IEEE Int. Conf. on Multimedia and Expo (ICME 2013), San Jose, CA, USA, July 2013.

VIDEO EVENT DETECTION USING A SUBCLASS RECODING ERROR-CORRECTING OUTPUT CODES FRAMEWORK

Nikolaos Gkalelis [1,2], *Vasileios Mezaris* [1], *Michail Dimopoulos* [1], *Ioannis Kompatsiaris* [1], *Tania Stathaki* [2]

[1] Information Technologies Institute / CERTH, Thermi 57001, Greece
[2] Electrical and Electronic Engineering Dept., Imperial College London, SW7 2AZ, UK

ABSTRACT

In this paper, complex video events are learned and detected using a novel subclass recoding error-correcting outputs (SRECOC) design. In particular, a set of pre-trained concept detectors along different low-level visual feature types are used to provide a model vector representation of video signals. Subsequently, a subclass partitioning algorithm is used to divide only the target event class to several subclasses and learn one subclass detector for each event subclass. The pool of the subclass detectors is then combined under a SRECOC framework to provide a single event detector. This is achieved by first exploiting the properties of the linear loss-weighted decoding measure in order to derive a probability estimate along the different event subclass detectors, and then utilizing the sum probability rule along event subclasses to retrieve a single degree of confidence for the presence of the target event in a particular test video. Experimental results on the large-scale video collections of the TRECVID Multimedia Event Detection (MED) task verify the effectiveness of the proposed method. Moreover, the effect of weak or strong concept detectors on the accuracy of the resulting event detectors is examined.

Index Terms— Semantic model vectors, event detection, subclass error-correcting output codes, loss-weighted decoding, recoding, concept detectors.

1. INTRODUCTION

High-level video event detection is now widely recognized as an essential step towards large-scale multimedia content analysis, indexing and search [1]. Due to the compositional nature of events (i.e., consisting of actions, actors, objects, locations, times and other components with possible relations among them) [2], this task is much more challenging than tasks dealing with the detection of elementary actions in video [3] or other, mostly static, semantic concepts. To deal with the inherent complexity of high-level events, typically several low-level features are extracted from the video signal in order to provide a more informative event representation. For instance, the authors in [4] exploit a late fusion strategy of three different feature types, namely, static visual (local image features extracted using a dense sampling strategy and the scale invariant feature invariant transform (SIFT)), audio (Mel-frequency cepstral coefficient (MFCC) descriptors) and dynamic visual features (dense trajectories described with the motion boundary histogram (MBH) descriptor). One support vector machine (SVM) is trained for each feature type and each event of the TRECVID 2011 Multimedia Event Detection (MED) dataset [5], and the weighted sum of the SVM output scores is used to detect the presence of an event in a test video. Similarly, in [6], a variety of features (Harris-SIFT, Hessian-SIFT, space time interest points-HOG (STIP-HOG), STIP-HOF, dense HOG, MFCC) are extracted, and a Gaussian mixture model (GMM) supervector is constructed for each feature and each video. The derived GMM supervectors are used to train one kernel SVM (KSVM) for each event in the TRECVID 2011 MED dataset, and the weighted average of the KSVM output scores is exploited for event detection.

Recently, some researchers started to exploit semantic model vectors [7] as a feature representation of high-level events, aiming at better event detection performance. The inspiration behind this modelling approach is that high-level events can be better recognized by looking at their constituting semantic entities. For instance, in [8] a set of pre-trained concept detectors are used for describing the video signal, and discriminant analysis is used to derive the most informative event concepts. These concepts are then used for describing the videos and for learning the target events. In [9, 10], large sets of low-level video features as well as semantic model vector features are extracted, and different fusion strategies are used to detect the target events. Experimental results in the above works showed that in some cases event detectors trained using the semantic model vector representation outperformed classifiers trained on state-of-the-art low-level feature representations alone [10], and that their combination with low-level features provides small but noticeable performance gains.

In the above works, fusion of different modalities is performed along different feature types in order to improve the detection performance. However, recent works on machine

This work was supported by the European Commission under contract FP7-287911 LinkedTV.

learning have shown that in various learning problems performance gains can also be achieved by combining multiple classifiers trained along different regions of the same feature space [11, 12]. Building on this, we propose in this work the combination of semantic model vectors for video event representation with a new event detection method that exploits a SRECOC framework and the loss weighted decoding (LWD) measure [11, 13, 14] to combine multiple classifiers trained at different regions of the same concept space.

