Metadata-weighted Score Fusion for Multimedia Event Detection

Scott McCloskey
Honeywell ACS Labs
1985 Douglas Drive North
Minneapolis, Minnesota, USA
scott.mccloskey@honeywell.com

Jingchen Liu
Laboratory for Perception, Action and Cognition
Department of Computer Science and Engineering
The Pennsylvania State University
jingchen@cse.psu.edu

***Abstract*—We address the problem of multimedia event detection from videos captured 'in the wild,' in particular the fusion of cues from multiple aspects of the video's content: detected objects, observed motion, audio signatures, etc. We employ score fusion, also known as late fusion, and propose a method that learns local weightings of the various base classifier scores which respect the performance differences arising from the video quality. Classifiers working with visual texture features, for instance, are given reduced weight when applied to subsets of the video corpus with high compression, and the weights associated with the other classifiers are adjusted to reflect this lack of confidence. We present a method to automatically partition the video corpus into relevant subsets, and to learn local weightings which optimally fuse scores on a particular subset. Improvements in event detection performance are demonstrated on the TRECVid Multimedia Event Detection (MED) MEDTest dataset, and comparisons are provided to several other score fusion methods.**

I. Introduction

Detecting and recognizing activities from unconstrained videos captured 'in the wild' is a fundamental problem in modern computer vision, and is often considered 'vision complete' in that it encompasses elements of scene recognition, object recognition, motion estimation, and even non-visual cues such as audio semantics. Having begun by recognizing short, simple activities (running, walking, gesturing, etc.) from fixed, high-quality cameras [1], modern research addresses longer and more complicated events (changing a vehicle tire, etc. [2]) in images scraped from the web. In order to address the semantic complexity of such events, and the intra-class variability introduced by viewpoints and quality changes, multiple features are used in order to combine various cues.

At a high level, methods to combine cues from various modalities can be broken down into feature-level fusion (also known as 'early fusion'), score fusion (also known as 'late fusion') and decision fusion, based on the stage at which the combination takes place; see Figure 1. The use of feature-level fusion methods, including Multiple Kernel Learning (MKL) [3], typically precludes the use of different classifier types for different features, and learning a single MKL model with many degrees of freedom can be more difficult when training data (particularly the number of positive instances) is limited. Decision fusion, where the binary (thresholded) outputs of a bank of base classifiers are combined, suffers from discretization issues when a small number of base classifiers are used to rank a large dataset. Score fusion, by contrast, naturally produces continuous outputs for ranking of large test sets, and the greatly-reduced model complexity (vs. feature fusion) makes it amenable to learning approaches even when the amount of training data is limited.

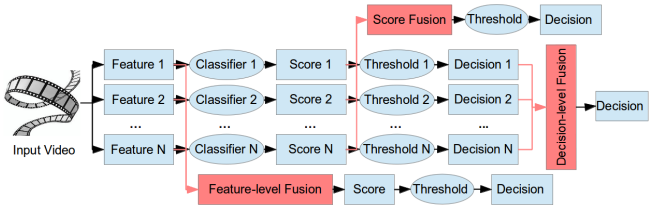

Figure 1. An illustration of different fusion approaches in the pipeline of a detection or classification system.

Within the domain of score fusion approaches, the simplest use mathematical functions - such as the arithmetic or geometric mean - which uniformly weight the outputs of each base classifier. In order to account for differences in the base classifiers' performance, learned score fusion models weight the output of a classifier by its performance on a set of training data [4], [5]. The need to allocate part of the (limited) labeled data for fusion training can be lessened in part by 'recycling' base classifier training data, using a method we proposed in [6]. More elaborate score fusion models, such as hierarchical models [7] that encode additional domain knowledge, can perform well but require manual intervention or exhaustive searching over a large configuration space. In past work [8], we have noted empirically that the performance difference between learned models and uniform score fusion decreases as the number of base classifiers grows.

