Exploiting Web Images for Event Recognition in Consumer Videos: A Multiple Source Domain Adaptation Approach

Lixin Duan Dong Xu
Nanyang Technological University
50 Nanyang Avenue, Singapore
{S080003,dongxu}@ntu.edu.sg

Shih-Fu Chang
Columbia University
New York, New York, 10027, USA
sfchang@ee.columbia.edu

Abstract

*Recent work has demonstrated the effectiveness of domain adaptation methods for computer vision applications. In this work, we propose a new multiple source domain adaptation method called Domain Selection Machine (DSM) for event recognition in consumer videos by leveraging a large number of loosely labeled web images from different sources (*e.g.*, Flickr.com and Photosig.com), in which there are no labeled consumer videos. Specifically, we first train a set of SVM classifiers (referred to as source classifiers) by using the SIFT features of web images from different source domains. We propose a new parametric target decision function to effectively integrate the static SIFT features from web images/video keyframes and the space-time (ST) features from consumer videos. In order to select the most relevant source domains, we further introduce a new data-dependent regularizer into the objective of Support Vector Regression (SVR) using the ϵ-insensitive loss, which enforces the target classifier shares similar decision values on the unlabeled consumer videos with the selected source classifiers. Moreover, we develop an alternating optimization algorithm to iteratively solve the target decision function and a domain selection vector which indicates the most relevant source domains. Extensive experiments on three real-world datasets demonstrate the effectiveness of our proposed method DSM over the state-of-the-art by a performance gain up to 46.41%.*

1. Introduction

Event recognition in consumer videos is an important research topic of computer vision, because of its broad applications in video indexing and retrieval. Unlike the videos in the KTH dataset (see [16] for the recent research progress), the highly unconstrained consumer videos may contain significant camera motion and large intra-class variations [11, 18], making event recognition in consumer videos an extremely challenging task.

Loui *et al.* [20] and Jiang *et al.* [15] collected two benchmark datasets for consumer videos. Liu *et al*. [18] used AdaBoost to integrate different features, and Ikizler-Cinbis and Sclaroff [14] proposed to exploit a multiple instance learning (MIL) approach to fuse multiple features from objects, scenes and people. Recently, Wang *et al.* [29] proposed to use dense trajectories to describe videos, which also leads to improved action recognition results for YouTube videos.

The above event recognition algorithms [13, 18, 29] followed the conventional framework, in which a large corpus of training data is required to learn the robust classifiers with the event class labels determined through time consuming and expensive human annotation. Given sufficient labeled training videos, these methods have demonstrated promising results. However, it is well-known that users are generally reluctant to annotate abundant consumer videos. When there are only few labeled videos available, the learned classifiers using these methods cannot generalize well, which may significantly degrade the event recognition performances.

Recently, Duan *et al.* [11] developed a new event recognition approach for consumer videos by using web videos from YouTube, in which a domain adaptation method is developed to explicitly handle the mismatch between data distributions of two domains (*i.e.*, web video domain and consumer video domain). In [13], Ikizler-Cinbis *et al.* employed web images for action recognition. However, their work did not cope with the feature distribution mismatch between different domains and cannot distinguish actions like "sitting_down" and "standing_up", because temporal information is not exploited in their image-based model.

In this work, we develop a new multiple source domain adaptation method called Domain Selection Machine (DSM) for event recognition in consumer videos by leveraging loosely labeled web images from different sources (*e.g.*, Flickr.com and Photosig.com), which is illustrated in Figure 1. Our work is motivated by [11, 13] and other recent domain adaptation work (see Section 2 for more details) and also based on the observation that there are much more web

images than videos available with loose labels.

We first learn an SVM classifier (referred to as source classifier) by using the SIFT features of web images from one source domain, which are retrieved by conducting keyword based search with an event name (*e.g.*, "show") as the textual query. For each video, we extract the space-time (ST) feature and also a set of static SIFT features from multiple keyframes, as suggested in [18, 11]. Each learned SVM classifier is then applied to multiple keyframes of each video to generate the decision values, which are further averaged as the prediction of this video. Recall that most domain adaptation methods [7, 8, 10, 11, 12, 26, 30] cannot work when the samples from different domains are represented by using different types of features. In our application, the videos in the target domain can be additionally represented by another type of features (*i.e.*, the ST features). In order to effectively integrate the static SIFT features from web images/video keyframes and the ST features from videos in this application, we propose a new parametric target decision function which is adapted from a weighted combination of the selected classifiers with the adaption error modeled by using the ST features.

It is well-known that irrelevant source domains may be harmful for the classification performances in the target domain (*i.e.*, the so-called negative transfer in [24]). In order to select the most relevant source domains, we additionally introduce a new data-dependent regularizer into the objective of Support Vector Regression (SVR) using the ϵ-insensitive loss. This new regularizer is based on the smoothness assumption which enforces the decision values on unlabeled consumer videos from the target classifier to be similar to the predictions from the selected source classifiers. Moreover, an alternating optimization algorithm is also developed, in which we iteratively solve for the target decision function and a domain selection vector which indicates the most relevant source domains. We conduct extensive experiments using three real-world datasets. Promising results clearly demonstrate the effectiveness of DSM.

The main contributions of this paper include: 1) We present the first domain adaptation method called DSM to take advantage of abundant freely available web images for event recognition in consumer videos. 2) We integrate different types of features from different domains by using a newly proposed target decision function in DSM. 3) With the newly introduced data-dependent regularizer, DSM can automatically select the most relevant source domains.

2. Related Work

Domain adaptation (*a.k.a.*, cross-domain learning or transfer learning) methods have been used for different applications [22]. A few SVM based approaches [3, 7, 10, 11, 30] were recently developed. In [10, 11], Duan *et al.* proposed to simultaneously learn the optimal linear combination of base kernels and the target SVM classifier by minimizing a regularized structural risk functional, in which the subsequent work in [11] additionally used the pre-learned classifiers as the prior. Bruzzone and Marconcini [3] proposed Domain Adaptation Support Vector Machine (DASVM) to iteratively learn the target classifier by labeling the unlabeled target samples and simultaneously removing some labeled samples in the source domain.

