Multimodal Feature Fusion for Robust Event Detection in Web Videos

Pradeep Natarajan, Shuang Wu, Shiv Vitaladevuni, Xiaodan Zhuang,
Stavros Tsakalidis, Unsang Park, Rohit Prasad and Premkumar Natarajan
Speech, Language and Multimedia Business Unit
Raytheon BBN Technologies, Cambridge, MA 02138
{pradeepn,swu,svitalad,xzhuang,stavros,upark,rprasad,pnatarajan}@bbn.com

Abstract

Combining multiple low-level visual features is a proven and effective strategy for a range of computer vision tasks. However, limited attention has been paid to combining such features with information from other modalities, such as audio and videotext, for large scale analysis of web videos. In our work, we rigorously analyze and combine a large set of low-level features that capture appearance, color, motion, audio and audio-visual co-occurrence patterns in videos. We also evaluate the utility of high-level (i.e., semantic) visual information obtained from detecting scene, object, and action concepts. Further, we exploit multimodal information by analyzing available spoken and videotext content using state-of-the-art automatic speech recognition (ASR) and videotext recognition systems. We combine these diverse features using a two-step strategy employing multiple kernel learning (MKL) and late score level fusion methods. Based on the TRECVID MED 2011 evaluations for detecting 10 events in a large benchmark set of ∼45000 videos, our system showed the best performance among the 19 international teams.

1. Introduction

There has been considerable interest in recent years for developing techniques that can rapidly analyze a large number of user-generated videos in websites like YouTube [34, 24, 32, 25]. Bag-of-words approaches [6] based on low-level visual features have shown promise in large scale image retrieval and action recognition in unstructured videos [31, 15]. However, limited attention has been paid to combining such features with information from other modalities, such as audio and videotext, for large scale analysis of web videos. Further, there are typically far more negative examples than positive examples (e.g., MED'11 dataset), which induces a strong bias toward the negative class in many traditional machine learning approaches.

In our work, we present an approach to address these challenges using a two stage feature fusion strategy. In the first stage, we combine multiple low-level audio and visual features using a state-of-the-art, fast multiple kernel learning (MKL) approach presented in [29]. We evaluate the individual performance of different low-level features modeling motion [11, 31], grayscale appearance [16, 1], color [27] and audio [7], as well as different combinations of them. As expected, combining diverse feature sets produces the best performance, but even combining similar features like STIP [11] and HoGHoF3D [31] produces better performance than individual features. To avoid learning trivial classifiers that declare all examples as negative, we perform an extensive grid search of parameters in MKL. At each parameter setting, we estimate a detection threshold using k-fold validation and evaluate performance using a metric corresponding to a weighted sum of missed detection and false alarm rates. This framework allows us to learn robust classifiers that show good generalization on unseen test data. Furthermore, it can be used to optimize for other metrics such as area under curve (AUC) or mean average precision (MAP).

In the second stage, we combine multiple MKL-based subsystems using a late, score-level fusion strategy. We do this in two steps: first, we use a *double sigmoid* score normalization to model the large number of negative samples clustered in a small range below the detection threshold, and the small number of positive samples distributed in a large range of scores above the threshold; and in the second step, we combine the normalized score from multiple sub-systems to get the final event detection score for each video. For this step, we tested a Bayesian model combination (BayCom) [23] approach using parametric models learned from training data, and we also propose a novel non-parametric fusion strategy based on *video specific* weighted average fusion. We compare our approach to several standard techniques based on average, maximum and voting of system scores, and found weighted average fusion to consistently outperform other approaches, including BayCom. The use of a large set of low and high level visual and audio

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

features and advanced early and late fusion strategies contrast our work from earlier studies, such as [25].

The rest of the paper is organized as follows - Sections 2 and 3 describe the suite of low-level audio and visual features that we use and different coding and pooling strategies. Sections 4∼6 describe our approaches to exploit object content, speech and videotext, respectively. Section 7 and 8 describe fusion strategies. Section 9 presents experimental results, and section 10 summarizes our work.

2. Low-level Features

We extract a large set of low-level features from video, which can be grouped into four broad classes: *Appearance*, *Color*, *Motion* and *Audio*.

2.1. Appearance Features

Appearance features model local shape patterns by aggregating quantized gradient vectors in grayscale images. We use the following appearance features in our system:

SIFT [16]: These features are among the most widely used in vision and use a difference of Gaussians (DoG) approach to detect interest points at different scales. A 128 dimensional feature descriptor is then extracted at each point to capture local image gradients. These descriptors are scale invariant and robust to affine distortion.

