International Journal of Multimedia Information Retrieval (2019) 8:241–252
https://doi.org/10.1007/s13735-019-00180-z

REGULAR PAPER

Semi-supervised domain adaptation for pedestrian detection in video surveillance based on maximum independence assumption

Ghazaleh Shojaei[1] · Farbod Razzazi[1]

Received: 1 July 2019 / Revised: 27 September 2019 / Accepted: 4 October 2019 / Published online: 2 November 2019

Abstract

In this paper, two domain adaptation approaches are utilized in a pedestrian detection application that is one of the most interesting and widely used fields in machine vision. In cases where the distributions of training and test data are different, the performance of pedestrian detection algorithms drops significantly. In this paper, employing two methods, namely transfer component analysis (TCA) and maximum independence domain adaptation (MIDA), the source and target domain data are represented in a new space where the distributions of two domains are more similar to each other, while the local geometry of data is preserved. Thereby, the classifier trained in the original space can be applied to the target data after the transformation. The experimental results of the proposed approach obtained on INRIA train dataset and CUHK test dataset show 82% about relative reduction in the classification error in the case of using TCA and about 84% in the case of using MIDA, compared to the baseline method where no domain adaptation method has been applied.

Keywords Domain adaptation · Pedestrian detection · Transfer learning · Semi-supervised learning · Transfer component analysis · Maximum independence domain adaptation

1 Introduction

Detecting an object in an image or a video sequence is one of the most important tasks in computer vision field. Among these, pedestrian detection is important for developing an intelligent transport system as well as automated video surveillance on traffic scenes [1–3]. Most of these researches have provided object detectors based on the appearance. These studies have increased the detection efficiency and also reduced the destructive effect of background subtraction such as merging and splitting blobs, detecting moving objects in the background, and detecting moving shadows. On the other hand, other studies have focused on the selection of appropriate features that reduce false positive rate and increase the detection rate [4, 5]. However, these methods have led to higher computational costs due to the use of multi-scale detection methods. Therefore, some researches have focused on reducing the time needed to compute features without adding complexity or particular hardware requirements to allow fast multi-scale detection [6, 7].

Besides, one of the key problems in appearance features-based detectors is the dataset issue, where thousands of labeled samples are required for training stage. The preferred dataset contains a wide range of variety, including different scales, viewpoints, lighting conditions, and resolution. In addition, training a specific pedestrian detector to be used in different situations is difficult. This is due to the high diversity in traffic scenes, including the presence of objects from different categories (e.g., various vehicles as well as different animals and plants), different road conditions in terms of infrastructure, the impact of climate on video quality and video recording hours (e.g., day or night conditions, during peak hours of traffic or during off hours).

The variety of positive and negative samples recorded in video surveillance in a specific scene is very limited in comparison with generic scenes. However, the accuracy of a generic detector shows a notable reduction when it is used in a specific scene. The main reason is the difference between the statistical distribution of target domain samples (specific scenes) and source domain samples (generic scenes). Indeed, it is preferred that a detector, which is trained on a generic scene, should be able to detect the intended target

✉ Farbod Razzazi
razzazi@srbiau.ac.ir

[1] Department of Electrical and Computer Engineering Science and Research Branch, Islamic Azad University, Tehran, Iran

(pedestrian) in a specific scene without a performance reduction. Hence, the use of domain adaptation methods has to be considered.

The main drawback of designing a scene-specific detector is the need of labeled samples from the target scene, which requires human efforts and can be a very difficult and time-consuming task due to the variety of available classes in the target domain. Therefore, a practical and reasonable solution is to label a few target domain samples that are combined with the source dataset's samples for training the detector in a semi-supervised manner.

1.1 Related work

In spite of many studies carried out on the generic detectors, research on the scene-specific detectors has not been presented extensively. Typically, researchers have designed a sample labeler that automatically selects positive and negative samples from the target scene to re-train the generic detector. In order to improve the performance and the effectiveness of the work, training samples which are selected by automatic labeler have to be reliable and informative for the generic detector.

The conventional simple method of self-learning has been used in [8]. In this study, the classified samples with high reliability have been used to re-train the classifier. At first, the generic classifier is trained with a small set of existing labeled data, and the new classifier is applied to categorize unlabeled data. Then, a subset of the most reliable categorized data is selected and combined with labeled data for re-training the classifier. To improve self-training approach, the target samples can be selected through a hard-threshold coefficient, obtained by appearance-based detectors or context-cues [9, 10]. The hard-threshold (or aggressive threshold) method is an unreliable method that may eliminate useful information and cause the detector to fail. On the other hand, a conservative (soft) threshold leads to inadequate training and convergence after a lot of repetitions.

In contrast, transfer learning provides a main solution to the problem of domain adaptation. This method has been used in object recognition, scene classification, action recognition, image retrieval [11, 12] and visual concept classification [13]. Pong has proposed a transfer learning approach that adapts the weights of the classifier trained on source samples to target samples [14].

Roth et al. have trained a separate detector for each local area [15, 16]. Ali et al. have proposed a flow boost method for training a scene-specific detector from a training video that has been labeled with sparse tracking approaches [17]. In this case, the possibility of labeling is very limited. Jane and Miller have adapted a general detector to the new test domain using the label propagation methods without the need of detector re-training [18].

Fortunately, the position of most surveillance cameras is fixed. When the scene is constant, the changes in the positive and negative samples are significantly reduced, and as long as only one camera is used, the limiting variations only include the view point, resolution, ambient light conditions, and backgrounds. Therefore, it is easier to train a pedestrian detector on specific target samples accurately. The direct approach is training of the detector on manually labeled target samples. However, repeating the manual labeling for each viewpoint of a camera is a very costly task. An appropriate method is to apply a generic detector automatically to the target scene, so that a number of video frames from the target scene are labeled with the least effort to be used as a training dataset [19–21]. The problems of this method, such as too many repetitions till convergence and drifting risk during training stage, have been solved in Wang et al.'s research [22].

Wang et al. [22] have solved scene-specific pedestrian detection problem by providing a transfer learning framework and assessing a set of context-cues for selecting scene-specific training samples. The work begins with a generic detector that applies to unlabeled samples in videos from the target scene. Positive and negative samples of the target scene are automatically selected based on detection results and context-cues. As labels of the selected samples are predicted according to the detection scores and context-cues, it is probable that they would be false. Therefore, some confidence scores are calculated and selected samples and their confidence scores are employed to re-train the scene-specific detector using the transfer learning method. The updated scene-specific detector is applied to target samples to provide more samples for the next training step. This procedure is repeated until convergence.

Mao has suggested a new method that automatically trains a scene-specific detector based on tracklets (i.e., a chain of traceable samples) [23]. In this approach, a pedestrian detector is first applied to a specific scene, which of course involves a large number of false positives and misdetections. In the next step, a multi-pedestrian detector is considered as the corresponding problem and the detected samples are connected in a single tracklet. In the third step, the tracklet characteristics are classified into positive, negative, and unreliable labels. Again, the unreliable tracklets are labeled in comparison with positive and negative samples. With the use of tracklets, it is possible to extract more reliable features. Also, unreliable informative samples that are close to the classification boundaries are thoroughly labeled through propagation in separate tracklets and between different types of tracklets. Labeled samples are combined with the generic dataset to train scene-specific detectors. As indicated in [23], this approach outperforms the old scene-specific detectors and does not require manual labeling.

