I.J. Information Engineering and Electronic Business, **2024, 2, 39-53**
Published Online on April 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2024.02.04

Unified Domain Adaptation with Discriminative Features and Similarity Preservation

Obsa Gilo*
CSE, Indian Institute of Technology Patna, 801103, Bihar, India
Email: obsa_1921cs33@iitp.ac.in
ORCID iD: https://orcid.org/0000-0002-9562-8129
*Corresponding Author

Jimson Mathew
CSE, Indian Institute of Technology Patna, 801103, Bihar, India
Email: jimson@iitp.ac.in
ORCID iD: https://orcid.org/0000-0001-8247-9040

Samrat Mondal
CSE, Indian Institute of Technology Patna, 801103, Bihar, India
Email: samrat@iitp.ac.in
ORCID iD: https://orcid.org/0000-0002-2159-3410

Received: 01 August, 2023; Revised: 13 September, 2023; Accepted: 17 October, 2023; Published: 08 April, 2024

Abstract: In visual domain adaptation, the goal is to train effective classifiers for the target domain by leveraging information from the source domain. In unsupervised domain adaptation, the source domain provides labeled data while the target domain lacks labels. However, it is crucial to recognize that the source and target domains have different underlying distributions despite sharing the same label space. Directly applying source domain information to the target domain often leads to poor performance due to the distribution gap between the two domains. Unsupervised do- main adaptation aims to bridge this gap and improve performance. We introduce a comprehensive UDADFSP (Unified Domain Adaptation with Discriminative Features and Similarity Preservation) de- signed explicitly for unsupervised domain adaptation to tackle these challenges. Our framework focuses on incorporating discriminative and invariant features. We employ clustering with entropy regularization on the unlabeled target domain to refine the neighbor relationships. This step significantly enhances the alignment between the target and source domains, facilitating a more effective adaptation. Furthermore, we seamlessly incorporate discriminative features while preserving similarity in the source and target domains. We carefully balance the discrimination and similarity aspects by considering linear and non-linear data representations. Extensive testing demonstrates that learning discriminative and similarity features in the same feature space yields significant improvements over several state-of-the-art domain adaptation techniques. In a comparative evaluation, our approach surpasses several existing methods across four diverse cross-domain visual tasks and the Amazon re- view sentiment analysis task.

Index Terms: Distribution shift, Discriminative features, Entropy regularization, Domain adaptation.

1. Introduction

The core principle of conventional machine learning hinges on the belief that training and testing data stem from a shared distribution, assuming independence and identically. Nevertheless, this assumption falls short due to the dynamic nature of data, which can vary over time and across different environments. Consequently, the model learned from the training data may exhibit inferior performance when con- fronted with biases or domain shifts in the dataset. Numerous researchers [1, 2, 3, 4] have proposed various algorithms and strategies to tackle these challenges. Recent studies have emphasized the importance of employing domain-aware algorithms to address cross-domain learning transfer issues [1].

Applications across domains such as computer vision, hyper-spectral images, natural language processing, and time series analysis have demonstrated potential for adaptability [5]. When source and target data distributions differ, domain adaptation becomes relevant. For example, in visual recognition, the testing and training distributions may differ due to various factors, such as camera quality, sensor type, resolution, background clutter, environmental conditions, and view angle. Obtaining labeled data for visual recognition tasks is time-consuming and demanding, often resulting in a significant portion of unstructured or unlabeled data. Domain adaptation offers a solution to these challenges by utilizing labeled data from a source domain to improve the performance of the target domain, even when labeled data in the target domain is limited or unavailable. Depending on the availability of labeled target data, domain adaptation can be classified into three forms: supervised, semi-supervised, and unsupervised domain adaptation. The target domain is labeled in a supervised setting. In semi-supervised adaptation, there is a labeled data, while in the unsupervised target domain, bridging the domain gap between components to align their distributions is a crucial challenge in domain adaptation. Fig. 1 illustrates the concepts of domain adaptation.

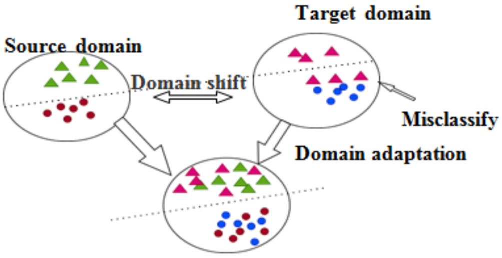


Fig. 1. In the illustrations of domain adaptation, circles and triangles are used as visual representations of different classes, representing distinct categories

Most recent studies have independently investigated three learning techniques for domain adaptation:

1. Feature-based finds domain-invariant features and de- creases distribution simultaneously [1, 6, 7, 8].
2. Instance re-weighting decreases distributions by reweighting the source domain, training the classifier on the re-weighted domain, and testing the target domain [9, 10, 11]. Re-weighting permits a sample of the data from the source domain to seek learning in the target domain.
3. Adversarial-based domain adaptation learns a biased representation from labels in the source domain, and then uses an asymmetric mapping known as a domain- adversarial loss to map the target data to the same space [12, 13].

This study employs an approach based on features, combining graph clustering and entropy regularization to handle unannotated target domains. The method utilizes adaptive neighbors to enhance the adaptation process effectively [14]. Unlike many previous studies that have prioritized maximizing domain invariance to improve feature transferability at the cost of discriminative features, we advocate for integrating both invariant and discriminative feature domains. This approach aims to balance domain adaptation and preserving discriminative information for more effective and accurate model performance. Clustering and entropy regularization techniques address the challenge of unlabeled target do- mains. Traditional k-means clustering allocates individual data points to a separate group, posing limitations. In contrast, FKM clustering provides probabilistic or fuzzy clustering, offering a more flexible approach [15]. Entropy regularization provides confidence in assigning each affinity matrix [14, 15]. The primary objective of this paper is to combine discriminative features and promote similarity preservation across domains in unsupervised domain adaptation (UDA) using entropy regularization and adaptive neighbor's clustering. Moreover, the study considers the domains' geometric structures and statistical distributions to achieve more robust and effective domain adaptation. To summarize, the following are our significant contributions:

- Retain the distinct features of both the source and the target domains.
- The geometrical structures are regulated and similarity is strengthened by reducing domain scatterings using Maximum Mean Discrepancy (MMD).
- For source domains, intra-class distance is minimized while inter-class distance is maximized.
- The clustering of the target domain is carried out using entropy regularization and adaptive neighbors, facilitating more accurate and effective adaptation.
- The experimental setup includes handling linear and non-linear data using kernel functions.

The rest of the paper is organized as follows. The related approach for domain adaptation is discussed briefly in section 2. Section 3 elaborates on our approaches. Then, we provide our approach, followed by objective function formulations in section 4. Section 5 discusses experimental findings and a thorough evaluation of five benchmark datasets. Lastly, in section 6, we conclude and provide a future direction for our study.

2. Related Works

The Joint Geometrical and Statistical Alignment (JGSA) approach aims to align both domains by projecting them into subspaces and minimizing geometrical and statistical distributions [16]. Nevertheless, the main emphasis of JGSA lies in achieving domain-invariant features and enhancing transferability between components, neglecting to incorporate domain-discriminant features and preserving similarity across domains. This omission can lead to negative transfer, resulting in poor performance and inconsistent results. Therefore, our study considers handling negatively transfer- able features to mitigate such issues.

Transfer Feature Learning with Joint Distribution Adaptation (JDA) aims to construct new feature representations by minimizing the divergence between the source and target do- mains regarding marginal and conditional distributions, using dimensional reduction techniques [17]. However, JDA does not explicitly consider specific features and the geometrical information of the data. These aspects are not directly addressed in the JDA method. Moreover, it fails to consider the significance of intra-class and inter-class distances for grasping appropriate information in the data. (TJM) [18] procedure seeks to jointly approximate feature representations with examples while re-sample instances across domains. It incorporates dimensional reduction principles and creates new feature representations invariant to domain distribution differences. While TJM prioritizes domain-consistent features, it ignores the preservation of discriminative features and the similarity between domains. These aspects are not explicitly addressed within the TJM approach. Furthermore, TJM does not maintain any source and target domain information.

The DJP-MMD approach for Domain Adaptation, as introduced in [19], seeks to simultaneously account for transferability and discriminability by leveraging joint Likelihood MMD. Nevertheless, DJP-MMD overlooks the consideration of sample dimensionality and the preservation of geometrical structures within domains. Deep CORAL [20] is a UDA technique that uses second-order statistics to lessen the issuance discrepancy between the source and target domains. It caters to situations where the target domain lacks labels while the source domain is labeled. CORAL [21], an approach, aligns the second-order statistics of the source and target distributions by employing linear transformations. Deep CORAL [22] builds upon this concept and extends it to learn non- linear further on this idea by introducing non-linear modifications into DNNs, allowing for the matching of correlations among layer activations.

