Complex Event Detection using Semantic Saliency and Nearly-Isotonic SVM

Xiaojun Chang CXJ273@GMAIL.COM
Yi Yang YEE.I.YANG@GMAIL.COM
Centre for Quantum Computation and Intelligent Systems, University of Technology Sydney, Sydney, Australia

Eric P. Xing EPXING@CS.CMU.EDU
Yao-Liang Yu YAOLIANG@CS.CMU.EDU
Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

We aim to detect complex events in long Internet videos that may last for hours. A major challenge in this setting is that only a few shots in a long video are relevant to the event of interest while others are irrelevant or even misleading. Instead of indifferently pooling the shots, we first define a novel notion of semantic saliency that assesses the relevance of each shot with the event of interest. We then prioritize the shots according to their saliency scores since shots that are semantically more salient are expected to contribute more to the final event detector. Next, we propose a new isotonic regularizer that is able to exploit the semantic ordering information. The resulting nearly-isotonic SVM classifier exhibits higher discriminative power. Computationally, we develop an efficient implementation using the proximal gradient algorithm, and we prove new, closed-form proximal steps. We conduct extensive experiments on three real-world video datasets and confirm the effectiveness of the proposed approach.

1. Introduction

Modern consumer electronics (*e.g.* smart phones) have made video acquisition convenient for the general public. Consequently, the number of videos on Internet has grown at an unprecedented rate, thanks also to the appearance of large video hosting websites (*e.g.* YouTube). How to store, index, classify, and eventually make sense of the vast information contained in these videos has become an important challenge for the machine learning and computer vision communities. Reflecting this challenge, the National Institute of Standards and Technology (NIST) hosts an annual competition on a variety of retrieval tasks, of which the multimedia event detection (MED) task has received considerable attention and is also the focus of this work.

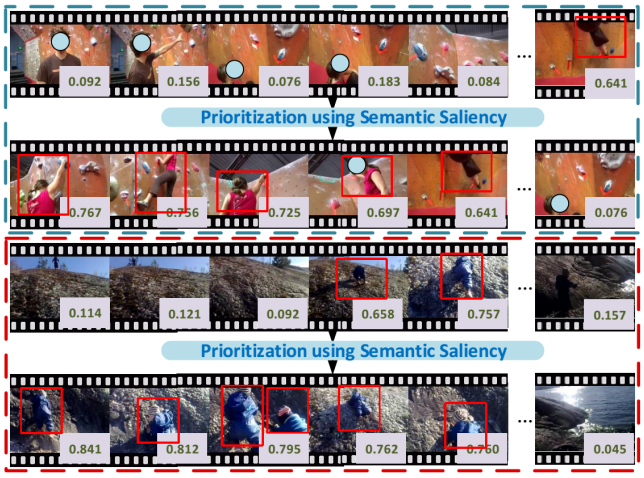

Figure 1. Two Internet video examples, where the same event *Rock Climbing* happened in very different time frames. The number in each frame indicates its saliency score, which describes how this keyframe is relevant to the specified event. We use this saliency information to prioritize the video shot representations.

In MED, a large number of *unseen* videos is presented and the learning algorithm must rank them according to their likelihood of containing an event of interest, such as *birthday party* or *dog show*. To start, a compact representation of the video is first sought using feature extraction. Deep learning approaches, *e.g.* convolutional neural networks (CNNs), have become increasingly popular in this regard. The standard way (Aly et al., 2013; Yu et al., 2014) is to extract local descriptors using CNNs on each frame of a video clip and then aggregate video-wise, through either average-pooling or max-pooling or even more complicated pooling strategies (*e.g.* Cao et al., 2012; Li et al., 2013). While effective in reducing size, pooling may result in the

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

loss of structural information, hence is less desirable. On the other hand, training a classifier on top of all frame features is also challenging, due to the limited number of positive examples. For instance, the MED datasets provided by the NIST contain only 100 positive examples (and $\approx$ 5,000 negatives).

Instead, we consider an intermediate strategy in this work. We first split each video into multiple shots, and for each shot we randomly sample one key frame whose extracted features will be used to represent the entire shot. Instead of conducting pooling on the shot-level, we prioritize the shots according to their "relevance" to the event of interest. To address the the small sample size issue of training data, we propose to train an "informed" classifier that puts larger weights on more relevant shots. Leveraging on this ordering bias, we are able to significantly enhance the discriminative power of the statistical classifier.

More precisely, in §3 we propose a new prioritizing procedure based on the notion of *semantic saliency*. Prioritizing objects according to saliency (Koch & Ullman, 1985) is ubiquitous in visual tasks such as segmentation (Rahtu et al., 2010) and video summarization (Lee et al., 2012). However, instead of borrowing an existing saliency algorithm, we prefer a more "supervised" version that is closely related to our event detection task. To this end, we first train 1,000 concept detectors using the ImageNet dataset (Russakovsky et al., 2014), resulting in a probability vector for each shot that indicates the relative presence of individual concepts. Then, using the skip-gram model (Mikolov et al., 2013) in natural language processing, we pre-learn a relevance vector that measures the *a priori* relevance of each concept name with the textual description (provided in most MED datasets) of the event of interest. Lastly, by taking a weighted combination of the probability vector (likelihood) and the relevance vector (prior), we obtain the proposed semantic saliency of each shot. Rearranging the shots according to their saliency scores yields the desired prioritization.

After prioritizing the multiple shots of each video, we feed them into a linear large margin classifier such as support vector machines (SVM). Intuitively, shots with higher semantic saliency scores are expected to be more relevant to the event, hence providing more discriminative information. To incorporate this carefully constructed side information, we propose, in §4, a new isotonic regularizer that encourages the classifier to put more weights on more salient shots. Our isotonic regularizer is not convex, but the popular proximal gradient algorithm can still be applied, with the convergence guarantee recently established in (Bolte et al., 2014). The key component, namely the proximal map of the isotonic regularizer, despite being nonconvex, is solved globally and exactly in linear time through a sequence of reductions. The final algorithm, which we call nearly-isotonic SVM (NI-SVM), is very efficient and runs quickly on large MED datasets. An alternative convex variant is also proposed, although its performance is found to be inferior.

In §5 we conduct extensive experiments on three real-world unconstrained video datasets (CCV, MED13, MED14), and achieve state-of-the-art performances measured by the mean average precision. Finally, §6 concludes the paper with some future directions.

2. Complex Event Detection

Event detection refers to the task in which the learning algorithm must rank a large number of *unseen* videos according to their likelihood of containing an event of interest. Events are complex, and may be composed of several scenes, objects, actions, and the rich interactions between them. On the application side, event detection is the first important step in video analysis towards automatic categorization, recognition, search, and retrieval (just to name a few) hence has attracted much attention in the machine learning and computer vision communities.