Maamatou has presented a new transfer learning framework based on the continuous Monte Carlo filter, which adapts the generic classifier to the specific scene classifier [24]. The proposed algorithm iteratively approximates the distribution of target samples as a dataset (from both target and source domains) to train the scene-specific classifier. The output classifier has then been used to detect pedestrians in traffic scenes. During the numerous experiments conducted on CHUK and MIT datasets, it has been shown that the performance of this scene-specific classifier is better than the generic ones.

One of the successful domain adaptation approaches extends the invariable properties of the source domain to the target domain. By doing so, inter-domain differences are reduced, while important features and information are preserved. One of these approaches is transfer component analysis (TCA) which attempts to learn the transfer components over the domains using the maximum mean difference in the Hilbert kernel space [25]. Also, there is possibility to extend it to a semi-supervised scenario. In this case, it is possible to use the labels' information as well as preserving main features of data.

Shi et al. [26] have measured domain differences using mutual information between all data and their binary labels. This method is considered as an introduction to domain adaptation methods such as MIDA. Also, they have minimized negative mutual information between target samples and their cluster labels to minimize classification error. Shao et al. [27] have presented a low-rank transfer subscriber learning algorithm (LTSL) that is a reconstructed data transfer method. In this method, each target sample is represented as a local combination of source samples in the new subspace. Information about the labels and the geometry of samples could be re-trained using generalization of various subspace learning methods to the LTSL.

1.2 Contribution of the paper

In this paper, we used two domain adaptation frameworks named as "transfer component analysis" (TCA) to learn a transformation from the source to target dataset to map target features into the source domain, and "maximum independence domain adaptation" (MIDA) to make a subspace in which domain features have maximum independence. This reduces the distributional difference between domains. These two methods were developed for simultaneous transformation-based domain adaptation and classification. TCA and MIDA were shown to be a good choice for pedestrian detection application. The tests were conducted to classify a video dataset that has a different distribution as that of the training data, in a semi-supervised scenario. Therefore, the main contributions of our paper include:

- To deal with significant differences between feature distributions of source and target domains, we used TCA procedure to select proper samples of the source domain with the best match to the distribution of the target domain. These samples are used to extract the adaptation transformation.
- To deal with the TCA problem that all samples are transferred to a common subspace and samples with the same appearance but different contents cannot be recognized, we used MIDA procedure. With the use of MIDA, firstly we defined the domain features for each sample to describe the corresponding background. Then, a final feature space was defined in which the samples have maximum independence from their domain features.
- Comprehensive experiments on INRIA and CUHK datasets demonstrate the effectiveness of these algorithms in real-world pedestrian detection applications, in the semi-supervised domain adaptation cases.

The rest of the paper is organized as follows: in Sect. 2, we introduce our proposed framework with TCA and MIDA methods. Section 3 provides a description of the employed test bench and discusses on simulation results. Finally, Sect. 4 concludes the paper and comments on how this algorithm can be further extended.

2 Proposed domain adaptation framework

2.1 Motivation and roadmap

The main motivation of the proposed approach is to maximize the classification accuracy in the target domain for pedestrian detection application. We employed two approaches based on two feature-based domain adaptation algorithms named TCA and MIDA. In this section, the proposed framework for using these two methods is explained. In addition, the proposed algorithms are mathematically formulated.

The aim of TCA and MIDA algorithms is to create a new feature space that significantly decreases the dimension of the original feature space, and simultaneously approximates the distribution of the source and target datasets along with preserving important features of the datasets. Although these methods do not require a lot of complex calculations and their implementations are simple, however, they achieve accurate results.

The overall block diagram of the employed TCA and MIDA algorithms is presented in Fig. 1.

At the first step of this algorithm, a preprocessing procedure is required. In the first step, the image frames were extracted from the target video. Then, RGB images were converted to grayscale images. In order to detect pedestrians

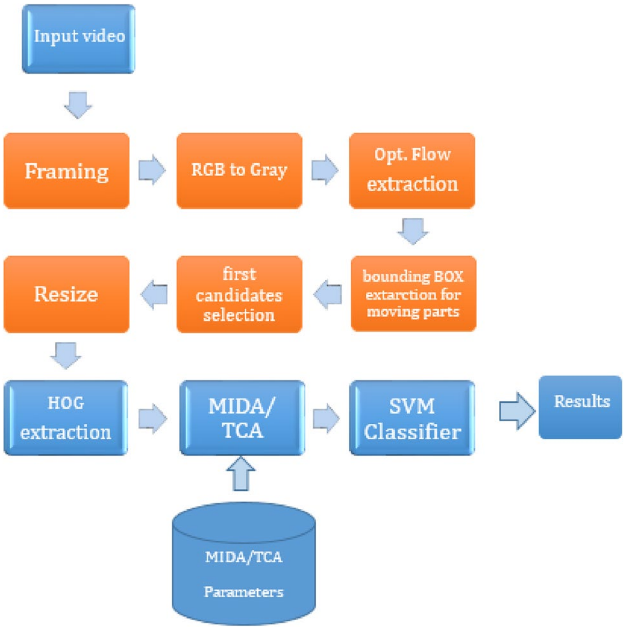

Fig. 1 The proposed block diagram of pedestrian detection algorithm using TCA/MIDA domain adaptation methods (color figure online)

in a video sequence, it is desirable to use cues that discriminate pedestrians from the other objects such as the background and cars. Due to the various researches in this field, utilizing the cues such as size and motion is a good choice. Optical flow extraction method was used as motion cues. Also, using a threshold, the moving objects that were larger or smaller than pedestrians were removed to make the task of detection become easier.

The selected candidates, which include both positive and negative samples, were converted to 128 × 64 pixels images. At the next stage, we extracted HOG (histogram of gradient) features that are known as useful and efficient features for pedestrian detection [2]. Then, the extracted features were transformed to the appropriate feature subspace (TCA or MIDA subspace). Moreover, the subspace parameters for the used datasets were carefully selected in this study. The final stage is the classification using an SVM classifier.

2.2 Mathematical formulation

We represent a domain by two main components: a feature space of inputs features $\mathcal{X}$ that are extracted from pattern images (either pedestrian or non-pedestrian) and a marginal probability distribution of inputs $P(X)$, where $X = \left\{x_1 \ldots x_n\right\} \in \mathcal{X}$ is a set of learning samples of both pedestrian and non-pedestrian objects. In general, if two domains are different, they may have different feature spaces or different marginal probability distributions. In this study, we focused on the settings where there are only one source and one target domain sharing the same feature space. We also assume that labeled data D_S are available in the source domain, while only a few labeled data D_T are available in the target domain. More specifically, let the source domain data be $D_S = \{\left(x_{S_1}.y_{S_1}\right) \ldots (x_{S_{n_1}}.y_{S_{n_1}})\}$, where $x_{S_i} \in x$ is the input and $y_{S_i} \in \mathcal{Y}$ is the corresponding label. Similarly, let the target domain data be $D_T = \left\{x_{T_1} \ldots x_{T_{n_2}}\right\}$, where the input x_{T_i} is also in $\mathcal{X}$. A few samples of D_T are in the form of the pair $\left(x_{t_i}.y_{t_i}\right)$.

