IBM Research and Columbia University TRECVID-2011 Multimedia Event Detection (MED) System

Liangliang Cao[†], Shih-Fu Chang[*], Noel Codella[†], Courtenay Cotton[*], Dan Ellis,[*]
Leiguang Gong[†], Matthew Hill[†], Gang Hua[‡], John Kender[§], Michele Merler[§], Yadong Mu[*],
Apostol Natsev[†], John R. Smith[†]

Abstract

The IBM Research/Columbia team investigated a novel range of low-level and high-level features and their combination for the TRECVID Multimedia Event Detection (MED) task. We submitted four runs exploring various methods of extraction, modeling and fusing of low-level features and hundreds of high-level semantic concepts. Our Run 1 developed event detection models utilizing Support Vector Machines (SVMs) trained from a large number of low-level features and was interesting in establishing the baseline performance for visual features from static video frames. Run 2 trained SVMs from classification scores generated by 780 visual, 113 action and 56 audio high-level semantic classifiers and explored various temporal aggregation techniques. Run 2 was interesting in assessing performance based on different kinds of high-level semantic infomation. Run 3 fused the low- and high-level feature information and was interesting in providing insight into the complementarity of this information for detecting events. Run 4 fused all of these methods and explored a novel Scene Alignment Model (SAM) algorithm that utilized temporal information discretized by scene changes in the video.

1 Introduction

The exploding volumes of image and video content is creating tremendous opportunities for exploiting this information for insights and information. However, the unconstrained nature of "video in the wild" makes it very challenging for automated computer-based analysis. Furthermore, the most interesting content in this video is often complex in nature reflecting a diversity of human behaviors, scenes, activities and events. And the research community is only beginning to tackle the problem of automatically recognizing, representing and searching for this kind of event information in unconstrained video.

The TRECVID Multimedia Event Detection (MED) task was designed to evaluate how well systems can detect events in "video in the wild." The IBM Research/Columbia team addressed this challenge by building on what we were good at from our prior work on video content classification and retrieval. Notable from our previous efforts, we have built powerful capabilities for analysis and semantic modeling of visual content as part of the IBM Multimedia Analysis and Retrieval System (IMARS). We used this as a foundation in terms of providing a starting set of visual feature descriptors and semantic classifiers. Given the challenge of detecting events, which are temporal and multi-modal in nature, we further developed additional spatial-temporal feature descriptors and dynamic motion-based and audio-semantic classifiers. We also explored numerous techniques for fusing this information over multiple scenes and segments of video in order to accurately detect events.

Overall, we submitted four runs for the MED task that explored the following, respectively, (1) low-level signal features, (2) high-level semantic features, (3) two types of fusion of low- and high-level features, (4) alignment aligning models with scenes. We explored several techniques for normalizing and fus-

[*]Columbia University, Dept. of Electrical Engineering
[†]IBM T. J. Watson Research Center
[‡]Stevens Institute of Technology
[§]Columbia University, Dept. of Computer Science

ing this information as well as incorporating temporal aspects for predicting events. We summarize the four runs as well as their key characteristics as follows, note that all of our run names were prefixed by IBM_MED11_MED11TEST_MEDFull_AutoEAG_:

1. **Run 1: c-CU-Fusion-Regression_1**
   Low-level signal features:
   Sparse SIFT + Dense SIFT + Color SIFT + STIP + MFCC features
   SVM with histogram intersection + Chi2 kernels
   Fusion weights based on ridge regression

2. **Run 2: c-Fusion-All-optimized15_1**
   High-level semantic features:
   780 visual + 113 action + 56 audio semantic features
   10 feature normalization and 2 feature aggregation methods
   SVM with RBF, Chi2, and histogram intersection kernels
   Greedy ensemble fusion with forward model selection

3. **Run 3: p-Fusion-All-Baseline-AdHoc_1**
   Weighted Average fusion of Run 1a and Run 2
   Run 1a like Run 1 but using manually-specified weights
   Weights of Run 1a and Run 2 proportional to their training set MAP scores

4. **Run 4: c-Semantic-Fusion-Baseline_1**
   Linear SVM-based fusion of 14 component runs
   14 components runs (8 low-level features + 5 semantic features + Scene Aligned Models)
   Weights learned with linear SVM

Overall, run 3 was our best-performing submission, benefitting from the fusion of high-level semantic features with the lower-level signal features they were based upon in accordance with MAP scores on our training data.

2 System

2.1 Overview

As described above, our system performed extraction of low- and high-level features and investigated multiple techniques for fusing this information to detect events. The overall system and process flow is illsutrated in Figure 1. As depicted, one process flow extracts and models low-level features using various scene descriptors (visual), SIFT/GIST (local visual features), MFCCs (audio), and STIP features (spatio-temporal). Similarly, a high-level feature process flow extracts and models visual, audio and action semantics. These provide the basis for the information used to detect events in the video. Subsequent modeling from this information provides prediction of the events. We explored various techniques for combing these predictions in late fusion to make the overal decision about event detection.

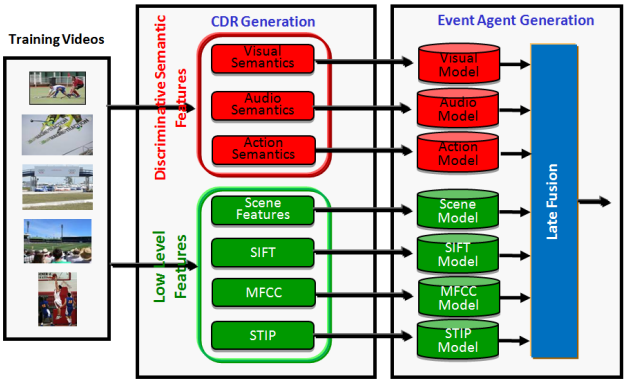

Figure 1: IBM/Columbia mutlimedia event detection system.

2.2 Low-level Feature Extraction

2.2.1 Video processing

Our system builds up semantic model vectors for videos from frame-level features. The first step for us is to decode videos into frames and select those which will be used in subsequent processes.

