Visual Event Recognition in Videos by Learning from Web Data

Lixin Duan Dong Xu Ivor W. Tsang
School of Computer Engineering
Nanyang Technological University
{S080003, DongXu, IvorTsang}@ntu.edu.sg

Jiebo Luo
Kodak Research Labs
Eastman Kodak Company, Rochester, NY, USA
Jiebo.Luo@Kodak.com

Abstract

We propose a visual event recognition framework for consumer domain videos by leveraging a large amount of loosely labeled web videos (e.g., from YouTube). First, we propose a new aligned space-time pyramid matching method to measure the distances between two video clips, where each video clip is divided 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. Second, we propose a new cross-domain learning method in order to 1) fuse the information from multiple pyramid levels and features (i.e., space-time feature and static SIFT feature) and 2) cope with the considerable variation in feature distributions between videos from two domains (i.e., web domain and consumer domain). For each pyramid level and each type of local features, we train a set of SVM classifiers based on the combined training set from two domains using multiple base kernels of different kernel types and parameters, which are fused with equal weights to obtain an average classifier. Finally, we propose a cross-domain learning method, referred to as Adaptive Multiple Kernel Learning (A-MKL), to learn an adapted classifier based on multiple base kernels and the prelearned average classifiers by minimizing both the structural risk functional and the mismatch between data distributions from 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.

1. Introduction

With the rapid adoption of digital cameras and mobile phone cameras, visual event recognition in personal videos produced by consumers has become an important research topic due to its usefulness in automatic video retrieval and indexing. It is a challenging computer vision task to recognize events in consumer domain videos from visual cues because such videos are captured by amateurs using hand-held cameras and generally contain considerable camera motion, occlusion, cluttered background, and large intra-class variations within the same type of events.

While a large number of video event recognition techniques have been proposed (see Section 2 for more details), few [3, 14, 16, 18] focused on event recognition in the highly unconstrained consumer domain. Loui *et al.* [16] 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.* [3] developed a multi-modal consumer video classification system by using visual features and audio features. In the web video domain, Liu *et al.* [14] employed strategies inspired by PageRank to effectively integrate both motion features and static features for action recognition in YouTube videos. In [18], action models were first learned from loosely labeled web images and then used for identifying human actions in YouTube videos. However, their work [18] cannot distinguish actions like "sitting_down" and "standing_up" because it did not utilize temporal information in its image-based model.

Most event recognition methods [3, 10, 14, 19, 22, 24, 26] followed the conventional framework. First, a 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 concepts in any test data. When sufficient and strong labeled training samples are provided, these event recognition methods have achieved promising results. However, it is 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 loosely labeled YouTube videos can be readily obtained by using keywords (also called tags) based

978-1-4244-6983-3/10/$26.00 ©2010 IEEE

search. However, the quality of YouTube videos is generally lower than consumer videos because YouTube videos are often down-sampled and heavily compressed by the web server. In addition, YouTube videos may have been selected and edited to attract attention while consumer videos are in their natural captured state. Therefore, the feature distributions of samples from the two domains (*i.e.*, web domain and consumer domain) may change considerably in terms of the statistical properties (such as mean, intra-class and inter-class variance).

Our proposed framework consists of two contributions. First, we extend the recent work on pyramid matching [6, 10, 12, 25, 26] and present a new aligned space-time pyramid matching method to effectively measure the distances between two video clips from different domains.

The second is our main contribution. We propose a new cross-domain learning method, referred to as Adaptive Multiple Kernel Learning (A-MKL), in order to cope with the considerable variation in feature distributions between videos from the web domain and consumer domain. Specifically, for each pyramid level and each type of local features, we train a set of SVM classifiers based on a combined training set from two domains by using multiple base kernels of different kernel types and parameters, which are further fused with equal weights to obtain an *average classifier*. We also propose a new objective function to learn an *adapted classifier* based on multiple base kernels and the prelearned average classifiers by minimizing both the structural risk functional and mismatch of data distributions from two domains.

