Local Expert Forest of Score Fusion for Video Event Classification

Jingchen Liu[1], Scott McCloskey[2], and Yanxi Liu[1]

[1] Penn State University, State College, PA, USA
[2] Honeywell Labs, Golden Valley, MN, USA

Abstract. We address the problem of complicated event categorization from a large dataset of videos "in the wild", where multiple classifiers are applied independently to evaluate each video with a 'likelihood' score. The core contribution of this paper is a local expert forest model for meta-level score fusion for event detection under heavily imbalanced class distributions. Our motivation is to adapt to performance variations of the classifiers in different regions of the score space, using a divide-and-conquer technique. We propose a novel method to partition the *likelihood-space*, being sensitive to local label distributions in imbalanced data, and train a pair of locally optimized experts each time. Multiple pairs of experts based on different partitions ('trees') form a 'forest', balancing local adaptivity and over-fitting of the model. As a result, our model disregards classifiers in regions of the score space where their performance is bad, achieving both local source selection and fusion. We experiment with the TRECVID Multimedia Event Detection (MED) dataset, detecting 15 complicated events from around $34k$ video clips comprising more than 1000 hours, and demonstrate superior performance compared to other score-level fusion methods.

1 Introduction

Content-based exploitation and retrieval of digital video from large datasets is an important topic in computer vision, with a wide range of potential applications. Recently, with the rapid growth of multimedia data shared on platforms such as YouTube, people have switched their focus from recognizing simple events, *e.g.* single person waving (KTH dataset [1]) from high-quality videos (static camera, clean background) [2,3] to more complicated events that contain multiple object-interactions, *e.g.* boxing (MSR dataset [4]) within uncontrolled videos (hand-held camera, cluttered background) [5] or movies [3].

In this paper, we address the detection of complex events from video clips in a large multimedia archive (1000+hr collection of about $34k$ clips from the Multimedia Event Detection task of TRECVID2011 [6])[1], where the videos are uncontrolled with respect to camera motion, background clutter and human editing. The major challenges that typically come with video event detection are

[1] `http://www.nist.gov/itl/iad/mig/med11.cfm`

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part V, LNCS 7576, pp. 397–410, 2012.

(a) (b) (c) (d) (e) (f) (g)

Fig. 1. Example frames from training clips. (a-e) illustrates the wide semantic intra-class variation of the same event category: '*attempting a board trick*. (f,g) illustrates inter-class variation, with example frames from '*feeding an animal*', and '*landing a fish*. These frames illustrate various challenges *e.g.* rapid motion blur and background clutter (a), insufficient lighting and captions from post-processing (f), as well as a wide ranges of camera viewpoint and scene scale.

(1) wide intra-class and inter-class variation; (2) high-dimensional features; and (3) imbalanced labeled data.

As shown in Fig.1, the event categories exhibit both wide intra-class variation (*attempting a board trick*), broad inter-class variation, and rich temporal structure (*e.g. changing a vehicle tire* or *making a sandwich*) which can't be estimated from a single frame. Moreover, given the variety of real-world videos, any particular event class, *e.g. wedding ceremony*, only composes of a tiny proportion in the entire video database, which results in an imbalanced labeled data for training one-versus-all classifiers to detect each particular event.

We first introduce the larger system which motivates our choice of score-fusion for event detection, and review other approaches to score fusion. We then propose the idea of local expert forest for score fusion that resolves the dilemma between local adaptivity and over-fitting from imbalanced training data. We then demonstrate the method's superior performance to alternative methods.

1.1 Video Retrieval System

The ultimate goal of a content-based image or video retrieval system is to allow people to browse large multimedia archives in useful ways. While our experiments are focused on activity-based browsing, i.e. finding video clips that show the same event type, our score fusion approach can be used for other modalities such as object-based browsing. In order to address the wide range of event categories illustrated above, a useful system must incorporate a range of visual features, audio features, and classification methods associated with the features. Fig.2 shows our system architecture, in which several types of features are extracted from the video archive, and are stored in a database. A bank of *base classifiers* follow, each of which are trained to produce a likelihood score based on a subset of the features. Their outputs for a particular event are then fused by our method, and the resulting fused likelihood is used to rank the clips in the archive relative to the operator's interest.