The paper is organized as follows: The event detection problem is formulated in Section 2, while the model vector approach for representing events in video signals is described in Section 3. The SRECOC framework along with the formulation of the LWD measure for providing probability estimates are presented in Section 4, and experimental results in TRECVID MED 2010 and 2011 video collections are discussed in Section 5. Conclusions and future extensions are considered in Section 6.

2. PROBLEM FORMULATION

Our goal is to learn an event detector $f : \mathcal{X} \rightarrow [0,1]$ and the respective threshold $\theta \in [0,1]$ for providing a hard decision regarding the presence of the target event in the video. For this, a concept-based representation of an annotated video database is used, $\{(\mathbf{x}^p, y^p) \in \mathcal{X} \times \{-1,1\}\}$, where, $\mathcal{X} \subset [0,1]^Q$, $\mathbf{x}^p = [x^{p,1}, \ldots, x^{p,Q}]^T$ is the model vector representation of the p-th video in the dataset. I.e., $x^{p,\kappa}$ is the degree of confidence (DoC) that the κ-th concept (out of Q concepts in total) is depicted in the p-th video, and y^p is the label of the p-th video denoting the target event class ($y^p = 1$) or the "rest of the world" class ($y^p = -1$).

3. VIDEO REPRESENTATION

3.1. Low-level visual features

For the extraction of low-level visual features, we follow an approach similar to the one described in [15], as explained in the following. The visual stream of a video is decoded and represented using temporal sequences of keyframes extracted from video at fixed intervals, i.e., one keyframe every 6 seconds.

The spatial information within each keyframe image is encoded using a 1×3 spatial pyramid decomposition scheme, i.e., the entire image is the pyramid cell at the first level, and three horizontal image bars of equal size are the pyramid cells at the second level [16]. For the detection of salient image patches at the pyramid cells we use either a dense sampling strategy or the Harris-Laplace detector. The statistical properties of a local patch are captured using a set of suitable descriptors to derive an 128- or 384-dimensional feature vector depending on the type of the descriptor. Specifically, we utilize the SIFT descriptor as well as two of its color variants, RGB-SIFT and oponentSIFT [16]. Subsequently, for each of the aforementioned sampling strategies, descriptor types and pyramid cells, a Bag-of-Words (BoW) model of 1000 visual words is derived using the k-means algorithm and a large set of automatically extracted feature vectors. The assignment of the derived local feature vectors to the codebook words is done using either hard or soft assignment [17]. Therefore, in total $I = 12$ feature extraction procedures are utilized (called hereafter channels [18]), derived from every combination of sampling strategy (2 options), descriptor type (3 options) and assignment technique (2 options) described above. Applying the above procedure, the l-th keyframe of the p-th video sequence is represented with a 4000-dimensional BoW feature vector $\mathbf{z}_i^{p,l}$ in the i-th channel feature space $\mathcal{Z}_i$.

3.2. From low-level features to model vectors

A set of $Q \cdot I$ pre-trained concept detectors, $\mathcal{G} = \{g_{\kappa,i} : \mathcal{Z}_i \rightarrow [0,1] | \kappa = 1, \ldots, Q,\ i = 1, \ldots, I\}$, is utilized to provide an intermediate level representation of a video keyframe based on Q semantic concepts [8, 10]. A *weak* concept detector $g_{\kappa,i}$ is designed using a linear SVM and a training set of low-level feature vectors referring to the i-th channel (Section 3.1) and the κ-th semantic concept. To derive a *strong* concept detector $g_\kappa : \mathcal{Z}_1 \times, \ldots, \times \mathcal{Z}_I \rightarrow [0,1]$ for the κ-th semantic concept, the relevant weak concept detectors $g_{\kappa,i}$, $i = 1, \ldots, I$, are combined at the score level using the harmonic mean operator. In this way, the l-th keyframe of the p-th video in the database is associated with the model vector $\mathbf{x}^{p,l} = [x^{p,1,l}, \ldots, x^{p,Q,l}]$, where, $x^{p,\kappa,l}$ is the response of the strong concept detector g_κ expressing the DoC that the κ-th concept is depicted in the keyframe. At this point we should note that a model vector can be similarly derived using the set of the Q weak concept detectors referring to a specific single channel i.