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, coupled HMM, and Dynamic Bayesian Network [20] to model the temporal evolution. Appearance-based approaches employed space-time features extracted from salient regions with significant local variations in both spatial and temporal dimensions [11, 19, 22]. Statistical learning methods including Support Vector Machine (SVM) [22], probabilistic Latent Semantic Analysis (pLSA) [19], and Boosting [8] were then applied to the above space-time features to obtain the final classification. Promising results [1, 13, 19, 22] have been reported on video data sets under controlled settings, such as Weizman [1] and KTH [22] data sets.

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 [10, 24] and broadcast news videos [26]. In [10], Laptev *et al.* integrated local space-time features (*i.e.*, HoG and HoF), space-time pyramid matching and SVM for action classification in movies. Sun *et al.* [24] employed Multiple Kernel Learning (MKL) to efficiently fuse three types of features including SIFT average descriptor and two trajectory-based features. To recognize events in diverse broadcast news videos, Xu and Chang [26] proposed a multi-level temporal matching algorithm for measuring video similarity.

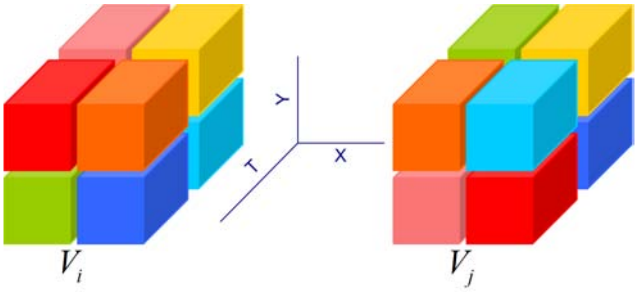


Figure 1. Illustration of aligned space-time pyramid matching at level 1. Two videos V_i and V_j are divided into 8 space-time volumes (the matched volumes by using our method are illustrated with the same colors).

However, all these methods followed the conventional learning framework by assuming that the training and test samples are from the same domain and distribution. When the total number of labeled training samples is limited, the performances of these methods would suffer. In contrast, the goal of this 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 of videos from two domains (*i.e.*, web domain and consumer domain). As a result, our algorithm can learn a robust classifier for event recognition when 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 recognitions, scene classification, and event recognition in movies and news videos [6, 10, 12, 25, 26]. 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 [12] and its space-time extension [10] used fixed block-to-block matching and fixed volume-to-volume matching (we refer to them as *unaligned space-time matching*), respectively. In contrast, our proposed aligned pyramid matching extends the methods of Spatially Aligned Pyramid Matching (SAPM) [25] and Temporally Aligned Pyramid Matching (TAPM) [26] from either spatial domain or temporal domain to joint space-time domain, where the volumes across different space and time locations may be matched.

Similar to [10], we divide each video clip into 8^l non-overlapped space-time volumes over multiple levels, $l = 0, \dots, L-1$, where the volume size is set as $1/2^l$ of the original video in width, height and temporal dimension. Fig. 1 illustrates the partition for two videos V_i and V_j at level-1. Following [10], 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 frames and then extract static local SIFT features from them [17].

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 $r, c = 1, \dots, R$ 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 [10], we use χ^2 distance to measure the distance D_{rc}. Note that each space-time volume consists of a set of image blocks. We also extract toke-frequency (tf) features from each image block by vector-quantizing the corresponding SIFT features into visual words. And based on the SIFT features, as suggested in [26], the pairwise distance D_{rc} between two volumes $V_i(r)$ and $V_j(c)$ is calculated by using Earth Mover's Distance (EMD), *i.e.*, $D_{rc} = \frac{\sum_{u=1}^{H}\sum_{v=1}^{I}\widehat{f}_{uv}d_{uv}}{\sum_{u=1}^{H}\sum_{v=1}^{I}\widehat{f}_{uv}}$, 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 with Integer-flow EMD to explicitly align the volumes. We try to solve a flow matrix $\widehat{F}_{rc}$ containing binary elements that represent unique matches between volumes $V_i(r)$ and $V_j(c)$. As suggested in [25, 26], such binary solution can be conveniently computed by using the standard Simplex method for linear programming. The following Theorem 1 is utilized:

Theorem 1 ([25, 26]). *The linear programming problem*

$$\widehat{F}_{rc} = \arg\min_{F_{rc} \in \{0,1\}} \sum_{r=1}^{R}\sum_{c=1}^{R} F_{rc} D_{rc},$$

$$\textit{s.t.} \sum_{c=1}^{R} F_{rc} = 1, \ \forall r; \ \sum_{r=1}^{R} F_{rc} = 1, \ \forall c,$$

will always have an integer optimum solution when solved with the Simplex method.

