Visual Event Recognition in Videos by Learning from Web Data

Lixin Duan, Dong Xu, *Member, IEEE,* Ivor Wai-Hung Tsang, Jiebo Luo, *Fellow, IEEE*

Abstract—We propose a visual event recognition framework for consumer videos by leveraging a large amount of loosely labeled web videos (*e.g.*, from YouTube). Observing that consumer videos generally contain large intra-class variations within the same type of events, we first propose a new method called Aligned Space-Time Pyramid Matching (ASTPM) to measure the distances between two video clips. Second, we propose a new cross-domain learning method, referred to as Adaptive Multiple Kernel Learning (A-MKL), in order to 1) fuse the information from multiple pyramid levels and features (*i.e.*, space-time features and static SIFT features) and 2) cope with the considerable variation in feature distributions between videos from two domains (*i.e.*, web video domain and consumer video domain). For each pyramid level and each type of local features, we first train a set of SVM classifiers based on the combined training set from two domains by using multiple base kernels from different kernel types and parameters, which are then fused with equal weights to obtain a prelearned average classifier. In A-MKL, for each event class we learn an adapted target classifier based on multiple base kernels and the prelearned average classifiers from this event class or all the event classes by minimizing both the structural risk functional and mismatch between data distributions of two domains. Extensive experiments demonstrate the effectiveness of our proposed framework that requires only a small number of labeled consumer videos by leveraging web data. We also conduct in-depth investigation on various aspects of the proposed method A-MKL, such as the analysis on the combination coefficients on the prelearned classifiers, the convergence of the learning algorithm, and the performance variation by using different proportions of labeled consumer videos. Moreover, we show that A-MKL using the prelearned classifiers from all the event classes leads to better performance when compared with A-MKL using the prelearned classifiers only from each individual event class.

Index Terms—Visual event recognition, cross-domain learning, transfer learning, adaptive MKL, aligned space-time pyramid matching

1 Introduction

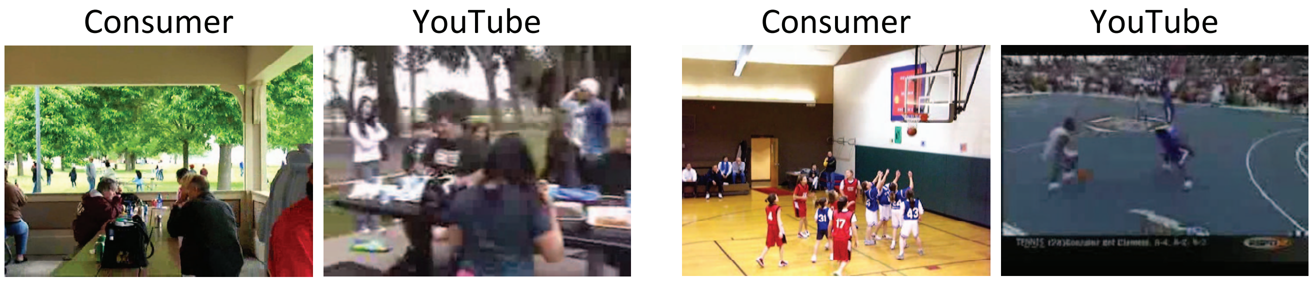
In recent years, digital cameras and mobile phone cameras are becoming popular in our daily life. Consequently, there is an increasingly urgent demand on indexing and retrieving from a large amount of unconstrained consumer videos. In particular, visual event recognition in consumer videos has attracted growing attention. However, this is an extremely challenging computer vision task due to two main issues. First, consumer videos are generally captured by amateurs using hand-held cameras of unstaged events and thus contain considerable camera motion, occlusion, cluttered background and large intra-class variations within the same type of events, making their visual cues highly variable and thus less discriminant. Second, these users are generally reluctant to annotate many consumer videos, posing a great challenge to the traditional video event recognition techniques that often cannot learn robust classifiers from a limited number of labeled training videos.

While a large number of video event recognition techniques have been proposed (see Section 2 for more details), few of them [5], [16], [17], [28], [30] focused on event recognition in the highly unconstrained consumer video domain. Loui *et al.* [30] developed a consumer video data set which was manually labeled for 25 concepts including activities, occasions, static concepts like scenes and objects, as well as sounds. Based on this data set, Chang *et al.* [5] developed a multi-modal consumer video classification system by using visual features and audio features. In the web video domain, Liu *et al.* [28] employed strategies inspired by PageRank to effectively integrate both motion features and static features for action recognition in YouTube videos. In [16], action models were first learned from loosely labeled web images and then used for identifying human actions in YouTube videos. However, the work in [16] cannot distinguish actions like "sitting_down" and "standing_up" because it did not utilize temporal information in its image-based model. Recently, Ikizler-Cinbis and Sclaroff [17] proposed to employ multiple instance learning to integrate multiple features of the people, objects and scenes for action recognition in YouTube videos.

Most event recognition methods [5], [25], [28], [32], [41], [43], [49] followed the conventional framework. First, a sufficiently large corpus of training data is collected, in which the concept labels are generally obtained through expensive human annotation. Next, robust classifiers (also called models or concept detectors) are learned from the training data. Finally, the classifiers are used to detect the presence of the events in any test

- *Lixin Duan, Dong Xu and Ivor Wai-Huang Tsang are with the School of Computer Engineering at the Nanyang Technological University, Singapore. Jiebo Luo is with the Computer Science Department at the University of Rochester, NY, USA.*

Digital Object Indentifier 10.1109/TPAMI.2011.265 0162-8828/11/$26.00 © 2011 IEEE



(a) “picnic” (b) “sports”

Fig. 1. Four sample frames from consumer videos and YouTube videos. Our work aims to recognize the events in consumer videos by using a limited number of labeled consumer videos and a large number of YouTube videos. The examples from two events (*i.e.*, “picnic” and “sports”) illustrate the considerable appearance differences between consumer videos and YouTube videos, which poses great challenges to conventional learning schemes but can be effectively handled by our cross-domain learning method Adaptive Multiple Kernel Learning (A-MKL).

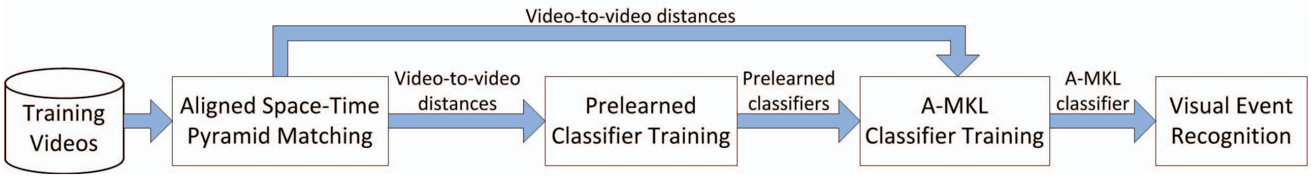


Fig. 2. The flowchart of the proposed visual event recognition framework. It consists of an *aligned space-time pyramid matching* method ASTPM that effectively measures the distances between two video clips and a *cross-domain learning* method A-MKL that effectively copes with the considerable variation in feature distributions between the web videos and consumer videos.

data. When sufficient and strong labeled training samples are provided, these event recognition methods have achieved promising results. However, for visual event recognition in consumer videos, it is time-consuming and expensive for users to annotate a large number of consumer videos. It is also well-known that the learned classifiers from a limited number of labeled training samples are usually not robust and do not generalize well.

In this paper, we propose a new event recognition framework for consumer videos by leveraging a large amount of loosely labeled YouTube videos. Our work is based on the observation that a large amount of loosely labeled YouTube videos can be readily obtained by using keywords (also called tags) based search. However, the quality of YouTube videos is generally lower than consumer videos because YouTube videos are often down-sampled and compressed by the web server. In addition, YouTube videos may have been selected and edited to attract attention while consumer videos are in their naturally captured state. In Fig. 1, we show four frames from two events (*i.e.*, “picnic” and “sports”) as examples to illustrate the considerable appearance differences between consumer videos and YouTube videos. Clearly, the visual feature distributions of samples from the two domains (*i.e.*, web video domain and consumer video domain) can change considerably in terms of the statistical properties (such as mean, intra-class and inter-class variance).

Our proposed framework is shown in Fig. 2 and consists of two contributions. First, we extend the recent work on pyramid matching [13], [25], [26], [48], [49] and present a new matching method called *aligned space-time pyramid matching* (ASTPM) to effectively measure the distances between two video clips that may be from different domains. Specifically, we divide each video clip into space-time volumes over multiple levels. We calculate the pair-wise distances between any two volumes and further integrate the information from different volumes with Integer-flow Earth Mover’s Distance (EMD) to explicitly align the volumes. In contrast to the fixed volume-to-volume matching used in [25], the space-time volumes of two videos across different space-time locations can be matched using our ASTPM method, making it better at coping with the large intra-class variations within the same type of events (*e.g.*, moving objects in consumer videos can appear at different space-time locations, and the background within two different videos even captured from the same scene may be shifted due to considerable camera motions).

The second is our main contribution. In order to cope with the considerable variation between feature distributions of videos from the web video domain and consumer video domain, we propose a new cross-domain learning method, referred to as Adaptive Multiple Kernel Learning (A-MKL). Specifically, we first obtain one prelearned classifier for each event class at each pyramid level and with each type of local features, in which existing kernel methods (*e.g.*, SVM) can be readily employed. In this work, we adopt the *prelearned average classifier* by equally fusing a set of SVM classifiers that are prelearned based on a combined training set from two domains by using multiple base kernels from different kernel types and parameters. For each event class, we then learn an *adapted classifier* based on multiple base kernels and the prelearned average classifiers from this event class or all event classes by minimizing both the structural risk functional and mismatch between data distributions of two domains. It is noteworthy that the utilization of the prelearned average classifiers from all event classes in A-MKL is based on the observation that some events may share common motion patterns [47]. For example, the videos from some events (such as "birthday", "picnic" and "wedding") usually contain a number of people talking with each other. Therefore, it is beneficial to learn an adapted classifier for "birthday" by leveraging the prelearned classifiers from "picnic" and "wedding".

The remainder of this paper is organized as follows. Section 2 will provide brief reviews of event recognition. The proposed methods ASTPM and A-MKL will be introduced in Sections 3 and 4, respectively. Extensive experimental results will be presented in Section 5, followed by conclusions and future work in Section 6.

2 Related Work on Event Recognition

Event recognition methods can be roughly categorized into model-based methods and appearance-based techniques. Model-based approaches relied on various models including HMM [35], coupled HMM [3], and Dynamic Bayesian Network [33] to model the temporal evolution. The relationships among different body parts and regions are also modeled in [3], [35], in which object tracking needs to be conducted at first before model learning.

