Compositional Models for Video Event Detection: A Multiple Kernel Learning Latent Variable Approach

Arash Vahdat, Kevin Cannons, Greg Mori
Simon Fraser University, Canada
{avahdat, kcannons, mori}@sfu.ca

Sangmin Oh, and Ilseo Kim
Kitware Inc., USA
{sangmin.oh, ilseo.kim}@kitware.com

Abstract

We present a compositional model for video event detection. A video is modeled using a collection of both global and segment-level features and kernel functions are employed for similarity comparisons. The locations of salient, discriminative video segments are treated as a latent variable, allowing the model to explicitly ignore portions of the video that are unimportant for classification. A novel, multiple kernel learning (MKL) latent support vector machine (SVM) is defined, that is used to combine and re-weight multiple feature types in a principled fashion while simultaneously operating within the latent variable framework. The compositional nature of the proposed model allows it to respond directly to the challenges of temporal clutter and intra-class variation, which are prevalent in unconstrained internet videos. Experimental results on the TRECVID Multimedia Event Detection 2011 (MED11) dataset demonstrate the efficacy of the method.

1. Introduction

Multimedia event detection in unconstrained video collections is a challenging problem. Event categories are diverse and exhibit large intra-class variation. Additionally, videos may be composed of a small number of important segments, while the remaining portions of the video are ineffective for classification.

Consider the example video from the *board trick* category in Fig. 1. This video contains segments focusing on the snowboard, the person jumping, is shot in an outdoor, ski-resort scene, and has fast-paced theme music. Together, all of these pieces of evidence can lead an algorithm to declare that this video is from the relevant category.

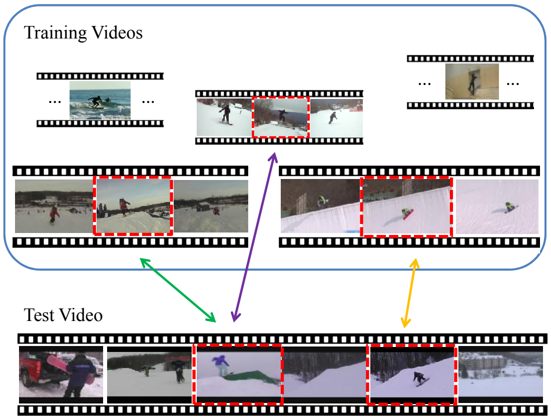

Figure 1: A test video can be described using pieces of similar training videos. Similarity might be defined from different perspectives. In this example, parts of the test video from the *board trick* event are similar to three different videos in terms of motion and sound (green), pure motion (purple) or motion and texture (yellow).

Building a model that can correctly categorize this type of video is challenging. Arguably, such a model must reason about which temporal segments within the video contain relevant evidence. Additionally, grouping these segments into different mid-level categories, or "scene types" may be beneficial. For the *board trick* event, a particular video may involve a surfboard, skateboard, or snowboard trick, but is unlikely to include all three. Grouping segments into their relevant scene types can improve recognition. Finally, the model must utilize a variety of different low-level features in order to make such a decision.

In this paper we present a novel, *compositional* model for video event detection. Our model uses a latent variable framework to localize the discriminative temporal segments of a video. These temporal segments are matched to training segments of the same scene type via kernels that combine information from several feature modalities. The test video is explained as a composition of related training videos.

The main contribution of this paper is the theoretical development of a formulation and learning algorithm for this

This work was supported by NSERC and the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20069. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/NBC, or the U.S. Government.

type of model. The proposed compositional method has two key novel aspects: (1) a weakly supervised method for localizing only the most salient evidence for classification in a video sequence. This method does not require manual marking of the salient segments – they are automatically extracted and labeled by scene type. (2) A novel multiple kernel learning algorithm with structured latent variables that permits the principled combination of multiple different low-level features in a single integrated framework.

2. Previous Work

Event detection in unconstrained internet videos is an active area of research. We consider the TRECVID MED11 dataset – a large, diverse, and challenging video collection. Among the top ranking methods on this dataset is the work of Natarajan et al. [7], which performs a principled combination of many low-level features using a global, video-level representation. It is arguable that engineering a combination of many complementary low-level features is necessary for excellent performance on this dataset, and the method we propose can be used with a multitude of features in this manner. Furthermore, our multiple kernel learning algorithm offers an extension that allows for such feature combination in conjunction with latent SVMs. With this novel approach, more detailed comparisons between latently selected video segments can be considered.

Other video classification work includes Niebles et al. [8], who developed a related model for human action recognition, but used a fixed, single temporal ordering of key poses around anchor points – which may break down in internet videos due to temporal clutter. Tang et al. [12] extended this line of work to consider temporal segmentation via a variant of an HMM. Cao et al. [1] considered a "scene aligned pooling" feature representation to capture the different scenes present in a single video. In contrast to the above, our method focuses on intra-class variation and temporal scatter of an event by using latent variables to compose a test video in a kernelized framework. In direct comparisons, we show empirically that our approach outperforms these previous methods.

The approach we take to modeling internet videos is weakly supervised – only a video-level category label is provided during training. Segments and their associated scene types that compose a video are learned in an unsupervised fashion. Izadinia and Shah [4] developed a similar method, but with manual annotations on the training data – extending the image-attribute method of Wang and Mori [17] to the video domain.