The incorporation of entropy regularization and adaptive neighbors in the ERCAN method [14, 23] allows it to surpass the limitations of k-means and spectral clustering in unlabeled datasets, leading to improved performance and greater robustness. ERCAN [24] also combines graph clustering with a Laplacian rank constraint matrix and introduces an affinity matrix. This further enhances the method’s ability to handle unsupervised domain adaptation challenges. While ERCAN utilizes the l-0 norm for adaptive neighbors, which enables sparse and robust segmentation, it does not consider domain adaptability. Our study addresses the challenges of an unsupervised target domain through entropy regularization and adaptive neighbor clustering.

3. Proposed Approach

3.1. Problem definition and notations

This section addresses the problem statement and introduces relevant terminologies and notations.

Table 1. Glossary of notations

Description	Notation
Target and Source domain	D_{tr} ,D_{sr}
Projection vectors for the source and target domains	C, D
Count of target/source examples	n_t , n_s
The marginal distribution of the target/source domain	P_t ,P_s
Source domain, along with their corresponding labels	X_s,Y_s
Target domain along with their corresponding labels	X_t ,Y_t
Transferred elements	m
Between class/Within-class scattered matrix	S_b, S_w
Basis vectors of the subspace	k
The conditional distribution of target domain	$p_t\ (y_t \mid x_t\ )$
Regularizer parameter	λ
The conditional distribution of source domain	$p_s(y_s \mid x_s)$
Kernel matrix of source /target	Ks, Kt
Dimension of each sample/samples in both domains	d, N
Transpose	T
Identity matrix	I
Affinity matrix	$s_{i\,j}$
Target and source column vectors have a value of one	1_t , 1_s
Input data includes target and source examples	X
Class, number of class c, C_l	
Parameters controlling the trade-off	β, μ, γ

The source domain, denoted by$X_s \in \mathbb{R}^{D\times n_s}$, originates from the distribution$P_s(X_s)$, while the target domain, represented as $X_t \in \mathbb{R}^{D\times n_t}$, is drawn from the distribution $P_t(X_t)$. In this context, D refers to the dimensionality of the examples, while n_s and n_t indicate the respective number of instances in the source and target domains. We focus on the UDA situation due to the availability of ample labeled data in the source domain, denoted as $\mathcal{D}_{sr} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_s}, \mathbf{x}_i \in \mathbb{R}^D$, and unlabeled data in the target domain, denoted as $\mathcal{D}_{tr} = \{(\mathbf{x}_j)\}_{j=1}^{n_t}, \mathbf{x}_j \in \mathbb{R}^D$, during the training stage. We assume that the feature and label spaces are the same across domains: $\mathcal{X}_s = \mathcal{X}_t$ and$\mathcal{Y}_s = \mathcal{Y}_t$, but the distributions $P_s(X_s)$ and $P_t(X_t)$ differ due to domain shift. The assumption of a unified transformation$\phi(\cdot)$, satisfying $P_s(\phi(X_s)) = P_t(\phi(X_t))$ and$P_s(Y_s \mid \phi(X_s)) = P_t(Y_t \mid \phi(X_s))$, is invalid in the presence of domain shift in the dataset. Table 1 briefly explains each symbol.

3.2. Methods

Our goal formulation involves the subsequent steps: A) Retaining discriminative information from the source domain. B) Maintaining discriminatory information of the target domain. C) Employing MMD (Maximum Mean Discrepancy) to ensure similarity preservation between the two domains. D) Minimization of subspace divergence. E) Incorporating entropy regularization enables clustering within the target domain with adaptive neighbors.

A. Retaining discriminative information from the source domain:

Let us denote the projected vectors for the source and target domains as C and D, respectively. The preservation of discriminative properties is feasible due to the availability of ample labeled source data for unsupervised domain adaptation. We employ inter-class and intra-class strategies to minimize variation within categories and maximize variance across categories, thereby retaining the discriminative characteristics. Given the labeled nature of the source domain, we utilize Linear Discriminant Analysis (LDA) to reduce dimensionality and construct a scatter matrix that captures both within-class and between-class variations

$$\max_{C} \mathrm{Tr}(C^T S_b C) \tag{1}$$

$$\min_{C} \mathrm{Tr}(C^T S_w C) \tag{2}$$

$$S_w = \sum_{c=1}^{C_l} X_s^{(c)} H_s^{(c)} \left(X_s^{(c)}\right)^T \tag{3}$$

$$\mathrm{S_b} = \sum_{\mathrm{c}=1}^{\mathrm{C_l}} \mathrm{n_s^{(c)}} \left(\mathrm{m_s^{(c)}} - \overleftarrow{\overline{\mathrm{m}}}_{\mathrm{s}}\right)\left(\mathrm{m_s^{(c)}} - \overleftarrow{\overline{\mathrm{m}}}_{\mathrm{s}}\right)^{\mathrm{T}} \tag{4}$$

where $\mathrm{X_s^{(c)}} \in \mathbb{R}^{\mathrm{D}\times \mathrm{n_s^{(c)}}}$ is the source samples belong to the categories $\mathrm{c}, \mathrm{m_s^{(c)}} = \frac{1}{\mathrm{n_s^{(c)}}}\sum_{\mathrm{i}=1}^{\mathrm{n_s^{(c)}}} \mathrm{x_i^{(c)}}, \overleftarrow{\overline{\mathrm{m}}}_{\mathrm{s}} = \frac{1}{\mathrm{n_s}}\sum_{\mathrm{i}=1}^{\mathrm{n_s}} \mathrm{x_i}$, $\mathrm{H_s^{(c)}} = \mathrm{I_s^{(c)}} - \frac{1}{\mathrm{n_s^{(c)}}} 1_{\mathrm{s}}^{(\mathrm{c})}\left(1_{\mathrm{s}}^{(\mathrm{c})}\right)^{\mathrm{T}}$ is the within class centering matrix $\mathrm{c}, \mathrm{I_s^{(c)}} \in \mathbb{R}^{\mathrm{n_s^{(c)}}\times \mathrm{n_s^{(c)}}}$ is the identity matrix, $1_{\mathrm{s}} \in \mathbb{R}^{\mathrm{n_s^{(c)}}}$.

B. Maintaining discriminatory information of the target domain:

In the absence of labeled information in the target domain, we employ pseudo-labeling data obtained through 1-NN classification. We apply PCA to map the destination domain features to a low-dimension subspace to capture the distinguishing features in the target domain.

$$\max_{\mathrm{D}} \mathrm{Tr}(\mathrm{D^T S_t D}) \tag{5}$$

$$\mathrm{S_t} = \mathrm{X_t H_t X_t^T} \tag{6}$$

$\mathrm{S_t}$ represents the scatter matrix of the target domain, and $\mathrm{H_t} = \mathrm{I_t} - \frac{1}{\mathrm{n_t}} 1_{\mathrm{t}} 1_{\mathrm{t}}^{\mathrm{T}}$ is the centering matrix, where $1_{\mathrm{t}} \in \mathbb{R}^{\mathrm{n_t}}$ is an all-ones vector of columns . By maximizing the variance in the target domain, we aim to capture discriminative features. Furthermore, we take into account both likeness and discriminability within classes and between classes in the target domain to achieve a more effective domain adaptation.

