Pattern Anal Applic (2018) 21:813–827
https://doi.org/10.1007/s10044-017-0664-1

SHORT PAPER

A semi-supervised domain adaptation assembling approach for image classification

Shruti Saxena[1] · Shreelekha Pandey[2] · Pritee Khanna[1]

Received: 13 May 2016 / Accepted: 3 October 2017 / Published online: 19 October 2017

Abstract Automatic annotation of images is one of the fundamental problems in computer vision applications. With the increasing amount of freely available images, it is quite possible that the training data used to learn a classifier has different distribution from the data which is used for testing. This results in degradation of the classifier performance and highlights the problem known as domain adaptation. Framework for domain adaptation typically requires a classification model which can utilize several classifiers by combining their results to get the desired accuracy. This work proposes depth-based and iterative depth-based fusion methods which are basically rank-based fusion methods and utilize rank of the predicted labels from different classifiers. Two frameworks are also proposed for domain adaptation. The first framework uses traditional machine learning algorithms, while the other works with metric learning as well as transfer learning algorithm. Motivated from ImageCLEF's 2014 domain adaptation task, these frameworks with the proposed fusion methods are validated and verified by conducting experiments on the images from five domains having varied distributions. Bing, Caltech, ImageNet, and PASCAL are used as source domains and the target domain is SUN. Twelve object categories are chosen from these domains. The experimental results show the performance improvement not only over the baseline system, but also over the winner of the ImageCLEF's 2014 domain adaptation challenge.

Keywords Domain adaptation · Image classification · Fusion methods · Domain adaptation frameworks

1 Introduction

The amount of freely available images is increasing day by day and their annotation is one of the fundamental problems in computer vision applications. Also, online image annotation tools like Amazon's mechanical Turk and Google's GoogleS are limited to a few number of objects categorization. For example, these tools generalize well for object categories like books, tables, but are not able to categorize food pictures as well as images having multiple objects [10]. This happens because of the differences in distribution of a source data used for learning and a different but related target data used for testing. Visual appearance of images and hence their data distribution varies in different domains even if their object labels are same. Due to this reason an image classification model fails when tested for images from other datasets. For example, Fig. 1a shows images of an object category *Cup* collected from Amazon and Webcam. The accuracies obtained with support vector machine (SVM) and naive Bayes nearest neighbor (NBNN) classifiers as given in Fig. 1b show a steep drop in the performance of both the classifiers when test images are drawn from the target domain instead of source domain [54]. The reason behind this performance is the difference in data distribution of object category *Cup* in the given domains, which is clearly visible in images shown in Fig. 1a. A classifier trained with

✉ Pritee Khanna
pkhanna@iiitdmj.ac.in

Shruti Saxena
shruti.saxena@iiitdmj.ac.in

Shreelekha Pandey
shreelekha.pandey@thapar.edu

1 PDPM Indian Institute of Information Technology, Design and Manufacturing Jabalpur, Dumna Airport Road, Mehgeon, Jabalpur, Madhya Pradesh 482005, India

2 Thapar University, Patiala, Punjab 147004, India

(a)

Train	Test	SVM	NBNN
Source	Source	$54\% \pm 2$	$61\% \pm 1$
Source	Target	$20\% \pm 1$	$19\% \pm 1$

(b)

Fig. 1 **a** Some images from Amazon and Webcam for object category *Cup*. **b** Accuracy of classifiers trained with the images from Amazon [54]

one dataset and tested with another dataset tends to produce poor results even if classes of images in both the datasets are same. This is known as domain adaptation or cross-domain learning problem. Domain adaptation is a challenging task and any solution to it would lead to generalization of vision-based annotation systems. Most of the available solutions to the domain adaptation tasks consider situations where source domain consists of one or two datasets; source and target domains have the same labels; and target domain has less number of annotated training data.

Aiming to push the state of art toward more realistic way, ImageCLEF's 2014 domain adaptation task presented the challenge of using more than two source domains [1, 10]. Three groups XRCE, Hubert Curien Lab, and Idiap (organizers) provided their results for this task. The XRCE group utilized the concept of fusion to combine the results of different classifiers and attained the highest number of correct image annotations. However, accuracy reported by XRCE group was less than 50% which motivated us to work on this problem [14]. Addressing the problem of domain adaptation, this work presents two classification models that utilizes the knowledge from multiple source domains in the learning phase. Testing is performed on testing phase only. Considering the importance of fusion methods, two new fusion methods are proposed in this work and their applicability in the field of domain adaptation is evaluated on both the proposed frameworks. One of the proposed framework works with traditional machine learning algorithms, while the other is able to work with either metric learning or transfer learning algorithm.

The rest of the paper is organized as follows. Section 2 presents the prior art of the domain adaptation work. Section 3 presents the proposed fusion methods. Overview of the proposed classification frameworks is given in Sect. 4. Along with the experimental setup, results are analyzed in Sect. 5. Finally, the work is concluded in Sect. 6.

2 Literature review

Domain adaptation plays an important role in many applications like face recognition [53], opinion mining [47], sentiment analysis [39], and information extraction [18]. Statistical machine learning algorithms build a classification model from training data to achieve accurate classification on the test data. The success of statistical machine learning algorithms is based on an assumption that testing data follow the same training data distribution. Such classification models fail if test data are drawn from a dataset with different distribution. To resolve the domain adaptation problem researchers have proposed variants of traditional machine learning algorithms like transductive support vector machine (TSVM), adaptive support vector machines (aSVM), localized SVM (LSVM) [38], semi-supervised support vector machines (S3VM) [5], and C-SVM (coupled SVM algorithm) [60]. Other solutions include an optimized inference algorithm using the technique of conditional expectation maximization and a simple mixture model [17], a co-training algorithm using model adaptation techniques [13], and many more.

Figure 2 shows a generalized framework of domain adaptation-based image annotation system. Features of images from all source domains are extracted and used to train the classifier. During testing, the same features for the query image are extracted and trained classifier predicts the most appropriate annotation for this query image. The target domain labels may or may not be used during training phase. In unsupervised domain adaptation, labeled source data along with unlabeled target data are employed in the training phase [29, 30]. In supervised domain adaptation, labeled source data along with some labeled target data are employed in the training phase

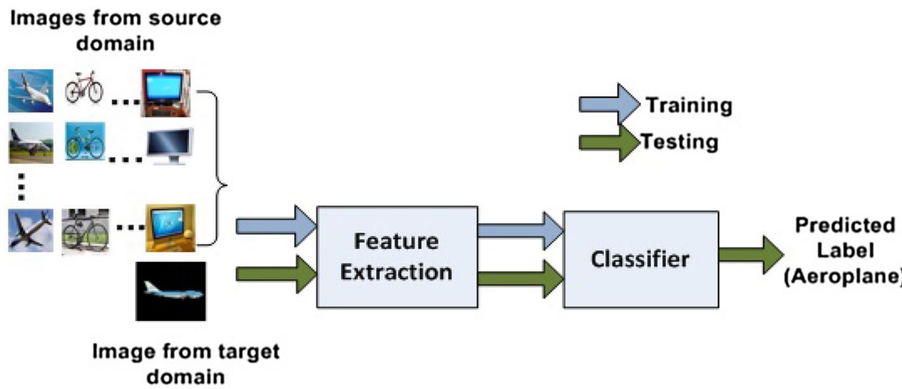

Fig. 2 Stages of domain adaptation method

[33]. But semi-supervised approaches use labeled source domain and also some labeled as well as unlabeled target domain data [16].