In the past we have experimented with using uniformly sampled frames along with keyframes (selected based on inter-frame color histogram differences) however this year we used solely uniform sampling. We decode the video clip, and uniformly save one frame every two seconds. These frames are then used to extract static visual descriptors: local (SIFT), GIST, global and Semantic Model Vectors. We chose 0.5 fps as a sampling rate based

on the data set size in order to yield a number of frames that we could process in a reasonable time. The STIP (Sec 2.2.6) features were extracted at the full video frame rate, up to 30fps.

2.2.2 Static Features

We extract over 100 types of static image features from each of the sampled frames. These features capture a wide range of image information including color, texture, edge, local appearances and scene characteristics. We build upon these features to extract the Semantic Model Vectors (Sec 2.3.2).

2.2.3 Local Descriptors

Local descriptors are extracted as SIFT [11] features with Harris Laplace interest point detection for the sampled frames. Each keypoint is described with a 128-dimensional vector containing oriented gradients. We obtain a "visual keyword" dictionary of size 1000 (by running K-means clustering on a random sample of approximately 300K Interest point features, we then represent each frame with a histogram of visual words. For keyframes we used soft assignment following Van Gemert et al. [16] using $\sigma = 90$.
In our combination runs we also included local features computed by Columbia University, which extracted SIFT features from interest points detected with both DoG and Hessian detectors at the sampled frames, employed two 500-d codebooks, and adopted spatial pyramid matching for the full frame + 4 quadrants, obtaining a 5000-D total feature length.

2.2.4 GIST

The GIST descriptor [13] describes the dominant spatial structure of a scene in a low dimensional representation, estimated using spectral and coarsely localized information. We extract a 512 dimensional representation by dividing the image into a 4x4 grid, we also extract histograms of the outputs of steerable filter banks on 8 orientations and 4 scales.

2.2.5 Global Descriptors

In addition to the SIFT bag-of-words and GIST descriptors, we extracted 13 different visual descriptors at 8 granularities and spatial divisions. SVMs are trained on each feature and subsequently linearly combined in an ensemble classifier. We include a summary of the main descriptors and granularities. Details on features and ensemble classifier training can be found in our prior report [1].

- **Color Histogram**: global color distribution represented as a 166-dimensional histogram in HSV color space.
- **Color Correlogram**: global color and structure represented as a 166-dimensional single-banded autocorrelogram in HSV space using 8 radii depths.
- **Color Moments**: localized color extracted from a 5x5 grid and represented by the first 3 moments for each grid region in Lab color space as a normalized 225-dimensional vector.
- **Wavelet Texture**: localized texture extracted from a 3x3 grid and represented by the normalized 108-dimensional vector of the normalized variances in 12 Haar wavelet sub-bands for each grid region.
- **Edge Histogram**: global edge histograms with 8 edge direction bins and 8 edge magnitude bins, based on a Sobel filter (64-dimensional).

Having a large and diverse set of visual descriptors is important for capturing different semantics and dynamics in the scene, as so far no single descriptor can dominate across a large vocabulary of visual concepts and events, and using a collection like this has shown robust performance [1, 15]. The spatial granularities include global, center, cross, grid, horizontal parts, horizontal center, vertical parts and vertical center – each of which is a fixed division of the image frame into square blocks (numbering from 1 up to 25), and then concatenating the descriptor vectors from each block. Such spatial divisions has been repeatedly shown robust performance in image/video retrieval benchmarks such as TRECVID [14].

2.2.6 Spatio-Temporal Features

To capture the spatiotemporal characteristics of the video events at the feature level, we extract both histogram of

oriented gradient and histogram of flow features around the space time interest point (STIP) [8]. The space time interest point is an extension of the Harris interest point detector in 2D images to the spatiotemporal 3D volume video data. It is extracted from a set of multiple combination of spatial and temporal scales. Once an space time interest point is detected, a 3D volume is formed around it based on the corresponding spatial and temporal scales. The 3D volume is partitioned into a grid of $3 \times 3 \times 2$ spatiotemporal blocks in x, y, and t axis, respectively. For each block, a 4 bin histogram of gradient and a 5 bin of histogram of flow are extracted. They are aggregated over all blocks to form a $3 \times 3 \times 2 \times 4 = 72$ dimensional histogram of gradient feature, and a $3 \times 3 \times 2 \times 5 = 90$ dimensional histogram of flow feature. These two feature vectors are concatenated to form a 162 dimensional feature vector for each STIP we detected from the video to form a bag-of-feature representation for each video clip.

2.2.7 Audio Features

For audio features, we conducted preliminary experiments using conventional MFCC features as well more experimental features based on extracting transient sound events (short duration energy concentrations) [2], and modeling the perceptually-salient textural aspects of the sound (based on recent work in the resynthesis of textures) [3]. These experiments, however, showed that MFCC remained the best single feature to use, so our current audio semantic system uses only these features. Specifically, we calculate 20 dimensional MFCCs over 32 ms windows with 16 ms hop. We calculate deltas and double-deltas over 10-frame windows for a 60 dimensional feature vector. We calculate the mean and full 60×60 covariance matrix over each video to give an 1890-dimensional representation of the video. We experimented with various subsets of these dimensions, and found that using only the 20 direct feature means and the 210 unique values of the covariance of the direct features performed nearly as well as larger representations, but at much less computational cost. The semantic results below are based on these 230-dimensional feature vectors. Each feature dimension was normalized to have zero mean and unit variance over the entire dataset before any further processing; these normalization constants were recorded and applied to the test data as well.

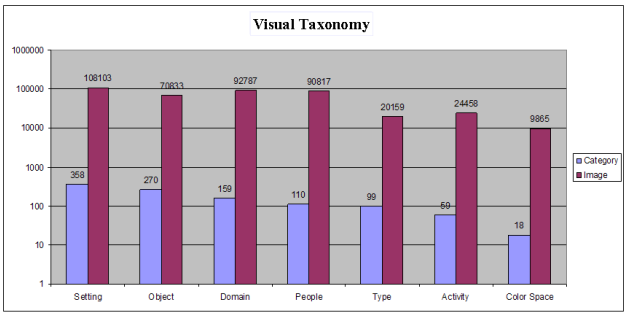

Figure 2: Visual taxonomy: distribution of categries and images across top facets

2.3 Semantic Modeling

While the traditional methods of modeling video events has been to train models directly from low level features, we believe event detection may be made more robust, simpler, and more space efficient, if the videos were described by their content in terms of higher level semantics. The bridge between low level features and high level events is referred to as the "Semantic Gap". We propose a technique that fills this gap with an additional semantic layer, connecting low level features to video events through a hierarchy of the visual, audio, and action semantic content of the video. The multimedia ontological modeling system developed for MED11 event detection consist of three taxonomies: image/video taxonomy, audio taxonomy and dynamic action taxonomy. Both audio and dynamic action taxonomies include 56 and 134 categories respectively. Their specific applications and performance will be discussed separately in subsequent sections, while in this section we will focus on the discussions of visual taxonomy.