Complex event detection on unconstrained Internet videos is very challenging for the following reasons: 1) Unlike professional video recordings (*e.g.* films), the quality of Internet videos varies considerably, making them difficult to model statistically; 2) Events are complex and can be ambiguous: the *wedding shower* event consists of multiple defining concepts such as hugging (action), laughing (action) and veil (object), and can take place indoors (*e.g.* in a house) or outdoors (*e.g.* in a park), resulting in dramatic intra-class variations; 3) Positive training examples are very limited. In the 10EX competition organized by NIST (NIST, 2013; 2014), only 10 positive training examples and 5000 negative examples are provided, creating a highly imbalanced ranking problem; 4) A video clip can last from a few minutes to several hours, with the evidence possibly scattering anywhere, see Figure 1 for an example.

A decent video event detection system usually consists of a good feature extraction module and a sophisticated statistical classification module. Various low-level features, *e.g.* SIFT (Lowe, 2004), Space-Time Interest Points (Laptev et al., 2007) and improved dense trajectories (Wang & Schmid, 2013) have been used. Improvements are obtained by aggregating complementary features at the video level, such as fusion (Natarajan et al., 2012), Fisher vector encoding (Oneaţă et al., 2013), and pooling (Cao et al., 2012; Tamrakar et al., 2012; Li et al., 2013; Tang et al., 2013). Combining multiple classifiers has also been observed to improve performance (Liu et al., 2012; 2013; Vahdat et al., 2013). Recently, Simonyan & Zisserman (2013); Karpathy et al. (2014) applied CNN for video classification but they

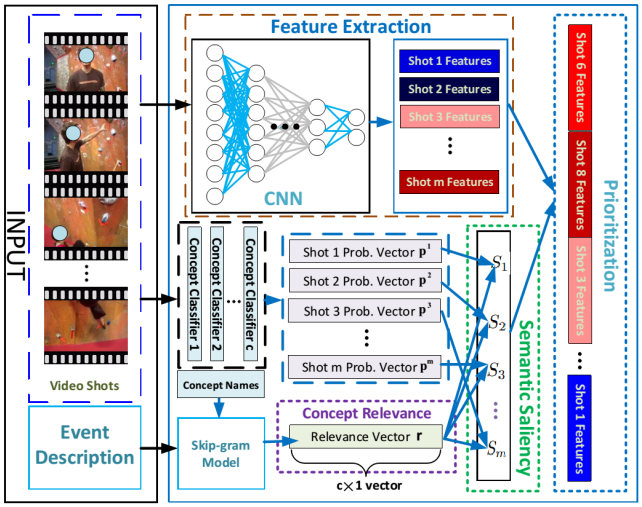

Figure 2. Each input video is divided into multiple shots, and each event has a short textual description. CNN is used to extract features (§3.1). ImageNet concept names and skip-gram model are used to derive a probability vector (§3.2) and a relevance vector (§3.3), which are combined to yield the new semantic saliency and used for prioritizing shots in the classifier training (§3.4).

did not consider semantic saliency nor isotonic regularization. Some recent works, *e.g.* (Tang et al., 2012; Lai et al., 2014), also tried to exploit temporal information. Lastly, Ramanathan et al. (2013) used video textual descriptions to refine actions and role models while Zhao et al. (2011) considered online event detection for surveillance videos.

3. Prioritization using Semantic Saliency

As we mentioned in §2, a good feature extraction module is vital for event detection. Thus we first describe our feature extraction method. Since not all video shots are equally relevant to the event of interest, we develop in this section a new prioritization procedure to reorder them. Then in §4 we propose the nearly-isotonic SVM classifier to exploit the ordering information. The overall system is illustrated in Figure 2 and we discuss it block by block in the sequel.

3.1. Feature extraction

To extract representative features from videos, we first segment each of them into m shots $[\mathbf{v}_1, \ldots, \mathbf{v}_m]$ using the color histogram difference as the indication of the shot boundary. Other segmentation or change-point detection algorithms may also be used. For simplicity, we randomly sample one key frame from each shot and extract the frame level CNN descriptors using the architecture of (Simonyan & Zisserman, 2015). The key insight in (Simonyan & Zisserman, 2015) is that by using smaller convolution filters (3×3) and very deep architecture (16-19 layers), significant improvement on the ImageNet Challenge 2014 can be achieved. Due to its excellent performance on images, we therefore choose to apply the same architecture to our video datasets by sampling key frames. With some abuse of notation, the extracted CNN features (from fc6, the first fully-connected layer) of all m shots are still written collectively as $[\mathbf{v}_1, \ldots, \mathbf{v}_m] \in \mathbb{R}^{d \times m}$. In our experiments, we set m as the smallest number of keyframes for all videos. For example, in the Trecvid MED14 dataset, the shortest video has $m = 535$ keyframes. We do not explicitly model temporal information in this work, although conceivably it could further aid our detection system.

3.2. Concept detectors

The ImageNet dataset (Russakovsky et al., 2014) consists of $c = 1,000$ classes/concepts, each accompanied with an entity description (*e.g.*, *lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens*). These concepts can be used to aid event detection. For example, we would expect concepts such as "chrysanthemum dog" or "shetland sheepdog" to be relevant to the event *dog show*. Thus we train a detector/classifier for each concept. All c concept detectors will be applied to each video shot $\mathbf{v}_j$, resulting in a c-dimensional probability vector $\mathbf{p} \in \mathbb{R}_+^c$, with the entry p_k standing for the (relative) probability of having the k-th concept appear in the shot $\mathbf{v}_j$. Conveniently, since the CNN architecture of (Simonyan & Zisserman, 2015) is also trained on ImageNet, we can simply extract the vector $\mathbf{p}$ from its probability layer (the last layer). This probability vector $\mathbf{p}$ will be combined with the concept relevance (defined next) to yield the semantic saliency scores.

3.3. Concept relevance

Events come with short descriptions. For example, the event *dog show* in the Trecvid MED14 (NIST, 2014) is defined as "a competitive exhibition of dogs". We exploit this textual information by learning a semantic relevance score between the event description and the individual ImageNet concept names. More precisely, we pre-learn a skip-gram model (Mikolov et al., 2013) using the English Wikipedia dump (http://dumps.wikimedia.org/enwiki/). The skip-gram model learns a D-dimensional vector space representation of words by fitting the joint probability of the co-occurrence of surrounding contexts on large unstructured text data, and places semantically similar words near each other in the embedding vector space. Thus it is able to capture a large number of precise syntactic and semantic word relationships. For short phases consisting of multiple words (*e.g.* event descriptions), we simply average its word-vector representations.

After properly normalizing the respective word-vectors, we compute the cosine distance of the event description and all the concept names in ImageNet, resulting in a relevance vector $\mathbf{r} \in \mathbb{R}^c$, where r_k measures the *a priori* relevance of the k-th concept to the event of interest.