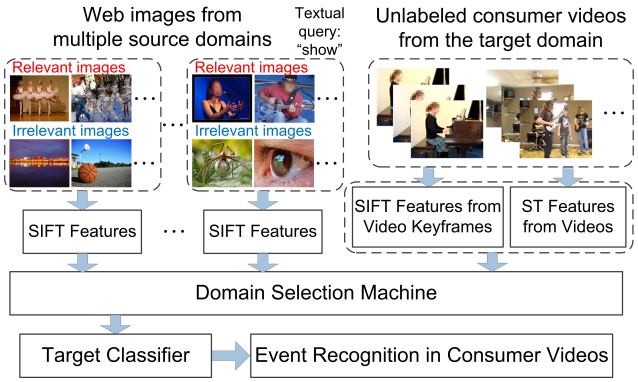

Figure 1. Illustration of our proposed method Domain Selection Machine (DSM) for event recognition in consumer videos.

Saenko *et al.* [25] proposed a metric learning method to enforce the intra-class samples from two domains become closer with each other, which is further extended in [17] by using an asymmetric kernel transformation. Moreover, their work [17, 25] can be employed for knowledge transfer by transferring the information learned from existing categories to unseen categories. Other knowledge transfer methods [1, 23, 28] assumed that the samples come from one domain only. Based on the Grassmann manifold assumption, Gopalan *et al.* [12] inferred the intermediate subspaces for domain adaptation by representing the data points in two domains as two subspaces.

The above methods [3, 7, 10, 11, 12, 17, 25, 30] mainly focus on the single source domain setting. When the training data come from different sources, researchers proposed multiple source domain adaptation methods [8, 9, 26]. While different weights may be assigned to different source domains as in [5, 9], these methods [8, 9, 26] employed all the source domains for domain adaptation. However, it is crucial to select the most relevant source domains for domain adaptation, because irrelevant source domains may be harmful for the classification performances in the target domain, which is known as negative transfer [24].

In this work, we develop a new multiple source domain adaptation method called Domain Selection Machine to automatically determine the most relevant source domains for domain adaption without requiring any labeled data from the target domain. Moreover, most existing domain adaptation methods [7, 8, 10, 11, 12, 26, 30] assumed that the data in the source and target domains share the same type of features. In contrast, this work employs the SIFT features of web images in the source domains to classify consumer

videos in the target domain, in which the ST features of videos can be additionally integrated.

3. Domain Selection Machine

In this work, we refer to the loosely labeled web images from different sources as multiple source domains and the consumer videos as the target domain. Our goal is to learn a robust classifier for the target domain (referred to as target classifier) by leveraging web images from multiple source domains, where there are no labeled training data in the target domain.

Let $\mathcal{D}^T$ be the target domain consisting of unlabeled samples $\mathbf{x}_1^T, \ldots, \mathbf{x}_m^T$ and $\mathcal{D}^s = \{(\mathbf{x}_i^s, y_i^s)|_{i=1}^{n_s}\}$ be the s-th source domain, where y_i^s is the label of $\mathbf{x}_i^s$, $s = 1, \ldots, S$, and S is the total number of source domains. We denote the transpose of vector/matrix by using the superscript $'$. We also define $\mathbf{0}_n, \mathbf{1}_n \in \mathbb{R}^n$ as the column vectors of all zeros and all ones, respectively. The inequality $\mathbf{u} = [u_1, \ldots, u_n]' \geq \mathbf{v} = [v_1, \ldots, v_n]'$ means that $u_i \geq v_i$ for $i = 1, \ldots, n$. Moreover, let us denote $\circ$ as the element-wise product between two vectors, *i.e.*, $\mathbf{u} \circ \mathbf{v} = [u_1 v_1, \ldots, u_n v_n]'$.

3.1. Motivation

Our work Domain Selection Machine (DSM) is motivated from two aspects.

Data-dependent regularizer for source domain selection: In Domain Adaptation Machine (DAM) [9], Duan *et al*. introduced a regularizer for multiple source domain adaptation, in which the decision values of unlabeled target samples $\mathbf{x}_i^T|_{i=1}^m$ from the target classifier should be similar to those from the source classifiers. Specifically, the regularizer in [9] is defined as follows:

$$\Omega(f) = \frac{1}{2} \sum_{s=1}^{S} \gamma_s \sum_{i=1}^{m} \left(f^T(\mathbf{x}_i^T) - f^s(\mathbf{x}_i^T) \right)^2, \qquad (1)$$

where $f^T(\mathbf{x}_i^T)$ is the decision value of $\mathbf{x}_i^T$ from the target classifier, $f^s(\mathbf{x}_i^T)$ is the decision value of $\mathbf{x}_i^T$ from the s-the source classifier, and γ_s is a pre-defined weight for measuring the relevance between the s-th source domain and the target domain. Ideally, we should enforce $f^s(\mathbf{x}_i^T)$ to be close to $f^T(\mathbf{x}_i^T)$ if two domains are relevant (*i.e.*, γ_s is large). However, it is a non-trivial task to effectively measure the relevance between two domains. In [9], the so-called Maximum Margin Discrepancy (MMD) criterion proposed in [2] is used to define γ_s. Based on the manifold assumption, Chattopadhyay *et al*. proposed to learn the optimal γ_s in [5].