We address score fusion of a small number of base classifiers (we use 5 in our experiments), and automatically learn local models that account for variations in classifier performance *within the video corpus*. Rather than weighting a classifier's output score based on its global performance

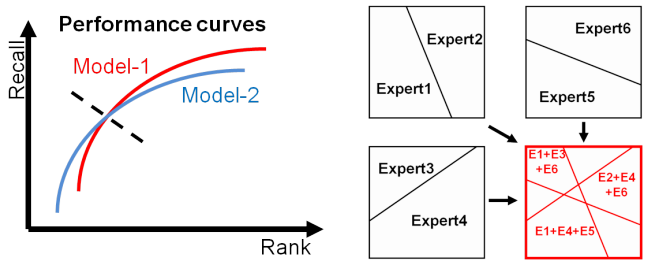

Figure 2. Our previous Expert Forest method for score fusion accounts for changes in the relative performance of base classifiers. In this simplified example (left), the performance ranking of two base classifier models are different on the opposing sides of the dashed line. The expert forest method automatically discovers multiple such partitions in the score space (right), and learns local 'experts' (i.e., linear weights) which optimally combine base classifier scores with respect to the local performance of each.

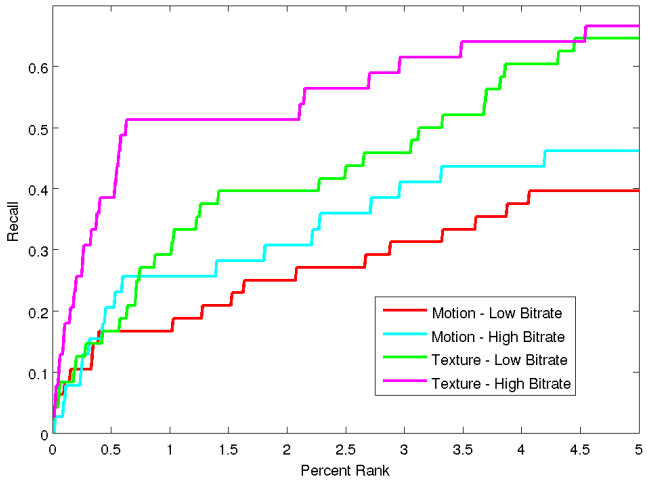

Figure 3. An illustration of the relationship between metadata and classifier performance, as observed in detection results for the event **grooming an animal**. The red and cyan performance curves show that motion features computed on low bitrate video are more reliable than those extracted from high bitrate videos. The green and magenta lines, by contrast, show that our texture feature is less reliable on lower bitrate videos than on higher bitrate videos. Here, 'high' and 'low' simply refer to values above/below the median value. This suggests that the fusion weight for the motion feature should increase with bitrate, and that the weight for the texture feature should decrease.

over the training data, we learn local models which account for local variations in base classifier performance. In previous work, we learned a model called an *Expert Forest* (EF) [9] where changes in the relative performances of a set of base classifiers is learned over local regions in the score space, i.e. the subset of $\Re^N$ spanned by the output ranges of the N base classifiers. As illustrated in Figure 2, this addresses cases where performance curves cross. While EF works well in many cases, it suffers from an interpretability problem: while we observe a certain ordering of classifier performance in different regions, it's not clear *why* the differences arise across the score space.

In the present work, we learn a different set of local fusion models, where we account for understandable differences in base classifier performance related to video quality metadata extracted from the header information. The metadata allow us to determine salient divisions of the data - e.g., separating the video corpus into highly- and lightly-compressed subsets - from which easily understood performance differences arise. We would expect, for instance, that visual features extracted from highly-compressed video will perform worse than those extracted from lightly-compressed videos, whereas an audio-based classifier would have consistent performance on both subsets. In the following sections, we develop a fully automated algorithm to partition the video corpus according to video metadata, and to learn local weightings which are applied to base classifier scores in order to improve detection performance.

II. Other Related Work

The basic observation motivating our metadata-based fusion algorithm is that the different base classifiers used in a multimedia system will inherently have different sensitivities to the variations in image quality expected in unconstrained videos. Conceptually, we are inspired by quality-based multibiometric systems [10] which fuse identifying features based on their reliability, estimated using sharpness metrics and the like. Compared to this method - and other quality-based fusion approaches in biometrics - a key difference is that we use metadata features which are directly observable from the video header file, whereas sharpness and other image quality metrics used in quality-based biometric fusion must be measured by comparing an image to the expected appearance of an eye/face/etc. image.

Our work is also related to the use of metadata values in detection and classification tasks. In [11], Boutell and Luo use metadata from EXIF headers in JPEG images to predict whether an image was taken indoors or outdoors, since cameras are designed to use (and record the use of) flashs and longer exposure times in light-constrained environments typical of indoor photographs. In [12], we used video metadata (clip duration, frame rate, audio/video bitrate, and pixel count) to detect events in web videos. There, the basic intuition was that the videographer's choice of camera is event-dependent (better cameras are used to record weddings than fishing trips, for instance), and the metadata turns out to be somewhat predictive of the content of a particular clip. Relative to that work, the present paper does not use the metadata directly in detection, but rather uses the metadata as an estimate of the relative performance of a number of different classifiers.