Let us assume that $P\left(X_S\right)$ and $Q\left(X_T\right)$ (or P and Q in short) be the marginal distributions of $X_S = \left\{x_{S_i}\right\}$ and $X_T = \left\{x_{T_i}\right\}$ from the source and target domains, respectively. In general, P and Q can be different. Our task is prediction of the labels y_{T_i}s corresponding to inputs x_{T_i}s in the target domain. Most of the domain adaptation methods assume that $\mathcal{P} \neq \mathcal{Q}$, but $P\left(Y_S|X_S\right) = P(Y_T|X_T)$. However, in real applications, the conditional probability $P(Y|X)$, due to dynamic or noisy factors that affect the observed data, will change across the domains. In this case, we assume a weaker assumption that $\neq \mathcal{Q}$, but there is a transformation $\emptyset$ such that $P\left(\emptyset\left(X_S\right)\right) \approx P\left(\emptyset\left(X_T\right)\right)$ and $P\left(Y_S|\emptyset\left(X_S\right)\right) \approx P\left(Y_T|\emptyset\left(X_T\right)\right)$. Standard supervised learning methods can be applied to the converted source data in the new space $\emptyset\left(X_S\right)$ along with the corresponding labels, and thus models are trained to be used on the converted target data $\emptyset\left(X_T\right)$.

One of the key issues is how to find the $\emptyset$ transformation. As we do not have any labels in the target domain, we are not able to obtain $\emptyset$ by directly minimizing the distance between $P\left(Y_S|\emptyset\left(X_S\right)\right)$ and $P(Y_T|\emptyset\left(X_T\right)$. Therefore, the transformation $\emptyset$ is learned so that the distance between the marginal distributions $P\left(\emptyset\left(X_S\right)\right)$ and $P\left(\emptyset\left(X_T\right)\right)$ is decreased. In addition, $\emptyset\left(X_S\right)$ and $\emptyset\left(X_T\right)$ preserve the important properties of X_S and X_T. We assume that $\emptyset$ satisfies the relation $P(Y_S|\emptyset\left(X_S\right)) \approx P(Y_T|\emptyset\left(X_T\right))$. Although the domain adaptation approach based on this assumption is more challenging, it is more realistic. Finally, the classifier f is trained on $\emptyset\left(X_S\right)$ and Y_Ss are used for the classification on $\emptyset\left(X_T\right)$.

Let us assume that $\emptyset$ is the transformation, which is created by a universal kernel function. The distance between two distributions $\mathcal{P}$ and $\mathcal{Q}$ can be obtained by measuring the distance between the empirical means of the two domains as follows:

$$\operatorname{Dist}\left(X_S', X_T'\right) = {}^1/_{n_1} \sum_{i=1}^{n_1} \emptyset\left(x_{Si}\right) - {}^1/_{n_2} \sum_{i=1}^{n_2} \emptyset\left(x_{Ti}\right)_{\mathcal{H}}^2. \tag{1}$$

Hence, the optimal nonlinear transformation $\emptyset$ is obtained by minimizing (1). As $\emptyset$ is strongly nonlinear, a direct optimization can cause the problem to be trapped in local minima.

Instead of finding a nonlinear $\emptyset$ transformation directly, we used a domain adaptation method based on the dimensionality reduction approach called maximum mean discrepancy embedding (MMDE) [28]. In this method, both the source and target domain data are transferred to a final common low-dimensional space using a nonlinear Ø transformation, and then the corresponding kernel matrix K is learned by solving a semi-definite programming (SDP) equation [29]. Assume that Gram matrices are denoted for the source, target, and cross-domain data with $\boldsymbol{K}^{S.S}$, $\boldsymbol{K}^{S.T}$ and $\boldsymbol{K}^{T.T}$, respectively. The main purpose is to learn the kernel matrix $\boldsymbol{K}$:

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{K}^{S.S} & \boldsymbol{K}^{S.T} \\ \boldsymbol{K}^{S.T} & \boldsymbol{K}^{T.T} \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)} \tag{2}$$

Equation (2) means that the kernel matrix on all data is defined in such a way that the distance between the mapped source and target domains data is minimized, while the data variance is increased as much as possible. The MMD distance can be rewritten as $tr(\boldsymbol{KL})$, where we have: $\boldsymbol{K} = \left[\emptyset^T(X_i)\emptyset(X_j)\right]$ and $\boldsymbol{L}_{ij} = 1/n_1^2$ if $x_i.x_j \in X_S$, or $\boldsymbol{L}_{ij} = 1/n_2^2$ If $x_i.x_j \in X_T$, otherwise, $\boldsymbol{L}_{ij} = -\left(1/n_1n_2\right)$.

Besides minimizing the trace of ψ, we also have the following constraints which are motivated from maximum variance unfolding (MVU):

1. The distance is preserved, i.e., $K_{ii} + K_{jj} - 2K_{ij} = d_{ij}^2$ for all i,j ψ such that $(i,j) \in N$;
2. The embedded data are centered;
3. The trace of $\boldsymbol{K}$ is maximized.

For all i,j, if x_i ψ and x_j are k-nearest neighbors, we denote this by using $(i,j) \in N$. Therefore, the MMDE objective function can be written as follows:

$$\begin{aligned} &\min_{K=\tilde{K}+\varepsilon I} tr(\boldsymbol{KL}) - \lambda tr(\boldsymbol{K}) \\ &s.t. K_{ii} + K_{jj} - 2K_{ij} = d_{ij}^2, \forall(i,j) \in N, \\ &K\mathbf{1} = -\varepsilon\mathbf{1}.\tilde{K} \geq 0 \end{aligned} \tag{3}$$

The first term in the objective function minimizes the distance between distributions, and the second term maximizes the variance in the feature space. $\lambda \geq 0$ is a parameter to hold a trade-off between these two terms.

To deal with the limitations of using MMDE, a useful framework has been used to find the nonlinear transformation Ø, based on the kernel feature extraction. This will help us to get rid of SDP's high computational complexity. In addition, the learned kernel can be used for unseen patterns as well. It should be noted that instead of using a two-step approach, such as MMDE, an integrated kernel learning approach has been used. The matrix K in (2) can be decomposed as $K = \left(KK^{-\frac{1}{2}}\right)\left(K^{-\frac{1}{2}}K\right)$, which is called the empirical kernel map. Consider the matrix $\tilde{W} \in \mathbb{R}^{(n_1+n_2)\times m}$ that projects the features of the empirical kernel map to an m-dimensional space ($m \ll n_1 + n_2$). The final kernel matrix is as follows:

$$\tilde{K} = \left(KK^{-\frac{1}{2}}\tilde{W}\right)\left(\tilde{W}^TK^{-\frac{1}{2}}K\right) = KWW^TK \tag{4}$$

where $W = K^{-\frac{1}{2}}\tilde{W}$. The corresponding kernel between each of the two patterns x_i and x_j is $\tilde{k}\left(x_i.x_j\right) = K_{x_i}^TWW^TK_{x_j}$, where $k_x = \left[k\left(x_1.x\right) \ldots k\left(x_{n_1+n_2}.x\right)\right]^T \in \mathbb{R}^{(n_1+n_2)}$. The kernel $\tilde{k}$ provides a parametric form for evaluations of out-of-samples kernels. Utilizing the definition of $\tilde{k}$ in (1), the MMD distance between the empirical means of the two domains X_S' and X_T' is written as follows:

$$\text{Dist}\left(X_S'.X_T'\right) = tr\left(\left(KWW^TK\right)L\right) = tr\left(W^TKLKW\right) \tag{5}$$