Summary of the domain adaptation algorithms existing in the literature is given in Table 1. A brief review of methods that try to learn the boundary of classifier for each domain using metric and transfer leaning techniques is presented here.

2.1 Metric learning

Metric learning is widely used for verification, classification, and ranking problems [3, 32, 49]. An extended version of k-NN classifier, i.e., large margin nearest neighbor (LMNN) is developed to learn metrics for image classification [63]. Its main goal is to separate the examples from different classes by a large margin and at the same time reduce this margin for the examples from the same class. Cost function is defined over triplets of points that are optimized using stochastic gradient descent (SGD) training procedure. LMNN is best suited for large datasets. Its another variant, large margin nearest local mean (LMNLM) assigns a class label based on the minimum distance between the test image and the means of all nearest neighborhood classes [12]. It uses semi-definite programming (SDP) and outperforms LMNN. However, it is time-consuming as it computes distance between all training examples, and therefore does not scale well for large datasets. Another method with the same drawback is TagProp, which assigns weights to training instances based on their distance with the test image [46]. During label prediction, total weight from neighborhood for each class is calculated and label of a class with maximum weight is the predicted label.

Metric learning also works by collapsing classes where objective is to learn a distance that assigns uniform weights to images from the same class and near to zero weights to images from other classes [28]. In comparison, neighborhood component analysis assigns only zero weight to other class images [35]. In another form of metric learning, e.g., NBNN classifier learns image-to-class distance metric. It uses a gradient descent method to solve the optimization problem of learning per-class mahalanobis distance metrics [59].

The next metric learning approach is nearest mean classifier (NCM), where query image is compared with the mean value of the classes from source domains, and the label of a class with the minimum distance is assigned to it. Instead of image descriptor, this method uses class descriptor and therefore, requires less training time as compared to other methods, e.g., k-NN where query image is compared to each image descriptor to get the class label. NCM also takes less time for inference, because image labels are selected based on the distance between the image and class descriptors only. In comparison with some well-known classifiers like SVM and their variants used in domain adaptation, NCM algorithm is a linear classifier. NCM is explored in this work, as it outperforms k-NN and is comparable to SVM [55].

2.2 Transfer learning

Transfer learning means applying previously learned skill in one condition into another condition. Domain adaptation is a part of the transfer learning which aims to transfer the shared knowledge across various domains [50]. TrAdaBoost [15] and Multisource TrAdaBoost [34] are instance transfer approaches where examples from sources are directly used for training. Another technique, feature representation transfer tries to find a representation of the feature space that minimizes the difference between source and target. Task TrAdaBoost is an example of parameter-transfer approach where target shares parameters with related source [34, 50]. Relational-knowledge transfer presumes that source data are correlated and try to transfer this correlation to target. Transfer boosting algorithms and TrAdaBoost [15] take advantage of both AdaBoost and the weighted majority algorithm. Attribute-based classification solves the image classification problem by transferring high-level semantic across the classes. Another

Table 1 Summary of domain adaptation algorithms

Author, year	Classifier	Domains or datasets	Features	Evaluation measures
Dai et al. [15]	TrAdaBoost	Three text datasets (20 Newsgroups, SRAA, Reuters 21578) and Mushroom (UCI machine learning repository)	-	Error rate[1]
Boiman et al. [7]	NBNN	Caltech-101, Caltech-256, Graz-01	SIFT	Accuracy[2]
Jiang [36]	Review paper	-	-	-
Duan et al. [21]	Multiple source domain adaptation method	Three English (CNN, MSNBC and NBC), two Chinese (CCTV and NTDTV) and one Arabic channel (LBC)	Grid color moment, Gabor texture and edge detection histogram	Mean average precision (MAP)
Pan et al. [50]	Review paper	-	-	-
Everingham et al. [63]	Multisource and Task TrAdaBoost	Caltech and a dataset from Google image search	Bag of words	Accuracy
Kulis et al. [40]	Asymmetric regularized cross-domain transformation	Amazon, Dslr, and Webcam	SURF and SIFT features	Accuracy
Geng et al. [27]	Metric learning	FERET, ORL, UMIST, and YALE	Bag of words	Recognition rate
Duan et al. [22]	Heterogeneous features augmentation	Amazon, Webcam, and Dslr	Visual words	Accuracy
Mensink et al. [55]	Nearest class mean classication	ImageNet	Fisher vector computed over SIFT descriptors and local color features	Average of top-1 and top-5 flat error
Donahue et al. [20]	Extended max-margin domain transforms	Webcam and Dslr	SURF bag of words (BoW)	Accuracy
Rodner et al. [25]	Large-scale max-margin domain transform	ImageNet and SUN	Bag of words	Accuracy
Tommasi et al. [56]	NBNN	Amazon, Webcam, and Dslr	Visual words	Accuracy
Hoffman et al. [37]	Large-scale detection through adaptation	ImageNet	Convolutional neural networks	MAP
Long et al. [42]	Adaptation regularization-based transfer learning	Three handwritten digits (USPS, MNIST and PIE) and two text datasets (20-Newsgroups and Reuters-21578)	-	Accuracy
Xu et al. [61]	Adaptive structural SVM	PASCAL and INRIA	-	Average miss rate
Chidlovskii et al. [14]	Fusion of different classifiers	Bing, Caltech, ImageNet, Pascal, and SUN	SIFT	Accuracy
Yao et al. [62]	Semi-supervised DA with subspace learning	Amazon, Webcam, and Dslr	Deep convolutional activation	Accuracy
Patel et al. [51]	Review paper	-	-	-

[1]**Error rate** defines the classification error rate with learner trained on source domains and tested on target domain

[2]**Accuracy** is the proportion of the total number of target domain images that are correctly annotated

variation of domain adaptation based on transfer learning methods is transfer component analysis (TCA) that learns the components from domains and tries to minimize the domain distance [45].

2.3 Fusion methods

Fusion of information is widely used in applications like biometrics, robot navigation, land mine detection, weather forecasting, and many more. Information is fused at different levels, hence fusion algorithms are categorized as measurement level, abstract level, and rank level. In domain adaptation, outputs of different classifiers can be fused to get more appropriate annotation for images from the target domain. Each classifier predicts the order of preference for all class labels. Different methods are applied in rank level fusion to combine the obtained order of preference and get the final label prediction. Dictator [2], HAT, FPP [58], Runoff [58], Exhaustive [23], Pref [9], Contingent, Coombs [48], Highest rank, Borda count [8], Nanson [11], Baldwin [4], Bucklin majority voting [24], and Condorcet [19] are some popular rank level fusion methods.

This work aims to develop better domain adaptation classification models built on traditional machine learning algorithms as well as domain adaptation algorithms. Two frameworks designed by using traditional and metric/transfer learning approaches are combined with the fusion methods to increase the rate of image classification system. Instead of exploring all the approaches, the better ones are identified in each of the two learning approaches. Among metric learning approaches, the most accurate NCM and from transfer learning, the most popular Transfer AdaBoost (TrAdaBoost) is selected. This work also presents two fusion methods that combine the class label outputs from the classifiers and assign the appropriate class label to the query image. The proposed fusion methods are designed using the concept of depth as explained in the next section.

3 The proposed fusion methods

Several rank-based fusion methods like highest rank, majority voting, nonlinear weighted ranks, Bucklin majority voting, and Borda count exist in literature [41]. These methods work quite well when only topmost prediction of classifier results are being fused, but fail when applied to all predicted values of the classifier. Instead of using all predicted labels from classifiers, the proposed fusion methods focus on fusion of a few top predicted values till a particular depth. This section explains the concept of *depth* used in this work and the proposed fusion algorithms.