3.4. Semantic saliency

Lastly, we define the semantic saliency score of each video shot as a weighted combination of the concept probability vector $\mathbf{p}$ (likelihood, different for each video shot, §3.2) and the *a priori* concept relevance vector $\mathbf{r}$ (prior, same for all shots, §3.3):

$$s = \sum_{k=1}^{c} p_k r_k. \tag{1}$$

Repeating this for each shot $\mathbf{v}_j, j = 1, \ldots, m$, of a video generates its saliency vector $\mathbf{s} = [s_1, \ldots, s_m]$. Intuitively, the saliency score s_j evaluates the relevance of the j-th shot to the event of interest. The most salient shots are those most likely to contain the specified event, hence they should carry more weight in the final classifier boundary. Thus we prioritize the shots by reordering them such that

$$s_1 \geq s_2 \geq \cdots \geq s_m, \tag{2}$$

i.e., the shots are ranked in a descending order. Importantly, note that different videos are reordered differently. After prioritization, it is desirable to train an event detector that exploits this valuable ordering information, which motivates the isotonic regularizer that we propose in the next section. Note that all of our results can be extended to a partial ordering, *i.e.*, allowing some shots to be incomparable (when their scores are very close, for instance).

The definition of our semantic saliency essentially follows the zero-shot learning framework of (Lampert et al., 2009). It is convenient because it is fully automatic. A potentially superior approach[1] is to extract relevant keyframes from the positive training exemplars and use these to define saliency. However, the downside of this alternative is that it requires some human intervention/labeling.

4. Nearly-Isotonic Support Vector Machines

As described above, we represent each video $V^i, i = 1, \ldots, n$, as a matrix $[\mathbf{v}_1^i, \ldots, \mathbf{v}_m^i]$, where $\mathbf{v}_j^i \in \mathbb{R}^d$ are the extracted CNN features from the j-th shot. These features are reordered according to their semantic saliency scores defined in §3.4. To perform event detection, we employ the large margin support vector machines (SVM):

$$\min_{W \in \mathbb{R}^{d \times m}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \langle V^i, W \rangle) + \lambda \cdot \Omega(W), \tag{3}$$

where $\lambda > 0$ is the regularization constant, and $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ measures the discrepancy between the true binary[2] label y_i and the prediction $\langle V^i, W \rangle := \sum_{j,k} V_{jk}^i W_{jk}$. For instance, the well-known hinge loss is $\ell(y, t) = (1 - yt)_+$, where as usual $(t)_+ := \max\{t, 0\}$. The regularizer Ω enforces some desirable structure on the classifier weight matrix W, and will play a major role here.

In vanilla SVM, $\Omega(W) = \|W\|_{\mathsf{F}}^2$ (the squared Frobenius norm), which penalizes large weight matrices to avoid overfitting. Another useful alternative is $\Omega(\mathbf{w}) = \|W\|_1$ (the ℓ_1-norm, sum of absolute values), which encourages sparsity. Hence is effective for feature selection. However, neither is able to exploit the order information that we carefully constructed in §3. In fact, both norms are invariant to column reorderings. Instead, we propose below a new isotonic regularizer that respects the prioritization we performed on the shots using their saliency scores.

4.1. The isotonic regularizer

Let us assume momentarily that $d = 1$, *i.e.*, there is only a single feature. This assumption, although unrealistic, simplifies our presentation and will be removed later. As mentioned, we want to learn a weight vector that respects the saliency order in our shot features, since more relevant shots are expected to contribute more to the final detection boundary. This motivates us to consider the following isotonic regularizer:

$$\|\mathbf{w}\|_\imath := \sum_{j=2}^{m} (|w_j| - |w_{j-1}|)_+. \tag{4}$$

To see the rationale behind, let us use the absolute value $|w_j|$ of the weight vector to indicate the contribution of the j-th shot to the final decision rule $\operatorname{sign}(\sum_j v_j w_j)$. Since the shots are arranged in decreasing order of relevance, we would expect roughly $|w_1| \geq |w_2| \geq \cdots \geq |w_m|$, *i.e.*, the weights (in magnitude) align well with the saliency order we constructed in §3.4. If this is the case, the regularizer $\|\mathbf{w}\|_\imath$ would be 0, *i.e.* incurring no penalty. On the other hand, we pay a linear cost for violating any of the saliency orders, *i.e.*, if instead $|w_j| > |w_{j-1}|$ for some j, we suffer a cost equal to the difference $|w_j| - |w_{j-1}|$. Clearly, the more we deviate from a saliency order, the more we are penalized. Equipping $\Omega(\mathbf{w}) = \|\mathbf{w}\|_\imath$ in (3) we obtain a new classification method which we call the nearly-isotonic SVM (NI-SVM):

$$\min_{W} \quad \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \langle V^i, W \rangle) + \lambda \cdot \|W\|_\imath. \tag{5}$$

Exploiting order information in statistical estimation has a long history, see the wonderful book (Barlow et al., 1972) for early applications. Similar regularizers to (4) have also appeared recently. For instance, Tibshirani et al. (2011) dropped the absolute values in (4), while Yang et al. (2012) replaced the positive part in (4) with the absolute value. Since the weight vector $\mathbf{w}$ has signed entries, and the order we aim to force is one-directional, we believe our formu-

[1] We thank an anonymous reviewer for suggesting this.

[2] While it may seem beneficial to jointly detect multiple events in a multi-task learning framework, we follow the NIST standard that requires the separate detection of each event.

lation (4) here is more appropriate (see §5.3 below for empirical verification). Indeed, the variation in (Yang et al., 2012) will always incur a cost except when $|w_i| = |w_{i-1}|$, a condition that is too stringent to be useful in MED. Similarly, for two negative entries $0 > w_i > w_{i-1}$, the variation in (Tibshirani et al., 2011) would incur an unnecessary penalty $w_i - w_{i-1} > 0$. While all three variations are intimately related, we note that both (4) and the variation in Yang et al. (2012) are not convex (nor smooth). Nevertheless, we can still design an efficient algorithm for solving NI-SVM. Before that, however, let us mention how to extend to multiple features ($d > 1$).

4.2. Extending to multiple features

When $d > 1$, each video representation V^i is a matrix in $\mathbb{R}^{d\times m}$, hence accordingly the linear classifier we learn is indexed by the weight matrix $W \in \mathbb{R}^{d\times m}$. Inspecting the NI-SVM formulation (5), we note first that the loss term extends immediately: the standard inner product $\langle V^i, W\rangle$ in $\mathbb{R}^{d\times m}$ extends straightforwardly for any d. For the isotonic regularizer, we need to summarize all d importance measures (each contributed by a feature). There are multiple ways to achieve this, and we consider two particularly convenient ones here:

$$\|W\|_{\imath,1} := \sum_{i=1}^{d} \|W_{i,:}\|_\imath = \sum_{i=1}^{d}\sum_{j=2}^{m} (|W_{i,j}| - |W_{i,j-1}|)_+, \quad (6)$$

$$\|W\|_{\imath,2} := \sum_{j=2}^{m} (\|W_{:,j}\|_2 - \|W_{:,j-1}\|_2)_+, \quad (7)$$

where $W_{i,:}$ (resp. $W_{:,j}$) is the i-th row (resp. j-th column) of the matrix W. The first regularizer (6) simply sums the vector isotonic regularizer along each feature dimension, while the second regularizer (7) first aggregates the shot importance by summing the d weights and then applies the vector isotonic regularizer on top. When $d = 1$, both (6) and (7) reduce to the vector isotonic regularizer (4), but we expect them to behave differently when $d > 1$. The corresponding SVM formulation with the matrix regularizers (6) and (7) will be called respectively NI-SVM$_1$ and NI-SVM$_2$.

