Multi-Source Domain Adaptation with Fuzzy-Rule based Deep Neural Networks

1st Keqiuyin Li
Faculty of Engineering and IT
University of Technology Sydney
Sydney, NSW, Australia
keqiuyin.li@student.uts.edu.au

2nd Jie Lu
Faculty of Engineering and IT
University of Technology Sydney
Sydney, NSW, Australia
jie.lu@uts.edu.au

3rd Hua Zuo
Faculty of Engineering and IT
University of Technology Sydney
Sydney, NSW, Australia
hua.zuo@uts.edu.au

4th Guangquan Zhang
Faculty of Engineering and IT
University of Technology Sydney
Sydney, NSW, Australia
guangquan.zhang@uts.edu.au

2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) | 978-1-6654-4407-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/FUZZ45933.2021.9494586

***Abstract*—Unsupervised domain adaptation provides a variety of methods to leverage the previously gained knowledge from a labeled source domain to help complete a task from a similar unlabeled target domain. Many existing methods focus on transferring knowledge across single source and single target domains, while few studies deal with multi-source domain adaptation, which is more realistic and challengeable. Existing multi-source domain adaptation methods rarely consider the uncertainty of the transformed knowledge resulting from limited information in target domain. A fuzzy system allows imprecision and ambiguity within transfer, thus it can deal with problems with uncertainty. This work proposes a multi-source domain adaptation method with fuzzy-rule based deep neural networks (MDAFuz). The proposed method first extracts multi-view adapted features and pretrains source classifiers. Using the learned features and classifiers, training samples are then split into multiple clusters, hence fuzzy rules can be built to learn new classifiers. At the same time, the cluster discriminator is trained to define the membership. Finally, by measuring the similarities among source and target domains using the pseudo target labels and a domain discriminator, the target task is completed by combining all source classifiers with regard to the learned weights. The experiment results on real-world visual datasets show the superiority of the proposed method.**

I. Introduction

Traditional deep learning requires that the training and testing data must be drawn from the same distribution, and there is a large quantity of training data to learn the model. However, these requirements cannot always be satisfied in many applications due to the high cost or the privacy issue, which means that the acquired labeled training data (source) and the unlabeled testing data (target) probably follow different distributions. Thus, transfer learning gains increasing attention in view of its capability to handle the dataset bias problem [1]. Domain adaptation is attractive and well-explored in transfer learning, including closed set [2], open-set [3], [4] and partial domain adaptation [5]. A widely studied approach is feature-based domain adaptation, which aims at handling domain shift in a latent feature space via reducing discrepancy between the source and target domains. Maximum mean discrepancy (MMD) [6] and Wasserstein distance [7] are two popular and regularly used distribution measurements, and adversarial training [8] captures considerable attention by building a two-player network to make source and target domains non-discriminatory.

According to the dimensions of the feature spaces of source and target domains, the feature-based approach can be divided into homogeneous and heterogeneous domain adaptation. In homogeneous domain adaptation, the source and target feature spaces have the same dimension. Structurally regularized deep clustering proposes a source regularized method for unsupervised domain adaption [9]. Motivated by the structural similarity, it employs a deep clustering framework to learn class centres of source and target domains, and generates an auxiliary target distribution to help explore the intrinsic discrimination in the target domain by matching it to the source distribution. Certainty-based attention for domain adaptation identifies adaptable regions by building a Bayesian discriminator [10]. The predominant areas that can benefit the matching of source and target data are highlighted by the class probabilities returned using a Bayesian classifier.

In heterogeneous domain adaptation, the source and target domains have different feature space dimensions. Completely heterogeneous transfer learning deals with domain adaptation where both the feature and label spaces of the source and target domains are different [11]. It discovers what and what not to transfer by selecting a subset of source samples, and the attention mechanism is employed to learn a set of weight vectors and determine its correlation with target domain. Deep matrix completion with adversarial kernel embedding employs an adversarial manner to learn the distribution kernel embedding in a latent feature space, and uses it to map distributions and train the classifiers [12].

However, the mentioned studies rarely consider the inherent

This work was supported by the Australian Research Council under grant FL190100149.(Corresponding author: Jie Lu.)

phenomenon of uncertainty in knowledge transfer [13]. Since target labels are inaccessible, there is a limit to the amount of information with certainty that can be extracted, causing a high level of uncertainty in the target domain. To address this problem, fuzzy logic is introduced to transfer learning [14]–[16]. Fuzzy multiple-source transfer learning deals with regression tasks in both homogeneous and heterogeneous scenarios with multiples source domains [17]. It determines dominant source domains which contain more suitable transferable information for the given target domain by measuring the distance between each source and target class centres. Multi-source heterogeneous unsupervised domain adaptation extracts shared information from multi-dimension spaces using a novel shared-fuzzy-equivalence-relations neural network [18], and then transforms the acquired shared fuzzy knowledge into latent feature spaces to match the distribution discrepancy among heterogeneous domains.