SURF [1]: The speeded-up robust features (SURF) are similar in principle to SIFT, but are several times faster to extract. They compute sums of 2D Haar Wavelet response and are potentially more robust to image transformations compared to SIFT.

D-SIFT [2]: This is a dense version of SIFT where, instead of detecting interest points, the 128-dimensional feature vectors are extracted by uniformly sampling over the image. D-SIFT typically generates 3 times the number of points generated by SIFT and has been shown to outperform SIFT for image classification [2].

CHoG [3]: The compressed histogram-of-oriented gradient features use a low bit-rate feature descriptor with a 20 times reduction in bit-rate compared to SIFT and other features. They have shown competitive performance in image retrieval tasks.

2.2. Color Features

These features, proposed in [27], extend SIFT features by splitting the image into color planes, computing descriptors in each plane, and then concatenating them for each detected interest point. We consider 3 different color features:

RGB-SIFT: This feature splits an image in the RGB color space and computes SIFT descriptors for each interest point, in each of the color planes, before concatenating them.

Opponent SIFT: Similar to RGB-SIFT, but splits each image in the Opponent color space.

C-SIFT: Builds on Opponent SIFT by using C-invariants to eliminate intensity in the opponent space.

2.3. Motion Features

We use two features for modeling motion patterns:

STIP [11]: The Space-Time Interest Points (STIP) extends the notion of spatial interest points to the spatio-temporal domain by detecting 3D corners, then computing scale-invariant spatio-temporal descriptors from image gradients and optical flow.

HoGHoF3D [31]: These features are similar to STIP, except that the points are sampled from a uniform spatio-temporal grid. They are the slowest features in our system to extract and take 3X as long as STIP to extract.

2.4. Audio Features

We extract the following low-level features from the audio stream:

MFCC [7]: The mel-frequency cepstral coefficients are widely used in speech processing and transform the raw audio into a 45 dimensional feature stream using the following steps. Features are extracted from overlapping frames of audio data, and each frame is windowed with a Hamming window to compute a power spectrum for the frequency band 80-6000 Hz. From this, 14 Mel-warped cepstral coefficients are computed. These base cepstral features with their first and second derivatives, together with audio energy and its first and second derivatives, compose the 45-dimensional feature vector.

FDLP [17]: We also incorporate the Frequency Domain Linear Prediction (FDLP) feature into the system. The FDLP model, in contrast to the short-term analysis by MFCC, is based on linear prediction on different frequency bands, and describes the perceptually dominant peaks and removes the finer-scale details. The resultant FDLP feature has 588 dimensions and has been shown to perform well when channel distortion varies.

Audio Transients [5]: In contrast to MFCC, these features do not have uniformly-spaced frames and instead focus on audio transients. First, spectrograms are extracted with varying window lengths and high-magnitude in any of the frequency bins of these windows proposes a candidate transient event. At each event time, a spectrogram is extracted from the temporal neighborhood, and represented using a vector representation.

3. Coding and Pooling Strategies

We represent the information from different feature descriptors using the popular bag-of-words representation. This consists of two steps - *coding*, where the descriptors are projected to a pre-trained codebook of descriptor vectors, and *pooling*, which aggregates the projections to a fixed length feature vector. Following the notations in [2],

let a video V be represented by a set of low-level descriptors x_i, where $i \in \{1 \dots N\}$ is the set of locations. Let M denote the different spatial/spatio-temporal regions of interest [13, 12], and N_m denote the number of descriptors extracted within region m. let f and g denote the coding and pooling operators respectively. Then, the vector z representing the whole video is obtained by sequentially coding, pooling over all regions and concatenating:

$$\alpha_i = f(x_i) \qquad i = 1..N \tag{1}$$

$$\mathbf{h}_m = g\left(\{\alpha_i\}_{i \in N_m}\right) \quad m = 1, \dots, M \tag{2}$$

$$\mathbf{z}^T = [h_1^T \dots h_M^T] \tag{3}$$

Coding: In hard quantization, we assign each feature vector x_i to the nearest codeword, from a codebook learnt using k-means or a similar unsupervised clustering algorithm:

$$\alpha_i \in \{0,1\}^K, \alpha_{i,j} = 1 \Leftrightarrow j = arg \min_{k \leq K} ||x_i - c_k||^2 \tag{4}$$

where c_k is the k^{th} codeword. In soft quantization [28], the assignment of the feature vectors to codewords is distributed as

$$\alpha_{i,j} = \frac{\exp\left(-\beta||x_i - c_j||^2\right)}{\sum_{k=1}^{K} \exp\left(-\beta||x_i - c_k||^2\right)} \tag{5}$$

where β controls the soft assignment. Sparse coding [18] uses a linear combination of a small number of codewords to approximate x_i. One popular approach that we use is to optimize:

$$\min_{\alpha} ||\mathbf{x} - \mathbf{D}\alpha||_2, \ s.t. \ ||\alpha||_0 \leq k \tag{6}$$