By including a wide range of video and audio features, the system can better handle semantically diverse events. In our experiments (Sec. 3), we have found that no single feature provides acceptable performance across all event categories, with acoustic features outperforming others on *birthday party*, motion features

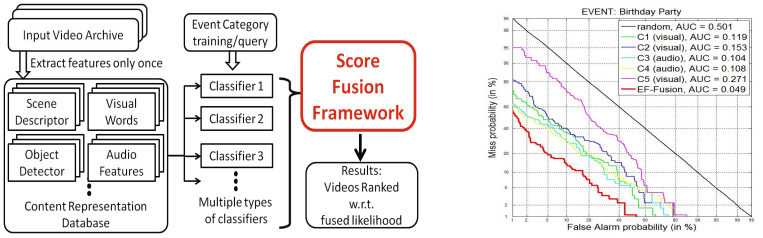


Fig. 2. (a): Architecture of a general content based video retrieval system, within which our score-fusion is a key part; **(b): DET curves showing base classifiers and fusion performances**, 3 visual classifiers and 2 audio classifiers are fused together, reducing both miss and false alarm rates. The overall performance of each curve is quantified using Area Under Curve (AUC).

outperforming others on *flash mob gathering*, and object detections outperforming others on *getting a vehicle unstuck*. Our database of rich features can also be used to classify ad hoc event categories without the need to re-process the archive clips. Having a large feature set complicates fusion, however, because not all features will be useful with respect to a particular event. Therefore, a good fusion system must identify and ignore such non-discriminative features (and their associated base classifiers).

Performing the fusion at the score level abstracts away the details of the underlying classifiers, and allows us to use different classification methods for the features to which they are best suited. For instance, score fusion allows numerical combination of temporal models (e.g. Dynamic Bayes Nets (DBNs)[7]) on 3D features for spatio-temporal matching with kernel methods (e.g., SVM) applied to bag of word-type features. The system should also permit the later introduction of scores from classification schemes as they are introduced (*e.g.*, [8,4,5]). In that sense, score-level fusion is preferred, as we need only re-train the fusion part when a new classifier's output is provided.

Because each base classifier layer produces a scalar likelihood value from a high-dimensional audio/video feature, fusion in the score space is generally faster than fusion at the feature level. This advantage in training complexity can be used to provide robustness to missing features, by training multiple models for base classifier combinations that may be given for any particular clip at evaluation time. While the full power set may not be necessary, TRECVID clips occasionally lack audio data, requiring separate fusion models for video-only and audio+video base classifier sets. As compared to voting methods for decision-level fusion, score-fusion is preferred because the output is still a continuous score (likelihood) for ranking the archive clips.

An example fusion result is plotted in detection-error-trade-off (DET) curve shown in Fig. 2(b), which is very similar with Receiver Operating Characteristic (ROC) curve but uses a nonlinear scale on the two axis (false alarm/miss detection prob.), so that the curves are more 'linear' [9]. This DET curve illustrates our fusion performance (from 3 visual classifiers and 2 audio classifiers) in

detecting *birthday party*, which gives 6% missed detections under 20% false alarm. This is a significant improvement over the best base classifier, which has 20% miss at the same false alarm level.

1.2 Other Work in Score Fusion

Discriminative score fusion differs from classification/regression problems in that it takes continuous and semantically meaningful input (likelihood scores) with discrete labels to produce continuous output (a fused likelihood score) for ranking. On the other hand, score fusion is similar to ensemble stacking, where separate training data are used for base classifier- and meta-level (fusion) training.

Fusion techniques are widely applied in various applications. In biometric systems, [10] combines the matching scores from multiple modalities based on generalized densities estimated from the score space, but requires a large amount of training data to approximate the density distribution. For object tracking, Yin [11] fuses multiple likelihood maps (of motion, saliency, template matching, etc.) using minimum mean-squared error (MMSE) linear fusion. For hand detection, Mittal [12] proposes to use a linear SVM classifier to fuse three independent detector scores based on shape, context and skin, respectively.

In large scale detection and ranking systems such as search engines, linear score fusion is widely used [13,14,15]. Here, area under an ROC curve (AUC), similar to average precision (AP), is widely used as a measure of the overall performance of the retrieval system [16,17], instead of evaluation at a single operating point. AUC optimization is also equal to the Wilcoxon-Mann-Whitney (WMW) ranking

$$AUC = \sum_{i=1}^{N^+} \sum_{j=1}^{N^-} I(p_i, n_j)/N^+N^-, \tag{1}$$

where $I(p_i, n_j)$ is 1 if $p_i > n_j$ and zero otherwise, p_i and n_j are the scores of N^+ positive and N^- negative samples, respectively. Because Eqn.1 is nonlinear and discontinuous, people have used continuous functions, *e.g.*, a sigmoid, to approximate the function and have optimized via gradient descent [17]. However, when the substitute function is too smooth, it no longer approximates Eqn.1. On the other hand, if the continuous function is sharp enough to approximate Eqn.1, gradient descent solutions may become unstable.