Technically, the proposed approach is most closely related to [18, 20, 3], but differentiates itself by presenting a novel multiple kernel learning approach that accommodates structured latent variables. In comparison, Wu and Jia [18] and Yang et al. [20] developed kernelized variants of the latent support vector machine [2, 21]. However, the algorithms for learning kernelized latent SVMs in these papers have two drawbacks: they are limited to cases where one can enumerate the set of latent variables and they are restricted to a single kernel or a set of summed kernels. Finally, Gu et al.[3] consider low level concept detection (e.g. flag, car, building) using a bag-instance relationship whereas ours examines high-level event recognition.

Kernelized classifiers often offer superior performance. A body of work has aimed at providing efficient training and evaluation with kernelized classifiers via algorithmic optimizations or additive linear approximations [15, 6, 10]. This line of work is promising, but has yet to be extended to latent variable models, as is done here.

3. Compositional Models for Video Retrieval

We are interested in the classification of high-level complex events in unconstrained internet videos. Two significant challenges in this domain are temporal clutter (i.e., the evidence of a complex event can occur in small, isolated video segments) and intra-class variation. In this paper, we target both the intra-class variation and temporal clutter challenges by leveraging a compositional model.

Early successes on the TRECVID MED11 dataset have often deferred to an approach where the output of an array of simple classifiers operating on a range of low-level features are combined [7]. These approaches have tended to employ simple, bag of words (BoW) representations with kernelized SVM classifiers. In such systems, the standard kernelized SVM can be thought of as a form of intelligent template matching, whereby a test video is compared directly against the set of support vectors. Such approaches can perform effective matching on global video-level representations, but are not well-suited for segment-level analysis. By introducing latent variables in our proposed method, kernelized latent SVMs are constructed that select particularly salient video segments. Thus, this intelligent template matching can now be completed not only at the video level, but also at the segment level. This approach provides our compositional model with the additional flexibility to mix and match segments from the pool of training videos when evaluating a test video, directly addressing the challenges of clutter and intra-class variation.

Additionally, to attain state-of-the-art performance on TRECVID MED11, it appears that multiple feature types must be combined. We further extend our model to combine multiple kernel learning with the kernelized latent SVM framework, adding the ability to weight feature types based on their relative importance.

3.1. Linear Model

To begin the exposition we describe the linear version of our model, which consists of two parts. The first part is a global model that captures the overall theme or "***subcategory***" of the video. It is assumed that each event category contains several subcategories (e.g., a wedding

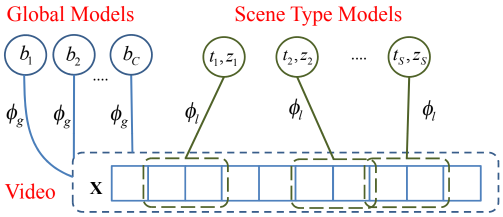

Figure 2: Depiction of our proposed model. The global model captures the subcategories of an event, and the scene model represents the different scene types observed in the category. The presence of a subcategory or scene type is represented using binary variables (b_c, z_s). The temporal position of scene types in a video is denoted by t_s.

ceremony at a church, house, or park). Further, it is assumed that a particular video corresponds to only one subcategory. The second part of our formulation is a "***scene type model***" that represents an event by a set of segment-level features. This part of the model is included to identify and localize discriminative segments of interest in a video. The model is depicted graphically in Fig. 2.

We consider eight second segments that correspond to scenes observed within the event category (e.g., for *wedding ceremony* videos, outdoor park scenes or people dancing, cutting a cake, or kissing). A weakly supervised setting is considered, meaning that we are only given a binary event label for each video that indicates the presence of a complex event in the sequence; the subcategory labels, scene type labels, and temporal locations of scene types are not provided. These are modeled as hidden variables and we employ a latent max-margin approach [2] to infer them during training.

Concretely, assume we are given a video sequence x, and want to classify it into an event category. The variables C and S denote the number of subcategories and scene types for an event, respectively. The presence of a subcategory $c \in \{1, 2, \dots, C\}$ is defined using the binary variable b_c; similarly, the presence of a scene type $s \in \{1, 2, \dots, S\}$ is denoted using the binary variable z_s.

We define $\phi_g(x)$, a global feature extracted from the whole sequence, and $\phi_l(x, t)$ a segment-level feature extracted from a temporal window of fixed size centered at time t in x. Multiple features are incorporated to improve accuracy: G global and L local (segment-level) features. Together, the linear version of our model is defined as:

$$f_w(x, \mathbf{b}, \mathbf{h}) = \sum_{c=1}^{C} \sum_{g=1}^{G} w_{cg}^T \phi_g(x) b_c + \sum_{s=1}^{S} \sum_{l=1}^{L} w_{sl}^T \phi_l(x, t_s) z_s \quad (1)$$

where w_{cg} is the learned weight vector for the c^{th} subcategory model on the global feature $\phi_g(\cdot)$, and w_{sl} is the weight vector for the s^{th} scene type model defined on the segment-level feature $\phi_l(\cdot)$. Use of the same set of feature types in the global and segment-level scales can be achieved by setting $G = L$. However, more generally, our model supports the added flexibility of using different sets of features for the two parts. For notational compactness, we represent the pair (t_s, z_s) using h_s for $s \in \{1, 2, ..., S\}$, and group them in vector $\mathbf{h} = \{h_1, h_2, ..., h_S\}$. We similarly group subcategory binary variables in $\mathbf{b} = \{b_1, b_2, ..., b_C\}$.

