Domain Confusion with Self Ensembling for Unsupervised Adaptation

Jiawei Wang, Zhaoshui He*, Chengjian Feng, Zhouping Zhu, Qinzhuang Lin, Jun Lv, Shengli Xie

March 31, 2018

Abstract

Data collection and annotation are time-consuming in machine learning, expecially for large scale problem. A common approach for this problem is to transfer knowledge from a related labeled domain to a target one. There are two popular ways to achieve this goal: adversarial learning and self training. In this article, we first analyze the training unstablity problem and the mistaken confusion issue in adversarial learning process. Then, inspired by domain confusion and self-ensembling methods, we propose a combined model to learn feature and class jointly invariant representation, namely Domain Confusion with Self Ensembling (DCSE). The experiments verified that our proposed approach can offer better performance than empirical art in a variety of unsupervised domain adaptation benchmarks.

1 Introduction

An essential task in visual recognition is to design a model that can adapt to dataset distribution bias [3, 37, 27], in which one attempts to transfer labeled source domain knowledge to unlabeled target domain. For example, we sometimes have a real world recognition task in one domain of interest, but we only have limitted training data in this domain. If we can use almost infinite simulation images in the 3D virtual world with labels to train a recognition model, and then generalize it to the real world, it would greatly reduce the cost of manual labelling [24, 29]. In order to obtain satisfactory

generalization capability, we turn to deep learning, which is the best known method having the robost generalization performance [26, 12, 10, 15, 28, 22]. However, deep learning models often needs millions of labeled data to fit millions of parameters. It is hard to obtain enough data to train in supervised setting where labeled data is hard to collect and annotate. As alternative methods, semi-supervised and unsupervised learning methods can reduce the large requirements [19, 20, 18]. Semi-supervised learning aims at combining labeled and unlabeled datasets for other unlabeled data from target to perform the adaptation [36, 30, 31]. Unsupervised domain adaptation is a similar problem, in which model attempts to exploit the knowledge from source domain and classify unlabeled dataset in target domain [34, 5, 35, 6, 33].

There have been extensive works in domain adaptation [5, 7, 34, 35], which focus on building an unify representation between source and target doamins for the domain adaptation problem. One of the domain adaptation methods is Domain Confusion (DC) [34]. We analyzed the DC methods based on conventional GAN-form loss function, which is approximately equivalent to RevGrad approach [5], and found that the model is hard to train stably. In this regard, we named it as training unstability problem. Additionally, we visualized the prediction results by confusion matrix (see Figure 1), which demonstrate the fact that DC methods only can align featured probability distribution rather than the feature distribution conditioning on certain class images. Figure 2 shows how the confused feature distribution looks like. We named it as feature conditional distribution misalignment problem. Apart from this, recent work presented another domain adaptation technique based on mean-teacher framework [4, 32], which achieved pretty good results in some relatively simple domain adaptation situations, such as USPS $\rightarrow$ MNIST, SVHN $\rightarrow$ MNIST and so on. But the task to transfer knowledge from MNIST to SVHN is still difficult, where the model should learn knowledge about gray-scale images and generalize it to RGB images.

In this paper, we aim at developing a "Domain Confusion with Self Ensembling (DCSE)" method for domain adaptation. The contributions are as follows (see Figure 4 for a schematic overview):

- To address the training unstability problem in unsupervised domain adaptation task, we introduce Wasserstein-GAN (WGAN) algorithm which is theoretically proved to be more stable than the conventional GAN by Arjovsky et al [2]. It shows us a way that we no longer need to balance the discriminator and feature extractor in our model. Therefore,

we are able to improve the model performance based on this stable WGAN model.

- To address the feature conditional distribution misalignment problem, we aim at forming a model that can find a joint representation of classes and features. Thus we not only need the feature informations within source domain, but also need to consider the label informations in target domain. The main intuition here is that we can use self-ensembling method to provide pseudo labels in target domain [32, 4].

- Based on these idea, we proposed an improved unsupervised domain adaptation method that combines two domain adaptation methods mentioned above. We simultaneously use domain confusion method and self-ensembling method to guide the target domain representation, so that the target representation distribution not only can regard as a whole to align with the source representation, but also can align with the feature conditional distribution of source domain. As an addtional benefit, we found these two methods can improve classification accuracy. For example, self-ensembling framework can utilize unlabeled data to improve accuracy in domain adaptation problem. But it often stucks in low accuracy due to the huge gaps of appearance between source and target datasets [4]. However, in our combined model, DC method can help it improve performance by finding a domain invariant representation despite the dataset bias. Details will be described in Section 4.2. As the result of this combination, we improve the state-of-the-art in cross tasks MNIST $\leftrightarrow$ SVHN and CIFAR $\leftrightarrow$ STL under unsupervised domain adaptation settings.

The rest of this paper is organized as follows: In Section 2, we will review the related works about domain confusion and self-ensembling techniques. And in Section 3, we will analyze the limitations of existing methods to elicit our motivation. Our approach is described in Section 4. Then we give experimental details and results in Section 5. And finally we present conclusions in Section 6.

2 Related Works

We first review image-based approaches in visual domain adaptation tasks. Paolo Russo et al. [23] presented SBADA-GAN, aiming at simultaneously

transforming the source images to target images and vice versa. It jointly optimizes bi-directional image mapping with classification loss, adversarial loss, and a class consistency loss, which aligns the generators in two directions. The main views here can be treated as data augmentation by style transfer, which avoids to decide a prior of which is the best strategy to augment data. Finally, SBADA-GAN promotes the performance greatly in case of MNIST $\rightarrow$ SVHN from prior art 52.8% to 61.1%.