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

4.1. Brief review of related learning work

Cross-domain learning methods have been proposed for many applications [4, 5, 15, 27]. To take advantage of all labeled patterns from both auxiliary and target domains, Daumé III [4] proposed Feature Replication (FR) by using augmented features for SVM training. In Adaptive SVM (A-SVM) [27], 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 $f^T(\mathbf{x}) = f^A(\mathbf{x}) + \Delta f(\mathbf{x})$, where $\Delta f(\mathbf{x})$ is the so-called *perturbation function*. While A-SVM can also employ multiple auxiliary classifiers, these auxiliary classifiers are equally fused to obtain $f^A(\mathbf{x})$. Moreover, the target classifier $f^T(\mathbf{x})$ is learned based on only one kernel. Recently, Duan [5] proposed Domain Transfer SVM (DTSVM) to simultaneously reduce the mismatch in the distributions between 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 samples from the auxiliary domain D^A and the target domain D^T in the Reproducing Kernel Hilbert Space (RKHS), namely:

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

where $\mathbf{x}_i^A$'s and $\mathbf{x}_i^T$'s are the samples from the auxiliary and target domains, respectively, and a 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 $\mathbf{s}$ 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. Thus, the MMD criterion in (1) can be simplified to [2, 5]:

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

where $\mathbf{S} = \mathbf{s}\mathbf{s}' \in \Re^{N\times N}$, and $\mathbf{K} = \begin{bmatrix} \mathbf{K}^{A,A} & \mathbf{K}^{A,T} \\ \mathbf{K}^{T,A} & \mathbf{K}^{T,T} \end{bmatrix} \in \Re^{N\times N}$, and $\mathbf{K}^{A,A} \in \Re^{n_A\times n_A}$, $\mathbf{K}^{T,T} \in \Re^{n_T\times n_T}$ and $\mathbf{K}^{A,T} \in \Re^{n_A\times n_T}$ 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. Proposed formulation and solution

Motivated by A-SVM and DTSVM, we propose a new cross-domain learning method, referred to as Adaptive Multiple Kernel Learning (A-MKL), to learn a target classifier adapted from a set of prelearned classifiers as well as a perturbation function which is based on multiple base kernels k_m's. The prelearned classifiers are used as prior for learning a robust adapted target classifier. Specifically, we train a set of independent classifiers for each pyramid level and each type of local features using the training data from two domains. We further equally fuse these classifiers to obtain **average classifiers** $f_l^{SIFT}(\mathbf{x})$ and $f_l^{ST}(\mathbf{x})$, $l = 0, \ldots, L-1$. These classifiers are then used as prelearned classifiers $f_p(\mathbf{x})|_{p=1}^{P}$. In our work, the kernel function k is a linear combination of base kernels k_m's, *i.e.*, $k = \sum_{m=1}^{M} d_m k_m$, where d_m is the linear combination coefficient, and the kernel function k_m is induced from the nonlinear feature mapping function $\varphi_m(\cdot)$, *i.e.*, $k_m(\mathbf{x}_i, \mathbf{x}_j) = \varphi_m(\mathbf{x}_i)'\varphi_m(\mathbf{x}_j)$. Inspired by semiparametric SVM [23], we define the target decision function on any sample $\mathbf{x}$ as follows:

$$f^T(\mathbf{x}) = \sum_{p=1}^{P} \beta_p f_p(\mathbf{x}) + \sum_{m=1}^{M} d_m \mathbf{w}_m' \varphi_m(\mathbf{x}) + b, \qquad (3)$$

where $f_p(\mathbf{x})$'s are the prelearned classifiers trained based on the labeled data from both domains, $\Delta f(\mathbf{x}) = \sum_{m=1}^{M} d_m \mathbf{w}_m' \varphi_m(\mathbf{x}) + b$ is the perturbation function with the bias term b. Let us define the coefficient vector $\mathbf{d} = [d_1, \ldots, d_M]'$ which belongs to $\mathcal{D} = \{\mathbf{d} \in \Re^M | \mathbf{d}'\mathbf{1} = 1, \mathbf{d} \geq \mathbf{0}\}$. In A-MKL, the first objective is to reduce the mismatch in data distributions between two domains. As shown in [5], (2) can be rewritten as:

$$\text{DIST}_k^2(D^A, D^T) = \Omega(\mathbf{d}) = \mathbf{h}'\mathbf{d}, \qquad (4)$$

where $\mathbf{h} = [\text{tr}(\mathbf{K}_1\mathbf{S}), \ldots, \text{tr}(\mathbf{K}_M\mathbf{S})]'$, and $\mathbf{K}_m = [\varphi_m(\mathbf{x})'\varphi_m(\mathbf{x})] \in \Re^{N\times N}$ is the mth base kernel matrix defined on the samples from both auxiliary and target domains.

The second objective of A-MKL is to minimize the structural risk functional. Given the labeled training samples $(\mathbf{x}_i, y_i)|_{i=1}^{n}$ from $D^A \cup D_l^T$, the optimization problem in A-MKL is then formulated as follows:

$$\min_{\mathbf{d}\in\mathcal{D}} \;\; G(\mathbf{d}) = \frac{1}{2}\Omega^2(\mathbf{d}) + \theta\, J(\mathbf{d}), \qquad (5)$$

where

$$J(\mathbf{d}) = \min_{\mathbf{w}_m, \boldsymbol{\beta}, b, \xi_i} \frac{1}{2}\sum_{m=1}^{M} d_m \|\mathbf{w}_m\|^2 + \lambda\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{n}\xi_i, \qquad (6)$$
$$\text{s.t.} \quad y_i f^T(\mathbf{x}_i) \geq 1 - \xi_i,\; \xi_i \geq 0,$$

$\boldsymbol{\beta} = [\beta_1, \ldots, \beta_P]'$ 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 $\boldsymbol{f}(\mathbf{x}_i) = [f_1(\mathbf{x}_i), \ldots, f_P(\mathbf{x}_i)]'$. Let us define $\tilde{\mathbf{v}}_m = d_m\tilde{\mathbf{w}}_m$. The optimization problem in (6) then becomes a quadratic programming (QP) problem [21]:

$$J(\mathbf{d}) = \min_{\tilde{\mathbf{v}}_m, b, \xi_i} \frac{1}{2}\sum_{m=1}^{M} \frac{\|\tilde{\mathbf{v}}_m\|^2}{d_m} + C\sum_{i=1}^{n}\xi_i, \qquad (7)$$
$$\text{s.t.} \quad y_i\left(\sum_{m=1}^{M}\tilde{\mathbf{v}}_m'\tilde{\varphi}_m(\mathbf{x}_i) + b\right) \geq 1 - \xi_i,\; \xi_i \geq 0.$$

By introducing the Lagrangian multipliers $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]'$, the dual of (7) becomes (see [21] for the detailed derivation):

$$J(\mathbf{d}) = \max_{\boldsymbol{\alpha}\in\mathcal{A}} \boldsymbol{\alpha}'\mathbf{1} - \frac{1}{2}(\boldsymbol{\alpha}\circ\mathbf{y})'\left(\sum_{m=1}^{M} d_m\tilde{\mathbf{K}}_m\right)(\boldsymbol{\alpha}\circ\mathbf{y}), \qquad (8)$$

where $J(\mathbf{d})$ is linear in $\mathbf{d}$, $\mathcal{A} = \{\boldsymbol{\alpha} \in \Re^n | \boldsymbol{\alpha}'\mathbf{y} = 0, \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}\}$, $\mathbf{y} = [y_1, \ldots, y_n]'$, $\tilde{\mathbf{K}}_m = [\tilde{\varphi}_m(\mathbf{x}_i)'\tilde{\varphi}_m(\mathbf{x}_j)] \in \Re^{n\times n}$ is defined by the labeled training data from both domains, and $\tilde{\varphi}_m(\mathbf{x}_i)'\tilde{\varphi}_m(\mathbf{x}_j) = \varphi_m(\mathbf{x}_i)'\varphi_m(\mathbf{x}_j) + \frac{1}{\lambda}\boldsymbol{f}(\mathbf{x}_i)'\boldsymbol{f}(\mathbf{x}_j)$. Surprisingly, the optimization problem in (8) 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.