4.3. The proximal gradient

The isotonic regularizers (6) and (7) are both nonsmooth and nonconvex, hence very challenging numerically. Fortunately, the proximal gradient algorithm, a.k.a. the forward-backward splitting procedure, has been recently extended in (Bolte et al., 2014) to handle semialgebraic functions that need not be convex or smooth. Recall that a function $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ is semialgebraic if its graph $\{(\mathbf{w}, f(\mathbf{w})) : \mathbf{w} \in \operatorname{dom} f\}$ is a semialgebraic set, *i.e.*, a finite union of finite intersections of the sets $\{\mathbf{z} \in \mathbb{R}^{d+1} : p_1(\mathbf{z}) < 0, p_2(\mathbf{z}) = 0\}$, where p_1, p_2 are polynomials. Of course, polynomials are semialgebraic. Many practical functions are semialgebraic including all the isotonic regularizers we use here (the proof is a routine exercise in real algebraic geometry hence is omitted). Restricting to semialgebraic functions f and g, we can now consider the general composite minimization problem:

$$\min_{\mathbf{w}} \; f(\mathbf{w}) + g(\mathbf{w}). \quad (8)$$

Since we do not assume convexity, we will be satisfied with convergence to a critical point[3].

Theorem 1 (Bolte et al. 2014, Proposition 3)**.** *Let f and g be semialgebraic functions, and f has L-Lipschitz gradient, then for any step size $\mu \in (0, 1/L)$, the following iteration converges to a critical point of the minimization problem (8), provided that the iterates are bounded:*

$$\mathbf{w} \leftarrow \mathbf{w} - \mu\nabla f(\mathbf{w}), \quad (9)$$

$$\mathbf{w} \leftarrow \mathsf{P}_g^\mu(\mathbf{w}) := \arg\min_{\mathbf{z}} \tfrac{1}{2\mu}\|\mathbf{w} - \mathbf{z}\|_2^2 + g(\mathbf{z}). \quad (10)$$

The above iterations (9)-(10), which we call proximal gradient (PG), are very simple: it amounts to a usual gradient step w.r.t. f first, and then a proximal step w.r.t. g using the *proximal map* P_g^μ. For instance, when g is the 1-norm, then $\mathsf{P}_{\|\cdot\|_1}^\mu(\mathbf{w}) = \operatorname{sign}(\mathbf{w}) \cdot (|\mathbf{w}| - \mu)_+$ is the well-known soft-shrinkage operator. The boundedness assumption in Theorem 1 is not restrictive: it is satisfied as long as the sublevel sets $\{\mathbf{w} : f(\mathbf{w}) + g(\mathbf{w}) \le \alpha\}$ are bounded.

Since evaluating the gradient ∇f is straightforward, the efficiency of PG (9)-(10) hinges on our capability of solving the subproblem (10) quickly. When specialized to our NI-SVM, we need to compute the proximal map for the isotonic regularizers in (6) and (7). A slight complication here is that the isotonic regularizers are not convex. Nevertheless, as we demonstrate in the next subsection, we can still compute the proximal maps exactly in linear time.

4.4. Proximal map for the isotonic regularizer

We address the proximal map for both matrix isotonic regularizers (6) and (7) through a sequence of reductions that allows us to directly exploit existing results.

Reducing to vector case We first reduce the matrix regularizers (6) and (7) to their vector cousin:

$$\mathsf{P}_{\|\cdot\|_\imath}^\mu(\mathbf{w}) := \arg\min_{\mathbf{z}} \tfrac{1}{2\mu}\|\mathbf{w} - \mathbf{z}\|_2^2 + \|\mathbf{z}\|_\imath, \quad (11)$$

where $\mathbf{w}, \mathbf{z} \in \mathbb{R}^m$ and $\|\cdot\|_\imath$ is defined in (4). For (6), this reduction is obvious as (6) is separable in rows of the matrix W, so we need only apply (11) to each row independently. For (7), let us write out the objective of its proximal map explicitly (Z is the optimization variable):

$$\frac{1}{2\mu}\sum_{j=1}^{m}\|W_{:,j} - Z_{:,j}\|_2^2 + \sum_{j=2}^{m}(\|Z_{:,j}\|_2 - \|Z_{:,j-1}\|_2)_+. \quad (12)$$

[3]For nonsmooth functions, we define the critical points to be the set $\{\mathbf{w} : 0 \in \partial(f + g)(\mathbf{w})\}$, where ∂f is the subdifferential of f, a strict generalization of the usual gradient, see (Rockafellar & Wets, 1998) for details.

Consider the polar decomposition $Z = \Theta\Lambda$, where each column of Θ has unit Euclidean norm and Λ is diagonal with z_i in the i-th diagonal. Clearly, the regularizer $\|Z\|_{\imath,2}$ only depends on Λ and for fixed Λ, the quadratic term in (12) is minimized precisely when $\Theta_{:,j} = \frac{W_{:,j}}{\|W_{:,j}\|_2}$ for all j. Plugging it back we can solve the diagonal Λ by:

$$\arg\min_{\mathbf{z}\in\mathbb{R}^m} \frac{1}{2\mu}\sum_{j=1}^{m}(z_j - \|W_{:,j}\|_2)^2 + \|\mathbf{z}\|_\imath. \tag{13}$$

Clearly, this is in the form of the vector problem (11), which we will focus on in the sequel. Note that the isotonic regularizer $\|\mathbf{w}\|_\imath$ is not convex, thus its proximal map in (11) is not a convex problem. Nevertheless, we will show how to solve it exactly and globally in linear time.