In this work, we argue that it is more beneficial to choose a few relevant source domains rather than use all the source domains as in [5, 9] for multiple source domain adaptation. To this end, we propose a new data-dependant regularizer for source domain selection:

$$\Omega(f) = \frac{1}{2} \sum_{s=1}^{S} d_s \sum_{i=1}^{m} \left(f^T(\mathbf{x}_i^T) - f^s(\mathbf{x}_i^T) \right)^2, \qquad (2)$$

where $d_s \in \{0, 1\}$ is a domain selection indicator for the s-th source domain. While our proposed data-dependent regularizer is similar to (1), it is intrinsically different from (1) because of the domain selection indicator d_s introduced in this work. After the learning process, we expect $d_s = 1$, if the s-th source domain is relevant to the target domain; and $d_s = 0$, otherwise.

Integrating SIFT and ST features in the target classifier: In this work, we extract SIFT features [21] from web images in the source domains. For consumer videos in the target domain, we extract the SIFT features from the video keyframes as well as space-time (ST) features like HOG and HOF [29], because both types of features have been demonstrated to be useful for event recognition in videos [11, 18].

Most existing domain adaptation methods [7, 8, 10, 11, 12, 26, 30] assume that the samples in the source and target domains are represented by the same type of features with the same dimension. In this work, the samples from the target domain can be additionally represented by another type of features (*i.e.*, the ST features). Motivated by the existing work [11], we propose a new parametric target decision function for any video $\mathbf{x}$, which is adapted from a weighted combination of the selected source classifiers with the adaption error modeled by using the ST features, namely:

$$\begin{aligned} f(\mathbf{x}) &= f_{\text{2D}}(\mathbf{x}) + f_{\text{3D}}(\mathbf{x}) \\ &= \sum_{s=1}^{S} d_s \beta_s f^s(\mathbf{x}) + \mathbf{w}'\varphi(\mathbf{x}) + b, \end{aligned} \qquad (3)$$

where $f_{\text{2D}}(\mathbf{x}) = \sum_{s=1}^{S} d_s \beta_s f^s(\mathbf{x})$ is a weighted combination of source classifiers f^s's based on SIFT features, β_s is a real-valued weight for the s-th source domain; $f_{\text{3D}}(\mathbf{x}) = \mathbf{w}'\varphi(\mathbf{x}) + b$ is the adaptation error function modeled by using the ST features, $\varphi(\cdot)$ is a feature mapping function that maps $\mathbf{x}$ into $\varphi(\mathbf{x})$, $\mathbf{w}$ is a weight vector, and b is a bias term. Note that each $f^s(\mathbf{x})$ represents the averaged decision values of all video keyfames from the video $\mathbf{x}$. For our method DSM, we will learn the domain selection indicator d_s and the weight β_s for each source classifier in $f_{\text{2D}}(\mathbf{x})$, as well as learn the parameters $\mathbf{w}$ and b in $f_{\text{3D}}(\mathbf{x})$.

Note that the target decision function in DAM [9] is in the form of $f_{\text{3D}}(\mathbf{x}) = \mathbf{w}'\varphi(\mathbf{x}) + b$, in which the pre-learned source classifiers are not employed. In order to clearly demonstrate the effectiveness of the newly proposed data-dependent regularizer (2) for source domain selection, in the experiments (see Sections 4.2 and 4.3), we further compare DAM [9] with a simplified version called DSM_{sim} in

which we also model the target decision function as $f(\mathbf{x}) = f_{3D}(\mathbf{x})$. Note that the optimization algorithm that will be introduced to solve DSM in Section 3.3 is still applicable for DSM_{sim} after some minor modifications. It is worth mentioning that the only difference between DSM_{sim} and DAM is that DSM_{sim} uses the newly proposed data-dependent regularizer in (2) (*i.e.*, the sparse binary selections of source domains), while DAM uses the data-dependent regularizer in (1) (*i.e.*, the non-sparse continuous selection weights).

In this work, we use the ST features in $f_{3D}(\mathbf{x})$ for DAM [9], DSM_{sim} and DSM, because of their effectiveness for event recognition as shown in [29].

3.2. Proposed formulation of DSM

Let us define $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_S]'$ as the weight vector for source classifiers and $\mathbf{d} = [d_1, \ldots, d_S]'$ as the domain selection vector for source domains. We then formally present the objective of our method DSM as follows:

$$\min_{\mathbf{d},\mathbf{w},b,\boldsymbol{\beta},\mathbf{f}^T} \frac{1}{2}\left(\|\mathbf{w}\|^2+\lambda\|\boldsymbol{\beta}\|^2\right)+C\sum_{i=1}^{m}\ell_\epsilon(f^T(\mathbf{x}_i^T)-f(\mathbf{x}_i^T)) + \frac{\theta}{2}\sum_{s=1}^{S} d_s \sum_{i=1}^{m}(f^T(\mathbf{x}_i^T) - f^s(\mathbf{x}_i^T))^2, \quad (4)$$

$$\text{s.t.} \quad \sum_{s=1}^{S} d_s \geq 1,\ d_s \in \{0,1\}, \quad (5)$$

where, $\ell_\epsilon(\cdot)$ is the ϵ-insensitive loss function[1], $\mathbf{f}^T = [f^T(\mathbf{x}_1^T), \ldots, f^T(\mathbf{x}_m^T)]'$ is a vector of decision values of unlabeled target samples from the target classifier, and $\lambda, \theta, C > 0$ are the regularization parameters.

Note that $\mathbf{f}^T$ can be considered as an intermediate variable which represents the regression values for the parametric form of the target classifier $f(\mathbf{x})$ in the ϵ-insensitive loss (see (4)). And it can be used for transductive inference (see Section 3.3 for the derived solution). We will show that with this intermediate variable, our proposed optimization problem in (4) with fixed $\mathbf{d}$ can be transformed into an SVR-like optimization problem which can thus be effectively solved by using the existing toolboxes such as LIBSVM [4].