The aforementioned existing fuzzy transfer learning methods focus on shallow neural networks. In this paper, we propose a multi-source domain adaptation method which adds fuzzy rules to deep neural networks. In the proposed method, a pre-trained network is used first to extract multi-view features of the source and target data by adapting multi-level domain distributions. Then the fuzzy sets and fuzzy rules are built based on the extracted features and the similarities predicted by the pre-trained classifiers to construct new source classifiers. Finally, all source classifiers are combined using fuzzy membership returned by a learned domain discriminator to complete the target task. Our main contributions are as follows:

- We propose a multi-source domain adaptation method with a fuzzy system, which applies fuzzy logic into very deep neural networks. It differs from previous fuzzy transfer learning which focuses on shallow models;
- We use a similarity estimation strategy to group source samples that contain similar information into multiple clusters and build the fuzzy rules. A cluster discriminator is learned to split samples according to the information level which rarely has been considered as a factor affecting performance;
- We develop a fuzzy combination rule for conjoining source classifiers to predict target labels. This is the first study to employ fuzzy membership to define the source contribution to the target task.

The remainder of this paper is designed as follows: Section II details the proposed method. Section III presents the experiment results and analyses on real-world visual datasets. Conclusion and future work are given in section IV.

II. The Proposed Method

In this paper, we deal with homogeneous unsupervised multi-source domain adaptation. The proposed method is detailed as follows: first, it employs a deep network to extract transferrable features in latent feature spaces and pre-trains source classifiers. Both domain-level and class-level distributions are considered to match the source and target domains. Then, the pre-trained classifiers estimate the similarity of a source sample belonging to a category. By dividing the estimated similarities into different groups, training samples in each source domain are split into multiple clusters, thus fuzzy rules are built to learn new source classifiers. Finally, all source classifiers are combined to complete the target task. In order to learn the combination weights, fuzzy membership is estimated using the domain discriminator. The whole framework is shown in Fig. 1.

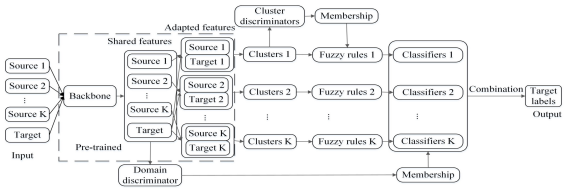

Fig. 1. The procedure of the proposed method.

A. Transferrable Feature Extraction and Source Classifier Pre-training

In this section, we pre-train the source classifiers and collect the transferrable features using our previous study [19]. Given multiple source domains $\{\mathcal{D}_{s_k}\}_{k=1}^{K}$ following distributions $\{\mathcal{P}_{s_k}\}_{k=1}^{K}$, where the kth source domain $\mathcal{D}_{s_k} = \{(\boldsymbol{x}_{s_k}^i, \boldsymbol{y}_{s_k}^i)\}_{i=1}^{n_{s_k}}$, and the target domain $\mathcal{D}_t = \{\boldsymbol{x}_t^j\}_{j=1}^{n_t}$ following distribution $\mathcal{P}_t$, $\boldsymbol{x}_{s_k}, \boldsymbol{x}_t \in \mathcal{X}$ represent samples, $\boldsymbol{y}_{s_k} \in \mathcal{Y}$ indicates corresponding label of $\boldsymbol{x}_{s_k}$, n_{s_k}, n_t indicate the number of samples in the kth source domain and target domain, respectively. With regard to the structural risk minimization principle [20], the learning processing of each source classifier P_{s_k} can be written as:

$$P_{s_k} = \underset{\substack{P_{s_k}\in\mathcal{H} \\ (\boldsymbol{x}_{s_k},\boldsymbol{y}_{s_k})\sim\mathcal{D}_{s_k}}}{\arg\min} \; L(P_{s_k}(\phi_k(\boldsymbol{x}_{s_k})), \boldsymbol{y}_{s_k}) + \lambda R. \quad (1)$$

L is the error between the predicted outputs and source labels, where cross-entropy loss is used to calculate the error:

$$L = -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \boldsymbol{y}_{s_k}^i \log(P_{s_k}(\phi_k(\boldsymbol{x}_{s_k})).$$

ϕ_k is the feature extraction operation, R indicates the regularization term, $\mathcal{H}$ represents reproducing kernel Hilbert space and λ is an optimal trade-off parameter. During training, n_{s_k} is replaced with the batch size. This operation applies to other equations in this paper.

In our setting, source classifiers are trained to be used on target data. To provide the cross-domain ability of the learned classifiers, here we adapt the distributions between each source and target domains, hence the extracted features of the source and target domains will follow similar distributions, meaning the classifiers can be applied to both of them.