3.1 The concept of 'depth'

A two-step process is proposed to identify the *depth* up to which fusion is to be used for an improved performance. For each image in the target domain, the correct classification depth is computed for all the classifiers. Depth at which the maximum number of images are correctly classified is then identified for each classifier. Let CR_j^i is the number of images correctly classified (i.e., classification rate) at ith depth by the jth classifier, then

$$i_{\max j} = \operatorname*{argmax}_i (CR_j^i) \tag{1}$$

where $i_{\max j}$ is the depth at which jth classifier has maximum correct classification.

From the depth values obtained for all the classifiers, the one with maximum correct classification is selected as the required *depth* parameter, given as

$$\text{depth} = \operatorname*{argmax}_{i_{\max j}} (CR_j^{i_{\max j}}) \tag{2}$$

Example The data given for this example is obtained from experimental evaluation only, but to make things more conceptual here the names of classifiers, class labels, and the actual query image are not mentioned. Let us consider four classifiers (CL_j, $j = 1..4$) and a target image corpus of 600 images belonging to 12 classes, namely *Aeroplane*, *Bike*, *Bird*, *Boat*, *Bottle*, *Bus*, *Car*, *Dog*, *Horse*, *Monitor*, *Motorbike*, and *People*. For each image, the depth of its correct label is found. Table 2 explains this concept of depth. 'Depth 1' means consider only those class labels which are predicted at the rank 1 by the four classifiers. This will take only a single-class label, which has the maximum probability or score value from each of the four lists. Similarly, 'depth 2' means consider the top two class label predictions by each of the four classifiers. So, 'depth n' means consider the top-n class labels from each of the four lists predicted by the four classifiers.

Finding the depth which reports the maximum correct classifications For each image in the source domain, find the depth at which its correct class label is predicted by each of the four classifiers. This depth is known as

Table 2 Relation between class labels and depth

	Classifier 1	Classifier 2	Classifier 3	Classifier 4
Depth 1	Bike	Bird	Bike	Aeroplane
Depth 2	Bike, Bird	Bird, Bike	Bike, Aeroplane	Aerplane, Car
Depth 3	Bike, Bird, Car	Bird, Bike, Aeroplane	Bike, Aeroplane, Book	Aerplane, Car, Book
.	...	...	...	...
.	...	...	...	...
Depth n	...	...	...	...

Table 3 Correct class label depth (CCLD) for different classifiers

CL_1		CL_2		CL_3		CL_4	
Image number	CCLD	Image number	CCLD	Image number	CCLD	Image number	CCLD
1	5	1	3	1	2	1	2
2	4	2	2	2	5	2	9
...	....	...	....	...	....	...	....
600	7	600	5	600	4	600	6

Table 4 Total number of correct classifications at each depth for different classifier

CL_1		CL_2		CL_3		CL_4	
Depth	CCLD	Depth	CCLD	Depth	CCLD	Depth	CCLD
1	52	1	44	1	23	1	52
2	31	2	48	2	51	2	39
...	....	...	....	...	....	...	....
6	**71**	...	....	**5**	**53**	**8**	**67**
...	....	**11**	**59**	...	....	...	....
12	26	12	32	12	39	12	11

Bold values indicate the best correct classification

Table 5 Depth at which maximum correct classification is obtained for different classifiers (using Eq. 1)

Classifiers	CL_1	CL_2	CL_3	CL_4
Maximum Correct classification	71	59	53	67
Depth	6	11	5	8

the Correct Class Label Depth (*CCLD*) and is shown in Table 3. The next step is to find the correct classification obtained at each depth. Table 4 shows the total number of correct classification obtained at each depth. The depth at which the maximum number of correct classifications is obtained for each classifier is shown in Table 5. The first row shows the frequency and the second row shows the corresponding depth, i.e., the depth at which the classifier has reported the maximum number of correct classifications. Classifier CL_1 is providing the maximum correct classification at *depth* 6 among all the classifiers. Therefore, the value of parameter *depth* is selected as 6 for the given example.

3.2 Proposed fusion method 1: depth-based fusion

The proposed depth-based fusion method uses the concept of 'depth' as: "Find the majority votes for a query image till computed *depth*. If a unique class label with maximum frequency is obtained, assign it to the query image. Otherwise, in case of clashing use the results of the best classifier. The best classifier is the one that obtains majority votes on the entire target data at the first prediction given by classifiers, i.e., at *depth 1.*"

The complete method given as Algorithm 1 is explained with an illustrative example.

Algorithm 1 Proposed depth based fusion to combine the results of classifiers

```
1: procedure DEPTH–BASED–FUSION
2:     Get the predicted labels from all classifiers for each
           image in the target domain
3:     Determine depth using Equation (1) and Equation (2)
4:     For each image, find the label frequency till depth
5:     if class label with maximum frequency is unique then
6:         Assign it to the image
7:     else
8:         Resolve clashing using results of the best classifier
9:     end if
10: end procedure
```

Example Let us continue with the same example. For a query image I_q, each of the 4 classifiers rank all the 12 labels according to their probability or score values as shown in Table 6. The label frequency is computed till the computed *depth* 6 and is shown in Table 7. The next step is to identify a label having maximum frequency and assign it to the query image. Here, label 1, 2, 4, 7, and 8 are clashing as they have the same frequency 3 as shown in bold in Table 7. Therefore, clashing would be resolved by ranking the labels according to the best classifier found through the majority votes method on the entire target data at *depth* 1. Table 8 shows that classifier CL_4 wins the election in majority votes method. Therefore, CL_4

Table 6 Predicted order of labels for an image I_q from different classifiers

CL_1	1	6	11	10	8	4	9	2	3	7	5	12
CL_2	2	1	3	7	4	8	9	5	12	11	10	6
CL_3	2	5	1	10	7	12	9	4	11	8	6	3
CL_4	2	8	4	7	3	6	5	9	11	10	12	1

Table 7 Frequency of each label computed till *depth* 6

Label	1	2	3	4	5	6	7	8	9	10	11	12
Freq	**3**	**3**	2	**3**	1	2	**3**	**3**	0	2	1	1

Table 8 Selection of the best classifier using majority votes method at *depth* 1

Classifiers	CL_1	CL_2	CL_3	CL_4
Majority votes method	41	10	32	47

is selected as the best classifier. Now the labels 1, 2, 4, 7, and 8 (which are clashing at frequency 3) are arranged from the top as 2, 8, 4, 7, and 1 for classifier CL_4 as given in Table 6. The final predicted label for image I_q is 2.

3.3 Proposed fusion method 2: iterative depth-based fusion

The next proposed fusion method is iterative depth-based fusion which differs from depth-based fusion in step 2 as: " After obtaining the *depth* parameter using Eqs. (1) and (2), find votes for all labels till the computed *depth* and apply majority voting method in iterative way. If any class label with majority votes is present then this class label is assigned to the query image. Otherwise, *depth* is decreased by one and again majority votes are computed against all class labels. This process continues till a unique class label with majority votes is found. When *depth* becomes one and still a unique class label with majority votes is not available, then the final decision comes from the best classifier obtained through majority vote method on the entire target data at *depth 1.*"

The flowchart for iterative depth-based fusion method to combine the results of different classifiers is shown in Fig. 3. The complete method given as Algorithm 2 is explained with an illustrative example.