In a similar fashion to [5, 21], we can prove that the optimization problem in (5) is jointly convex with respect to $\mathbf{d}$, $\tilde{\mathbf{v}}_m$, b and ξ_i (we omit the detailed proof due to space limitation). Then, we employ the alternative coordinate descent procedure proposed in [21] to update different variables ($\boldsymbol{\alpha}$ and $\mathbf{d}$) in (5) with (8) iteratively to obtain the globally optimal solution. With a fixed $\mathbf{d}_t$ at the tth iteration, the dual variables $\boldsymbol{\alpha}_t$ can be solved by using LIBSVM. According to [5], $\mathbf{d}$ is updated at iteration $t+1$ by:

$$\mathbf{d}_{t+1} = \mathbf{d}_t - \eta_t\mathbf{g}_t \in \mathcal{D}, \qquad (9)$$

where $\mathbf{g}_t = (\nabla_t^2 G)^{-1}\nabla_t G$ is the updating direction and η_t is the learning rate which can be obtained by standard line-search methods [21]. With respect to $\mathbf{d}_t$ at the tth iteration, $\nabla_t G = \mathbf{h}\mathbf{h}'\mathbf{d}_t + \theta\nabla_t J$ is the gradient of G in (5), where $\nabla_t J$ is the gradient of J in (8). And the Hessian of G is $\nabla_t^2 G = \mathbf{h}\mathbf{h}'$. Note that $\mathbf{h}\mathbf{h}'$ is not full rank. Therefore, we replace $\mathbf{h}\mathbf{h}'$ by $\mathbf{h}\mathbf{h}' + \epsilon\mathbf{I}$ to avoid numerical instability, where ϵ is set as 10^{-4} in the experiments. The whole procedure is summarized in Algorithm 1 as follows:

Algorithm 1 Adaptive Multiple Kernel Learning

```
Initialization: $\mathbf{d} \Leftarrow \frac{1}{M}\mathbf{1}$.
for $t = 1, \ldots, T_{\max}$ do
    1) Solve the dual variables $\boldsymbol{\alpha}_t$ by the dual of SVM using LIBSVM with the kernel matrix $\sum_{m=1}^{M} d_m \tilde{\mathbf{K}}_m$.
    2) Update the base kernel coefficients $\mathbf{d}_t$ using (9).
end for
```

Note that by setting the derivative of the Lagrangian obtained from (6) to zero, we can obtain $\tilde{\mathbf{w}}_m = \frac{\tilde{\mathbf{v}}_m}{d_m} = \sum_{i=1}^{n} \alpha_i y_i \tilde{\varphi}_m(\mathbf{x}_i)$. Thus, with the optimal dual variables $\boldsymbol{\alpha}$ and linear combination coefficients $\mathbf{d}$, the target decision function (3) of our method A-MKL can be rewritten as $f^T(\mathbf{x}) = \sum_{m=1}^{M} d_m \tilde{\mathbf{w}}'_m \tilde{\varphi}_m(\mathbf{x}) + b = \sum_{i=1}^{n} \alpha_i y_i \sum_{m=1}^{M} d_m \tilde{\mathbf{K}}_m(\mathbf{x}_i, \mathbf{x}) + b = \sum_{i=1}^{n} \alpha_i y_i \left(\sum_{m=1}^{M} d_m \mathbf{K}_m(\mathbf{x}_i, \mathbf{x}) + \frac{1}{\lambda} \boldsymbol{f}(\mathbf{x}_i)' \boldsymbol{f}(\mathbf{x}) \right) + b$.

4.3. Differences from related learning work

A-SVM [27] also 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 equally fused in the target classifier, *i.e.*, $f^T(\mathbf{x}) = \frac{1}{P} \sum_{p=1}^{P} f_p^A(\mathbf{x}) + \Delta f(\mathbf{x})$. In contrast, A-MKL learns the optimal combination coefficients β_p's in (3); 2) 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 (3) is based on multiple kernels, and the optimal kernel combination is automatically determined during the learning process; 3) A-SVM cannot utilize the unlabeled data in the target domain. In contrast, the valuable unlabeled data in the target domain is used in the MMD criterion of A-MKL for measuring the distribution mismatch of two domains.