Besides, recent works also focused on feature-based method aiming at transfering deep feature representation of neural network from labeled source domain to unlabeled target domain. Ganin et al. [5] proposed a Gradient Reversal Layer, achieving unsupervised domain adaptation through aligning the distributions of features over the source and target domains, where it considers the classification task as finding domain invariant representation in the feature space. Furthermore, Eric Tzeng et al. [35] used a unified framework ADDA for unsupervised domain adaptation task. It provides a simple and easy understanding view for recently proposed domain adaptation researches, which combines discriminative modeling, untied weight sharing, and a GAN loss to form a general framework. ADDA first pretrains a source encoder CNN using labeled source data, and then learns a target encoder CNN by confusing domain features with adversarial learning method. In a result, the discriminator can not properly distinguish the feature representation of source and target. In other words, ADDA learns a joint invariant representation between domains. During testing, target domain images feature are computed with the target encoder CNN, in which the features are classified by the source domain classifier.

On the contrary, there is a Self-Ensembling (SE) method [4] which is compeletly diffrent from the previous. It is derived from mean-teacher [32] and used in unsupervised domain adaptation problem. The model is formed of a student network and a teacher network. The student is trained using the cross entropy loss as usual and the teacher's weights are equal to the exponential moving average weights of the student. Meanwhile, when the maximum predicted probability of a sample is greater than a predefined confidence threshold of 0.968, the teacher gives the self-ensembling predicted probability as the consistency labels of the student. Their approach achieved some state-of-the-art results in a variety of benchmarks.

3 Limitations of existing methods

Before introducing our domain confusion with self-ensembling approach, we first give the intuition behind our proposed method. In domain adaptation task, we only have the labels in source domain. We aim to train a feature extractor to learn the joint representation of source and target domains so that the classifier can also classify the target domain image. One of the most popular methods is DC [34] which exploits a domain classifier to predict the domain of the feature representation, and a Gradient Reversal Layer (GRL) to minimize the domain discrepancy. The process that GRL reverses the gradient from domain classifier can be considered as backpropagation with a loss function multiplied by a negative constant, consequently, the model is equivalent to conventional GAN. So the following we will discuss GAN-form DC methods, showing its two main limitations in domain adaptation task. The first cons is training unstability problem, the second one is feature conditional distribution misalignment problem.

3.1 Training unstability problem of conventional GAN-form techniques

First we construct a feature extractor, a classifier and a discriminator (see Figure 4 except the teacher part for an overview). When source and target domain image simultaneously come into the feature extractor, the classifier will try to recognize the category of source images by using cross entropy loss function as usual. And the discriminator attempts to discriminate the feature representation come from which domain. Meanwhile, the feature extractor tries to use the feature representation from target domain to fool the discriminator with logit "1". So here we formalize a conventional GAN loss described in [8]. The GAN objective is defined as follows:

$$\max_{D} \mathbb{E}_{x\sim p_s}[logD(F(x))] + \mathbb{E}_{x\sim p_t}[log(1 - D(F(x)))] \tag{1}$$

$$\min_{F} \mathbb{E}_{x\sim p_t}[log(1 - D(F(x)))] \tag{2}$$

where F is the feature extractor mapping function. It uses the image sampled from source domain distribution p_s and target domain distribution p_t as input, and outputs their feature representation with distribution $p(F)$. D is the discriminator, which maximize the probability of all training examples that assign the correct domain label.

After evaluating this method in MNIST $\rightarrow$ SVHN domain adaptation task, we find that the model often gets the unstable result even after we carefully tune the hyperparameters and add some regularization skills. We follow the discussions in [9] to explain the unstable problem. In unsupervised domain adaptation task, when we use GAN-form DC technique, if we train the discriminator too well until the discriminator is optimal, optimizing the feature extractor will be the same as minimizing the Jensen Shannon divergence of the feature distributions of source and target [8]. In addtion, we could simply assume the support set of natural images are lying on low dimensional manifolds, so their feature maps can't "fullfill" the whole feature space. Therefore, the probability measure of the intersection manifolds between the source and target domains tends to be zero. When the manifolds of these two feature distributions are not overlap, the Jensen Shannon divergence term will always be a constant, thus the feature extractor can not receive adequate gradients to update itself.

Therefore, when we optimize the forementioned GAN loss functions in domain adaptation task, the feature distribution of target domain $p(F_t)$ would be far from the $p(F_s)$ of source domain, so that the target domain images are hard to be classified correctly by the classifier. For this reason, the model will get a worse accuracy in target domain recognition task. According to the above analysis, it is hard to improve the model performance based on the GAN-form domain confusion methods.

Table 1: Feature Extractor architecture

Description	Shape
Feature extractor	
Conv $3 \times 3 \times 128$ stride 1, pad 1, instance norm	$32 \times 32 \times 128$
Conv $3 \times 3 \times 128$ stride 1, pad 1, instance norm	$32 \times 32 \times 128$
Conv $3 \times 3 \times 128$ stride 1, pad 1, instance norm	$32 \times 32 \times 128$
Dropout, 50%	$32 \times 32 \times 128$
Conv $3 \times 3 \times 256$ stride 2, pad 1, instance norm	$16 \times 16 \times 256$
Conv $3 \times 3 \times 256$ stride 1, pad 1, instance norm	$16 \times 16 \times 256$
Conv $3 \times 3 \times 256$ stride 1, pad 1, instance norm	$16 \times 16 \times 256$

Table 2: Classifier/Critic architecture

Description	Shape
Classifier/Critic	
Dropout, 50%	$16 \times 16 \times 256$
Conv $3 \times 3 \times 512$ stride 2, pad 1, instance norm	$8 \times 8 \times 512$
Conv $3 \times 3 \times 512$ stride 1, pad 1, instance norm	$8 \times 8 \times 512$
Conv $3 \times 3 \times 512$ stride 1, pad 1, instance norm	$8 \times 8 \times 512$
Global pooling layer	$1 \times 1 \times 512$
Fully connected layer, 10 (critic:1) units, softmax	10 (critic:1)

