# Learning Non-linear Calibration for Score Fusion with Applications to Image and Video Classification

Tianyang Ma
Temple University

Sangmin Oh
Kitware Inc.

Amitha Perera
Kitware Inc.

Longin Jan Latecki
Temple University

## Abstract

*Image and video classification is a challenging task, particularly for complex real-world data. Recent work indicates that using multiple features can improve classification significantly, and that score fusion is effective. In this work, we propose a robust score fusion approach which learns non-linear score calibrations for multiple base classifier scores. Through calibration, original base classifiers scores are adjusted to reflect their true intrinsic accuracy and confidence, relative to the other base classifiers, in such a way that calibrated scores can be simply added to yield accurate fusion results. Our approach provides a unified approach to jointly solve score normalization and fusion classifier learning. The learning problem is solved within a max-margin framework to globally optimize performance metric on the training set. Experiments demonstrate the strength and robustness of the proposed method.*

## 1. Introduction

The goal of image and video classification is typically detecting or retrieving images or videos with particular content such as object classes (e.g., flower) or complex events (e.g., flash mob). It is particularly challenging for real-world datasets which exhibit significant visual clutter, small inter-class variations, and large intra-class variations. To deal with such challenges, multiple features are frequently considered and fused to improve classification. Many algorithms have been proposed for combining multiple features, and their effectiveness have been proved on various visual classification tasks [1, 4, 12, 6, 30].

**Acknowledgement**: We thank Arash Vahdat, Greg Mori, and Scott McCloskey for sharing features. 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 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.

To fuse multiple features, there are two main strategies. Early fusion [1, 6] directly learns a single classifier across multiple features, and outputs a single confidence value on a test sample. In early fusion (also called *feature fusion*), learning is accomplished mostly by concatenating features, or using multiple kernel combinations (e.g., [1]). In contrast, late fusion trains multiple *base classifiers* independently on different features, and then combines their output [4, 10, 16, 9, 28]. The most common type of late fusion is *score fusion*, where the scores of the base classifiers are combined. (The alternative is *decision fusion*, where the binary decisions of the base classifiers are combined.) Compared to early fusion, late fusion approaches have a number of advantages: (1) they are less memory intensive and more parallelizable because entire features need not be loaded onto memory at the same time; (2) late fusion provides a practical framework to combine additional off-the-shelf classifiers which can not be easily incorporated into early fusion; and (3) late fusion approaches are of superior or comparable accuracy to early fusion [4, 27].

However, one major challenge with score fusion is that scores generated by different base classifiers may exhibit different ranges and even substantially different distributions. This variation in the score profiles makes it suboptimal to blindly apply any fixed score fusion rule (e.g., sum or product from [10]) on the raw base classifier scores.

Score normalization schemes [18, 5, 24] have been widely studied to partly address this challenge. A common characteristic of these schemes is that each set of base classifier scores is independently normalized into a $[0, 1]$ range using *a priori* assumptions, and then are summed to produce fusion scores. Some normalization schemes include Platt scaling [18] (to account for the different behaviors of, e.g. SVM and boosted trees vs neural networks), and other scaling techniques using different assumptions on the score distribution model, including Gaussian [5], sigmoid [5], and Weibull [24]. Two key limitations of these schemes are that fusion is sensitive to the accuracy of the assumed score distribution model, which is difficult to judge in practice for black-box, off-the-shelf classifiers; and that each base classifier is treated equally even though individual accuracy can