C. Employing MMD (Maximum Mean Discrepancy):

MMD (Maximum Mean Discrepancy) is a common method for minimizing distribution discrepancies between two samples (i.e., the source domain and target domain) by comparing sample means. When the means of two sample distributions are equivalent, it signifies they came from the same distribution. We used the following formulas to minimize divergence:-

$$\min_{C,D} \left\| \frac{1}{n_s} \sum_{x_i \in K_s} C^T x_i - \frac{1}{n_t} \sum_{x_j \in K_l} D^T x_j \right\|_F^2 \tag{7}$$

If minimizing the marginal distribution across domains and the K-dimension of the subspace is insufficient, we incorporate conditional distributions after pseudo-labeling the target domains. It is recommended to utilize pseudo-labels predicted by the source domain classifier to exploit conditional class distributions in the target domains. Subsequently, the pseudo-labels in the target domain are iteratively redefined to reduce the discrepancies in the conditional distributions between the two domains. This approach allows for a more refined and accurate domain adaptation process.

$$\min_{C,D} \sum_{c=1}^{C_l} \left\| \frac{1}{n_s^{(c)}} \sum_{\mathbf{x}_i \in X_s^{(c)}} C^T \mathbf{x}_i - \frac{1}{n_t^{(c)}} \sum_{\mathbf{x}_j \in X_t^{(c)}} D^T \mathbf{x}_j \right\|_F^2 \tag{8}$$

We fuse both marginal and conditional distributions (7) and (8)

$$\min_{\mathrm{C,D}} \left\| \frac{1}{\mathrm{n_s}} \sum_{\mathrm{x_i} \in \mathrm{K_s}} \mathrm{C}^{\mathrm{T}} \mathrm{x_i} - \frac{1}{\mathrm{n_t}} \sum_{\mathrm{x_j} \in \mathrm{K_t}} \mathrm{D}^{\mathrm{T}} \mathrm{x_j} \right\|_{\mathrm{F}}^2 + \min_{\mathrm{C,D}} \sum_{\mathrm{c}=1}^{\mathrm{C_l}} \left\| \frac{1}{\mathrm{n_s^{(c)}}} \sum_{\mathrm{x_i} \in \mathrm{X_s^{(c)}}} \mathrm{C}^{\mathrm{T}} \mathrm{x_i} - \frac{1}{\mathrm{n_t^{(c)}}} \sum_{\mathrm{x_j} \in \mathrm{X_t^{(c)}}} \mathrm{D}^{\mathrm{T}} \mathrm{x_j} \right\|_{\mathrm{F}}^2 \tag{9}$$

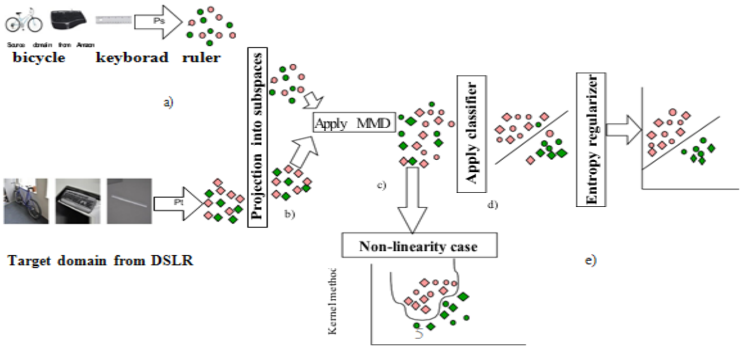


Fig. 2. The proposed approach involves the following steps: a) Extract features and obtain the marginal distributions of the source and target domains, denoted as Ps and Pt, respectively. b) Project both domains into subspaces. c) Utilize Maximum Mean Discrepancy (MMD) to align the domains. d) Apply a classifier, which may not perform well initially. e) Introduce entropy regularization to improve classification accuracy. f) For non-linear datasets, apply a kernel function and entropy regularization.

The final distribution divergence is as follows.

$$\min_{C,D} \mathrm{Tr}\left(\begin{bmatrix} C^T & D^T \end{bmatrix} \begin{bmatrix} M_s & M_{st} \\ M_{ts} & M_t \end{bmatrix} \begin{bmatrix} C \\ D \end{bmatrix} \right) \tag{10}$$

$$\begin{gathered} M_s = X_s\left(O_s + \sum_{c=1}^{C_l} O_s^{(c)}\right) X_s^T, \quad O_s = \frac{1}{n_s^2} 1_s 1_s^T, \\ \left(O_s^{(c)}\right)_{ij} = \begin{cases} \frac{1}{\left(n_s^{(c)}\right)^2} & \mathbf{x}_i, \mathbf{x}_j \in X_s^{(c)} \\ 0 & \text{otherwise} \end{cases} \\ M_t = X_t\left(O_t + \sum_{c=1}^{C_l} O_t^{(c)}\right) X_t^T, \quad O_t = \frac{1}{n_t^2} 1_t 1_t^T, \\ \left(O_t^{(c)}\right)_{ij} = \begin{cases} \frac{1}{\left(n_t^{(c)}\right)^2} & \mathbf{x}_i, \mathbf{x}_j \in X_t^{(c)} \\ 0 & \text{otherwise} \end{cases} \end{gathered} \tag{11}$$

$$\begin{gathered} M_{st} = X_s\left(O_{st} + \sum_{c=1}^{C_l} O_{st}^{(c)}\right) X_t^T, \quad O_{st} = -\frac{1}{n_s n_t} 1_s 1_t^T \\ \left(O_{st}^{(c)}\right)_{ij} = \begin{cases} -\frac{1}{n_s^{(c)} n_t^{(c)}} & \mathbf{x}_i \in X_s^{(c)}, \mathbf{x}_j \in X_t^{(c)} \\ 0 & \text{otherwise} \end{cases} \\ M_{ts} = X_t\left(O_{ts} + \sum_{c=1}^{C_l} O_{ts}^{(c)}\right) X_s^T, \quad O_{ts} = -\frac{1}{n_s n_t} 1_t 1_s^T \\ \left(O_{ts}^{(c)}\right)_{ij} = \begin{cases} -\frac{1}{n_s^{(c)} n_t^{(c)}} & \mathbf{x}_j \in X_s^{(c)}, \mathbf{x}_i \in X_t^{(c)} \\ 0 & \text{otherwise} \end{cases} \end{gathered} \tag{12}$$

D. Minimizing Subspace divergence:

In traditional machine learning, it is assumed that the source and target domains belong to the same distribution, allowing us to project them onto the same subspaces. However, in unsupervised domain adaptation, we independently project the source and target domains into their respective subspaces to minimize their distribution divergence. We utilize the Frobenius norm and simultaneously optimize C and D, preserving discriminative information and similarity among the domains to achieve this. As a result, the two subspaces gradually converge toward each other. The following expressions illustrate how the two subspaces approach each other during this process.

$$\min_{\mathrm{C,D}} \parallel \mathrm{C} - \mathrm{D} \parallel_{\mathrm{F}}^{2} \tag{13}$$

E. Incorporating entropy regularization enables clustering within the target domain with adaptive neighbors:

Standard clustering approaches frequently rely on hard clustering, in which data points are assigned to certain clusters. In real-world circumstances, however, probabilistic or confidence-based cluster allocations may be preferable. For unsupervised target domains, we use graph clustering with entropy regularization. This entropy regularization gives confidence to the affinity matrix assignment. To do this, we employ a simple Gaussian kernel function with Euclidean distance, which is defined as follows:

$$\mathrm{s_{ij}} = \mathrm{s_{ji}} = \mathrm{e}^{\frac{-\left\|\mathrm{x_i}-\mathrm{x_j}\right\|^2}{\mathrm{h}}} \tag{14}$$

Where h heat kernel function is similar to ε neighbors. The Gaussian technique generates adaptable graphs, and self-tuning graphs seek to minimize the number of modified parameters. The likeness between $\mathrm{x_i}$ and $\mathrm{x_j}$ can be detailed as

$$\mathrm{s_{ij}} = \mathrm{e}^{\frac{-\mathrm{d}^2\left(\mathrm{x_i},\mathrm{x_j}\right)}{\sigma_\mathrm{i}\sigma_\mathrm{j}}} \tag{15}$$

Where σ_i shows distance between vertex/nodes $\mathrm{x_i}$ and indicate farthest neighbours $\left(\sigma_\mathrm{i} = \mathrm{d}(\mathrm{x_i}, \mathrm{x_k})\right)$ and $\mathrm{x_k}$ means the kth neighbor for $\mathrm{x_i}$.

