages in different kinds of application domains. For example, in unsupervised DA where there is no labeled training data provided for the target task, it is therefore essential to adapt the source data to the target test distribution in order to learn a target model with the source data. While in the "Life long learning" scenario where the main goal is to preserve previously learned knowledge for new incoming tasks, it would then be preferable to retain knowledge in a more compact form as model parameters.

Nevertheless, in some cases these two kinds of knowledge transfer methods could work together to get a better performance. For example, a pre-learned transferable feature extractor (as shown in section 4.1) could be applied to other transfer learning methods that focus on predictor learning (*e.g.* methods in section 3.1 or section 4.2).

In table 5 we show some common knowledge transfer settings that we have encountered in this survey, we compare their assumptions about the input data and the objective, and we show some possible knowledge transfer flows for each setting. From this table, the table 2 and the previous sections we can get some hints about the future research directions of knowledge transfer in vision recognition:

- In section 3.2 we show some source selection methods for transferable feature learning. Although there has not been much work studying unlabeled source data selection when learning a transferable feature extractor for the target task, we believe that this could be a research direction worth to explore. Since data selection has already shown its effectiveness when transferring knowledge from labeled source data to target feature extractor, it should show equal importance when transferring knowledge from unlabeled source data to target feature extractor.
- In section 4.3.1 we introduce some knowledge distillation methods which reduce the sizes of deep neural networks. Although these methods are proposed for distillation of knowledge between models that tackle the same task, they have the potential to be extended to knowledge transfer between different tasks, especially in the lifelong learning or meta learning situations. Actually "policy distillation" is already applied in lifelong learning methods as shown in [115].
- As shown in section 4.3.2, meta learning is a promising research direction for knowledge transfer, especially when combined with the rapidly growing DNN techniques.
- As shown in section 5, transferability study is important for performing knowledge transfer between different tasks. For vision recognition problems, transferability study is also a promising research direction. Some works have already shown their success in this direction [140] [30] [1] [111], but there are still much more left to be explored.

7 CONCLUSION

In this paper we have introduced a new methodology to describe and categorize knowledge transfer methods for vision recognition problems. Unlike existing surveys for transfer learning, we focus on where the knowledge comes from and where the knowledge goes. Based on this principle, we derive six major categories as shown in table 2, each containing detailed sub categories. For each category we introduce its basic idea by illustration and description and we also introduce some representative works related to this category. Then we introduce some research works on transferability study in section 5. In section 6 we give a discussion to compare the different kinds of knowledge transfer categories, and based on this discussion we give some possible future research directions in this field. We hope this paper could be a good guide for researchers to get an overview about knowledge transfer in vision recognition, to find suitable methods for their applications, or to find directions for their future research.

REFERENCES

[1] Achille, A., Lam, M., Tewari, R., Ravichandran, A., Maji, S., Fowlkes, C. C., Soatto, S., and Perona, P. Task2vec: Task embedding for meta-learning. *ArXiv abs/1902.03545* (2019).

[2] Amit, Y., Fink, M., Srebro, N., and Ullman, S. Uncovering shared structures in multiclass classification. In *Proceedings of the 24th international conference on Machine learning* (2007), ACM, pp. 17–24.

[3] Ando, R. K., and Zhang, T. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research 6*, Nov (2005), 1817–1853.

[4] Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems* (2016), pp. 3981–3989.

[5] Argyriou, A., Evgeniou, T., and Pontil, M. Multi-task feature learning. In *Advances in neural information processing systems* (2007), pp. 41–48.

[6] Aytar, Y., and Zisserman, A. Tabula rasa: Model transfer for object category detection. In *Proc. 2011 Int. Conf. Computer Vision* (Washington, DC, USA, 2011), IEEE Computer Society, pp. 2252–2259.

[7] Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., and Salzmann, M. Domain adaptation on the statistical manifold. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 2481–2488.

[8] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. A theory of learning from different domains. *Machine learning 79*, 1 (2010), 151–175.

[9] Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence 35*, 8 (2013), 1798–1828.