Pooling: The two most popular pooling strategies are average and max. In average pooling, we take the average of the α_i assigned to different codewords for different feature vectors as

$$\mathbf{h} = \frac{1}{N} \sum_{i=1}^{N} \alpha_i. \tag{7}$$

In max pooling, we take the maximum of the α_i's as

$$\mathbf{h} = \max_{i=1..N} \alpha_i. \tag{8}$$

These pooling techniques, in effect map the set of projections to a codeword, to a single scalar value and have shown good performance for image classification tasks, where we pool hundreds of feature projections. In videos, the number of interest points extracted is of the order of $10^5 \sim 10^6$ and pooling all these features to a single scalar is suboptimal. Recently, a pooling technique called *alpha histogram* was introduced in [30], where the α_i values are aggregated in a histogram, and showed promising results for a video classification task. In our work, we evaluate performance of these different coding and pooling strategies for each feature type.

4. Object Detection

Intuitively, object cues seem likely to be important for activity analysis; there have been several studies in recent years, e.g., [9] showed the mutual dependence between movement dynamics and objects involved in the action, and [26] extended bag-of-words models with object and scene context and applied it the challenging Hollywood 2 action dataset. In a related study [14] used a large set of 200 object detectors for high-level scene classification using the maximum response of the detectors as a feature.

Drawing upon work such as [26] and [14], we define a probabilistic representation of object detections, referred to as spatial probability map, and computed as follows. A state-of-the-art object detector [8], is used to compute detections in each video frame and the pixels within the detections' bounding boxes are set to 1 to create a detection mask for each object concept. The detections masks are averaged over the duration of the video, and normalized to an $n \times n$ grid to get the spatial probability map. The feature provides robustness to missed and false detection due to averaging over time, and encodes the expected location and spatial extent of the various object concepts. In our experiments, a 16×16 grid was used giving a 256 dimensional vector for each video for each object concept. Figure 1 shows an example frame from a "vehicle getting unstuck" video with the detection results, and the spatial probability map obtained by average over the duration of the video. Notice how the map encodes presence of two "blobs" and their approximate locations.

Figure 1. Example car detection result and spatial probability map.

5. Automatic Speech Recognition

Human language content is often present in consumer videos in the form of the spoken content in the audio track. Such content could potentially provide useful information for detecting events of interest. For example, in videos of tutorials about making dishes and documentaries about particular expeditions, the accompanying spoken narrative provides information about the category of the video. Besides, the semantic information from human language is typically complementary to the information from low-level visual features. Our approach for using the spoken language information in the audio track involves the following three

modules.

First, within the video clips, the speech segments are identified by a speech activity detection (SAD) system. The SAD system employs two Gaussian mixture models (GMM), for speech and non-speech observations respectively. A small subset of 101 video clips is annotated for speech segments, which are used for training the speech GMM. Besides the non-speech segments in this set, we also use 500 video clips with no speech at all to enrich the non-speech model, in order to handle the heterogeneous audio data in MED'11.

Second, we apply a large-vocabulary automatic speech recognition (ASR) system to the speech data to produce a transcript of the spoken content. This system is adapted from an ASR system trained on 1700-hour broadcast news. In particular, we adapt the lexicon and language model using text descriptions of the events of interest and related web text data. The acoustic models are adapted during ASR decoding for each video clip in an unsupervised fashion.

Finally, to leverage the hypothesized speech transcripts in event detection, we use the distribution of a set of event-discriminating keywords within each video clip. The hypothesized speech transcripts, stop words removed, are normalized and then stemmed by the Porter stemmer [21]. We identify event-discriminating keywords according to a revised TF-IDF criteria:

$$\left(\frac{n}{t}\right)^a \log\left(\frac{d}{h}\right) \tag{9}$$

where n is the number of times a word appears in a video clips belonging to a particular event category; t is the total number of words in that category; d is the total number of categories considered; h is the number of categories containing the word; a is an exponential weight. For each video clip, the counts of these keywords are normalized to form a histogram of keywords within that clip. We choose a to be 0.25 and the top 2251 keywords are selected, forming a histogram of length 2251.