III. Video Quality Impacts on Classifier Performance

Figure 3 illustrates the type of performance variation captured by our algorithm. The four curves show the perfor-

mance of two classifiers - one based on a motion feature and another based on texture - on disjoint subsets of high- and low-bitrate videos from a test set of 25k clips. From this, we see that the performance differs substantially, with the texture feature having 8% lower average precision (AP) on the low-bitrate videos than on the high-bitrate subset and the motion feature having 5% *higher* performance on the low-bitrate videos. In a simplified detection system incorporating only these two features, then, the weighting of the motion feature should be higher on the low-bitrate samples than on the high-bitrate clips, while the opposite should be true of the texture feature.

While this example illustrates an ideal case - one classifier improving while another worsens - the same approach works even when all of the base classifiers have performance changes in the same direction. This is roughly the case for the video length, for instance, where all of the base classifiers work better on longer clips. The key is that the *rates* of base classifier performance changes will differ, so that the relative weighting of their outputs should be adjusted.

IV. Metadata-weighted Fusion Algorithm

The high-level steps of the training algorithm are given in Algorithm 1. Inputs from the training set of N_c video clips consist of the associated ground truth labels L, scores S from N_b base classifiers, and metadata values M over N_m different quantities. The final trained model consists of N_m *pairs* of weights W^i, and performance metrics p^i associated with each. Details of these high-level steps are given in the following sub-sections, and the use of the model in testing is described in the final sub-section.

A. Partitioning

Whereas the example in Figure 3 simply divided the video corpus around the median value of video bitrate, the model learning searches over several different soft partitions in order to determine a boundary with a useful performance variation. In order to avoid degenerate partitions (e.g., a threshold above which all training clips are negatively labeled), we generate $N_t = 20$ candidate thresholds linearly spaced between the 25th and 75th percentiles of the metadata values of positive clips. In limited experimentation, we have found that the results of the training are relatively insensitive to either the number of thresholds or the exact percentile range over which they are spaced. Consistent with other detection problems, we assume that the number of negative training examples far outweighs the number of positive examples, so no explicit mechanism is used to avoid degenerate partitions of the negative examples.

As in [9], we use the general mixture of expert model formulated as

$$P^i(L|M,S) = \sum_{E \in \hat{W}^i_{\{h,l\}}} P^i(E|M) P^i(L|S,E), \quad (1)$$

Input: Labels L, base classifier scores S, and metadata values M for training videos
Output: Fusion weights W, thresholds τ, and performance metrics p^i
foreach *Metadata feature i (columns of M)* **do**
  Generate N_t partition thresholds;
  foreach *partition threshold* $\hat{\tau}^i_j,\ 1 \le j \le N_T$ **do**
    Generate subsets $L_l,\ S_l$ of L, S with metadata features below threshold;
    Determine fusion weights $\hat{W}^i_l$ from $L_l,\ S_l$;
    Generate subsets $L_h,\ S_h$ of L, S with metadata features above threshold;
    Determine fusion weights $\hat{W}^i_h$ from $L_h,\ S_h$;
    Compute performance metric $\hat{p}^i$ of weights $\hat{W}$;
    if $j = 1$ *or* $\hat{p}^i > p^i$ **then**
      $p^i = \hat{p}^i$;
      $W^i = \hat{W}^i$;
      $\tau^i = \hat{\tau}^i_j$;
    end
  end
end

Algorithm 1: Metadata-weighted fusion model learning.

where $P^i(E|M)$ is the 'gate' function, indicating which set(s) of weights is responsible for generating each data. Adopting that notation, E is an *expert* on clips with *high* or *low* (h/l) metadata values

The simple formulation for the gate function would be

$$P^i(h|M) = \begin{cases} 1 & \text{if } M \ge \hat{\tau}^i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

but a hard decision boundary may lead to unwanted sharp variations around the threshold. So we introduce a transition region to the gate function.

$$P^i(h|M) = \begin{cases} \frac{M-\hat{\tau}^i}{k} & \text{if } M \ge \hat{\tau}^i \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where

$$k = min(\hat{\tau}^i - \overline{\{m \in M : m < \hat{\tau}^i\}}, \overline{\{m \in M : m > \hat{\tau}^i\}} - \hat{\tau}^i), \quad (4)$$

and $P^i(l|M) = 1 - P^i(h|M)$.