2.3.1 Semantic Taxonomy

For the MED11 event detection task, our team utilized an improved taxonomy of visual concepts/categories based on the IBM Multimedia Analysis and Retrieval System (IMARS) taxonomy [4]. About 400 new categories has been structured into the IMARS taxonomy including many event related (directly or indirectly) concepts (e.g. appliance, fishing gear, toolbox, etc.). Various number of image/video-frame examples are associated with each

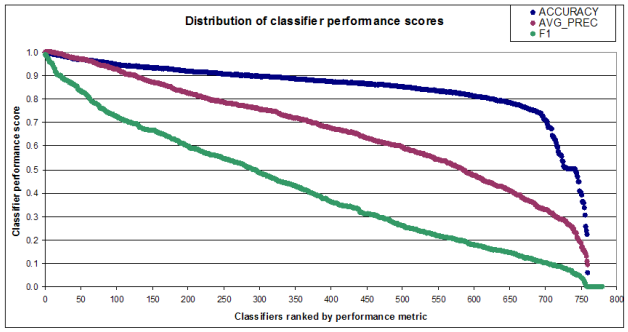

Figure 3: Semantic modeling performance: validation scores for 780 categories

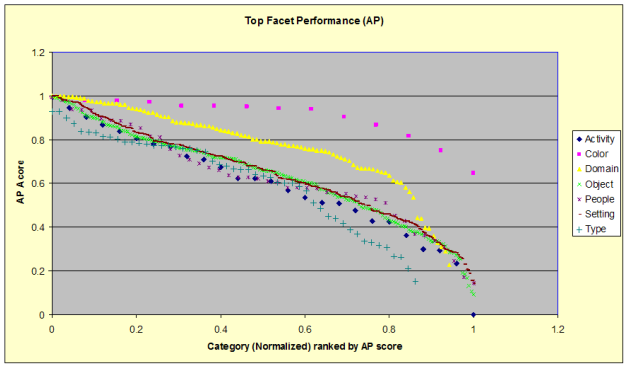

Figure 4: Facets performance: average precision (AP) scores for 780 categories

leaf node (concept) for training. The current version of the taxonomy has total 1000 categories and 500K positive training examples. After filtering out categories with small number of training examples, 780 categories (semantic features) were used in the runs for MED11TEST submission.

The visual taxonomy has been designed and built with four conceptual constructs: entity (node), facet (node), is-a (link) and facet-of (link). Adopting the facet node type and "facet-of" link type allows greater flexibility in modeling mutually non-exclusive concepts, which represent different view perspective of a same entity (e.g. people - number of people, age of people). More specifically, sibling concepts (nodes) in the taxonomy tree that are not mutually exclusive are denoted as facet nodes, while mutually exclusive sibling concepts as entity nodes. A relationship of "facet-of" links two facet nodes or an entity node to a facet node, while "is-a" relationship is used to link two entity nodes or a facet node to an entity node. By inferencing the structure and semantic relationships, the taxonomy system can perform efficient labeling of training images by associating images with the each entity node in the hierarchy, and allocates negative training examples accordingly with the recognition of exclusiveness of entity nodes and non-exclusiveness of facet nodes.

Currently we have seven top facets (setting, domain, object, people, activity, type, and color). Figure 3 shows the top level composition of the visual taxonomy in terms of top facets and the distributions of categories and image examples among them. Various experiments were performed to examine the performance characteristics of the visual taxonomy for concept modeling. Figure 3 plots the three validation score distributions for the 780 categories in the taxonomy. As it shows, over 64% of them received 0.6 or higher average precision scores, and about 75% received 0.5 or better scores.

Our experiments show that the modeling performance varies among the facets and categories, which allow us to identify the weak spots in our taxonomy. We are currently conducting a series of evaluation and revision to further improve the visual taxonomy. A parallel effort is being made to develop a semiautomatic tool to assist and reduce the cycle in the development of the taxonomy.

2.3.2 Visual Semantic Modeling

Once features have been extracted from the taxonomy and organized into relevant positive and negative categories for each concept, training data is split into two partitions: Learning and Validation. The proportions of this split are typically 67% and 33%, respectively. Visual Semantic Modeling then occurs in two phases, and is based on the Robust Subspace Bagging (RB-SBag) method [17]. In the first phase, models are trained for multiple "bags" of data in the Learning partition. A "bag" is a sub-sampling of features and training data for a given semantic concept (image granularity, feature type, randomly selected subset of positive samples, and randomly selected subset negative samples). For each bag, a single SVM model, referred to as a "Unit Model," is generated. In training, the best performing SVM model parameters out of 29 for each is

selected. In summary, for each concept, 7 image granularities were chosen, 18 image features were extracted, and 5 bags were sampled. The list of granularities is as follows: global, center, cross, horizontal center, horizontal parts, layout, and vertical center. The list of features extracted is as follows described in Section 2.2.2.

Once all the unit models for a concept are trained, they are combined in the second phase to form a "Fusion Model." When the fusion model is used to score images, the score is determined by a weighted sum of the outputs of all the unit models that compose the fusion model. The weights are determined by the Average Precision (AP) scores of each unit model evaluated against the Validation partition of the training data.

2.3.3 Audio Semantic Modeling

We developed a set of 55 semantic audio models based on previously-available manually labeled video. 25 classifiers came from an earlier project on classifying consumer video that involved labeling 1873 unedited consumer videos downloaded from YouTube with 25 consumer-relevant labels such as "Crowd", "Animal", "Museum" etc. [9] 20 more classifiers came from a similar but larger dataset of 9413 videos released last year as the Columbia Consumer Video (CCV) dataset [5], and included some overlap in labels with the first 25, although some concepts were eliminated, and others were expanded (e.g., "Sports" became "Soccer", "Basketball", and "Baseball"). The final 10 concepts were based on an in-house labeling performed as part of MED2010 in which 6626 10-second segments cut from the MED2010 development data were annotated with 10 audio-related labels including "outdoor - rural", "outdoor - urban", "speech", and "cheer". Each concept defined a one-versus-all Support Vector Machine classifier based on a Gaussian kernel; the C and γ parameters were chosen by grid search over a portion of the training data.