Let us represent the feasible set of $\mathbf{d}$ as $\mathcal{M} = \{\mathbf{d} | \mathbf{1}_S'\mathbf{d} \geq 1, \mathbf{d} \in \{0,1\}^S\}$. We define $\tilde{\mathbf{w}} = [\mathbf{w}', \sqrt{\lambda}\boldsymbol{\beta}']'$ and $\tilde{\varphi}(\mathbf{x}_i^T) = [\varphi'(\mathbf{x}_i^T), \frac{1}{\sqrt{\lambda}}(\mathbf{d} \circ \boldsymbol{f}(\mathbf{x}_i^T))']'$, where $\boldsymbol{f}(\mathbf{x}_i^T) = [f^1(\mathbf{x}_i^T), \ldots, f^S(\mathbf{x}_i^T)]'$. Then, our parametric form of the target classifier in (3) can be rewritten as $f(\mathbf{x}) = \tilde{\mathbf{w}}'\tilde{\varphi}(\mathbf{x}) + b$. We also define $\mathbf{f}^s = [f^s(\mathbf{x}_1^T), \ldots, f^s(\mathbf{x}_m^T)]'$ as a vector of decision values of unlabeled target samples obtained from the source classifier f^s. Since the ϵ-insensitive loss ℓ_ϵ is non-smooth, we then transform the loss on the unlabeled target samples $\mathbf{x}_i^T$'s in (4) into constraints in which the slack variables ξ_i's and ξ_i^*'s are also introduced. Then, we rewrite the optimization problem in (4) as follows:

$$\min_{\substack{\mathbf{d}\in\mathcal{M},\tilde{\mathbf{w}},\\ b,\mathbf{f}^T,\boldsymbol{\xi},\boldsymbol{\xi}^*}} \frac{1}{2}\|\tilde{\mathbf{w}}\|^2+\frac{\theta}{2}\sum_{s=1}^{S} d_s\left\|\mathbf{f}^T-\mathbf{f}^s\right\|^2+C\sum_{i=1}^{m}(\xi_i+\xi_i^*), \quad (6)$$

$$\text{s.t.} \quad \tilde{\mathbf{w}}'\tilde{\varphi}(\mathbf{x}_i^T) + b - f^T(\mathbf{x}_i^T) \leq \epsilon + \xi_i,\ \xi_i \geq 0; \quad (7)$$

$$f^T(\mathbf{x}_i^T) - \tilde{\mathbf{w}}'\tilde{\varphi}(\mathbf{x}_i^T) - b \leq \epsilon + \xi_i^*,\ \xi_i^* \geq 0, \quad (8)$$

3.3. Detailed solution

Note that the optimization problem in (6) is a mixed integer programming problem. To the best of our knowledge, there is no closed-form solution. Before presenting our developed algorithm, let us first obtain the Lagrangian of (6) with respect to the primal variables (*i.e.*, $\tilde{\mathbf{w}}, b, \mathbf{f}^T, \xi_i$ and ξ_i^*) by introducing the dual variables α_i's and α_i^*'s for the constraints in (7) and (8), respectively. Taking the derivatives of the Lagrangian with respect to $\tilde{\mathbf{w}}, b, \mathbf{f}^T, \xi_i$ and ξ_i^* and setting them to zeros, respectively, we obtain the Karush-Kuhn-Tucker (KKT) conditions as: $\mathbf{f}^T = \frac{1}{\sum_{s=1}^{S} d_s}\left(\sum_{s=1}^{S} d_s\mathbf{f}^s + \frac{\boldsymbol{\alpha}-\boldsymbol{\alpha}^*}{\theta}\right)$, $\tilde{\mathbf{w}} = -\sum_{i=1}^{m}(\alpha_i - \alpha_i^*)\tilde{\varphi}(\mathbf{x}_i^T)$, $\mathbf{1}_m'\boldsymbol{\alpha} = \mathbf{1}_m'\boldsymbol{\alpha}^*$, and $\mathbf{0}_m \leq \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \leq C\mathbf{1}_m$, where $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]'$, $\boldsymbol{\alpha}^* = [\alpha_1^*, \ldots, \alpha_m^*]'$. Based on the KKT conditions, (6) can be converted into the following optimization problem:

$$\min_{\mathbf{d}\in\mathcal{M}} h(\mathbf{d}) = g(\mathbf{d}) + \frac{\theta}{2}\left(\mathbf{u}'\mathbf{d} - \frac{\mathbf{d}'\mathbf{F}'\mathbf{F}\mathbf{d}}{\mathbf{1}_S'\mathbf{d}}\right), \quad (9)$$

where $\mathbf{u} = [u_1, \ldots, u_S]'$ is a vector with each entry as $u_s = \|\mathbf{f}^s\|^2$, $\mathbf{F} = [\mathbf{f}^1, \ldots, \mathbf{f}^S]$ is a matrix, and $g(\mathbf{d})$ is solved by

$$g(\mathbf{d}) = \max_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} -\frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)'\hat{\mathbf{K}}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)-\hat{\mathbf{y}}'(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) -\epsilon\mathbf{1}_m'(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*), \quad (10)$$
$$\text{s.t.} \quad \mathbf{1}_m'\boldsymbol{\alpha} = \mathbf{1}_m'\boldsymbol{\alpha}^*,\ \mathbf{0}_m \leq \boldsymbol{\alpha}, \boldsymbol{\alpha}^* \leq C\mathbf{1}_m,$$

where $\hat{\mathbf{K}}$ is a newly obtained kernel matrix with each entry as $\hat{\mathbf{K}}(\mathbf{x}_i^T, \mathbf{x}_j^T) = \tilde{\varphi}(\mathbf{x}_i^T)'\tilde{\varphi}(\mathbf{x}_j^T) + \frac{1}{\theta\mathbf{1}_S'\mathbf{d}}\delta_{ij} = \mathbf{K}(\mathbf{x}_i^T, \mathbf{x}_j^T) + \frac{1}{\lambda}\sum_{s=1}^{S} d_s f^s(\mathbf{x}_i^T)f^s(\mathbf{x}_j^T) + \frac{1}{\theta\mathbf{1}_S'\mathbf{d}}\delta_{ij}$, and $\hat{\mathbf{y}} = [\hat{y}_1, \ldots, \hat{y}_m]'$ is a vector of newly obtained regression values with each entry as $\hat{y}_i = \frac{\sum_{s=1}^{S} d_s f^s(x_i^T)}{\mathbf{1}_S'\mathbf{d}}$. Note that $\mathbf{K}(\mathbf{x}_i^T, \mathbf{x}_j^T) = \varphi'(\mathbf{x}_i^T)\varphi(\mathbf{x}_j^T)$. Moreover, $\delta_{ij} = 1$, if $i = j$; and $\delta_{ij} = 0$, otherwise.