Although there are similarities mentioned above, the matrices are generated solely based on distance measures, lacking clear physical interpretations for the combination variables. Additionally, setting these matrices in subsequent graph analysis can make them sensitive to noise in the raw data. Finding a suitable graph with unambiguous physical meaning, explicit neighbor assignment, and a local link remains challenging. The clustering technique is inspired by Shannon's entropy measurement , and it is denoted as follows:

$$\varphi(\mathrm{s_i}) = \textstyle\sum_{\mathrm{j}=1}^{\mathrm{n}}\left(-\mathrm{s_{ij}}\ln\mathrm{s_{ij}}\right) \tag{16}$$

When there are n elements in $\mathrm{s_i}$ and $\mathrm{s_{ij}} \geq 0$ under a probability constraint, $\varphi(\mathrm{s_i})$ reaches its minimum only if one of the elements is equal to one, and the rest are zeros. This represents the most informative distribution state of $\mathrm{s_i}$ under the probability model. Entropy maximization is then formulated as follows:

$$\begin{array}{ll} \min\limits_{S} & \sum_{i,j=1}^{n} \parallel x_i - x_j \parallel_2^2 s_{ij} + \gamma s_{ij} \ln s_{ij} \\ \text{s.t.} & \forall i, \quad s_i^T \mathbf{1} = 1, \quad 0 \leq s_i \leq 1, \quad s_{ii} = 0 \end{array} \tag{17}$$

Where $\gamma > 0$ is a regularizer for entropy. If the parameter γ is large enough, the problem can be written as:

$$\begin{array}{ll} \min\limits_{S} & \sum_{i,j=1}^{n} s_{ij} \ln s_{ij} \\ \text{s.t.} & \forall i, \quad s_i^T \mathbf{1} = 1, \quad 0 \leq s_i \leq 1, \quad s_{ii} = 0 \end{array} \tag{18}$$

Let $d_{ij}^{X} = \parallel x_i - x_j \parallel_2^2$, then the problem can be written as:

$$\begin{array}{ll} \min\limits_{S} & \sum_{i,j=1}^{n} d_{ij}^{x} s_{ij} + \gamma s_{ij} \ln s_{ij} \\ \text{s.t.} & \forall i, \quad s_i^T \mathbf{1} = 1, \quad 0 \leq s_i \leq 1, \quad s_{ii} = 0 \end{array} \tag{19}$$

4. General Objective

The UDADFSP method is constructed by integrating the aforementioned mathematical equations (3), (4), (5), (6), (7,8), (11), (12), and (13) as follows:

$$\max \frac{\mu\{\text{target aff. matrix}\}+\gamma\{\text{entropy reg}\}+\beta\{\text{Between Class Var}\}}{\{\text{Dist shift}\}+\lambda\{\text{Subspace shift}\}+\beta\{\text{intra-class Var}\}} \tag{20}$$

The trade-off parameters $\mu, \gamma, \beta, \lambda$ are utilized in the optimization process, where var represents variance and entropyreg denotes entropy regularization for target clustering. By solving the following optimization function, we aim to obtain two coupled projections C and D:

$$\max_{C,D} \frac{\mathrm{Tr}\left(\begin{bmatrix} C^T & D^T \end{bmatrix}\begin{bmatrix} \beta S_b & 0 \\ 0 & \mu S_t \end{bmatrix}\begin{bmatrix} C \\ D \end{bmatrix}\right)}{\mathrm{Tr}\left(\begin{bmatrix} C^T & D^T \end{bmatrix}\begin{bmatrix} M_s+\lambda I+\beta S_w & M_{st}-\lambda I+\gamma \mathrm{REnt} \\ M_{ts}-\lambda I+\gamma \mathrm{REnt} & M_t+(\lambda+\mu)I+\gamma \mathrm{REnt} \end{bmatrix}\begin{bmatrix} C \\ D \end{bmatrix}\right)} \tag{21}$$

where REnt represents regularize entropy. In order to improve (20), we modify $[C^T \quad D^T]$ as G^T. Once this is achieved, the goal function and related restrictions may be modifying as follows:

$$\min_{G} \frac{\mathrm{Tr}\left(G^T\begin{bmatrix} \beta S_b & 0 \\ 0 & \mu S_t \end{bmatrix} G\right)}{\mathrm{Tr}\left(G^T\begin{bmatrix} M_s+\lambda I+\beta S_w & M_{st}-\lambda I+\gamma \mathrm{REnt} \\ M_{ts}-\lambda I+\gamma \mathrm{REnt} & M_t+(\mu+\lambda)I+\gamma \mathrm{REnt} \end{bmatrix} G\right)} \tag{22}$$

It should be noted that G rescaling does not impact the objective function. Analytical solutions for the relevant eigenvectors can be found using extended eigenvalue decomposition. The subspaces C and D are simple to acquire once the transformation matrix G has been established. The UDADFSP pseudo code is summarized by Algorithm 1.

Algorithm 1.
Input: X_s, Y_s, X_t **Parameters:** $\mu = 1$, k, T, γ, $\lambda = 1$, β, σ **Output:** Y_t (Predicted labels for target domain)

Build S_b, S_t, S_w, M_{s1}, M_{t1}, M_{st1}, and M_{ts1} using Equations (2), (3), (4), (5), (6), (9), (10), (11), (12) Set up the initial pseudo-labels for the target domain, expected Y_t, employing a learner trained on data from the source domain Apply entropy regularization (Equation (20)) using parameter γ Repeat steps 3 and 4 until $i < T$

Solve generalized eigenvalue decomposition in Equations (22) and (23) Select k corresponding eigenvectors of leading eigenvalues as transformations w and obtain subspaces C and D To obtain the embeddings, map the original data to the appropriate subspaces: $F_s = C^T X_s$, $F_t = D^T X_t$ until convergence Finalize the adaptive classifier f on $\{F_s, F_t, Y_s\}$ Predict $Y_t \leftarrow f([F_s, F_t, Y_s])$

Kernel methods: PCA and LDA may be used effectively when our data can be linearly separated. However, data may not be separable in a linear fashion in real-world applications. As a result, we must first transform our input data to a higher dimensional space before linearly separating it. For this reason, we used kernel functions in this project, demonstrating the Primal approach, Linear kernel, and Radial Basis Functions in practice. Kernel functions convert input data into higher-dimensional spaces that are separated linearly.

$h(X) = W^T X + b$ where X is input data, W is transformation weight, and b is biased $h(X) = Wf(X) + b$. This is a Kernel function using a decision boundary surface function $f(X)$, which refers to the data's non-linear decision boundary surfaces. Kernel function $k(x, y)$ satisfies Mercer's condition by dot product or inner product as follows:

$$W^T X = \sum_{i=1}^{d} w_i x_i \tag{23}$$

Where d is the dimension of each sample. For a compact input space X and the set of all square integral functions then expanded as

$$k(x, x') = \sum_{i=1}^{n_{\mathcal{H}}} \phi_i(x)\phi_i(x') \tag{24}$$

Where $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS). Numerous kernel functions, such as polynomial kernels, radial basis procedures, etc., satisfy the Mercer condition. Linear kernel. In this work, we included linear and Radial basis functions as follows:

$$k(X, W) = W^T X. \tag{25}$$

Radial Basis Function (RBF) Radial Basis Function (RBF) is also known as the Gaussian Kernel function. RBF is widely used to find the similarity and dissimilarity between non-linear data samples, such as approximate warping of 2D facial expressions. The RBF kernel function can be defined as follows:

$$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{\sigma}\right) \tag{26}$$

Where x_i and x_j are two sample vectors, and σ is a parameter that determines the impact of each sample xi.

5. Experiments

In comparison to the most prominent and widely used do- main adaptation dataset benchmarks, our technique demonstrates better performance when compared to various cutting- edge approaches, based on the average results. SA(subspace alignment[29], SDA(subspace distribution alignment)[30], GFK(Geodesic flow kernel)[31], TCA(Transfer component Analysis)[9], JDA(Joint distribution alignment)[17], TJM(Transfer joint matching)[18], SCA(Scatter Component Analysis)[32], Optimal Transport(OTGL)[33],Kernel manifold alignment(KEMA)[34], JGSA(Joint geometrical and statistical alignment)[16], JPDA(Discriminative joint probability maximum mean discrepancy for do- main adaptation)[19] and BW-JGSA(Balanced weight- JGSA)[35]. We use parameters that are included in the expression as follows. In all evaluations, we fixed $\lambda = 1$ and $\mu = 1$, while treating β, T, k, and K as free parameters for the number of iterations (T-iteration), the dimensionality of subspaces (k-dimension), and the number of neighbors in clustering (K).

5.1 Real-World Datasets

Our strategy undergoes testing using three commonly employed datasets in optical recognition challenges: Object recognition handwritten digits (USPS, MNIST), (Office, Cal- tech), COIL20 dataset, and Multi-PIE facial images (PIE07, PIE05, PIE09, PIE29, PIE27).

Table 2. Detailed information about the datasets used in the experiments.

Dataset	Samples	Features	Classes	Data	Domains
Office	1410	800	10	A,W,D	Object
Caltech	1123	800	10	C	Object
USPS	1800	256	10	USPS	Digit
MNIST	2000	256	10	MNIST	Digit
COIL20	1440	1024	20	COIL1, COIL2	Object

A. Object Recognition:

We have used datasets for object recognition and datasets from Caltech [31]. It contains do- mains: Amazon, Webcam, and DSLR. Each has images from Amazon.com or an office environment with varying lighting and poses changes using a webcam or a DSLR camera.