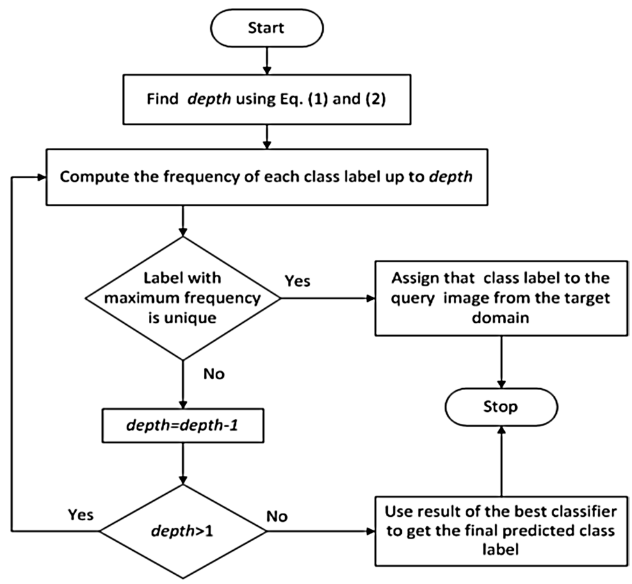

Fig. 3 The flowchart for the proposed iterative depth-based fusion method

Algorithm 2 Proposed Iterative depth based fusion to combine the results of classifiers

```
procedure ITERATIVE–DEPTH–BASED–FUSION
    Get the predicted labels from all classifiers for each
            image in the target domain
    Determine depth using Equation (1) and Equation (2)
    for D = 1 to depth do
        if D > 1 then
            For each image, find the label frequency till
                    depth
            if class label with maximum frequency is
                    unique then
                Assign it to the image
            else
                D = D − 1
            end if
        else
            Resolve clashing using results of the best
                    classifier
        end if
    end for
end procedure
```

Example (contd..) Let us consider the same example again. The *depth* = 6 is obtained as discussed earlier. As a result, the frequencies of all the class labels at *depth* values 6, 5, 4, 3, 2 are given in Table 9. Following the majority voting method for frequency values at *depth* = 6, the class labels 1, 2, 4, 7, and 8 are selected as they have the maximum votes of 3. As a unique label with maximum frequency could not be obtained, the *depth* is decremented by 1. At *depth* = 5 as well a unique class label with majority votes is not present. Observe the frequency values at *depth* = 5 in Table 9, class labels 1, 2, and 7 have equal votes, i.e., 3. Again the *depth* is decremented, and at *depth* = 4 there are two class labels, viz., 1 and 2, with the same number of votes. The scenario remains the same at *depth* = 3 as the class labels 1 and 2 are still clashing at the majority vote value of 3. Another decrement makes *depth* = 2 and as can be observed from the frequency values of the last row, there is a unique class label 2 which has the maximum number of votes. Finally, the image I_q is assigned a class label 2 with this method.

4 The proposed frameworks for domain adaptation

Two frameworks proposed for domain adaptation problem are discussed here. The first framework works with traditional machine learning algorithms. As traditional machine learning algorithms have limitations, therefore, the second framework is proposed for domain adaptation algorithms. The usefulness of the proposed fusion methods is evaluated on both of these frameworks.

4.1 Fusion on traditional machine learning algorithms

Figure 4 shows architecture of the proposed system which combines outputs of traditional machine learning algorithms. In the training phase features of images from all the domains are used to train each selected classifier and testing is accomplished with images from the target dataset.

A wide range of traditional machine learning algorithms exist in state of the art [6]. A few popular ones are Linear Regression, Logistic Regression, Naive Bayes, Support Vector Machine, Decision Tree, Random Forest, and K-Nearest Neighbor algorithms. The applicability of these algorithms vary with the datasets, so it's important to identify an algorithm or a combination of algorithms that fits the best in the considered problem domain. This analysis can easily be accomplished using a tool, WEKA (Waikato Environment for Knowledge Analysis) [44]. WEKA not only provides a collection of different machine learning algorithms, but also incorporates several ways with the help of which a quick analysis of the performance of these algorithms on different datasets can be done. Based on their performance on the datasets, some algorithms are selected. Every classifier predicts the order of labels. These predictions are combined using the proposed rank-based fusion methods to further improve the accuracy and results are discussed in the next section.

Table 9 Frequency of all the labels at different *depth*

Label	1	2	3	4	5	6	7	8	9	10	11	12
Freq @ *depth* = 6	3	3	2	3	1	2	3	3	0	2	1	1
Freq @ *depth* = 5	3	3	2	2	1	1	3	2	0	2	1	0
Freq @ *depth* = 4	3	3	1	1	1	1	2	1	0	2	1	0
Freq @ *depth* = 3	3	3	1	1	1	1	0	1	1	0	1	0
Freq @ *depth* = 2	2	3	0	0	1	1	0	1	1	0	0	0

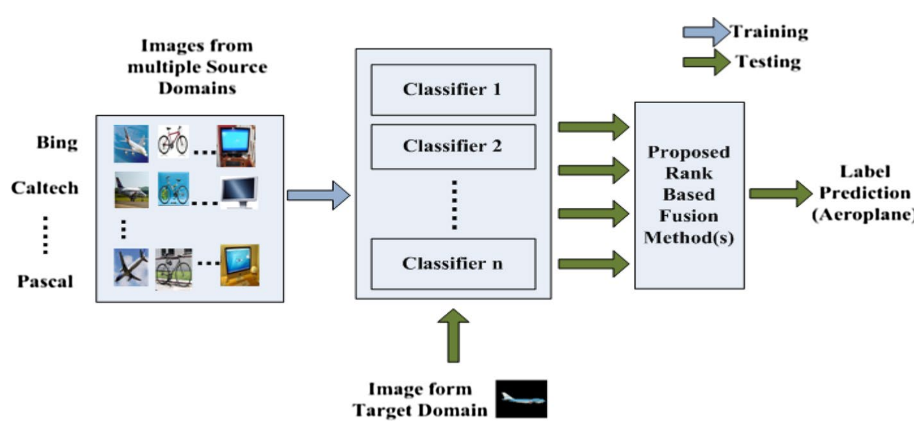

Fig. 4 The proposed framework for traditional machine learning algorithms

4.2 Fusion on domain adaptation algorithms

Figure 5 shows architecture of the system proposed for domain adaptation algorithms which is slightly different from the previous one. Instead of combined images from all the datasets, the classifier is individually trained with each domain and tested on target domain. Thus, problem statement here is 'Given either the same or differently distributed training data, the goal is to build a classifier that will minimize the prediction error or maximize predictions for the target data.'

Suppose d_s is the number of source datasets and the target dataset has N classes. At the fusion time, fusion algorithm gets the matrix of $d_s \times N$. This $d_s \times N$ dimension matrix represents the order of class labels from different classifiers. The proposed fusion methods are used over this matrix to find the final predicted label of the image. The framework is executed separately for two popular domain adaptation algorithms. Nearest class mean (NCM) for metric learning and TrAdaBoost for transfer learning are chosen due to their relative advantages over other methods in these categories as discussed in Sect. 2.

4.2.1 Fusion with metric learning

The NCM classifier [55] assigns query image I_q to the class c^* from source domains having the closest mean.

$$c^* = \underset{c\epsilon\{1,2,3,\ldots,N\}}{\operatorname{argmin}} \; d(I_q, \mu_c) \tag{3}$$

$$\mu_c = \frac{1}{N_c} \sum_{i:y_i \epsilon c} I_i \tag{4}$$

where $d(I_q, \mu_c)$ is Euclidean distance between the image I_q and mean (μ_c) of all the classes c, y_i is ground truth value of image I_i, and N_c is the number of training images belong to class c.