Note that the model in Eq. 1 assumes the temporal location for the s^{th} scene type is shared among all segment-level features types – they are all extracted from the same temporal window in the sequence.

It is assumed that a sequence can belong to only one global subcategory, but multiple scene types might be observed in a sequence, corresponding to the various segments. Therefore, two hard constraints are imposed on the selecting binary variables: $\sum_{c=1}^{C} b_c = 1$, and $\sum_{s=1}^{S} z_s = K$, where K is a constant parameter.

The subcategory variables, b_c, and scene model configurations, h_s, are latent variables, unobserved on both training and testing data. Next, we develop a novel multiple kernel learning approach for learning with these latent variables.

3.2. Multiple Kernel Latent SVM

Latent SVMs have been successfully used in many computer vision tasks. They were originally proposed for linear models [21, 2], where the similarity of two samples is measured using a simple dot product. Recently, LSVMs were extended to kernelized versions [20, 18] resulting in significant boosts in recognition accuracy. However, both [20, 18] assumed simple models with few latent variables that could be enumerated during inference. In our proposed model, latent variables are defined in a structured framework such that enumeration is not tractable.

The use of multiple complementary features can lead to improved recognition accuracy. With multiple features, fusion is a challenge because the importance of feature types is variable. Multiple kernel learning is a standard approach to address this challenge. A linear MKL SVM framework (e.g., [16]) typically performs such fusion by linearly combining a set of kernels $K = \sum_i d_i K_i$, which corresponds to re-scaling feature maps of the kernel, Ψ_i, by $\sqrt{d_i}$.

The linear model in Eq. 1 is also defined with respect to multiple features. We require a training framework that can accommodate both latent variables and feature re-scaling simultaneously. We propose a novel multiple kernel latent SVM framework that extends standard MKL and can be used to train models of the form proposed in this paper.

Consider a set $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ of training videos where $x_i \in \mathcal{X}$ is the i^{th} video and $y_i \in \{-1, 1\}$ its label. Our goal is to learn a scoring function $F : \mathcal{X} \rightarrow \mathcal{R}$ that can be used to classify a video. Similar to the standard latent SVM, the proposed multiple kernel latent SVM (MKL-KLSVM[1]) operates upon a set of base feature maps, $\Psi_i(x, \mathbf{v})$, defined on a sample x and its latent variables $\mathbf{v} \in \mathcal{V}$, where $\mathcal{V}$ is the set of all

[1] We use MKL-KLSVM for Multiple Kernel Latent SVM to prevent confusion with Multiple Kernel Learning SVM (MKL SVM)

possible latent variables. We define the scoring function $F(x) = \max_{\mathbf{v}} \sum_{i=1}^{I} \sqrt{d_i} w_i^T \Psi_i(x, \mathbf{v})$ where d_i is the normalizing factor for the i^{th} base feature map. Training of the MKL-KLSVM is then formulated as:

$$\min_{w,b,\xi\geq 0,d\geq 0} \frac{1}{2} \sum_i w_i^T w_i + \rho \sum_n \xi_n + \frac{\lambda}{2} \sum_i d_i^2 \tag{2}$$
$$s.t. \quad y_n(\max_{\mathbf{v}\in\mathcal{V}_n} \sum_i \sqrt{d_i} w_i^T \Psi_i(x_n, \mathbf{v}) + b) \geq 1 - \xi_n \quad \forall n,$$

where λ is a regularizer on the kernel weights, d_i to prevent them from diverging to infinity, and ρ is a trade-off parameter to penalize error on the training data. Note that our multiple kernel latent SVM framework becomes a standard latent SVM [2] if the kernel coefficients, d_i, are set to one and will become a standard MKL classifier if the hidden variables $\mathbf{v}_n$ are observed.

The objective function in Eq. 2 is not convex; however, convexity is attained if the latent variables for positive samples are available (semi-convexity of latent SVM [2]) and if w_i is replaced with $\sqrt{d_i} w_i$. Here we limit the possible latent variables of positive samples to a single configuration $\mathcal{V}_n = \{\mathbf{v_n^*}\} \quad \forall n : y_n = 1$, but allow negative samples to consider all possible latent variables, $\mathcal{V}_n \quad \forall n : y_n = -1$. Given that the latent variable configuration has been specified, the max operator can be omitted from Eq. 2, yielding,

$$\min_{w,b,\xi\geq 0,d\geq 0} \frac{1}{2} \sum_i \frac{w_i^T w_i}{d_i} + \rho \sum_n \xi_n + \frac{\lambda}{2} \sum_i d_i^2 \tag{3}$$
$$s.t. \quad y_n(\sum_i w_i^T \Psi_i(x_n, \mathbf{v}) + b) \geq 1 - \xi_n \quad \forall n, \forall \mathbf{v} \in \mathcal{V}_n$$

The objective function in Eq. 3 addresses the problem of learning parameters of a structural SVM with multiple kernels. It has $N^- |\mathcal{V}| + N^+$ constraints, where N^- and N^+ are the number of negative and positive samples respectively. If the latent variables are structured, $|\mathcal{V}|$ will be exponential. The same problem of exponential constraints is confronted with linear latent SVMs as well. Yu and Joachims [21] use the cutting plane algorithm [13] to ameliorate this challenge by mining hard constraints and iteratively optimizing with and updating the current constraints.