[10] Bhatt, H. S., Rajkumar, A., and Roy, S. Multi-source iterative adaptation for cross-domain classification. In *IJCAI* (2016), pp. 3691–3697.

[11] Burl, M. C., and Perona, P. Recognition of planar object classes. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on* (1996), IEEE, pp. 223–230.

[12] Busto, P. P., and Gall, J. Open set domain adaptation. In *The IEEE International Conference on Computer Vision (ICCV)* (2017), vol. 1, p. 3.

[13] Cao, Z., Long, M., Wang, J., and Jordan, M. I. Partial transfer learning with selective adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018).

[14] Chen, M., Xu, Z., Weinberger, K. Q., and Sha, F. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Coference on International Conference on Machine Learning* (2012), Omnipress, pp. 1627–1634.

[15] Chen, T., Goodfellow, I., and Shlens, J. Net2net: Accelerating learning via knowledge transfer. *arXiv preprint arXiv:1511.05641* (2015).

[16] Chen, Y., Wang, J., Huang, M., and Yu, H. Cross-position activity recognition with stratified transfer learning. *Pervasive and Mobile Computing 57* (2019), 1–13.

[17] Chen, Z., and Liu, B. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning 10*, 3 (2016), 1–145.

[18] Collier, E., DiBiano, R., and Mukhopadhyay, S. Cactusnets: Layer applicability as a metric for transfer learning. *2018 International Joint Conference on Neural Networks (IJCNN)* (2018), 1–8.

[19] Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems* (2017), pp. 3733–3742.

[20] Courty, N., Flamary, R., and Tuia, D. Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* (2014), Springer, pp. 274–289.

[21] Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2016).

[22] Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems* (2013), pp. 2292–2300.

[23] Dai, W., Yang, Q., Xue, G., and Yu, Y. Boosting for transfer learning. In *Proc. 24th Int. Conf. Machine Learning* (New York, NY, USA, 2007), ACM, pp. 193–200.

[24] Day, O., and Khoshgoftaar, T. M. A survey on heterogeneous transfer learning. *Journal of Big Data 4*, 1 (2017), 29.

[25] Ding, Z., and Fu, Y. Robust transfer metric learning for image classification. *IEEE Transactions on Image Processing 26*, 2 (2017), 660–670.

[26] Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1422–1430.

[27] Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., and Darrell, T. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning* (2014), pp. 647–655.

[28] Donahue, J., Krähenbühl, P., and Darrell, T. Adversarial feature learning. *arXiv preprint arXiv:1605.09782* (2016).

[29] Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779* (2016).

[30] Dwivedi, K., and Roig, G. Representation similarity analysis for efficient task taxonomy & transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 12387–12396.

[31] Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research 11*, Feb (2010), 625–660.

[32] Fei-Fei, L., Fergus, R., and Perona, P. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence 28*, 4 (2006), 594–611.

[33] Fernando, B., Habrard, A., Sebban, M., and Tuytelaars, T. Unsupervised visual domain adaptation using subspace alignment. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013* (2013), pp. 2960–2967.

[34] Fink, M. Object classification from a single example utilizing class relevance metrics. In *Advances in neural information processing systems* (2005), pp. 449–456.

[35] Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), JMLR. org, pp. 1126–1135.

[36] Finn, C., Levine, S., and Abbeel, P. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning* (2016), pp. 49–58.

[37] Finn, C., Tan, X. Y., Duan, Y., Darrell, T., Levine, S., and Abbeel, P. Deep spatial autoencoders for visuomotor learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (2016), IEEE, pp. 512–519.

[38] Finn, C., Yu, T., Fu, J., Abbeel, P., and Levine, S. Generalizing skills with semi-supervised reinforcement learning. *arXiv preprint arXiv:1612.00429* (2016).

[39] Finn, C., Yu, T., Zhang, T., Abbeel, P., and Levine, S. One-shot visual imitation learning via meta-learning. *arXiv preprint arXiv:1709.04905* (2017).