B. Learning Weights

Let $S = (s_1, \ldots, s_M, 1_n)$ be an N-by-$M+1$ likelihood matrix with entry $s(i,j)$ the score from the jth base classifier on the ith clip, and 1_{N_c} a N_c-by-1 vector appended for adjusting the global offset. $L \in \{0,1\}^{N_C}$ is the binary vector of training labels, and Λ a diagonal matrix with the ith entry indicating the gate response on the metadata of the ith video clip.

The weights for each (low/high) subset of the training data are computed to minimize the mean squared error (MMSE)

on the training data, as

$$W = \text{argmin}(S * W - L)^T \Lambda (S * W - L). \quad (5)$$

The regularized *MMSE* solution is given by $w = (X^T \Lambda X + \lambda I)^{-1} X^T \Lambda L$. Because we do not assume that the scores are normalized across the base classifiers, our model includes 1_{N_C} in the score matrix S and learns an extra weight. Without constraining $\|W\|$ to be a unit vector, the local fusion expert simultaneously adjusts the offset and scale variance of each source.

C. Performance Metrics

In order to select the optimal partition threshold, the training algorithm computes and compares performance metrics for different candidates, and the algorithm chooses the set of weightings which are best for each metadata value. For this step, we have experimented with various measures, including

1) **Performance variation of the base classifiers** - consistent with the intuition developed around Figure 3, we can measure each base classifier's performance on each subset of the training data, and favor those thresholds which provide a wider performance disparity. In particular, we measured the variance (over the base classifiers) of the difference in area under the curve (AUC) measured on the two sub-sets.
2) **Quality of MMSE fit** - Having used the MMSE weights, we attempted to measure the performance of a weight set by the MMSE residual, i.e. $\|L - P^i(L|M,S)\|$.
3) **Average precision** - In this option, we generate the fused scores $P^i(L|M,S)$ for a candidate pair of weightings, and compute the average precision of the ranked list, compared to the ground truth labels L.

Because the later two options are measured on the same training data that was used to learn the weights, we expect that, due to over-fitting, these metrics may not accurately reflect performance on the test data. Metric #1 should suffer less from this but, in practice, we find very little difference between the fusion performance of models trained using the three metrics. For the experiments, we use metric #3.

In addition to computing the performance metric for each set of weightings, we compute metric on global model to determine whether the local models improve overall performance. The performance metric of the global model is stored as p_g. At test time, the local weightings are used for a given metadata feature i only if $p^i > p_g$.

D. Using Models in Test

Having trained the fusion model as described above, the algorithm producing the fused score on test clips is described in Algorithm 2. For each metadata feature, we compute its estimated label as a linear combination of base classifier scores, with weights determined by the metadata (working through the gating function). The overall fused score for a particular clip is the average of the scores predicted given each metadata feature, the average using only those features whose weighting pairs out-perform the global model.

Input: Base classifier scores s and metadata values m.
The learned model: fusion weights W,
thresholds τ, performance metrics p
Output: The predicted detection probability $P(L|m,s)$
foreach *Metadata feature* i **do**
  Compute the gate functions $P^i(\{h,l\}|m)$ by eq. 3;
  Extend the score vector s by adding a 1 at the end;
  Set $P^i(L|m,s) = P^i(h|m) \cdot W_h^i s + P^i(l|m) \cdot W_l^i s$;
  Generate N_t partition thresholds;
  if $p^i > p_g$ **then**
    $\chi^i = 1$;
  end
  else
    $\chi^i = 0$;
  end
end
$P(L|m,s) = \frac{1}{\sum \chi} \cdot \sum_i \chi^i \cdot P^i(L|m,s)$

Algorithm 2: Metadata-weighted fusion model use in test.