We use the cutting plane algorithm to extract the set of most violated constraints for negative samples during training, while the latent variables of positive videos remain fixed. Here, $\tilde{\mathcal{V}}_n$ denotes the set of current active constraints (instead of $\mathcal{V}_n$, which represents all the constraints defined over all possible latent variables). The set of active constraints, $\tilde{\mathcal{V}}_n$, contains just a single constraint per positive sample, but can have multiple constraints for negative samples, extracted using the cutting plane algorithm.

Given a current set of constraints, a method is required for optimizing Eq. 3. By forming the Lagrangian of Eq. 3 and minimizing the objective function with respect to w_i, ξ and b, we obtain

$$w_i = d_i \sum_{n,\mathbf{v}\in\tilde{\mathcal{V}}_n} \alpha_{n,\mathbf{v}} y_n \Psi_i(x_n, \mathbf{v}) \tag{4}$$

where $\alpha_{n,\mathbf{v}}$ is the Lagrangian variable for the n^{th} sample and the latent variables, $\mathbf{v}$. Substituting w_i in Eq. 3 yields

$$\min_{d\geq 0} \max_{\alpha} L(\alpha, d) = \sum_{n,\mathbf{v}} \alpha_{n,\mathbf{v}} + \frac{\lambda}{2} \sum_i d_i^2 \tag{5}$$
$$-\frac{1}{2} \sum_i d_i \left[\sum_{n,\mathbf{v}'} \sum_{m,\mathbf{v}'} \alpha_{n,v} \alpha_{m,v'} y_n y_m \Psi_i(x_n, \mathbf{v})^T \Psi_i(x_m, \mathbf{v}') \right]$$
$$s.t. \quad 0 \leq \sum_{n,\mathbf{v}} \alpha_{n,v} \leq \rho, \; \sum_{n,\mathbf{v}} y_n \alpha_{n,v} = 0,$$

which is an instance of the saddle point problem. In Eq. 5, $\Psi_i(x_n, \mathbf{v})^T \Psi_i(x_m, \mathbf{v}')$ can be replaced with a kernel $k(x_n, \mathbf{v}, x_m, \mathbf{v}')$ that measures the similarity of x_n and x_m, given their latent configurations. If the kernel weights, d, are fixed in Eq. 5, the inner maximization will become the Quadratic Program (QP) of a kernelized structural SVM [13]. We solve the saddle point problem by iteratively updating d and subsequently performing QP optimization for α with a fixed d. The kernel weights can be updated using a Newton descent step or the cutting plane approach [5]. Alternatively, the Lagrangian of Eq. 5 can be derived to form the dual problem, which is differentiable and can be optimized using the sequential minimal optimization (SMO) algorithm [11], similar to [16].

Here, we elect to use the simple Newton descent approach. Given the optimum, α^*, from iteration τ, in iteration $\tau+1$ an update is computed as $d^{\tau+1} = d^{\tau} - \mu H^{-1} \nabla L$, where $\mu = \frac{1}{\tau}$ is the step size. Additionally, $H = \lambda I$ is the Hessian matrix of $L(\alpha^*, d)$ (I is the identity matrix), and $\nabla L_i(\alpha^*, d) = \lambda d_i^{\tau} - \frac{1}{2} \| \sum_{n,\mathbf{v}} y_n \alpha_{n,v}^* \Psi_i(x_n, \mathbf{v_n}) \|^2$ is the the derivative of L with respect to d^{τ}. If a Newton descent update results in a negative kernel weight, it is back projected using $d_i^{\tau+1} = 0$ if $d_i^{\tau+1} < 0$.

After updating the kernel weights, the inner quadratic program in Eq. 5 is solved by assuming d is fixed. We iterate between these two steps until the optimization converges and the objective function does not change. Given the final α^* and d^* (which together represent w), we infer the latent variables on the positive examples using $\mathbf{v}_n^* = \arg\max_{\mathbf{v}} \sum_i w_i^T \Psi_i(x_n, \mathbf{v})$. It has been shown for standard linear latent SVMs that iteratively updating the latent variables of positive samples and learning the latent SVM model parameters will minimize the objective function to a local optimum [21, 2]. The same argument holds for multiple kernel latent SVM. Algorithm 1 provides a summary of our proposed training algorithm.

3.3. Kernelized Model

We use multiple kernel latent SVM to train the parameters of our model defined in Eq. 1. However, we still must define $\Psi_i(x, \mathbf{v})$, the base features, and their corresponding kernels that have an associated re-scaling coefficient d_i as in Eq. 2. For the linear model defined in Eq. 1 global models were defined on G global features while scene type models employed L segment-level feature types. Specifically, the

Algorithm 1 Training a multiple kernel latent SVM

```
Input : $\{(x_1, y_1), (x_2, y_2) \ldots, (x_N, y_N)\}$
Output : $\alpha^*$, $d^*$
$\tilde{\mathcal{V}}_n = \{\mathbf{v}_n^0\} \; \forall n : y_n = 1, \tilde{\mathcal{V}}_n = \{\} \; \forall n : y_n = -1$
repeat
  repeat
    Optimize Eq. 3 using iterative Newton descent and QP given the current $\tilde{\mathcal{V}}_n$
    $\forall n : y_n =$ -1 add the most violated constraint to $\tilde{\mathcal{V}}_n$
  until no change in objective function of Eq. 3
  $\forall n : y_n = 1$ update $\tilde{\mathcal{V}}_n = \arg\max_{\mathbf{v}} \sum_i w_i^T \Psi_i(x_n, \mathbf{v})$
until no change in $\tilde{\mathcal{V}}_n \forall n : y_n = 1$
```

base features in Eq. 2, Ψ_i, are defined as $\sum_{c=1}^{C} \phi_g(x) b_c$ for the global features and $\sum_{s=1}^{S} \phi_l(x, t_s) z_s$ for the segment-level features, which are derived from Eq. 1. Thus, $G + L$ kernels are defined as

$$K_g(x, \mathbf{b}, x', \mathbf{b}') = \sum_{c=1}^{C} b_c k_g(x, x') b'_c,$$
$$K_l(x, \mathbf{h}, x', \mathbf{h}') = \sum_{s=1}^{S} z_s k_l(x, t_s, x', t'_s) z'_s. \qquad (6)$$