3.3. From frame-level to video-level representation

The procedure described above provides a set of model vectors for each video (i.e., one model vector for each keyframe). In order to derive a model vector representation of the overall video, the model vectors of the individual keyframes referring to it are averaged. For instance, when using the strong concept detectors, the model vector $\mathbf{x}^p$ referring to the p-th video is computed using $\mathbf{x}^p = \frac{1}{L_p} \sum_{l=1}^{L_p} \mathbf{x}^{p,l}$, where L_p is the length of the p-th video in keyframes.

4. EVENT DETECTION

Event detectors are learned separately for each event following a target-event versus rest-of-the-world approach. A detector is derived using a splitting algorithm to partition the event class to several subclasses, then learning a number of subclass event detectors, and finally embedding the pool of the trained subclass detectors within a new variant of the ECOC framework [11, 13, 12], as explained in the following.

4.1. Subclass divisions

An iterative algorithm is applied in order to derive a subclass division of the target event class [12, 19, 20]. Starting from the initial, one subclass partition $\mathcal{X}_+^{(1)} = \mathcal{X}_+$, where $\mathcal{X}_+$ is the set of the videos that belong to the target event class, at the r-th iteration the k-means algorithm is used to divide $\mathcal{X}_+$ to r subclasses, $\mathcal{X}_+^{(r)} = \{\mathcal{X}_j^{(r)} | j = 1, \dots, r\}$.

At each iteration the following nongaussianity measure is computed along the partitions [19]

$$\Phi^{(r)} = \frac{1}{r}\sum_{j=1}^{r}(\gamma_j + \beta_j), \tag{1}$$

where, $\gamma_j = \frac{1}{Q}\sum_{\kappa=1}^{Q} |\gamma_j^\kappa|$, $\beta_j = \frac{1}{Q}\sum_{\kappa=1}^{Q} |\beta_j^\kappa - 3|$ are estimates of the multivariate standardized skewness and kurtosis of the j-th subclass respectively. These are based on estimates of their one-dimensional counterparts, which along the κ-th dimension can be calculated using $\gamma_j^\kappa = (\frac{1}{P_j}\sum_{x^{p,\kappa} \in \mathcal{X}_j^{(r)}} (x^{p,\kappa} - \mu_j^\kappa)^3)/(\sigma_j^\kappa)^3$ and $\beta_j^\kappa = (\frac{1}{P_j}\sum_{x^{p,\kappa} \in \mathcal{X}_j^{(r)}} (x^{p,\kappa} - \mu_j^\kappa)^4)/(\sigma_j^\kappa)^4$ respectively. In the above equations P_j is the number of videos of the j-th subclass, $x^{p,\kappa}$ is the κ-th element of the p-th model vector belonging to the j-th subclass, and $\mu_j^\kappa, \sigma_j^\kappa$, are the sample mean and standard deviation of the j-th subclass along the κ-th dimension, respectively.

At the end of this iterative algorithm, the best subclass partition $\mathcal{X}_+^{(H_1)}$ is selected according to the following rule

$$\mathcal{X}_+^{(H_1)} = \underset{r \in [1,R]}{\text{argmin}}(\Phi^{(r)}), \tag{2}$$

where, R is the total number of iterations and H_1 is the number of subclasses of the target event class corresponding to the derived optimal subclass partition.

4.2. SRECOC framework

The application of the iterative algorithm presented above will provide a subclass division of the overall training dataset $\mathcal{X} = \{\mathcal{X}_1, \dots, \mathcal{X}_{H_1}, \mathcal{X}_-\}$, of $H = H_1 + 1$ total subclasses, where $\mathcal{X}_-$ is the set of videos that belong to the "rest of the world" class. Thus, the video dataset is described at subclass level, $\{(\mathbf{x}^p, u^p) \in \mathcal{X} \times \{1, \dots, H_1, -1\}\}$, where, u^p is the subclass label of the p-th video denoting that it belongs to one of the subclasses of the target event class ($u^p \in [1, H_1]$) or to the "rest of the world" class ($u^p = -1$). The derived subclass division is exploited using a ternary SRECOC framework. As explained in [14], the advantage of using a recoding step is that the generalization capability of ECOC (or SECOC) is improved without considerably affecting its training and testing time.