Reducing to convex case Crucially, we observe that the vector isotonic regularizer $\|\mathbf{z}\|_\imath$ is invariant to the sign changes of any component w_i, but the quadratic term $\frac{1}{2\mu}\|\mathbf{w}-\mathbf{z}\|_2^2$ is minimized when the signs of $\mathbf{w}$ and $\mathbf{z}$ match. Thus, at any minimizer of (11) we must have $\mathrm{sign}(w_i) = \mathrm{sign}(z_i)$ for all i, further reducing the vector problem (11) to:

$$\mathsf{P}^{\mu}_{\kappa+\|\cdot\|_\imath}(|\mathbf{w}|) := \arg\min_{\mathbf{z}\geq 0} \tfrac{1}{2\mu}\|\mathbf{z}-|\mathbf{w}|\|_2^2 + \|\mathbf{z}\|_\imath, \tag{14}$$

where $|\mathbf{w}|$ is the component-wise absolute value of $\mathbf{w}$, and

$$\kappa(\mathbf{z}) = \begin{cases} 0, & \text{if } \mathbf{z}\geq 0 \\ \infty, & \text{otherwise} \end{cases}. \tag{15}$$

If we can solve (14), now a convex problem thanks to the nonnegative constraint, then we can immediately recover

$$\mathsf{P}^{\mu}_{\|\cdot\|_\imath}(\mathbf{w}) = \mathsf{P}^{\mu}_{\kappa+\|\cdot\|_\imath}(|\mathbf{w}|)\cdot\mathrm{sign}(\mathbf{w}). \tag{16}$$

Reducing to total variation norm Two elementary observations turn out to be key in solving (14) efficiently: a). Under the nonnegative constraint $\mathbf{w}\geq 0$, we have

$$2\|\mathbf{z}\|_\imath = \|\mathbf{z}\|_{\mathsf{tv}} + z_m - z_1,\ \ \|\mathbf{z}\|_{\mathsf{tv}} := \sum_{j=2}^{m}|z_j - z_{j-1}|,$$

which follows from applying the identity $2(t)_+ = t + |t|$ to each term $(|z_j| - |z_{j-1}|)_+$. b). The function κ in (15), *i.e.*, the nonnegative constraint, is invariant to permutations.

Reducing to known results Denote $h(\mathbf{w}) = w_m - w_1$, and recall that we need to solve the proximal map (14) of the function $\kappa(\mathbf{z}) + \|\mathbf{z}\|_\imath = \kappa(\mathbf{z}) + \frac{1}{2}(\|\mathbf{z}\|_{\mathsf{tv}} + h(\mathbf{z}))$. Then, by applying the results in (Yu, 2013) we arrive at the following decomposition rule (proof in Appendix A):

Theorem 2. *Denote $\mathbf{e}_i$ the i-th canonical basis in $\mathbb{R}^m$, then for any $\mu\geq 0$ and for all $\mathbf{w}\in\mathbb{R}^m_+$,*

$$\mathsf{P}^{\mu}_{\kappa+\|\cdot\|_\imath}(\mathbf{w}) = \mathsf{P}^{\mu}_{\kappa}\Big(\mathsf{P}^{\mu/2}_{\|\cdot\|_{\mathsf{tv}}}\big(\mathsf{P}^{\mu/2}_{h}(\mathbf{w})\big)\Big),\ \textit{where} \tag{17}$$

$$\mathsf{P}^{\mu}_{\kappa}(\mathbf{w}) = (\mathbf{w})_+, \tag{18}$$

$$\mathsf{P}^{\mu/2}_{h}(\mathbf{w}) = \mathbf{w} + \tfrac{\mu}{2}(\mathbf{e}_1 - \mathbf{e}_m). \tag{19}$$

Algorithm 1: Proximal Gradient for NI-SVM

1 Input: $W\in\mathbb{R}^{d\times m}$, regularization λ,γ, step size μ.
2 **repeat**
3 $\quad W \leftarrow W - \frac{\mu}{n}\sum_i \ell'(y_i,\langle V^i, W\rangle)V^i$; `// grad`
4 $\quad W \leftarrow \begin{cases} \texttt{prox\_row}\,(W,\mu,\lambda,\gamma)\,; & \texttt{// for (6)} \\ \texttt{prox\_col}\,(W,\mu,\lambda,\gamma)\,; & \texttt{// for (7)} \end{cases}$
5 **until** *convergence*;
6 **Procedure** `prox_row` (W,μ,λ,γ)
7 $\quad$ **for** *j= 1, . . . , d* **do**
8 $\quad\quad W_{j,:} \leftarrow \texttt{prox\_vec}\,(W_{j,:},\mu,\lambda,\gamma)$
9 **Procedure** `prox_col` (W,μ,λ,γ)
10 $\quad \mathbf{w} \leftarrow (\|W_{:,1}\|_2,\ldots,\|W_{:,m}\|_2)$
11 $\quad \mathbf{w} \leftarrow \texttt{prox\_vec}\,(\mathbf{w},\mu,\lambda,\gamma)$
12 $\quad W \leftarrow W\cdot\mathrm{diag}\left(\frac{w_1}{\|W_{:,1}\|_2},\ldots,\frac{w_m}{\|W_{:,m}\|_2}\right)$
13 **Procedure** `prox_vec` $(\mathbf{w},\mu,\lambda,\gamma)$
14 $\quad \mathbf{s}\leftarrow\mathrm{sign}(\mathbf{w}), \mathbf{w}\leftarrow|\mathbf{w}|$; `// omitted for (20)`
15 $\quad \mathbf{w} \leftarrow \mathbf{w} + \frac{\lambda\mu}{2}(\mathbf{e}_1 - \mathbf{e}_m)$
16 $\quad \mathbf{w} \leftarrow \mathsf{P}^{\mu\lambda/2}_{\|\cdot\|_{\mathsf{tv}}}(\mathbf{w})$
17 $\quad \mathbf{w} \leftarrow (\mathbf{w})_+$
18 $\quad \mathbf{w} \leftarrow \begin{cases} (\mathbf{w}-\gamma\mu)_+ & \texttt{// for (21)} \\ \frac{1}{1+2\gamma\mu}\mathbf{w} & \texttt{// for (22)} \end{cases}$
19 $\quad \mathbf{w} \leftarrow \mathbf{s}\cdot\mathbf{w}$; `// omitted for (20)`

The only term left unspecified in Theorem 2, $\mathsf{P}^{\mu/2}_{\|\cdot\|_{\mathsf{tv}}}$, has a well-known linear time algorithm, see *e.g.* (Davies & Kovac, 2001). We summarize the above reductions and steps in Algorithm 1, which computes the proximal maps of the matrix isotonic regularizers (6) and (7), globally and exactly in linear time.

4.5. The convex alternative

We also propose a convex alternative, mainly as a comparison baseline against our nonconvex NI-SVM formulation in (5). We simply add a nonnegative constraint on the classifier weight matrix W:

$$\min_{W\geq 0}\ \ \tfrac{1}{n}\textstyle\sum_{i=1}^{n}\ell(y_i,\langle V^i, W\rangle) + \lambda\cdot\|W\|_\imath, \tag{20}$$

where $\|W\|_\imath$ can be either $\|W\|_{\imath,1}$ (NI-SVM$_{1+}$) or $\|W\|_{\imath,2}$ (NI-SVM$_{2+}$). Note that the convexity in (20) is gained by placing a restriction on the classifier which may jeopardize its prediction performance (verified in our experiments). On the other hand, the nonnegative constraint encourages a sparse weight matrix W, in a spirit similar to nonnegative matrix factorization (Lee & Seung, 1999), since our video representation V^i is nonnegative as well. This may be beneficial in interpretation tasks (not considered in this work). Conveniently, the proximal gradient algorithm we developed above for NI-SVM can be easily recycled, with only a single slight change: We do not backup or restore the sign (*e.g.* omitting line 14 and 19 in Algorithm 1).