Given two videos, x and x', K_g measures the kernelized similarity of their global feature if they belong to the same subcategory; otherwise, it assigns zero similarity. Analogously, K_l measures the kernelized similarity of segment-level feature l for sequences x and x' at times t_s and t'_s for the scene models that are present in both x and x'.

Given the kernels defined in Eq. 6, Alg. 1 is used to learn α^* and d^*, the parameters of the proposed kernelized model. We can substitute these parameters in Eq. 1 to rewrite our scoring function for the kernelized model:

$$F(x) = \max_{\mathbf{b},\mathbf{h}} \Big[\sum_{n,(\mathbf{h}_n,\mathbf{b}_n)} \sum_{g=1}^{G} \alpha^*_{n,(\mathbf{h}_n,\mathbf{b}_n)} y_n d^*_g K_g(x_n, \mathbf{b}_n, x, \mathbf{b})$$
$$+ \sum_{n,(\mathbf{h}_n,\mathbf{b}_n)} \sum_{l=1}^{L} \alpha^*_{n,(\mathbf{h}_n,\mathbf{b}_n)} y_n d^*_l K_l(x_n, \mathbf{h}_n, x, \mathbf{h}) \Big], \qquad (7)$$

where $(\mathbf{h}_n, \mathbf{b}_n) \in \tilde{\mathcal{V}}_n$ are latent variables defined for the n^{th} training sample.

The completed model in Eq. 7 is the full, proposed compositional model. Given the sequence, x, maximization matches the sequence to the training videos by choosing segment locations, $\mathbf{h}$, and the subcategory model, $\mathbf{b}$, that are well-explained by the training videos. A test video, x, is assigned a high score for an event category if it is similar to its associated positive training videos using two criteria. First, the global features from the test video should be similar to the global features from training videos. Second, the test video should contain segments that are similar to those in the training set. Under this framework, the test video can be composed using components from numerous training videos at both the global and segment scale. The learned kernel coefficients, d, allow for the re-scaling of the similarity measures on different parts of model. This rescaling can give higher weights to important feature types while allowing for the extraction of the most discriminative evidence from the training set, using $(\mathbf{h}_n, \mathbf{b}_n)$.

3.4. Implementation Details

Simple heuristics are used to initialize the latent variables for the positive samples. For the subcategory labels, we cluster the concatenated global features of the positive videos into C clusters. Subsequently, we assign a video to the closest cluster. For the scene models, we similarly cluster the concatenated segment-level features of all segments from the positive training videos. Then, we choose the K closest clusters to the video segments, and set the temporal location of each, t_s, to the closest segment.

Inference: For inferring latent variables, we first need to compute the global and scene model scores for each subcategory and scene type. For a general kernel type, there is no explicit form of w_i and direct comparison to support vectors is necessary to compute the scores. Kernel comparison can significantly slow down the inference. Given N_s support vectors, considering Eq. 7, Eq. 6 and sparsity of b_n and z in $\mathbf{h}_n$, $O(N_s G + N_s KLT)$ kernel comparisons will be required to compute the scores for a sequence. However, with additive kernels we can approximate the embedding feature [14], and form an approximated w_i using Eq. 4. Thus, the number of linear kernel computations becomes $O(CG + SLT)$.

Consider the model in Fig 1. Now, given global and scene type model scores, we need to infer the subcategory variables b_c and temporal locations t_s of the K best scene type models. The subcategory can be found in $O(C)$. For a video with T segments, the best location for each scene type is found in $O(T)$, and then the K best scenes are selected in $O(S \log(K))$ using a min heap. So, the complexity of inference is $O(C + ST + S\log(K))$ in addition to the score computation. In our experiments, this inference takes 0.05 seconds for a 120-second video on an Intel CPU E7450 @2.40GHz.

4. Experiments

We evaluate our model on the challenging TRECVID MED11 dataset [9], following a standard evaluation protocol used in previous work [12]. The TRECVID MED11 dataset contains 15 events that are divided across two collections, DEV-T and DEV-O. The DEV-T dataset consists of 10,723 videos including videos from five event categories: *board trick (E1)*, *feeding animal (E2)*, *landing fish (E3)*, *wedding ceremony (E4)*, and *woodworking project (E5)*. The DEV-O collection is significantly larger, 32,061 videos, and includes ten categories: *birthday party (E6)*, *changing a tire (E7)*, *flash mob (E8)*, *getting a vehicle unstuck (E9)*, *grooming animal (E10)*, *making sandwich*

Table 1: Performance variation on the DEV-T dataset as a function of model parameters: the number of subcategories (C), number of scene types (S), and number of selected scenes (K). Selection is done for each parameter in turn and is fixed for subsequent parameters, as shown in red.