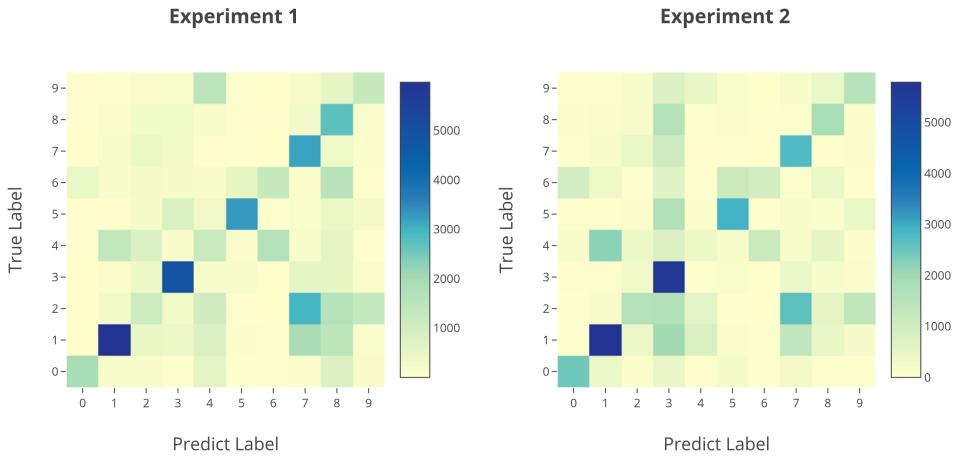
3.2 Feature conditional distribution misalignment problem

There are some unsupervised domain adaptation tasks remain undefeated due to the big gaps between domains. It is still a challenge even if we have used the adversarial training paradigm because of feature conditional distribution misalignment problem. We make the following analysis on this. First we build the model shown in table 1,2, which is jointly trained with cross entropy classification loss and adversarial loss.

According to the repeated experiments using the same model described in Section 3.2, we find the results showed in confusion matrix (Figure 1) are very different although we train the model in completely the same architecture and same hyperparameters. One possible explanation is that adversarial loss can only push target feature distribution as a whole to align with source feature distribution instead of aligning the feature distribution conditioning on different category images between source and target domains, as shown in Figure 2.

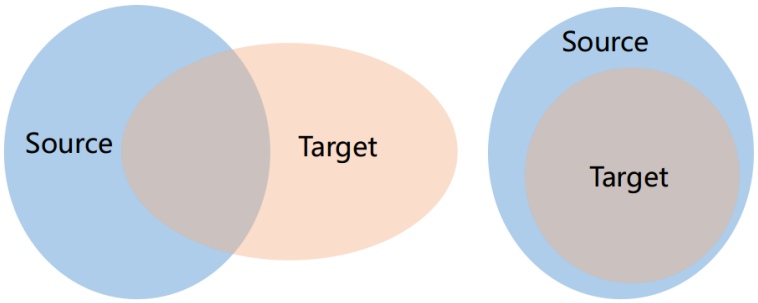
Formally speaking, let X_s^i be the i^{th} class images of source domain, so is target domain images X_t^i. Our target of domain adaptation task is to make the source domain conditional distribution $p(Y_s|X_s^i)$ as similar as target domain conditional distribution $p(Y_t|X_t^i)$, where Y_s and Y_t are the class random variables over label space. Let F_s be the feature variable of $F(x;\theta)$ where x is a random variable of source images with density p_s and θ is the parameters of feature extractor F, so is F_t. Decomposition of the above formula produces:

$$p(Y_s|X_s^i) = \sum_{F_s} p(Y_s, F_s|X_s^i) = \sum_{F_s} p(F_s|X_s^i)p(Y_s|F_s) \tag{3}$$



(a) (b)

Figure 1: Confusion Matrix visualization of the same domain confusion model runs twice. Note the most often mistaken digits are different in these two experiments, which are "7" and "8" in (a), but with "3" and "7" in (b).



(a) (b)

Figure 2: Domain confusion only can align feature distribution despite the category. (a) and (b) are the feature distributions of source and target domains before or after adversarial training respectively

$$p(Y_t|X_t^i) = \sum_{F_t} p(Y_t, F_t|X_t^i) = \sum_{F_t} p(F_t|X_t^i)p(Y_t|F_t) \tag{4}$$

where $p(F|X^i)$ is the feature distribution conditioning on i^{th} class image. And $p(Y|F)$ is the class distribution conditioning on features, which is parameter-

ized by classifier. It is easy to see when $p(F_s|X_s^i)$ is similar to $p(F_t|X_t^i)$ and $p(Y_s|F_s)$ is similar to $p(Y_t|F_t)$, our goal of domain adaptation is achieved. In our conventional GAN-form domain adaptation model, we have used source domain classification loss to determine the conditional probability distribution $p(Y_s|X_s^i)$, it fixes both domain distributions of $p(F_s|X_s^i)$ and $p(Y_s|F_s)$. Then we use the GAN loss to restrict $p(F_t) \approx p(F_s)$. The rest of the question is, although $p(F_s) \approx p(F_t)$, only $p(F_s|X_s^i) \approx p(F_t|X_t^i)$ holds partly because of $p(F) = \sum_{X^i} p(X^i)p(F|X^i)$, in which marginal distributions $p(X_s^i) \neq p(X_t^i)$. This means $p(F_t|X_t^i)$ sometimes would align mistakenly with $p(F_s|X_s^j)(j \neq i)$. Therefore, $p(Y_t|X_t^i)$ is difficult to approximate $p(Y_s|X_s^i)$.

The above analysis tells us adversarial learning method is congenitally deficient, so that it is hard to substantially surpass the best results in unsupervised domain adaptation benchmarks.