In domain adaptation, it is desired to learn the transformation Ø by both reducing the distance between $P\left(\emptyset\left(X_S\right)\right)$ and $P\left(\emptyset\left(X_T\right)\right)$ as well as preserving the data main properties (e.g., maximizing data variance). The representation of data in the final space is W^TK, where the ith column $[W^TK]_i$ denotes the embedding coordinates of x_i. Therefore, the variance of the mapped samples is W^TKLKW, where $\boldsymbol{H} = I_{n_1+n_2} - \left(\frac{1}{n_1+n_2}\right)11^T$ is the centering matrix, $1 \in \mathbb{R}^{(n_1+n_2)}$ is an all-one column vector, and $I_{n_1+n_2} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}$ is the identity matrix.

2.2.1 Pedestrian detection using semi-supervised TCA

A useful and an efficient dimensionality reduction method for domain adaptation has been introduced by TCA [25], where it is possible to preserve the variance of data as much as possible and to reduce the distance between different distributions across domains.

Unsupervised TCA method does not employ label information for learning. However, utilizing an effective kernel function, the unsupervised TCA can be extended to a semi-supervised TCA with the aim of increasing the dependence with the data labels. Thereby, the label information is propagated from the labeled samples (usually source domain) to the unlabeled samples (usually target domain). In semi-supervised domain adaptation methods, labeled and unlabeled samples are drawn from two different domains.

The three main objectives of the semi-supervised TCA learning method are firstly maximum adaptation of source and target domains data distributions in the new space, secondly high dependency on data labels and finally preserving local geometry of data. The first goal is to minimize MMD between the source and target domains in the embedded space as stated in (5). The second goal is to increase the dependency between labels and the embedding. The kernel matrix in this case is defined as:

$$\tilde{K}_{yy} = \gamma K_l + (1-\gamma)K_v \tag{6}$$

where $\gamma \geq 0$ is a trade-off parameter. In Eq. (6), the first term increases label dependence on the labeled data, and the second term aids to maximize the variance on data in both domains. In this case, the goal is to maximize the following equation:

$$tr\left(H\left(KWW^{T}K\right)H\tilde{K}_{yy}\right) = tr\left(W^{T}KH\tilde{K}_{yy}HKW\right) \tag{7}$$

The third goal is to preserve local geometry of data. MMDE preserves the local geometry of data by imposing distance constraints on the desired kernel matrix. More specifically, we consider $\mathcal{N} = \left\{\left(x_i.x_j\right)\right\}$ as the data pair set in the k-nearest neighbor of each other, where $d_{ij} = ||x_i - x_j||_2$ is the distance between the data in the original input space. For each pair $\left(x_i.x_j\right)$ in $\mathcal{N}$, the constraint $K_{ii} + K_{jj} - 2K_{ij} = d_{ij}^2$ is added to the optimization problem. Therefore, we have a semi-definite optimization problem with a large number of constraints.

For this purpose, a graph is defined as $m_{ij} = \exp\left(-d_{ij}^2/2\sigma^2\right)$, in which x_i belong to the k nearest neighbors of x_j or vice versa. The Laplacian matrix of the graph is defined as $\mathcal{L} = D - M$, in which D is a diagonal matrix with the value $d_{ii} = \sum_{j=1}^{n} m_{ij}$. If x_i and x_j are neighbors to each other in the original input space, the distance between their embedding coordinates is preferred to be small. The third goal is to minimize the following equation:

$$\sum_{(i.j)\in\mathcal{N}} m_{ij}||\left[W^{T}K\right]_i - \left[W^{T}K\right]_j||^2 = tr\left(W^{T}K\mathcal{L}KW\right) \tag{8}$$

By combining three goals, the ultimate optimization problem is as follows:

$$\begin{aligned} &\min_{W} tr\left(W^{T}KLKW\right) + \mu tr\left(W^{T}W\right) + \frac{\lambda}{n^2} tr\left(W^{T}K\mathcal{L}KW\right) \\ &\qquad s.t. \quad W^{T}KH\tilde{K}_{yy}HKW = I \end{aligned} \tag{9}$$

where $\lambda \geq 0$ and $\mu > 0$ are trade-off parameters, $n^2 = \left(n_1 + n_2\right)^2$ is a normalization term, and $I \in \mathbb{R}^{(m\times m)}$ is the identity matrix. This optimization problem actually contains a non-convex constraint $W^{T}KH\tilde{K}_{yy}HKW = I$.Equation (9) is written as follows:

$$\max_{W} tr\left\{\left(W^{T}K(L + \lambda\mathcal{L})KW + \mu I\right)^{-1}\left(W^{T}KH\tilde{K}_{yy}HKW\right)\right\} \tag{10}$$

This method is known as semi-supervised TCA (SSTCA). The solution to this problem is obtained by eigendecomposition of $(K(L + \lambda\mathcal{L})K + \mu I)^{-1}KH\tilde{K}_{yy}HK$.

2.3 Maximum independence domain adaptation (MIDA)

TCA algorithm has two basic problems. First, TCA method is for cases where the source and target domains are discrete. However, in case samples come in a stream the data distribution changes continuously. The next problem is that it is possible the conditional probability $P(Y|X)$ changes for samples with different backgrounds. In methods such as TCA, all samples are transferred to a common subspace; therefore samples with the same appearance but different concepts cannot be discriminated.

One of the main reasons leading to the changes between train and test domain data is the difference between measuring instruments (cameras). The maximum independence domain adaptation method (MIDA) covers the change in continuous or discrete data distributions due to different domains [32]. In MIDA method, domain features are first extracted to describe the background of each sample. Then, a subspace is created in which samples and their domain features are maximally independent regarding Hilbert–Schmidt independence criterion (HSIC) [30]. Using this strategy, any difference between domain distributions is considered. It is also possible to extend the adaptation algorithm to include different types of distributions.

To map the samples based on the background, feature augmentation is performed by combining the main features with the domain features. It is also possible to use a semi-supervised MIDA algorithm, so that data labels can also been utilized. Both unsupervised and semi-supervised MIDA methods can be used for the case of one or more source and target domains. Although the MIDA method is designed for unsupervised conditions (when none of the target samples are labeled), it can be used in unlabeled or labeled conditions in both domains. Label information can also be continuous (two or multiple classes classification) or discrete (regression).