V. Experiments

We test our score fusion algorithm on 10 semantically-complex events, using base classifier scores derived from 5 different multimedia features. The TRECVid data we use for testing has 5 different metadata values. Because our fusion method requires training data, we present Average Precision (AP) performance measures over 10 random 50/50% splits of a 25k clip archive into training and testing sets. Results are given in Sec. VI. The following sub-sections provide additional detail about the events, metadata, and classifiers

A. TRECVid Events

The Multimedia Event Detection (MED) task was added to the annual TRECVid[1] evaluation in 2011 to assess the performance of event detection techniques on open source video clips. The evaluation provides training and testing video clips for several semantically rich events. Our results are presented on the MEDTest data, and ten events:

- **E06** - Birthday party
- **E07** - Changing a vehicle tire
- **E08** - Flash mob gathering
- **E09** - Getting a vehicle unstuck
- **E10** - Grooming an animal
- **E11** - Making a sandwich
- **E12** - Parade
- **E13** - Parkour
- **E14** - Repairing an appliance
- **E15** - Working on a sewing project

[1] http://trecvid.nist.gov/

The evaluation is quite challenging for several reasons. First, pairs of similar events - such as flash mob and parade, both of which involve large numbers of people moving in formation - are tested in order to evaluate fine semantic categorization. Second, the range of video types - from broadcast-like videos in *making a sandwich*, through heavily post-processed consumer video in *parkour*, to more informal video in *birthday party* - is much broader than other activity recognition datasets consisting only of a certain type (e.g., UCF Sports [13] consisting mostly of broadcast video).

Note that, while MEDTest contains annotated examples of an additional 10 events, they are not used in our evaluation because - as we and other TRECVid teams have observed - the positive annotations seem incomplete when results are manually viewed. For each event type, approximately 100 positive training examples are given. The testing set consists of about 25,000 clips, with labeled instances of the events.

B. Metadata Features Used

All TRECVid video clips have been transcoded to MPEG4, and certain metadata (camera make, model, etc.) has been deleted. The remaining metadata, which we use in our experiments, are: clip duration, video framerate, video bitrate, audio bitrate, and frame resolution (which we quantify as pixel count, i.e. frame width times height). In a more general application, such as event detection from YouTube videos, we expect that additional metadata (camera type, time/date, etc.) would be available and would help improve the results presented below.

C. Base Classifiers and Scores

Our experiments were performed with $M = 5$ base classifiers, each of which estimates event probability based on a different multimedia feature.

- **C1 (audio)** Low-level audio information is captured using Mel-Frequency Cepstral Coefficients (MFCCs), computed using the HTK Speech Recognition Toolkit[2], and a Support Vector Machine (SVM) with Histogram Intersection Kernel (HIK) is trained using a bag of words quantization of the MFCC features.
- **C2 (visual - motion)** A Bag of Words (BOW) Histogram of Optical Flow (HOF) feature extracted on Dense Trajectories [14], classified with a bagged HIK SVM. Note: this is the motion feature whose performance is shown in Fig. 3.
- **C3 (visual - objects)** Max frame-level Object Bank [15] scores maximized across video, classified with a bagged HIK SVM.
- **C4 (visual - texture)** Spatial pyramid feature on a Histogram of Oriented Gradient (HOG 2D) [16], classified with an HIK SVM. Note: this is the texture feature whose performance is shown in Fig. 3.
- **C5 (visual - color)** Color SIFT BoW feature [17] with 4096 codewords and spatial pyramid (1 global and 3 vertical layers), classified with an NGD kernel SVM.

While each of the base classifiers is trained to produce test scores as likelihoods in the range $[0, 1]$, the use of different kernels and other training options result in different score distributions. In this situation, we have found in [9] that learned models out-perform uniform weighting fusion (via the geometric or arithmetic mean) because the weights provide a way to account for the varying distributions. More recently, we have found that fusion by geometric mean can out-perform many learned models *as long as the scores are normalized to have similar distributions*; in particular, the means and standard deviations of the scores from each base classifier must match. In order to provide high performance with untrained geometric mean-based fusion, the scores from each of the base classifiers listed above are normalized with respect to an arbitrarily-chosen classifier's distribution.

Each base classifier is trained on a set of 100 positive instance clips, and a common set of negative (background) clips. Each is then tested against the 24k clips in MEDTest, and scores are stored in intermediate files for use in fusion. Base classifier training and testing was carried out by researchers at several partner institutions listed in the acknowledgements below.

VI. Results and Analysis

As mentioned above, the MEDTest scores from each of the base classifiers is randomly partitioned into a training half and testing half, in order to evaluate various trained fusion models (including ours). In order to evaluate the performance of each fusion method, we compute the Average Precision (AP) of each fusion method on each of the 10 events. This process is repeated 10 times for different random partitions of MEDTest into training and testing portions, and the AP scores are averaged over these 10 runs.