6. Videotext Recognition

We detect text in videos, namely, videotext and use it in our overall system. Unlike speech, the occurrence of videotext is quite sparse. Hence, it is difficult to learn sophisticated dictionary and language models from the available training data. Therefore, we create a small 128-word dictionary based on words occurring in short textual descriptions of each event. In order to reduce the false positives from videotext, we also construct salient word pairs the co-occur in each event's description.

During classification, given a test video, we perform videotext detection/OCR and eliminate all the special characters to get the final OCR output for event detection. We then match each word w in the output with dictionary words d using string edit distance. These scores are then normalized using word lengths to get the normalized edit distance $NED(w,d)$ measure.
Then we compute co-occurrence scores for each word pair (w_1,w_2) occurring in a frame, with each chosen dictionary word pair (d_1,d_2) as

$$(1 - NED(w_1, d_1)) * (1 - NED(w_2, d_2)). \tag{10}$$

Finally, for each event we compute the probability of an event by taking the maximum of the co-occurrence scores corresponding to that event over the entire video.

7. Kernel-based Feature Fusion

For our early fusion strategy, we combine multiple features using $p-$norm Multiple Kernel Learning (MKL), with $p>1$. For each feature, we first compute χ^2 kernels using the samples in the training set as

$$K(\mathbf{x}, \mathbf{y}) = e^{-\rho \sum_i \frac{(x_i - y_i)^2}{x_i + y_i}}. \tag{11}$$

Given a set of kernels $\{K_k\}$ for a set of features, our aim is to learn a linear combination of the base kernels as $K{=}\sum_k d_k K_k$. This is equivalent to concatenating the corresponding weighted feature maps $\sqrt{d_k}\phi_k$ and then learning a standard support vector machine (SVM) classifier. The primal of the MKL problem can be formulated as

$$\min_{w,b,\xi\geq 0,d\geq 0} \frac{1}{2}\sum_k \mathbf{w}_k^t \mathbf{w}_k + C\sum_i \xi_i + \frac{\lambda}{2}\left(\sum_k d_k^p\right)^{\frac{2}{p}}$$
$$s.t.\ y_i\left(\sum_k \sqrt{d_k}\mathbf{w}_k^t \phi_k(x_i) + b\right) \geq 1 - \xi_i. \tag{12}$$

This primal can be made convex by substituting w_k for $\sqrt{d_k}w_k$ to get

$$\min_{w,b,\xi\geq 0,d\geq 0} \frac{1}{2}\sum_k \mathbf{w}_k^t \mathbf{w}_k/d_k + C\sum_i \xi_i + \frac{\lambda}{2}\left(\sum_k d_k^p\right)^{\frac{2}{p}}$$
$$s.t.\ y_i\left(\sum_k \mathbf{w}_k^t \phi_k(x_i) + b\right) \geq 1 - \xi_i. \tag{13}$$

With this formulation, smaller values of p, close to 1 learns a sparse set of kernels, while larger values of p choose denser kernel combinations. Recently, an efficient algorithm for optimizing this objective function was proposed in [29]. This allows use of the sequential minimal optimization (SMO) [20] for training large scale SVMs, within the MKL framework. This is done by first computing the La-

grangian

$$L = \frac{1}{2}\sum_k \mathbf{w}_k^t \mathbf{w}_k / d_k + \sum_i (C - \beta_i)\,\xi_i + \frac{\lambda}{2}\left(\sum_k d_k^p\right)^{\frac{2}{p}} - \sum_i \alpha_i \left[y_i \left(\sum_k \mathbf{w}_k^t \phi_k(x_i) + b \right) - 1 + \xi_i \right] \quad (14)$$

and then computing the l_p-MKL dual as

$$D = \max_{\alpha \in A} \mathbf{1}^t \alpha - \frac{1}{8\lambda}\left(\sum_k \left(\alpha^t H_k \alpha\right)^q\right)^{\frac{2}{q}} \quad (15)$$

where $\frac{1}{p}+\frac{1}{q}$=1, A=$\{\alpha | 0 \leq \alpha \leq C\mathbf{1}, \mathbf{1}^t Y\alpha = 0\}$, $H_k = Y K_k Y$ and Y is a diagonal matrix with labels on the diagonal. The kernel weights can then be computed as

$$d_k = \frac{1}{2\lambda}\left(\sum_k (\alpha^t H_k \alpha)^q\right)^{\frac{1}{q}-\frac{1}{p}} (\alpha^t H_k \alpha)^{\frac{q}{p}}. \quad (16)$$

Since the dual objective (equation (15)) is differentiable with respect to α, the SMO algorithm can be applied by selecting two variables at a time and optimizing until convergence.