4 Method

According to the descriptions in the previous section, there are two problems when the GAN adversarial learning method is used in domain adaptation task. In this section, we first intorduce Wasserstein GAN technique to address the training unstability problem. Then we use self-ensembling method to tackle the feature conditional distribution misalignment problem. In the following subsections, we will introduce our method in details.

4.1 Domain confusion using Wasserstein-GAN

Instead of using GAN-form domain confusion method with training unstability problem, following the works of Arjovsky et al. [2], we now introduce the advantages of Earth-Mover distance (Wasserstein distance) in domain adaptation task. Let Π be the sets of all joint distributions $\gamma(x, y)$ whose marginal distributions are $p(F_s)$ and $p(F_t)$. Here are the Wasserstein distance:

$$W(p(F_s), p(F_t)) = \inf_{\gamma \sim \Pi(p(F_s), p(F_t))} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||] \tag{5}$$

Compared with Jensen Shannon (JS) divergence which has been discussed in section 3.1, Wasserstein distance is still able to reflect the distance between two distributions without overlap of their manifolds [2]. For this purpose, we choose Wasserstein distance as the measurement of two distributions $p(F_s)$

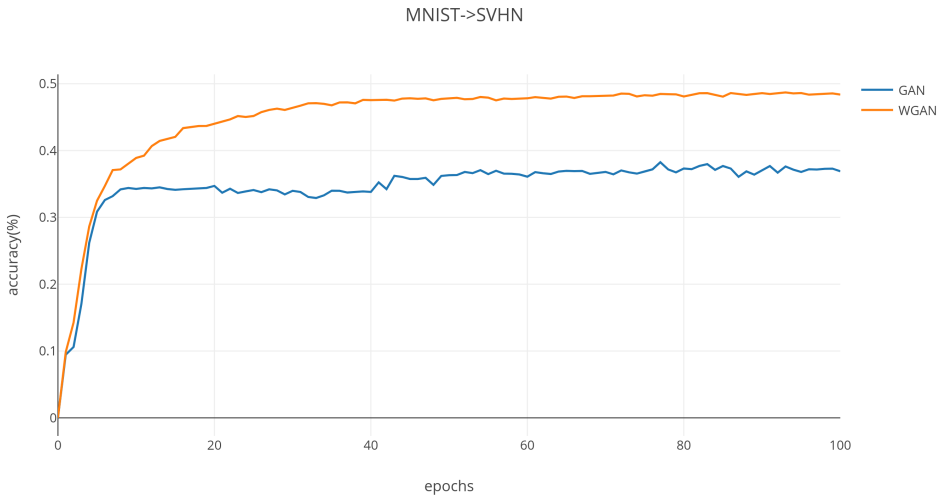


Figure 3: GAN loss vs WGAN loss training process

and $p(F_t)$. This superior property can make the feature extractor trained stably. Thus by the way of minimizing the Wasserstein distance between feature distributions, we can reduce the shift of datasets. Here is the WGAN loss function [9]:

$$\min_D \mathbb{E}_{x\sim p_t}[D(F(x))] - \mathbb{E}_{x\sim p_s}[D(F(x))] + \lambda \mathbb{E}_{x\sim\Omega}[(||\nabla_x D(x)||_2 - K)^2] \quad (6)$$

$$\max_F \mathbb{E}_{x\sim p_t}[D(F(x))] \quad (7)$$

In this case, we call it critic instead of discriminator when we use WGAN. But for convenience, the critic is still wrote as D in the formula. The last term of Formula (6) claims the parameters of critic must obey K-Lipschitz continuity. The hyperparameters λ and K are empirically set to 10 and 1 respectively in our expriments, which just simply follow the setting of Gulrajani et al. [9]. We find it very robust after we experienced this WGAN loss in many different settings, see Figure 3 as a case. Impressively, now we can use domain confusion technique easily without considering stability problem. This characteristic allows us making improvements based on this WGAN model.

4.2 Domain confusion with self-ensembling

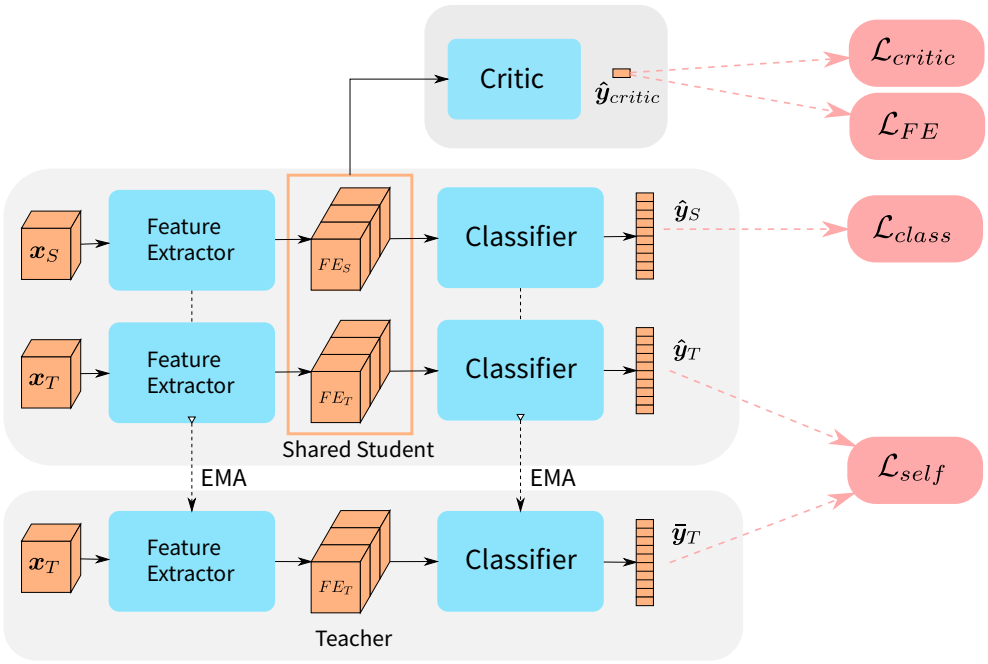


Figure 4: DCSE model architecture. The dashed line means weight shared, and the EMA means the teacher's parameters is the exponential moving average of the student's parameters. $\mathcal{L}_{critic}$ and $\mathcal{L}_F$ is the WGAN loss for update $Critic$ and F. $\mathcal{L}_{class}$ is the classification loss of source domain images. $\mathcal{L}_{se}$ is the self-ensembling loss of $\hat{\boldsymbol{y}}_t$ and $\bar{\boldsymbol{y}}_t$.