Linear fusion, producing a fused score s from base classifier scores s_k as $s = \sum_k s_k w_k$, is a popular method to combine likelihoods with non-negative weights $w_k \geq 0$. This is intuitive as a high likelihood output from the base-classifier should also indicate a high likelihood after fusion, and the different weights account for the relative performance of the classifiers. This model has demonstrated strong performance ([11,13,14,15]) especially in terms of generalization on unseen testing data (Sec.3).

On the other hand, the obvious drawback of linear fusion is that it offers limited degrees of freedom, as each base-classifier is assigned a fixed weighting. This is illustrated in Fig. 3(a), where the notional performances of two base classifiers are plotted in red and blue DET curves respectively. Classifier C_2 has relatively

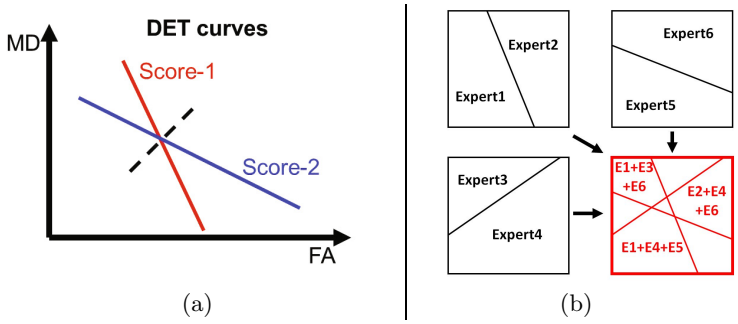


Fig. 3. Motivation and high-level approach. In order to account for local performance variations, as illustrated by crossing DET curves in (a), we learn a local expert forest over the score space. Multiple partitions and expert pairs within the score space (black plots in (b)) are combined to an expert forest (red part) which accounts for local performance variations.

better performance at the top-left region (data with high likelihood score), and C_1 outperforms C_2 at the bottom-right region (data with low likelihood score).

This observation motivated us to weight base-classifiers differently in regions of the M-dimensional score space defined by the outputs of M base classifiers. A mixture of local expert model (*MoE*) [18] appears to be a promising solution since it provides such a local flexibility, and because non-negative linear fusion can still be performed within each local region to provide good generalization.

However, there are several drawbacks of *MoE* when applied to our system. First, *MoE* is typically solved using expectation maximization (EM) iteration, the performance of which heavily depends on the initialization, especially when the base-classifier score distribution is more scattered than clustered, as in our case. More importantly, when positive samples of each class are limited and vastly outnumbered by negative samples, splitting the entire space into multiple pieces may still result in over-fitting within some local regions. This imbalance is particularly acute in the TRECVID data, where each event category has 100+ positively labeled training clips, as compared to $2k+$ negative training clips.

We propose a novel Expert Forest model resolving the above concerns. We use a combination of the basic *MoE* unit – a one-layer binary partition with two local experts. We carefully divide the score space in two, being sensitive to label distributions in each cluster so that the experts have enough data to avoid over-fitting, while still being able to adapt to local data properties. We apply linear fusion with non-negative weight constraints on local clusters, so that each local model has strong generalization while allowing for base classifier selection. The training of an expert with a linear binary partition is much simpler than EM optimization in a high-dimensional space. Multiple local experts that have overlapping regions jointly contribute to the weight set used (Fig.3(b)), so the overall model gains a much higher degree of freedom to adapt to local properties while maintaining good generalization ability.

2 Random Forest of Local Fusion Experts

The general framework is shown in Fig. 4, where we have multiple 1-layer binary partition trees. Each 'tree' divides the score space in two, and then handles them independently. At test time, input score vectors first go through multiple 'gates' to determine which weight sets are applied to data in that part of the score space. The weighted scores from each of the trees are combined (averaged) in order to generate the final fused output. In this section, we first address the key problem of how to find good partitions in the presence of heavily imbalanced data for a *MoE* model, and then give solutions to expert forest (*EF*) model optimization under this guided space partition.