For comparison, we use the same methodology to determine the performance of several other fusion methods:

- **Geometric mean,** using Matlab's geomean function.
- **Global MMSE,** i.e. a global model of linear weights learned by minimizing the MSE (eq. 5) on the training data.
- **Expert Forest,** using our code from [9].
- **Best base classifier,** i.e. the performance of the best base classifier. While not a fusion scheme, these results are shown in order to demonstrate how the fusion improves on the performance of the base classifiers.

AP results are shown in Table I. Considering all of the events, the mean average precision (mAP) is highest using the metadata-based fusion method presented here. Geometric mean-based fusion is second best, followed by expert forest; all of the fusion methods out-perform the best base classifier for each event, demonstrating the utility of using fusion

[2]http://htk.eng.cam.ac.uk/

Event	Geometric Mean	Global MMSE	Expert Forest	Best Base	Ours
Birthday party	0.2516	0.2712	0.2405	0.2160	**0.3208**
Changing a vehicle tire	0.3719	0.3215	0.3179	0.3279	**0.4179**
Flash mob gathering	**0.6383**	0.6193	0.6058	0.5835	0.6216
Getting a vehicle unstuck	**0.4038**	0.3864	0.3403	0.3744	0.3865
Grooming an animal	**0.2111**	0.1564	0.1846	0.1602	0.1839
Making a sandwich	0.1827	0.1694	0.1853	0.1806	**0.2072**
Parade	0.4045	0.3263	0.3764	0.3168	**0.4097**
Parkour	0.4741	0.4664	**0.5037**	0.3622	0.4865
Repairing an appliance	**0.5346**	0.5014	0.5176	0.4005	0.5180
Working on a sewing project	0.2043	**0.2188**	0.1972	0.1790	0.1906
Average of Events	0.3677	0.3437	0.3469	0.3101	**0.3743**

Table I

Fusion performance: average precision (AP) on events 6-15, AND MEAN AVERAGE PRECISION (MAP). FOR EACH EVENT, THE BEST AP IS SHOWN IN BOLD. OUR METHOD IS COMPARED TO AN UNTRAINED GEOMETRIC MEAN, THE USE OF A GLOBAL SET OF WEIGHTS MINIMIZING MMSE ON THE TRAINING SET, AND OUR PREVIOUS EXPERT FOREST METHOD. WE ALSO SHOW THE PERFORMANCE OF THE BEST-PERFORMING BASE CLASSIFIER, TO DEMONSTRATE THAT THE FUSION ALWAYS ADDS TO SYSTEM PERFORMANCE.

in detection systems. Though the performance metrics are different, the improvement of expert forest relative to the global MMSE is narrower than noted in [9]. Part of the reason for this is the reduction in the number of positive training examples from an average of about 130 per event to 100 per event in these results.

Looking at the individual events, there is no clear relationship between the method's performance and any particular quality of the data. Our metadata-based fusion algorithm performs well on *birthday party,* where positive examples are of diverse quality, and also performs better than others on *making a sandwich,* where many of the positive instances in the test data come from broadcast video. Given that the algorithm selects a variable number of metadata features while building the model, one potential hypothesis is that there may be a relationship between the number of such models and the performance of the metadata-based fusion. In order to assess this, table II shows whether any particular metadata feature was used for each event. This explains, for instance, that our new algorithm and the global MMSE method perform very similarly on *getting a vehicle unstuck* because the metadata model consists of only two sets of weights based on a partitioning of the data with respect to audio bitrate. More surprisingly, the global model outperforms all other methods on *working on a sewing project,* despite the metadata-based model computing sets of weights for each different feature. There seems to be no clear relationship between the number of metadata features used in the model and its relative performance.

Table II also indicates that each metadata feature was used in at least half of the events. The least-used metadata feature, clip duration, was somewhat surprising given that each base classifier's performance correlates strongly with the duration of the clip. However, since each of the base classifiers have the same tendency to better performance on long clips, partitioning the set with respect to this value doesn't shift their relative performance.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a method to learn fusion models which incorporate the relationship between a video's quality and the expected relative performance of several base classifiers using different multimedia descriptors. The video quality is quantified using metadata values incorporated in the video header, and we show that several easily-understood relationships - between video bitrate and the performance of texture descriptors, for instance - have an impact on system performance. By incorporating this dependence in our model, we show that fusion performance is improved relative to several other methods.