Appearance-based approaches employed space-time features extracted from volumetric regions that can be densely sampled or from salient regions with significant local variations in both spatial and temporal dimensions [24], [32], [41]. In [19], Ke *et al.* employed boosting to learn a cascade of filters based on space-time features for efficient visual event detection. Laptev and Lindeberg [24] extended the ideas of Harris interest point operators and Dollar *et al.* [7] employed separable linear filters to detect the salient volumetric regions. Statistical learning methods including SVM [41] and probabilistic Latent Semantic Analysis (pLSA) [32] were then applied by using the aforementioned space-time features to obtain the final classification. Recently, Kovashka and Grauman [20] proposed a new feature formation technique by exploiting multi-level vocabularies of space-time neighborhoods. Promising results [12], [20], [27], [32], [41] have been reported on video data sets under controlled conditions, such as Weizman [12] and KTH [41] data sets. Interested readers may refer to [45] for a recent survey.

Recently, researchers proposed new methods to address the more challenging event recognition task on video data sets captured under much less uncontrolled conditions, including movies [25], [43] and broadcast news videos [49]. In [25], Laptev *et al.* integrated local space-time features (*i.e.*, Histograms of Oriented Gradient and Histograms of Optical Flow), space-time pyramid matching and SVM for action classification in movies. In order to locate the actions from movies, a new discriminative clustering algorithm [11] was developed based on the weakly-labeled training data that can be readily obtained from movie scripts without any cost of manual annotation. Sun *et al.* [43] employed Multiple Kernel Learning (MKL) to efficiently fuse three types of features including a so-called SIFT average descriptor and two trajectory-based features. To recognize events in diverse broadcast news videos, Xu and Chang [49] proposed a multi-level temporal matching algorithm for measuring video similarity.

However, all these methods followed the conventional learning framework by assuming that the training and test samples are from the same domain and feature distribution. When the total number of labeled training samples is limited, the performances of these methods would be poor. In contrast, the goal of our work is to propose an effective event recognition framework for consumer videos by leveraging a large amount of loosely labeled web videos, where we must deal with the distribution mismatch between videos from two domains (*i.e.*, web video domain and consumer video domain). As a result, our algorithm can learn a robust classifier for event recognition requiring only a small number of labeled consumer videos.

3 Aligned Space-Time Pyramid Matching

Recently, pyramid matching algorithms were proposed for different applications, such as object recognition, scene classification, and event recognition in movies and news videos [13], [25], [26], [48], [49]. These methods involved pyramidal binning in different domains (*e.g.*, feature, spatial, or temporal domain), and improved performances were reported by fusing the information from multiple pyramid levels. Spatial pyramid matching [26] and its space-time extension [25] used fixed block-to-block matching and fixed volume-to-volume matching (we refer to it as *unaligned space-time matching*), respectively. In contrast, our proposed *Aligned Space-Time Pyramid Matching* (ASTPM) extends the methods of Spatially Aligned Pyramid Matching (SAPM) [48] and Temporally Aligned Pyramid Matching (TAPM) [49] from either the spatial domain or the temporal domain to the joint space-time domain, where the volumes across different space and time locations can be matched.

Similar to [25], we divide each video clip into 8^l non-overlapped space-time volumes over multiple levels, $l = 0, \ldots, L-1$, where the volume size is set as $1/2^l$ of the original video in width, height and temporal dimension. Fig. 3 illustrates the partitions of two videos V_i and V_j at level-1. Following [25], we extract the local space-time (ST) features including Histograms of Oriented Gradient (HOG) and Histograms of Optical Flow (HOF), which are further concatenated together to form lengthy feature vectors. We also sample each video clip to extract image

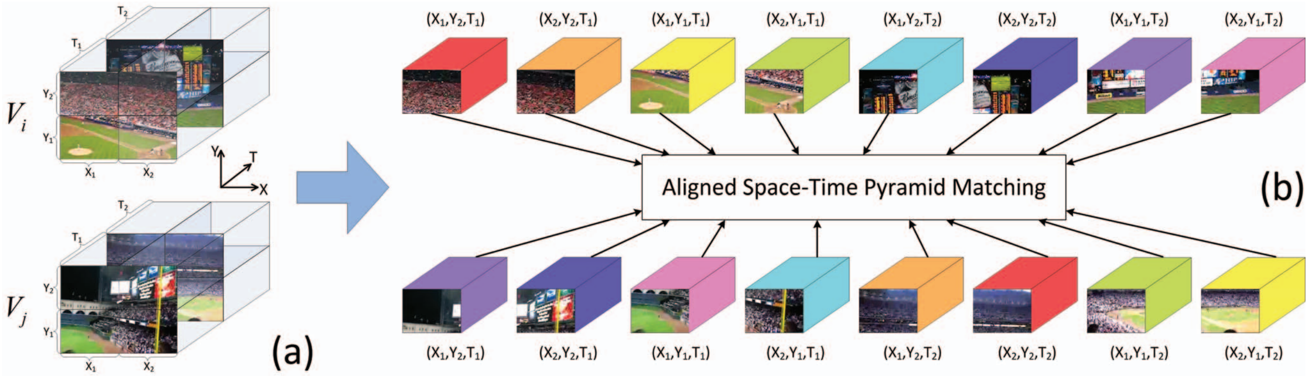


Fig. 3. Illustration of the proposed aligned space-time pyramid matching (ASTPM) method at level-1: (a) Each video is divided into 8 space-time volumes along the width, height and temporal dimensions; (b) The matching results are obtained by using our ASTPM method. Each pair of matched volumes from two videos is highlighted in the same color. For better visualization, please see the colored PDF file.

frames and then extract static local SIFT features from them [31].

Our method consists of two matching stages. In the first matching stage, we calculate the pairwise distance D_{rc} between each two space-time volumes $V_i(r)$ and $V_j(c)$, where $r, c = 1, \ldots, R$ with R being the total number of volumes in a video. The space-time features are vector-quantized into *visual words* and then each space-time volume is represented as a token-frequency feature. As suggested in [25], we use χ^2 distance to measure the distance D_{rc}. Noting that each space-time volume consists of a set of image blocks, we also extract toke-frequency features from each image block by vector-quantizing the corresponding SIFT features into visual words. And based on the token-frequency features, as suggested in [49], the pairwise distance D_{rc} between two volumes $V_i(r)$ and $V_j(c)$ is calculated by using Earth Mover's Distance (EMD) [39] as follows:

$$D_{rc} = \frac{\sum_{u=1}^{H}\sum_{v=1}^{I}\widehat{f}_{uv}d_{uv}}{\sum_{u=1}^{H}\sum_{v=1}^{I}\widehat{f}_{uv}},$$

where H, I are the numbers of image blocks in $V_i(r), V_j(c)$ respectively, d_{uv} is the distance between two image block (Euclidean distance is used in this work), and $\widehat{f}_{uv}$ is the optimal flow that can be obtained by solving the linear programming problem as follows:

$$\widehat{f}_{uv} = \arg\min_{f_{uv}\geq 0}\sum_{u=1}^{H}\sum_{v=1}^{I} f_{uv}d_{uv},$$
$$\text{s.t.} \sum_{u=1}^{H}\sum_{v=1}^{I} f_{uv} = 1; \sum_{v=1}^{I} f_{uv} \leq \frac{1}{H}, \forall u; \sum_{u=1}^{H} f_{uv} \leq \frac{1}{I}, \forall v$$

In the second stage, we further integrate the information from different volumes by using integer-flow EMD to explicitly align the volumes. We try to solve a flow matrix $\widehat{F}_{rc}$ containing binary elements which represent unique matches between volumes $V_i(r)$ and $V_j(c)$. As suggested in [48], [49], such binary solution can be conveniently computed by using the standard Simplex method for linear programming, which is presented in the following theorem:

Theorem 1 ([18]): The linear programming problem

$$\widehat{F}_{rc} = \arg\min_{F_{rc}\in\{0,1\}}\sum_{r=1}^{R}\sum_{c=1}^{R} F_{rc}D_{rc},$$
$$\text{s.t.} \sum_{c=1}^{R} F_{rc} = 1, \ \forall r; \ \sum_{r=1}^{R} F_{rc} = 1, \ \forall c,$$

will always have an integer optimal solution when solved by using the Simplex method.

Fig. 3 illustrates the matching results of two videos after using our ASTPM method, indicating the reasonable matching between similar scenes (*i.e.*, the crowds, the playground and the Jumbotron TV screens in the two videos). It is also worth mentioning that our ASTPM method can preserve the space-time proximity relations between volumes from two videos at level-1 when using the ST or SIFT features. Specifically, the ST features (*resp.*, SIFT features) in one volume can only be matched to the ST features (*resp.*, SIFT features) within another volume at level-1 in our ASTPM method rather than arbitrary ST features (*resp.*, SIFT features) within the entire video as in the classical bag-of-words model (*e.g.*, ASTPM at level-0).

Finally, the distance $D_l(V_i, V_j)$ between two video clips V_i and V_j at level-l can be directly calculated by

$$D_l(V_i, V_j) = \frac{\sum_{r=1}^{R}\sum_{c=1}^{R}\widehat{F}_{rc}D_{rc}}{\sum_{r=1}^{R}\sum_{c=1}^{R}\widehat{F}_{rc}}.$$

In the next section, we will propose a new cross-domain learning method to fuse the information from multiple pyramid levels and different types of features.

4 Adaptive Multiple Kernel Learning

Following the terminology from prior literature, we refer to the web video domain as *auxiliary domain* $\mathcal{D}^A$ (*a.k.a.*, *source domain*) and consumer video domain as *target domain* $\mathcal{D}^T = \mathcal{D}_l^T \cup \mathcal{D}_u^T$, where $\mathcal{D}_l^T$ and $\mathcal{D}_u^T$ represent the labeled and unlabeled data in the target domain, respectively. In this work, we denote $\mathbf{I}_n$ as the $n \times n$ identity matrix and $\mathbf{0}_n, \mathbf{1}_n \in \mathbb{R}^n$ as $n \times 1$ column vectors of all zeros and all ones, respectively. The inequality $\mathbf{a} = [a_1, \ldots, a_n]' \geq \mathbf{0}_n$ means that $a_i \geq 0$ for $i = 1, \ldots, n$. Moreover, the element-wise product between vectors $\mathbf{a}$ and $\mathbf{b}$ is defined as $\mathbf{a} \circ \mathbf{b} = [a_1 b_1, \ldots, a_n b_n]'$.