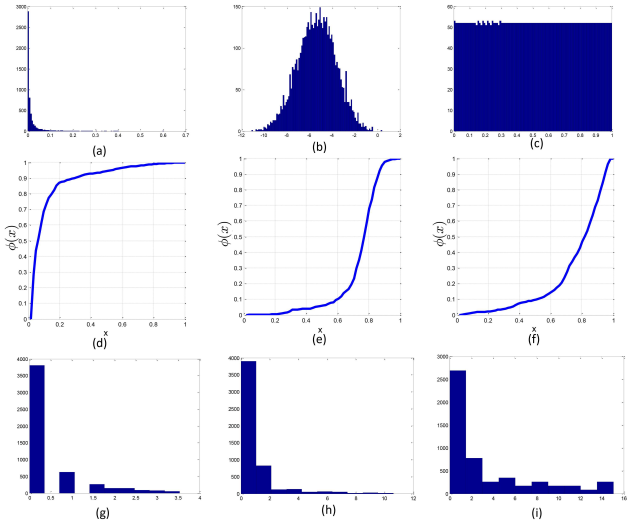



Figure 1. First row: distributions of three different score types by a same classifier on the same dataset: (a) SVM posterior probability using Platt scaling [18] on margins (b) SVM margins (c) Rank scores (uniform in [0,1]). Second row: corresponding calibration functions learned by our method where $x$ and $y$ axes correspond to original and calibrated scores. Third row: the distributions of the calibrated scores, which are clearly comparable.

vary substantially.

In this work, we propose a robust score fusion approach which learns *non-linear score calibration functions* for multiple base classifiers. By calibration, we mean the process of adjusting raw base classifiers scores to reflect their true intrinsic accuracy and confidence, in such a way that calibrated scores can be simply added to yield accurate fusion results. The key innovation compared to existing score normalization approaches is that supervised learning is used to learn the non-linear calibration functions $\phi(\cdot)$ which translate different types of scores to a *common score space* (CSS) where they are accurately calibrated. An important property of our approach is that score calibration functions are learned simultaneously in a max-margin framework to globally optimize a performance metric on the training set.

A key observation driving our approach is that different classifiers, or even a same classifier applied to the same data, often yield very different score distributions depending on how scores are processed. For example, the first row of Fig. 1 shows three different score distributions generated by the same SVM classifier on the same data, but with three different ways to represent scores (all of which are quite common). Accordingly, the *intrinsic* score distribution of all three examples are actually same. Existing approaches to address this issue are tailored for very specific classes of score distributions, e.g., [22, 5, 24], and cannot be applied automatically. As the number of base classifiers (and consequently the number of score representation schemes) increases, manually defining proper calibration functions for each becomes daunting; automatic methods become necessary. Linear methods [28, 9] provide only sub-optimal solutions; for example, no uniform scaling can accurately align the three score distributions in Fig. 1. Our approach addresses all these issues by automatically learning non-linear calibration functions from the training data without making any assumptions about the score distributions. For example, the learned calibrations are illustrated in the second row of Fig. 1, and the third row shows clearly comparable resulting score distributions after applying calibrations, despite their original difference.

Another key innovation in our approach is that all the calibration functions are learned simultaneously to ensure optimality for fusion. Most existing approaches [5, 24] calibrate each base classifier scores independently, and do not guarantee fusion optimality. While there exist approaches [16, 26] which learn fusion functions jointly across multiple base classifier scores, they typically learn a mixture of localized fusion functions in the multi-dimensional score space, resulting in models which are opaque to understanding the contribution of each classifier. On the other hand, our method explicitly learns transparent calibration functions which can aid the users with useful information on fusion, while also guaranteeing the optimality of fusion. In terms of the optimality measure, we use the area under an ROC curve (AUC) to measure the overall performance, which is equal to Wilcoxon-Mann-Whitney ranking [3]:

$$\text{AUC} = \sum_{i=1}^{N^+} \sum_{j=1}^{N^-} \mathbb{1}(s_j < s_i)/(N^+N^-) \qquad (1)$$

where $s_i$ are the scores of $N^+$ positive samples, and $s_j$ are the scores of $N^-$ negative samples. In particular, we show (in Sec. 3) that the supervised learning of optimal non-linear calibration functions can be formulated as solving a max-margin optimization problem, which can be efficiently solved by a modified Newton method.

The proposed method has been evaluated through extensive experiments (see Sec. 4), including both large-scale video classification, as well as image classification. In addition, for the video classification task, we have also evaluated the robustness of our method to changes in the base classifier score distributions. On all experiments, the proposed method showcased notable performance compared to state-of-the-art methods, which clearly demonstrates its benefits.