Our experiments, and indeed the TRECVid test data, are somewhat contrived in that every clip had exactly the same five metadata features. In a real world setting, where additional video metadata may be available on a clip-by-clip basis, our models would need to account for the possibility that metadata for test clips might be missing. If a different number of weight pairs were used for different clips while scoring the same event, this might lead to a mis-match of fused score distributions, so additional methods would be needed to avoid this problem.

ACKNOWLEDGEMENTS

The work described in this paper was done as part of a collaboration with Kitware Inc., Simon Fraser University, and Georgia Tech, who provided base classifier scores. The authors would like to specifically thank Ilseo Kim, Megha Pandy, Sangmin Oh, and Amitha Perera from Kitware; Arash Vahdat, Kevin Cannons, and Greg Mori from SFU; and You-Chi Cheng from Georgia Tech.

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20069. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

Event	Clip Duration	Frame Rate	Video Bitrate	Audio Bitrate	Pixel Count
Birthday party		X	X	X	X
Changing a vehicle tire	X	X	X		
Flash mob gathering		X			X
Getting a vehicle unstuck				X	
Grooming an animal	X	X	X	X	X
Making a sandwich		X	X	X	
Parade	X	X	X	X	X
Parkour	X	X	X	X	
Repairing an appliance			X	X	X
Working on a sewing project	X	X	X	X	X
Times Used	5	8	8	8	6

Table II
WHICH METADATA FEATURES WERE USED FOR EACH EVENT? THIS TABLE INDICATES WHETHER A PAIR OF LOCAL WEIGHTS OUT-PERFORMED THE GLOBAL MMSE-MINIMIZING WEIGHTS FOR EACH METADATA FEATURE. THE LAST ROW SUMMARIZES THE NUMBER OF EVENTS WHICH USED A PARTICULAR METADATA FEATURE.

Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DOI/NBC, or the U.S. Government.

REFERENCES

[1] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *ICPR*, 2004.

[2] P. Over, G. Awad, J. Fiscus, B. Antonishek, A. F. Smeaton, W. Kraaij, and G. Quenot, "TRECVID 2010 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics," in *TRECVID*, 2011.

[3] A. Kembhavi, B. Siddiquie, R. Miezianko, S. McCloskey, and L. Davis, "Scene it or not? incremental multiple kernel learning for object detection," in *IEEE International Conference on Computer Vision*, 2009.

[4] I. Kim, S. Oh, B. Byun, A. G. A. Perera, and C.-H. Lee, "Explicit performance metric optimization for fusion-based video retrieval," in *ECCV Workshops*, 2012.

[5] I.-H. Jhuo, D. Liu, G. Ye, and S.-F. Chang, "Robust late fusion with rank minimization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[6] J. Liu, S. McCloskey, and Y. Liu, "Training data recycling for multi-level learning," in *IEEE International Conference on Pattern Recognition*, 2012.

[7] P. Natarajan, S. Wu, S. N. P. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, R. Prasad, and P. Natarajan, "Multimodal feature fusion for robust event detection in web videos." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[8] S. Oh, S. McCloskey, I. Kim, A. Vahdat, K. Cannons, H. Hajimirsadeghi, G. Mori, A. G. A. Perera, M. Pandey, and J. J. Corso, "Multimedia event detection with multimodal feature fusion and temporal concept localization," *Machine Vision and Applications*, vol. 25, no. 1, 2014.

[9] J. Liu, S. McCloskey, and Y. Liu, "Local expert forest of score fusion for video event classification," in *European Conference on Computer Vision*, 2012.

[10] K. Nandakumar, Y. Chen, A. Jain, and S. Dass, "Quality-based score level fusion in multibiometric systems," in *IEEE International Conference on Pattern Recognition*, vol. 4, 2006.

[11] M. Boutell and J. Luo, "Bayesian fusion of camera metadata cues in semantic scene classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[12] S. McCloskey and P. Davalos, "Activity detection in the wild using video metadata," in *IEEE International Conference on Pattern Recognition*, 2012.

[13] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach: a spatio-temporal maximum average correlation height filter for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[14] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[15] L.-J. Li, H. Su, E. P. Xing, and F.-F. Li, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in *Neural Information Processing Systems (NIPS)*, 2010.

[16] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo." in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[17] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Empowering visual categorization with the gpu," *IEEE Transactions on Multimedia*, vol. 13, no. 1, pp. 60–70, 2011.