4.1 Brief review of related learning work

Cross-domain learning (*a.k.a.*, *transfer learning* or *domain adaptation*) methods have been proposed for many applications [6], [8], [9], [29], [50]. To take advantage of all labeled patterns from both auxiliary and target domains, Daumé III [6] proposed Feature Replication (FR) by using augmented features for SVM training. In Adaptive SVM (A-SVM) [50], the target classifier $f^T(\mathbf{x})$ is adapted from an existing classifier $f^A(\mathbf{x})$ (referred to as auxiliary classifier) trained based on the samples from the auxiliary domain. Specifically, the target decision function is defined as follows:

$$f^T(\mathbf{x}) = f^A(\mathbf{x}) + \Delta f(\mathbf{x}), \tag{1}$$

where $\Delta f(\mathbf{x})$ is called as a *perturbation function* that is learned by using the labeled data from the target domain only (*i.e.*, $\mathcal{D}_l^T$). While A-SVM can also employ multiple auxiliary classifiers, these auxiliary classifiers are fused with predefined weights to obtain $f^A(\mathbf{x})$ [50]. Moreover, the target classifier $f^T(\mathbf{x})$ is learned based on only one kernel. Recently, Duan [8] proposed Domain Transfer SVM (DTSVM) to simultaneously reduce the mismatch between the distributions of two domains and learn a target decision function. The mismatch was measured by Maximum Mean Discrepancy (MMD) [2] based on the distance between the means of the samples respectively from the auxiliary domain $\mathcal{D}^A$ and the target domain $\mathcal{D}^T$ in a Reproducing Kernel Hilbert Space (RKHS) spanned by a kernel function k, namely:

$$\text{DIST}_k(\mathcal{D}^A, \mathcal{D}^T) = \left\| \frac{1}{n_A} \sum_{i=1}^{n_A} \varphi(\mathbf{x}_i^A) - \frac{1}{n_T} \sum_{i=1}^{n_T} \varphi(\mathbf{x}_i^T) \right\|_{\mathcal{H}}, \tag{2}$$

where $\mathbf{x}_i^A$'s and $\mathbf{x}_i^T$'s are the samples from the auxiliary and target domains, respectively, and the kernel function k is induced from the nonlinear feature mapping function $\varphi(\cdot)$, *i.e.*, $k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)'\varphi(\mathbf{x}_j)$. We define a column vector s with $N = n_A + n_T$ entries, in which the first n_A entries are set as $1/n_A$ and the remaining entries are set as $-1/n_T$, respectively. With the above notions, the square of MMD in (2) can be simplified as follows [2], [8]:

$$\text{DIST}_k^2(\mathcal{D}^A, \mathcal{D}^T) = \text{tr}(\mathbf{KS}), \tag{3}$$

where tr($\mathbf{KS}$) represents the trace of $\mathbf{KS}$, $\mathbf{S} = \text{ss}' \in \mathbb{R}^{N \times N}$, and $\mathbf{K} = \begin{bmatrix} \mathbf{K}^{A,A} & \mathbf{K}^{A,T} \\ \mathbf{K}^{T,A} & \mathbf{K}^{T,T} \end{bmatrix} \in \mathbb{R}^{N \times N}$, and $\mathbf{K}^{A,A} \in \mathbb{R}^{n_A \times n_A}$, $\mathbf{K}^{T,T} \in \mathbb{R}^{n_T \times n_T}$ and $\mathbf{K}^{A,T} \in \mathbb{R}^{n_A \times n_T}$ are the kernel matrices defined for the auxiliary domain, the target domain and the cross-domain from the auxiliary domain to the target domain, respectively.

4.2 Formulation of A-MKL

Motivated by A-SVM [50] and DTSVM [8], we propose a new cross-domain learning method to learn a target classifier adapted from a set of prelearned classifiers as well as a perturbation function that is based on multiple base kernels k_m's. The prelearned classifiers are used as prior for learning a robust adapted target classifier. In A-MKL, the existing machine learning methods (*e.g.*, SVM, FR and so on) using different types of features (*e.g.*, SIFT and ST features) can be readily used to obtain the prelearned classifiers. Moreover, in contrast to A-SVM [50] which uses the predefined weights to combine the prelearned auxiliary classifiers, we learn the linear combination coefficients $\beta_p|_{p=1}^P$ of the prelearned classifiers $f_p(\mathbf{x})|_{p=1}^P$ in this work, where P is the total number of the prelearned classifiers. Specifically, we use the average classifiers from one event class or all the event classes as the prelearned classifiers (see Sections 5.3 and 5.6 for more details). We additionally employ multiple *predefined* kernels to model the perturbation function in this work, because the utilization of multiple *base* kernels k_m's instead of a single kernel can further enhance the interpretability of the decision function and improve performances [23]. We refer to our cross-domain learning method based on multiple base kernels as Adaptive Multiple Kernel Learning (A-MKL), because A-MKL can handle the distribution mismatch between the web video domain and the consumer video domain.

Following the traditional MKL assumption [23], the kernel function k is represented as a linear combination of multiple base kernels k_m's as follows:

$$k = \sum_{m=1}^{M} d_m k_m, \tag{4}$$

where d_m's are the linear combination coefficients, $d_m \geq 0$ and $\sum_{m=1}^M d_m = 1$; each base kernel function k_m is induced from the nonlinear feature mapping function $\varphi_m(\cdot)$, *i.e.*, $k_m(\mathbf{x}_i, \mathbf{x}_j) = \varphi_m(\mathbf{x}_i)'\varphi_m(\mathbf{x}_j)$, and M is the total number of base kernels. Inspired by semiparametric SVM [42], we define the target decision function on any sample $\mathbf{x}$ as follows:

$$f^T(\mathbf{x}) = \sum_{p=1}^{P} \beta_p f_p(\mathbf{x}) + \underbrace{\sum_{m=1}^{M} d_m \mathbf{w}_m' \varphi_m(\mathbf{x}) + b}_{\Delta f(\mathbf{x})}, \tag{5}$$

where $\Delta f(\mathbf{x}) = \sum_{m=1}^M d_m \mathbf{w}_m' \varphi_m(\mathbf{x}) + b$ is the perturbation function with b as the bias term. Note that multiple base kernels are employed in $\Delta f(\mathbf{x})$.

As in [8], we employ the MMD criterion to reduce the mismatch between the data distributions of two domains in this work. Let us define the linear combination coefficient vector as $\mathbf{d} = [d_1, \ldots, d_M]'$ and the feasible set of $\mathbf{d}$ as $\mathcal{M} = \{\mathbf{d} \in \mathbb{R}^M | \mathbf{1}_M' \mathbf{d} = 1, \mathbf{d} \geq \mathbf{0}_M\}$. With (4), (3) can be rewritten as:

$$\text{DIST}_k^2(\mathcal{D}^A, \mathcal{D}^T) = \Omega(\mathbf{d}) = \mathbf{h}'\mathbf{d}, \tag{6}$$

where $\mathbf{h} = [\mathrm{tr}(\mathbf{K}_1\mathbf{S}), \ldots, \mathrm{tr}(\mathbf{K}_M\mathbf{S})]'$, $\mathbf{K}_m = [\varphi_m(\mathbf{x})'\varphi_m(\mathbf{x})] \in \mathbb{R}^{N\times N}$ is the m-th base kernel matrix defined on the samples from both auxiliary and target domains. Let us denote the labeled training samples from both the auxiliary and target domains (*i.e.*, $\mathcal{D}^A \cup \mathcal{D}_l^T$) as $(\mathbf{x}_i, y_i)|_{i=1}^n$, where n is the total number of labeled training samples from the two domains. The optimization problem in A-MKL is then formulated as follows:

$$\min_{\mathbf{d}\in\mathcal{M}} \quad G(\mathbf{d}) = \frac{1}{2}\Omega^2(\mathbf{d}) + \theta\, J(\mathbf{d}), \tag{7}$$

where

$$\begin{aligned} J(\mathbf{d}) = \min_{\mathbf{w}_m,\boldsymbol{\beta},b,\xi_i} \; & \frac{1}{2}\left(\sum_{m=1}^{M} d_m\|\mathbf{w}_m\|^2 + \lambda\|\boldsymbol{\beta}\|^2\right) + C\sum_{i=1}^{n}\xi_i, \\ \text{s.t.} \quad & y_i f^T(\mathbf{x}_i) \geq 1 - \xi_i, \; \xi_i \geq 0, \end{aligned} \tag{8}$$

$\boldsymbol{\beta} = [\beta_1, \ldots, \beta_P]'$ is the vector of β_p's and $\lambda, C > 0$ are the regularization parameters. Denote $\tilde{\mathbf{w}}_m = [\mathbf{w}_m', \sqrt{\lambda}\boldsymbol{\beta}']'$ and $\tilde{\varphi}_m(\mathbf{x}_i) = [\varphi_m(\mathbf{x}_i)', \frac{1}{\sqrt{\lambda}}\boldsymbol{f}(\mathbf{x}_i)']'$, where $\boldsymbol{f}(\mathbf{x}_i) = [f_1(\mathbf{x}_i), \ldots, f_P(\mathbf{x}_i)]'$. The optimization problem in (8) can then be rewritten as follows:

$$\begin{aligned} J(\mathbf{d}) = \min_{\tilde{\mathbf{w}}_m,b,\xi_i} \; & \frac{1}{2}\sum_{m=1}^{M} d_m\|\tilde{\mathbf{w}}_m\|^2 + C\sum_{i=1}^{n}\xi_i, \\ \text{s.t.} \quad & y_i\left(\sum_{m=1}^{M} d_m\tilde{\mathbf{w}}_m'\tilde{\varphi}_m(\mathbf{x}_i) + b\right) \geq 1 - \xi_i, \; \xi_i \geq 0. \end{aligned} \tag{9}$$

By defining $\tilde{\mathbf{v}}_m = d_m\tilde{\mathbf{w}}_m$, we rewrite the optimization problem in (9) as a quadratic programming (QP) problem [37]:

$$\begin{aligned} J(\mathbf{d}) = \min_{\tilde{\mathbf{v}}_m,b,\xi_i} \; & \frac{1}{2}\sum_{m=1}^{M} \frac{\|\tilde{\mathbf{v}}_m\|^2}{d_m} + C\sum_{i=1}^{n}\xi_i, \\ \text{s.t.} \quad & y_i\left(\sum_{m=1}^{M} \tilde{\mathbf{v}}_m'\tilde{\varphi}_m(\mathbf{x}_i) + b\right) \geq 1 - \xi_i, \; \xi_i \geq 0. \end{aligned} \tag{10}$$

Theorem 2 ([8], [37]): The optimization problem in (7) is jointly convex with respect to $\mathbf{d}$, $\tilde{\mathbf{v}}_m$, b and ξ_i.