To collect features for adapting domain distributions, for kth source domain, the feature extraction operation ϕ_k contains shared feature extraction ϕ and specific multi-view feature

extraction ϕ_{c_k} and ϕ_{d_k}. Employing a pre-trained deep neural network structure, the formula of shared feature extraction is:

$$\boldsymbol{f}_{s_k}^i = \phi(\boldsymbol{x}_{s_k}^i, \boldsymbol{\theta}), \boldsymbol{f}_t^j = \phi(\boldsymbol{x}_t^j, \boldsymbol{\theta}),$$
$$i = 1, 2, \ldots, n_{s_k}, j = 1, 2, \ldots, n_t, k = 1, 2, \ldots, K,$$

$\boldsymbol{\theta}$ represents the parameter of deep network ϕ.

To represent data from different aspects, the shared features are then divided into common view and diverse view features. Corresponding feature extraction can be expressed as:

$$\boldsymbol{f}_{c_{s_k}}^i = \phi_{c_k}(\boldsymbol{f}_{s_k}^i, \boldsymbol{\theta}_{c_k}), \boldsymbol{f}_{c_{t_k}}^j = \phi_{c_k}(\boldsymbol{f}_t^j, \boldsymbol{\theta}_{c_k}),$$
$$\boldsymbol{f}_{d_{s_k}}^i = \phi_{d_k}(\boldsymbol{f}_{s_k}^i, \boldsymbol{\theta}_{d_k}), \boldsymbol{f}_{d_{t_k}}^j = \phi_{d_k}(\boldsymbol{f}_t^j, \boldsymbol{\theta}_{d_k}),$$
$$i = 1, 2, \ldots, n_{s_k}, j = 1, 2, \ldots, n_t, k = 1, 2, \ldots, K,$$

$\boldsymbol{\theta}_{c_k}, \boldsymbol{\theta}_{d_k}$ are the corresponding parameters of ϕ_{c_k}, ϕ_{d_k}. Features for adaptation and classification can then be written as $\boldsymbol{F}_{s_k} = [\boldsymbol{f}_{c_{s_k}}; \boldsymbol{f}_{d_{t_k}}]$, $\boldsymbol{F}_{t_k} = [\boldsymbol{f}_{c_{t_k}}; \boldsymbol{f}_{d_{t_k}}]$.

MMD is used to minimize the discrepancy between the source and target distributions [6]. Both domain-level and class-level distributions are considered. Domain-level matching can be expressed as:

$$\begin{aligned} L_d &= \mathcal{MMD}(\mathcal{H}, \mathcal{P}_{s_k}, \mathcal{P}_t) \\ &= \sup_{\psi \in \mathcal{H}} (\mathbb{E}_{\boldsymbol{F}_{s_k} \sim \mathcal{P}_{s_k}} \psi(\boldsymbol{F}_{s_k}) - \mathbb{E}_{\boldsymbol{F}_{t_k} \sim \mathcal{P}_t} \psi(\boldsymbol{F}_{t_k})). \end{aligned}$$

Employing the same strategy proposed in [21], MMD can be calculated as:

$$\mathcal{MMD}(\mathcal{H}, \mathcal{P}_{s_k}, \mathcal{P}_t) = \left\| \frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \psi(\boldsymbol{F}_{s_k}^i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \psi(\boldsymbol{F}_{t_k}^j) \right\|_{\mathcal{H}}^2,$$

ψ is a nonlinear function transforming the extracted features into RKHS with a universal kernel.

Class-level matching minimizes the discrepancy between the same classes and maximizes it among different classes. Since misalignment often occurs to boundary samples, we only enlarge the distance between the nearest two classes with the highest probability during training, which can reduce computational complexity. The class-level matching can be written as:

$$\begin{aligned} L_c = &\frac{1}{C} \sum_{r=1}^{C} \mathcal{MMD}(\mathcal{H}, \mathcal{P}_{s_k}^r, \mathcal{P}_t^r) - \\ &\big(\frac{\alpha}{2}(\mathcal{MMD}(\mathcal{H}, \mathcal{P}_{s_k}^{r_{s1}}, \mathcal{P}_{s_k}^{r_{s2}}) + \\ &\mathcal{MMD}(\mathcal{H}, \mathcal{P}_t^{r_{t1}}, \mathcal{P}_t^{r_{t2}}))\big), \end{aligned}$$

C is the number of classes, $s1, s2$ and $t1, t2$ are indices of the nearest class in the source and target domains. The solution of MMD is:

$$\mathcal{MMD}(\mathcal{H}, \mathcal{P}_{*_1}^{\star_1}, \mathcal{P}_{*_2}^{\star_2}) = \left\| \frac{1}{n_{*_1}^{\star_1}} \sum_{i=1}^{n_{*_1}^{\star_1}} \psi(p_{*_1}^{\star_1} \cdot \boldsymbol{F}_{*_1}^{\star_1}) - \frac{1}{n_{*_2}^{\star_2}} \sum_{j=1}^{n_{*_2}^{\star_2}} \psi(p_{*_2}^{\star_2} \cdot \boldsymbol{F}_{*_2}^{\star_2}) \right\|_{\mathcal{H}}^2,$$

$*$ and $\star$ denote the subscript and superscript in L_c, p is the probability of a sample belonging to a class.