Model Settings	*E1*	*E2*	*E3*	*E4*	*E5*	mAP
$C = 1, S = 0$	14.2	3.8	16.7	34.4	8.4	15.5
$C = 2, S = 0$	14.1	3.9	17.6	35.8	8.5	16.0
$C = 4, S = 0$	14.3	3.7	16.8	34.3	13.7	16.6
$C = 8, S = 0$	13.8	3.8	18.3	40.7	16.6	18.6
$C = 16, S = 0$	12.1	3.9	17.3	38.8	15.1	17.4
$C = 8, S = K = 4$	12.3	2.8	24.0	44.4	13.3	19.4
$C = 8, S = K = 8$	11.1	2.6	25.3	44.6	12.8	19.2
$C = 8, S = K = 16$	13.3	2.3	26.8	43.9	14.8	20.2
$C = 8, S = K = 32$	13.1	2.1	27.2	44.6	14.3	20.2
$C = 8, S = 16, K = 1$	15.3	3.3	20.1	42.3	16.6	19.5
$C = 8, S = 16, K = 2$	14.8	3.4	24.1	46.1	18.4	21.4
$C = 8, S = 16, K = 4$	17.4	3.2	26.3	46.3	17.5	22.1
$C = 8, S = 16, K = 8$	12.8	2.9	29.0	48.5	17.9	22.2
$C = 8, S = 16, K = 16$	13.3	2.3	26.8	43.9	14.8	20.2

(E11), *parade (E12)*, *parkour (E13)*, *repairing appliance (E14)*, and *sewing project (E15)*. Both DEV-T and DEV-O are dominated by videos of the null category (i.e., background videos that do not contain the events of interest). For training, an Event-Kit data collection, containing roughly 150 positive videos per category, is also provided. A classifier is trained for each event category versus all other categories, similar to [12].

For TRECVID MED11, DEV-T is used for development, whereas DEV-O is utilized for testing. Thus, we performed cross validation of all system parameters and hyper parameters on DEV-T and held them constant when considering DEV-O. We use mean average precision (mAP) as the performance metric to remain comparable with recently published works [1, 12].

4.1. Comparisons using HOG3D Features

First, we evaluated our proposed method against several baselines. This evaluation uses HOG3D features, k-means quantized into a 1,000 word codebook for all methods. For this experiment, we use the following set of baselines: **Linear-SVM**, a linear SVM using HOG3D BoW features; **KSVM**, same video-level features with histogram intersection kernel (HIK) SVM; **Niebles** [8]; **Tang** [12]; **Linear-SAP**, the scene-aligned pooling method [1] using a linear SVM; and **K-SAP**, the same method using a HIK-SVM. Results for **Niebles** and **Tang** are reproduced from [12] and we obtained exactly the same quantized features to be directly comparable. Also, note that we re-implemented the scene aligned pooling method [1] using parameters suggested by the authors to permit direct comparisons.

Two variants of our proposed model were considered: **Linear-LSVM**, using a linear latent SVM, and **KLSVM**, using a HIK latent SVM. For the proposed models, selection of appropriate parameters is required, including the number of subcategories (C), number of scene types (S), and number of selected scenes (K). We used the kernelized version of our model with a HIK kernel to choose the best parameters on DEV-T (E1 to E5) and fixed them for all subsequent experiments using our model in this paper. Parameters were selected based on the criteria of mAP performance and model complexity. Interestingly, as Table 1 shows, as the various components of our model are added, mAP is improved. In particular, our latent model with selected parameters ($C = 8, S = 16, K = 4$) outperforms the standard kernelized SVM ($C = 1, S = 0$) by 6.6% in mAP.

In this section, our novel multiple kernel learning formulation is not employed, since the number of kernels used is very small. Section 4.2 considers experiments with the full model, using MKL for multi-feature fusion.

Results for the six baselines and two variants of the proposed method on DEV-O are shown in Table 2. When considering only models that employ linear SVMs (i.e., **Linear-SVM**, **Niebles**, **Tang**, **Linear-SAP**, and **Linear-LSVM**), the recently proposed scene aligned pooling method provides highest performance with a mean AP of 6.28%. The linear variant of the proposed model offers midrange performance. However, the simple **KSVM** baseline significantly outperforms all variants that use a linear SVM classifier, including **Niebles** and **Tang**, which model complex structure. It appears that use of a kernelized SVM is critical for the task of accurate event detection.

A second performance trend can be identified from considering the models that use kernelized SVMs (i.e., **KSVM**, **K-SAP**, and **KLSVM**). Specifically, the proposed model, **KLSVM**, outperforms all other baselines, including **K-SAP** by 3.72% and **KSVM** by 4.22%. Further, **KLSVM** attains best performance on eight out of ten event categories, often by a significant margin (e.g., 11.43% gap for E14). These results emphasize the importance of using a compositional framework. Note that a kernelized version of **Tang** was not considered because it is not clear how the computationally expensive inference could be done for an extension to kernel SVMs, especially for a large data collection.

4.2. Comparisons using Multiple Features

In this section, we demonstrate the effectiveness of the full, multiple kernel learning-based model by extending from a single feature modality to six features.

To demonstrate the full **MKL-KLSVM** model, HOG3D was supplemented with five additional features from the Sun09 set [19]. The additional features were: sparse SIFT, dense SIFT, HOG2x2, self-similarity descriptors (SSIM), and color histograms. Here, the same set of features was used for both the global and scene type parts of our model (i.e., $G = L = 6$). These particular features were selected because we empirically found them to offer best performance on TRECVID MED11. Features were extracted at four second time increments, synchronized with

Table 2: Performance comparison against several baselines using HOG3D features on DEV-O for E6-E15. Numbers denote the average precision, in %. Best results for a particular event category are shown in bold.