Self-ensembling model builds upon WGAN domain confusion model which is mentioned previously. In general, in order to solve feature conditional distribution misalignment problem, we add a self-ensembling loss on the basis of classification loss and WGAN loss, which is given by adding a teacher structure. Detailed model is shown in Figure 4.

4.2.1 Self-Ensembling technique

Following the works of French et al. [4], first we constrct a student network and a teacher network which are share the same architecture. The teacher's parameters t_i are weighted by the student's s_i using formula $t_i = \alpha t_{i-1} + (1-\alpha)s_i$ over every batch during training, α is set with 0.99. We use

self-ensembling label given by the teacher network to construct class label consistency loss, which guides the target domain distribution $p(F_t|X_t^i)$ to align with the source domain feature representation $p(F_s|X_s^i)$. The way to caculate self-ensembling loss is, for each target domain unlabeled sample $\boldsymbol{x}_t^i$, let $\boldsymbol{x}_t^i$ pass through the student network with data augmentation and dropout in the network, while without any modification for the teacher side. Next we can obtain their predicted vector $\hat{\boldsymbol{y}}_t^{ij}$ and pseudo predicted vector $\bar{\boldsymbol{y}}_t^{ij}$, where j indicates the j^{th} predicted class probability. Then we compute their self-ensembling loss of each sample, precisely, mean square errors. At last, we compute the confidence $\bar{\boldsymbol{c}}_t^i = \max_j \bar{\boldsymbol{y}}_t^{ij}$ of the i^{th} sample. If $\bar{\boldsymbol{c}}_t^i$ is below the confidence threshold of 0.968, the self-ensembling loss for the i^{th} sample $\boldsymbol{x}_t^i$ will be masked to 0, which follows the setting of French et al. [4].

Self-ensembling technique we used here can be treated as soft-clustering algorithm. The teacher in our model can be seen as an ensemble model weighted over time. So the samples with high confidence prediction from teacher are more likely to be classified correctly. Therefore their predictions can be used as pseudo labels of the student. Moreover, the noisy student can be seen as a regularizer to smooth the decision boundary in feature space. For example, when images from target domain pass a teacher network then get high confidence outputs, we believe the student's features manifold should have high probability of being around the teacher's, because the data augmentation and dropout won't actually change the category of images. When we push the noisy student's outputs as similar as the teacher's, the features with same category would be clustered. This discussion exactly follows the manifold assumption, wherein unlabeled data is able to make the feature space more dense. It facilitates the local catagory division in the feature space and helps the decision boundary perform data fitting more satisfactorily.

4.2.2 Class balance

As French et al. [4] demonstrated, the challenging MNIST $\leftrightarrow$ SVHN benchmarks remain undefeated because of the training unstability problem of normal self-ensembling technique. During the model training without using class balance, the error rate first decreases and then rises to high values before training stops. The authors hypothesize class imbalance in the SVHN dataset cause the fact that the self-ensembling pseudo labels always predict the "1"

class more often than the others, which gives rise to the model degenerate in local minimum.

They addressed this problem by using a class balance technique that average different class loss in each mini-batch. For example, let $\hat{y}_t^i$ indicate the class probability of the i^{th} target domain sample, then compute its mean μ_t by average $\hat{y}_t^i$ over the batch dimension. The class balance loss is computed as the binary cross entropy between the μ_t probability vector and a uniform probability vector. At last, they consider the self-ensembling loss is only applied to the confident training samples, the class balance loss should be weighted by the ratio of confident samples to all training samples. By the way, we also weight it with 0.005 for not overwhelming the other loss. We use this technique in MNIST $\leftrightarrow$ SVHN experiments.

4.2.3 Noise & Data augmentation

Recent works have showed that using noise and data augmentation in self-ensembling model can significantly improve the model generalization capability [13, 32]. Our goal, then, is to add more priors to expand the manifolds of data points in the input space or feature space. Thus the model can force decision boundaries pass through the sparse data manifolds rather than the dense data manifolds. This technique can avoid the wrong predictions leading the model to degenerate performance, which helps the teacher model reduce the misclassification on target domain. Therefore, we apply 5 augmentation strategies in our DCSE algorithm: 1) Using dropout on student model with random drop rates of 0.5. 2) Random brightness in the range of [0.7, 1.3]. 3) Random contrast in the range of [0.7, 1.3]. 4) Random saturation in the range of [0.7, 1.3]. 5) Random affine augmentation using affine transformation matrix which is shown as follows [4].

$$\begin{bmatrix} 1+\mathcal{N}(0,0.1) & \mathcal{N}(0,0.1) & 0 \\ \mathcal{N}(0,0.1) & 1+\mathcal{N}(0,0.1) & 0 \\ \mathcal{U}(-2,2) & \mathcal{U}(-2,2) & 1 \end{bmatrix} \tag{8}$$

4.3 Functions of different losses

To stablize the training process, we analyze the effects of different losses in details and propose an adaptive technique to address it. In our domain adaptation problem, we have the labeled images in source domain and unla-

beled images in target domain. In training process, we design three different losses, which are cross entropy as classification loss in supervised setting as usual, domain confusion loss for aligning source and target domains feature representation, and self-ensembling loss for aligning each feature conditional distribution $p(F|X^i)$ between domains. We think our self-ensembling model works by relying on model assumption: when the model assumption is positive, unlabeled samples will help to improve model performance [25, 13]. In our case, when supervised term leads the model to a great model assumption in source domain, the first unsupervised term domain confusion loss will help the classifier much easier to classify the target domain images by aligning the feature distribution. The second unsupervised term self-ensembling loss makes the model gaining a much better result by aligning feature conditional distribution.