[40] Friedjungová, M., and Jirina, M. Asymmetric heterogeneous transfer learning: A survey. In *DATA* (2017), pp. 17–27.

[41] Ganin, Y., and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning* (2015), pp. 1180–1189.

[42] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research 17*, 1 (2016), 2096–2030.

[43] Gatys, L. A., Ecker, A. S., and Bethge, M. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015).

[44] Ge, L., Gao, J., Ngo, H., Li, K., and Zhang, A. On handling negative transfer and imbalanced distributions in multiple source transfer learning. *Statistical Analysis and Data Mining: The ASA Data Science Journal 7*, 4 (2014), 254–271.

[45] Ge, W., and Yu, Y. Borrowing treasures from the wealthy: Deep transfer learning through selective joint fine-tuning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI* (2017), vol. 6.

[46] Gong, B., Shi, Y., Sha, F., and Grauman, K. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), IEEE, pp. 2066–2073.

[47] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems* (2014), pp. 2672–2680.

[48] Gopalan, R., Li, R., and Chellappa, R. Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision* (2011), IEEE, pp. 999–1006.

[49] Gretton, A., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K., and Sriperumbudur, B. K. Optimal kernel choice for large-scale two-sample tests. In *Advances in neural information processing systems* (2012), pp. 1205–1213.

[50] Ha, D., Dai, A., and Le, Q. V. Hypernetworks. *arXiv preprint arXiv:1609.09106* (2016).

[51] He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722* (2019).

[52] Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[53] Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations* (2019).

[54] Hochreiter, S., Younger, A. S., and Conwell, P. R. Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks* (2001), Springer, pp. 87–94.

[55] Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A. A., and Darrell, T. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213* (2017).

[56] Jaderberg, M., Mnih, V., Czarnecki, W. M., Schaul, T., Leibo, J. Z., Silver, D., and Kavukcuoglu, K. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397* (2016).

[57] Jiang, W., Zavesky, E., Chang, S.-F., and Loui, A. Cross-domain learning methods for high-level visual concept classification. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on* (2008), IEEE, pp. 161–164.

[58] Kempka, M., Wydmuch, M., Runc, G., Toczek, J., and Jaśkowski, W. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)* (2016), IEEE, pp. 1–8.

[59] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105.

[60] Kuzborskij, I., and Orabona, F. Stability and hypothesis transfer learning. In *International Conference on Machine Learning* (2013), pp. 942–950.

[61] Kuzborskij, I., Orabona, F., and Caputo, B. From n to n+1: Multiclass transfer incremental learning. In *Proc. 2013 IEEE Conf. Computer Vision and Pattern Recognition* (June 2013), pp. 3358–3365.

[62] Lampert, C. H., Nickisch, H., and Harmeling, S. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*, 3 (2013), 453–465.

[63] Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning* (2009), ACM, pp. 609–616.

[64] Lee, H., Pham, P., Largman, Y., and Ng, A. Y. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems* (2009), pp. 1096–1104.

[65] Li, K., and Malik, J. Learning to optimize. *arXiv preprint arXiv:1606.01885* (2016).

[66] Li, S., Li, K., and Fu, Y. Self-taught low-rank coding for visual learning. *IEEE Trans. Neural Netw. Learn. Syst.* (2017).

[67] Li, X. *Regularized adaptation: Theory, algorithms and applications*, vol. 68. Citeseer, 2007.

[68] Liu, M.-Y., and Tuzel, O. Coupled generative adversarial networks. In *Advances in neural information processing systems* (2016), pp. 469–477.

[69] Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. S. Transfer feature learning with joint distribution adaptation. In *2013 IEEE Int. Conf. Computer Vision* (Dec. 2013), pp. 2200–2207.

[70] Long, M., Zhu, H., Wang, J., and Jordan, M. I. Unsupervised domain adaptation with residual transfer networks. In *Adv. Neural Inf. Process Syst. (NIPS)* (2016).