Proof: Note that the first term $\frac{1}{2}\Omega^2(\mathbf{d})$ of $G(\mathbf{d})$ in (7) is a quadratic term with respect to $\mathbf{d}$. And other terms in (10) are linear except the term $\frac{1}{2}\sum_{m=1}^{M}\frac{\|\tilde{\mathbf{v}}_m\|^2}{d_m}$. As shown in [37], this term is also jointly convex with respect to $\mathbf{d}$ and $\tilde{\mathbf{v}}_m$. Therefore, the optimization problem in (7) is jointly convex with respect to $\mathbf{d}$, $\tilde{\mathbf{v}}_m$, b and ξ_i. □

With Theorem 2, the objective in (7) can reach its global minimum. By introducing the Lagrangian multiplier $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]'$, we solve the dual form of the optimization problem in (10) as follows:

$$J(\mathbf{d}) = \max_{\boldsymbol{\alpha}\in\mathcal{A}} \mathbf{1}_n'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\left(\sum_{m=1}^{M} d_m\tilde{\mathbf{K}}_m\right)(\boldsymbol{\alpha}\circ\mathbf{y}), \tag{11}$$

where $\mathbf{y} = [y_1, \ldots, y_n]'$ is the label vector of the training samples, $\mathcal{A} = \{\boldsymbol{\alpha} \in \mathbb{R}^n | \boldsymbol{\alpha}'\mathbf{y} = 0, \mathbf{0}_n \leq \boldsymbol{\alpha} \leq C\mathbf{1}_n\}$ is the feasible set of the dual variable $\boldsymbol{\alpha}$, $\tilde{\mathbf{K}}_m = [\tilde{\varphi}_m(\mathbf{x}_i)'\tilde{\varphi}_m(\mathbf{x}_j)] \in \mathbb{R}^{n\times n}$ is defined by the labeled training data from both domains, and $\tilde{\varphi}_m(\mathbf{x}_i)'\tilde{\varphi}_m(\mathbf{x}_j) = \varphi_m(\mathbf{x}_i)'\varphi_m(\mathbf{x}_j) + \frac{1}{\lambda}\boldsymbol{f}(\mathbf{x}_i)'\boldsymbol{f}(\mathbf{x}_j)$. Recall that $\boldsymbol{f}(\mathbf{x})$ is a vector of the predictions on $\mathbf{x}$ from the prelearned classifiers f_p's, which resembles the label information of $\mathbf{x}$ and can be used to construct the *idealized* kernel [22]. Thus, the new kernel matrix $\tilde{\mathbf{K}}_m$ can be viewed as the integration of both the visual information (*i.e.*, from $\mathbf{K}_m$) and the label information, which can lead to better discriminative power. Surprisingly, the optimization problem in (11) is in the same form as the dual of SVM with the kernel matrix $\sum_{m=1}^{M} d_m\tilde{\mathbf{K}}_m$. Thus, the optimization problem can be solved by existing SVM solvers, such as LIBSVM [4].

4.3 Learning Algorithm of A-MKL

In this work, we employ the reduced gradient descent procedure proposed in [37] to iteratively update the linear combination coefficient $\mathbf{d}$ and the dual variable $\boldsymbol{\alpha}$ in (7).

Updating the dual variable $\boldsymbol{\alpha}$: Given the linear combination coefficient $\mathbf{d}$, we solve the optimization problem in (11) to obtain the dual variable $\boldsymbol{\alpha}$ by using LIBSVM [4].

Updating the linear combination coefficient $\mathbf{d}$: Suppose the dual variable $\boldsymbol{\alpha}$ is fixed. With respect to $\mathbf{d}$, the objective function $G(\mathbf{d})$ in (7) becomes:

$$\begin{aligned} G(\mathbf{d}) &= \frac{1}{2}\mathbf{d}'\mathbf{h}\mathbf{h}'\mathbf{d} + \theta\left(\mathbf{1}_n'\boldsymbol{\alpha} - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\left(\sum_{m=1}^{M} d_m\tilde{\mathbf{K}}_m\right)(\boldsymbol{\alpha}\circ\mathbf{y})\right) \\ &= \frac{1}{2}\mathbf{d}'\mathbf{h}\mathbf{h}'\mathbf{d} - \theta\mathbf{q}'\mathbf{d} + \text{const}, \end{aligned} \tag{12}$$

where $\mathbf{q} = [\frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\mathbf{K}_1(\boldsymbol{\alpha}\circ\mathbf{y}), \ldots, \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\mathbf{K}_M(\boldsymbol{\alpha}\circ\mathbf{y})]'$ and the last term is a constant term that is irrelevant to $\mathbf{d}$, namely, const $= \theta\left(\mathbf{1}_n'\boldsymbol{\alpha} - \frac{1}{2\lambda}\sum_{i,j=1}^{n}\alpha_i\alpha_j y_i y_j \boldsymbol{f}(\mathbf{x}_i)'\boldsymbol{f}(\mathbf{x}_j)\right)$.

We adopt the second-order gradient descent method to update the linear combination coefficient $\mathbf{d}$ at iteration $t+1$ by:

$$\mathbf{d}_{t+1} = \mathbf{d}_t - \eta_t\mathbf{g}_t, \tag{13}$$

where η_t is the learning rate which can be obtained by using a standard line search method [37], $\mathbf{g}_t = (\nabla_t^2 G)^{-1}\nabla_t G$ is the updating direction, and $\nabla_t G = \mathbf{h}\mathbf{h}'\mathbf{d}_t - \theta\mathbf{q}$ and $\nabla_t^2 G = \mathbf{h}\mathbf{h}'$ are the first-order and second-order derivatives of G in (12) with respect to $\mathbf{d}$ at the t-th iteration, respectively. Note that $\mathbf{h}\mathbf{h}'$ is not of full rank, and therefore we replace $\mathbf{h}\mathbf{h}'$ by $\mathbf{h}\mathbf{h}' + \epsilon\mathbf{I}_M$ to avoid numerical instability, where ϵ is set as 10^{-5} in the experiments. Then, the updating function (13) can be rewritten as follows:

$$\mathbf{d}_{t+1} = (1-\eta_t)\mathbf{d}_t + \eta_t\mathbf{d}_t^{\text{new}}, \tag{14}$$

where $\mathbf{d}_t^{\text{new}} = \theta(\mathbf{h}\mathbf{h}' + \epsilon\mathbf{I}_M)^{-1}\mathbf{q}$. Note that by replacing $\mathbf{h}\mathbf{h}'$ with $\mathbf{h}\mathbf{h}' + \epsilon\mathbf{I}_M$, the solution to $\nabla_t G = \mathbf{h}\mathbf{h}'\mathbf{d}_t - \theta\mathbf{q} = \mathbf{0}_M$ becomes $\mathbf{d}_t^{\text{new}}$. Given $\mathbf{d}_t \in \mathcal{M}$, we project $\mathbf{d}_t^{\text{new}}$ onto the feasible set $\mathcal{M}$ to ensure $\mathbf{d}_{t+1} \in \mathcal{M}$ as well.

Algorithm 1 Adaptive Multiple Kernel Learning

```
Input: labeled training samples $(\mathbf{x}_i, y_i)|_{i=1}^n$, prelearned classifiers $f_p(\mathbf{x})|_{p=1}^P$ and predefined base kernel functions $k_m|_{m=1}^M$
Initialization: $t \leftarrow 1$ and $\mathbf{d}_t \leftarrow \frac{1}{M}\mathbf{1}_M$
Solve for the dual variables $\boldsymbol{\alpha}_t$ in (11) by using SVM.
While $t < T_{\max}$ Do
    $\mathbf{q}_t \leftarrow [\frac{1}{2}(\boldsymbol{\alpha}_t \circ \mathbf{y})'\mathbf{K}_1(\boldsymbol{\alpha}_t \circ \mathbf{y}), \ldots, \frac{1}{2}(\boldsymbol{\alpha}_t \circ \mathbf{y})'\mathbf{K}_M(\boldsymbol{\alpha}_t \circ \mathbf{y})]'$
    $\mathbf{d}_t^{\text{new}} \leftarrow \theta(\mathbf{h}\mathbf{h}' + \epsilon\mathbf{I}_M)^{-1}\mathbf{q}_t$ and project $\mathbf{d}_t^{\text{new}}$ onto the feasible set $\mathcal{M}$.
    Update the base kernel combination coefficients $\mathbf{d}_{t+1}$ by using (14) with standard line search.
    Solve for the dual variables $\boldsymbol{\alpha}_{t+1}$ in (11) by using SVM.
    If $|G(\mathbf{d}_{t+1}) - G(\mathbf{d}_t)| \le \tau$ then break
    $t \leftarrow t + 1$
End While
Output: $\mathbf{d}_t$ and $\boldsymbol{\alpha}_t$
```

The whole optimization procedure is summarized in Algorithm 1[1]. We terminate the iterative updating procedure, once the objective in (7) converges or the number of iterations reaches $T_{\max}$. We set the tolerance parameter $\tau = 10^{-5}$ and $T_{\max} = 15$ in the experiments.

Note that by setting the derivative of the Lagrangian obtained from (9) with respect to $\tilde{\mathbf{w}}_m$ to zero, we obtain $\tilde{\mathbf{w}}_m = \sum_{i=1}^n \alpha_i y_i \tilde{\varphi}_m(\mathbf{x}_i)$. Recall that $\sqrt{\lambda}\boldsymbol{\beta}$ and $\frac{1}{\sqrt{\lambda}}\boldsymbol{f}(\mathbf{x}_i)$ are the last P entries of $\tilde{\mathbf{w}}_m$ and $\tilde{\varphi}_m(\mathbf{x}_i)$, respectively. Therefore, the linear combination coefficient $\boldsymbol{\beta}$ of the prelearned classifiers can be obtained as follows:

$$\boldsymbol{\beta} = \frac{1}{\lambda}\sum_{i=1}^n \alpha_i y_i \boldsymbol{f}(\mathbf{x}_i).$$

With the optimal dual variables $\boldsymbol{\alpha}$ and linear combination coefficients $\mathbf{d}$, the target decision function (5) of our method A-MKL can be rewritten as follows:

$$f^T(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i \left(\sum_{m=1}^M d_m \mathbf{K}_m(\mathbf{x}_i, \mathbf{x}) + \frac{1}{\lambda}\boldsymbol{f}(\mathbf{x}_i)'\boldsymbol{f}(\mathbf{x})\right) + b.$$

4.4 Differences from related learning work

A-SVM [50] assumes that the target classifier $f^T(\mathbf{x})$ is adapted from existing auxiliary classifiers $f_p^A(\mathbf{x})$'s. However, our proposed method A-MKL is different from A-SVM in several aspects: 1) In A-SVM, the auxiliary classifiers are learned by using only the training samples from the auxiliary domain. In contrast, the prelearned classifiers used in A-MKL can be learned by using the training samples either from the auxiliary domain or from both domains; 2) In A-SVM, the auxiliary classifiers are fused with predefined weights γ_p's in the target classifier, *i.e.*, $f^T(\mathbf{x}) = \sum_{p=1}^P \gamma_p f_p^A(\mathbf{x}) + \Delta f(\mathbf{x})$. In contrast, A-MKL learns the optimal combination coefficients β_p's in (5); 3) In A-SVM, the perturbation function $\Delta f(\mathbf{x})$ is based on one single kernel, *i.e.*, $\Delta f(\mathbf{x}) = \mathbf{w}'\varphi(\mathbf{x}) + b$. However, in A-MKL, the perturbation function $\Delta f(\mathbf{x}) = \sum_{m=1}^M d_m \mathbf{w}_m' \varphi_m(\mathbf{x}) + b$ in (5) is based on multiple kernels, and the optimal kernel combination is automatically determined during the learning process; 4) A-SVM cannot utilize the unlabeled data in the target domain. On the contrary, the valuable unlabeled data in the target domain are used in the MMD criterion of A-MKL for measuring the data distribution mismatch between two domains.