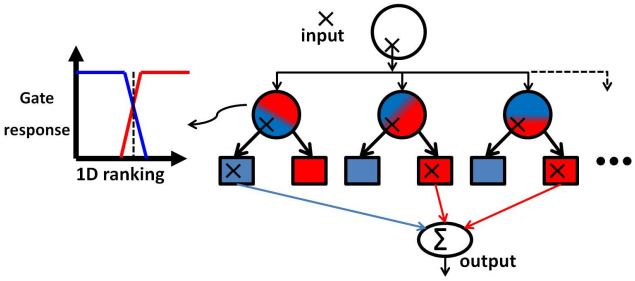


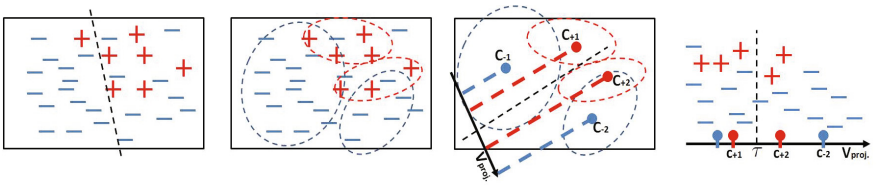
Fig. 4. Local expert fusion model framework, in which an input vector containing M base classifier scores is fused to a single scalar output. The input's position X in the score space (notionally shown by the white circle) determines, for each tree, which set of fusion weights are applied. The fused weights from each of the trees are combined in order to generate the final fused output.

2.1 Partitioning for MoE under Imbalanced Data

One problem with the *MoE* model data is that, when class labels are not balanced, space partitioning may locally exacerbate that imbalance. This is illustrated in Fig.5(a), where a blind K-means partition ignoring the labels produces one cluster containing mostly negatively-labeled data. Unlike linear discriminant analysis (LDA) or decision trees, which look for a separation between positive and negative labels, our training prefers a balanced distribution of labels in order to prevent local over-fitting.

To address this, we apply K-means separately to the positive (+) and negative (−) samples to partition both types of labels into 2 clusters (c_1, c_2) and merge them in a later stage. Given different random initializations of K-means[2], we can obtain various partitions on both the(+) and (−) samples. Ideally, if a (+)cluster and a (−)cluster spatially overlap completely, we can merge them to form a local space that is rich with both (+/−) labels. In order to evaluate the consistency of merging a binary partition on (+) labels with a partition on (−) labels, we

[2] To get random partition of clusters with more diversity, we typically run K-means in a random subspace and don't require full convergence.



(a) (b) (c) (d)

Fig. 5. Partitioning the score space in the presence of imbalanced class labels. Naive K-means clustering (a) may exacerbate imbalance, leading to over-fitting. We separately partition both positive and negative samples (b), project them on a 1D axis (c), and choose a threshold τ along that line (d) to partition the samples.

adopt the idea of mutual information, which indicates the spatial overlap of $(+/-)$clusters to be merged.

Let a binary random variable '+'$\in \{c_{1+}, c_{2+}\}$ indicate a sample data can belong to 1 of the $(+)$clusters, with prob. $p(+ = c_{1+}) + p(+ = c_{2+}) = 1$. Similarly the same sample can also belong to either of the negative clusters with $p(- = c_{1-}) + p(- = c_{2-}) = 1$. Therefore the mutual information between 2 random variable '+' and '−' is given by

$$I(+;-) = \sum_{\substack{\text{`}+\text{'}\in\{c_{1+},c_{2+}\} \\ \text{`}-\text{'}\in\{c_{1-},c_{2-}\}}} p(+,-)\log\left(\frac{p(+,-)}{p(+)p(-)}\right) \tag{2}$$

$I(+;-)$ here can also be interpreted as 'co-occurrence' character: given one data sample from a particular '+' cluster, how much do we know about which '−' cluster that it belongs to. A higher co-occurrence indicates stronger overlapping of the cluster areas, and is thus preferred.

Given K different binary partitions on $(+/-)$ samples respectively, we evaluate K^2 pairs of associations according to Eqn. 2 and then select the top K associations (we use $K = 20$ through all the experiments). For each of top ranked associations, let the cluster centers be C_{+1}, C_{-1}, C_{+2} and C_{-2}, we use LDA to find the 1D projection vector v_{proj} that best separate one pair of positive and negative cluster centers from the other pair (Fig.5(c)). A partition of the score space is thus defined by the projection vector v_{proj} and a 1D threshold τ. To avoid the partition becoming ill-posed again as in Fig.5(a), we fix v_{proj} for our system and, during the model optimization stage, we optimize the threshold τ within the range of the middle of the two projected old cluster centers.