A key challenge in web video retrieval is the large imbalance in the available training data for positive and negative examples. In our experiments on the MED'11 dataset, the ratio of positive to negative examples is about 1:49. Since the MKL optimization function in equation (12) optimizes for accuracy, the solution often converges to the trivial classifier that declares all examples as negative, which has 98% accuracy in our case. To address this, we train models for combining multiple features by performing an extensive grid search over C and p. At each parameter setting, we perform a k-fold validation on the available training data and estimate a threshold that minimizes the Normalized Detection Cost (NDC) score that is defined as

$$f = \min_{Th} \{ w_{MD} P_{MD}(Th) + w_{FA} P_{FA}(Th) \} \quad (17)$$

where, Th is the detection threshold, $P_{MD}(Th)$ and $P_{FA}(Th)$ are the missed detection and false alarm rates at the detection threshold, and w_{MD}, w_{FA} are the relative weights for missed detections and false alarms. This score is often used in machine learning with imbalanced datasets and for the TRECVID MED dataset, different systems are compared with w_{MD}=1.0, w_{FA}=12.49.

8. Score Level Late Fusion

Late fusion of scores from multiple systems has been shown to improve performance in several studies (e.g., [4]). In our work, we combine different combinations of features using MKL, and then combine these different subsystems using score level fusion.

8.1. Score Normalization

One challenge with score fusion is that different systems have different detection thresholds and score profiles. Normalizing scores from different systems has been widely studied, particularly in biometrics [10]. *z-norm* and *sigmoid-norm*, which normalize scores using a Gaussian and sigmoid function respectively, have been the most popular. In recent work, a $w - score$ based on extreme value theory was proposed in [22] for score normalization, which does not make any assumption on the score distribution.
We normalize score p_i' for system i, given the original score p_i and threshold Th_i, by using a *double sigmoid* function as

$$p_i' = \begin{cases} \left(1 + e^{-\frac{2(p_i - Th_i)}{Th_i}}\right)^{-1} & p_i < Th_i \\ \left(1 + e^{-\frac{2(p_i - Th_i)}{1 - Th_i}}\right)^{-1} & p_i \geq Th_i\ . \end{cases} \quad (18)$$

The double sigmoid normalizes the scores around 0.5, which enables score level fusion of multiple systems.

8.2. Bayesian Model Combination

Scores obtained from multiple systems can be combined using Bayesian decision theoretic approach (BayCom) as suggested in [23]. Let M be the number of models to combine, and r_i=(c_i,s_i) denote the output generated by model i. Here, c_i is the classification by system i, and s_i is the confidence score. Let C be the set of unique classes proposed by all systems. Then, the model selects the optimal hypothesis according to

$$c^\star = arg \max_{c \in C} P(c \mid r_1, \ldots, r_M). \quad (19)$$

Using Bayes theorem, we have

$$P(c \mid r_1, \ldots, r_M) = P(c) \frac{P(r_1, \ldots, r_M \mid c)}{P(r_1, \ldots, r_M)}. \quad (20)$$

Assuming that the system hypotheses are independent of each other and ignoring the denominator since it is independent of c, equation (20) becomes

$$P(c) \prod_{i=1}^{M} P(r_i \mid c) = P(c) \prod_{i=1}^{M} P(s_i \mid c_i, c) P(c_i \mid c). \quad (21)$$

The first term $P(c)$ is the prior probability of c being the correct class. The second term $P(s_i \mid c_i, c)$ is the conditional score distribution and can be decomposed in two disjoint events. If $c_i = c$, then this term denotes the probability the system i is correct with a score s_i. If $c_i \neq c$, then this term denotes the probability the system i is incorrect with a hypothesis c_i and a score s_i. Similarly, the third term $P(c_i \mid c)$ can be decomposed in two disjoint events.

The approach followed in the original BayCom formulation [23] assumes that the the conditional probabilities are independent of the particular hypothesis c_i and determined only by whether the hypothesis is correct or incorrect. In this work, we use class specific conditional probabilities. To overcome data sparseness we smooth the conditional probabilities with the class independent probabilities using the Witten-Bell smoothing [33]. For example, we smooth $P(c_i \mid c)$ as

$$P(c_i \mid c) = (1 - \lambda) * P_{ML}(c_i \mid c) + \lambda * P(c_i) \quad (22)$$

where $P_{ML}(c_i|c)$ is the Maximum Likelihood (ML) estimate of the conditional probability and

$$\lambda = \frac{N_{c_i,c}}{N_{c_i,c} + \sum_{c_i} c(c_i, c)} \quad (23)$$

where $N_{c_i,c} = |\{c_i : \#(c_i, c) > 0\}|$. The term $P(c_i)$ in equation 22 denotes the class independent probability.