In this work, we develop an effective alternating optimization algorithm to solve (9) by iteratively updating the domain selection vector $\mathbf{d}$ and the dual variables $\boldsymbol{\alpha}, \boldsymbol{\alpha}^*$.

Updating $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$: Note that when $\mathbf{d}$ is fixed in (9), we solve the optimization problem in (10) for $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$. Observing that (10) is analogous to the dual problem of Support Vector Regression [27], we directly use the existing

[1] $\ell_\epsilon(a) = \begin{cases} |a| - \epsilon, & \text{if } |a| > \epsilon; \\ 0, & \text{otherwise.} \end{cases}$

Algorithm 1 Domain Selection Machine

Input: unlabeled target training samples $\{\mathbf{x}_i^T|_{i=1}^m\}$, source classifiers $\{f^s(\mathbf{x})|_{s=1}^S\}$ and the feasible set $\mathcal{M}$
Initialization: $t \leftarrow 1$ and $\mathbf{d}_t \leftarrow \mathbf{1}_S$
Solve for $\boldsymbol{\alpha}_t, \boldsymbol{\alpha}_t^*$ in (10)
While $t \leq T_{\max}$ **Do**
  $k \leftarrow 1$
  Calculate $h(\mathbf{d}^{(j)})$'s in (11) by using $\boldsymbol{\alpha}_t$ and $\boldsymbol{\alpha}_t^*$
  Sort $h(\mathbf{d}^{(j)})$'s as $h(\mathbf{d}^{(1)}) \leq \cdots \leq h(\mathbf{d}^{(|\mathcal{M}|)})$
  While $k \leq |\mathcal{M}|$ **Do**
    $\mathbf{d}_{t+1} \leftarrow \mathbf{d}^{(k)}$ and solve for $\boldsymbol{\alpha}^{(k)}, \boldsymbol{\alpha}^{*(k)}$ in (10)
    If (9) decreases **then** break
    $k \leftarrow k + 1$
  End While
  If $k = |\mathcal{M}| + 1$ **then** break
  $\boldsymbol{\alpha}_{t+1} \leftarrow \boldsymbol{\alpha}^{(k)}$ and $\boldsymbol{\alpha}_{t+1}^* \leftarrow \boldsymbol{\alpha}^{*(k)}$
  $t \leftarrow t + 1$
End While
Output: $\mathbf{d}_t$, $\boldsymbol{\alpha}_t$ and $\boldsymbol{\alpha}_t^*$

solver LIBSVM [4] to solve (10) by using the newly obtained kernel $\hat{\mathbf{K}}$ and regression value $\hat{\mathbf{y}}$.

Updating d: When $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ are obtained after solving the optimization problem in (10), we fix them and consequently the optimization problem in (9) can be rewritten as the following integer programming problem:

$$\min_{\mathbf{d}\in\mathcal{M}} h(\mathbf{d}) = \min_{\mathbf{d}\in\mathcal{M}} \mathbf{p}'\mathbf{d} - \frac{r + 2\theta\mathbf{q}'\mathbf{d} + \theta^2\mathbf{d}'\mathbf{F}'\mathbf{F}\mathbf{d}}{2\theta\mathbf{1}_S'\mathbf{d}}, \quad (11)$$

where $\mathbf{p} = [p_1, \ldots, p_S]'$ is a vector with each entry as $p_s = \frac{\theta}{2}\|\mathbf{f}^s\|^2 - \frac{1}{2\lambda}\|(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)'\mathbf{f}^s\|^2$, $r = \|\boldsymbol{\alpha} - \boldsymbol{\alpha}^*\|^2$ is a scalar, $\mathbf{q} = [q_1, \ldots, q_S]'$ is a vector with each entry as $q_s = (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)'\mathbf{f}^s$. In our experiments, considering we only have a limited number of source domains (we set $S = 10$), we enumerate all possible candidates of $\mathbf{d} \in \mathcal{M}$ in order to find the exact solution to the integer programming problem in (11).

It is worth mentioning that the optimization problem in (9) may not even converge, if we simply solve the subproblems in (10) and (11) alternatively. To avoid such an issue, we update $\mathbf{d}$ in a line-search-like fashion to enforce the convergence of (9). Specifically, we first initialize $\mathbf{d}$ as $\mathbf{1}_S$ and then update $\boldsymbol{\alpha}, \boldsymbol{\alpha}^*$ by solving the optimal solution to (10). When updating $\mathbf{d}$, we sort the values of objective function $h(\mathbf{d})$ in (11) with different $\mathbf{d}$ as $h(\mathbf{d}^{(1)}) \leq \cdots \leq h(\mathbf{d}^{(|\mathcal{M}|)})$, where $\mathbf{d}^{(j)}$'s are the candidate solutions of $\mathbf{d}$ and $|\mathcal{M}|$ represents the cardinality of $\mathcal{M}$. Following the order of $h(\mathbf{d}^{(j)})$, we then use each candidate $\mathbf{d}^{(j)}$ one by one to solve (10) and also obtain the objective value of (9). We terminate this process when the objective value of (9) decreases by using $\mathbf{d}^{(k)}$, and set $\mathbf{d}$ as the current candidate (*i.e.*, $\mathbf{d}^{(k)}$) for updating.