Specifically, a variant of the one-versus-one subclass strategy is used, where binary problems are defined only for subclasses of different classes, similar to [12]. During the coding step, a set of binary subclass classifiers $\mathcal{A} = \{a_j : \mathcal{X} \rightarrow [0,1] | j = 1, \dots, H_1\}$ are utilized, where, the j-th detector is trained using as positive samples the model vectors of the j-th subclass ($u^p = j$) and as negative samples the videos with negative label ($u^p = -1$). In addition to the above set of detectors, a last detector a_H is trained, using as positive samples all samples of the target event, and as negative the rest of the world event samples. Consequently, a codeword $\mathbf{m}_k \in \{1, 0, -1\}^{1 \times H}$, $k \in [1, H]$ is designed for each subclass, where the codeword referring to the rest of the world event class is defined as $\mathbf{m}_H = [-1, -1, \dots, -1]$. In contrary, the elements of the codewords referring to the target event subclasses receive one of the other two ternary digits, i.e,

$$m_{k,j} = \begin{cases} 1 & \text{if } j = k \text{ or } j = H; \\ 0 & \text{else,} \end{cases} \tag{3}$$

where $k \in [1, H_1], j \in [1, H]$. The above codewords are then used as rows of the so-called coding matrix $\mathbf{M} \in \{1, 0, -1\}^{H \times H}$.

Moreover, in order to update $\mathbf{M}$, following the conventional recoded ECOC (RECOC) [14] and pursuing a Loss-Weighted decoding (LWD) scheme, the weighting matrix $\tilde{\mathbf{M}} \in \mathbb{R}^{H \times H}$ is calculated using the training set and the derived subclass classifiers [13]. This is done by firstly computing the performance matrix $\mathbf{B} \in \mathbb{N}^{H \times H}$, whose element $b_{k,j}$ corresponds to the performance of a_j on classifying the training samples belonging to the k-th subclass

$$b_{k,j} = \frac{1}{P_k}\sum_{p=1}^{P_k} s_{k,j}^p, \tag{4}$$

$$s_{k,j}^p = \begin{cases} 1 & \text{if } a_{k,j}^p \geq \theta_j; \\ 0 & \text{else,} \end{cases} \tag{5}$$

where, $s_{k,j}^p$, $a_{k,j}^p$ are the response and DoC of the j-th indicator function and detector respectively, with respect to the p-th model vector of the k-th subclass, θ_j is the detection threshold referring to the j-th detector, and P_k is the number of videos of k-th subclass. The weighting matrix is then obtained by normalizing each row $\mathbf{b}_k$ of $\mathbf{B}$ to unit $l1$ norm, i.e., $\tilde{m}_{k,j} = b_{k,j} / \parallel \mathbf{b}_k \parallel_1$ so that $\parallel \tilde{\mathbf{m}}_k \parallel_1 = 1$, where $\| \|_1$ is the $l1$ norm function. The above normalization effectively allows the treatment of $\tilde{\mathbf{M}}$ as a discrete probability density function. Subsequently, a performance threshold $\varphi \in [0.1, 1]$ is used to update (recode) the positions of $\mathbf{M}$ coded with zero according to the following rule

$$\breve{m}_{k,j} = \begin{cases} 1 & \text{if } \tilde{m}_{k,j} > \varphi \cdot \tilde{m}_{k,k} \ \ \& \ \ m_{k,j} = 0 \\ m_{k,j} & \text{else,} \end{cases} \tag{6}$$

where, $\breve{\mathbf{M}}$ is the recoded matrix, and $k \in [1, H_1], j \in [1, H]$.

During the decoding stage, a test model vector $\mathbf{x}^t$ is classified to one of the subclasses by first evaluating the H_1 subclass detectors in order to create a codeword for it, and then comparing the derived codeword with the base codewords in the coding matrix referring only to the target event subclasses. For the comparison of the codewords we use the linear LWD measure considering the intersection of the confidence intervals derived from the subclass classifiers [13]

$$d_k^t = -\sum_{j=1}^{H_1} \breve{m}_{k,j} a_j^t \tilde{m}_{k,j},\ k = 1, \dots, H_1\,, \tag{7}$$

where, $\breve{m}_{k,j}, \tilde{m}_{k,j}$ are the elements of the recoded and weighting matrix, respectively, that correspond to the j-th subclass and the detector that separates the k-th subclass from the "rest of the world" class. Note that $\breve{m}_{k,j} \in \{0, 1\}$, $\tilde{m}_{k,j}, a_j^t \in [0, 1]$, $\sum_{j=1}^{H_1} \tilde{m}_{k,j} = 1$, $\forall k, j$, and therefore $d_k^t \in [-1, 0]$. To this end, in order to derive a probability estimate for the j-th subclass, we negate the LWD distance $\pi_k^t = -d_k^t$. Finally, considering that all detectors refer to subclasses of the target event, i.e., they can be considered as expert detectors of the event in a subregion of the concept space, an overall DoC f^t regarding the presence of the event in the test video is obtained using the sum probability rule under the equal prior assumption along the event subclasses [21]