In this method, image is represented by the feature vector and each class is represented by a mean feature vector. Mean feature vector is the mean of feature vectors of all the images that correspond to a class. When a new image has to be annotated, its feature vector is compared with mean feature vectors of all other classes based on Euclidean distance. The basic idea is to learn the projection matrix W such that the distance between the images (feature vector) of the same class decreases and the distance between the images of other classes increases. NCM classifier is formulated as multi-class probabilistic model where the probability of an image I_i belonging to class c can be defined as

$$p(y_i = c \mid I_i) = \frac{\exp(-0.5 d_w(I_i, \mu_c))}{\sum_{c\epsilon N} \exp(-0.5 d_w(I_i, \mu_c))} \tag{5}$$

Optimization of any learning algorithm has objective to minimize the loss function. With respect to this, Eq. (6) is used to show the loss function of the learning algorithm and loss function is presented in term of the log-likelihood of the probability function. Equation (7) is minimum value of the loss function. To learn the projection matrix W, minimization of negative log-likelihood of the correct prediction of the image is required.

$$L = -\frac{1}{N} \sum_{i=1}^{N} \ln p(y_i \mid I_i) \tag{6}$$

For optimization purpose, stochastic gradient descend (SGD) algorithm is used. At each step, W is updated with some fixed learning rate for minimization of negative log-likelihood of correct prediction of images. Thus, gradient of this objective function can be formulated as

$$\nabla_w L = \frac{2}{N} \sum_{i=1}^{N} \sum_{c=1}^{c} ([\![y_i = c]\!] - p(c \mid I_i)) W(\mu_c - I_i)(\mu_c - I_i)^T \tag{7}$$

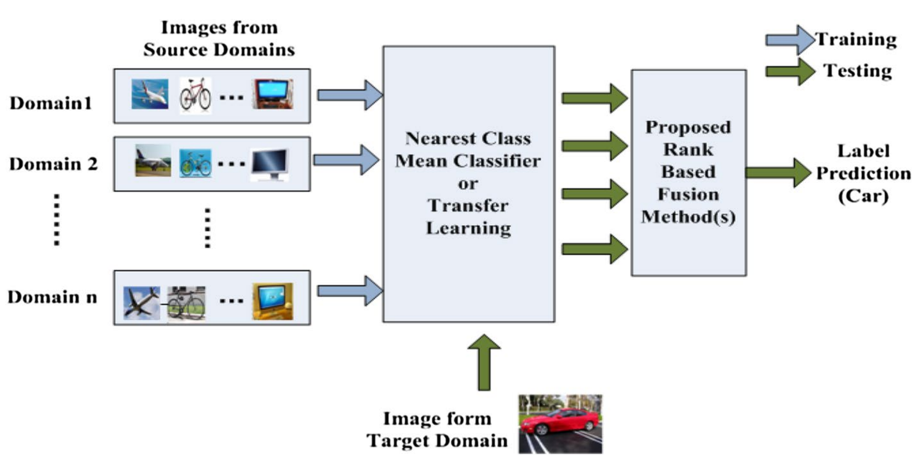

Fig. 5 The proposed framework for domain adaptation algorithms

Being a linear classifier, nearest class mean (NCM) has an advantage over nonlinear k-NN classifier. The cost of computing the mean is negligible as compared to the cost of feature extraction and this operation does not require accessing images of other classes also. However, the performance of NCM degrades over large dataset because only the mean feature vector is used to represent the distribution of training data of a class.

4.2.2 Fusion with transfer learning

TrAdaBoost is an instance-based Transfer AdaBoost learning, which extends AdaBoost for transfer learning. AdaBoost [26] is a meta-machine learning algorithm aims to increase the accuracy of a weak learner by adjusting the weights of training instances. It combines both source and target domain for learning but somewhat treats them differently. In each iteration, it trains the weak classifier on the combination of the source and target instances, and then target instances are used to find error. This error is used for up-weighting and down-weighting the weights of instances. TrAdaBoost algorithm assumes that the same categories and the same set of features are used in both training and test datasets. Because of the difference in the distribution of both source and target domains, some image instances will play an important role in the learning and some will not. Thus, it uses the basic idea of re-weighting the examples using AdaBoost. Similar to metric learning, each domain is trained one by one corresponding to one target domain. At the time of testing, test instance from target domain are passed through all the classifiers which are trained using different domains. A predicted label for the image from target domains is determined using the proposed fusion methods [15].

5 Results and discussion

As stated earlier, this work is motivated from ImageCLEF's domain adaptation 2014 task. Accordingly, five datasets: Caltech-256 [31], ImageNet ILSVRC2012 [38], PASCAL VOC2012 [43], Bing [5], and SUN [60] are used for experiments. SUN dataset is used as the target domain, while the other four are used as source domains. The proposed frameworks and fusion algorithms are evaluated on these five datasets. The results are compared with the ground truth. Each correct image classification earns one point [1]. Total points earned by a classifier give its accuracy.

5.1 Datasets and feature extraction

The five datasets used for experiments are briefly discussed here. **Caltech-256** is a large-scale dataset containing 30,607 images divided in 256 object categories. Each category consists of at least 80 images. Left and right alignment of objects in this dataset gives better idea of object categorization for real world conditions [31]. **ImageNet ILSVRC2012** is larger in scale and diversity than the other image classification datasets. It contains 14,197,122 annotated images organized by the semantic hierarchy of WordNet. English words form a set of synonyms, and that set is termed as synset. Therefore, each concept in WordNet, described by multiple words, is called synonym set or synset. ImageNet populates 21,841 synsets of WordNet with an average of 650 manually verified and full resolution images. [38]. **PASCAL VOC2012** dataset consists of annotated photographs collected from the flickr photo-sharing web site. This dataset is considered as a benchmark for visual object category recognition in vision and machine learning communities. It has 11,530 images, organized in 20 object classes [43]. Similar to Caltech-256, **Bing** contains 256 categories. In addition, 300 web images are selected per category from Bing search engine [5]. It consists of 209,826 number of images. **SUN** dataset contains 108,754 annotated images of environmental scenes, places, and objects. It has 397 categories with a minimum of 100 images per category. It also uses the concept of WordNet English dictionary. The images for this dataset are collected using an online image search engine corresponding to the textual queries of scenes, places, and environments [60].

Twelve categories from all five datasets, namely *Aeroplane*, *Bike*, *Bird*, *Boat*, *Bottle*, *Bus*, *Car*, *Dog*, *Horse*, *Monitor*, *Motorbike*, and *People* are selected. From each of the selected category, 50 images are randomly chosen. In total 600 images are selected from each dataset. Table 10 shows images of these object categories from five different domains. It can be easily seen that the visual appearance of the objects varies in different domains even if the object labels are same. Due to this reason an image classification model fails when tested for images from other datasets.

To maintain consistency and worth of the results, dense SIFT features of the images and their category as provided by ImageCLEF2014DA task are used for experiments. For feature extraction, dense SIFT descriptors are computed on 2×2 patches over a grid with a spacing of 128 pixels for all the datasets. The dense features are vector quantized into 256 visual words using k-means clustering. In summary, an image is first converted in to 2×2 spatial histograms over the 256 visual words which finally resulted in 1024-dimensional feature [1].