[71] Long, M., Zhu, H., Wang, J., and Jordan, M. I. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning* (International Convention Centre, Sydney, Australia, 06–11 Aug 2017), D. Precup and Y. W. Teh, Eds., vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 2208–2217.

[72] Long, M.-S., Cao, Y., Wang, J.-M., and Jordan, M. Learning transferable features with deep adaptation networks. In *Proc. 32nd Int. Conf. Machine Learning* (2015), pp. 97–105.

[73] Lu, H., Zhang, L., Cao, Z., Wei, W., Xian, K., Shen, C., and van den Hengel, A. When unsupervised domain adaptation meets tensor representations. In *The IEEE International Conference on Computer Vision (ICCV)* (2017), vol. 2.

[74] Lu, Y., Chen, L., Saidi, A., Dellandrea, E., and Wang, Y. Discriminative transfer learning using similarities and dissimilarities. *IEEE Transactions on Neural Networks and Learning Systems 29*, 7 (July 2018), 3097–3110.

[75] Lu, Z., Zhu, Y., Pan, S. J., Xiang, E. W., Wang, Y., and Yang, Q. Source free transfer learning for text classification. In *Twenty-Eighth AAAI Conference on Artificial Intelligence* (2014).

[76] Luo, L., Chen, L., Hu, S., Lu, Y., and Wang, X. Discriminative and geometry aware unsupervised domain adaptation. *CoRR abs/1712.10042* (2017).

[77] Luo, Y., Wen, Y., Duan, L., and Tao, D. Transfer metric learning: Algorithms, applications and outlooks. *arXiv preprint arXiv:1810.03944* (2018).

[78] Manjunatha, V., Ramalingam, S., Marks, T. K., and Davis, L. Class subset selection for transfer learning using submodularity. *arXiv preprint arXiv:1804.00060* (2018).

[79] Munkhdalai, T., and Yu, H. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (2017), JMLR. org, pp. 2554–2563.

[80] Naik, D. K., and Mammone, R. J. Meta-neural networks that learn by learning. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks* (1992), vol. 1, IEEE, pp. 437–442.

[81] Nichol, A., Achiam, J., and Schulman, J. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999* (2018).

[82] Noroozi, M., and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision* (2016), Springer, pp. 69–84.

[83] Noroozi, M., Pirsiavash, H., and Favaro, P. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 5898–5906.

[84] Pal, A., and Balasubramanian, V. N. Zero-shot task transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019).

[85] Pan, S. J. Transfer learning. In *Data Classification: Algorithms and Applications*, C. C. Aggarwal, Ed. CRC Press, 2014, pp. 537–570.

[86] Pan, S. J., Kwok, J. T., and Yang, Q. Transfer learning via dimensionality reduction. In *AAAI* (2008), vol. 8, pp. 677–682.

[87] Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks 22*, 2 (2011), 199–210.

[88] Pan, S. J., and Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng. 22* (Oct. 2010), 1345–1359.

[89] Parameswaran, S., and Weinberger, K. Q. Large margin multi-task metric learning. In *Advances in neural information processing systems* (2010), pp. 1867–1875.

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

[91] Perrot, M., Courty, N., Flamary, R., and Habrard, A. Mapping estimation for discrete optimal transport. In *Advances in Neural Information Processing Systems* (2016), pp. 4197–4205.

[92] Perrot, M., and Habrard, A. A theoretical analysis of metric hypothesis transfer learning. In *International Conference on Machine Learning* (2015), pp. 1708–1717.

[93] Qi, G.-J., Aggarwal, C., Rui, Y., Tian, Q., Chang, S., and Huang, T. Towards cross-category knowledge propagation for learning visual concepts. In *Proc. 2011 IEEE Conf. Computer Vision and Pattern Recognition* (June 2011), pp. 897–904.

[94] Quattoni, A., Collins, M., and Darrell, T. Learning visual representations using images with captions. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), IEEE, pp. 1–8.

[95] Raina, R., Battle, A., Lee, H., Packer, B., and Ng, A. Y. Self-taught learning: Transfer learning from unlabeled data. In *Proc. 24th Int. Conf. Machine Learning* (2007).