Our work is also different from the prior work DTSVM [8], where the target decision function $f^T(\mathbf{x}) = \sum_{m=1}^M d_m \mathbf{w}_m' \varphi_m(\mathbf{x}) + b$ is only based on multiple base kernels. In contrast, in A-MKL, we use a set of prelearned classifiers $f_p(\mathbf{x})$'s as the parametric functions, and model the perturbation function $\Delta f(\mathbf{x})$ based on multiple base kernels in order to better fit the target decision function. To fuse multiple prelearned classifiers, we also learn the optimal linear combination coefficients β_p's. As shown in the experiments, our A-MKL is more robust in real applications by utilizing optimally combined classifiers as the prior.

MKL methods [23], [37] utilize the training data and the test data drawn from the same domain. When they come from different distributions, MKL methods may fail to learn the optimal kernel. This would degrade the classification performance in the target domain. On the contrary, A-MKL can better make use of the data from two domains to improve the classification performance.

5 Experiments

In this section, we first evaluate the effectiveness of the proposed method Aligned Space-Time Pyramid Matching (ASTPM). We then compare our proposed method Adaptive Multiple Kernel Learning (A-MKL) with the baseline SVM, and three existing cross-domain learning algorithms: Feature Replication (FR) [6], Adaptive SVM (A-SVM) [50] and Domain Transfer SVM (DTSVM) [8], as well as a Multiple Kernel Learning (MKL) method discussed in [8]. We also analyze the learned combination coefficients β_p's of the prelearned classifiers, illustrate the convergence of the learning algorithm of A-MKL and investigate the performance variations of A-MKL using different proportions of labeled consumer videos.

[1] The source code can be downloaded at our project web page http://vc.sce.ntu.edu.sg/index_files/VisualEventRecognition/VisualEventRecognition.html.

Moreover, we show that A-MKL using the prelearned classifiers from all event classes is better than A-MKL using the prelearned classifiers from one event class.

For all methods, we train one-versus-all classifiers with a fixed regularization parameter $C = 1$. For performance evaluation, we use the non-interpolated Average Precision (AP) as in [25], [49] which corresponds to the multi-point average precision value of a precision-recall curse and incorporates the effect of recall. Mean Average Precision (MAP) is the mean of APs over all the event classes.

5.1 Data set description and features

In our data set, part of the consumer videos are derived (under a usage agreement) from the Kodak Consumer Video Benchmark Data Set [30] which was collected by Kodak from about 100 real users over the period of one year. There are 1358 consumer video clips in the Kodak data set. A second part of the Kodak data set contains web videos from YouTube collected using keywords based search. After removing TV commercial videos and low-quality videos, there are 1873 YouTube video clips in total. An ontology of 25 semantic concepts were defined and keyframe based annotation was performed by the students at Columbia University to assign binary labels (presence or absence) for each visual concept for both sets of videos (see [30] for more details).

In this work, six events "birthday", "picnic", "parade", "show", "sports" and "wedding" are chosen for experiments. We additionally collected new consumer video clips from real users on our own. Similarly to [30], we also downloaded new YouTube videos from the website. Moreover, we also annotate the consumer videos to determine whether a specific event occurred by asking an annotator, who is not involved in the algorithmic design, to watch each video clip rather than just look at the key frames as done in [30]. For video clips in the Kodak consumer data set [30], only the video clips receiving positive labels in their keyframe based annotation are reexamined. We do not additionally annotate the YouTube videos[2] collected by ourselves and Kodak because in a real scenario we can only obtain loosely labeled YouTube videos and cannot use any further manual annotation. It should be clear that our consumer video set comes from two sources – the Kodak consumer video data set and our additional collection of personal videos, and our web video set is a combined set of YouTube videos as well. We confirm that the quality of YouTube videos is much lower than that of consumer videos directly collected from real users. Therefore, our data set is quite challenging for cross-domain learning algorithms. The total numbers of consumer videos and YouTube videos are 195 and 906, respectively. Note that our data set is a single-label data set, *i.e.*, each video belongs to only one event.

TABLE 1
Means and standard deviations (%) of MAPs over six events at different levels using SVM with the default kernel parameter for SIFT features.

	Gaussian	Laplacian	ISD	ID
Level-0	41.4 ± 3.7	44.2 ± 3.8	45.0 ± 3.5	46.2 ± 4.0
Level-1 (Unaligned)	43.0 ± 2.7	47.7 ± 1.7	49.0 ± 1.6	48.2 ± 1.5
Level-1 (Aligned)	50.4 ± 3.7	53.8 ± 1.8	52.9 ± 3.6	51.0 ± 2.5

TABLE 2
Means and standard deviations (%) of MAPs over six events at different levels using SVM with the default kernel parameter for ST features.

	Gaussian	Laplacian	ISD	ID
Level-0	22.2 ± 1.8	36.1 ± 0.8	22.0 ± 3.8	35.6 ± 0.7
Level-1 (Unaligned)	20.1 ± 1.0	33.9 ± 0.6	21.8 ± 0.7	33.4 ± 0.7
Level-1 (Aligned)	20.6 ± 0.7	35.8 ± 1.7	22.3 ± 1.1	35.9 ± 1.8

In real-world applications, the labeled samples in the target domain (*i.e.*, consumer video domain) are usually *much fewer* than those in the auxiliary domain (*i.e.*, web video domain). In this work, all 906 loosely labeled YouTube videos are used as labeled training data in the auxiliary domain. We randomly sample three consumer videos from each event (18 videos in total) as the labeled training videos in the target domain, and the remaining videos in the target domain are used as the test data. We sample the labeled target training videos for five times and report the means and standard deviations of MAPs or per-event APs for each method.

For all the videos in the data sets, we extract two types of features. The first one is the local space-time (ST) feature [25], in which 72-dimensional Histograms of Oriented Gradient (HOG) and 90-dimensional Histograms of Optical Flow (HOF) are extracted by using the online tool[3]. After that, they are concatenated together to form a 162-dimensional feature vector. We also sample each video clip at a rate of 2 frames per second to extract image frames from each video clip (we have 65 frames per video on average). For each frame, we extract 128-dimensional SIFT features from salient regions, which are detected by Difference-of-Gaussian (DoG) interest point detector [31]. On the average, we have 1385 ST features and 4144 SIFT features per video. Then, we build *visual vocabularies* by using k-means to group the ST features and SIFT features into 1000 and 2500 clusters, respectively.

5.2 Aligned Space-Time Pyramid Matching vs. Unaligned Space-Time Pyramid Matching

We compare our proposed Aligned Space-Time Pyramid Matching (ASTPM) discussed in Section 3 with the fixed volume-to-volume matching method, referred to as Unaligned Space-Time Pyramid Matching (USTPM), used in [25]. In [25], the space-time volumes of one video clip are matched with the volumes of the other video at the

[2] The annotator felt that at least 20% of YouTube videos are incorrectly labeled after checking the video clips.

[3] http://www.irisa.fr/vista/Equipe/People/Laptev/download.html

same spatial and temporal locations at each level. In other words, the second matching stage based on Integer-flow EMD is not applied, and the distance between two video clips is equal to the sum of diagonal elements of the distance matrix, *i.e.*, $\sum_{r=1}^{R} D_{rr}$. For computational efficiency, we set the total number of levels $L = 2$ in this work. Therefore, we have two ways of partitions, in which one video clip is divided into $1 \times 1 \times 1$ and $2 \times 2 \times 2$ space-time volumes, respectively.

We use the baseline SVM classifier learned by using the combined training data set from two domains. We test the performances with four types of kernels: Gaussian kernel (*i.e.*, $K(i,j) = \exp\left(-\gamma D^2(V_i, V_j)\right)$), Laplacian kernel (*i.e.*, $K(i,j) = \exp\left(-\sqrt{\gamma} D(V_i, V_j)\right)$), inverse square distance (ISD) kernel (*i.e.*, $K(i,j) = \frac{1}{\gamma D^2(V_i,V_j)+1}$) and inverse distance (ID) kernel (*i.e.*, $K(i,j) = \frac{1}{\sqrt{\gamma} D(V_i,V_j)+1}$), where $D(V_i, V_j)$ represents the distance between video V_i and V_j, and γ is the kernel parameter. We use the default kernel parameter $\gamma = \gamma_0 = \frac{1}{A}$, where A is the mean value of the square distances between all training samples as suggested in [25].

Tables 1 and 2 show the MAPs of the baseline SVM over six events for SIFT and ST features at different levels according to different types of kernels with the default kernel parameter. Based on the means of MAPs, we have the following three observations: 1) In all cases, the results at level-1 using aligned matching are better than those at level-0 based on SIFT features, which demonstrates the effectiveness of space-time partition and it is also consistent with the findings for prior pyramid matching methods [25], [26], [48], [49]; 2) At level-1, our proposed ASTPM outperforms USTPM used in [25], thanks to the additional alignment of space-time volumes; 3) The results from space-time features are not as good as those from static SIFT features. As also reported in [15], a possible explanation is that the extracted ST features may fall on cluttered backgrounds because the consumer videos are generally captured by amateurs with hand-held cameras.

5.3 Performance comparisons of cross-domain learning methods

We compare our method A-MKL with other methods including the baseline SVM, FR, A-SVM, MKL and DTSVM. For the baseline SVM, we report the results of SVM_AT and SVM_T, in which the labeled training samples are from two domains (*i.e.*, the auxiliary domain and the target domain) and only from the target domain, respectively. Specifically, The aforementioned four types of kernels (*i.e.*, Gaussian kernel, Laplacian kernel, ISD kernel and ID kernel) are adopted. Note that in our initial conference version [10] of this paper, we have demonstrated that A-MKL outperforms other methods by setting the kernel parameter as $\gamma = 2^l \gamma_0$, where $l \in \mathcal{L} = \{-6, -4, \ldots, 2\}$. In this work, we test A-MKL by using another set of kernel parameters, *i.e.*, $\mathcal{L} = \{-3, -2, \ldots, 1\}$. Note that the total number of base kernels is $16|\mathcal{L}|$ from two pyramid levels and two types of local features, four types of kernels and $|\mathcal{L}|$ kernel parameters, where $|\mathcal{L}|$ is the cardinality of $\mathcal{L}$.