5.2 Baseline approaches

ImageCLEF domain adaptation challenge 2014 provided one set of results considered as the baseline approach. The participants in this competition were supposed to produce

Table 10 Images of various object categories from five different domains

Dataset/Object	Bike	Bird	Boat	Bottle	Bus	Car	Dog	Horse	Monitor	People
Bing										
Caltech										
ImageNet										
Pascal										
Sun										

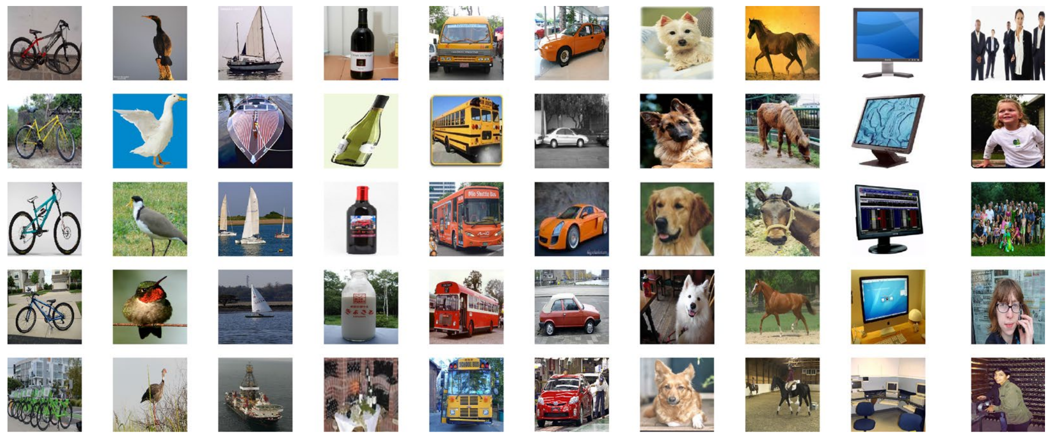

Table 11 Accuracy of the traditional machine learning algorithms on four source domains as well as combined dataset

Algorithm	Accuracy				
	Bing	Caltech	ImageNet	Pascal	Combined datasets
IB1	100.00	99.83	100.00	99.00	99.70
Random forest	99.67	99.50	100.00	98.16	99.20
SMO	100.00	99.83	99.83	98.66	98.67
AdaBoost	87.33	88.33	88.66	87.50	90.04
Decision tree	89.16	90.33	89.16	88.16	89.04
Dagging	30.00	48.16	33.16	27.30	33.50
SimpleLogistic	21.67	35.67	27.16	26.80	32.78
Naive Bayes	18.16	29.18	22.50	14.66	19.70
Classifiaction via regression	15.83	33.83	16.33	14.00	17.50

results better than this base line system. Nineteen groups participated in this domain adaptation challenge, but only three groups have submitted their runs and results. **I Rank: XRCE group** was the winner of this task by reporting the best classifier for this task. They used two domain adaptation algorithms named as adaptive boosting and metric learning methods. In order to increase the overall accuracy of the framework, the results obtained with both the approaches were fused with different fusion methods. The fusion methods used by them are unweighted majority voting, weighted majority voting, and Bayesian rule [14]. **II Rank: Hubert Curien Lab (HCL) group** although submitted the accuracy obtained for each object category, but did not provide any type of working notes or even the details of their algorithm. **III Rank: Idiap group** were the organizers of this task. This group used high-level learning to learn (H-L2L) algorithm that combines the confidence output with high-level integration scheme. They have not tried parameter tuning because their main objective was to provide the baseline model to the participants of the challenge [52]. As the problem focused in this work is adapted from ImageCLEF domain adaptation challenge 2014, the results mentioned in this challenge are used as baseline approaches here to compare the results obtained in this work.

5.3 Results obtained with the framework for traditional machine learning algorithms

The performance of various machine learning algorithms is carefully observed on four source domain datasets individually as well as on their combination by using WEKA tool as discussed in Sect. 4.1. For tenfold cross-validation, some of the results are given in Table 11. It is clearly visible that IB1, Random Forest, SMO, AdaBoost, and Decision Tree (J48) perform well with more than 85% accuracy. *IB1* is instance-based learning algorithm and extension of k-NN classification algorithm. *Decision Tree* is a graph that learns the behavior of all possible outcome of the variable. *Random forest* is an ensemble learning method for classification, regression and other tasks. It is trained for constructing multiple decision trees at training time. *SMO* is heuristic to solve the support vector machine (SVM) problem. *AdaBoost* is meta learning algorithm that combines weak classifiers into strong classifier. Since Random Forest is an ensemble decision tree and also gives better results as compared to Decision Tree (J48), it is preferred over Decision Tree (J48). Based on this discussion, four traditional machine learning algorithms, i.e., IB1, Random Forest, SMO, and AdaBoost are selected to train the system using concatenated feature vectors of all the datasets. During testing, all classifiers are tested using images from target dataset. Every classifier gives some rank label corresponding to probability or score values. Suppose M classifiers are selected and testing dataset has N classes, then every classifier predicts the $1 \times N$ vector as a result where the first value is the most appropriate label predicted by a particular classifier. The $M \times N$ dimension matrix represents the order of class labels from different classifiers. The proposed fusion methods are used over this matrix to find the final predicted label of the image.

Table 12 presents the results obtained for the proposed fusion methods used with the framework for traditional machine learning algorithms. Based on the overall accuracy, this framework is more accurate as compared to the base system as well as IDIAP. However, it performs far below than HCL and XRCE. Iterative depth-based fusion extends depth-based fusion method, and the possible cause of performance degradation here is that most of the time classification is coming from the best classifier instead of fusion of the results from all classifiers. Although performance of traditional machine learning algorithms improves with fusion methods, still obtained results highlight the limitation of it to address the issue of domain adaptation.

5.4 Results obtained with the framework for domain adaptation algorithms

The framework given in Fig. 5 is used to develop the system by fusing the results obtained with NCM and transfer learning domain adaptation algorithms on different domains. The obtained results are summarized in Table 12.

Results obtained with Metric Learning At any moment in the training phase, each NCM classifier takes single-source domain only and learns the projection matrix W using SGD training. All the trained classifiers predict the labels for the same target domain during testing. This method takes approximately $20,000 - 30,000$ iterations to learn the required projection matrix. The accuracy of NCM classifier as shown in Table 12 tells that depth-based fusion gives slightly better results as compared to iterative depth-based fusion. Also, the results obtained through NCM classifier are better than those obtained with the framework on traditional machine learning algorithms. Still, it fails to achieve better accuracy as compared to the first and second winner of the competition. Still, some more improvement in classification accuracy is required that lead us to see the performance obtained with transfer learning framework for domain adaptation.