The within-set average precision of all classifiers was tested for every label within that set on a 5-way split (40% used for train, 20% for tuning, and 40% for test, with each item appearing in the test set for two folds). The results are shown in figure 5. Mean APs vary from 0.3 to 0.48, although the results are hard to compare since they depend on the difficulty of the particular labels and the priors in the test set.

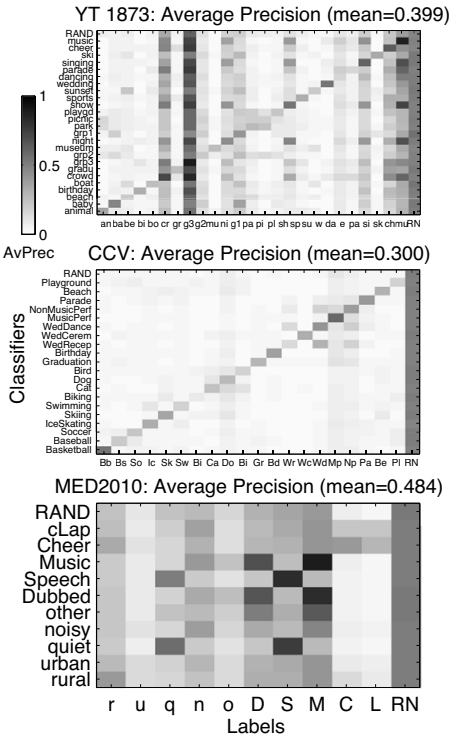

Figure 5: Average Precision results for each of the 3 subgroups of semantic audio classifiers.

Each of the 55 classifiers was run on each MED2011 video soundtrack to create a 55-dimensional vector composed of the distance-to-margin from each classifier. These semantic audio feature vectors were then the basis of further classification. Figure 6 shows the Average Precision of SVM classifiers trained for each of the 15 MED2011 events. (These results are again on a five-fold cross-validation, using only the 2062 event example videos as "background"). The plot also compares against similar classifiers trained directly on raw MFCC summary statistics instead of the semantic classifier vectors. In this case, there is little or no gain obtained by introducing the semantic audio classifier layer. Notice, however, that for some events such as E004 Wedding Ceremony, E006 Birthday Party, and E012 Parade, the classifiers based on semantic features perform substantially better than their raw MFCC counterparts. These partic-

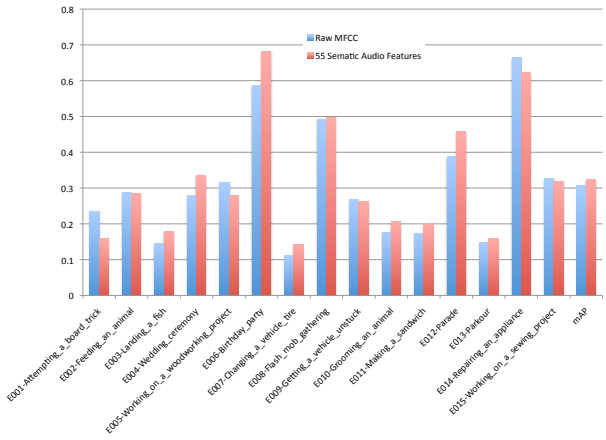

Figure 6: Average Precision results for classifiers trained and tested the MED2011 event examples, comparing classifiers based on the 55 semantic audio classifier outputs and those based directly on the raw MFCC summary statistics.

ular categories are aligned closely with labels present in the underlying classifier set. One conclusion is that, given a larger and more comprehensive set of semantic audio classifiers, such gains could be obtained for a wider range of events.

2.3.4 Dynamic Action Modeling

The video events we investigate are generally composed of a combination of people and objects interacting in a scene through actions. In this sense, the dynamic information is as important as the audio and static visual one, and in certain cases represents the major cue to distinguish among events. For example the events *Parade* and *Flash mob gathering* present quite similar visual appearance (people on a street), but the movements of the group of people is what allows to distinguish between the two.

Naturally low-level motion descriptors can encompass such information to a certain degree, however also in the dynamic domain there is a need to fill the semantic gap. To this end we adopt the Dynamic Action Model Vector, an intermediate semantic representation which is similar is spirit to the Visual Semantic Modeling, but explores the complementary context of action semantics.

The generation process for the Dynamic Action Model Vector is illustrated in Figure 7. Starting from a video clip, we extract histogram of gradient (HOG) and histogram of flow (HOF) features around the STIP [8] spatio-temporal interest points, and concatenate them into a single 162 dimensional representation (72-d HOG plus 90-d HOG). Following the popular bag of words framework, we extract a histogram of codewords occurrences for each video clip, using a 5000-words codebook obtained through K-means clustering from the MED11 development videos. In order to build such histograms, we adopt the popular and effective soft assignment strategy proposed by Jiang et al. [6]. We build action models by training one versus all classifiers for the action categories found in three state of the art datasets:

- UCF50[1]: an extension of the UCF Youtube [10] dataset, consisting of 6,681 videos obtained from Youtube and personal videos, representing 50 actions mainly related to sports
- HMDB[2] [7]: a collection of 6,766 videos from various internet sources, with 51 actions mostly focusing on human movements (i.e. kiss, hug, situp, drink)
- Hollywood2[3] [12]: 12 actions from 1,707 videos clips taken from movies

Action Models are trained for each dataset separately, and each model is an ensemble SVM with RBF kernel. The learning procedure follows that of the Visual Semantic Models (bagging, parameter selection through cross-validation, forward model selection on a held-out set), with the objective of maximizing performance in terms of Average Precision. In Table 1 are reported the training Mean Average Precision rates across the action categories on the three sets. Consistently with the results of the literature, the models trained on the HMDB dataset are significantly less accurate then the ones representing the UCF50 classes.