$$f^t = \frac{1}{H_1} \sum_{k=1}^{H_1} \pi_k^t. \tag{8}$$

The test video is then classified to the target event according to the rule $f^t \geq \theta$, where, $\theta \in [0, 1]$ is the detection threshold value estimated using a cross-validation procedure.

5. EXPERIMENTAL EVALUATION

5.1. Dataset description

The video datasets of the TRECVID MED 2010 and 2011 tasks are used for the evaluation of the proposed algorithm and for comparison with the kernel SVM (KSVM) [10, 22]. The former dataset (TRECVID MED 2010) consists of 1745 development and 1742 test videos belonging to one of 3 target events ("assembling a shelter", "batting a run in" and "making a cake") or to the "rest of the world" event class. For the annotation of the videos we employ the labelling information provided in [10]. The TRECVID MED 2011 consists of 13871 development videos, 32061 test videos and 11 event classes, i.e, the "rest of the world" event class and 10 target event classes: "birthday party", "changing a vehicle tire", "flash mob gathering", "getting a vehicle unstuck", "grooming an animal", "making a sandwich", "parade", "parkour", "repairing an appliance", "working on a sewing project". On average, around 50 and 130 videos per event of interest are included in the development collection of the TRECVID MED 2010 and MED 2011 dataset, respectively.

5.2. Evaluation measure

For assessing the performance of the individual target event detectors the average precision (AP) is used. The AP summarizes the shape of the precision recall curve and for the n-th event it is computed as follows

$$AP_n = \frac{1}{M_n} \sum_{s=1}^{S} \frac{M_n^s}{s} R_s, \tag{9}$$

where, S is the total number of test samples, M_n is the number of samples of the n-th event in the test set, M_n^s is the number of samples of the n-th event in the top s ranked samples returned by the detection method, and R_s is an indicator function with $R_s = 1$ if the s-th video in the ranked list belongs to the n-th event and $R_s = 0$ otherwise. The overall performance of a method along all events in a dataset is measured using the mean average precision (MAP) defined as the mean AP along all the events in the database, i.e., $MAP = \sum_{n=1}^{N} AP_n$, where N is the total number of the target events in the dataset.

5.3. Experimental setup

The TRECVID SIN 2012 dataset is used to derive one weak concept detector for each of the $Q = 346$ TRECVID SIN 2012 Task concepts and for each of the $I = 12$ channels. Additionally, a set of $Q = 346$ strong concept detectors is also formulated as described in Section 3.2. Subsequently, following the procedure described in Section 3, each video in the evaluation set is decoded, and one keyframe every 6 seconds is uniformly selected. A set of 13 model vectors for each keyframe is then retrieved using the 12 weak concept detectors as well as the strong concept detector described above. Finally, the model vectors referring to the same video and the same type of concept detectors are averaged, providing 13 model vectors in $\mathbb{R}^{346}$ for each video. Then, we form 3 evaluation sets of model vectors:

1) TRECVID MED 2010 - 346 weak concept detectors: this set consists of the TRECVID MED 2010 model vectors derived using the weak concept detectors referring to the channel combining the dense sampling strategy, the oponentSIFT descriptor and the soft assignment BoW technique (called hereafter O_1 channel).

2) TRECVID MED 2010 - 346 strong concept detectors: this set consists of the TRECVID MED 2010 model vectors derived using the strong concept detectors.

3) TRECVID MED 2011 - 172 weak concept detectors: this set consists of the TRECVID MED 2011 model vectors created using 172 of the weak concept detectors of the O_1 channel.

Our choice to exploit the O_1 channel in our experiments with weak concept detectors is based on the recommendation by several researchers that this channel provides the best results (e.g., see [16]). In this way, for MED 2010 we can com-

Table 1. Performance evaluation on the TRECVID MED 2010 dataset using weak concept detectors.