All methods are compared in three cases: (a) Classifiers learned based on SIFT features; (b) Classifiers learned based on ST features; (c) Classifiers learned based on both SIFT and ST features. For both SVM_AT and FR (*resp.* SVM_T), we train $4|\mathcal{L}|$ independent classifiers with the corresponding $4|\mathcal{L}|$ base kernels for each pyramid level and each type of local features using the training samples from two domains (*resp.*, the training samples from target domain). And we further fuse the $4|\mathcal{L}|$ independent classifiers with equal weights to obtain the *average classifier* f_l^{SIFT} or f_l^{ST}, where $l = 0$ and 1. For SVM_T, SVM_AT and FR, the final classifier is obtained by fusing average classifiers with equal weights (*e.g.*, $\frac{1}{2}\left(f_0^{SIFT} + f_1^{SIFT}\right)$ for case (a), $\frac{1}{2}\left(f_0^{ST} + f_1^{ST}\right)$ for case (b) and $\frac{1}{4}\left(f_0^{SIFT} + f_1^{SIFT} + f_0^{ST} + f_1^{ST}\right)$ for case (c)). For A-SVM, we learn $4|\mathcal{L}|$ independent auxiliary classifiers for each pyramid level and each type of local features using the training data from the auxiliary domain and the corresponding $4|\mathcal{L}|$ base kernels, and then we independently learn four adapted target classifies from two pyramid levels and two types of features by using the labeled training data from the target domain based on Gaussian kernel with the default kernel parameter [50]. Similar to SVM_T, SVM_AT and FR, the final A-SVM classifier is obtained by fusing two (*resp.*, four) adapted target classifiers for cases (a) and (b) (*resp.*, case (c)). For MKL and DTSVM, we simultaneously learn the linear combination coefficients of $8|\mathcal{L}|$ base kernels (for cases (a) or (b)) or $16|\mathcal{L}|$ base kernels (for case (c)) by using the combined training samples from both domains. Recall that for our method A-MKL, we make use of prelearned classifiers as well as multiple base kernels (see (5) in Section 4.2). In the experiment, we consider each average classifier as one prelearned classifier and learn the target decision function of A-MKL based on two average classifiers $f_l^{SIFT}|_{l=0}^{1}$ or $f_l^{ST}|_{l=0}^{1}$ for cases (a) or (b) (*resp.*, all the four average classifiers for case (c)) as well as $8|\mathcal{L}|$ base kernels based on SIFT or ST features for cases (a) or (b) (*resp.*, $16|\mathcal{L}|$ base kernels based on both types of features for case (c)). For A-MKL, we empirically fix $\theta = 10^{-5}$ and set $\lambda = 20$ for all three cases. Considering that DTSVM and A-MKL can take advantage of both labeled and unlabeled data by using the MMD criterion to measure the mismatch in data distributions between two domains, we use semi-supervised setting in this work. More specifically, all the samples (including test samples) from the target domain and auxiliary domain are used to calculate $\mathbf{h}$ in (6). Note that all test samples are used as unlabeled data during the learning process.

Table 3 reports the means and standard deviations of MAPs over all six events in three cases for all methods. From Tables 3, we have the following observations based on the means of MAPs:

1) The best result of SVM_T is worse than that of SVM_AT, which demonstrates that the learned SVM

TABLE 3
Means and standard deviations (%) of MAPs over six events for all methods in three cases.

	SVM_T	SVM_AT	FR	A-SVM	MKL	DTSVM	A-MKL
MAP-(a)	42.32 ± 5.50	53.93 ± 5.58	49.98 ± 5.63	38.42 ± 7.93	47.19 ± 2.59	52.36 ± 1.88	57.14 ± 2.34
MAP-(b)	32.56 ± 2.08	24.73 ± 2.22	28.44 ± 2.61	24.95 ± 1.25	35.34 ± 1.55	31.07 ± 2.60	37.24 ± 1.58
MAP-(c)	42.00 ± 4.94	36.23 ± 3.37	44.11 ± 3.57	32.40 ± 4.99	46.92 ± 2.53	53.78 ± 2.99	$\mathbf{58.20 \pm 1.87}$

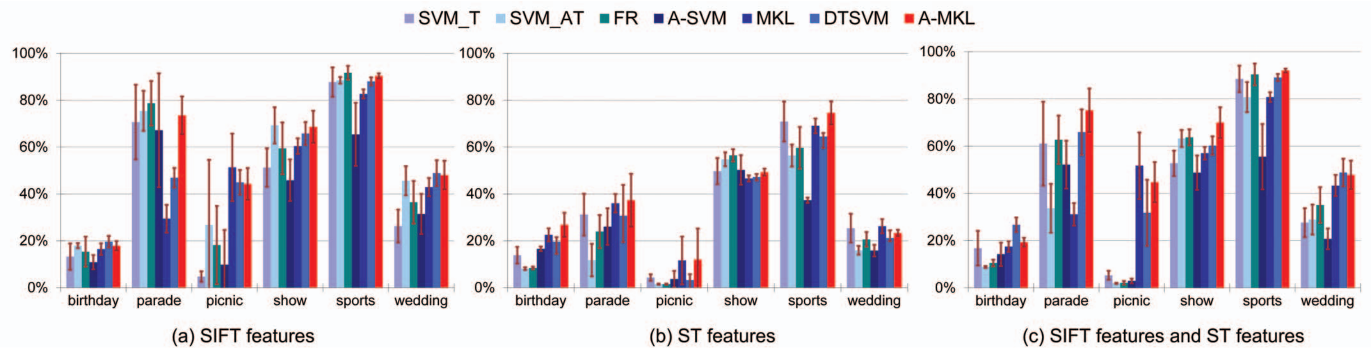


Fig. 4. Means and standard deviations of per-event APs of six events for all methods.

classifiers based on a limited number of training samples from the target domain are not robust. We also observe that SVM_T is always better than SVM_AT for cases (b) and (c). A possible explanation is that the ST features of video samples from the auxiliary and target domains distribute sparsely in the ST feature space, which makes the ST feature not robust and thus it is more likely that the data from the auxiliary domain may degrade the event recognition performances in the target domain for cases (b) and (c).

2) In this application, A-SVM achieves the worst results in cases (a) and (c) in terms of the mean of MAPs, possibly because the limited number of labeled training samples (*e.g.*, three positive samples per event) in the target domain are not sufficient for A-SVM to robustly learn an adapted target classifier which is based on only one kernel.

3) DTSVM is generally better than MKL in terms of the mean of MAPs. This is consistent with [8].

4) For all methods, the MAPs based on SIFT features are better that those based on ST features. In practice, the simple ensemble method, SVM_AT, achieves good performances when only using the SIFT features in case (a). It indicates that SIFT features are more effective for event recognition in consumer videos. However, the MAPs of SVM_AT, FR and A-SVM in case (c) are much worse compared with case (a). It suggests that the simple late fusion methods using equal weights are not robust for integrating strong features and weak features. In contrast, for DTSVM and our method A-MKL, the results in case (c) are improved by learning optimal linear combination coefficients to effectively fuse two types of features.

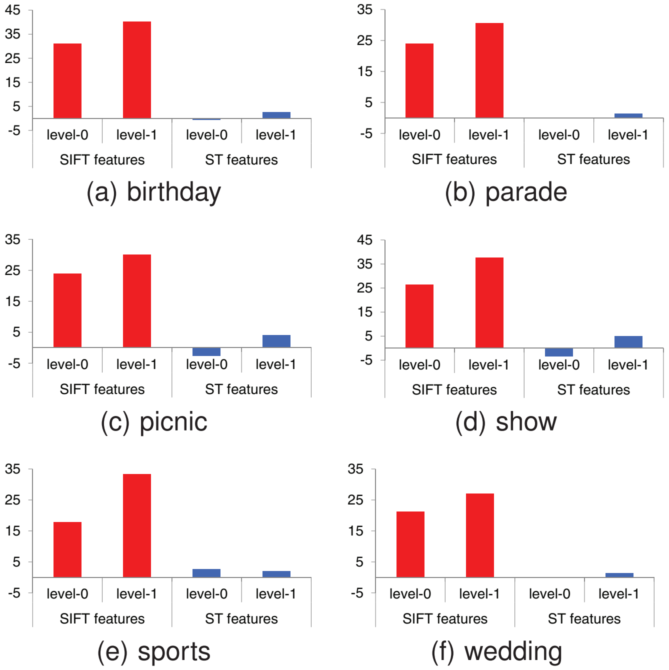
5) For each of three cases, our proposed method A-MKL achieves the best performance by effectively fusing average classifiers (from two pyramid levels and two types of local features) and multiple base kernels as well as reducing the mismatch in the data distributions between two domains. We also believe the utilization of multiple base kernels and prelearned average classifiers can also well cope with YouTube videos with noisy labels. In Table 3, compared with the best means of MAP-s of SVM_T (42.32%), SVM_AT (53.93%), FR (49.98%), A-SVM (38.42%), MKL (47.19%) and DTSVM (53.78%), the relative improvements of our best result (58.20%) are 37.52%, 7.92%, 16.54%, 51.48%, 23.33% and 8.22%, respectively.

In Fig. 4, we plot the means and standard deviations of per-event APs for all methods. Our method achieves the best performances in 3 out of 6 events in case (c) and some concepts enjoy large performance gains according to the means of per-event APs, *e.g.*, the AP of "parade" significantly increases from 65.96% (DTSVM) to 75.21% (A-MKL).

5.4 Analysis on the combination coefficients β_p's of the prelearned classifiers

Recall that we learn the linear combination coefficients β_p's of the prelearned classifiers f_p's in A-MKL. And the absolute value of each β_p reflects the importance of the corresponding prelearned classifier. Specifically, the larger $|\beta_p|$ is, the more f_p contributes in the target decision function. For better representation, let us denote the corresponding average classifiers f_0^{SIFT}, f_1^{SIFT}, f_0^{ST} and f_1^{ST} as f_1, f_2, f_3 and f_4, respectively.

Taking one round of training/test data split in the target domain for example, we draw the combination coefficients β_p's of the four prelearned classifiers f_p's for all events in Fig. 5. In this experiment, we again set $\mathcal{L} = \{-3, -2, \ldots, 1\}$. We observe that the absolute values of β_1 and β_2 are always much larger than those of β_3 and β_4, which shows that the prelearned classifiers (*i.e.*, f_1 and f_2) based on SIFT features play dominant roles among all the prelearned classifiers. This is not surprising, because SIFT features are much more robust than ST features as demonstrated in Section 5.3. From Fig. 5, we also observe that the values of β_3 and β_4 are generally not close to zero, which demonstrates that A-MKL can



(a) birthday

(b) parade

(c) picnic

(d) show

(e) sports

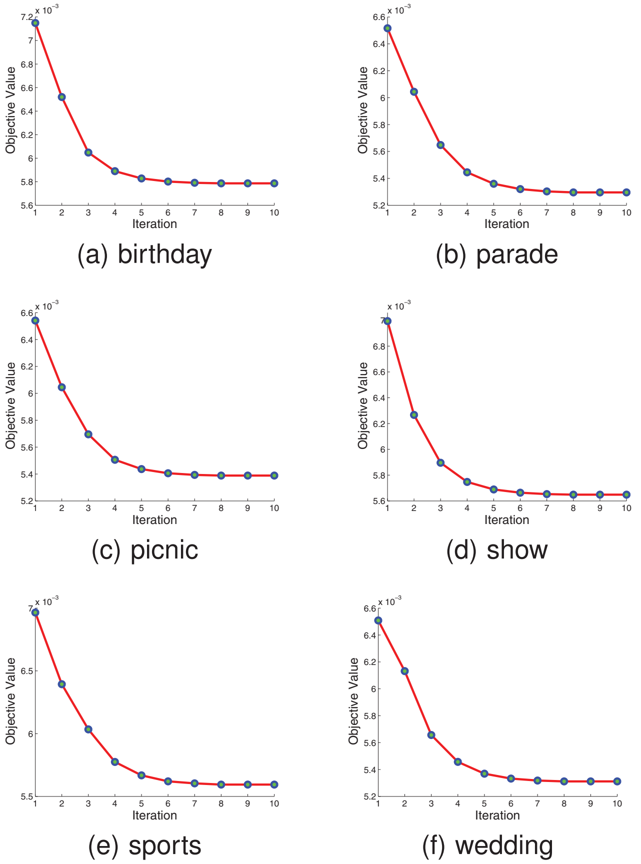
(f) wedding

Fig. 5. Illustration of the combination coefficients β_p's of the prelearned classifiers for all events.

further improve the event recognition performance by effectively integrating strong and weak features. Recall that A-MKL using both types of features outperforms A-MKL with only SIFT features (see Table 3). We have similar observations for other rounds of experiments.