Event	Chance	**Linear-SVM**	**Niebles** [8]	**Tang** [12]	**Linear-SAP** [1]	**Linear-LSVM**	**KSVM**	**K-SAP** [1]	**KLSVM**
E6	0.54	1.97	2.25	4.38	2.77	2.34	**6.08**	4.73	5.73
E7	0.35	1.25	0.76	0.92	2.11	1.33	2.87	2.26	**4.81**
E8	0.42	6.48	8.30	15.29	25.48	10.30	20.75	22.99	**35.82**
E9	0.26	2.15	1.95	2.04	4.14	1.79	6.25	7.61	**8.38**
E10	0.25	0.81	0.74	0.74	1.03	0.76	1.43	1.34	**2.12**
E11	0.43	1.10	1.48	0.84	1.93	1.41	2.29	2.65	**4.65**
E12	0.58	5.83	2.65	4.03	7.06	5.71	8.44	8.70	**10.99**
E13	0.32	2.58	2.05	3.04	10.38	2.57	9.44	10.43	**13.11**
E14	0.27	1.18	4.39	10.88	6.69	4.58	10.00	11.89	**23.32**
E15	0.26	0.92	0.61	**5.48**	1.21	1.09	2.49	2.4	3.29
mAP	0.37	2.43	2.52	4.77	6.28	3.19	7.00	7.50	**11.22**

the HOG3D features. The two coarser scales of a three level spatial pyramid were retained for dense SIFT, HOG2x2, and SSIM. Sparse SIFT and color histograms were extracted on the whole frame. Global and segment-level features are formed by averaging the histograms.

Three baselines are compared against the full **MKL-KLSVM**, all systems using the identical set of six features. The first baseline, **KSVM**, is trained on a summation of six χ^2 kernels on the global features. The second baseline, **MKL-SVM**, is similar to **KSVM**, but the weights on the kernels are trained. **KLSVM** and **MKL-KLSVM** are variants of our model that consider both the global and segment-level features. Global models and scene type models are formed using χ^2 and HIK, respectively. In the **KLSVM**, the weights of all kernels are fixed to one, while in the **MKL-KLSVM**, the kernel weights are learned.

Table 3 presents the results of these systems for DEV-O. A progression in the mAP performance is demonstrated as the different components of our model are added. By allowing the model to learn the kernel weights for the various feature modalities, **MKL-SVM** shows slight performance gains over **KSVM**. **KLSVM** improves performance by incorporating our proposed compositional model that performs latent segment selection. Finally, when considering the full model, **MKL-KLSVM**, which allows the various kernel weights to be adapted for the global and segment components across multiple features, highest overall accuracy is attained.

Table 3: Performance comparison against several baselines using multiple features on DEV-O for E6-E15. Numbers denote the average precision, in %.

Event	**KSVM**	**MKL-SVM**	**KLSVM**	**MKL-KLSVM**
E6	6.36	**6.77**	5.36	6.24
E7	22.04	22.22	23.47	**24.62**
E8	31.23	31.40	31.99	**37.46**
E9	**18.13**	17.49	16.18	15.72
E10	2.48	**2.55**	2.36	2.09
E11	3.88	4.03	**7.98**	7.65
E12	10.90	11.00	10.77	**12.01**
E13	13.31	**14.54**	13.70	10.96
E14	12.97	12.34	31.22	**32.67**
E15	3.98	3.81	7.47	**7.49**
mAP	12.53	12.62	15.05	**15.69**

4.3. Results Visualizations

Figure 3 shows qualitative results for our model on four test videos, where eight second segments are visualized using their center frames. The frames that are latently selected tend to be discriminative and ignore temporal clutter inherent in many test videos. For example, in the *sewing project* video, the latter frames where the individual is walking in an outdoor environment are not selected because such scenes are not typically associated with a video of a sewing project.

Latently selected frames of the same scene type model also often have similar overall appearance characteristics. For instance, in the *grooming animal* test video, the frame in the green box shows a view of a dog's backside with human hands moving its tail. A support vector containing a frame for this scene type showing a comparable view of a dog with extended human arms is also selected.

The visualizations also demonstrate the compositional approach. For example, in the *changing a tire* test sequence, two of the top three support vector videos offer good matches for three of the latently selected frames in the test sequence (corresponding to the test frames highlighted with red, yellow, and blue boxes). However, for the fourth test frame that was selected (green box), only one of the top three support vectors provides a particularly discriminative match. The proposed model is able to accumulate evidence for classification from different video segments in the pool of training videos.