4.6. Adding more regularizers

In some applications it may be desirable to add other regularizers. For instance, the squared 2-norm can be used to avoid overfitting and the 1-norm may be needed for feature selection. Pleasantly, we can easily incorporate these additional regularizers, without complicating the algorithm at all, thanks to the following result (proof in Appendix A):

Theorem 3. *With the same setup as in Theorem 2, we have for any $\mu, \gamma \geq 0$ and for all $\mathbf{w} \in \mathbb{R}^m_+$:*

$$\mathsf{P}^{\mu}_{\kappa+\|\cdot\|_1+\gamma\|\cdot\|_2^2}(\mathbf{w}) = \mathsf{P}^{\mu}_{\gamma\|\cdot\|_2^2}\left[\mathsf{P}^{\mu}_{\kappa}\left(\mathsf{P}^{\mu/2}_{\|\cdot\|_{\mathrm{tv}}}\left(\mathsf{P}^{\mu/2}_{h}(\mathbf{w})\right)\right)\right] \quad (21)$$

$$\mathsf{P}^{\mu}_{\kappa+\|\cdot\|_1+\gamma\|\cdot\|_1}(\mathbf{w}) = \mathsf{P}^{\mu}_{\gamma\|\cdot\|_1}\left[\mathsf{P}^{\mu}_{\kappa}\left(\mathsf{P}^{\mu/2}_{\|\cdot\|_{\mathrm{tv}}}\left(\mathsf{P}^{\mu/2}_{h}(\mathbf{w})\right)\right)\right]. \quad (22)$$

5. Experiments

In this section we carry out extensive experiments to validate the proposed approach.

Datasets We test on three real event detection datasets:

- MED14: The TRECVID MEDTest 2014 dataset (NIST, 2014) contains approximately 100 positive training exemplars per event, and all events share ($\sim$ 5000) negative training exemplars. The test set has approximately 23,000 videos. There are in total 20 events, whose descriptions can be found in (NIST, 2014). To our best knowledge, this is the largest (35,914 videos in total) public dataset for event detection.
- MED13 (NIST, 2013): Similar as MED14. Note that 10 of its 20 events overlap with those of MED14.
- $\mathrm{CCV}_{\mathrm{sub}}$: The official Columbia Consumer Video dataset (Jiang et al., 2011) contains 9,317 videos in total with 20 semantic categories, including scenes like "beach", objects like "cat", and events like "baseball" and "parade". For our purpose we only use the 15 event categories. For each event we use its own training data as positive and all other training data as negative, totaling 4,659 training videos and 4,658 testing videos.

Experiment Setup As mentioned in §3.1 we use the CNN architecture in (Simonyan & Zisserman, 2015) to extract 4096 features on one keyframe per video shot. The regularization constants λ and γ are selected using cross-validation from the range $\{10^{-4}, 10^{-3}, \ldots, 10^3, 10^4\}$. According to NIST standard, we detect each event separately and use mean Average Precision (mAP) for evaluation (the larger the better).

Competitors We consider both the least squares loss $\ell(y,t) = \frac{1}{2}(t-y)^2$ and the squared[4] hinge loss $\ell(y,t) = (1-yt)_+^2$. We will use the subscript $_1$ and $_2$ to differentiate the matrix isotonic regularizers (6) and (7). A further subscript $_+$ is used to denote the convex alternative in §4.5. More precisely, we compare the following:

- LS_{A}: least squares loss with average-pooling on the video shots. Note that pooling is performed on the selected m keyframes, for fairness and efficiency.
- LS_{M}: least squares loss with max-pooling.
- LS_{T}: least squares loss without pooling, but the shots are reordered according to their saliency scores.
- NI-LS_1: least squares loss with isotonic regularizer (6).
- NI-LS_2: least squares loss with isotonic regularizer (7).
- NI-LS_{1+}: nonnegative convex version of NI-LS_1.
- NI-LS_{2+}: nonnegative convex version of NI-LS_2.

Similarly, for the squared hinge loss, we replace "LS" throughout with "SVM". As suggested in §4.6, additional ℓ_2^2 and ℓ_1 regularizers can be incorporated. We also compare against some state-of-the-art alternatives in §5.2.

5.1. Against standard baselines

We report the experimental results in Table 1, where full details on the MED14 dataset are documented (for the least squares loss). The average performances on MED13 and $\mathrm{CCV}_{\mathrm{sub}}$ are also recored at the bottom of Table 1, with full details deferred to Appendix B. The average performances for the squared hinge loss are given in Table 2, again with full details deferred to Appendix B.

We make a few observations from Table 1 and Table 2:

1) Average-pooling outperforms max-pooling on average and in most events.
2) LS_{T} and $\mathrm{SVM}_{\mathrm{T}}$ perform significantly better than their pooling counterparts. This confirms that pooling, if naively done, can be detrimental. However, LS_{T} and $\mathrm{SVM}_{\mathrm{T}}$ do not directly benefit from prioritizing the shots: their classifier weights ignore the ordering information.
3) NI-LS, with either matrix isotonic regularizers, further outperforms LS_{T}, demonstrating that properly exploiting the ordering information can significantly improve the performance. Moreover, the matrix isotonic regularizer (7) (subscript $_2$) generally performs better than the matrix isotonic regularizer (6) (subscript $_1$).
4) The squared hinge loss on average performs better than the least squares loss, unanimously across all methods.
5) Additional ℓ_2^2-norm regularization (left panel) generally outperforms additional ℓ_1-norm regularization (right panel). We hypothesize that it is because the CNN features we use are very discriminative hence sparsity does not help here.
6) The convex variants (with subscript $_+$) have poorer performance than the nonconvex counterparts (but still competitive against average-pooling), possibly because the nonnegative constraint is too restrictive. Empirically (results not shown), we also found that the nonconvex variants are quite robust against initializations (random or using the convex variant), likely because we are able to solve the proximal maps globally and in closed-form.

[4] The convergence guarantee in Theorem 1 requires the loss to be smooth, hence excludes the usual hinge loss.

Table 1. Mean average precisions (mAP) with least squares loss on MED14 (full details), MED13 (summary), and CCV_{sub} (summary).