Finally, we train Event classifiers for the MED11 categories by learning an SVM with RBF kernel on top of the Dynamic Action Model Vectors(DAMV), which result from the concatenation of the action models classifiers responses to the MED11 video clips.

[1] http://server.cs.ucf.edu/~vision/data.html
[2] http://serre-lab.clps.brown.edu/resources/HMDB/index.htm
[3] http://www.irisa.fr/vista/actions/hollywood2

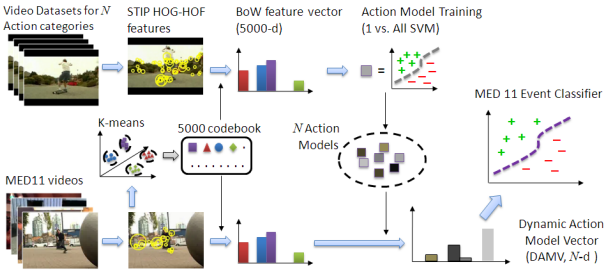

Figure 7: Dynamic Action Model Vector generation pipeline.

Dataset	#Actions	#Vids.	Secs.	MAP
UCF50	50	133	7.44	0.651
HMDB	51	133	2.60	0.311
Hollywood2	12	156	14.8	0.389

Table 1: Statistics of the three action datasets used to train the Dynamic Action Models: number of action classes, average number of clips per class, average clip duration (in seconds) and model training Mean Average Precision.

In order to estimate the impact of the number of action classes applied to the Event detection domain, we tested the contribution of different sets of action models in the following framework. Performance was evaluated in terms of Mean Average Precision (MAP) on the 15 events of the MED11 data, on a training/test split from the Event Kit and DEV-T sets with 7K videos used for training and 5K videos for testing. We registered a significant improvement from using only the 50 Models coming from the UCF50 dataset (MAP = 0.074) to employing the full set of 113 models (MAP = 0.108), which is the configuration adopted in the official submitted runs. Details of the comparison for each Event category are reported in Figure 8. Event modeling clearly favors a larger number of actions in the model vector, therefore suggesting to increase the number of action classes to model. Which or how many still remains an open question that we intend to investigate in the next iterations of the program.

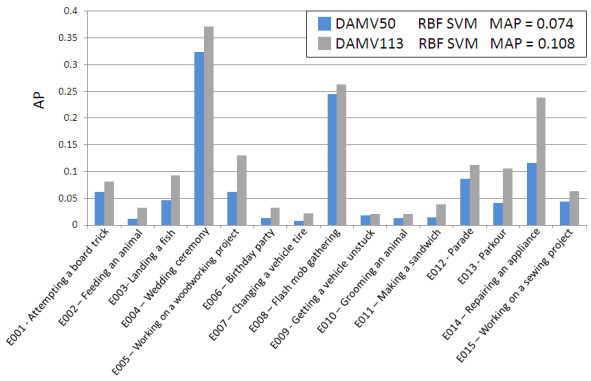

Figure 8: MED11 Event modeling on a train/test split on 12K video clips from the Event Kit and DEV-T sets. Comparison between Dynamic Action Model Vectors using 50 or 113 action models.

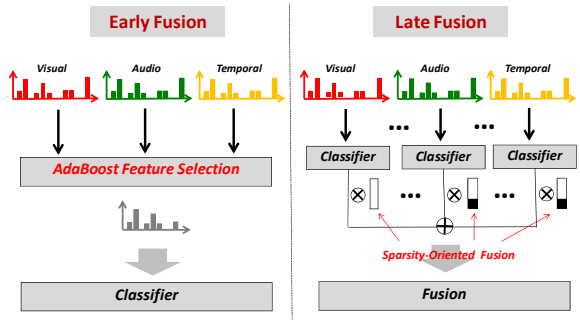

Figure 9: Multimodal models via early fusion (AdaBoost) and late fusion (Sparse optimization).

2.4 Event Modeling

2.4.1 Multi-modal Models

Detecting complex multimedia events is typically beyond the discrimination ability of single modality (*e.g.*, visual, audio, or spatial-temporal feature). Therefore, it motivates the research on multi-modality fusion. Generally, fusion methods can be categorized according to the stage that it takes place in the event detection pipeline, including both early fusion and late fusion. The input of late fusion is the decision scores of each pre-trained components.

As shown in Figure 9, we attack the fusion problem by

Fusion Runs	Components	Performance	Sparsity
AdaBoost	6	.3850 / .2781	0.0%
Uniform	6	.3743 / .2676	0.0%
Ad Hoc	6	.3847 / .2719	0.0%
Ridge Reg.	9	.4032 / .2786	0.0%
Ridge Reg.	15	.4112 / .2838	0.0%
Ridge Reg.	24	.4159 / .2799	0.0%
Lasso	9	.4025 / .2789	43.7%
Lasso	15	.4132 / .2833	48.4%
Lasso	24	.4113 / .2792	55.8%
Tree Lasso	24	.4038 / .2781	62.5%

Table 2: Late fusion performances on the IBM internal test set and the final evaluation set (column three). The performances are reported in terms of mean-average-precision (MAP). In the fourth column, "Sparsity" denotes the percentage of zero coefficients. See text for detailed explanation.

Concept	Mean AP for E001-E015
Parade	0.0419
Protest	0.0406
Team Photo	0.0404
Big Group	0.0403
Bicycling	0.0375
Crowd	0.0373

Table 3: Top 6 relevant static visual semantics models to the 15 MED'11 events.

1) compact early-stage feature fusion via AdaBoost, and 2) late-stage multi-modal fusion via sparse models.

Most of traditional early fusion methods suffer from the explosion of feature dimension when concatenating heterogeneous sources. A significant level of redundancy can be discovered and removed by exploiting machine learning techniques. We perform such a study via AdaBoost. It was used to sequentially choose a subset of most discriminative feature dimensions from the concatenated feature pool. Our preliminary study using the internal evaluation confirms the promise of such early fusion approach - a compact fused feature (1500 dimensions) still outperforms the original concatenated feature of a much higher dimension (14,000 D).