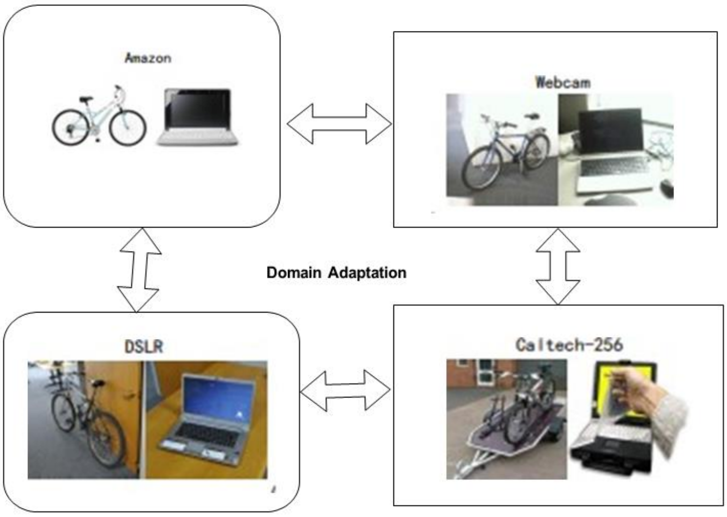


Fig. 3. This figure shows Office31 and Caltech256 data set sample images.

Table 3. PIE samples and COIL20

COIL1	COIL2	PIE05	PIE07	PIE09	PIE27	PIE29
720	720	3332	1629	1632	3329	1632
20	20	68	68	68	68	68
1024	same	same	same	same	same	Same

The Office-Caltech data set, which includes the Caltech-256 data set (which contains 256 classes of C) and the Office-10 data set is made up of ten standard classes of data sets (which include ten categories of A, D, and W). SURF BoW histogram features, vector quantized to 800 dimensions (unless indicated as 600), and Decaf6 features (which are activations of the 6th fully connected layer of convolutional neural networks (CNNs) that trained on Ima- genet). The classifier is built on the one-nearest neighbor (1-NN) principle [31]. We fix the unrestricted parameters,

$k = [20,100], T = 10, \beta = [0.0000000001,0.5], \mu = 1$ and $\alpha = 1$. Office Caltech dataset contains ten overlapping categories with the Office31 and Caltech-256. Surf Bow histogram features and vector quantized to 800 dimensions are also available for this dataset [36].

B. Multi-PIE dataset:

The multi-PIE(multi-pose, Illumination, expression) dataset contains 68 classes and 337 faces of image subjects taken under different Illumination, pose, and presentation. The pose range includes 15 discrete views, capturing a face profile-to-profile. Illumination changes using 19 flashlights located in the room [37].

C. Digit Recognition dataset:

We used USPS and MNIST datasets to assess our handwritten digit recognition task approach. MNIST has 6,000 training and 10,000 testing sets. The size of the input is 28x28x1. USPS contains 7,291 training and 2,007 testing sets of size 16X16X1. They have both ten classes and categories [0-9] digits. We follow [17, 18] to construct pairs of cross-domain datasets UPS-MINST and vice versa also proper. The sample of USPS images is 1,800, and for MNIST 2000, one is a source and target, respectively. All images are uniformly rescaled to 16x16 and are gray scales. We set T, k, and beta for arbitrary parameters according to Table 11.

D. COIL20 cross-domain dataset:

COIL20 is a dataset of 1140 photos and 20 categories [38]. The item is in the center of a steady turntable arrangement with a black background. The 72 photos obtained for each item are rotated 360 degrees (one of every five degrees). The input size is 32x32 pixels, and the picture size is normalized. COIL20 is subdivided into two parts. Each subset has 720 images: COIL1 and COIL2.COIL1 contains all images taken in the directions [0, 85] and [180,265], as well as COIL2 [90,175] and [270,355]. Experiments are conducted on the COIL1 and COIL2 source and target domains, and vice versa.

E. Sentiment Analysis:

We assessed the effectiveness of our standard Amazon review dataset for sentiment analysis [39], in which bag of words features' dimensionality was reduced to 400 top words with no missing data. The dataset is divided into four categories: books, DVDs, electronics, and kitchen equipment. Each section has 1000 positives and negatives.

Table 4. Accuracy (%) using the multi-pie dataset

Features	JDA	TCA	JPDA	BDA	Proposed
pie05-09	54.23	41.79	**66.67**	52.82	61.46
pie05-07	58.81	40.76	59.36	58.2	**68.39**
pie05-27	84.5	59.63	83.99	83.03	**89.25**
pie07-05	57.62	41.81	63	57.35	**76.89**
pie05-29	49.75	29.35	49.51	49.14	**58.70**
pie07-09	62.93	51.47	60.85	62.75	**75.00**
pie07-29	39.89	33.7	47.67	39.71	**63.48**
pie07-27	75.82	64.73	77.05	75.76	**86.33**
pie09-05	50.96	34.69	59.78	51.35	**70.41**
pie09-27	68.46	56.23	74.47	67.86	**83.63**
pie09-07	57.95	47.7	63.35	56.41	**72.50**
pie09-29	39.95	33.15	52.7	42.4	**69.49**
pie27-07	82.63	67.83	83.24	83.06	**88.83**
pie27-05	80.58	55.64	84.87	80.52	**87.82**
pie27-09	87.25	75.86	**87.44**	87.25	84.50
pie27-29	54.66	40.26	65.38	54.53	**74.63**
pie29-07	42.05	29.9	51.32	43.22	**69.98**
pie29-09	53.31	29.9	55.76	47.92	**73.35**
pie29-05	46.46	26.98	53.63	47.99	**61.34**
pie29-27	57.01	33.64	58.49	57.1	**78.04**
Average	60.24	44.75	65.52	59.91	**74.70**

Results not found in the paper are denoted by (-), and our reported results for UDADFSP with primal, linear, and radial basis functions (RBF) are indicated as UDADFSP_p, UDADFSP_l, and UDADFSP_rb f , respectively.

5.2. Result and Discussion

We employed the most widely used cross-domain visual recognition datasets (Office31 (Amazon, Webcam, DSL), and Caltech), digital recognition datasets (USPS+MNIST), and Multi-PIE face picture datasets with varying Pos, Illumination, and Expressions. Also, we have compared our method to the COIL20 dataset, which has other distributions and views of objects, and Amazon reviews sentiment analysis datasets. Our proposed method outperforms the state-of-the-art, achieving high accuracy in most domains. Also, in implementation, we included non-linearity using kernel function (Linear and RBF) kernels and entropy regularization.

Our method UDADFSP result in Table 5 uses SURF features with 800 dimensions, and we compare it with the more recent state of arts regarding domain adaptation for visual tasks. Table 5 result shows that our result average at original space (Primal) and kernels (Linear and RBF) outperform the state of arts. Our approaches are able to achieve 51.37 average accuracy more than the state of arts. Also, if we see Table 6's result, the features generated using CNN and taken at 6-layers known as DECAF6, which has 4096 dimensions. The proposed method UDADFSP average result yields more accuracy than the state of art(SOTA) when we include both linearity and non-linearity. Our methods achieve better 90.49 average accuracy and beat the state of arts. In Table 7, we adapt our approach with the Multi-PIE dataset and compare it with the current state of the arts. Our method performs well. We observed that when state-of-the-art accuracy 65.52 and our proposed model gives 74.70 average accuracy. Table 8 shows USPS–MNIST or MNIST-USPS. Our approach shows more than other SOT (State of skills). Our proposed method gives 75.94 average accuracy and is superior to the state of the arts. We have compared our process in Table 9 using the COIL20 dataset with the SOATs, and our method shows that it outperforms well. Our proposed model achieved a better average accuracy of the 95.90. Table 10 shows our ways against state-of-the-art performance for the sentiment analysis in natural language processing [40].

A. t-SNE Feature Visualization

To ensure the robustness and clarity of our experiments be- fore and after domain adaptation, we have employed t-SNE for visualizations. Fig. 4 and 5 indicate that we utilized the MNIST dataset as the input space and the USPS dataset as the target task. We have plotted clearly what our datasets look like before and after adaptation. We have proven our methods work properly when compared with state-of-arts. The following plot, Fig. 6(a) and 6(b) illustrate PIE datasets. We took PIE05 as the Source and PIE27 as the target domain. As we observe from visualization before and after domain adaptation, our techniques work very well. After transformation, the source and target domains close together. For Fig. 7(a) and 7(b), we apply the Office-caltech10 Surf features. We have used Amazon as a source and Webcam as the target domain. We plot visualization before and after domain adaptations and show that our methods work correctly. Fig. 8(a-d) shows t-SNE visualization of the Office-caltech10 Surf features. Fig.8 (a) Input Surf features before applying domain adaptation techniques, whereas the source and target domain features are distributed along spaces and not appropriately classified. Fig. 8(b–d) illustrates the data features of the two classes (Source and target domain) that match using the primal, linear, and RBF, respectively. Following classification, the data points of each distinct class are collected and aggregated in a specific location. For example, they are gathered in a space at one location too close together after domain adaptation, and ten different color points denoting ten distinct classes are clustered at random locations, indicating they are classified. However, as seen in the RBF and primal methods, the points are not clustered as effectively as in the linear way. This is because the classification accuracy of UDADFSP is higher than that of RBF and primal methods.