2.2 MoE Model

The general mixture of expert model is formulated by

$$P(Y|X) = \sum_{E} P(E|X)P(Y|X,E), \tag{3}$$

where $P(E|X)$ is the 'gate' function, indicating which model is responsible for generating each data. The output of the gate function directly depends on the input X, which differentiates between *MoE* methods and boosting-based models.

In our case of score fusion, we are looking for a score mapping $s(X) \propto P(Y = 1|X)$, and can still adopt the probability representation for a *maximum likelihood* solution of the model parameter $\theta = (w^{(L)}, w^{(R)}, \tau)$

$$\begin{aligned} L &\sim \prod_X P(Y|X,\theta) \\ &= \prod_X \big[\sum_{i=L,R} G^{(i)}(X,\tau) P(Y|E^{(i)}(X,w^{(i)})) \big], \end{aligned} \tag{4}$$

where, the gate of left child as an example is given by

$$G^{(L)}(X,\tau) = \begin{cases} 1 \text{ if } X \cdot w \leq \tau \\ 0 \text{ otherwise} \end{cases} \tag{5}$$

The hard decision in Eqn.5 can also be made soft by introducing a transition region (Fig.4), so that the fusion output will be smooth across the boundary.

We use linear models for local experts, with likelihood function from experts being

$$P(Y|E^{(i)}(X,w^{(i)})) = \exp\{-\|X \cdot w^{(i)} - Y\|^2\}. \tag{6}$$

The *maximum likelihood* model solution of Eqn.4 cannot be solved directly because of the summation term within the multiplication loop. Therefore, we have to iteratively update the 'expectation' of the gate response and 'maximization' of the likelihood at local experts. Because G_i only involves a single parameter τ, once the local experts are updated at each iteration, we can enumerate τ along 1D and directly obtain the optimal value according to Eqn.4.

2.3 Local Expert Training

Let $X = (x_1, \ldots, x_M, 1_n)$ be an N-by-$M+1$ likelihood matrix with entry $X(n,m)$ the score output on clip n from base classifier m, and 1_n a n-by-1 vector appended for adjusting the global offset. $Y \in \{0,1\}^n$ is the binary vector of training labels, and Λ a diagonal matrix with $\Lambda(n,n) = G(x_{(n)})$, indicating the gate response on the score vector of video clip n. $\Lambda(n,n) = 1/0$ (or something in the middle), indicates the video clip n is within/outside the local region (or in the transition region).

The maximum likelihood solution for w of Eqn. 6 is in the same form of the *MMSE* representation:

$$w^* = \text{argmin}(X * w - Y)^T \Lambda (X * w - Y). \tag{7}$$

The regularized *MMSE* solution is given by $w = (X^T \Lambda X + \lambda I)^{-1} X^T \Lambda Y$. Instead of λ-regularization, we apply a non-negative constraint on all weights $w_i \geq 0, i = 1, \ldots, M$. Note that the weight w_{M+1}, corresponding to the offset term,

Table 1. Pseudo-code of training (top) and testing (bottom) our model

1. Generate K random binary partitions on both the positive and negative samples
2. Take K positive and K negative partitions to form K^2 pairs of associations
3. Extract projection vectors from top K association pairs based on Eqn.2
4. **For:** each projection vector v_k
5. Learn kth *MoE* model according to Eqns. 4,5,6.
1. **For:** each *MoE* model k
2. estimate fused score s_k according to Eqns. 3,5,6.
3. Compute the final score $s = \text{avg}\{s_k \vert k = 1, \ldots, K\}$

may be negative. When applied to a bank of base classifiers which perform no worse than random chance, this constraint enforces the intuition that no such classifier should be discredited by the fusion model. With this constraint, we have found that the system has equivalent (or even better) generalization compared to regularization, while still behaving as a convex optimization problem which can be solved efficiently using existing toolboxes such as [19]. When base classifiers have random performance for a particular event, the non-negative constraint can produce a sparse solution (i.e., $\exists\ i\ s.t.\ w_i = 0$).

Because we do not assume that the scores are normalized across the base classifiers, our model includes 1_n in the likelihood matrix X and learns an extra weight w_{M+1}. Without constraining $\|w\|$ to be a unit vector, the local fusion expert simultaneously adjusts the offset and scale variance of each source.