[96] Ravi, S., and Larochelle, H. Optimization as a model for few-shot learning. In *ICLR* (2017).

[97] Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550* (2014).

[98] Rosenstein, M. T., Marx, Z., Kaelbling, L. P., and Dietterich, T. G. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning* (2005), vol. 898, p. 3.

[99] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV) 115*, 3 (2015), 211–252.

[100] Salakhutdinov, R., Tenenbaum, J., and Torralba, A. One-shot learning with a hierarchical nonparametric bayesian model. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning* (2012), pp. 195–206.

[101] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. Meta-learning with memory-augmented neural networks. In *International conference on machine learning* (2016), pp. 1842–1850.

[102] Schmidhuber, J. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[103] Schmidhuber, J. Learning to control fast-weight memories: An alternative to dynamic recurrent networks. *Neural Computation 4*, 1 (1992), 131–139.

[104] Seah, C.-W., Ong, Y.-S., and Tsang, I. W. Combating negative transfer from predictive distribution differences. *IEEE transactions on cybernetics 43*, 4 (2012), 1153–1165.

[105] Shao, L., Zhu, F., and Li, X. Transfer learning for visual categorization: A survey. *IEEE Trans. Neural Netw. Learn. Syst. PP*, 99 (July 2014), 1–1.

[106] Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops* (2014), pp. 806–813.

[107] Si, S., Tao, D., and Geng, B. Bregman divergence-based regularization for transfer subspace learning. *IEEE Trans. Knowl. Data Eng. 22* (July 2010), 929–942.

[108] Silver, D. L., and Bennett, K. P. Guest editor's introduction: special issue on inductive transfer learning. *Machine Learning 73*, 3 (2008), 215–220.

[109] Silver, D. L., Yang, Q., and Li, L. Lifelong machine learning systems: Beyond learning algorithms. In *2013 AAAI spring symposium series* (2013).

[110] Srinivas, S., and Fleuret, F. Knowledge transfer with Jacobian matching. In *Proceedings of the 35th International Conference on Machine Learning* (Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018), J. Dy and A. Krause, Eds., vol. 80 of *Proceedings of Machine Learning Research*, PMLR, pp. 4730–4738.

[111] Standley, T., Zamir, A. R., Chen, D., Guibas, L., Malik, J., and Savarese, S. Which tasks should be learned together in multi-task learning? *arXiv preprint arXiv:1905.07553* (2019).

[112] Sun, B., and Saenko, K. Subspace distribution alignment for unsupervised domain adaptation. In *BMVC* (2015), pp. 24–1.

[113] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 1199–1208.

[114] Tang, Y., Wang, J., Wang, X., Gao, B., Dellandréa, E., Gaizauskas, R., and Chen, L. Visual and semantic knowledge transfer for large scale semi-supervised object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2017).

[115] Tessler, C., Givony, S., Zahavy, T., Mankowitz, D. J., and Mannor, S. A deep hierarchical approach to lifelong learning in minecraft. In *Thirty-First AAAI Conference on Artificial Intelligence* (2017).

[116] Thrun, S. Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems* (1996), The MIT Press, pp. 640–646.

[117] Thrun, S., and Pratt, L. *Learning to learn*. Springer Science & Business Media, 2012.

[118] Tommasi, T., Orabona, F., and Caputo, B. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Proc. 2010 IEEE Conf. Computer Vision and Pattern Recognition* (June 2010), pp. 3081–3088.

[119] Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 4068–4076.

[120] Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)* (2017), vol. 1, p. 4.

[121] Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474* (2014).

[122] Vapnik, V. Principles of risk minimization for learning theory. In *Advances in neural information processing systems* (1992), pp. 831–838.

[123] Villani, C. *Optimal transport: old and new*, vol. 338. Springer Science & Business Media, 2008.

[124] Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning* (2008), ACM, pp. 1096–1103.

[125] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research 11*, Dec (2010), 3371–3408.