2.3.1 Pedestrian detection using semi-supervised MIDA

HSIC is an appropriate method to evaluate the dependence between two sample sets X and Y. Assume that k_x and k_y are two kernel functions associated with reproducing kernel Hilbert spaces $\mathcal{F}$ and $\mathcal{G}$, respectively, and p_{xy} is the joint distribution. HSIC is described as the square of the Hilbert–Schmidt norm of the cross-covariance operator C_{xy}:

$$\begin{aligned} \mathrm{HSIC}\left(p_{xy}, \mathcal{F}, \mathcal{G}\right) &= ||C_{xy}||_{H}S^2 \\ &= E_{xx'yy'}\left[k_x\left(x, x'\right)k_y\left(y, y'\right)\right] \\ &\quad + E_{xx'}\left[k_x\left(x, x'\right)\right]E_{yy'}\left[k_y\left(y, y'\right)\right] \\ &\quad - 2E_{xy}\left[E_{x'}\left[k_x\left(x, x'\right)\right]E_{y'}\left[k_y\left(y, y'\right)\right]\right] \end{aligned}$$

where $E_{xx'yy'}$ is the expectation over independent pairs (x, y) and $\left(x', y'\right)$ drawn from p_{xy}. It can be shown that if and only if x and y are independent, $HSIC\left(p_{xy}, \mathcal{F}, \mathcal{G}\right)$ is zero [31]. The larger the HSIC is, the stronger the dependence with respect to the choice of kernels will be. Let $Z = X \times Y = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, and$K_x, K_y \in R^{n \times n}$ be the kernel matrices of X and Y, respectively. HSIC can then be written as:

$$\mathrm{HSIC}(Z, \mathcal{F}, \mathcal{G}) = (n-1)^{-2} tr\left(K_x H K_y H\right) \quad (11)$$

where $H = I - n^{-1} 1_n 1_n^T$ is the centering matrix [31].

Consider that our setup includes n_{dev} cameras. Domain feature vectors are $d \in R^{n_{dev}}$, so that if the sample is generated from the pth camera, $d_p = 1$, and otherwise, $d_p = 0$. According Eq. (11), the kernel matrix K_d of the domain features should be computed for HSIC. Consider $\mathrm{D} = \left[d_1 \ldots d_n\right] \in R^{m_d \times n}$ and m_d is the dimension of domain feature vector. We have: $K_d = D^T D$. In the usual domain adaptation problems that involve different discrete domains, the above equation can be used to build domain features.

Consider that $X \in R^{m \times n}$ is a matrix containing n samples. The training and test samples are combined together without the requirement of domain labels. Then, feature vectors are augmented. A linear/nonlinear mapping function Φ can be used to map X to a new space. It is not necessary to know the function Φ exactly; however, the inner product of the function Φ *(X)* is represented by the kernel matrix $K_x = \boldsymbol{\Phi}(X)^T \boldsymbol{\Phi}(X)$. Then, a projection matrix $\tilde{W}$ is used to project Φ *(X)* to a subspace with the dimension h. The result is samples projected to the new space in the form of $Z = \tilde{W}^T \boldsymbol{\Phi}(X) \in R^{h \times n}$. The main idea is to express each projection direction as a linear combination of all samples in the new space. This equation is defined as $\tilde{W} = \boldsymbol{\Phi}(X) W$, where $W \in R^{h \times n}$ is the projection matrix that has to be learned. The projected samples are defined as follows:

$$Z = W^T \boldsymbol{\Phi}(X)^T \boldsymbol{\Phi}(X) = W^T K_x \quad (12)$$

If the projected features are independent from the domain features, then we are no longer able to identify the background of samples using the projected features. In this case, it is suggested that the difference between the domains in the new subspace is reduced. Therefore, after removing the scale factor in the Hilbert–Schmidt space, we minimize the following equation:

$$tr\left(K_Z H K_d H\right) = tr\left(K_x W W^T K_x H K_d H\right) \quad (13)$$

where K_z is the kernel matrix of Z. The goal of retaining important data properties can be achieved by maximizing the trace of the covariance matrix of the projected samples. The covariance matrix is defined as:

$$\mathrm{cov}(Z) = \mathrm{cov}\left(W^T K_x\right) = W^T K_x H K_x W \quad (14)$$

MIDA aligns samples with completely different backgrounds without any need to label information. However, if the labels of some samples are available, they can be used in the subspace learning process, which can be useful in predicting the data label. In semi-supervised MIDA method, as there is no explicit difference between the samples' domain labels, labeled and unlabeled samples can exist in both domains. HSIC is designed to maximize the dependency between mapped features and labels.

In classification problems, we define the label matrix $Y \in R^{c \times n}$, for c classes. If x_i has a label belonging to the class j, we have $y_{i,j} = 1$; otherwise $y_{i,j} = 0$. In regression problems, the label matrix $Y \in R^{1 \times n}$ is equivalent to the target value of x_i if it has a label, otherwise it will be zero. For the label kernel matrix, linear kernel function is: $K_y = Y^T Y$.

The SMIDA objective function is defined as follows:

$$\begin{aligned} &\max_W tr\left(W^T K_x(-HK_d H + \mu H + \gamma H K_y H\right) K_x W), \\ &\quad s.t. \quad W^T W = I \end{aligned} \quad (15)$$

where $\gamma > 0$ and $\mu > 0$ are trade-off parameters. By applying the Lagrange multipliers method, it is concluded that W is the eigenvectors corresponding to the h largest eigenvalues of $K_x(-HK_d H + \mu H + \gamma H K_y H) K_x$.

To calculate K_x, an appropriate kernel function has to be selected. The widely used kernels are linear kernel $k(x.y) = x^T y$, polynomial kernel $k(x.y) = \left(\sigma x^T y + 1\right)^d$, and Gaussian radial basis function kernel (RBF) $k(x.y) = \exp x - y^2/2\sigma^2$. Polynomial and RBF kernels project the original features to a space with a higher dimension. In such a space, it is possible to detect more types of dependency. The most important point in these methods is to select the kernel with a proper width σ. In our paper, a polynomial kernel has been used and an appropriate σ has been empirically selected during the simulation.

In TCA, the maximum mean discrepancy (MMD) criterion is used to measure the difference between distributions. TCA can only be used when there are two discrete domains, while MIDA can be used in a variety of conditions, including multiple domains and continuous changes in distributions. MIDA method is also used in a completely unsupervised manner.

3 Experiments

In this section, we present some experiments to evaluate the behavior of TCA and MIDA algorithms in two cases: (1) The source dataset is totally different from target dataset (domain adaptation task), and (2) only a few labeled data are available from the target (semi-supervised learning task). All experiments were conducted on two different datasets, and the results were compared against several methods.

3.1 Dataset description

We ran all experiments on INRIA dataset [2] and CUHK dataset [22]. The source data are from INRIA for pedestrians, which contains 3634 labeled data of which 2161 are positive samples and 1218 are negative samples. The target data are from CUHK dataset, a 60-min video taken by a fixed camera by the rate of 25 frames per second. The great challenge of our research is a remarkable difference between the source and target domains datasets, which severely degrades the classifier performance. All methods were tested with the same data and parameters. In this paper, the histogram of gradients (HOG) was used to extract features for the pedestrian detection application [2].