Results obtained with transfer learning As discussed in 4.2.2, transfer learning framework (TLF) is using TrAdaBoost algorithm as classifier. Further, TrAdaBoost algorithm

Table 12 Comparison of the results obtained with the proposed frameworks (implemented with the proposed fusion methods) with the winners of ImageCLEF's 2014 domain adaptation task

Category	Base line Approaches				MLF		NCMF		TLF	
	Base system	IDIAP (3rd)	HCL (2nd)	XRCE (1st)	DB	IDB	DB	IDB	DB	IDB
Aeroplane	4	3	36	41	18	24	4	4	31	39
Bike	2	1	7	12	12	4	7	4	31	25
Bird	0	0	15	15	3	6	9	11	31	37
Boat	1	4	5	18	2	3	3	6	26	37
Bottle	2	3	25	20	5	6	12	8	30	26
Bus	4	6	10	23	0	0	8	5	29	23
Car	10	7	13	17	8	6	7	9	19	24
Dog	5	3	8	8	1	1	1	6	13	15
Horse	6	2	6	17	1	1	7	5	21	15
Monitor	6	3	15	28	2	3	4	6	17	26
Motorbike	3	3	7	12	4	6	10	4	16	19
People	5	10	11	17	10	4	1	3	23	14
Average	48	45	158	228	66	64	73	71	277	300

DB Depth-based fusion

IDB Iterative depth-based fusion

MLF Traditional Machine Learning Framework

NCMF NCM framework

TLF Transfer learning framework

uses decision stump as a weak learner here. After several iterations decision stumps gives higher weights to those instances that are similar to target instance and less weight to dissimilar instances. Decision stump [57] is a machine learning algorithm consisting of a one-level decision tree. It has one root node (internal node) and two leaf nodes (terminal nodes). This algorithm is also known as 1-rules, because it makes prediction-based value of single feature vector. Number of decision stumps is equal to the length of the feature vector. One leaf node represents a class and the other leaf node represents other classes. Since features are continuous, threshold value of feature is to be selected. The trained classifier predicts probability values of an image from the target domain to be in each of the twelve classes. It is an instance-based learning which assigns higher weights to similar instances and lower weights to dissimilar ones. The algorithm runs on 1-vs-all strategy, i.e., for every class a model is created by considering labels of the same class as +1 and labels from other classes as −1. Twelve models corresponding to each of the twelve classes are created here.

As shown in Table 12, TLF with both the proposed fusion algorithms outperforms other frameworks proposed for domain adaptation problem. With both the fusion methods, it successfully increases the class label predictions by almost 20%–30% in comparison with the base system. The accuracy reported is superior to HCL and XRCE, who are ranked I and II in ImageCLEF 2014 domain adaptation challenge. TLF correctly classifies 277 and 300 images as compared to only 228 correct classifications by XRCE. Although both depth and iterative depth-based fusion methods perform better, but iterative depth-based fusion gives the best results. It is able to achieve 300 correct classification for 600 test images drawn from the SUN dataset.

It can be seen that other methods perform well on some of the categories while their performance drastically drops for other categories. For example, base systems reports 10 correct classifications for *Car* category as compared to all incorrect classifications for *Bike* category. Similarly, the winner of the competition, XRCE achieved only 8 correct classifications for *Dog* category as compared to 41 correct classifications in case of *Aeroplane* category. It is interesting to see that TLF performs equally well for almost all classes as compared to other methods, because it is an instance-based method and learns the boundaries of all the classes.

5.5 Time complexity of the proposed methods

The computational cost of three frameworks and two fusion methods are proposed in this work is discussed here. Let, n is the size of the dataset, f is its dimensionality, and k is the number of class labels, then Depth-Based Fusion method is $O(\log k)$. Iterative Depth-Based Fusion iterates over the depth till the unique class label for an image is identified. Thus, the complexity of this method is $O(k \log k)$. The traditional machine learning framework is tested with four algorithms. Complexity of SVM, random forest, and k-NN are same and is equivalent to $O(nf)$ whereas the complexity of AdaBoost is $O(fn \log n)$. So, the overall complexity of this framework is $O(fn \log n)$. Lastly, the complexity of Metric Learning Framework and Transfer Learning Framework are $O(f \log(1/n^{0.5}) + \alpha)$ and $O(nf \log n + \alpha)$, respectively, where α is the complexity of fusion algorithm used within the framework.

6 Conclusion

Most of the fusion methods work quite well when only topmost prediction of classifier results are being fused, but fail when applied to all predicted values of the classifier. The results obtained with the proposed depth and iterative depth-based fusion methods show that "the concept of 'depth' can be effectively utilized to design fusion methods which may lead to development of more accurate classifiers for domain adaptation". Usually traditional machine learning algorithms do not perform well for domain adaptation task. But their performance improves by applying the proposed depth and iterative depth-based fusion methods. Still, it is not sufficient to work with, and therefore domain adaptation-based frameworks are developed. These frameworks are explored with the most popular and efficient metric learning (i.e., NCM) and transfer learning (i.e., TrAdaBoost) algorithms. Although NCM-based framework performs better as compared to traditional machine algorithms-based framework, but due to its dependence on mean it does not perform well in the presence of images with large difference in their distribution. However, adaptive nature of transfer learning helps to deal with such situations in a better way. The same is also examined with the transfer learning framework. The proposed transfer learning framework with both the fusion algorithms outperforms the results obtained by the winner of ImageCLEF2014 challenge. This opens up further research directions for domain adaptation problem.

References

1. http://www.imageclef.org/2014/adaptation
2. Bade S (2013) Random serial dictatorship: the one and only. In: Unpublished manuscript. Available from http://www.coll.mpg.de/sites/www.coll.mpg.de/files/workshop/newproof5.pdf
3. Bai B, Weston J, Grangier D, Collobert R, Sadamasa K, Qi Y, Chapelle O, Weinberger K (2010) Learning to rank with (a lot of) word features. Inf Retr 13(3):291–314
4. Baldwin JM (1926) The technique of the nanson preferential majority system of election. Proc R Soc Vic 39:42–52
5. Bergamo A, Torresani L (2010) Exploiting weakly-labeled web images to improve object classification: a domain adaptation