Event	*KSVM*	*SRECOC*	*% Boost*
Assembling a shelter	0.20371	0.20472	0.4%
Batting a run in	0.64855	0.65492	1%
Making a cake	0.28803	0.30448	5.7%
MAP	0.3801	0.38804	2.1%

Table 2. Performance evaluation on the TRECVID MED 2010 dataset using strong concept detectors.

Event	*KSVM*	*SRECOC*	*% Boost*
Assembling a shelter	0.25102	0.26869	7%
Batting a run in	0.74314	0.75356	1.4%
Making a cake	0.20375	0.25396	24.6%
MAP	0.3993	0.4254	6.5%

Table 3. Performance evaluation on the TRECVID MED 2011 dataset using weak concept detectors.

Event	*KSVM*	*SRECOC*	*% Boost*
Birthday party	0.02601	0.02967	14.1%
Changing a vehicle tire	0.13865	0.13823	-0.3%
Flash mob gathering	0.26711	0.27328	2.3%
Getting a vehicle unstuck	0.11441	0.12168	6.3%
Grooming an animal	0.02705	0.04902	81%
Making a sandwich	0.05381	0.06525	21.2%
Parade	0.10639	0.11798	10.1%
Parkour	0.09069	0.09565	5.6%
Repairing an appliance	0.16934	0.19155	13.1%
Working on a sewing project	0.07105	0.09184	29.3%
MAP	0.10645	0.11742	10.3%

pare the event detection performance of a method that uses strong concept detectors with the one using the best weak concept detectors. The event detectors for each method and for each of the 3 evaluation sets described above are then created using the corresponding development set. For the KSVM and the base classifiers of SRECOC we used the KSVM implementation provided in the libsvm package [23] with Gaussian radial basis function (RBF) kernel. During training, we need to estimate the scale parameter σ of the RBF kernel and the penalty term C of the SVM, while for the SRECOC we additionally require the estimation of the recoding threshold φ. Following the recommendation in [1], we set the scaling parameter σ to the mean of the pairwise distances between the model vectors in the development set. The other two parameters C and/or φ are estimated through a grid search on a 3-fold cross-validation procedure, where at each fold the development set is split to 70% training set and 30% validation set. The estimated parameters are then applied to the overall development set in order to derive the target event detectors.

5.4. Results

The performance of the SRECOC and KSVM in terms of AP and MAP on the 3 evaluation sets described above are shown in Tables 1, 2 and 3. From the analysis of the obtained results we observe that in the case of the weak concept detectors, SRECOC provides an approximate boost in performance over KSVM of 2.1% and 10.3% in terms of MAP for the TRECVID MED 2010 and TRECVID MED 2011 dataset respectively; when the strong concept detectors are used, the boost in performance in the TRECVID MED 2010 dataset is increased to 6.5%. The small improvement in TRECVID MED 2010 dataset with weak concept detectors is explained by considering the fact that this dataset is small and noisy (due to the weak concept detectors) and thus the base subclass KSVMs of SRECOC overfit the data. Increasing the robustness of the features by applying the strong concept detectors in TRECVID MED 2010, or using the much larger TRECVID MED 2011 development set, a noticeable performance gain is achieved by SRECOC over KSVM.

Another important conclusion is inferred by the comparison of the performance between the strong concept detectors and the weak concept detectors in the TRECVID MED 2010 dataset. In terms of MAP, the strong concept detectors outperform their weak counterpart. However, in the case of the "making a cake" event the weak concept detectors are superior. We attribute this paradox to the fact that the procedure for building strong concept detectors from the weak ones (which is concept-independent; see Section 3.2) indeed increases the accuracy of concept detectors on average, but does not necessarily do so for every single one of the considered concepts. Therefore, the set of strong concept detectors may include, for specific concepts, detectors that are actually weaker than the corresponding detectors of the weak detector set, and this may affect performance for events that depend a lot on these concept detectors.

Finally, we should also note that a model vector approach (exploiting 280 semantic concepts) in combination with KSVMs (which is the approach that we use as our baseline for comparison) was proposed in [10] and was used for the detection of the 3 events in TRECVID MED 2010 dataset, achieving MAP $\simeq 0.4$. The performance of our method, exploiting the strong concept detectors in combination with KSVM or SRECOC is equivalent or better, respectively, compared to the corresponding performance reported in [10].