In the implementations, the HOG features were first extracted for all positive and negative samples of INRIA (source) and CUHK (target) datasets. Before we extract the HOG features from the INRIA and CUHK images, we reduced the size of the images to 64×128 pixels. As a result, the size of the input image is $64 \times 128 \times 3$ and the output HOG feature vector's length is 3780. In addition, according to the selection method that can be either semi-supervised or unsupervised, we divided the mixed source and target samples into training and test sets using K-FOLD method with $\mathrm{K} = 5$. We used the training data to train a nonlinear SVM with a polynomial kernel. After this step, we applied the proposed domain adaptation method. We then employed the trained SVM to evaluate the test samples.

3.2 Experimental setup

The most effective and decisive parameters in the work are the adaptation parameters (λ, γ, σ, μ, m), whose impact is studied on the results. Finally, the parameters' values associated with the best results are used in the final experiments.

The kernel type in the employed domain adaptation methods is a polynomial kernel, and σ is an important factor in the formulation of this kernel function. Also, among all the noted parameters σ has the most influence on the results. Therefore, particular attention was paid to select this parameter using K-FOLD evaluation method. The tested values for the sigma parameter are selected from the set $\{0.0001, 0.001, 0.01, 0.1, 1, 10\}$, and the optimum values of the other parameters are shown in Table 1.

In the ITL method, no kernel function has been used and the important factor in this method is the rate of dimensionality reduction in PCA method. We set the threshold in PCA so that 95% of signal energy is reserved. Also, the number of iterations was set to 20. These values were obtained empirically as a result of repeated testing.

Classification accuracy is computed as the ratio of the number of correctly classified test samples to the total number of test samples. Equivalently, the classification error is defined as 1-accuracy.

Table 1 The optimum values of parameters in domain adaptation methods

m	γ	μ	λ
100	0.9	1	1

We used four different domain adaptation methods, none of which has previously employed pedestrian detection. The results were compared with the general SVM method that is trained on the labeled source data and some of the labeled target data. For this purpose, we first mapped the samples to the new reduced-dimension feature space. After training SVM classifier by the transformed samples, it is applied to the test samples for comparison purpose. The four methods for comparison in this research are as follows:

- **TCA** [25]: Transfer component analysis
- **MIDA** [32]: Maximum independence domain adaptation
- **KPCA** [33]: Kernel component analysis
- **ITL** [26]: Information-theoretical learning.

All these four methods learn a feature transformation that transforms training data from both the source and target domains into a common feature space.

We carried out our experiments on two domains, the first one as the source and the other one as the target. In the unsupervised setting, the domains are disjoint, while in the semi-supervised setting a few images per class of the target domain are added to the source. Also, source and target data were randomly collected from the datasets. The target data were from 60 min of the CUHK video containing 3335 images, of which 2668 samples were labeled and used along with the INRIA source dataset in the semi-supervised method for training step. The remaining 667 unlabeled samples were also used for testing. K-FOLD method ($K = 5$) was used to divide the target data into training and test sets. The average results of five repetitions are reported in the following figures and tables. The block diagram of the implementation process and testing of the proposed method is also shown in Fig. 2.

INRIA dataset includes accessible labeled images, but the CUHK dataset contains a video sequence, which it is necessary to first convert the video to image frames. In addition, as the negative and positive labeled samples of this dataset are not available, it is required to detect and label the pedestrians and backgrounds in the images. For this purpose, as explained in Wang's proposed method [22], proper background–pedestrian discriminating factors have to be utilized.

In this research, two main factors to discriminate pedestrians from backgrounds (tree, building, traffic light, machine, etc.) were used. These factors include: 1) motion to distinguish moving objects (cars, pedestrians, motorcycles, etc.)

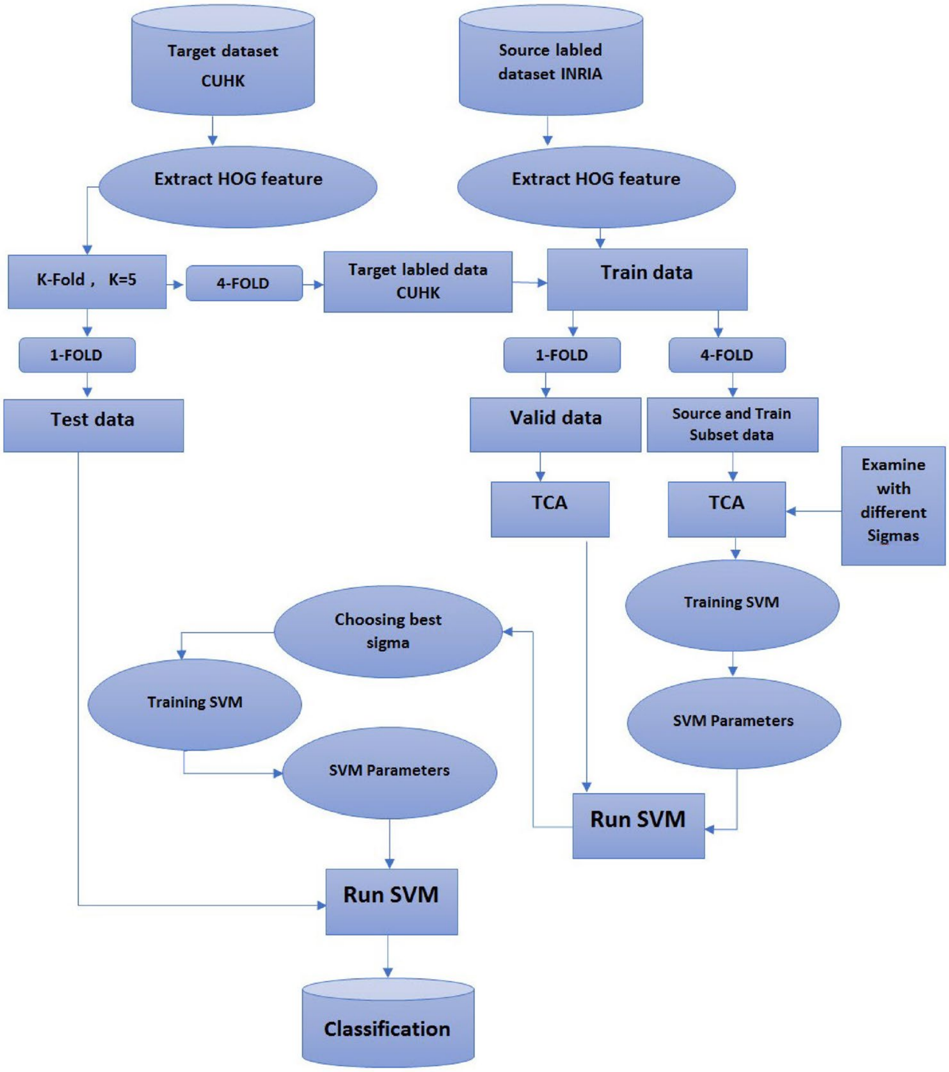

Fig. 2 Block diagram of the implemented proposed method

from non-moving objects (trees and the background), and 2) size to distinguish moving objects from each other. Optical flow method [34, 35] was used to apply the motion factor. Also, to apply the size factor, a threshold was employed to remove items that are smaller or bigger than the normal size of the pedestrians.