## 2. Related Work

There exist numerous late fusion methods, which can be mainly grouped into four categories. The first category can be understood as *blind* fusion where fixed rules are applied regardless of actual base classifier score distributions, prior to simple score summation. As one of the pioneering works, multiple classifier combination rules are studied in [10], where extensive experiments showed that Sum and Product

are top two best performing methods. In recent work [27], geometric mean is reported to be highly effective despite its simplicity. Both product and geometric mean can be still understood to belong to the first category where a logarithm transformation is used prior to summation. While simplicity is the main advantage of these methods, as reported in [27], more sophisticated fusion methods can outperform them at the expense of additional computation in many occasions.

The second category of late fusion methods [5, 24, 18] are formulated within a score normalization framework, where particular assumptions are made on score distributions, and used to align base classifier scores. However, most of these methods require the normalization transforms to be determined manually, based on expert knowledge. Hence, it is difficult to build a robust fusion model from a large set of black-box classifiers. In addition, fusion learning needs to be conducted separately from score normalization, which does not guarantee optimality in fusion.

The third category aims to learn a linear weighting of base classifiers where score normalization is ignored. (Or, more precisely, the normalization scheme is a simple scaling by the learned weight.) In [28, 9], weights are learned by minimizing different target error metrics with different regularizations. In [17], a linear dependency between features is proposed to address the independent assumption issue in fusion process. Compared to these linear models, our method can learn weights and non-linear calibration functions jointly, allowing more flexibility and providing for scores with very different underlying distributions.

Finally, the fourth late fusion category looks into building a mixture of fusion models which are optimized locally across multi-dimensional score space. In Local Expert Forrest (LEF) [16], multiple local expert fusion classifiers are built across score partitions, and final fused scores are computed as the average of outputs from many local experts. Smith et al. [26] treat the confidence scores from multiple models as a feature vector, and then learn a classifier for different classes using a sample-based approach. Lan et al. [13] introduced a double fusion technique, which unifies both feature level fusion and score level fusion. While these approaches work well in practice, the resulting models are opaque and do not provide transparent insight into the learned fusion model, which is crucial to understanding the directions for further improving the classification system.

Our method is related to SVM$^{perf}$ [7] in that multivariate SVMs is used to optimize the ROC area. However, both our goal and problem formulation are very different from [7], which is not related to late score fusion.

## 3. Method

We are given a set of base classifiers $B_k$, $k = 1, \ldots, K$, each with a possibly different range of score values $R_k$. The classifiers share the common property that a higher score for a given data sample implies a higher likelihood of belonging to the target class. However, their score values are not necessarily comparable, therefore, they cannot be directly combined for fusion.

We associate with each classifier a function $\phi_k : R_k \rightarrow C$, called a *calibration function*, where $C$ denotes a (new) common score space. The sets $R_k$ and $C$ are subsets of real numbers. For each data sample $j$, given its base classifier scores $s_j^k$, $k = 1, \ldots, K$, the combined score is

$$s_j = \sum_{k=1}^{K} \phi_k(s_j^k) \tag{2}$$

Our goal is to simultaneously learn all calibration functions $\phi_k$, $k = 1, \ldots, K$, such that AUC of the combined score in (1) is maximized over the training data.

A unique property of the proposed approach is that functions $\phi_k$ can be non-linear. (As far as we are aware, ours is the first approach with this property.) In order to provide meaningful structure in the calibration functions and prevent overfitting to the training data, we assume that functions $\phi_k$ are non-decreasing, i.e., we require that $x < x'$ implies $\phi_k(x) \leq \phi_k(x')$. Hence scores of each classifier $B_k$ can be stretched non-uniformly, but with the restriction that their order cannot be swapped.