Our work is also different from the prior work of DTSVM [5], 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, thanks to the very few target labeled patterns, we use a set of prelearned classifiers $f_p(\mathbf{x})$'s as the parametric functions, and model the perturbation function $\Delta f(x)$ based on multiple base kernels in order to better fit the target decision function. To fuse multiple prelearned average classifiers from multiple pyramid levels and different types of features, 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 average classifiers as the prior.

MKL methods [9, 21] 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 aligned space-time pyramid matching method. 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) [4], Adaptive SVM (A-SVM) [27] and Domain Transfer SVM (DTSVM) [5], as well as a Multiple Kernel Learning (MKL) method discussed in [5]. For all methods, we train one-versus-all classifiers with a fixed regularization parameter $C = 1$. For performance evaluation, we use the same non-interpolated Average Precision (AP) as in [10, 26]. Mean Average Precision (MAP) is the mean of APs over all the event classes.

5.1. Data set description

Part of the test data set is derived (under a usage agreement) from the Kodak Consumer Video Benchmark Data Set [16], 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 [16] for more details).

In this work, six events "wedding", "birthday", "picnic", "parade", "show", and "sports" are chosen for experiments. We additionally collected new consumer video clips from real users on our own. Similarly to [16], 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 algorithmic design, to watch each video clip rather than just look at the key frames as done in [16]. For video clips in the Kodak consumer data set [16], only the video clips receiving positive labels in their keyframe based annotation are re-examined. We do not additionally annotate the YouTube videos[1] 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

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

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 quality of YouTube videos are 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.

In real-world applications, the labeled samples in the target domain (*i.e.*, consumer video domain) are *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 the 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 rest 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.

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

We compare our proposed aligned space-time pyramid matching method discussed in Section 3 with the fixed volume-to-volume matching method (referred to as unaligned space-time pyramid matching) used in [10]. In [10], the space-time volumes of one video clip are matched with the volumes of the other video at the 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 types of partitions, in which one video clip is divided into 1 and $2 \times 2 \times 2$ space-time volumes, respectively.

For all the videos in the data sets, we extract two types of features. The first one is the local space-time (ST) feature [10], in which 72-dimensional Histograms of Oriented Gradient (HoG) and 90-dimensional Histograms of Optical Flow (HoF) are extracted by using the online tool[2]. 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 [17]. On 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.

We use the baseline SVM classifier based on the combined training data set from two domains (consumer domain and web domain). 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 $\gamma = \gamma_0 = \frac{1}{A}$ with A being the mean value of square distances between all training samples, as suggested in [10].

Tables 1 and 2 show the MAPs for SIFT and ST features at different levels. 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 [10, 12, 25, 26]; 2) At level-1, our proposed aligned space-time pyramid matching method outperforms the unaligned space-time pyramid matching method used in [10], 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 [7], 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. Comparisons of cross domain learning methods

We also evaluate the performance of our proposed cross-domain learning method discussed in Section 4. In this experiment, we make use of 20 base kernels from four kernel types (*i.e.*, Gaussian kernel, Laplacian kernel, ISD kernel and ID kernel) and five kernel parameters. We set $\gamma = 4^{n-1}\gamma_0$, where $n \in \{-2, -1, \ldots, 2\}$, $\gamma_0 = \frac{1}{A}$ is the default kernel parameter. In total, we have 80 kernels from two pyramid levels, two types of local features, and 20 base kernels.