The similarity and diversity among source domains should also be taken into account. We use the common view features to represent similarity and the diverse view features to indicate diversity, then the source domain adaptation can be expressed by minimizing the similarity and maximizing the diversity:

$$\begin{aligned} \mathcal{L}_s = &\frac{2}{K(K-1)} \sum_{k_1=1}^{K-1} \sum_{k_2=k_1+1}^{K} \\ &(\mathcal{MMD}(\mathcal{H}, \mathcal{P}_{s_{k_1}}^c, \mathcal{P}_{s_{k_2}}^c) - \mathcal{MMD}(\mathcal{H}, \mathcal{P}_{s_{k_1}}^d, \mathcal{P}_{s_{k_2}}^d)), \end{aligned}$$

where

$$\mathcal{MMD}(\mathcal{H}, \mathcal{P}_{s_{k_1}}^{\cdot}, \mathcal{P}_{s_{k_2}}^{\cdot}) = \left\| \frac{1}{n_{s_{k_1}}} \sum_{i=1}^{n_{s_{k_1}}} \psi(\boldsymbol{f}_{\cdot s_{k_1}}^i) - \frac{1}{n_{s_{k_2}}} \sum_{j=1}^{n_{s_{k_2}}} \psi(\boldsymbol{f}_{\cdot s_{k_2}}^j) \right\|_{\mathcal{H}}^2,$$

symbol $\cdot$ represents superscript c or d.

Cross-domain constraint is applied to ensure that multiple source classifiers return the same labels of the same target samples:

$$\begin{aligned} \mathcal{L}_{cro} = &\frac{2}{K(K-1)} \sum_{k_1=1}^{K-1} \sum_{k_2=k_1+1}^{K} \\ &\big(\frac{1}{n_t} \sum_{j=1}^{n_t} \big| P_{s_{k_1}}(\boldsymbol{F}_{t_{k_1}}^j) - P_{s_{k_2}}(\boldsymbol{F}_{t_{k_2}}^j) \big| \big). \end{aligned} \tag{2}$$

The source classifier in equation (1) can be re-written as:

$$\begin{aligned} P_{s_k} = &\mathop{\arg\min}_{\substack{P_{s_k} \in \mathcal{H} \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \sim \mathcal{D}_{s_k}}} L(P_{s_k}(\phi_k(\boldsymbol{x}_{s_k})), \boldsymbol{y}_{s_k}) \\ &+ \lambda_1 L_d + \lambda_2 L_c + \lambda_3 L_s + \lambda_4 L_{cro}. \end{aligned}$$

B. Fuzzy-rule based Classification

The Takagi–Sugeno fuzzy model is a popular fuzzy architecture. For data pair $(\boldsymbol{x}, \boldsymbol{y})$, the rule is:

$$\text{if } \boldsymbol{x} \text{ is } A_m, \text{then } \boldsymbol{y} \text{ is } P_m(\boldsymbol{x}), m = 1, 2, \cdots, M. \tag{3}$$

A_m is the fuzzy set of the mth rule, P_m is the corresponding output function. The output of the fuzzy system is expressed as:

$$\boldsymbol{y} = \sum_{m=1}^{M} p_m \cdot P_m(\boldsymbol{x}) \tag{4}$$

p_m is the membership of data belonging to a set.

In the classification task, the classifier can identify an item in different views, for example, front view, partial view, rotate view and so on. It cannot distinguish the different views of the item but only "remembers" its features during learning. The information level of the same item in different views is actually different, and samples with the same level information are more similar to each other compared with those with different level information. Hence, according to the information level, to construct a fuzzy model for classification, we divide the samples into multiple groups to learn the multiple classifiers

of each source domain, which is expected to benefit the classification.

Using the estimated similarity to represent the information level contained in a sample, the similarity of each sample belonging to the class in kth source domain can be estimated by the pre-trained classifier:

$$p_{s_k} = \max(P_{s_k}(\boldsymbol{F}_{s_k})), p_{s_k} \in [0, 1]$$

Divide the closed interval $[0, 1]$ into M sub-intervals, $[0, a_1)$, $\cdots$, $[a_{k-1}, a_k)$, $\cdots$, $[a_{M-1}, 1]$, the source samples are split into different clusters according to the value of the estimated similarity. For the mth cluster, a classifier $P_{s_{km}}$is trained by minimizing the cross-entropy loss:

$$L_m = -\frac{1}{n_{s_{km}}} \sum_{i=1}^{n_{s_{km}}} \boldsymbol{y}_{s_k}^i \log(P_{s_{km}}(\boldsymbol{F}_{s_k})).$$

$n_{s_{km}}$ is the number of cluster samples.