The pseudo-code for training and testing of the *Expert Forest* model is given in Table 1.

3 Experiments

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

- **C1.(visual)** Motion information is captured by a bag of words feature on 3D histograms of oriented gradients [20], which is classified by an SVM with Histogram Intersection Kernel (HIK).
- **C2.(visual)** The relationship between events and objects is captured using the Object Bank feature [21], computed using the reference code, and the maximum response of each detector across the clip's frames is classified with an SVM using HIK.
- **C3.(audio)** Low-level audio information is captured using Mel-Frequency Cepstral Coefficients (MFCCs), computed using the HTK Speech Recognition Toolkit[3], and an SVM with HIK is trained using a bag of words quantization of the MFCC features.
- **C4.(audio)** Higher-level audio information is captured by Acoustic Segment Models (ASM), which is classified using an SVM with HIK.

[3] `http://htk.eng.cam.ac.uk/`

– **C5.(visual)** The relationship between events and their environments is captured using the Gist feature [22], which is computed on a random 20 frame subset of the video, and the 20 outputs of a per-frame linear SVM are averaged to give the C5 base classifier score.

3.1 Experiment Design

We conducted 4 experiments to validate our proposed fusion method on different video events and different base classifiers on the *TRECVID2011* dataset. (a) Detecting five events (E1-E5) from the fusion of four base-classifiers (all except C4); (b) Detecting ten events (E6-E10) from the fusion of all five base-classifiers; (c) A stress test to evaluate the fusion system's robustness by adjusting the quality of the base-classifiers (E7); (d) A stress test on imbalanced label distribution. In training the score fusion for E1-E15, the model is learned on an average of 140 positive instances and 2000 negative instances per event category. For E1-E5, the model is tested on 4292 video clips with on average 2.3% of positive instances; for E6-E15, the model is tested on 32037 video clips with on average 0.37% of positive instances.

Our expert forest model uses $K = 20$ pairs of local experts, and we run bootstrapping (with replacement) on the training data 20 times, each time using the same number of labeled samples, and evaluate the area under the DET curve (AUC) score each time, where lower numbers indicate better performance.

3.2 Baseline Models for Comparison

We compare the results of our score fusion model to several methods using the same base classifier likelihoods. These methods are: score averaging, nonlinear-SVM, RBF network, MMSE- and MFoM-based linear fusion, and a naive MoE fusion without the partitioning of Sec. 2.1. The nonlinear-SVM and RBF network are trained using LibSVM [23], using a Gaussian kernel, and we perform cross-validation to optimize both the kernel width and the different weighting for positive/negative instances (to handle label imbalance). The MoE model uses 4 local experts initialized using K-means and optimized using EM.

As mentioned in Sec.1.2, we use AUC to evaluate the fusion methods across a range of operating points[4]; a random system will have $AUC = 0.5$. Such a discrete metric is equal to the normalized Wilcoxon-Mann-Whitney ranking statistic [24,17] and also similar to the average precision.

3.3 Results and Comparisons

The average AUC over the 20 runs on the 15 event categories are given in Tables 2 and 3. Our approach gives on average the best performance on all but one of

[4] We do this instead of measuring performance at a particular operating point on the curve, which may be evaluated as the proportion of inconsistent score pairs, *e.g.* a negative clip ranked higher than a positive clip.

Table 2. Fusion performance (AUC) on event 1-5, with 4 base classifiers. For each event, the best AUC is shown in bold.

Event	Best Base	Avg.	SVM	RBF	MoE	MMSE	MFoM	Ours
Attempting a board trick	.078	.075	.103	.078	.060	.062	.071	**.055**
Feeding an animal	.199	.191	.209	.212	.172	.172	.175	**.167**
Landing a fish	.065	.084	.112	109	.082	.061	.067	**.055**
Wedding ceremony	.046	.042	.065	.043	.055	**.030**	.046	.035
Woodworking	.124	.096	.135	.089	.079	.083	.089	**.075**

Table 3. Fusion performance (AUC) on events 6-15, with 5 base classifiers. For each event, the best AUC is shown in bold.