5. Conclusion

We presented a novel, compositional model for video event detection that leverages a novel multiple kernel learning algorithm that incorporates structured latent variables. The kernelized latent variable framework allows the model to select and match test video segments with those that are extracted from the pool of training of videos. The compositional nature of the model allows it to respond to the challenges of intra-class variation and temporal clutter, which

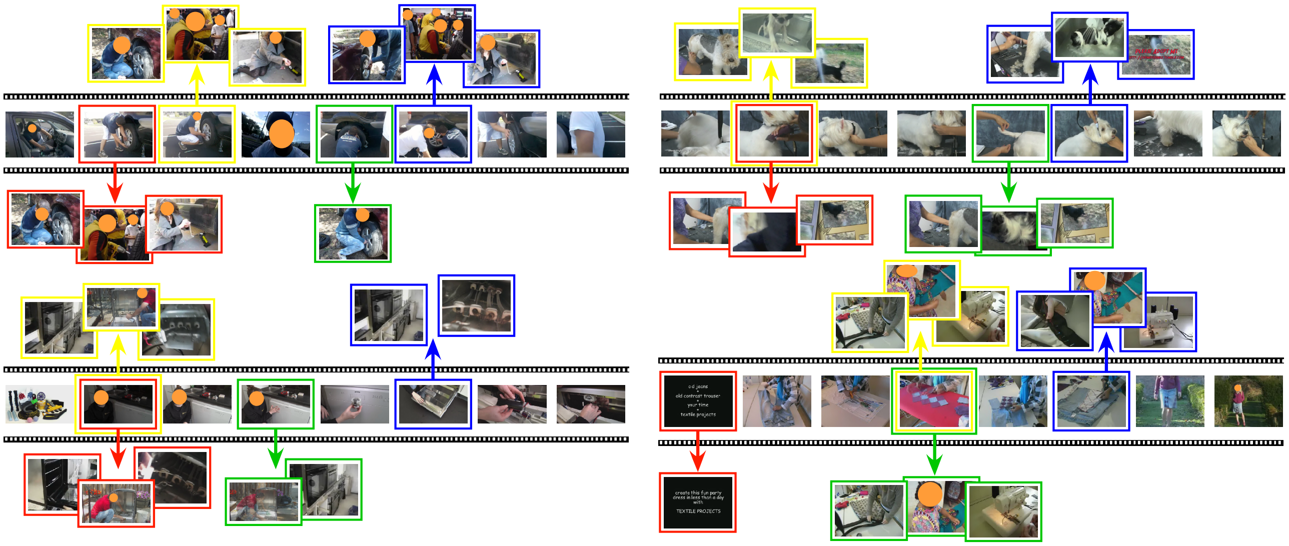

Figure 3: Qualitative visualization of results. Individual images denote the center frame from an eight second window. Each subfigure shows frames from a testing video along with frames from the three support vectors that produce the overall best match to that test video (i.e., frames from only three support vector videos are shown for each test sequence). For a test video, the $K = 4$ frames that were latently selected are highlighted with colored boxes, where color denotes the particular scene type model. Latently selected frames from the the top three support vectors are grouped using colored boxes, where color corresponds to the same scene types selected for the test video. From top-to-bottom, left-to-right, the testing videos correspond to *changing tire (E7)*, *grooming animal (E10)*, *repairing appliance (E14)*, and *sewing project (E15)*. Faces have been obscured for privacy considerations. Best viewed magnified and in color.

are inherent in unconstrained internet videos. Additionally, since multiple feature types are required to attain state-of-the-art performance on TRECVID MED11, a principled approach to feature fusion via multiple kernel learning with structured latent variables is proposed. Experimental results showed that this approach outperforms state-of-the-art baselines on the challenging TRECVID MED11 dataset.

References

[1] L. Cao, Y. Mu, P. Natsev, S.-F. Chang, G. Hua, and J. R. Smith. Scene aligned pooling for complex video recognition. In *ECCV*, 2012. 2, 6, 7

[2] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010. 2, 3, 4

[3] Z. Gu, T. Mei, X.-S. Hua, J. Tang, and X. Wu. Multi-layer multi-instance learning for video concept detection. *IEEE Transactions on Multimedia*, 10(8):1605–1616, 2008. 2

[4] H. Izadinia and M. Shah. Recognizing complex events using large margin joint low-level event model. In *ECCV*, 2012. 2

[5] M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskov, K.-R. Müller, and A. Zien. Efficient and accurate lp-norm multiple kernel learning. In *NIPS*, 2009. 4

[6] S. Maji and A. C. Berg. Max-margin additive classifiers for detection. In *ICCV*, pages 40–47, 2009. 2

[7] P. Natarajan, S. Wu, S. N. P. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, R. Prasad, and P. Natarajan. Multimodal feature fusion for robust event detection in web videos. In *CVPR*, 2012. 2

[8] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010. 2, 6, 7

[9] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, and G. Quenot. Trecvid 2011 — an overview of the goals, tasks, data, evaluation mechansims and metrics. In *Proceedings of TRECVID 2011*, 2011. 5

[10] F. Perronnin, J. Sánchez, and Y. Liu. Large-scale image categorization with explicit data embedding. In *CVPR*, 2010. 2

[11] J. Platt. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods*, pages 185–208. MIT Press, 1999. 4

[12] K. Tang, F.-F. Li, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012. 2, 5, 6, 7

[13] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004. 4

[14] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 5

[15] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *PAMI*, 34(3):480–492, 2012. 2

[16] S. V. N. Vishwanathan, Z. Sun, N. Ampornpunt, and M. Varma. Multiple kernel learning and the SMO algorithm. In *NIPS*, 2010. 3, 4

[17] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010. 2

[18] X. Wu and Y. Jia. View-invariant action recognition using latent kernelized structural SVM. In *ECCV*, 2012. 2, 3

[19] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 6

[20] W. Yang, Y. Wang, A. Vahdat, and G. Mori. Kernel latent svm for visual recognition. In *NIPS*, 2012. 2, 3

[21] C.-N. J. Yu and T. Joachims. Learning structural SVMs with latent variables. In *ICML*, 2009. 2, 3, 4