[126] Wang, H., Nie, F., and Huang, H. Robust and discriminative self-taught learning. In *International Conference on Machine Learning* (2013), pp. 298–306.

[127] Wang, J., Chen, Y., Hao, S., Feng, W., and Shen, Z. Balanced distribution adaptation for transfer learning. In *2017 IEEE International Conference on Data Mining (ICDM)* (2017), IEEE, pp. 1129–1134.

[128] Wang, J., Feng, W., Chen, Y., Yu, H., Huang, M., and Yu, P. S. Visual domain adaptation with manifold embedded distribution alignment. In *2018 ACM Multimedia Conference on Multimedia Conference* (2018), ACM, pp. 402–410.

[129] Wang, J., Zheng, V. W., Chen, Y., and Huang, M. Deep transfer learning for cross-domain activity recognition. In *Proceedings of the 3rd International Conference on Crowd Science and Engineering* (2018), ACM, p. 16.

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

[131] Weiss, K., Khoshgoftaar, T. M., and Wang, D. A survey of transfer learning. *Journal of Big data 3*, 1 (2016), 9.

[132] Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 3733–3742.

[133] Xiang, E. W., Pan, S. J., Pan, W., Su, J., and Yang, Q. Source-selection-free transfer learning. In *Twenty-Second International Joint Conference on Artificial Intelligence* (2011).

[134] Yang, J., Yan, R., and Hauptmann, A. G. Cross-domain video concept detection using adaptive svms. In *Proc. 15th Int. Conf. Multimedia* (New York, NY, USA, 2007), ACM, pp. 188–197.

[135] Yao, Y., and Doretto, G. Boosting for transfer learning with multiple sources. In *Proc. 2010 IEEE Conf. Computer Vision and Pattern Recognition* (June 2010), pp. 1855–1862.

[136] Yim, J., Joo, D., Bae, J., and Kim, J. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017).

[137] Ying, W., Zhang, Y., Huang, J., and Yang, Q. Transfer learning via learning to transfer. In *International Conference on Machine Learning* (2018), pp. 5072–5081.

[138] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *Advances in neural information processing systems* (2014), pp. 3320–3328.

[139] Yu, X., and Aloimonos, Y. Attribute-based transfer learning for object categorization with zero/one training example. *Computer Vision–ECCV 2010* (2010), 127–140.

[140] Zamir, A. R., Sax, A., Shen, W., Guibas, L. J., Malik, J., and Savarese, S. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 3712–3722.

[141] Zamir, A. R., Wekel, T., Agrawal, P., Wei, C., Malik, J., and Savarese, S. Generic 3d representation via pose estimation and matching. In *European Conference on Computer Vision* (2016), Springer, pp. 535–553.

[142] Zhang, J., Ding, Z., Li, W., and Ogunbona, P. Importance weighted adversarial nets for partial domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018).

[143] Zhang, J., Li, W., and Ogunbona, P. Joint geometrical and statistical alignment for visual domain adaptation. *arXiv preprint arXiv:1705.05498* (2017).

[144] Zhang, J., Li, W., Ogunbona, P., and Xu, D. Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective. *ACM Computing Surveys (CSUR) 52*, 1 (2019), 7.

[145] Zhang, L. Transfer adaptation learning: A decade survey. *arXiv preprint arXiv:1903.04687* (2019).

[146] Zhang, R., Isola, P., and Efros, A. A. Colorful image colorization. In *European conference on computer vision* (2016), Springer, pp. 649–666.

[147] Zhang, Y., and Yang, Q. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* (2017).

[148] Zhao, B., Fu, Y., Liang, R., Wu, J., Wang, Y., and Wang, Y. A large-scale attribute dataset for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2019), pp. 0–0.

[149] Zhuang, C., Zhai, A. L., and Yamins, D. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE International Conference on Computer Vision* (2019), pp. 6002–6012.

[150] Zohrizadeh, F., Kheirandishfard, M., and Kamangar, F. Class subset selection for partial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2019).