Based on the above analysis, we find that these losses play different roles in different training stages. First, following the previous analysis in Section 3.2, we consider using source domain classification loss to construct $p(Y_s|X_s^i)$ distribution along the whole training phase. Second, domain confusion loss should lead the model to a feasible assumption in the early stage of training, so that it can help self-ensembling loss leading the model to a better performance. In addition, because different datasets are suitable for different weights of domain confusion loss, we apply a weight decayed factor λ_{dc} using cosine ramp-down method [14]. Thus the factor λ_{dc} would gradually decrease till the zero during training. Then, to tackle the feature conditional distribution misalignment problem, we introduce self-ensembling loss. Due to the fact that it strongly relys on the self-ensembling pseudo labels which should be relatively accurate enough, we apply it to align feature conditional distribution when pseudo predict probability is larger than 0.968, which makes it become zero in the early unstable training stage. Furthermore, we also set a factor λ_{se} to weight the self-ensembling loss for fitting different datasets. The program flow of our proposed approach DCSE is described in Algorithm 1.

5 Experiments

Our implementation was developed using Pytorch [21].

Algorithm 1 DCSE, our proposed algorithm.

Require: epoch, the iteration number. lr, the learning rates. m, the batch size. λ_{dc}, the decayed factor for domain confusion loss. λ_{se}, the factor for self-ensembling loss. **Require**: Initialize the parameters of F_{Stu}, F_{Tch}, $Classifier_{Stu}$, $Classifier_{Tch}$, $Critic$.

for e=1,...,epoch **do**
  Sample $\boldsymbol{X}_s$, $\boldsymbol{Y}_s$ a batch from source domain
  Sample $\boldsymbol{X}_t$ a batch from target domain
  $\boldsymbol{F}_s$, $\boldsymbol{F}_t = F_{Stu}(\boldsymbol{X}_s, \boldsymbol{X}_t)$
  $\hat{\boldsymbol{y}}_s$, $\hat{\boldsymbol{y}}_t = Classifier_{Stu}(\boldsymbol{F}_s, \boldsymbol{F}_t)$
  $\mathcal{L}_{class}$ = ClassificationLoss($\hat{\boldsymbol{y}}_s$, $\boldsymbol{Y}_s$)
  $\lambda_{dc} \leftarrow \lambda_{dc}$ * CosineRampdown(e)
  $\hat{\boldsymbol{y}}_{critic_s}$, $\hat{\boldsymbol{y}}_{critic_t} = Critic(\boldsymbol{F}_s, \boldsymbol{F}_t)$
  $\mathcal{L}_{critic}$ = CriticLoss($\hat{\boldsymbol{y}}_{critic_s}$, $\hat{\boldsymbol{y}}_{critic_t}$, $\boldsymbol{F}_s$, $\boldsymbol{F}_t$) $\cdot$ λ_{dc}
  Use $\mathcal{L}_{critic}$ to update the $critic$ network
  $\hat{\boldsymbol{y}}_{critic_t} = Critic(\boldsymbol{F}_t)$
  $\mathcal{L}_F$ = -$\hat{\boldsymbol{y}}_{critic_t}$.mean() $\cdot$ λ_{dc}
  $\bar{\boldsymbol{y}}_t = Classifier_{Tch}(F_{Tch}(\boldsymbol{X}_t))$
  $\mathcal{L}_{se}$ = SelfEnsemblingLoss($\hat{\boldsymbol{y}}_t$, $\bar{\boldsymbol{y}}_t$) * λ_{se}
  Use $\mathcal{L}_{class}$, $\mathcal{L}_F$, $\mathcal{L}_{se}$ to update model
  EMA:$[F_{Tch}, C_{Tch}] \leftarrow 0.99$ * $[F_{Tch}, C_{Tch}] + 0.01$ * $[F_{Stu}, C_{Stu}]$
end for

5.1 Setting

Figure 4 shows a student network and a teacher network, which recept source data and target data as input, then output the class probability predictions. The training losses are the sum of supervised term and unsupervised term. Supervised term consists of classification loss which uses the cross entropy as the loss function for labeled source images. Unsupervised term consists of WGAN loss and self-ensembling loss. WGAN loss is used for confusing source and target domain feature representation. Self-ensembling loss is computed as mean square error between $\hat{\boldsymbol{y}}_t$ and $\bar{\boldsymbol{y}}_t$ when $\bar{\boldsymbol{c}}_t^i$ exceeds the threshold 0.968. It is worth noting that gradient descent only applies in student network. And the teacher is the exponential moving average of weights of the student over training batches. We test the results in experiments by teacher network. More details of the model architecture see table 1,2.

5.2 Hyperparameters

In all experiments, we used RMSprop optimizer with learning rates of 0.001 to execute gradient descent. The mini-batches of our algorithm composed of 128 samples. For hyperparameters λ_{dc} and λ_{se}, we have not found an applicable way to optimize them in unsupervised domain adaptation, because there is no target domain labels can be used for evaluation, which is still an open research question, and out of scope in our study. So here we just use the test sets example for evaluation and our experimental accuracy can be seen as upper bounds of our algorithm. Besides, because the other researches also have the same problem, our results are fair and comparable. The hyperparameters are shown in Table 3.