5.5 Convergence of A-MKL learning algorithm

Recall that we iteratively update the dual variable $\boldsymbol{\alpha}$ and the linear combination coefficient $\mathbf{d}$ in A-MKL (see Section 4.3). We take one round of training/test data split as an example to discuss the convergence of the iterative algorithm of A-MKL, in which we also set $\mathcal{L}$ as $\{-3, -2, \ldots, 1\}$, and we use both types of features. In Fig. 6, we plot the change of the objective value of A-MKL with respect to the number of iterations. We observe that A-MKL converges after about eight iterations for all events. We have similar observations for other rounds of experiments.



(a) birthday

(b) parade

(c) picnic

(d) show

(e) sports

(f) wedding

Fig. 6. Illustration of the convergence of A-MKL learning algorithm for all events.

5.6 Utilization of additional prelearned classifiers from other event classes

In the previous experiments, for a specific event class, we only utilize the prelearned classifiers (*i.e.*, average classifiers $f_l^{SIFT}|_{l=0}^{1}$ and $f_l^{ST}|_{l=0}^{1}$) from this event class. As a general learning method, A-MKL can readily incorporate additional prelearned classifiers. In our event recognition application, we observe that some events may share common motion patterns [47]. For example, the videos from some events (like “birthday”, “picnic” and “wedding”) usually contain a number of people talking with each other. Thus, it is beneficial to learn an adapted classifier for “birthday” by leveraging the prelearned classifiers from “picnic” and “wedding”. Based on this observation, for each event, we make use of the prelearned classifiers from all event classes for the learning of the adapted classifier in A-MKL. Therefore, the total number of the prelearned classifiers is 24 for each event when using both types of features. For better representation, we refer to A-MKL with four prelearned average classifiers discussed in Sections 5.3, 5.4 and 5.5 (*resp.*, A-MKL with all 24 prelearned average classifiers) as A-MKL_4 (*resp.*, A-MKL_24).

In Sections 5.3, 5.4 and 5.5, the same kernel parameter set (*i.e.* $\mathcal{L} = \{-3, -2, \ldots, 1\}$) is used for the base kernels and also employed to obtain the prelearned average classifiers in A-MKL. In this experiment, we also use the same set of kernel parameters (*i.e.* $\mathcal{L} = \{-3, -2, \ldots, 1\}$) for the base kernels but we additionally vary the set of kernel parameters (denoted as $\mathcal{H}$ for better representation) to obtain the prelearned average classifiers for A-MKL_4 and A-MKL_24. Specifically, for each pyramid level and each type of features, we learn $4|\mathcal{H}|$ independent SVM classifiers from the parameter set $\mathcal{H}$ and four types of kernels (*i.e.*, Gaussian kernel, Laplacian kernel, ISD kernel and ID kernel) by using the training samples from both the auxiliary and target domains, which are further averaged to obtain one prelearned classifier (*i.e.*, $f_l^{SIFT}|_{l=0}^{1}$ or $f_l^{ST}|_{l=0}^{1}$).

In Table 4, we compare the results of A-MKL_4 and A-MKL_24 when using 1) $\mathcal{H} = \{-3, -2, \ldots, 1\}$; 2) $\mathcal{H} = \{-4, -3, \ldots, 1\}$; 3) $\mathcal{H} = \{-5, -4, \ldots, 1\}$ and 4)

TABLE 4
Means and standard deviations (%) of MAPs of A-MKL (referred to as A-MKL_4) using the prelearned average classifiers from the same event class and A-MKL (referred to as A-MKL_24) using the prelearned average classifiers from all six event classes. Different sets of kernel parameters (*i.e.*, $\mathcal{H}$) are employed to obtain the prelearned average classifiers.

	$\mathcal{H} = \{-3, -2, \ldots, 1\}$	$\mathcal{H} = \{-4, -3, \ldots, 1\}$	$\mathcal{H} = \{-5, -4, \ldots, 1\}$	$\mathcal{H} = \{-6, -5, \ldots, 1\}$
A-MKL_4	58.20 ± 1.87	58.33 ± 2.33	58.56 ± 2.53	57.16 ± 3.02
A-MKL_24	58.74 ± 3.30	59.04 ± 3.53	59.25 ± 3.73	$\mathbf{59.28} \pm 2.57$

$\mathcal{H} = \{-6, -5, \ldots, 1\}$. From Table 4, We observe that while the performances of A-MKL_4 and A-MKL_24 change when using different $\mathcal{H}$, A-MKL_24 is consistently better than A-MKL_4 in terms of the mean of MAPs. It clearly demonstrates that A-MKL can learn a more robust target classifier by effectively leveraging the prelearned average classifiers from all the event classes. The performance of A-MKL_24 is the best, when setting $\mathcal{H} = \{-6, -5, \ldots, 1\}$. Compared with the other methods such as SVM_T, SVM_AT, FR, A-SVM, MKL and DTSVM in terms of the mean of per-event APs for case (c), A-MKL_24 achieves the best performances in 4 out of 6 events. The relative improvements of the best mean of MAPs from A-MKL_24 (59.28%) over those from SVM_AT (53.93%) and DTSVM (53.78%) in Table 3 are 9.92% and 10.23%, respectively.

5.7 Performance variations of A-MKL using different proportions of labeled consumer videos

We also investigate the performance variations of A-MKL when using different proportions of labeled training samples from the target domain. Specifically, we randomly choose a proportion (*i.e.*, r) of positive samples from the target domain for each event class. All the randomly chosen samples are considered as the labeled training data from the target domain, while the remainder of samples in the target domain are used as the test data. Again, we sample the labeled target training videos for five times and report the means and standard deviations of MAPs. Considering that the users are reluctant to annotate a large number of consumer videos, we set r as $5\%, 10\%, 20\%$ and 30%. By using both the SIFT and ST features (*i.e.*, case (c)), we compare our methods A-MKL_4 and A-MKL_24 with the baseline method SVM_T and the existing cross-domain learning method DTSVM that achieves the second best results in case (c) (see Tables 3). For DTSVM, A-MKL_4 and A-MKL_24, we use the same settings as in Sections 5.6 and 5.6 by setting the kernel parameter set $\mathcal{L}$ for the base kernels as $\{-3, -2, \ldots, 1\}$. For A-MKL_4, we set the kernel parameter set $\mathcal{H}$ for the prelearned average classifiers as $\{-3, -2, \ldots, 1\}$; and for A-MKL_24, $\mathcal{H}$ is set as $\{-6, -5, \ldots, 1\}$, with which A-MKL_24 achieves the best result (see Table 4).

From Fig. 7, we have the following observations based on the mean of MAPs. First, the results of all methods generally increase, when using more labeled training samples from the target domain. Second, the cross-domain learning methods DTSVM, A-MKL_4 and A-MKL_24 consistently outperform the baseline method SVM_T. Third, our methods A-MKL_4 and A-MKL_24 consistently perform better than DTSVM, which shows the effectiveness of the utilization of prelearned average classifiers. Finally, A-MKL_24 is consistently better than A-MKL_4, which demonstrates that the information from other event classes is helpful for improving the event recognition performance for an individual class.

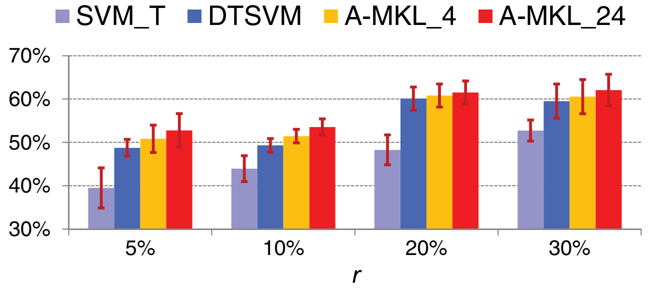


Fig. 7. Means and standard deviations of MAPs over six events for SVM_T, DTSVM, A-MKL_4 and A-MKL_24 when using different proportions (*i.e.*, r) of labeled training consumer videos.

5.8 Running time and memory usage

Finally, we report the running time and memory usage of our proposed framework. All the experiments are conducted on a server machine with Intel Xeon 3.33GHz CPUs and 32GB RAM by using a single thread. The main costs in running time and memory usage are from feature extraction and our proposed ASTPM method. Specifically, on the average it takes about 63.3 seconds (*resp.*, 246.5 seconds) to extract the SIFT features (*resp.*, ST features) from a one-minute-long video. For each video, its SIFT features (*resp.*, ST features) occupy 41.7 megabytes (*resp.*, 17.9 megabytes) on the average. In this work, each type of features are vector-quantized into visual words by using k-means. Considering the quantization process for the SIFT and ST features from training videos can be conducted in an offline manner and the quantization process for the SIFT and ST features from a test video is very fast, we do not count the running time of this process. For our ASTPM using the SIFT and ST features, it respectively takes about 20.9 milliseconds and 0.1 milliseconds (*resp.*, 1213.6 milliseconds and 0.4 milliseconds) to calculate the distance between a pair of videos at level-0 (*resp.*, level-1) on the average. For each event class, on the average it takes about 68.4 seconds to learn one A-MKL classifier, which includes 7.1 seconds for obtaining the prelearned average classifiers. The average prediction time for each test video is only about 11 milliseconds. To accelerate our framework for a median or large scale video event recognition task, we can extract the SIFT and ST features by using multiple threads in a parallel fashion and employ the fast EMD algorithm [34] in ASTPM.

6 Conclusions and Future Work

In this paper, we propose a new event recognition framework for consumer videos by leveraging a large amount of loosely labeled YouTube videos. Specifically, we propose a new pyramid matching method called Aligned Space-Time Pyramid Matching (ASTPM) and a novel cross-domain learning method, referred to as Adaptive Multiple Kernel Learning (A-MKL), to better fuse the information from multiple pyramid levels and different types of local features, and to cope with the mismatch between the feature distributions of consumer videos and web videos. Experiments clearly demonstrate the effectiveness of our framework. To the best of our knowledge, our work is the first to perform event recognition in consumer videos by incorporating cost-effective cross-domain learning.

To put it in a larger perspective, our work falls into the recent research trend of "Internet Vision", where the massive web data including images and videos together with rich and valuable contextual information (e.g., tags, categories and captions) are employed for various computer vision and computer graphics applications such as image annotation [44], [46], image retrieval [29], scene completion [14], and so on. By treating the "*web data as the king*", these methods [14], [44] have achieved promising results by adopting the simplistic learning methods such as the *k*NN classifier. In this work, we have demonstrated that it is beneficial to *learn from web data* by developing more advanced machine learning methods (specifically the cross-domain learning method A-MKL in this work) to further improve the classification performances. A possible future research direction is to develop effective methods to select more useful videos from a large number of low-quality YouTube videos to construct the auxiliary domain.