The automatically detected objects were framed, and the framed objects' sizes were compared with a pre-defined threshold. Finally, the selected samples were manually labeled as positive and negative samples.

3.3 Evaluating transfer component analysis as a domain adaptation method

By applying the TCA domain adaptation method to the INRIA (source) and the CUHK (target) datasets, the classifier performance was notably improved, and the average accuracy increased from 80.51 to 96.58%, as shown in Table 2. As the TCA adaptation method is a dimension reduction method, it efficiently reduces computational cost (time and complexity) that is one of the most important advantages of the proposed approach.

The effect of the parameter σ was also been studied on the classifier performance. As depicted in Fig. 3. The best σ in the TCA method is 0.01.

Table 2 Comparison of the SVM classifier performance on the INRIA and the CUHK datasets before and after applying TCA

Repetition	Detection results	
	Before TCA (%)	After TCA (%)
$i=1$	79.16	97.60
$i=2$	80.51	96.25
$i=3$	81.56	95.95
$i=4$	79.01	96.10
$i=5$	82.31	97.00
Average	80.51	96.58

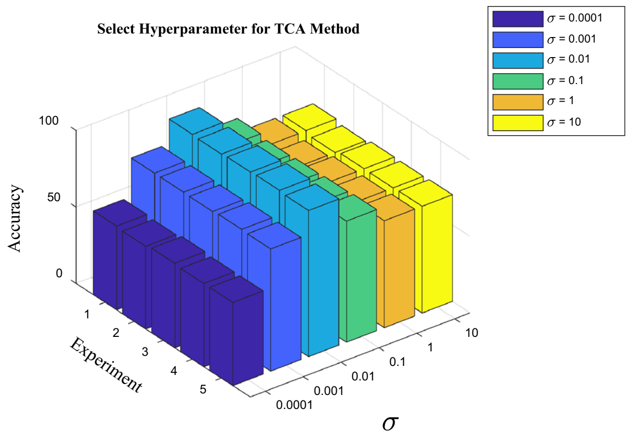

Fig. 3 The effect of σ-parameter on the classifier performance in the TCA method

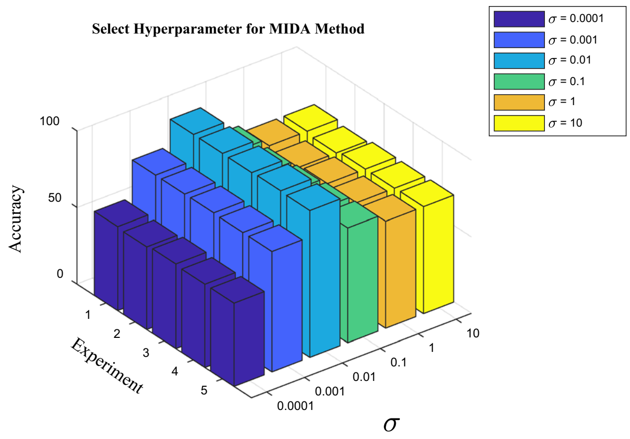

Fig. 4 The effect of σ-parameter on the classifier performance in the MIDA method

Table 3 Comparison of the SVM classifier performance on the INRIA and the CUHK datasets before and after applying MIDA

Repetition	Detection results	
	Before MIDA (%)	After MIDA (%)
$i=1$	79.16	97.60
$i=2$	80.51	97.00
i=3	81.56	96.10
$i=4$	79.01	96.40
$i=5$	82.31	97.30
Average	80.51	96.88

Table 4 Comparison of the SVM classifier performance on the INRIA and the CUHK datasets before and after applying ITL

Repetition	Detection results	
	Before ITL (%)	After ITL (%)
$i=1$	79.16	85.91
$i=2$	80.51	86.21
$i=3$	81.56	86.96
$i=4$	79.01	86.81
$i=5$	82.31	86.36
Average	80.51	86.44

3.4 Evaluating maximum independence as a domain adaptation method

The purpose of the conducted experiment in this section is to compare the performance of the SVM classifier on the source and target datasets before and after applying MIDA method. By applying the MIDA adaptation method, the SVM classifier performance has remarkably improved, and the average accuracy increased from 80.51 to 96.88%. The results are brought in Table 3. As MIDA adaptation method reduces the feature space dimension same as the TCA adaptation method, the computational time is distinguishably decreased. The σ-parameter in the presented results in Table 3 was selected 0.01, according to the graph depicted in Fig. 4.

3.5 Comparing two selected methods with other common methods of domain adaptation

The purpose of this experiment is to compare the employed domain adaptation methods, MIDA and TCA, with the other existing domain adaptation methods for improving pedestrian detection performance. Typical domain adaptation techniques used in the comparisons were KPCA and ITL.

Comparing with the employed methods, it is observed that ITL and KPCA methods also lead to a significant improvement over the general SVM classifier. In using ITL, the average accuracy increased from 80.51 to 86.44% and in using KPCA the average accuracy increased from 80.51 to 96.79%. As shown in Table 4, the classification accuracy of ITL method is weaker and is not competitive with the other ones. However, because this method does not include any kernel function, it is possible to employ it for online pedestrian detection applications. The KPCA method provides satisfactory results as presented in Table 5.

The effect of changing the parameter σ on the classifier performance was also studied. As shown in Fig. 5, it is observed that the optimum σ-choice in the KPCA method is 1. Also, the change in this parameter in the ITL method has not been considered as there is no kernel function in this method.

Table 5 Comparison of the SVM classifier performance on the INRIA and the CUHK datasets before and after applying KPCA

Repetition	Detection results	
	Before KPCA (%)	After KPCA (%)
$i = 1$	79.16	98.35
$i = 2$	80.51	96.40
$i = 3$	81.56	96.10
$i = 4$	79.01	95.50
$i = 5$	82.31	97.60
Average	80.51	96.79

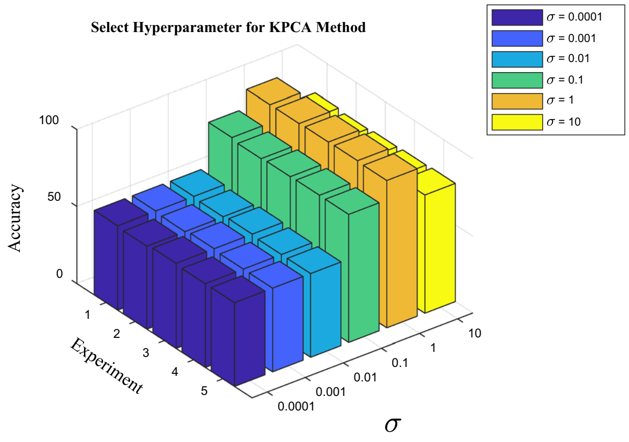

Fig. 5 The effect of σ-parameter on the classifier performance in the KPCA method

Table 6 The average classifier performance results before and after applying domain adaptation methods

Methods	Baseline	TCA	MIDA	ITL	KPCA
Average accuracy	80.51%	96.58%	96.88%	86.44%	96.79%

3.6 Comparison between different domain adaptation methods and standard SVM

The average performance results of the standard SVM classifier before and after applying different domain adaptation methods, TCA, MIDA, KPCA and ITL, are compared in Table 6. According to this table, it is obvious that employing domain adaptation techniques provides a significant improvement in the classifier performance for pedestrian detection applications. Among these methods, MIDA, TCA and KPCA demonstrate greater improvement in classifier accuracy in comparison with ITL. However, ITL method is more appropriate for online applications because it is not essential to apply a kernel function on new input samples.