6. CONCLUSIONS AND FUTURE WORK

A method that uses a concept-based representation and exploits an error-correcting output framework for detecting high-level events in video was proposed. Experimental results on the TRECVID MED task datasets verified the effectiveness of the proposed method for event detection in large-scale video collections and showed that it favorably compares to the corresponding state-of-the-art KSVM approach [10]. Moreover, the effect of weak and strong concept detectors in the performance of the event detection system was examined, indicating that a concept-dependent method for combining weak detectors may be useful for improving event detection.

Straightforward extensions of the proposed method include the incorporation of event detectors trained along subclasses of different feature spaces [10] and/or the exploitation of a more suitable weighting scheme for combining the weak concept detectors, as explained above. We plan to investigate this possibility, as well as the possibility of extending the proposed method for the task of multimedia event recounting.

7. REFERENCES

[1] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah, "High-level event recognition in unconstrained videos," *Int. J. Multimed. Info. Retr.*, Nov. 2013.

[2] N. R. Brown, "On the prevalence of event clusters in autobiographical memory," *Social Cognition*, vol. 23, no. 1, 2005.

[3] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1473–1488, Nov. 2008.

[4] M. Ayari et al., "INRIA at TRECVID'2011: Copy detection & multimedia event detection," in *Proc. TRECVID 2011 Workshop*, Gaithersburg, MD, USA, Dec. 2011.

[5] P. Over et al., "TRECVID 2011 - goals, tasks, data, evaluation mechanisms and metrics," in *Proc. TRECVID 2011 Workshop,*, Gaithersburg, MD, USA, Dec. 2011.

[6] Y. Kamishima et al., "Tokyotech+canon at TRECVID 2011," in *Proc. TRECVID 2011 Workshop*, Gaithersburg, MD, USA, Dec. 2011.

[7] J. Smith, M. Naphade, and A. Natsev, "Multimedia semantic indexing using model vectors," in *Proc. ICME*, Baltimore, MD, USA, July 2003, pp. 445–448.

[8] N. Gkalelis, V. Mezaris, and I. Kompatsiaris, "High-level event detection in video exploiting discriminant concepts," in *Proc. 9th Int. Workshop CBMI*, Madrid, Spain, June 2011, pp. 85–90.

[9] P. Natarajan et al., "BBN VISER TRECVID 2011 multimedia event detection system," in *Proc. TRECVID 2011 Workshop*, Gaithersburg, MD, USA, Dec. 2011.

[10] M. Merler, B. Huang, L. Xie, G. Hua, and A. Natsev, "Semantic model vectors for complex video event recognition," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 88–101, Feb. 2012.

[11] S. Escalera, D. M. Tax, O. Pujol, P. Radeva, and R. P. Duin, "Subclass problem-dependent design for error-correcting output codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1041–1054, June 2008.

[12] N. Gkalelis, V. Mezaris, I. Kompatsiaris, and T. Stathaki, "Linear subclass support vector machines," *IEEE Signal Process. Lett.*, vol. 19, no. 9, pp. 575–578, Sept. 2012.

[13] S. Escalera, O. Pujol, and P. Radeva, "On the decoding process in ternary error-correcting output codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 120–134, Jan. 2010.

[14] S. Escalera, O. Pujol, and P. Radeva, "Recoding error-correcting output codes," in *Proc. 8th Int. Workshop on Multiple Classifier Systems*, Reykjavik, Iceland, June 2009, pp. 11–21.

[15] A. Moumtzidou et al., "ITI-CERTH participation to TRECVID 2012," in *Proc. TRECVID 2012 Workshop*. Nov. 2012, Gaithersburg, MD, USA.

[16] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, Sept. 2010.

[17] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek, "Visual word ambiguity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1271–1283, Sept. 2010.

[18] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *Int. J. Comput. Vision*, vol. 73, no. 2, pp. 213–238, June 2007.

[19] N. Gkalelis, V. Mezaris, and I. Kompatsiaris, "Mixture subclass discriminant analysis," *IEEE Signal Process. Lett.*, vol. 18, no. 5, pp. 319–332, May 2011.

[20] N. Gkalelis, V. Mezaris, I. Kompatsiaris, and T. Stathaki, "Mixture subclass discriminant analysis link to restricted gaussian model and other generalizations," *IEEE Trans. Neural Netw. Learn. Syst*, vol. 24, no. 1, pp. 8–21, Jan. 2013.

[21] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.

[22] V. Vapnik, *Statistical learning theory*, New York: Willey, 1998.

[23] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. on Intell. Syst. and Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.