We also systematically investigate different strategies of late fusion. Table 2 presents the comparison between six strategies, where "AdaBoost" and "Uniform" denote using fusion weights learnt during the AdaBoost procedure and uniform weights across components respectively. Our evaluation over the NIST evaluation data set resulted in the following observations: 1) more features cannot guarantee better performance due to the risk of over-fitting. An example is that ridge regression with 15 components outruns that with 24 components in the evaluation set. It is found that less-reliable components (those using LBP, P-HOG and GIST features) possibly hurt the accuracy after fusion, and 2) we incorporate sparse models that encourages a large proportion of components to have zero coefficients. Our empirical study confirmed both the robustness and compactness of the sparse method. Note the ridge regression method with 15 components is the fusion method used in our official submission Run 1.

2.4.2 Semantic Feature Selection

Our semantic model features are extracted from 780 static visual semantic models, such as parade, protest, team photo, big group (8 or more), bicycling, and performance, etc.; 113 dynamic action semantic models such as basketball, benchpress, fencing, golfswing, etc.; and 56 audio semantic models such as rural, urban, noisy, speech, music, etc.. A natural question to ask is *how many semantic models we will need to effectively model the events?*

Our observation is that the different semantic models are not equally relevant to the visual events of interest. Table 3 shows some of the top relevant static visual semantics in descendant order with respect to the the 15 target events in MED'11. In answering this question, we combined the top k, e.g., $k = 5, 10, 15, 20, \ldots, 780$, static visual semantics model vectors with all dynamic action and audio semantic, and we conducted event modeling with all the different combined sets. The evaluation results on our internal IBM Test datasets are shown in Figure 10. As we can clearly observe, the event recognition accuracy saturates at around top 200 static visual semantic models. This implies a big potential efficiency improvement in testing time as we do not actually need to run the other 580 static visual semantic models.

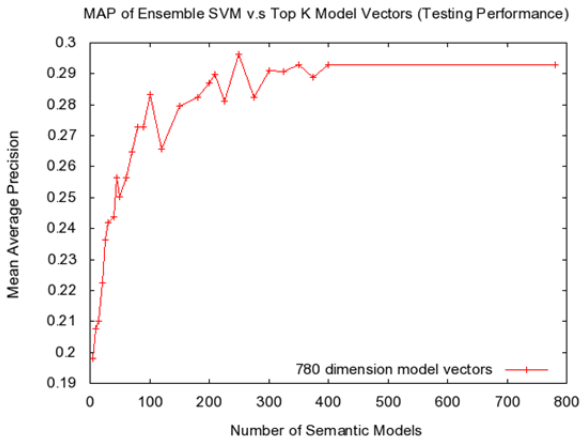

Figure 10: Average precision for event recognition with different number of static visual semantic model vectors.

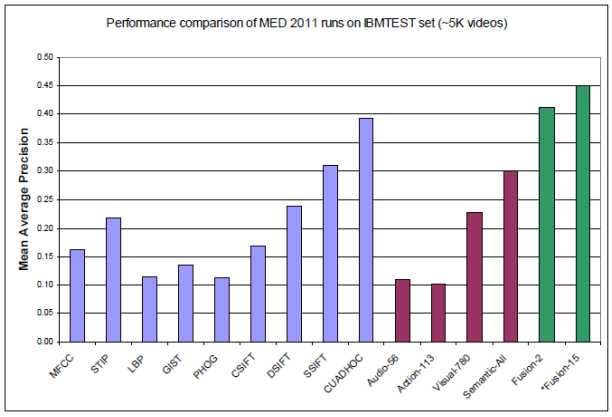

Figure 11: Mean average precision of event modeling with different features with different kernel SVMs

2.4.3 Event Models

When used for event modeling, different features, either low level features such SIFT, STIP, or MFCC, or mid-level features such as those semantic model vector features, may need to use different modeling techniques. In our experiments, we have examined different techniques for event modeling. In particular, we tested *linear regression* and *support vector machines* with different kernel functions for event modeling. The reason we tested linear regression for event modeling is to understand how much gain we can obtain through non-linear modeling such as kernel SVMs. The specific kernel functions we examined include RBF kernel, Chi-square kernel, and histogram intersection kernel. As a reference, linear SVM is also studied. Figure 11 presented the mean average precision of event modeling with different features (as well as the fusion results) on the internal IBM test data set. Our experiments reveal that most of the features work the best with Chi-square kernels in the end for event modeling.

2.4.4 Feature Aggregation

A video always conveys rich information including both temporal and spatial features. How to capture such rich information is a challenging task. A computationally efficient approach is to employ "bag-of-words" model, which quantizes local patches into histogram indices, and concatenate all the indices into a histogram representation. The limitation of bag-of-words model is two-fold: First, a lot of information is lost during the quantization process. Second, both the spatial and temporal information are neglected in the histogram representation. Another approach is to extract features for each frame, and then compute the average or maximum vector of these frame-level features. The mean or maximum vector, however, may not work well when there is large diversity in the video contents.

Our work is partially motivated by the recent process in image classification field. Lazebnik et al. developed spatial pyramid matching model to capture the spatial layout of image features. It is attempting to generalize the pyramid matching idea to the temporal domain. However, after a close examination of video features, we can easily see the flaw of temporal matching. The objects in image often appear in organized locations, e.g., subjects of interests are usually in the middle of a photograph, faces are usually up-right, heaven and cloud are usually in the upper position, and so on. On the contrary, there is no such regularity in temporal domain. A video event can either happen at the beginning part of the video or the second half of the video, or it can last throughout the whole sequence. A video can be of different length and connecting multiple sub-events. A temporal pyramid does not work

for video classification task.

Our idea is also motivated by the simple observation that a video clip is composed with shots of different scenes. Here a "scene" means the environmental description of video frames, including not only the simplest indoor and outdoor categories, but also characteristic environments which differ significantly with each other. Psychology studies show that human vision can easily distinguish different scenes and the results of scene recognition can be helpful for image understanding. We will discuss how to generalize these psychological theories to practical computer vision algorithms.

Our new model aims to aggregate frame-level representation into video-level representation according to different scenes. Intuitively, scene aligned model includes two key components: First, a complex video clip could be partitioned into simpler groups, and each group shares homogeneous scene appearances. Second, the task of video classification is carried out subject to the aligned scenes. The fundamental assumption is to reduce diversity of environments by aligning video frames with different scenes, so that we name this new model as *Scene Aligned Model* (SAM).