4 Conclusion

In this study, we evaluated the usage of TCA and MIDA algorithms in the context of pedestrian detection application. In domain adaptation algorithms, the goal is to adapt a trained detector in a general domain to a specific domain so that the detector's performance does not fall in the new domain. Thereby, an arbitrary detector that is trained in the source domain with available labels can be used in the target domain with no/enough labeled data. Among these methods, TCA and MIDA techniques have never been used to improve pedestrian detection methods. The final results from the proposed method suggest that two employed domain adaptation algorithms increase the performance of pedestrian detection methods. The experimental results of the proposed approach obtained on INRIA train dataset and CUHK test dataset demonstrate improvement in the classification accuracy from 81.5 to 96.58% by using TCA. Moreover, on the same datasets and employing MIDA, the classification accuracy has been increased from 80.51 to 96.88%. Moreover, we proposed the use of appropriate samples from the source domain data instead of using all samples. The results of this study can be employed in the situations where the training samples, cameras, or environmental conditions are different from the test scenario. As a future work, other feature-based methods and domain adaptation algorithms using more precise objective metric can be employed in multi-domain scenarios.

References

1. Sun D, Watada J (2015) Detecting pedestrians and vehicles in traffic scene based on boosted HOG features and SVM. In: IEEE 9th International symposium on intelligent signal processing, WISP, pp 1–4
2. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE computer society conference on computer vision and pattern recognition, CVPR, vol 1, pp 886–893
3. Felzenszwalb PF, Girshick RB, McAllester D (2010) Cascade object detection with deformable part models. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 2241–2248
4. Dollár P, Tu Z, Perona, P et al (2009) Integral channel features. BMVC http://authors.library.caltech.edu
5. Felzenszwalb P, McAllester D, Ramanan D (2008) A discriminatively trained, multiscale deformable part model. In: IEEE conference on computer vision and pattern recognition (CVPR), pp 1–8
6. Dollár P, Belongie SJ, Perona P (2010) The fastest pedestrian detector in the west. In: Proceedings of the British machine vision conference, pp 68.1–68.11
7. Dollár P, Appel R, Belongie S et al (2014) Fast feature pyramids for object detection. IEEE Trans Pattern Anal Mach Intell 36(8):1532–1545
8. Rosenberg C, Hebert M, Schneiderman H (2005) Semi-supervised self-training of object detection models. WACV/MOTION, (2)

9. Levin A, Viola PA, Freund Y (2003) Unsupervised improvement of visual detectors using co-training. In: IEEE international conference on computer vision, vol 1, pp 626–633
10. Javed O, Ali S, Shah M (2005) Online detection and classification of moving objects using progressively improving detectors. In: IEEE conference on computer vision and pattern recognition, CVPR, vol 1, pp 696–701
11. Yang J, Yan R, Hauptmann AG (2007) Cross-domain video concept detection using adaptive SVMs. In: Proceedings of 15th ACM international conference on multimedia, pp 188–197
12. Qi GJ, Aggarwal C, Huang T (2011) Towards semantic knowledge propagation from text corpus to web images. In: Proceedings of conference world wide web, pp 297–306
13. Duan L, Tsang IW, Xu D et al (2009) Domain transfer SVM for video concept detection. In: IEEE conference on computer vision and pattern recognition, pp 1375–1381
14. Pang J, Huang Q, Yan S et al (2011) Transferring boosted detectors towards viewpoint and scene adaptiveness. IEEE Trans Image Process 20(5):1388–1400
15. Roth PM, Sternig S, Grabner H et al (2009) Classifier grids for robust adaptive object detection. In: Computer vision and pattern recognition, CVPR, IEEE Conference pp 2727–2734
16. Roth PM, Grabner H, Bischof H, et al (2005) On-line conservative learning for person detection. IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance: 223-230
17. All K, Hasler D, Fleuret F (2011) Flow boost appearance learning from sparsely annotated video. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 1433–1440
18. Jain V, Learned-Miller E (2011) Online domain adaptation of a pre-trained cascade of classifiers. In: IEEE conference on computer vision and pattern recognition, CVPR, pp 577–584
19. Rosenberg C, Hebert M, Schneiderman H (2005) Semi-supervised self-training of object detection models. In: 7th IEEE workshops on applications of computer vision (WACV/MOTION'05),vol 1, pp 29–36
20. Levin A, Viola PA, Freund Y (2003) Unsupervised improvement of visual detectors using co-training. In: Proceedings of the 9th IEEE international conference on computer vision, vol 1, pp 626–633
21. Wu B, Nevatia R (2007) Improving part based object detection by unsupervised, online boosting. In: IEEE conference on computer vision and pattern recognition, pp 1–8
22. Wang X, Wang M, Li W (2014) Scene-specific pedestrian detection for static video surveillance. IEEE Trans Pattern Anal Mach Intell 36(2):361–374
23. Mao Y, Yin Z (2015) Training a scene-specific pedestrian detector using tracklets. In: IEEE winter conference on computer vision (WACV), pp 170–176
24. Maâmatou H, Chateau T, Gazzah S et al (2016) Sequential Monte Carlo filter based on multiple strategies for a scene specialization classifier. EURASIP J Image Video Process 1:40
25. Pan SJ, Tsang IW, Kwok JT et al (2011) Domain adaptation via transfer component analysis. IEEE Trans Neural Netw 22(2):199–210
26. Shi Y, Sha F (2012) Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In: Proceedings of the 29th international conference on machine learning, pp 1275–1282
27. Shao M, Kit D, Fu Y (2014) Generalized transfer subspace learning through low-rank constraint. Int J Comput Vision 109(1–2):74–93
28. Pan SJ, Kwok JT, Yang Q (2008) Transfer learning via dimensionality reduction. In: Proceedings of the 23rd national conference on artificial intelligence, pp 677–682
29. Lanckriet GR, Cristianini N, Bartlett P et al (2004) Learning the kernel matrix with semidefinite programming. Mach Learn Res 5:27–72
30. Song L, Smola A, Gretton A, Bedo J et al (2012) Feature selection via dependence maximization. Mach Learn Res 13:1393–1434
31. Gretton A, Bousquet O, Smola A, et al (2005) Measuring statistical dependence with Hilbert-Schmidt norms. In: Conference algorithmic learning theory, Berlin, Heidelberg, pp 63–77
32. Yan K, Kou L, Zhang D (2018) Learning domain-invariant subspace using domain features and independence maximization. IEEE Trans Cybern 48(1):288–299
33. Schölkopf B, Smola A, Müller KR (1998) Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput 10(5):1299–1319
34. Horn BK, Schunck BG (1981) Determining optical flow. Artif Intell 17(1–3):185–203
35. Beauchemin SS, Barron JL (1995) The computation of optical flow. ACM Comput Surv (CSUR) 27(3):433–466