All methods are compared in three cases: (a) classifiers learned based on SIFT features; (b) classifiers learned based on ST features; and (c) classifiers learned based on both SIFT and ST features. For SVM_AT and FR (*resp.* SVM_T), we train 20 independent classifiers 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 the corresponding 20 base kernels, which are further fused with equal weights to obtain the average classifier f_l^{SIFT} or f_l^{ST}, l=0, 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) and

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

Table 1. Means and standard deviations (%) of MAPs 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 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

$\frac{1}{4}\left(f_0^{SIFT} + f_1^{SIFT} + f_0^{ST} + f_1^{ST}\right)$ for case (c)). For A-SVM, we learn 20 independent auxiliary classifiers for each pyramid level and each type of local features using the training data from the auxiliary domain and the corresponding 20 base kernels, and then we independently learn four adapted target classifies using the labeled training data from the target domain based on Gaussian kernel with the default kernel parameter [27]. The final classifier is obtained by fusing four adapted target classifiers. For MKL and DTSVM, we simultaneously learn the linear combination coefficients of 40 base kernels (for cases (a) or (b)) or 80 base kernels (for case (c)) by using the combined training samples from both domains. For our method A-MKL, we learn an adapted classifier based on two average classifiers $f_l^{SIFT}|_{l=0}^1$ or $f_l^{ST}|_{l=0}^1$ (for cases (a) and (b)), or all the four average classifiers (for case (c)) as well as multiple base kernels (40 base kernels for cases (a) and (b), and 80 base kernels for case (c)). For A-MKL, we empirically fix $\theta = 10^{-4}$ and set $\lambda = 10$ 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 (4). Note that all test samples are used as unlabeled data during the learning process.

In Fig. 2 and Table 3, we compare our proposed method A-MKL with SVM_T, SVM_AT, FR, A-SVM, MKL and DTSVM. We have the following observations:

1) The best result of SVM_T is worse than that of SVM_AT, which demonstrates that the learned SVM classifiers based on a limited number of training samples from the target domain are not robust. We also observe that SVM_T is better than SVM_AT for cases (b) and (c). A possible explanation is that the ST feature is not robust enough so that the samples from auxiliary domain and target domain distribute sparsely in this feature space. Therefore, it is more likely that the data from the auxiliary domain may degrade the event recognition accuracies for cases (b) and (c).

2) In this application, A-SVM achieves the worst results in terms of the means of MAPs in three cases, possibly because the limited number of labeled training samples in the target domain (*e.g.*, 3 samples per event) are not sufficient for A-SVM to robustly learn an adapted target classifier, which is only based on one Gaussian kernel.

3) Similarly to the prior work [5], DTSVM outperforms MKL in almost all cases in terms of the means of per-event APs in Fig. 2. And DTSVM is also better than MKL in terms of the means of MAPs in Table 3. This is consistent with [5].

4) For all methods, the MAPs based on SIFT features are better compared with ST features. In practice, two simple ensemble methods, SVM_AT and FR, achieve 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 method using equal weights are not robust for integrating strong features and weak features. In contrast, for MKL, DTSVM and our method, the results in case (c) are improved by learning optimal linear combination coefficients to effectively fuse two types of features.

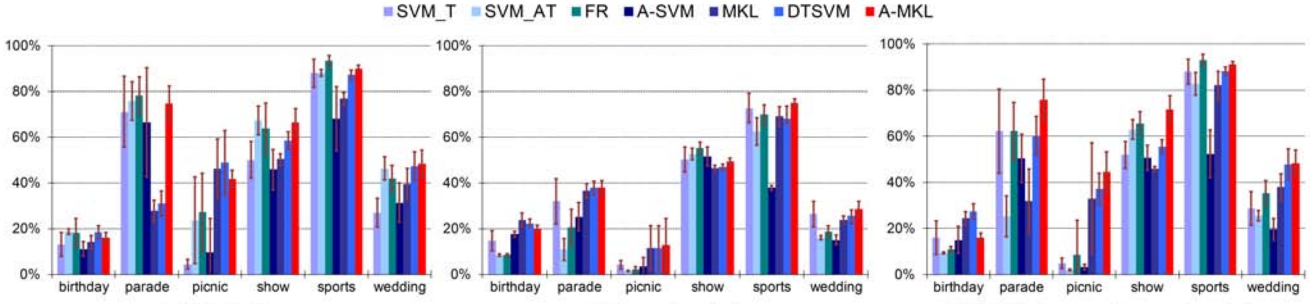
5) Our proposed method A-MKL achieves the best MAPs in all three cases by effectively fusing four average classifiers (from two pyramid levels and two types of local features) and multiple base kernels as well as reducing the mismatch in the data distributions between two domains. We also believe the utilization of multiple base kernels and prelearned average classifiers can also well cope with noisy YouTube videos. In case (c), our method achieves the best performances in 4 out of 6 events and some concepts enjoy large performance gains according to the means of Per-event APs. Compared with the best means of MAPs of SVM_T (42.3%), SVM_AT (53.3%), FR (53.8%), A-SVM (38.7%), MKL (42.5%) and DTSVM (52.7%), the relative improvements of our best result (57.9%) are 36.9%, 8.6%, 7.6%, 49.6%, 36.2% and 9.9%, respectively.