A cluster discriminator is trained using samples from each cluster to estimate the membership of new inputs. The cluster discriminator of the kth source domain P_{c_k} is parameterized by:

$$L_{Pc} = -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \boldsymbol{y}_{c_k}^i \log(P_{c_k}(\boldsymbol{F}_{s_k})).$$

$\boldsymbol{y}_{c_k}$ is cluster label. The membership vector is:

$$\boldsymbol{p}_{c_k} = P_{c_k}(\boldsymbol{F}_{s_k}).$$

The fuzzy model for each source domain in equation (3)-(4) can be re-written as:

if $\boldsymbol{F}_{s_k}$ *is* A_m, then $\boldsymbol{y}_{s_k}$ *is* $P_{s_{km}}(\boldsymbol{F}_{s_k}), m = 1, 2, \cdots, M.$

The prediction of kth source domain is expressed as:

$$\boldsymbol{y}_{s_k} = \boldsymbol{p}_{c_k}^T \cdot \boldsymbol{P}_{s_k}(\boldsymbol{F}_{s_k}) = \boldsymbol{p}_{c_k}^T \cdot \begin{bmatrix} P_{s_{k1}}(\boldsymbol{F}_{s_k}) \\ \cdots \\ P_{s_{kM}}(\boldsymbol{F}_{s_k}) \end{bmatrix},$$

cross-entropy loss of $\boldsymbol{P}_{s_k}$ is:

$$L_P = -\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \boldsymbol{y}_{s_k}^i \log(\boldsymbol{p}_{c_k}^T \cdot \boldsymbol{P}_{s_k}(\boldsymbol{F}_{s_k})).$$

As in equation (2), the cross-domain constraint for learning source classifiers in the proposed fuzzy model is re-written as:

$$\mathcal{L}_{crof} = \frac{2}{K(K-1)} \sum_{k_1=1}^{K-1} \sum_{k_2=k_1+1}^{K} \Big(\frac{1}{n_t} \sum_{j=1}^{n_t} \big|\boldsymbol{p}_{c_{k_1}}^T \cdot \boldsymbol{P}_{s_{k_1}}(\boldsymbol{F}_{t_{k_1}}^j) - \boldsymbol{p}_{c_{k_2}}^T \cdot \boldsymbol{P}_{s_{k_2}}(\boldsymbol{F}_{t_{k_2}}^j)\big|\Big).$$

The total loss of learning the fuzzy rule-based source classifier is:

$$\boldsymbol{P}_{s_k} = \underset{\substack{\boldsymbol{P}_{s_k} \in \mathcal{H} \\ (\boldsymbol{x}_{s_k}, \boldsymbol{y}_{s_k}) \sim \mathcal{D}_{s_k}}}{\arg\min} \sum_{m=1}^{M} L_m + \gamma_1 L_P + \gamma_2 L_{crof}.$$

C. Target Task Completion

To complete the target task, all source classifiers are combined to predict the target labels, which can be expressed as a fuzzy model:

if $\boldsymbol{F}_t$ *is* $\mathcal{D}_{s_k}$, then $\boldsymbol{y}_t$ *is* $\boldsymbol{p}_{c_k}^T \cdot \boldsymbol{P}_{s_k}(\boldsymbol{F}_t), k = 1, 2, \cdots, K.$

The final prediction of the target data is:

$$\boldsymbol{y}_t = \boldsymbol{p}_d^T \cdot \begin{bmatrix} \boldsymbol{p}_{c_1}^T \cdot \boldsymbol{P}_{s_1}(\boldsymbol{F}_t) \\ \cdots \\ \boldsymbol{p}_{c_K}^T \cdot \boldsymbol{P}_{s_K}(\boldsymbol{F}_t) \end{bmatrix},$$

$\boldsymbol{p}_d$ is the membership vector, indicating the probability of the target samples belonging to a source domain.

To define the membership, pseudo label-based and feature-based strategies are used to determine the combination rule. First, source classifiers directly pseudo label the target data, noting the number of target samples which obtain the same results from multiple source classifiers in each batch as n_c, batch size as n_b, the frequency of $n_c = n_b$ is a_c, and a threshold a is defined to identify if there is a significant difference among the predictions. If $a_c > a$, which means multiple source domains contribute similarly to the target domain, the averaged combination is then used, the element value of $\boldsymbol{p}_d$ is $\frac{1}{K}$, if $a_c \leq a$, a domain discriminator is used to estimate the element values.