Given the scores of the training samples generated by a classifier $B_k$, we first evenly discretize the score values into $M$ bins with the bin boundaries, which we call *nodes*, denoted by $\eta_0^k < \eta_1^k < \cdots < \eta_M^k$. We fully define $\phi_k$ by its values at the node points, $\phi_k(\eta_0^k), \ldots, \phi_k(\eta_M^k)$, by interpolating between the node points. We use linear interpolation: the calibration for a score value $x$ obtained by $B_k$ is

$$\phi_k(x) = \phi_k(\eta_n^k) + \frac{x - \eta_n^k}{\eta_{n+1}^k - \eta_n^k}(\phi_k(\eta_{n+1}^k) - \phi_k(\eta_n^k)) \tag{3}$$

if $x \in [\eta_n^k, \eta_{n+1}^k]$; $\phi_k(x) = \phi_k(\eta_0^k)$ if $x < \eta_0^k$; or $\phi_k(x) = \phi_k(\eta_M^k)$ if $x > \eta_M^k$. The intuition behind the formulation of the calibration functions is illustrated in Fig. 2.

Observe that since our fusion process is to simply sum the calibrated scores (Eq. (2)), a constant offset in each calibration function does not make a material difference. In

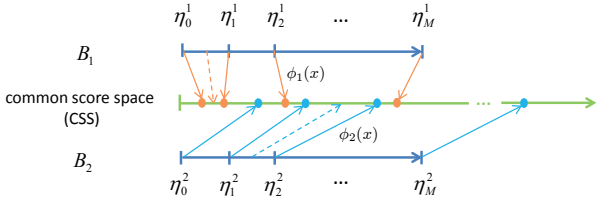



Figure 2. An illustration of our non-linear calibration functions. Two base classifiers $B_1$ and $B_2$ are calibrated into common score space based on the learned calibration values at nodes, where $B_2$ is (learned as being) more accurate. Calibration on nodes are shown as solid lines, while interpolated values are shown as dashed lines

particular, defining a new fusion score $s'_j = s_j - \sum_k \phi_k(\eta_0^k)$ does not make a material difference. Therefore, without loss of generality, we set $\phi_k(\eta_0^k) = 0$ for every $k$.

Given an auxiliary variable $\mathbf{w}^k = (w_1^k, \ldots, w_M^k)$ defined as $w_n^k = \phi_k(\eta_n^k) - \phi_k(\eta_{n-1}^k)$ for $n = 1, \ldots, M$, Eq. (3) can be written in the equivalent form

$$\phi_k(x) = \sum_{i=1}^{n} w_i^k + \frac{x - \eta_n^k}{\eta_{n+1}^k - \eta_n^k} w_{n+1}^k \tag{4}$$

if $x$ belongs to bin $[\eta_n^k, \eta_{n+1}^k]$. We also observe that for $i = 1, \ldots, M$, $w_i^k \geq 0$, because the functions $\phi_k$ are non-decreasing by our assumption. Intuitively, each value $w_i^k$ corresponds to the contribution of that bin towards fusion.

A key idea of our approach is that learning of the non-linear score calibration functions $\phi_k$ is expressed as learning the vectors $\mathbf{w}^k$, which we learn in a maximum-margin framework. For this, we first concatenate vectors $\mathbf{w}^k$ to a single vector $\mathbf{w} = (\mathbf{w}^1, \ldots, \mathbf{w}^K)$ of length $K \times M$, since we have $K$ base classifiers with scores discretized into $M$ bins each.

Given the node values $\{\eta_0^k, \ldots, \eta_M^k\}$ of base classifier $B_k$ and given a score value $x$ belonging to bin $[\eta_n^k, \eta_{n+1}^k]$, we represent it as an indicator vector $\mathbf{x}^k = (x_1^k, \ldots, x_M^k)$ such that $\phi_k(x) = \mathbf{w}^k \cdot \mathbf{x}^k$. In particular, we define

$$x_i^k = \begin{cases} 1, & 1 \leq i \leq n \\ \dfrac{x - \eta_n^k}{\eta_{n+1}^k - \eta_n^k}, & i = n+1 \\ 0, & n+1 < i \leq M \end{cases} \tag{5}$$