	ℓ_2^2 regularized								ℓ_1 regularized						
	LS_A	LS_M	LS_T	NI-LS_1	NI-LS_{1+}	NI-LS_2	NI-LS_{2+}	ID	LS_A	LS_M	LS_T	NI-LS_1	NI-LS_{1+}	NI-LS_2	NI-LS_{2+}
MED14	0.149	0.114	0.205	0.223	0.218	0.222	0.213	21	0.132	0.105	0.186	0.218	0.226	0.163	0.157
	0.106	0.094	0.126	0.157	0.136	0.144	0.143	22	0.097	0.084	0.103	0.099	0.112	0.121	0.084
	0.735	0.714	0.814	0.853	0.808	0.831	0.789	23	0.722	0.698	0.732	0.821	0.743	0.713	0.714
	0.027	0.025	0.031	0.022	0.058	0.042	0.043	24	0.026	0.019	0.035	0.047	0.041	0.053	0.039
	0.009	0.009	0.011	0.011	0.015	0.038	0.021	25	0.008	0.010	0.008	0.018	0.009	0.009	0.007
	0.077	0.063	0.104	0.110	0.111	0.094	0.099	26	0.066	0.062	0.078	0.085	0.092	0.087	0.081
	0.142	0.148	0.135	0.205	0.143	0.188	0.195	27	0.133	0.117	0.154	0.182	0.178	0.158	0.139
	0.349	0.308	0.378	0.410	0.399	0.384	0.389	28	0.334	0.304	0.331	0.382	0.326	0.351	0.334
	0.233	0.151	0.314	0.385	0.348	0.359	0.297	29	0.218	0.200	0.255	0.296	0.269	0.288	0.256
	0.099	0.115	0.126	0.099	0.143	0.126	0.141	30	0.091	0.085	0.092	0.099	0.089	0.089	0.086
	0.761	0.760	0.738	0.779	0.746	0.819	0.812	31	0.744	0.728	0.725	0.778	0.735	0.778	0.759
	0.207	0.113	0.218	0.231	0.225	0.299	0.264	32	0.194	0.178	0.199	0.197	0.208	0.185	0.147
	0.511	0.499	0.536	0.571	0.542	0.632	0.509	33	0.502	0.493	0.487	0.532	0.495	0.513	0.452
	0.366	0.343	0.402	0.431	0.428	0.515	0.428	34	0.352	0.331	0.398	0.388	0.396	0.411	0.375
	0.423	0.329	0.418	0.463	0.435	0.503	0.413	35	0.418	0.369	0.312	0.364	0.318	0.404	0.365
	0.122	0.127	0.138	0.135	0.142	0.172	0.176	36	0.111	0.103	0.098	0.082	0.099	0.113	0.109
	0.347	0.279	0.326	0.412	0.331	0.443	0.325	37	0.332	0.318	0.338	0.361	0.342	0.312	0.284
	0.035	0.031	0.029	0.043	0.025	0.045	0.071	38	0.042	0.036	0.041	0.061	0.043	0.048	0.033
	0.417	0.341	0.388	0.524	0.411	0.521	0.453	39	0.425	0.409	0.427	0.465	0.422	0.446	0.413
	0.075	0.072	0.132	0.123	0.128	0.195	0.098	40	0.068	0.061	0.072	0.094	0.099	0.098	0.093
	0.259	0.232	0.273	0.309	0.289	0.329	0.294	mAP	0.251	0.236	0.254	0.278	0.262	0.267	0.246
MED13	0.298	0.281	0.356	0.369	0.351	0.383	0.360	mAP	0.306	0.295	0.337	0.342	0.341	0.332	0.308
CCV_{sub}	0.727	0.705	0.741	0.767	0.739	0.773	0.757	mAP	0.718	0.703	0.735	0.738	0.733	0.716	0.743

Table 2. Mean average precisions (mAP) with squared hinge loss on MED14 (summary), MED13 (summary), and CCV_{sub} (summary).

ℓ_2^2 regularized							ℓ_1 regularized							
SVM_A	SVM_M	SVM_T	NI-SVM_1	NI-SVM_{1+}	NI-SVM_2	NI-SVM_{2+}	Dataset	SVM_A	SVM_M	SVM_T	NI-SVM_1	NI-SVM_{1+}	NI-SVM_2	NI-SVM_{2+}
0.272	0.245	0.301	0.322	0.304	0.344	0.310	MED14	0.265	0.249	0.276	0.291	0.278	0.282	0.261
0.311	0.292	0.360	0.381	0.364	0.392	0.374	MED13	0.318	0.309	0.342	0.354	0.382	0.344	0.320
0.738	0.716	0.753	0.779	0.750	0.783	0.768	CCV_{sub}	0.731	0.716	0.745	0.751	0.745	0.729	0.755

Table 3. Comparing against state-of-the-art alternatives, with additional ℓ_2^2 regularization and matrix isotonic regularizer (7).

Dataset	Tang et al. (2012)	Lai et al. (2014)	Li et al. (2013)	Karpathy et al. (2014)	NI-LS_2 (Ours)	NI-SVM_2 (Ours)
MED14	0.275	0.296	0.288	0.304	*0.329*	**0.344**
MED13	0.346	0.362	0.353	0.371	*0.383*	**0.392**
CCV_{sub}	0.734	0.747	0.741	0.758	*0.773*	**0.783**

5.2. Against state-of-the-art alternatives

We further compare the performance of the proposed approaches to some state-of-the-art alternatives in Table 3. Clearly, the proposed method, with either least squares loss or the squared hinge loss, achieves the best accuracy on all three datasets. For instance, on the most recent (and challenging) MED14 dataset, the proposed method (with squared hinge loss) outperforms the second best approach by a margin as large as 4%.

5.3. Against different isotonic regularizers

Lastly we compare different isotonic regularizers that have appeared in the literature:

- $\|\mathbf{w}\|_\imath := \sum_{j=2}^m (|w_j| - |w_{j-1}|)_+$, proposed in this work.
- $\|\mathbf{w}\|_+ := \sum_{j=2}^m (w_j - w_{j-1})_+$ (Tibshirani et al., 2011).
- $\|\mathbf{w}\|_a := \sum_{j=2}^m ||w_j| - |w_{j-1}||$ (Yang et al., 2012).
- $\|\mathbf{w}\|_{\text{tv}} := \sum_{j=2}^m |w_j - w_{j-1}|$, this is the well-known total variational norm.

All of the above isotonic regularizers are extended to the matrix setting as illustrated in §4.2. Table 4 summarizes the average results on all three datasets, under the setting with the least squares loss, additional ℓ_2^2 regularizer, and the matrix extension (7). As expected, our isotonic regularizer $\|\cdot\|_\imath$ achieves the best overall performance.

Table 4. Comparing different isonotnic regularizers.

Dataset	$\|\cdot\|_\imath$	$\|\cdot\|_+$	$\|\cdot\|_a$	$\|\cdot\|_{\text{tv}}$
MED14	0.329	0.299	0.313	0.318
MED13	0.383	0.357	0.366	0.371
CCV_{sub}	0.773	0.743	0.749	0.755

6. Conclusion

Based on the observation that not all video shots are equally relevant to the event of interest, in this work we propose to prioritize the video shots using a novel notion of semantic saliency. Through a suitable isotonic regularizer we further design the "informed" nearly-isotonic SVM classifier that is able to exploit the carefully constructed ordering information. An efficient proximal gradient implementation, with new closed-form proximal steps, is developed. Extensive experiments conducted on three real-world video datasets confirm the effectiveness of the proposed approach. In the future, we plan to incorporate temporal and spatial information in a more refined notion of saliency. We also plan to deploy NI-SVM to other applications.