We collected the shared features $\{\boldsymbol{f}_{s_k}\}_{k=1}^K$ and $\boldsymbol{f}_t$, the domain discriminator P_d is controlled by:

$$L_{Pd} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \boldsymbol{y}_d^i \log(P_d(\boldsymbol{f}_s)).$$

$\boldsymbol{y}_d$ is domain label, $\boldsymbol{f}_s = \bigcup_{k=1}^{K} \{\boldsymbol{f}_{s_k}\}$, $n_s = \sum_{k=1}^{K} n_{s_k}$. The membership vector is:

$$\boldsymbol{p}_d = P_d(\boldsymbol{f}_t), \boldsymbol{p}_d = [p_{d_1}, \cdots, p_{d_K}]^T.$$

The combination rule of target classifier can be formulated as:

$$\boldsymbol{y}_t = \begin{cases} \frac{1}{K} \sum_{k=1}^{K} (\boldsymbol{p}_{c_k}^T \cdot \boldsymbol{P}_{s_k}(\boldsymbol{F}_t)), \text{if } a_c > a, \\ \sum_{k=1}^{K} p_{d_k} \cdot (\boldsymbol{p}_{c_k}^T \cdot \boldsymbol{P}_{s_k}(\boldsymbol{F}_t)), \text{if } a_c \leq a, \end{cases}$$
$$k = 1, 2, \cdots, K.$$

III. Experiments

A. Datasets and Parameter Setting

The proposed method is evaluated on real-world visual datasets ImageCLEF-DA and Office-31.

ImageCLEF-DA is a balanced dataset containing 1800 images collected from 12 categories, where every category contains 50 images. It has three image libraries: Caltech-256 (C), ImageNet ILSVRC 2012 (I) and Pascal VOC 2012 (P), and each library is regarded as a domain. We test the proposed model by building three tasks: $I, C \rightarrow P$; $I, P \rightarrow C$; $C, P \rightarrow I$.

Office-31 is an unbalanced dataset containing 4110 images collected from 31 categories, and the number of images in every category is different. It has three libraries: Amazon

(A), Webcam (W) and DSLR (D). Amazon has 2817 images, Webcam has 795 images, and DSLR has 498 images taken by different devices. Regarding each library as a domain, we test the proposed model by building three tasks: $A, W \rightarrow D$; $A, D \rightarrow W$; $D, W \rightarrow A$.

$ResNet50$ is employed as the backbone ϕ, multi-view feature extraction networks ϕ_{c_k}, ϕ_{d_k} contain three convolutional layers and reduce the dimension of the shared features from 2048 to 256, batch size $n_b = 32$; learning rate η is $\eta = \frac{\eta_0}{(1+10\epsilon)^{0.75}}$, $\eta_0 = 0.01$, ϵ is the training progress changing linearly from 0 to 1, the momentum is 0.9 and weight decay is $5e-4$. The trade-off parameter $\alpha = 0.01$, λ_*, γ_* follow existing work [22], that is $\frac{2}{1+exp(-10\epsilon)} - 1$, threshold $a = 0.5$. To reduce the experimental complexity, a_c of each task is calculated using the pre-trained source classifiers rather than fuzzy source classifiers, and directly applies to target classification. a_c of tasks $I, C \rightarrow P$, $I, P \rightarrow C$, $C, P \rightarrow I$ from ImageCLEF-DA and $A, D \rightarrow W$ from Office-31 is larger than 0.5, while that of tasks $A, W \rightarrow D$ and $D, W \rightarrow A$ is smaller than 0.5.

The comparison methods are as follows and single source domain adaptation methods include:

- DAN: Deep adaptation network [23];
- RevGrad: Reverse gradient [24];
- D-CORAL: Correlation alignment for domain adaptation [25];
- MRAN: Multi-representation adaptation network [26];
- MDDA: Manifold dynamic distribution adaptation [27];
- DDAN: Dynamic distribution adaptation network [27].

Multi-source domain adaptation methods include:

- DCTN: Deep cocktail network [28];
- MFSAN: Multiple feature spaces adaptation network [22];
- DFRE: Distribution fusion and relationship extraction network [29].

B. Comparison and Analysis

All experiments are repeated for three times and the results are averaged accuracy. Tables I and II show the results on ImageCLEF-DA and Office-31 respectively. Here "Single Best" means the result is the best performance of a single source domain; "Source Combine" means combining all the source domains as one; "Multi-Source" means the results of the multiple source domains.

It can be seen the proposed method achieves the highest performance on most tasks. Generally, multi-source domain adaptation outperforms single source domain adaptation. Knowledge transfer with considering domain shift is superior to which simply mixes all source training samples. Sometimes, single source domain adaptation performs best, for example, tasks $I, C \rightarrow P$ using MDDA and $A, W \rightarrow D$ using DDAN, which means when combining all source classifiers or mixing source samples following different distributions, negative transfer may occurr. We will investigate this as future work to avoid negative transfer when combining source domains.

TABLE I
COMPARISON OF CLASSIFICATION ACCURACY (%) ON IMAGECLEF-DA.