6. Conclusion

In this paper, we propose a new event recognition framework for consumer domain videos by leveraging a large amount of loosely labeled YouTube videos. Specifically, we propose a new aligned space-time pyramid matching method and a novel cross-domain learning method to better fuse the information from multiple pyramid levels and different types of local features and to cope with the mismatch in data distribution of consumer video domain and web video domain. 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.

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

	SVM_T	**SVM_AT**	**FR**	**A-SVM**	**MKL**	**DTSVM**	**A-MKL**
MAP-(a)	42.3 ± 5.2	53.3 ± 4.4	53.8 ± 1.8	38.7 ± 7.6	42.4 ± 2.4	48.5 ± 2.7	56.2 ± 2.7
MAP-(b)	33.4 ± 1.3	25.3 ± 0.5	29.2 ± 1.5	25.1 ± 0.7	35.2 ± 1.5	35.3 ± 1.0	37.2 ± 2.0
MAP-(c)	42.0 ± 4.9	34.6 ± 1.4	46.0 ± 1.6	31.9 ± 4.4	42.5 ± 4.6	52.7 ± 2.4	$\mathbf{57.9 \pm 1.7}$



(a) SIFT features

(b) Space-time features

(c) SIFT features and space-time features

Figure 2. Means and standard deviations of per-event APs of six events for all methods.

Acknowledgements: This work is funded by Singapore A*STAR SERC Grant (082 101 0018) and MOE AcRF Tier-1 Grant (RG15/08).

References

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[2] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. Smola. Integrating structured biological data by kernel maximum mean discrepancy. In *ISMB*, 2006.

[3] 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. In *MIR*, 2007.

[4] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007.

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

[6] K. Grauman and T. Darrell. The pyramid match kernel: Discriminantive classification with sets of image features. In *ICCV*, 2005.

[7] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang. Action detection in complex scenes with spatial and temporal ambiguities. In *ICCV*, 2009.

[8] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, 2005.

[9] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 5:27–72, 2004.

[10] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[11] L. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003.

[12] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[13] Z. Lin, Z. Jiang, and L. S. Davis. Recognizing actions by shape-motion prototype trees. In *ICCV*, 2009.

[14] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009.

[15] Y. Liu, D. Xu, I. W. Tsang, and J. Luo. Textual Query of Consumer Photos Facilitated by Large-scale Web Data. To appear in *T-PAMI*, 2010.

[16] A. C. Loui *et al.* Kodak's consumer video benchmark data set: Concept definition and annotation. In *MIR*, 2007.

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

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

[19] J. Niebles, H. Wang, and F. Li. Unsupervised learning of human action categories using spatial temporal words. In *BMVC*, 2005.

[20] N. Oliver, B. Rosario, and A. Pentland. A bayesian computer vision system for modeling human interactions. *T-PAMI*, 22(8):831–843, 2000.

[21] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *JMLR*, 9:2491–2521, 2008.

[22] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions, a local svm approach. In *ICPR*, 2004.

[23] A. J. Smola, T. T. Frieß, and B. Schölkopf. Semiparameteric support vector and linear programming machines. In *NIPS*, 1999.

[24] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li. Hierarchical spatio-temporal context modeling for action recognition. In *CVPR*, 2009.

[25] D. Xu, T.-J. Cham, S. Yan, and S.-F. Chang. Near duplicate image identification with spatially aligned pyramid matching. In *CVPR*, 2008.

[26] D. Xu and S.-F. Chang. Video event recognition using kernel methods with multi-level temporal alignment. *T-PAMI*, 30(11):1985–1997, 2008.

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