8.3. Weighted Average Fusion

The second late fusion strategy we consider is weighted average fusion. The BayCom approach, as well as several other fusion techniques (e.g., [4]) use a fixed weight based on the overall performance on validation data, for each system using a parametric model learnt from training data. This approach has two limitations - first, the confidence of a system's detection score for a particular video can be significantly different from the average system performance. For example, on the small set of videos with relevant speech/videotext, the output of these features have high performance. But overall, these systems have low performance since most videos do not have such content. Second, learning the parameters for late fusion requires a large training set. This should be separate from the set used for training the individual systems, to estimate generalizable models. Given the limited amount of training data, it is challenging to obtain suitable partitions.

To address these limitations, we adopt a novel weighted average fusion strategy that assigns video specific weights based on each system's detection threshold. This is based on the intuition that a system has low confidence when its score for a particular video is close to the detection threshold, and high confidence when the scores are significantly different from the threshold. Given the confidence score p_i from system i for a particular video, the weight for that system is computed as

$$w_i = \begin{cases} \frac{Th_i - p_i}{Th_i} & p_i < Th_i \\ \frac{p_i - Th_i}{1 - Th_i} & p_i \geq Th_i \ . \end{cases} \quad (24)$$

This weighting matches our intuition and assigns higher weights to systems whose scores are much higher or lower than the detection threshold Th_i and lower weights to systems with weights closer to the threshold, for each video. Further, if we use normalized scores from equation (18), the system thresholds are normalized to 0.5, and the systems are weighted based on the absolute distance from the threshold. Given these weights, we compute the final score P for a video as

$$P = \frac{\sum_i w_i p_i}{\sum_i w_i}. \quad (25)$$

We also explore generalizations of the weighted average fusion technique using polynomials and L_p norms of the scores, respectively as

$$\langle \sum_i p_i, \sum_i p_i^2, \ldots, \sum_i p_i^k \rangle \text{ and}$$

$$\langle \sum_i p_i, (\sum_i p_i^2)^{1/2}, \ldots, (\sum_i p_i^k)^{1/k} \rangle.$$

These feature vectors are fed to a linear SVM. We observed that the obtained results were comparable to those obtained by weighted average fusion defined above.

9. Experimental Results

We tested our approach on a large, benchmark dataset of ∼45000 videos used in the TRECVID MED 2011 evaluations [19]. It consists of a 13000 video development set containing ∼2000 videos from 15 events of interest, with 100-200 examples per event, and the rest of the videos are from the background class. The evaluation set consists of ∼32000 videos, with ∼100 videos from each event of interest, and the rest from the background class.

9.1. MKL-based Feature Fusion

In the first set of tests, we trained different feature combinations using MKL on a $train$ partition and tested on a dev partition. Overall, using a diverse set of features (3 color features, D-SIFT, STIP, MFCC) produced the best performance. However, even combining somewhat redundant features like SIFT/D-SIFT and STIP/HoGHoF3D, which different only in feature point detection strategy produces consistent gains as illustrated in figure 2. The same pattern generalized to the final 32000 evaluation set as well.

9.2. Impact of High-Level Features

Next, we evaluated the impact of using high level information from object detection and speech. Figure 3, compares the performance of classifiers trained using person/car detector and ASR, with RGB-SIFT, which is the best individual low-level feature. As can be seen, for class 8, which corresponds to the event "Getting a vehicle unstuck" with a large number of vehicles, car detection is the single best feature. Further, ASR has the best performance for 4 of the 15 classes.

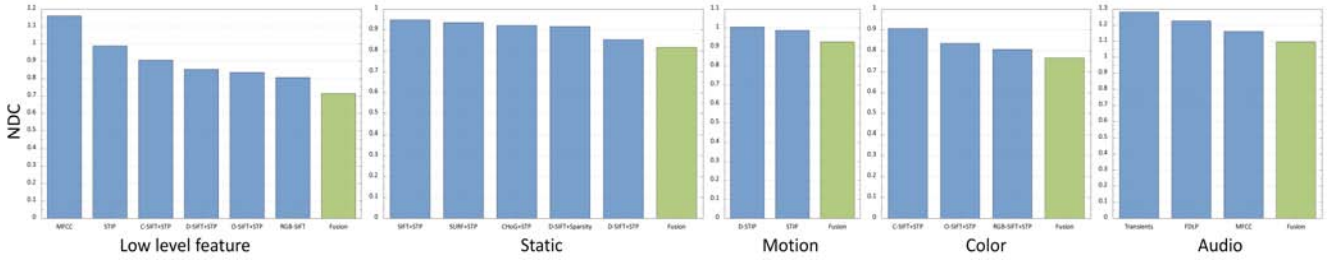

Figure 2. Performance summary of MKL-based low level feature combinations in terms of Average ANDC.