Table 3: Hyperparameters for different adaptation paths

	λ_{dc}	λ_{se}
MNIST $\rightarrow$ SVHN	3	5
SVHN $\rightarrow$ MNIST	0.3	3
CIFAR $\rightarrow$ STL	0.05	1
STL $\rightarrow$ CIFAR	0.1	1

Table 4: Four datasets benchmark in domain adaptation task, each result is shown as accuracy(%).

	MNIST - SVHN	SVHN - MNIST	CIFAR - STL	STL - CIFAR
RevGrad [5]	35.67	73.91	66.12	56.91
DRCN [7]	40.05	81.97	66.37	58.65
SE [4]	41.98	99.22	75.51	69.15
SBADA-GAN [23]	61.1	76.1	-	-
DC (ours)	48.65	66.10	71.38	59.83
SE (ours)	32.80	99.23	77.68	66.73
DCSE (ours)	**83.65**	**99.53**	**78.64**	**72.98**

5.3 Datasets

All results can be seen in Table 4. We evaluated our algorithm over 4 cross domain task pairs:1)MNIST $\rightarrow$ SVHN, 2)SVHN $\rightarrow$ MNIST, 3)CIFAR $\rightarrow$ STL, 4)STL $\rightarrow$ CIFAR.

MNIST [16] is a grayscale handwritten digit dataset (see Figure 5a). All images were converted to RGB channels for matching the colorful dataset SVHN (see Figure 5b) [17]. They were rescaled to 32 $\times$ 32 RGB images, and applied by [0,1] normalization, which forms MNIST 60,000 images for training and 10,000 for testing, 73,257 labeled SVHN for training, and 26,032 for testing.

CIFAR (see Figure 5c) [11] and STL (see Figure 5d) [1] consist of RGB images that share nine object classes:plane, car, bird, cat, deer, dog, horse, ship, truck. There are 45000 samples for training and 9000 for testing in CIFAR while STL only has 4500 for training and 7,200 for testing. They were all scaled to 32 $\times$ 32 images and also were applied by [0,1] normalization.

Our experiments aim at the most challenging datasets in small image domain adaptation task. For the convenience of comparison, in each of these cases, we applied inductive unsupervised learning in domain adaptation task like French et al. did [4]. Only the training sets were used during training, while the test sets only were used for reporting accuracy. By the way, all the results were obtained by the same architecture, as shown in Table 1,2.

MNIST $\rightarrow$ SVHN, This adaptation path is a difficult task for the reason

(a) MNIST (b) SVHN

(c) CIFAR (d) STL

Figure 5: Domain adaptation example images

that the model must learn knowledge from gray-scale and generalize it to varicolored images. We first used Wasserstein GAN as domain confusion technique to mitigate training unstable problem, achieving 48.65% accuracy. For the sake of solving class misalignment problem, we introduced self-ensembling model to guide the target class feature representation as similar as possible to the source. With this modification, we achieved a result that strongly outperforms prior art from 61.1% to 83.65%. To compare with self-ensembling based model, we only used classification loss and self-ensembling loss, which resulted in 32.80% accuracy only. It is worth noting that the result of our approach DCSE is highly surpass the domain confusion method. It also confirms the feature misalignment problem in unsupervised domain adaptation task, which has a huge effect on the model performance. However, self-ensembling loss can partly solve this problem by generating pseudo predictions to correct the feature conditional distribution.

SVHN → MNIST In this adaptation direction, we found our algorithm DCSE sometimes may unstable at the beginning of training. We diagnosed the problem by removing the domain confusion loss, observing the model can be trained stably and got 99.23% accuracy. We hypothesized that domain confusion loss may lead the model to a poor model assumption at the initial stage, so the self-ensembling pseudo label is highly possible incorrect, which leads the model to degenerate. We overcame this problem by linear ramping up the λ_{se} at the first 30 epochs with interval 0.1. With this modification, we significantly surpassed the domain-alignment techniques like DC (ours) and RevGrad [5] and self-ensembling techniques with 99.53% accuracy. To the best of our knowledges, this is a state-of-the-art result in domain adaptation setting, which is close to the supervised results.

CIFAR → STL The gaps in CIFAR → STL path are greater than the gaps

in MNIST $\leftrightarrow$ SVHN path since the former looks much more morphological changes than the latter. Furthermore, The source domain CIFAR has 45,000 images to perform supervised training, while unlabeled target domain STL only has 4,500 images. We obtained a strong performance with 78.64% in this experiment which surpassed the prior art. For fair comparison, we removed the WGAN loss and got 77.68% accuracy. This shows that our approach can stably outperforms the results of SE-only approaches. Furthermore, based on the analysis of model assumption, it also proves the fact that domain-alignment technique can help the model reach a better model assumption. As a result, self-ensembling technique can lead the model to a higher performance.

STL $\rightarrow$ CIFAR In this experiment, we achieved 72.98% accuracy. This result significantly exceeds the accuracy of SE model using self-ensembling method only. We consider that WGAN loss can gain much more informations by aligning feature distribution since STL has only 4,500 images be trained supervisely, which leads the model arrive in a better model assumption. Thus the unlabeled images are able to help the model become more accurate by self-ensembling.

6 Conclusions

In this paper, we have proposed an algorithm named as DCSE by combining domain confusion and self-ensembling, which aims at addressing training unstability problem and feature conditional distribution misalignment problem. Our model works by finding an unify class representation between domains, which presents better results in MNIST $\leftrightarrow$ SVHN and CIFAR $\leftrightarrow$ STL cross domain adaptation benchmarks.