The above alternating optimization algorithm will repeat for a few iterations until the the objective value of (9) stops decreasing or the maximal number of iterations reaches. We summarize the whole optimization procedure for DSM in Algorithm 1.

With the learned dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$, we use the following target decision function to predict any new test sample $\mathbf{x}$ as: $f_{\text{3D}}(\mathbf{x}) = \sum_{i=1}^m (\alpha_i - \alpha_i^*)\mathbf{K}(\mathbf{x}_i^T, \mathbf{x}) + b$. Note that we do not use the source classifiers here, because the information from the source domains has already been transferred when learning the optimal $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$. Moreover, the prediction process will also become much faster.

4. Experiments

In the experiments, we evaluate different methods using three real-world consumer video datasets. Specifically, we compare our proposed method DSM with the standard SVM, Domain Adaptation SVM (DASVM) [3] and several existing multiple source domain adaptation methods Multiple Kernel Mean Matching (Multi-KMM) [26], Domain Adaptation Machine (DAM) [9] and Conditional Probability based Multi-source Domain Adaptation (CP-MDA) [5], as these methods can work when there are no labeled training data in the target domain.

We also report the results of the simplified version of DSM (referred to as DSM_{sim}) that has been discussed in Section 3.1. We do not compare DSM with the multiple source domain adaptation method proposed in [8], because it requires the labeled training data from the target domain.

4.1. Datasets and Features

The first web image dataset is the NUS-WIDE dataset [6] which consists of 269,648 images downloaded from Flickr. The second web image dataset is collected from the photo forum called photosig.com and it contains about 1.3 million images. The images in both datasets are associated with surrounding textual descriptions (*e.g.*, title, tags, etc.) provied by the users. These two large-scale web image datasets are combined into one training database from which multiple source domains can be constructed (see Section 4.2 for more details). For each image, we extract 128-dimensional SIFT features from salient regions detected by the Difference of Gaussians (DoG) detector [21].

We use the following three real-world consumer video datasets as the test datasets for performance evaluation.

Kodak dataset: This dataset was used in [11] which contains 195 consumer videos with their ground truth labels of six event classes (*e.g.*, "birthday", "picnic", "parade", "show", "sports" and "wedding").

YouTube dataset: Following [11], we collect another consumer video dataset from YouTube using the same six event classes as in the Kodak dataset. Annotators manually checked the downloaded YouTube videos, and we finally

obtain 561 videos from six event classes for performance evaluation.

CCV dataset: It is a newly released consumer video dataset collected by Columbia University [15]. It contains a training set of 4,659 videos and a test set of 4,658 videos which are annotated to 20 semantic categories. Since our work focuses on event recognition, we do not consider the non-event categories (*i.e.*, "playground", "bird", "beach", "cat" and "dog"). In order to facilitate the keyword based search in the image database to collect training data, we merge "wedding ceremony", "wedding reception" and "wedding dance" as "wedding" and also merge "non-music performance" and "music performance" as "performance". Considering there are only a few hundreds of training images for some sports categories (*e.g.*, "biking"), we additionally merge different types of sports like "baseball", "basketball", "biking", "ice skating", "skiing", "soccer" and "swimming" into a single category "sports". Finally, we evaluate different algorithms using 2726 videos from the six event classes (*i.e.*, "birthday", "graduation", "parade", "performance", "sports" and "wedding").

For each video, we sample one keyframe per two seconds. For all the sampled keyframes, we extract the static SIFT features by using the DoG detector [21]. Moreover, space-time (ST) features are also extracted from each video. Specifically, for the videos in the Kodak and YouTube datasets, we extract three types of space-time features: 96-dimensional Histograms of Oriented Gradients (HOG), 108-dimensional Histograms of Optical Flow (HOF) and 192-dimensional Motion Boundary Histogram (MBH) by using the software from the recent work [29], in which we set the trajectory length as 50, the sampling stride as 16 and other parameters as their default values. For the videos in the CCV dataset, we use the 144-dimensional 3D Space-Time Interest Point (STIP) feature and one additional audio feature called Mel-Frequency Cepstral Coefficients (MFCC) provided in [15].

4.2. Experimental setup

In the experiments, we use the bag-of-words representations for SIFT features and ST features, respectively. Specifically, we randomly select one hundred thousand training images from the training image database and partition the corresponding SIFT features into 4000 clusters by using k-means. These 4000 clusters are considered as a visual codebook, and each image/video keyframe is then represented as a 4000-dimensional token frequency (TF) features by quantizing its SIFT features with respect to the visual codebook. We perform k-means clustering twice by using different randomly selected training images to construct two visual codebooks for the generation of multiple source domains. For the videos in the Kodak and YouTube datasets, we similarly group each type of ST features into 2000 clusters by using k-means as well. Finally, we represent each video as a 2000-dimensional TF features for each type of ST features. Moreover, for each video in the CCV dataset, we directly use the 5000-dimensional and 4000-dimensional features provided by the authors [15] based on the 3D STIP features and MFCC features, respectively.

Using a given event name as the textual query (*e.g.*, "show"), we first search for a set of relevant images which are associated with the textual query and randomly choose the same number of irrelevant images which are not associated with the textual query from our training image database, as suggested in [19]. After that, we construct five source domains for each SIFT feature based visual codebook by randomly sampling 100 relevant images and 100 irrelevant images for five times. Then, we have ten source domains in total (*i.e.*, $S = 10$) from two visual codebooks. In this work, the relevant images are used as positive samples and the irrelevant images are considered as negative samples. Our initial experiments show that the classification results using SVM with the training images from different source domains are different, making it suitable for evaluating our domain selection method DSM.