2.4.5 Fusion Modeling

Since different event models are modeling the events from different perspective. Therefore, fusing the different event models together would usually boost the recognition accuracy (i.e., late fusion). In our system, we have examined different options for fusing the different event models. We discuss each of them below:

1. **Weighted Average**: We combine the predication scores from different event models by weighted average. The weights could be uniform, manually tweaked, or proportional to their individual prediction average precision on the target events.

2. **Greedy Ensemble Fusion**: We greedily obtain the best k fusing component (fused with average) by examine the best event model to be added to the best $k-1$ fusing models obtained before. Obviously, we will start with the best performed event model at the beginning.

3. **AdaBoost**: We run AdaBoost to additively select the fusing event models and to simultaneously learn the weights of each fusing component.

4. **Linear regression**: We use linear regression to learn the weights of the different fusing event models. Additional regularization terms such as L2 penalty (ridge regression) and L1 penalty (Lasso regression.

5. **Linear SVM**: We simply learn a linear SVM to obtain the weights for each fusing event model for predicting the target event.

We choose the fusion method based on the experimental results in our internal IBM Test set, which can be regarded as the validation dataset for the DEV-O set of the MED'11 task. The last two bar chart in Figure 11 presented some of the fusion results on the internal IBM Test set.

2.5 Score Calibration

To facilitate the selection of the optimal threshold for event recognition, the prediction scores from the event models need to be normalized. In our experiments, we normalize the prediction scores from the event models using a Sigmoid function, i.e.,

$$p(s) = \frac{1}{1 + e^{-as}} \tag{1}$$

where s is score from the event model, $p(s)$ is the normalized score between 0 and 1, a is a scaling factor learned from the collection statistics on the internal IBM Test dataset. To pick up the optimal threshold for event prediction, we first obtain the ROC curve of the event recognition results on the internal IBM test dataset. Then we pick up the threshold corresponding to the operating point with the minimum NDC score on the ROC curve. The final results we obtained from NIST on the DEV-O dataset indicated that our threshold generalize well on the DEV-O dataset.

2.6 Scaling

We used computing clusters for two aspects of our system: learning semantic model classifiers from a training set and applying those models to evaluation sets to produce semantic model vectors. We also call this process of

applying the models "scoring". We made use of two platforms: Apache Hadoop, and IBM InfoSphere Streams, which are both part of IBM BigInsights[4].

2.6.1 Learning Visual Semantic Models (Noel)

The IBM Multimedia Analysis and Retrieval System (IMARS) was used for semantic model training and classification of satellite images. IMARS is configured on a dual-rack Apache Hadoop 0.20.2 system, with 224 CPU cores, 448 GB of total RAM, and 14 TB of HDFS storage. The Hadoop job structure is organized into two phases: a Map step, and a Reduce step. Parallelization occurs across Unit Models: each Unit Model is assigned to one Map task. Once all the Mappers have finished training Unit Models from the Learning data partition and scoring them against the Validation data partition, the models are passed to Reducers in the second stage, keyed by their corresponding semantic. Late-fusion occurs at this step, weighted by the unit model's individual validation scores, to generate Fusion Models. This architecture allows arbitrary scalability in training both semantic classifiers and event models: no algorithmic changes are necessary to scale.

2.6.2 Scoring Visual Semantic Models (Matt)

The process of applying the previously learned models to a test set of frames is computationally intensive. The memory requirements are not exceptionally large, but with 780 semantic models, scoring takes approximately 100 times longer than feature extraction. Furthermore, some of the 780 models had many more support vectors in total than others, which was reflected linearly in the relative amount of time required to evaluate them, as noted in Yan et al.[17]

InfoSphere Streams is a scalable IBM software platform which fits video processing tasks well. It is designed for data in motion, such as frame by frame processing, and defines primitive stream operators for common uses that make it easy to filter and process streaming data in a cluster. We had temporary access to a cluster of 50 nodes containing a total of 800 virtual CPU cores. The nodes each had 8 physical cores, employing hyper-threading to emulate 16 cores. Each node had 16 GB of RAM and access to shared disk storage where the learned models were stored, which were about 80GB on disk.

In order to balance the processing time, we used a bin-packing algorithm to allocate groups of the 780 models into groups that required about 750 MB of RAM each. This resulted in 99 model groups, which contained from 3 to about 100 models each. Then we assigned 16 groups to each node, 1 per virtual core, taking 12 GB of that nodes' RAM when loaded. This meant that we could load all 780 models onto 99 cores. With 800 cores, we replicated this 8 times. Then we split the input image frames into 8 distinct sets and ran everything in parallel, scoring 1.8 million frames in about 42 hours. This amounts to scoring 780 complex semantic models at nearly 12 frames per second. We found that the Streams platform added minimal overhead to the processing.

3 Experimental Results

In our official MED'11 run submissions, all our four runs indeed are the fusion results a subset of event models we explored, we list some detailed information of each of these four runs, along with the fusion methods we adopted for each run:

Run 1: Low-level signal features: In this run, we directly model the events from low level signal features such as sparse SIFT, dense SIFT, color SIFT, STIP, and MFCC. For event modeling we utilized SVM with histogram intersection kernel and Chi-square Kernel. The fusion is performed by linear ridge regression.

Run 2: High-level semantic features: In this run, we model the events from low from the semantic model vectors, including the 780 static visual semantics, the 113 dynamic action semantics, and the 56 audio semantic features. We used 10 different normalization and 2 different feature aggregation methods. We learn SVMs with different kernels including RBF, Chi-square, and histogram intersection, from each different normalized and aggregated semantic features. All these models are fused together by greedy ensemble fusion with forward model selection.

Run 3: Weighted Average Fusion of Run 1a and Run 2: our Run 1a is similar to Run 1 but with manually tweaked weights for each of the fusion models. We fuse Run 1a and Run 2 by weighted average where the weights are

[4]http://www.ibm.com/software/data/infosphere/biginsights

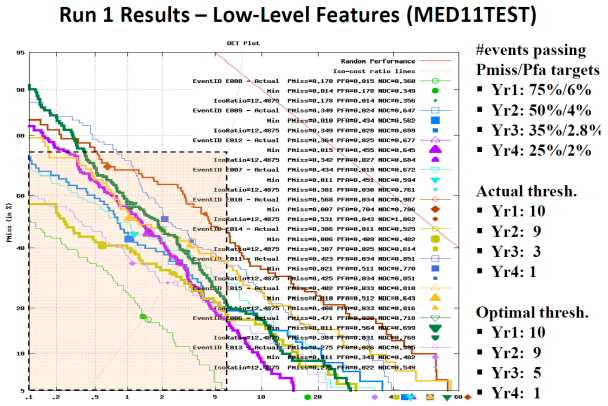

Figure 12: The results of Run 1.