Acknowledgements

The authors are grateful to all reviewers for their very insightful comments and suggestions. This work was supported in part by National Natural Science Foundation of China under Grants 61773127 and 61727810, Ten Thousand Talent Program approved in 2018, Scientific Funds approved in 2016 for Higher Level Talents by Guangdong Provincial universities Grant 2015TX01X232 and Project supported by GDHVPS 2014, Guangdong Province Science Foundation for Program of Research Team under Grant 2018A030313306, and

Guangzhou Science and Technology Foundation under Grants 201802010037 and 20150810007.

References

[1] Andrew Y. Ng Adam Coates, Honglak Lee. An analysis of single layer networks in unsupervised feature learning. *AISTATS*, 2011.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *ArXiv e-prints*, January 2017.

[3] Bo Du, Liangpei Zhang, Dacheng Tao, and Dengyi Zhang. Unsupervised transfer learning for target detection from hyperspectral images. *Neurocomputing*, 120:72 – 82, 2013. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2012.08.056. URL `http://www.sciencedirect.com/science/article/pii/S092523121300297X`.

[4] Geoffrey French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. *ArXiv e-prints 1706.05208*, 2017.

[5] Y. Ganin and V. Lempitsky. Unsupervised Domain Adaptation by Backpropagation. *ArXiv e-prints*, September 2014.

[6] Marzieh Gheisari and Mahdieh Soleymani Baghshah. Unsupervised domain adaptation via representation learning and adaptive classifier learning. *Neurocomputing*, 165:300 – 311, 2015. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2015.03.020. URL `http://www.sciencedirect.com/science/article/pii/S0925231215002921`.

[7] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *Proceedings of European Conference on Computer Vision*, pages 597–613, 2016.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. 2014. URL `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

[9] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of Wasserstein GANs.

Computing Research Repository, abs/1704.00028, 2017. URL http://arxiv.org/abs/1704.00028.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[11] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[13] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *Computing Research Repository*, abs/1610.02242, 2016. URL http://arxiv.org/abs/1610.02242.

[14] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. *Computing Research Repository*, abs/1608.03983, 2016. URL http://arxiv.org/abs/1608.03983.

[15] M.-T. Luong, H. Pham, and C. D. Manning. Effective Approaches to Attention-based Neural Machine Translation. *ArXiv e-prints*, August 2015.

[16] Yann Lcun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

[17] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[18] Jianhan Pan, Xuegang Hu, Peipei Li, Huizong Li, Wei He, Yuhong Zhang, and Yaojin Lin. Domain adaptation via multi-layer transfer learning.

Neurocomputing, 190:10 – 24, 2016. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2015.12.097. URL http://www.sciencedirect.com/science/article/pii/S0925231216000096.

[19] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[20] Weike Pan. A survey of transfer learning for collaborative recommendation with auxiliary data. *Neurocomputing*, 177:447 – 453, 2016. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2015.11.059. URL http://www.sciencedirect.com/science/article/pii/S0925231215018640.

[21] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, and Trevor Darrell. Pytorch. https://github.com/pytorch/pytorch, 2017.

[22] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2014.

[23] Paolo Russo, Fabio Maria Carlucci, Tatiana Tommasi, and Barbara Caputo. From source to target and back: symmetric bi-directional adaptive GAN. *Computing Research Repository*, abs/1705.08824, 2017. URL http://arxiv.org/abs/1705.08824.

[24] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *Computing Research Repository*, abs/1606.04671, 2016. URL http://arxiv.org/abs/1606.04671.

[25] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Computing Research Repository*, abs/1606.04586, 2016. URL http://arxiv.org/abs/1606.04586.

[26] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv e-prints*, September 2014.

[27] Xin Sun, Junyu Shi, Lipeng Liu, Junyu Dong, Claudia Plant, Xinhua Wang, and Huiyu Zhou. Transferring deep knowledge for object recognition in low-quality underwater videos. *Neurocomputing*, 275: 897 – 908, 2018. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2017.09.044. URL http://www.sciencedirect.com/science/article/pii/S0925231217315631.

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *Computing Research Repository*, abs/1409.4842, 2014. URL http://arxiv.org/abs/1409.4842.

[29] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *Computing Research Repository*, abs/1804.10332, 2018. URL http://arxiv.org/abs/1804.10332.

[30] Qiaoyu Tan, Yanming Yu, Guoxian Yu, and Jun Wang. Semi-supervised multi-label classification using incomplete label information. *Neurocomputing*, 260:192 – 202, 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2017.04.033. URL http://www.sciencedirect.com/science/article/pii/S092523121730704X.

[31] Xin Tang, Fang Guo, Jianbing Shen, and Tianyuan Du. Facial landmark detection by semi-supervised deep learning. *Neurocomputing*, 2018. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2018.01.080. URL http://www.sciencedirect.com/science/article/pii/S0925231218301139.

[32] Antti Tarvainen and Harri Valpola. Weight-averaged consistency targets improve semi-supervised deep learning results. *Computing Research Repository*, abs/1703.01780, 2017. URL http://arxiv.org/abs/1703.01780.

[33] Alessio Tonioni, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Unsupervised adaptation for deep stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1614–1622, 2017.

[34] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *Computing Research Repository*, abs/1412.3474, 2014. URL `http://arxiv.org/abs/1412.3474`.

[35] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. *Computing Research Repository*, abs/1702.05464, 2017. URL `http://arxiv.org/abs/1702.05464`.

[36] Jing Wang, Xin Zhang, Xueqing Li, and Jixiang Du. Semi-supervised manifold alignment with few correspondences. *Neurocomputing*, 230: 322 – 331, 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2016.12.010. URL `http://www.sciencedirect.com/science/article/pii/S0925231216314850`.

[37] Baodi Yuan, Jian Tu, Rui-Wei Zhao, Yingbin Zheng, and Yu-Gang Jiang. Learning part-based mid-level representation for visual recognition. *Neurocomputing*, 275:2126 – 2136, 2018. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2017.10.062. URL `http://www.sciencedirect.com/science/article/pii/S0925231217317137`.