For fair comparison, we fix the regularization parameter C as 1 for all methods. For images/video keyframes with SIFT features, we use the non-linear χ^2 kernel, *i.e.*, $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{A} D(\mathbf{x}_i, \mathbf{x}_j)\right)$, where $D(\mathbf{x}_i, \mathbf{x}_j)$ is the χ^2 distance between two image samples $\mathbf{x}_i$ and $\mathbf{x}_j$, and A is the mean value of square distances between all pairs of training samples. For videos, we also use the χ^2 kernel by combining different types of ST features or the audio feature using the method in [29] as $\mathbf{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\sum_c \frac{1}{A_c} D(\mathbf{x}_i^c, \mathbf{x}_j^c)\right)$, where $D(\mathbf{x}_i^c, \mathbf{x}_j^c)$ is the χ^2 distance between two videos $\mathbf{x}_i$ and $\mathbf{x}_j$ using the c-th type of features, and A_c's are similarly defined as for A.

Note that the standard SVM, DASVM and Multi-KMM can only handle the domain adaptation problem, when the data from the source and target domains are with the same type of features. Therefore, they can only use the static SIFT features to learn classifiers for the target domain. Recall that in this work, we do not have any labeled videos or keyframes in the target domain. We then learn the standard SVM by using the labeled training data only from the source/auxiliary domains, which is referred to as SVM_A here. Specifically for SVM_A, we learn one SVM classifier f^s by using the training images from the s-th source domain, and then we obtain the final SVM_A classifier by equally fusing all the source classifier f^s's. Considering that DASVM is a semi-supervised learning method and cannot handle the multiple source domain setting, we train one DASVM classifier by using the labeled images from one source domain and the keyframes of unlabeled test videos from the target domain. Similar to SVM_A, we fuse the DASVM classifiers from multiple source domains to obtain

Table 1. Mean Average Precisions (MAPs) of all methods on the Kodak dataset.

	SVM_A	DASVM	Multi-KMM	DAM	CP-MDA	DSM_{sim}	DSM
MAP	27.95%	25.68%	24.22%	27.66%	24.41%	33.67%	**35.46%**

Table 2. Mean Average Precisions (MAPs) of all methods on the YouTube dataset.

	SVM_A	DASVM	Multi-KMM	DAM	CP-MDA	DSM_{sim}	DSM
MAP	31.17%	29.40%	31.98%	32.58%	30.27%	33.75%	**35.26%**

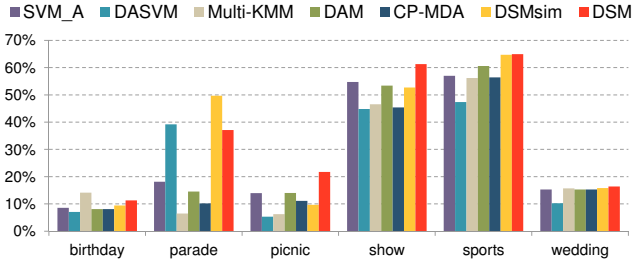

Figure 2. Per-event Average Precisions (APs) of all methods on the Kodak dataset.

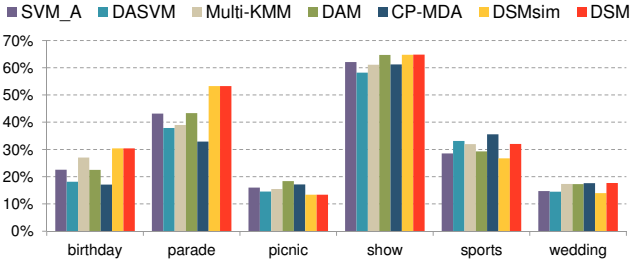

Figure 3. Per-event Average Precisions (APs) of all methods on the YouTube dataset.

the final DASVM classifier. For Multi-KMM, the labeled images from all source domains and unlabeled keyframes of test videos are used for training. Since SVM_A, DASVM and Multi-KMM can only predict video keyframes based on SIFT features, we obtain the final prediction for each test video by averaging the decision values of all its keyframes for each of the three methods. For DAM, CP-MDA, and our methods DSM_{sim} and DSM, we average the decision values of the video keyframes by using pre-learned SVM classifier f^s's to generate the prediction for each target video x_i^T (*i.e.*, $f^s(\mathbf{x}_i^T)$ in (4)) and the training images are not used when learning the target classifiers. As DAM, CP-MDA, DSM_{sim} and DSM can make use of the unlabeled samples for model learning, we then consider all the test videos from the target domain as the unlabeled training samples for all the four methods.

For performance evaluations, we use the non-interpolated Average Precision (AP) as in [11] and define Mean Average Precision (MAP) as the mean of APs over all event classes.

4.3. Comparisons of different methods

We plot the per-event APs of all methods on the Kodak, YouTube and CCV datasets in Figures 2, 3 and 4, respectively. We also show the MAPs of all methods on the three datasets in Tables 1, 2 and 3, respectively. From the results, we have the following observations:

1) There is no consistent winner among SVM_A and the existing domain adaptation methods DASVM, Multi-KMM, DAM and CP-MDA in terms of MAPs. On the Kodak dataset, SVM_A is better than the existing domain adaptation methods DASVM, Multi-KMM, DAM and CP-MDA in terms of MAPs, which indicates that there may exist some irrelevant source domains which cannot be well handled by the four domain adaptation methods. Therefore, the four domain adaptation methods cannot learn good target classifiers, which is known as negative transfer [24]. In contrast, on the YouTube dataset (*resp.*, the CCV dataset), Multi-KMM and DAM (*resp.*, Multi-KMM, DASVM and CP-MDA) outperform SVM_A in terms of MAPs, which demonstrates that the domain adaptation methods can successfully make use of the data from the source domains to help learn better classifiers for the target domain.

2) In terms of MAPs, the performance of the multiple source domain adaptation method CP-MDA is worse than that of DAM on the Kodak and YouTube datasets. A possible explanation is that the manifold assumption is employed in CP-MDA when learning the optimal weight (*i.e.*, γ_s in (1)) for each source domain. However, the manifold assumption may not hold well for real-world consumer videos on the two datasets, which thus degrades the recognition performances of CP-MDA.