approach. In: Advances in neural information processing systems, pp 181–189
6. Bishop CM (2006) Pattern recognition and machine learning. Springer, Berlin
7. Boiman O, Shechtman E, Irani M (2008) In defense of nearest-neighbor based image classification. In: IEEE conference on computer vision and pattern recognition, 2008. CVPR 2008. IEEE, pp 1–8
8. Borda JC (1784) Memoire sur les elections au scrutin. Histoire de l'Academie Royale des Science
9. Camps R, Mora X, Saumell L (2012) A continuous rating method for preferential voting: the complete case. Soc Choice Welf 39(1):141–170
10. Caputo B, Patricia N (2014) Overview of the imageclef 2014 domain adaptation task. In: ImageCLEF 2014: overview and analysis of the results
11. Cha SH, An YJ (2012) Taxonomy and nomenclature of preferential voting methods. In: Proceedings of the world congress on engineering and computer science, vol 1
12. Chai J, Liu H, Chen B, Bao Z (2010) Large margin nearest local mean classifier. Signal Process 90(1):236–248
13. Chen M, Weinberger KQ, Blitzer J (2011) Co-training for domain adaptation. In: Advances in neural information processing systems, pp 2456–2464
14. Chidlovskii B, Csurka G, Gangwar S (2014) Assembling heterogeneous domain adaptation methods for image classification. In: ImageCLEF Domain Adaptation Challenge, pp 1–14
15. Dai W, Yang Q, Xue GR, Yu Y (2007) Boosting for transfer learning. In: Proceedings of the 24th international conference on machine learning ACM, pp 193–200. doi:10.1145/1273496.1273521
16. Daumé III H, Kumar A, Saha A (2010) Frustratingly easy semi-supervised domain adaptation. In: Proceedings of the 2010 workshop on domain adaptation for natural language processing. Association for Computational Linguistics, pp 53–59
17. Daume H III, Marcu D (2006) Domain adaptation for statistical classifiers. J Artif Intell Res 26:101–126
18. Davis JV, Kulis B, Jain P, Sra S, Dhillon IS (2007) Information-theoretic metric learning. In: Proceedings of the 24th international conference on machine learning. ACM, pp 209–216
19. De Condorcet N et al (2014) Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix. Cambridge University Press, Cambridge
20. Donahue J, Hoffman J, Rodner E, Saenko K, Darrell T (2013) Semi-supervised domain adaptation with instance constraints. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 668–675
21. Duan L, Tsang IW, Xu D, Chua TS (2009) Domain adaptation from multiple sources via auxiliary classifiers. In: Proceedings of the 26th annual international conference on machine learning ACM, pp 289–296. doi:10.1145/1553374.1553411
22. Duan L, Xu D, Tsang I (2012) Learning with augmented features for heterogeneous domain adaptation. arXiv preprint arXiv:1206.4660
23. Durand F, Mathieu F, Noirie L (2013) On the manipulability of voting systems: application to multi-operator networks. In: 9th international conference on network and service management. IEEE, pp 292–297
24. Erdélyi G, Piras L, Rothe J (2010) Bucklin voting is broadly resistant to control. arXiv preprint arXiv:1005.4115
25. Rodner E, Hoffman J, JDTDaKS (2013) Towards adapting imagenet to reality: scalable domain adaptation with implicit low-rank transformations. arXiv preprint arXiv:1308.4200 pp 289–296
26. Freund Y, Schapire R, Abe N (1999) A short introduction to boosting. J Jpn Soc Artif Intell 14(771–780):1612
27. Geng B, Tao D, Xu C (2011) Daml: domain adaptation metric learning. IEEE Trans Image Process 20:2980–2989. doi:10.1109/TIP.2011.2134107
28. Globerson A, Roweis ST (2005) Metric learning by collapsing classes. In: Advances in neural information processing systems, pp 451–458
29. Gong B, Shi Y, Sha F, Grauman K (2012) Geodesic flow kernel for unsupervised domain adaptation. In: IEEE conference on computer vision and pattern recognition. IEEE, pp 2066–2073
30. Gopalan R, Li R, Chellappa R (2011) Domain adaptation for object recognition: an unsupervised approach. In: IEEE international conference on computer vision. IEEE, pp 999–1006
31. Griffin G, Holub A, Perona P (2007) Caltech-256 object category dataset. California Institute of Technology, Pasadena, pp 1473–1480
32. Guillaumin M, Verbeek J, Schmid C (2009) Is that you? Metric learning approaches for face identification. In: IEEE 12th international conference on computer vision. IEEE, pp 498–505
33. Hara T, Miyao Y, Tsujii JI (2005) Adapting a probabilistic disambiguation model of an HPSG parser to a new domain. In: IJCNLP, pp 199–210
34. Huang P, Wang G, Qin S (2012) Boosting for transfer learning from multiple data sources. Pattern Recognit Lett 33(5):568–579
35. Jacob Goldberger Sam Roweis GH, Salakhutdinov R (2004) Neighbourhood components analysis. In: Advances in neural information processing systems, pp 513–520
36. Jiang J (2008) A literature survey on domain adaptation of statistical classifiers. URL: http://sifaka.cs.uiuc.edu/jiang4/domainadaptation/survey
37. Hoffman J, Guadarrama S, Tzeng ES, Hu R, Donahue J, Girshick R, Darrell T, Saenko K (2014) Lsda: Large scale detection through adaptation. In: Advances in neural information processing systems, pp 3536–3544
38. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
39. Kranjc J, Smailović J, Podpečan V, Grčar M, Žnidaršič M, Lavrač N (2015) Active learning for sentiment analysis on data streams: methodology and workflow implementation in the clowdflows platform. Inf Process Manag 51(2):187–203
40. Kulis B, Saenko K, Darrell T (2011) What you saw is not what you get: domain adaptation using asymmetric kernel transforms. In: IEEE conference on computer vision and pattern recognition (CVPR 2011), vol 24. pp 1785–1792. doi:10.1109/CVPR.2011.5995702
41. Kumar A, Shekhar S (2011) Personal identification using multibiometrics rank-level fusion. IEEE Trans Syst Man Cybern Part C Appl Rev 41(5):743–752
42. Long M, Wang J, Ding G, Pan SJ, Yu PS (2014) Adaptation regularization: A general framework for transfer learning. IEEE Trans Knowl Data Eng 26(5):1076–1089
43. Everingham M, Van Gool L, Williams C, Winn J, Zisserman A (2005) Pascal visual object classes challenge results, vol 1, p 7. Available from http://www.pascal-network.org/
44. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD Explor Newsl 11(1):10–18
45. Matasci G, Volpi M, Kanevski M, Bruzzone L, Tuia D (2015) Semisupervised transfer component analysis for domain adaptation in remote sensing image classification. IEEE Trans Geosci Remote Sens 53(7):3550–3564
46. Guillaumin M, Mensink T, Verbeek J, Schmid C (2009) Tagprop: discriminative metric learning in nearest neighbor models for

image auto-annotation. In: IEEE 12th international conference on computer vision. IEEE, pp 309–316
47. Molina-González MD, Martínez-Cámara E, Martín-Valdivia MT, Ureña-López LA (2015) A spanish semantic orientation approach to domain adaptation for polarity classification. Inf Process Manag 51(4):520–531
48. Morais DC, de Almeida AT (2012) Group decision making on water resources based on analysis of individual rankings. Omega 40(1):42–52
49. Nowak E, Jurie F (2007) Learning visual similarity measures for comparing never seen objects. In: IEEE conference on computer vision and pattern recognition, 2007. CVPR'07. IEEE, pp 1–8
50. Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359
51. Patel VM, Gopalan R, Li R, Chellappa R (2015) Visual domain adaptation: a survey of recent advances. IEEE Signal Process Mag 32(3):53–69
52. Patricia N, Caputo B (2014) Learning to learn, from transfer learning to domain adaptation: a unifying perspective. In: 2014 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 1442–1449
53. Rai P, Khanna P (2014) A gender classification system robust to occlusion using gabor features based (2d) 2 pca. J Vis Commun Image Represent 25(5):1118–1129
54. Saenko K, Kulis B, Fritz M, Darrell T (2010) Adapting visual category models to new domains. In: Computer vision–ECCV 2010. Springer, pp 213–226
55. Mensink T, Verbeek J, Perronnin F, Csurka G (2013) Distance-based image classification: generalizing to new classes at near-zero cost. IEEE Trans Pattern Anal Mach Intell 35(11):2624–2637
56. Tommasi T, Caputo B (2013) Frustratingly easy nbnn domain adaptation. In: 2013 IEEE international conference on computer vision (ICCV). IEEE, pp 897–904
57. Torralba A, Murphy KP, Freeman WT (2007) Sharing visual features for multiclass and multiview object detection. IEEE Trans Pattern Anal Mach Intell 29(5):854–869
58. Van Erp M, Vuurpijl L, Schomaker L (2002) An overview and comparison of voting methods for pattern recognition. In: Proceedings of eighth international workshop on frontiers in handwriting recognition. IEEE, pp 195–200
59. Wang Z, Hu Y, Chia LT (2010) Image-to-class distance metric learning for image classification. In: Computer vision–ECCV 2010. Springer, pp 706–719
60. Xiao J, Ehinger KA, Hays J, Torralba A, Oliva A (2016) Sun database: exploring a large collection of scene categories. Int J Comput Vis 119(1):3–22
61. Xu J, Ramos S, Vazquez D, Lopez AM (2014) Domain adaptation of deformable part-based models. IEEE Trans Pattern Anal Mach Intell 36(12):2367–2380
62. Yao T, Pan Y, Ngo CW, Li H, Mei T (2015) Semi-supervised domain adaptation with subspace learning for visual recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2142–2150
63. Yao Y, Doretto G (2010) Boosting for transfer learning with multiple sources. In: IEEE conference on computer vision and pattern recognition, pp 1855–1862