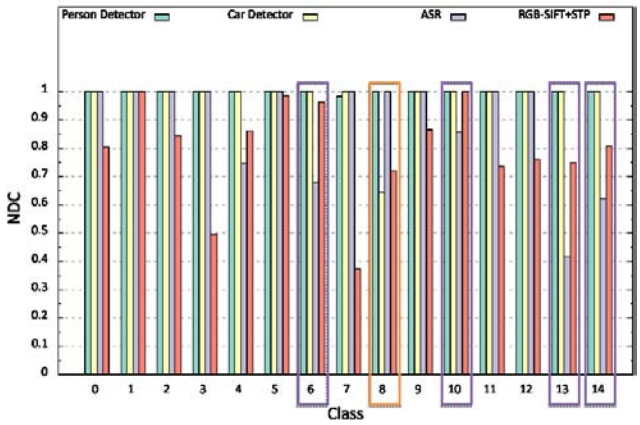

Figure 3. Performance summary of MKL-based high level feature combinations

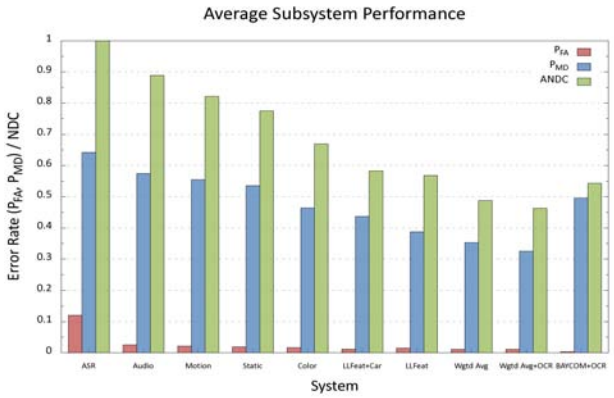

Figure 4. Performance comparison of individual features and three different fusion systems.

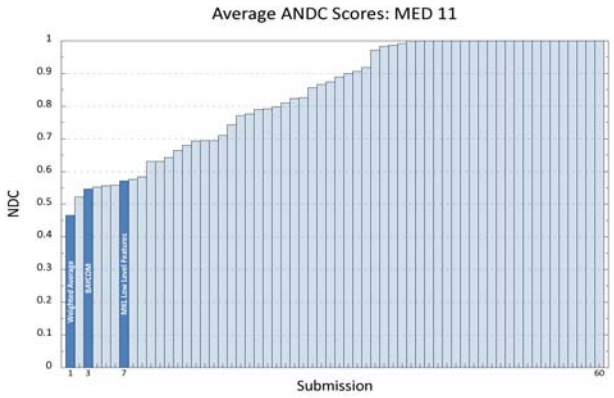

Figure 5. Performance of our MED'11 runs and all 60 official submissions. The vertical axis shows the performance measured by average ANDC scores across the 10 MED'11 events.

9.3. Comparison of Late Fusion Strategies

Next, we combined different sub-systems trained using MKL-based early fusion using different late fusion strategies. Overall, the weighted average fusion approach had the best NDC score. BayCom on the other hand had the lowest false alarm, but higher missed detection rate. Both these late fusion approaches improved over our low-level feature based system (LLFeat). This is illustrated in Figure 4.

Figure 5 summarizes the relative performance of our systems compared to all other systems evaluated in MED'11, in terms of NDC at the detection threshold. Weighted average system has the best NDC score. Further, MKL-based low level feature system has strong performance. This indicates that low-level features have strong stand-alone performance. However, high-level information from ASR, video-text, and object detection produce large gains over the low-level features.

10. Conclusion

To summarize our work, we evaluated a large set of low-level audio and visual features as well as high-level information from object detection, speech and video text OCR. We combined multiple features using a multi-stage feature fusion strategy with feature level early fusion using multiple kernel learning (MKL) and score level fusion using Bayesian model combination (BayCom) and weighted average fusion using video specific weights. We conducted rigorous evaluation of different features and fusion strategies on a large benchmark dataset of ≈45000 unstructured web videos used in the TRECVID MED 2011 evaluations.