3) On all three datasets, our proposed method DSM_{sim} is consistently better than the two most related methods DAM and CP-MDA that use the pre-defined weight or the learned weight (*i.e.*, γ_s in (1)) for each source domain. The results clearly demonstrate that it is beneficial to employ the selected relevant source domains for domain adaptation rather than use all the source domains. Our proposed method DSM outperforms DSM_{sim}, which further demonstrates the effectiveness of the proposed target decision function by integrating SIFT features and ST features.

4) Our proposed method DSM achieves the best results on all three datasets. We believe that the selection of optimal source domains can also well cope with noisy web images. Compared with the MAPs of SVM_A, DASVM, Multi-KMM, DAM and CP-MDA on the Kodak dataset, the relative improvements of our result are 26.87%, 38.08%, 46.41%, 28.20% and 45.27%, respectively. On the YouTube dataset (*resp.*, the CCV dataset), the relative improvemen-

Table 3. Mean Average Precisions (MAPs) of all methods on the CCV dataset.

	SVM_A	DASVM	Multi-KMM	DAM	CP-MDA	DSM_{sim}	DSM
MAP	17.14%	18.38%	19.77%	17.01%	17.49%	17.80%	**21.76%**

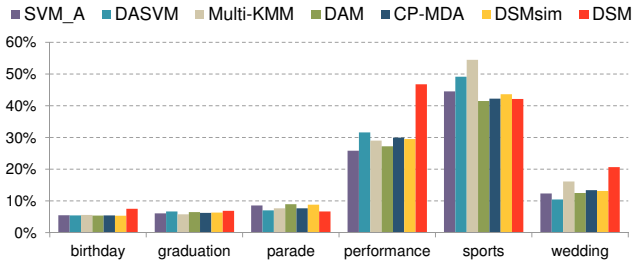

Figure 4. Per-event Average Precisions (APs) of all methods on the CCV dataset.

t of our method DSM over the best existing method DAM (*resp.*, Multi-KMM) are no less than 8.23% (*resp.*, 10.07%). Moreover, according to per-event APs, DSM achieves the best performances in 4 out of 6 event classes over other existing methods on each of the three datasets.

5. Conclusion

We have developed a new multiple source domain adaptation method for event recognition by leveraging a large number of freely available web images from different sources. By introducing a new data-dependent regularizer and a new target classifier, our work called Domain Selection Machine (DSM) can select the most relevant source domains when the samples in the target domains are additionally represented by another type of features (*i.e.*, the ST features). Extensive experiments on three real-world consumer video datasets clearly demonstrate the effectiveness of our proposed method DSM.

Acknowledgement: This research is supported by the Singapore National Research Foundation under its Interactive & Digital Media (IDM) Public Sector R&D Funding Initiative and administered by the IDM Programme Office.

References

[1] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *ICCV*, 2011. 2

[2] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(4):49–57, July 2006. 3

[3] L. Bruzzone and M. Marconcini. Domain adaptation problems: a dasvm classification technique and a circular validation strategy. *T-PAMI*, 32(5):770–787, 2010. 2, 5

[4] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001. 4, 5

[5] R. Chattopadhyay, J. Ye, S. Panchanathan, W. Fan, and I. Davidson. Multi-source domain adaptation and its application to early detection of fatigue. In *KDD*, 2007. 2, 3, 5

[6] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng. Nus-wide: A real-world web image database from national university of singapore. In *CIVR*, 2009. 5

[7] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007. 2, 3

[8] G. Doretto and Y. Yao. Boosting for transfer learning with multiple auxiliary domains. In *CVPR*, 2010. 2, 3, 5

[9] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *ICML*, 2009. 2, 3, 4, 5

[10] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank. Domain transfer svm for video concept detection. In *CVPR*, 2009. 2, 3

[11] L. Duan, D. Xu, I. W. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. In *CVPR*, 2010. 1, 2, 3, 5, 7

[12] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011. 2, 3

[13] N. Ikizler-Cinbis, R. G. Cinbis, and S. Sclaroff. Learning actions from the web. In *ICCV*, 2009. 1

[14] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *ECCV*, 2010. 1

[15] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *ICMR*, 2011. 1, 6

[16] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, 2010. 1

[17] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011. 2

[18] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos. In *CVPR*, 2009. 1, 2, 3

[19] Y. Liu, D. Xu, I. W. Tsang, and J. Luo. Textual query of personal photos facilitated by large-scale web data. *T-PAMI*, 33(5):1022–1036, 2011. 6

[20] A. C. Loui, J. Luo, S.-F. Chang, D. Ellis, W. Jiang, L. Kennedy, K. Lee, and A. Yanagawa. Kodak's consumer video benchmark data set: Concept definition and annotation. In *MIR*, 2007. 1

[21] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 3, 5, 6

[22] S. J. Pan and Q. Yang. A survey on transfer learning. *T-KDE*, 22(10):1345–1359, 2010. 2

[23] G.-J. Qi, C. Aggarwal, Y. Rui, Q. Tian, S. Chang, and T. Huang. Towards cross-category knowledge propagation for learning visual concepts. In *CVPR*, 2011. 2

[24] M. T. Rosenstein, Z. Marx, and L. P. Kaelbling. To transfer or not to transfer. In *NIPS 2005 Workshop on Inductive Transfer: 10 Years Later*, 2005. 2, 7

[25] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010. 2

[26] G. Schweikert, C. Widmer, B. Schölkopf, and G. Rätsch. An empirical analysis of domain adaptation algorithms for genomic sequence analysis. In *NIPS*, 2009. 2, 3, 5

[27] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004. 4

[28] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *CVPR*, 2010. 2

[29] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 1, 3, 4, 6

[30] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM MM*, 2007. 2, 3