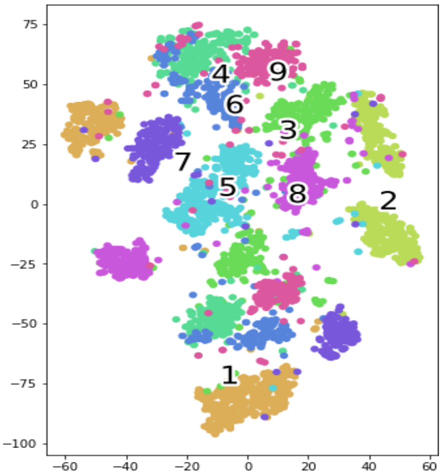


Fig. 4. MNIST-USPS before domain adaptation

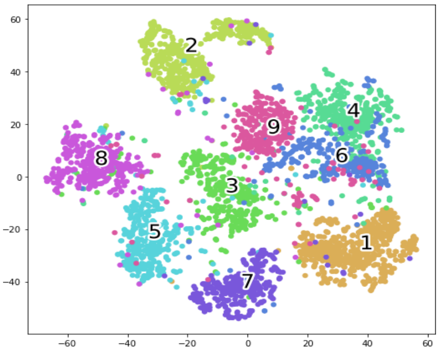
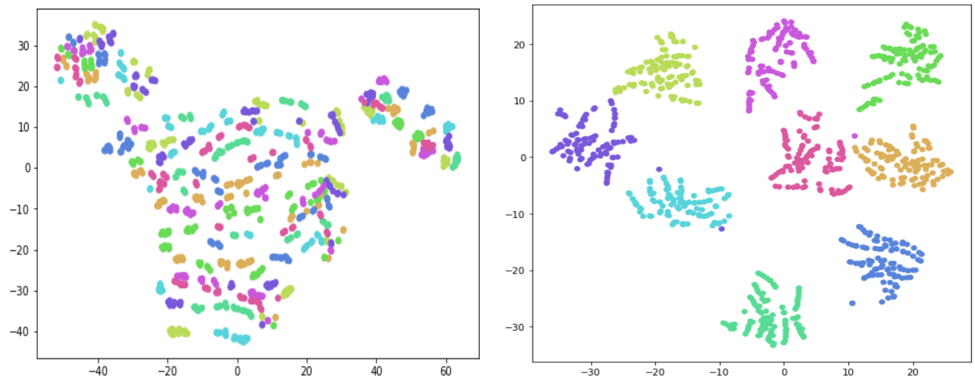
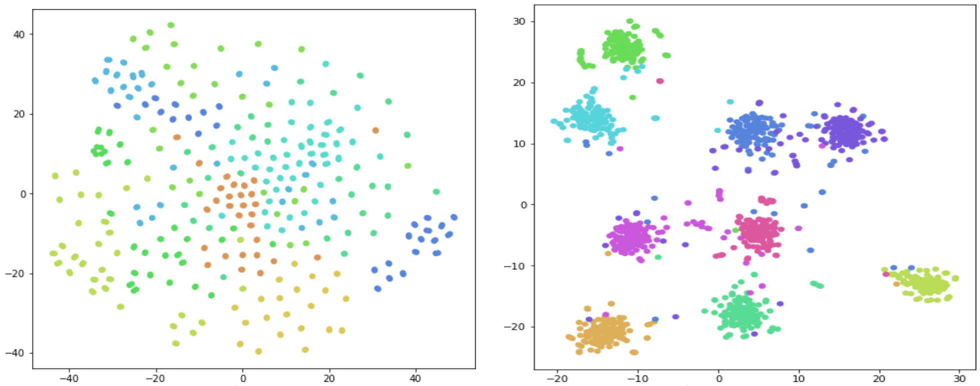


Fig. 5. MNIST-USPS after domain adaptation



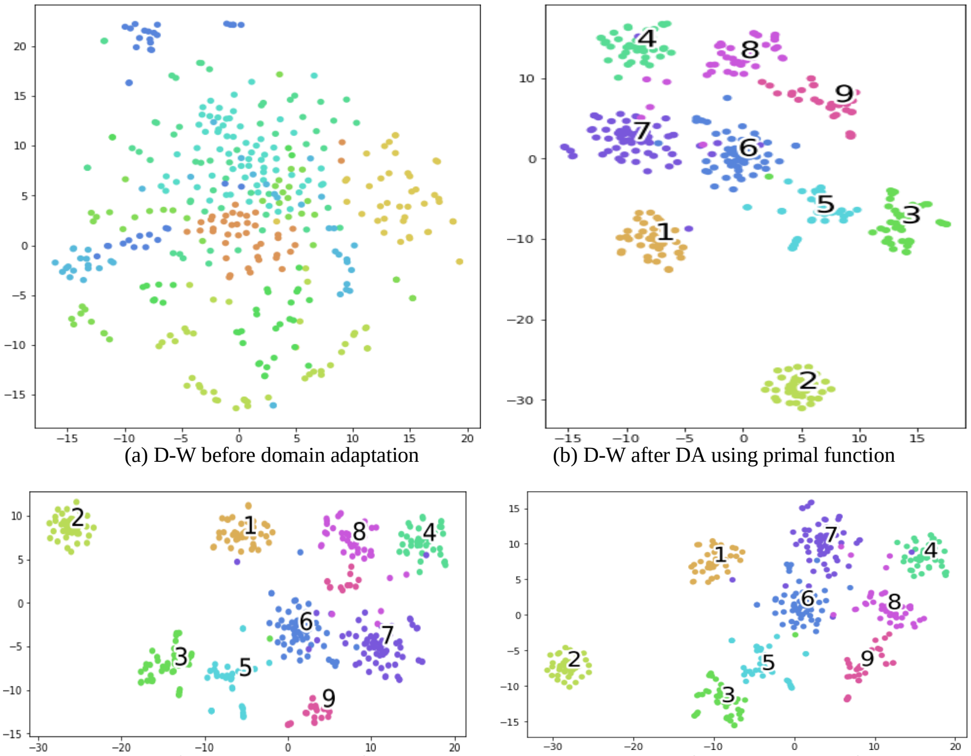
(a) PIE05-PIE27 before domain adaptation (b) PIE05-PIE27 after domain adaptation.

Fig. 6. Multi-PIE dataset (PIE05-PIE27) tasks t-SNE visualization before domain and after domain adaptation.



(a) A-W before domain adaptation (b) A-W after domain adaptation

Fig. 7. Office-caltech10 Surf features (A-W) t-SNE visualization before and after domain adaptation



(a) D-W before domain adaptation (b) D-W after DA using primal function

(c) D-W after DA using Linear function (d) D-W after using the RBF function

Fig. 8. t-SNE visualization of the Office-caltech10 Surf feature (D-W) tasks by applying various kernel functions.