Our results indicate that low-level audio and visual features have strong performance and form the core of our system. Combination of multiple, diverse features produce significant improvements over the individual features. More importantly, combining even similar features such as SIFT and SURF, or STIP and HoGHoF3D improves over the individual features. Further, score level fusion of multiple early fusion systems produce additional performance gains. The novel video-specific weighted average fusion strategy we presented outperforms several standard late fusion strategies. High level visual features from object detection are promising but performance gains from them are inconsistent. Speech and videotext OCR provide complementary information and produce large gains over the low-level features. We plan to build on these results by developing more robust high-level visual features to effectively leverage scene, object and action concepts.

11. Acknowledgment

Supported by the Intelligence Advanced Research Projects Activity (IARPA) via the Department of Interior National Business Center contract number D11PC20071. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. 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.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *CVIU*, 110(3):346–359, 2008.

[2] Y. Boureau, F. Bach, Y. Le Cun, and J. Ponce. Learning mid-level features for recognition. In *CVPR*, pages 2559–2566, 2010.

[3] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, R. Grzeszczuk, and B. Girod. Chog: Compressed histogram of gradients a low bitrate feature descriptor. In *CVPR*, pages 2504–2511, 2009.

[4] S.-F. Chang, D. Ellis, W. Jiang, K. Lee, A. Yanagawa, A. C. Loui, and J. Luo. Large-scale multimodal semantic concept detection for consumer video. *ACM SIGMM International Workshop on Multimedia Information Retrieval*, 2007.

[5] C. V. Cotton, D. P. W. Ellis, and A. C. Loui. Soundtrack classification by transient events. In *ICASSP*, pages 473–476, 2011.

[6] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[7] S. Davis and P. Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume 28, pages 357–66, 1980.

[8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. In *JAIR*, volume 32, 2007.

[9] A. Gupta and L. S. Davis. Objects in action: An approach for combining action understanding and object perception. In *CVPR*, 2007.

[10] A. K. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems. *Pattern Recognition*, 38(12):2270–2285, 2005.

[11] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[12] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008.

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages II: 2169–2178, 2006.

[14] L.-J. Li, H. Su, E. P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *NIPS*, 2010.

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

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

[17] P. Motlícek, S. Ganapathy, H. Hermansky, and H. Garudadri. Wideband audio coding based on frequency-domain linear prediction. *EURASIP J. Audio, Speech and Music Processing*, 2010.

[18] B. A. Olshausen and D. J. Fieldt. Sparse coding with an overcomplete basis set: a strategy employed by v1. *Vision Research*, 37:3311–3325, 1997.

[19] 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 mechanisms and metrics. In *Proceedings of TRECVID 2011*. NIST, USA, 2011.

[20] J. C. Platt. *Fast training of support vector machines using sequential minimal optimization*, pages 185–208. MIT Press, 1999.

[21] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[22] W. J. Scheirer, A. Rocha, R. J. Micheals, and T. E. Boult. Robust fusion: Extreme value theory for recognition score normalization. In *ECCV (3)*, pages 481–495, 2010.

[23] A. Shankar. Bayesian model combination (baycom) for improved recognition. *ICASSP*, pages 845–848, 2005.

[24] Y. Song, M. Zhao, J. Yagnik, and X. Wu. Taxonomic classification for web-based videos. In *CVPR*, pages 871–878, 2010.

[25] G. Toderici, H. Aradhye, M. Pasca, L. Sbaiz, and J. Yagnik. Finding meaning on youtube: Tag recommendation and category discovery. In *CVPR*, pages 3447–3454, 2010.

[26] M. M. Ullah, S. N. Parizi, and I. Laptev. Improving bag-of-features action recognition with non-local cues. In *BMVC*, 2010.

[27] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1582–1596, 2010.

[28] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J.-M. Geusebroek. Visfual word ambiguity. *PAMI*, 32(7):1271–1283, 2010.

[29] S. V. N. Vishwanathan, N. T.-A. Z. Sun, and M. Varma. Multiple kernel learning and the smo algorithm. *NIPS*, pages 3311–3325, 2010.

[30] S. Vitaladevuni, P. Natarajan, R. Prasad, and P. Natarajan. Efficient orthogonal matching pursuit using sparse random projections for scene and video classification. In *ICCV*, 2011.

[31] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[32] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li. Youtubecat: Learning to categorize wild web videos. In *CVPR*, pages 879–886, 2010.

[33] I. Witten and T. Bell. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *ieeetit*, 37(4):1085–1094, 1991.

[34] S. Zanetti, L. Zelnik Manor, and P. Perona. A walk through the web's video clips. In *InterNet*, pages 1–8, 2008.