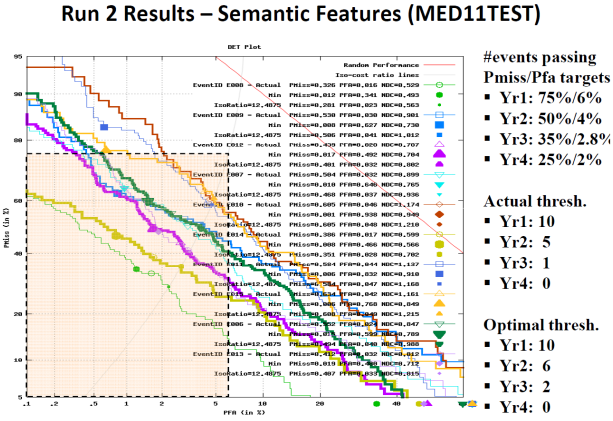

Figure 13: The results of Run 2.

proportional to their predication Average Precision scores on our IBM internal test dataset.

Run 4: Linear SVM-based fusion of 14 component runs: We fuse 14 component runs, including 8 low-level feature runs, 5 semantic feature runs, and the scene alignment model run using linear SVM.

The results of the four runs on the DEV-O dataset are shown in Figure

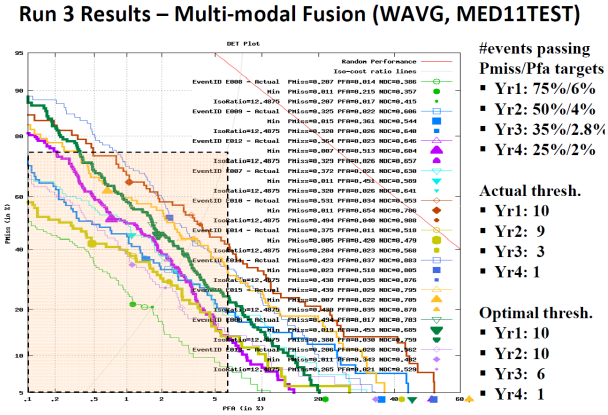

Figure 14: The results of Run 3.

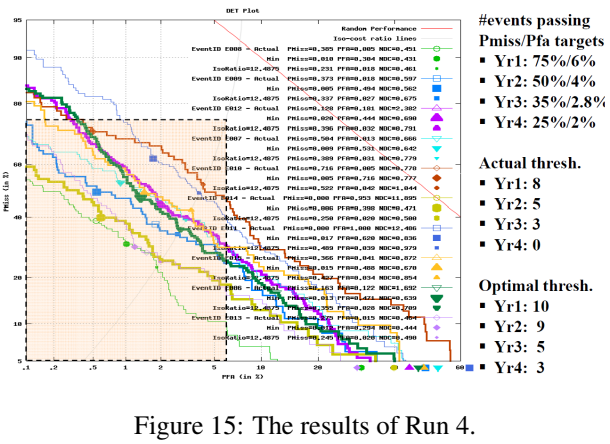

Figure 15: The results of Run 4.

4 Conclusion

We investigated novel techniques for multimedia event detection that utilize low- and high-level information. Overall, we obtained good results on the TRECVID MED task. We observed that low-level visual feature information on its own provides a good baseline for predicting the events in the evaluation. However, we also observe that high-level semantic information and the ability to detect semantic concepts provides complementary information that further improves performance. We expect as we continue to investigate multimedia event detection that further

development of the semantic classification capability is a key direction for improving performance.

References

[1] Murray Campbell, Alexander Haubold, Ming Liu, Apostol Natsev, John R. Smith, Jelena Tesic, Lexing Xie, Rong Yan, and Jun Yang. Ibm research trecvid-2007 video retrieval system. *Proc. NIST TRECVID Workshop*, 2007.

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

[3] D. P. W. Ellis, X. Zheng, and J. H. McDermott. Classifying soundtracks with audio texture features. In *Proc. ICASSP*, pages 5880–5883, Prague, May 2011.

[4] A. Haubold and A. Natsev. Web-based information content and its application to concept-based video retrieval. In *ACM International Conference on Image and Video Retrieval (ACM CIVR)*, 2008.

[5] Y.-G. Jiang, G. Ye, S.-F. Chang, D. P. W. Ellis, and A. C. Loui. Consumer video understanding: A benchmark database and an evaluation of human and machine performance. In *Proc. ICMR*, page article #29, Trento, Apr 2011.

[6] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *ACM International Conference on Image and Video Retrieval (CIVR)*, 2007.

[7] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *International Conference on Computer Vision (ICCV)*, 2011.

[8] Ivan Laptev. On space-time interest points. *Intl Jnl of Computer Vision*, 64(2):107–123, 2005.

[9] Keansub Lee and D. P. W. Ellis. Audio-based semantic concept classification for consumer video. *IEEE Tr. Audio, Speech, Lang. Proc.*, 18(6):1406–1416, Aug 2010.

[10] Jingen Liu, Jiebo Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1996 –2003, 2009.

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

[12] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in Context. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2929–2936, 2009.

[13] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, 2001.

[14] Alan F. Smeaton, Paul Over, and Wessel Kraaij. High level feature detection from video in trecvid: a 5-year retrospective of achievements. *Multimedia Content Analysis*, pages 151–174, 2009.

[15] 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*, in press, 2010.

[16] Jan C. van Gemert, Cees G. M. Snoek, Cor J. Veenman, Arnold W. M. Smeulders, and Jan-Mark Geusebroek. Comparing compact codebooks for visual categorization. *Computer Vision and Image Understanding*, 2010. In press.

[17] R. Yan, M. Fleury, M. Merler, A. Natsev, and J. R. Smith. Large-scale multimedia semantic concept modeling using robust subspace bagging and mapreduce. In *ACM Multimedia Workshop on Large-Scale Multimedia Retrieval and Mining (LS-MMRM)*, Oct. 2009.