Table 5. Accuracy on office-caltech10 using surf features

Features	Source	SA	SDA	GFK	TCA	JDA	TJM	SCA	JGSA_p	JGSA_l	JGSA_rbf	DJPA	BW_jgsa	UDADFSP_p	UDADFSP_l	UDADFSP_rbf
C→A	36.01	9.27	49.69	46.03	45.82	45.62	46.76	45.62	51.46	52.3	53.13	47.6	51.88	53.97	52.61	**55.95**
C→W	29.15	40	38.98	36.95	31.19	41.69	38.98	40	45.42	45.76	48.47	45.76	46.1	49.83	**50.17**	49.49
C→D	38.22	39.49	40.13	40.76	34.39	45.22	44.59	47.13	45.86	48.41	48.41	46.5	**49.68**	45.86	44.59	47.77
A→C	34.19	39.98	39.54	40.69	42.39	39.36	39.45	39.72	41.5	38.11	**41.5**	40.78	42.03	40.69	40.78	40.87
A→W	31.19	33.22	30.85	36.95	36.27	37.97	42.03	34.92	45.76	49.49	45.08	40.68	45.42	45.76	**51.86**	44.07
A→D	35.67	33.76	33.76	40.13	33.76	39.49	45.22	39.49	47.13	45.86	45.22	36.94	45.85	47.77	**48.41**	45.86
W→C	28.76	35.17	34.73	24.76	29.39	31.17	30.19	31.08	33.21	32.68	33.57	34.55	**34.55**	32.59	31.34	31.70
W→A	31.63	39.25	39.25	27.56	28.91	32.78	29.96	29.96	39.87	41.02	40.81	33.82	40.81	41.34	41.34	**41.86**
W→D	84.71	75.16	75.8	85.35	89.17	89.17	89.17	87.26	90.45	90.45	88.54	88.54	91.71	89.81	89.81	**91.08**
D→C	29.56	34.55	35.89	29.3	30.72	31.52	31.43	30.72	29.92	30.19	30.28	**34.73**	34.11	33.21	34.02	31.97
D→A	28.29	39.87	38.73	28.71	31	33.09	32.78	31.63	38	36.01	38.73	34.66	33.29	40.81	**41.02**	40.40
D→W	83.73	76.95	76.95	80.34	86.1	89.49	85.42	84.41	91.86	91.86	93.22	91.19	**91.84**	91.53	90.51	91.53
Average	40.93	44.72	44.52	43.13	43.26	46.38	46.33	45.16	50.04	50.18	50.58	47.98	50.61	**51.10**	**51.37**	**51.05**

Table 6. Office+caltech accuracy (%) with decaf6 features

Features	JDDA	OTGL	JGSA(P)	JGSA(l)	JGSA(rbf)	UDADFSP(p)	UDADFSP(l)	UDADFSP(rbf)
C→A	90.19	92.15	91.44	91.75	91.13	**93**	92.80	92.80
C→W	85.42	84.17	86.78	85.08	83.39	85.76	**87.80**	85.76
C→D	85.99	87.25	93.63	92.36	92.36	91.72	94.27	**94.27**
A→C	81.92	85.51	84.86	85.04	**84.86**	86.46	86.20	85.31
A→W	80.68	83.05	81.02	**84.75**	80	80.68	80.00	79.66
A→D	81.53	85	**88.54**	85.35	84.71	84.71	87.90	86.62
W→C	81.21	81.45	84.95	84.68	84.51	**86.46**	85.66	85.31
W→A	90.71	90.62	90.71	91.44	91.34	**91.86**	91.65	91.23
W→D	100	96.25	100	100	100	100	100	100
D→C	80.32	84.11	86.2	85.75	84.77	**88.25**	87.36	87.44
D→A	91.96	92.31	91.96	92.28	91.96	92.80	92.90	**92.90**
D→W	99.32	96.29	99.66	98.64	98.64	**100.00**	99.32	99.32
Average	87.44	88.18	89.98	89.76	88.97	**90.13**	**90.49**	**90.05**

Table 7. Digit dataset accuracy (%) across domains

Features	Raw	GFK	SDA	SA	TCA	JDA	TJM	SCA	JGSA(p)	UDADFSP(p)
usps-mnist	44.7	46.45	35.7	48.8	51.2	59.65	52.25	48	68.15	**68.15**
mnist-usps	65.94	61.22	65	67.78	56.33	67.28	63.28	65.11	80.44	**84**
Average	55.32	56.84	50.35	58.29	53.77	63.47	57.77	56.56	74.3	**75.94**

Table 8. Cross-domain using coil20 dataset accuracy (%)

Data	PCA	GFK	NN	TSL	LISL	LRSR	JDA	TCA	UDADFSP(l)	UDADFSP(p)	UDADFSP(rbf)
C1-C2	84.72	72.5	83.61	88.06	75.69	88.61	89.31	88.47	**96.81**	92.50	91.39
C2-C1	84.03	74.17	82.78	87.92	72.22	89.17	88.47	85.83	**95.00**	90.83	90.83
Average	84.38	73.34	83.2	87.99	73.96	88.89	88.89	87.15	**95.90**	**91.67**	**91.11**

Table 9. Accuracy (%) of the amazon review data set on four typical domain shifts with various features.

Features	GFK	TCA	BDA	UDADFSP(l)	UDADFSP(p)	UDADFSP(rbf)
B-K	65.1	-	63.4	67.58	67.48	70.64
B-E	74.7	68.4	-	67.02	66.67	73.02
B-D	62.4	66.3	64.3	65.78	66.88	**70.64**
K-B	64	65.5	62.5	64.10	66.25	**68.25**
K-E	-	-	-	67.58	**71.22**	70.64
K-D	60.4	69	-	67.02	67.48	**73.02**
E-B	59.5	63	58.7	65.78	66.67	**70.64**
E-K	68.7	**73.3**	74.5	64.10	66.88	68.25
E-D	62.7	-	62.1	67.58	66.25	**70.64**
D-B	62.2	66.5	62.7	67.02	71.22	**73.02**
D-K	-	-	-	65.78	67.48	**70.64**
D-E	66.3	64	67	64.10	66.67	**68.25**
Average	67	64.6	64.4	**67.58**	**66.88**	**70.64**

B. Parameter Sensitivity

We investigate parameter sensitivity and ablation tests in this work to validate our technique UDADFSP on diverse datasets and uncover parameters that produce relevant results. We set $\lambda = 1$ and $\mu = 1$. Three parameters are used to evaluate all tasks and others (K, β, and T). We take range k [30, 50], and T [10,20] for all evaluation tasks, and β are trade-off parameters for within class, between class variance, and discriminative loss. We analyzed and did ablation studies range of β between [0.000000001 to 0.5] can be chosen to achieve acceptable results. As a result, parameter sensitivity and selection significantly impact the classification result. Our paper parameter details are as follows in Table 10, where O+C and DR represent office Caltech and Digit recognition, respectively.

Table 10. Parameters sensitivity

Parameters	O+C(surf)	O+C(decaf6)	PIE	COIL20	DR	Amazon review
k	30	30	50	30	30	30
T	10	20	10	10	10	10
α	1	1	1	1	1	1
μ	1	1	1	1	1	1
β	0.5	0.5	0.01	0.01	0.01	0.7
γ	2	2	2	2	2	2
σ	1	1	1	1	1	1
K1	30	30	60	40	40	40

C. Ablation Study

In this section, we conducted ablation studies to evaluate the impact of different parameters. The results revealed that k, K1, and β are highly sensitive across all datasets, emphasizing the importance of precise tuning. Here, k refers to the subspace dimension, K1 represents the number of neighbors in clustering, and β controls the trade-off between within and between-class variance of the source domain. A too- small value for β disregards the classification of the source domain, while an excessively large β can lead to overfitting the classifier to the source domain. To establish a baseline performance, we explored k values within the range of [20, 100]. The remaining parameters had minimal influence on the model's performance and were set according to the specifications outlined in Table 11. The subsequent tables display the ablation studies' results on various datasets.

Table 11. Parameter sensitivity on pie dataset

Data Set	*k*	*K*1	*β*	Average Accuracy
Multi-PIE	30	60	0.01	73.43
Multi-PIE	50	60	0.0001	74.70
Multi-PIE	60	100	0.01	76.34

6. Conclusion

This study presents UDADFSP (Unified Domain Adaptation with Discriminative Feature and Similarity Preservation), a novel and effective approach to tackle domain shifts in unsupervised domain adaptation. The primary focus of UDADFSP lies in capitalizing on discriminative features and domain similarity to counter domain shift effectively. We conducted comprehensive experiments and analyses to verify the effectiveness of our proposed method. The evaluations were performed on five diverse domains, which include the office+caltech dataset using Decaf6 and Surf features, the COIL20 dataset, digit recognition with USPS+MNIST, the Multi-PIE facial dataset, and Amazon review sentiment analysis. A comparative assessment is performed against various existing domain adaptation methods. Additionally, to accommodate non-linear data, we incorporate the kernel method, and for handling unlabeled target domains, we introduce entropy regularization. We further conduct comprehensive ablation studies to gain insights into the specific contributions of our methods, leading to significant performance enhancements. Future work will involve applying our pro- posed techniques to other types of data, including time series and medical images. Additional extensions could handle more complex scenarios, such as multi-source domain adaptation, where the source domains have different distributions and single target domains.

Disclosure Statement

The authors state that they have no known financial or personal conflicts of interest that may have influenced the study's findings.

References

[1] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10):1345–1359, 2009.
[2] Hal Daum éIII. Frustratingly easy domain adaptation. arXiv preprint arXiv:0907.1815, 2009.
[3] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. Neurocomputing, 312:135–153, 2018.
[4] Gabriela Csurka. A comprehensive survey on domain adaptation for visual applications. Domain adaptation in computer vision applications, pages 1–35, 2017.