Given a data sample $j$ and its original classifier scores $s_j^1, \ldots, s_j^K$, by setting $x = s_j^k$, we obtain vector $\mathbf{x}_j^k$ defined by (5) for $k = 1, \ldots, K$. We then concatenate these vectors into a single vector $\mathbf{x}_j = (\mathbf{x}_j^1, \ldots, \mathbf{x}_j^K)$ of length $K \times M$. The combined fusion score $s_j$ in Eq. (2) is then given by

$$s_j = \sum_{k=1}^{K} \phi_k(s_j^k) = \mathbf{w} \cdot \mathbf{x_j} \tag{6}$$

Consequently, observing that AUC in Eq. (1) is proportional to $\sum_{i=1}^{N^+} \sum_{j=1}^{N^-} \mathbb{1}(\mathbf{w} \cdot (\mathbf{x_i} - \mathbf{x_j}))$, we can maximize AUC by solving the following max-margin minimization problem:

$$\min_{\mathbf{w}} \ \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{N^+} \sum_{j=1}^{N^-} \mathbb{1}(\mathbf{w} \cdot (\mathbf{x_j} - \mathbf{x_i})) \quad \text{s.t.} \quad w_n \geq 0 \tag{7}$$

in which $C$ is the parameter to balance the L2 regularization term on $\mathbf{w}$ and the loss function. Since (7) is difficult to optimize as a hard margin loss (or 0-1 loss), we need to relax it. We consider two relaxation options, L1 hinge loss or L2 hinge loss, given as

$$\min_{\mathbf{w}} \ \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{N^+} \sum_{j=1}^{N^-} \max(0, \mathbf{w} \cdot (\mathbf{x_j} - \mathbf{x_i}))^p \quad \text{s.t.} \quad w_n \geq 0, \tag{8}$$

where power $p = 1$ yields L1 and $p = 2$ yields L2 hinge loss. The L1 hinge loss can be solved by many efficient solvers, such as Pegasos [25]. To enforce the non-negative constraints on $\mathbf{w}$, an additional projection to the constraint set need to be performed after every gradient step [12].

Compared to L1-loss, L2 hinge loss gives more penalty to large violation of the margin. This is a desirable property in our setting, since the penalty for swapping the order of positive and negative examples with a large difference in calibrated scores is much larger. Our experimental results also confirm that L2 hinge loss delivers better performance[1].

Our algorithm to solve the L2 version of the minimization in Eq. (8) is outlined in Alg. 1. Key items about it are:

- it is based on a Modified Newton Method [15, 8];
- with $I^t$ fixed for the current iteration as in line 2, Eq. (8) becomes

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C \sum_{(i,j) \in I^t} (\mathbf{w} \cdot \mathbf{z_{ij}})^2 \quad \text{s.t. } w_n \geq 0; \tag{9}$$

  without the non-negativity constraint this is a simple regularized least squared problem, and can be solved by the linear system in line 4;
- we solve the system in line 4 using a conjugate graident method following [8];
- line 5 projects the solution to the non-negative feasible set;

**Algorithm 1** Modified Newton Method for L2-SVM with Non-Negative Constraints

**Input:** $\mathbf{w}^0$ and convergence threshold $\epsilon$

1: **repeat**
2: $I^t \leftarrow \{(i,j) \,|\, \mathbf{w}^t \cdot (\mathbf{x}_j - \mathbf{x}_i) > 0\}$
3: $\mathbf{z}_{ij} \leftarrow \mathbf{x}_j - \mathbf{x}_i \quad \forall (i,j) \in I^t$
4: $\bar{\mathbf{w}} \leftarrow \mathbf{w} \,|\, \sum_{(i,j) \in I^t} (\mathbf{I} + 2C\mathbf{z}_{ij}^\top \mathbf{z}_{ij})\mathbf{w} = 0$
5: $\bar{\mathbf{w}} \leftarrow \left(\max(0, \bar{w}_1^t), \ldots, \max(0, \bar{w}_{K \times M}^t)\right)$
6: $\alpha_t \leftarrow \arg\min_{0 \leq \alpha \leq 1} f(\mathbf{w}^t + \alpha(\bar{\mathbf{w}} - \mathbf{w}^t))$
7: $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t + \alpha(\bar{\mathbf{w}} - \mathbf{w}^t)$
8: **until** $|\mathbf{w}^{t+1} - \mathbf{w}^t| < \epsilon$