Standards	Method	I, C-P	I, P-C	P, C-I	Avg
Single best	ResNet	74.8	91.5	83.9	83.4
	DAN	75.0	93.3	86.2	84.8
	D-CORAL	76.9	93.6	88.5	86.3
	RevGard	75.0	96.2	87.0	86.1
	MRAN	78.8	95.0	93.5	89.1
	MDDA	**79.8**	95.7	92.0	89.2
	DDAN	78.0	94.0	91.0	87.7
Source Combine	DAN	77.6	93.3	92.2	87.7
	D-CORAL	77.1	93.6	91.7	87.5
	RevGard	77.9	93.7	91.8	87.8
Multi-Source	DCTN	75.0	95.7	90.3	87.0
	MFSAN	79.1	95.4	93.6	89.4
	DFRE	79.5	95.8	93.7	89.7
	MDAFuz	79.4	**96.3**	**94.5**	**90.1**

TABLE II
COMPARISON OF CLASSIFICATION ACCURACY (%) ON OFFICE-31.

Standards	Method	A, W-D	A, D-W	W, D-A	Avg
Single best	ResNet	99.3	96.7	62.5	86.2
	DAN	99.5	96.8	66.7	87.7
	D-CORAL	99.7	98.0	65.3	87.7
	RevGard	99.1	96.9	68.2	88.1
	MRAN	99.8	96.9	70.9	89.2
	MDDA	99.2	97.1	73.2	89.8
	DDAN	**100.0**	96.7	65.3	87.3
Source Combine	DAN	99.6	97.8	67.6	88.3
	D-CORAL	99.3	98.0	67.1	88.1
	RevGard	99.7	98.1	67.6	88.5
Multi-Source	DCTN	99.3	98.2	64.2	87.2
	MFSAN	99.5	98.5	72.7	90.2
	DFRE	99.6	98.7	73.1	90.5
	MDAFuz	99.7	**99.0**	**74.0**	**90.9**

Tables III and IV show the performance without and with a fuzzy system, "S" means single source domain, "M" means multi-source domain. Source order is the same as described, for example, $S1$ is A in task $A, W \rightarrow D$. It indicates that for many tasks, both single source and multi-source domain adaptation, the performance with fuzzy rules is better than that without fuzzy rules. For some tasks like $A, W \rightarrow D$, the accuracy without fuzzy rules is higher. The reason for this is that source domains show different levels of correlation with the target domain, and for some weakly connected source samples, transferrable information from each cluster is not enough for learning the target task, in other words, the auxiliary among training samples may be lost. We will try to solve this in the future.

IV. CONCLUSION AND FUTURE STUDY

In this paper, we propose a fuzzy system based method for multi-source domain adaptation. A pre-trained model is employed to extract the adapted features of the source and target domains, and the pre-trained source classifiers are used to develop a similarity estimation strategy, which divides training samples from each source domain into multiple clusters to construct fuzzy rules. Using the extracted features and fuzzy rules, new source classifiers are learned. To predict target

TABLE III
COMPARISON OF CLASSIFICATION ACCURACY (%) ON IMAGECLEF-DA WITHOUT AND WITH FUZZY RULES.

Standards		I, C-P	I, P-C	P, C-I	Avg
Without fuzzy	S1	78.8	95.4	93.2	89.2
	S2	79.0	95.2	93.3	
	M	79.1	95.7	93.4	89.4
With fuzzy	S1	78.9	**96.5**	94.3	89.8
	S2	78.7	95.7	**94.8**	
	M	**79.4**	96.3	94.5	**90.1**

TABLE IV
COMPARISON OF CLASSIFICATION ACCURACY (%) ON OFFICE-31 WITHOUT AND WITH FUZZY RULES.

Standards		A, W-D	A, D-W	W, D-A	Avg
Without fuzzy	S1	96.3	97.9	73.0	89.8
	S2	**99.8**	98.4	73.6	
	M	98.9	98.6	73.3	90.3
With fuzzy	S1	95.5	98.2	73.0	89.9
	S2	99.7	**99.0**	**74.2**	
	M	99.7	**99.0**	74.0	**90.9**

labels, the combination rule is defined with regard to the pseudo target labels and a domain discriminator, which are employed to measure the similarity among source and target domains. The experiment results and analysis on real-world visual datasets show that the proposed method outperforms the other comparison methods.

In future studies, we expect to avoid the negative transfer when combining source domains. Selecting source samples or domains that show strong correlation with the target domain is worth trying to achieve this. In addition, we also aim at enhancing the auxiliary among multiple clusters to facilitate information transfer among both data and tasks.

REFERENCES

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2009.

[2] S. Zhao, B. Li, X. Yue, P. Xu, and K. Keutzer, "Madan: Multi-source adversarial domain aggregation network for domain adaptation," *arXiv*, pp. arXiv–2003, 2020.

[3] P. Panareda Busto and J. Gall, "Open set domain adaptation," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, October 22 - 29 2017, pp. 754–763.