While cross-domain learning (*a.k.a.*, transfer learning or domain adaptation) has been studied for years in other fields (*e.g.*, natural language processing [1], [6]), it is still an emerging research topic in computer vision [40]. In some vision applications, there is an existing domain (*i.e.*, auxiliary domain) with a large number of labeled data but we want to recognize the images or videos in another domain of interest (*i.e.*, target domain) with very few labeled samples. Besides the adaption between the web domain and consumer domain studied in this work and [29], other examples that vision researchers are recently working on include the adaptation of cross-category knowledge to a new category domain [36], the knowledge transfer by mining semantic relatedness [38], and the adaption between two domains with different feature representations [21], [40]. In the future, we will extend our A-MKL for those interesting vision applications.

Acknowledgments

This work is supported by Singapore A*STAR SERC Grant (082 101 0018).

References

[1] J. Blitzer, R. McDonald, and F. Pereira, "Domain Adaptation with Structural Correspondence Learning," *Proc. Conf. Empirical Methods in Natural Language*, pp. 120–128, 2006.

[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*, vol. 22, no. 4, pp. e49–e57, 2006.

[3] M. Brand, N. Oliver, and A. Pentland, "Coupled Hidden Markov Models for Complex Action Recognition," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 994–999, 1997.

[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.

[5] S.-F. Chang, D. Ellis, W. Jiang, K. Lee, A. Yanagawa, A. C. Loui, and J. Luo, "Large-Scale Multimodal Semantic Concept Detection for Consumer Video," *Proc. ACM Int'l Workshop on Multimedia Information Retrieval*, pp. 255–264, 2007.

[6] Hal Daumé III, "Frustratingly Easy Domain Adaptation," *Proc. Annual Meeting of the Association for Computational Linguistics*, pp. 256–263, 2007.

[7] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior Recognition via Sparse Spatio-Temporal Features," *Proc. IEEE Int'l Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp.65–72, 2005.

[8] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank, "Domain Transfer SVM for Video Concept Detection," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 1375–1381, 2009.

[9] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua, "Domain Adaptation from Multiple Sources via Auxiliary Classifiers," *Proc. Int'l Conf. Machine Learning*, pp. 289–296, 2009.

[10] L. Duan, D. Xu, I. W. Tsang, and J. Luo, "Visual Event Recognition in Videos by Learning from Web Data," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 1959–1966, 2010.

[11] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, "Automatic Annotation of Human Actions in Video," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1491–1498, 2009.

[12] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as Space-Time Shapes," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1395–1402, 2005.

[13] K. Grauman and T. Darrell, "The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 1458–1465, 2005.

[14] J. Hays and A. A. Efros, "Scene Completion using Millions of Photographs," *ACM Trans. Graphics (SIGGRAPH 2007)*, vol. 26, no. 3, 2007.

[15] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. Huang, "Action Detection in Complex Scenes with Spatial and Temporal Ambiguities," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 128–135, 2009.

[16] N. Ikizler-Cinbis, R. G. Cinbis, and S. Sclaroff, "Learning Actions From the Web," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 995–1002, 2009.

[17] N. Ikizler-Cinbis and S. Sclaroff, "Object, Scene and Actions: Combining Multiple Features for Human Action Recognition," *Proc. European Conf. Computer Vision*, pp. 494–507, 2010.

[18] P. A. Jensen and J. F. Bard, "Operations Research Models and Methods," *John Wiley and Sons*, 2003.

[19] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient Visual Event Detection using Volumetric Features," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 166–173, 2005.

[20] A. Kovashka and K. Grauman, "Learning a Hierarchy of Discriminative Space-Time Neighborhood Features for Human Action Recognition," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 2046–2053, 2010.

[21] B. Kulis, K. Saenko, and T. Darrell, "What You Saw is Not What You Get: Domain Adaptation Using Asymmetric Kernel Transforms," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 1785–1792, 2011.

[22] J. T. Kwok and I. W. Tsang, "Learning with Idealized Kernels," *Proc. Int'l Conf. Machine Learning*, pp. 400–407, 2003.

[23] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the Kernel Matrix with Semidefinite Programming," *J. Machine Learning Research*, vol. 5, 27–72, 2004.

[24] I. Laptev and T. Lindeberg, "Space-Time Interest Points," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 432–439, 2003.

[25] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning Realistic Human Actions from Movies," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[26] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 2169–2178, 2006.

[27] Z. Lin, Z. Jiang, and L. S. Davis, "Recognizing Actions by Shape-Motion Prototype Trees," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 444–451, 2009.

[28] J. Liu, J. Luo, and M. Shah, "Recognizing Realistic Actions from Videos 'in the Wild'," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 1996–2003, 2009.

[29] Y. Liu, D. Xu, I. W. Tsang, and J. Luo, "Textual Query of Personal Photos Facilitated by Large-Scale Web Data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 1022–1036, 2011.

[30] 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," *Proc. Int'l Workshop on Multimedia Information Retrieval*, pp. 245–254, 2007.

[31] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[32] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words," *Int'l J. Computer Vision*, vol. 79, no. 3, pp. 299–318, 2008.

[33] N. Oliver, B. Rosario, and A. Pentland, "A Bayesian Computer Vision System for Modeling Human Interactions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, 2000.

[34] O. Pele and M. Werman, "Fast and Robust Earth Mover's Distances," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 460–467, 2009.

[35] P. Peursum, S. Venkatesh, G. A. W. West, and H. H. Bui, "Object Labelling from Human Action Recognition," *Proc. IEEE Int'l Conf. Pervasive Computing and Communications*, pp. 399–406, 2003.

[36] G.-J. Qi, C. Aggarwal, Y. Rui, Q. Tian, S. Chang, and T. Huang, "Towards Cross-Category Knowledge Propagation for Learning Visual Concepts," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 897–904, 2011.

[37] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[38] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele, "What Helps Where – And Why? Semantic Relatedness for Knowledge Transfer," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 910–917, 2010.

[39] Y. Rubner, C. Tomasi, and L. J. Guibas, "The Earth Mover's Distance as a Metrix for Image Retrieval," *Int'l J. Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[40] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting Visual Category Models to New Domains," *Proc. Euro. Conf. Computer Vision*, pp. 213–226, 2010.

[41] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing Human Actions: A Local SVM Approach," *Proc. Int'l Conf. Pattern Recognition*, pp. 32–36, 2004.

[42] A. J. Smola, T. T. Frieß, and B. Schölkopf. "Semiparametric Support Vector and Linear Programming Machines," *Advances in Neural Information Processing System*, pp. 585–591, 1999.

[43] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, "Hierarchical Spatio-Temporal Context Modeling for Action Recognition," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 2004–2011, 2009.

[44] A. Torralba, R. Fergus and W. T. Freeman, "80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

[45] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine Recognition of Human Activities: A Survey," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 18, pp. 1473–1488, 2008.

[46] X.-J. Wang, L. Zhang, X. Li, and W.-Y. Ma, "Annotating Images by Mining Image Search Results," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1919–1932, 2008.

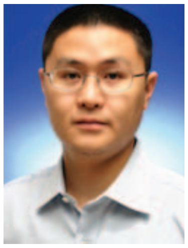
[47] X. Wu, D. Xu, L. Duan, and J. Luo, "Action Recognition using Context and Appearance Distribution Features," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 489–496, 2011.

[48] D. Xu, T. J. Cham, S. Yan, L. Duan and S.-F. Chang, "Near Duplicate Identification with Spatially Aligned Pyramid Matching," *IEEE Trans. Circuits Systems for Video Technology*, vol. 20, no. 8, pp. 1068–1079, 2010.

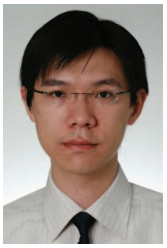
[49] D. Xu and S.-F. Chang, "Video Event Recognition Using Kernel Methods with Multilevel Temporal Alignment," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1985–1997, 2008.

[50] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-Domain Video Concept Detection Using Adaptive SVMs," *Proc. ACM Int'l Conf. Multimedia*, pp. 188–197, 2007.

Lixin Duan received the B.E. degree from the University of Science and Technology of China, Hefei, China, in 2008. He is currently working toward the Ph.D. degree at the School of Computer Engineering, Nanyang Technological University, Singapore. Mr. Duan was a recipient of the Microsoft Research Asia Fellowship in 2009 and the Best Student Paper Award in the IEEE Conference on Computer Vision and Pattern Recognition 2010.



Dong Xu (M'07) received the B.E. and Ph.D. degrees from the University of Science and Technology of China, in 2001 and 2005, respectively. While pursuing his Ph.D., he was with Microsoft Research Asia, Beijing, China, and the Chinese University of Hong Kong, Shatin, Hong Kong, for more than two years. He was a Post-Doctoral Research Scientist with Columbia University, New York, NY, for one year. He is currently an Assistant Professor with Nanyang Technological University, Singapore. His current research interests include computer vision, statistical learning, and multimedia content analysis. Dr. Xu was the co-author of a paper that won the Best Student Paper Award in the prestigious IEEE International Conference on Computer Vision and Pattern Recognition in 2010.



Ivor W. Tsang received the Ph.D. degree in computer science from the Hong Kong University of Science and Technology in 2007. He is currently an Assistant Professor with the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. He is also the Deputy Director of the Center for Computational Intelligence, NTU. Dr. Tsang received the IEEE Transactions on Neural Networks Outstanding 2004 Paper Award in 2006, and the second class prize of the National Natural Science Award 2008, China in 2009. He was awarded the Microsoft Fellowship in 2005 and the Best Paper Award from the IEEE Hong Kong Chapter of Signal Processing Postgraduate Forum in 2006. His work was also awarded the Best Student Paper Prize at CVPR'10.

Jiebo Luo (S'93–M'96–SM'99–F'09) received his B.S. degree from the University of Science and Technology of China in 1989 and Ph.D. degree from the University of Rochester in 1995. He was a Senior Principal Scientist with the Kodak Research Laboratories in Rochester before joining the Computer Science Department at the University of Rochester in Fall 2011. His research interests include image processing, machine learning, computer vision, social multimedia data mining, biomedical informatics, and ubiquitous computing. He has authored over 180 technical papers and holds over 60 U.S. patents. Dr. Luo has been actively involved in numerous technical conferences, including recently serving as the general chair of ACM CIVR 2008, program co-chair of IEEE CVPR 2012 and ACM Multimedia 2010, area chair of IEEE ICASSP 2009–2011, ICIP 2008–2011, and ICCV 2011. He has served on the editorial boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, Pattern Recognition, Machine Vision and Applications, and the Journal of Electronic Imaging. He is a Kodak Distinguished Inventor, a winner of the 2004 Eastman Innovation Award, and a Fellow of the SPIE and IAPR.