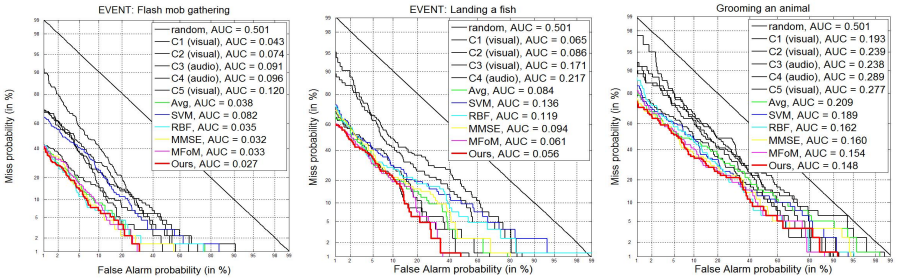
Event	Best Base	Avg.	SVM	RBF	MoE	MMSE	MFoM	Ours
Birthday party	.115	.082	.138	.089	.071	.062	.061	**.056**
Changing a vehicle tire	.144	.130	.106	.110	.112	.089	.113	**.087**
Flash mob gathering	.043	.038	.076	.037	.028	.033	.031	**.024**
Getting a vehicle unstuck	.105	.073	.115	.088	.060	.058	.057	**.050**
Grooming an animal	.193	.209	.175	.159	.150	.153	.156	**.148**
Making a sandwich	.123	.135	.128	.107	.106	.113	**.101**	**.101**
Parade	.072	.063	.127	.072	.056	.055	.051	**.047**
Parkour	.070	.092	.135	.099	.062	.067	.065	**.058**
Repairing an appliance	.087	.066	.112	.057	.039	.074	.040	**.035**
Working on a sewing project	.152	.190	.186	.174	.142	.156	.137	**.133**

the event categories. Averaged over the 20 runs on 15 event categories, our score fusion produces an AUC of 0.075, which is 30% lower than the best base classifier. MMSE and MFoM share the second tier of performance, with average AUCs that are 10.0% and 9.4% higher than ours, on average. We also found that the MFoM method, which optimizes the AUC score on the training data, does not provide optimal AUC on unseen testing data. Moreover, MFoM's optimization of AUC is prone to termination at local minima, resulting in poorer performance. The results also show that the nonlinear classification/regression methods of SVM and RBFNN performed much worse than linear models as measured by AUC. Although the non-linear methods may have stronger discriminative ability and give better performance at a particular operating point, the nonlinear kernel mappings applied on the data break the original ranking from all the base classifiers, reducing ranking performance of the system especially on unseen data.

Sample DET curves are shown in Fig 6, and illustrate that, while different fusion methods may occasionally perform slightly better than ours at certain operating points, we show better performance across the range of fused scores.

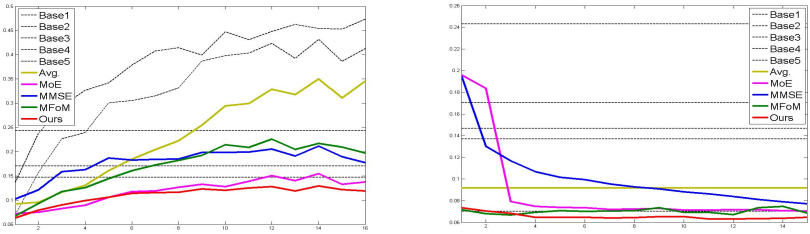
In addition, we stress test our algorithm by gradually decreasing the performance of several base-classifiers, adding Gaussian noise to their scores. From Fig. 7(a) it can be seen that both regularized MMSE fusion and MFoM are more sensitive to such noise. On the other hand, MoE and our approach both perform classifier selection, and are thus more robust to bad sources.

We conduct a second stress test to illustrate our robustness to label imbalance. Let r be the ratio of negative samples versus positive samples for model training.



(a) (b) (c)

Fig. 6. Sample DET curves on selected event categories. (a) Flash mob gathering is the category with best performance, and (c) Grooming an animal is the worst.



(a) (b)

Fig. 7. Stress tests performed on the *parkour* event. With respect to both (a) decreasing base-level classification accuracy and (b) unbalanced labels, our approach out-performs other score fusion techniques.

We keep the positive samples fixed and re-sample the negative samples (with replacement) to manipulate the negative sample ratio from $r = 1$ to $r = 15$ (Fig. 7b). *MMSE* fusion performs poorly when there are insufficient negative samples, and only improves gradually. Though comparable to our method in terms of robustness to noise, MoE has poor performance at low r, and lags our method at higher r. On the other hand, MFoM and our approach show stable performances over a range of r, with our method consistently performing better.