[4] Z. Fang, J. Lu, F. Liu, J. Xuan, and G. Zhang, "Open set domain adaptation: Theoretical bound and algorithm," *IEEE Trans. Neural Netw. Learn. Syst*, 2020.

[5] H. Liu, Z. Cao, M. Long, J. Wang, and Q. Yang, "Separate to adapt: Open set domain adaptation via progressive separation," in *Proc. IEEE. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, California, USA, June 16 - 20 2019, pp. 2927–2936.

[6] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. Mar, pp. 723–773, 2012.

[7] Q. Dai, X. Shen, X.-M. Wu, and D. Wang, "Network transfer learning via adversarial domain adaptation with graph convolution," *arXiv preprint arXiv:1909.01541*, 2019.

[8] J. Wen, R. Liu, N. Zheng, Q. Zheng, Z. Gong, and J. Yuan, "Exploiting local feature patterns for unsupervised domain adaptation," in *Proc. Am. Assoc. Artif. Intell. (AAAI)*, vol. 33, Honolulu, Hawaii, USA, January 27 - February 1 2019, pp. 5401–5408.

[9] H. Tang, K. Chen, and K. Jia, "Unsupervised domain adaptation via structurally regularized deep clustering," in *Proc. IEEE. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, California, USA, June 16 - 20 2019, pp. 8725–8735.

[10] V. K. Kurmi, S. Kumar, and V. P. Namboodiri, "Attending to discriminative certainty for domain adaptation," in *Proc. IEEE. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, California, USA, June 16 - 20 2019, pp. 491–500.

[11] S. Moon and J. G. Carbonell, "Completely heterogeneous transfer learning with attention-what and what not to transfer." in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, vol. 1, no. 1, Melbourne, Australia, August 19 - 25 2017, pp. 1–7.

[12] H. Li, S. J. Pan, R. Wan, and A. C. Kot, "Heterogeneous transfer learning via deep matrix completion with adversarial kernel embedding," in *Proc. Am. Assoc. Artif. Intell. (AAAI)*, vol. 33, no. 01, Honolulu, Hawaii, USA, January 27 - February 1 2019, pp. 8602–8609.

[13] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: a survey," *Knowl. Based. Syst.*, vol. 80, pp. 14–23, 2015.

[14] H. Zuo, J. Lu, G. Zhang, and W. Pedrycz, "Fuzzy rule-based domain adaptation in homogeneous and heterogeneous spaces," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 2, pp. 348–361, 2018.

[15] J. Shell and S. Coupland, "Fuzzy transfer learning: methodology and application," *Inf. Sci.*, vol. 293, pp. 59–79, 2015.

[16] P. Xu, Z. Deng, J. Wang, Q. Zhang, K.-S. Choi, and S. Wang, "Transfer representation learning with tsk fuzzy system," *IEEE Trans. Fuzzy Syst.*, 2019.

[17] J. Lu, H. Zuo, and G. Zhang, "Fuzzy multiple-source transfer learning," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 12, pp. 3418–3431, 2019.

[18] F. Liu, G. Zhang, and J. Lu, "Multi-source heterogeneous unsupervised domain adaptation via fuzzy-relation neural networks," *IEEE Trans. Fuzzy Syst.*, 2020.

[19] K. Li, J. Lu, H. Zuo, and G. Zhang, "Multi-source contribution learning for domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst*, 2021.

[20] V. Vapnik and V. Vapnik, "Statistical learning theory," *New York*, vol. 1, p. 624, 1998.

[21] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, 2010.

[22] Y. Zhu, F. Zhuang, and D. Wang, "Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources," in *Proc. Am. Assoc. Artif. Intell. (AAAI)*, vol. 33, Honolulu, Hawaii, USA, January 27 - February 1 2019, pp. 5989–5996.

[23] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*. Lille, France: PMLR, July 6 - 11 2015, pp. 97–105.

[24] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn. (ICML)*. JMLR, July 6 - 11 2015, pp. 1180–1189.

[25] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Amsterdam, The Netherlands: Springer, October 8 - 16 2016, pp. 443–450.

[26] Y. Zhu, F. Zhuang, J. Wang, J. Chen, Z. Shi, W. Wu, and Q. He, "Multi-representation adaptation network for cross-domain image classification," *Neural. Netw.*, vol. 119, pp. 214–221, 2019.

[27] J. Wang, Y. Chen, W. Feng, H. Yu, M. Huang, and Q. Yang, "Transfer learning with dynamic distribution adaptation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 1, pp. 1–25, 2020.

[28] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin, "Deep cocktail network: Multi-source unsupervised domain adaptation with category shift," in *Proc. IEEE. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, Utah, USA, June 18 - 22 2018, pp. 3964–3973.

[29] K. Li, J. Lu, H. Zuo, and G. Zhang, "Multi-source domain adaptation with distribution fusion and relationship extraction," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*. Virtual online: IEEE, July 19 - 24 2020, pp. 1–6.