[5] Arash Saboori and Hassan Ghassemian. Adversarial discriminative active deep learning for domain adaptation in hyperspectral images classification. Interna- tional Journal of Remote Sensing, 42(10):3981–4003, 2021.

[6] A. Gretton K. M. Borgwardt J. Huang, A. J. Smola and B. Scholkopf. Correcting sample selection bias by un- labeled data. 2006.

[7] J. Wang J. Sun Y. Guo M. Long, G. Ding and P. S. Yu. Transfer sparse coding for robust image representation. In CV-PR, 2013.

[8] Obsa Gilo, Jimson Mathew, and Samrat Mondal. Integration of discriminate features and similarity preserving for unsupervised domain adaptation. In 2022 IEEE 19th India Council International Conference (IN- DICON), pages 1–6. IEEE, 2022.

[9] J. T. Kwok S. J. Pan, I. W. Tsang and Q. Yang. Domain adaptation via transfer component analysis. IEEE transactions on neural networks, (22(2):):199–210, 2011.

[10] Domain adaptation problems: A dasvm classification technique and a circular validation strategy. 2009.

[11] F. De la Torre W.-S. Chu and J. F. Cohn. Selective transfer machine for personalized facial action unit dete tion. In CVPR, 2013.

[12] Hoffman J. Saenko K. Tzeng, E. and T. Darrell. Adversarial discriminative domain adaptation. In IEEE CV PR, 2017.

[13] Jianfei Yang, Han Zou, Yuxun Zhou, and Lihua Xie. Robust adversarial discriminative domain adaptation for real-world cross-domain visual recognition. Neurocomputing, 433:28–36, 2021.

[14] Jingyu Wang, Zhenyu Ma, Feiping Nie, and Xuelong Li. Entropy regularization for unsupervised clustering with adaptive neighbors. Pattern Recognition, page 108517, 2022.

[15] Rui Zhang, Xuelong Li, Hongyuan Zhang, and Feiping Nie. Deep fuzzy k-means with adaptive loss and entropy regularization. IEEE Transactions on Fuzzy Sys tems, 28(11):2814–2824, 2019.

[16] Li W. Zhang, J. and P. Ogunbona. Joint geometrical and statistical alignment for visual domain adaptation. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.

[17] Wang J. Ding G. Sun J. Long, M. and P. S. Yu. Transfer feature learning with joint distribution adaptation. Proceedings of the IEEE international conference on com- puter vision recognition, 2013.

[18] Wang J. Ding G. Sun J. Long, M. and P. S. Yu. Trans- fer joint matching for unsupervised domain adaptation. In Proceedings of the IEEE international conference on computer vision recognition, 2014.

[19] Wu D. A. (2020 July). Zhang, W. Discriminative joint probability maximum mean discrepancy (djp-mmd) for domain adaptation. In 2020 International Joint Conference on Neural Networks (IJCNN), pages 1–8.

[20] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 30, 2016.

[21] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In 9ECCV 2016 Workshops, 2016.

[22] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In Domain Adaptation in Computer Vision Applications, pages 153–171. Springer, 2017.

[23] Przemysław Spurek, Krzysztof Byrski, and Jacek Tabor. Online updating of active function cross- entropy clustering. Pattern Analysis and Applications, 22(4):1409–1425, 2019.

[24] Wenming Cao, Zhongfan Zhang, Cheng Liu, Rui Li, Qianfen Jiao, Zhiwen Yu, and Hau-San Wong. Unsupervised discriminative feature learning via finding a clustering-friendly embedding space. Pattern Recognition, 129:108768, 2022.

[25] Aleix M Martinez and Avinash C Kak. Pca versus lda. IEEE transactions on pattern analysis and machine intelligence, 23(2):228–233, 2001.

[26] Uri Lipowezky. Selection of the optimal prototype sub- set for 1-nn classification. Pattern Recognition Letters, 19(10):907–918, 1998.

[27] Richard O Duda, Peter E Hart, et al. Pattern classifi- cation and scene analysis, volume 3. Wiley New York, 1973.

[28] Arash Abdi, Mohammad Rahmati, and Mohammad M Ebadzadeh. Entropy based dictionary learning for im- age classification. Pattern Recognition, 110:107634, 2021.

[29] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In Proceedings of the IEEE international conference on computer vision, pages 2960–2967, 2013.

[30] Baochen Sun and Kate Saenko. Subspace distribution alignment for unsupervised domain adaptation. In BMVC, volume 4, pages 24–1, 2015.

[31] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In 2012 IEEE conference on computer vision and pattern recognition, pages 2066–2073. IEEE, 2012.

[32] Muhammad Ghifary, David Balduzzi, W Bastiaan Kleijn, and Mengjie Zhang. Scatter component anal- ysis: A unified framework for domain adaptation and domain generalization. IEEE transactions on pattern analysis and machine intelligence, 39(7):1414–1430, 2016.

[33] Nicolas Courty, Rémi Flamary, and Devis Tuia. Do- main adaptation with regularized optimal transport. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 274–289. Springer, 2014.

[34] Devis Tuia and Gustau Camps-Valls. Kernel manifold alignment for domain adaptation. PloS one, 11(2):e0148655, 2016.

[35] MS Rizal Samsudin, Syed AR Abu-Bakar, and Musa M Mokji. Balanced weight joint geometrical and statistical alignment for unsupervised domain adaptation. Journal of Advances in Information Technology Vol, 13(1), 2022.

[36] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new do- mains. In European conference on computer vision, pages 213–226. Springer, 2010.

[37] Iain Matthews. Fast and accurate active appearance models. In Proceedings of the HCSNet workshop on Use of vision in human-computer interaction-Volume 56, pages 3–3, 2006.

[38] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al.Columbia object image library (coil-100). 1996.

[39] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In Pro- ceedings of the 45th annual meeting of the association of computational linguistics, pages 440–447, 2007.

[40] Zixuan Cao, Yongmei Zhou, Aimin Yang, and Sancheng Peng. Deep transfer learning mechanism for fine-grained cross-domain sentiment classification. Connection Science, 33(4):911–928, 2021.

[41] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(11), 2008.

Authors' Profiles

Obsa Gilo received the BSc in 2014 and MSc. degrees in computer science from the Wallaga University, Ethiopia, in 2018. Currently, pursuing a Ph.D. degree in computer science and engineering at IIT Patna, India. He worked in a faculty position as a lecturer for more than four years at Wallaga University. He is exploring current research areas, computer vision, machine learning, Pattern recognition, transfer learning, and domain adaptation. He published and contributed to reputable journals and conferences.

Jimson Mathew received his master's degree in computer engineering from Nanyang Technological University (NTU), Singapore, and his Ph.D. degree in computer engineering from the University of Bristol, Bristol, U.K. Throughout his career, and he has held positions at various prestigious institutions including the Centre for Wireless Communications, National University of Singapore; Bell Laboratories Research, Lucent Technologies North Ryde, Australia; Royal Institute of Technology KTH, Stockholm, Sweden; and the Department of Computer Science, University of Bristol, U.K. He worked as Head of Computer Science and Engineering at IIT, Patna. Currently, he serves as a Professor in the Computer Science and Engineering Department at IIT Patna, India. He is a member of IET (Institution of Engineering and Technology). He has made significant contributions to the field of computer engineering and has a strong academic portfolio. He holds multiple patents, has coauthored three books, and has published over 100 papers in renowned international journals and conferences. His re- search interests span various areas, including fault-tolerant computing, hardware security, large-scale integration design, and design automation. His expertise and contributions have had a notable impact on computer engineering.

Samrat Mondal received a Ph.D. degree in computer science and engineering from the School of Information Technology, Indian Institute of Technology Kharagpur, Kharagpur, India, in 2010. Since December 2010, he has been currently a Faculty with the Department of Computer Science and Engineering, Indian Institute of Technology Patna, Patna, India. He was a Visiting Faculty with the University of Denver, Colorado, CO, USA, for 11 months. Earlier, he was also a Visiting Research Scholar with National Semiconductor Corporation, Santa Clara, California, CA, and USA. He has authored or co-authored several research papers in reputed international journals and conferences. He received Research Grants from the Science and Engineering Research Board, Government of India on multiple occasions. His research interests include security and privacy, database and data mining applications, and smart energy management-related applications.

How to cite this paper: Obsa Gilo, Jimson Mathew, Samrat Mondal, "Unified Domain Adaptation with Discriminative Features and Similarity Preservation", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.16, No.2, pp. 39-53, 2024. DOI:10.5815/ijieeb.2024.02.04