4 Conclusion and Future Work

We introduce a local expert forest model for score fusion that exploits changes in the relative performance of a bank of base classifiers by partitioning the score space and learning local weight sets which optimally combine scores in the resulting regions. We demonstrate this method on the TRECVID MED task, fusing scores produced by 5 different base classifiers in order to detect 15 complex events from an archive of more than 1000 hours of video. Our model shows a significant performance advantage over other fusion methods, in terms of average AUC over 300 trial runs. Since TRECVID performance, generally speaking,

correlates strongly with the number of low level features (and base classifiers) fused, we plan additional experiments with more fusion inputs.

To date, our fusion weights have been determined based on the relative performance of the base classifiers over regions of the score space, without taking into consideration the properties of individual clips. We have found that, in addition to the output probability, the performance of the base classifiers correlates with video metadata. As an example, the performance of the base classifier using HOG3d features has poorer performance on highly-compressed videos, as compared to those with relatively less compression. In order to capture such performance dependencies in our fusion model, we will investigate the use of clip-level metadata in weighting, *e.g.* reducing the weight given to the HOG3d classifier probability for highly-compressed video. This may be achieved, for instance, by expanding the score space to include dimensions representing relevant metadata measures and applying the existing partitioning method.

Acknowledgements. We thank our teammates for providing base classifier scores: Byungki Byun, Ilseo Kim, Ben Miller, Greg Mori, Sangmin Oh, Amitha Perera, and Arash Vahdat.

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. 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. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: ICPR (2004)
2. Wong, S., Kim, T., Cipolla, R.: Learning motion categories using both semantics and structural information. In: CVPR (2007)
3. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
4. Yuan, J., Liu, Z., Wu, Y.: Discriminative subvolume search for efficient action detection. In: CVPR (2009)
5. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: CVPR (2009)
6. Over, P., Awad, G., Fiscus, J., Antonishek, B., Smeaton, A., Kraaij, W., Quenot, G.: Trecvid 2010 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In: Proceedings of TRECVID 2010, NIST, USA (2011)
7. Gong, S., Xiang, T.: Recognition of group activities using dynamic probabilistic networks. In: ICCV (2003)
8. Yu, G., Goussies, N.A., Yuan, J., Liu, Z.: Fast action detection via discriminative random forest voting and top-k subvolume search. Multimedia 13, 507–517 (2011)

9. Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M.: The det curve in assessment of detection task performance. In: European Conf. on Speech Communication and Technology (1997)
10. Dass, S., Nandakumar, K., Jain, A.: A Principled Approach to Score Level Fusion in Multimodal Biometric Systems. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 1049–1058. Springer, Heidelberg (2005)
11. Yin, Z., Porikli, F., Collins, R.: Likelihood map fusion for visual object tracking. In: BMVC (2008)
12. Mittal, A., Zisserman, A., Torr, P.: Hand detection using multiple proposals. In: BMVC (2011)
13. Ma, C., Lee, C.: An efficient gradient computation approach to discriminative fusion optimization in semantic concept detection. In: ICPR (2008)
14. Gao, S., Wu, W., Lee, C., Chua, T.S.: A maximal figure-of-merit (mfom)-learning approach to robust classifier design for text categorization. ACM Trans. on Information Systems 42, 145–175 (2006)
15. Tseng, B., Lin, C., Naphade, M., Natsev, A., Smith, J.: Normalized classifier fusion for semantic visual concept detection. In: ICIP (2003)
16. Bach, F., Heckerman, D., Horvitz, E.: On the path to an ideal roc curve: considering cost asymmetry in learning classifiers. In: Artificial Intelligence and Statistics (2005)
17. Gao, S., Lee, C., Lim, J.: An ensemble classifier learning approach to roc optimization. In: ICPR (2006)
18. Jordan, M.I.: Hierarchical mixtures of experts and the em algorithm. Neural Computation 6, 181–214 (1994)
19. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 1.21 (2011), `http://cvxr.com/cvx`
20. Klaser, A., Marszalek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients. In: BMVC (2008)
21. Li, L., Su, H., Xing, E., Li, F.: Object bank: A high-level image representation for scene classification & semantic feature sparsification. In: Neural Information Processing Systems (NIPS), Vancouver, Canada (2010)
22. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV 42, 145–175 (2001)
23. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
24. Herschtal, A., Raskutti, B.: Optimizing area under the roc curve using gradient descent. In: ICML (2004)