Acknowledgment

We thank Bin Zhao, Zhongwen Xu, and the anonymous reviewers for their valuable comments. This work was supported by NIH R01GM087694 and P30DA035778, the 973 program (2012CB316400), the ARC DECRA project, and the open project program of the state key lab of CAD&CG (GrantNo. A1402).

References

Aly, Robin, Arandjelovic, Relja, Chatfield, Ken, et al. The axes submissions at Trecvid 2013. 2013. 1

Barlow, R. E., Bartholomew, D. J., Bremner, J. M., and Brunk, H. D. *Statistical inference under order restrictions: The theory and application of isotonic regression*. John Wiley & Sons, 1972. 4

Bolte, Jérôme, Sabach, Shoham, and Teboulle, Marc. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming, Series A*, 146:459–494, 2014. 2, 5

Cao, Liangliang, Mu, Yadong, Natsev, Apostol, et al. Scene aligned pooling for complex video recognition. In *ECCV*, pp. 688–701, 2012. 1, 2

Davies, P. L. and Kovac, A. Local extremes, runs, strings and multiresolution. *The Annals of Statistics*, 29(1):1–65, 2001. 6

Jiang, Yu-Gang, Ye, Guangnan, Chang, Shih-Fu, Ellis, Daniel P. W., and Loui, Alexander C. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *ICMR*, pp. 29:1–29:8, 2011. 7

Karpathy, Andrej, Toderici, George, Shetty, Sanketh, et al. Large-scale video classification with convolutional neural networks. In *CVPR*, pp. 1725–1732, 2014. 2, 8

Koch, Christof and Ullman, Shimon. Shifts in selective visual attention: Towards the underlying neural circuitry. *Human Neurobiology*, 4:219–227, 1985. 2

Lai, Kuan-Ting, Liu, Dong, Chen, Ming-Syan, and Chang, Shih-Fu. Recognizing complex events in videos by learning key static-dynamic evidences. In *ECCV*, pp. 675–688, 2014. 3, 8

Lampert, Christoph H., Nickisch, Hannes, and Harmeling, Stefan. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 4

Laptev, Ivan, Caputo, Barbara, Schüldt, Christian, and Lindeberg, Tony. Local velocity-adapted motion events for spatio-temporal recognition. *Computer Vision and Image Understanding*, 108(3):207–229, 2007. 2

Lee, Daniel D and Seung, H Sebastian. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999. 6

Lee, Yong Jae, Ghosh, Joydeep, and Grauman, Kristen. Discovering important people and objects for egocentric video summarization. In *CVPR*, pp. 1346–1353, 2012. 2

Li, Weixin, Yu, Qian, Divakaran, Ajay, and Vasconcelos, Nuno. Dynamic pooling for complex event recognition. In *ICCV*, pp. 2728–2735, 2013. 1, 2, 8

Liu, Dong, Lai, Kuan-Ting, Ye, Guangnan, Chen, Ming-Syan, and Chang, Shih-Fu. Sample-specific late fusion for visual category recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 803–810, 2013. 2

Liu, Jingchen, McCloskey, Scott, and Liu, Yanxi. Local expert forest of score fusion for video event classification. In *European Conference on Computer Vision (ECCV)*, pp. 397–410, 2012. 2

Lowe, David G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Gregory S., and Dean, Jeffrey. Distributed representations of words and phrases and their compositionality. In *NIPS*, pp. 3111–3119, 2013. 2, 3

Natarajan, Pradeep, Wu, Shuang, Vitaladevuni, Shiv, et al. Multimodal feature fusion for robust event detection in web videos. In *CVPR*, pp. 1298–1305, 2012. 2

NIST. The Trecvid MED 2013 dataset. http://nist.gov/itl/iad/mig/med13.cfm, 2013. 2, 7

NIST. The Trecvid MED 2014 dataset. http://nist.gov/itl/iad/mig/med14.cfm, 2014. 2, 3, 7

Oneaţă, Dan, Verbeek, Jakob, and Schmid, Cordelia. Action and event recognition with Fisher vectors on a compact feature set. In *ICCV*, pp. 1817–1824, 2013. 2

Rahtu, Esa, Kannala, Juho, Salo, Mikko, and Heikkilä, Janne. Segmenting salient objects from images and videos. In *ECCV*, pp. 366–379, 2010. 2

Ramanathan, Vignesh, Liang, Percy, and Li, Fei-Fei. Video event understanding using natural language descriptions. In *ICCV*, pp. 905–912, 2013. 3

Rockafellar, Ralph Tyrell and Wets, Roger J-B. *Variational Analysis*. Springer, 1998. 5

Russakovsky, Olga, Deng, Jia, Su, Hao, et al. ImageNet Large Scale Visual Recognition Challenge, 2014. 2, 3

Simonyan, Karen and Zisserman, Andrew. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2013. 2

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3, 7

Tamrakar, Amir, Ali, Saad, Yu, Qian, et al. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, pp. 3681–3688, 2012. 2

Tang, Kevin, Li, Fei-Fei, and Koller, Daphne. Learning latent temporal structure for complex event detection. In *CVPR*, pp. 1250–1257, 2012. 3, 8

Tang, Kevin, Yao, Bangpeng, Li, Fei-Fei, and Koller, Daphne. Combining the right features for complex event recognition. In *ICCV*, pp. 2696–2703, 2013. 2

Tibshirani, Ryan J., Hoefling, Holger, and Tibshirani, Robert. Nearly-isotonic regression. *Technometrics*, 53 (1):54–61, 2011. 4, 5, 8

Vahdat, Arash, Cannons, Kevin, Mori, Greg, Oh, Sangmin, and Kim, Ilseo. Compositional models for video event detection: A multiple kernel learning latent variable approach. In *ICCV*, pp. 1185–1192, 2013. 2

Wang, Heng and Schmid, Cordelia. Action recognition with improved trajectories. In *ICCV*, pp. 3551–3558, 2013. 2

Yang, Sen, Yuan, Lei, Lai, Ying-Cheng, et al. Feature grouping and selection over an undirected graph. In *KDD*, pp. 922–930, 2012. 4, 5, 8

Yu, Shoou-I, Jiang, Lu, Xu, Zhongwen, et al. Informedia@TRECVID 2014 MED and MER. 2014. 1

Yu, Yaoliang. On decomposing the proximal map. In *NIPS*, pp. 91–99, 2013. 6, 11

Zhao, Bin, Li, Fei-Fei, and Xing, Eric P. Online detection of unusual events in videos via dynamic sparse coding. In *CVPR*, pp. 3313–3320, 2011. 3