**Output:** $\mathbf{w}^{t+1}$

[1] For brevity, these results are not documented in this paper.

- line 6 is a line search with $f$ being the error function in Eq. (9); and
- the iterations stop when the change in $\mathbf{w}$ is less than a threshold $\epsilon$.

While our model has great flexibility, it also has the potential overfit to the training data, so that the generalization performance may degrade. We employ both early stopping and bagging strategy to counter overfitting [23]. When no separate validation dataset is available, we first divide training data into $Q$ folds. The training is performed on $Q-1$ folds, with 1 fold left out as the validation set. In each iteration of Alg. 1, we also compute the target function value (AUC) on the samples in validation set. The $\mathbf{w}^t$ which provides the lowest target function value on the validation set is chosen as the output for that fold. This is performed $Q$ times, and the average of the $Q$ $\mathbf{w}^t$ values is taken as the final solution.

Solving Eq. (8) yields the optimal $\mathbf{w}^*$. Given a sample for testing, same as we did for training samples, we first derive its indicator vector $\mathbf{x}$ according to Eq. (5), and then simply compute $\mathbf{w}^* \cdot \mathbf{x}$ as its final fused score.

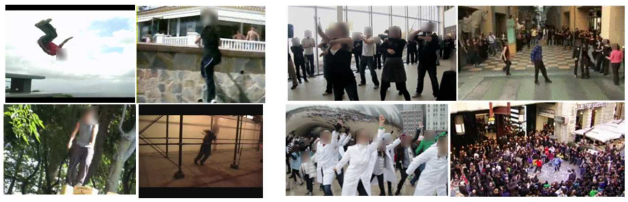
## 4. Experimental Results

In this section, we evaluate our non-linear calibration fusion method on various visual classification tasks, such as object categorization and multimedia event detection.

We compare our results with both learning-free approaches and learning-based approaches, including: averaging, geometric mean, linear SVM, RBF network, MMSE [32] and MFoM [9]. Both MMSE (Minimum Mean Square Error) and MFoM (Maximal Figure-of-Merit) are linear fusion methods, and the optimal weight for each base classifier are learned. We also compare with a very recent fusion method Local Expert Forrest (LEF) [16]. For linear SVM, we concatenate base classifier scores of each sample to a single vector, and use LibSVM [2] for training. For RBF network, we used a Gaussian kernel, and the kernel width is optimized through validation set.

We use the one-vs-all SVM as the model for generating the confidence scores. And we use AUC, i.e., Eq (1), as the main metric to evaluate the performance. For datasets consisting of multiple categories, we calculate the average AUC across all the categories as the final evaluation metric.

Sec. 4.1 describes experiments on a challenging large scale video classification task on the TRECVID Multimedia Event Detection (MED) 2011 [21] dataset. In addition, we also evaluate the robustness of the fusion methods in Sec. 4.1.1. In particular, we evaluate their performance under changes in the distribution of base classifier scores. Sec. 4.2 describes the evaluation of our method on an image classification task on the Oxford Flower 17 [19] dataset. In all of our experiments, we used $M = 60$ bins.



(a) Parkour (b) Flash mob gathering

Figure 3. Example images showcasing two MED11 events

### 4.1. Large-Scale Video Retrieval on MED 2011

In this section, we evaluate our method on a challenging task, multimedia event detection (TRECVID MED 2011 [21]). The goal is to detect complex events from video clips in a very large multimedia archive (1000+ hour collection of about 34000 clips). The videos are unconstrained in terms of camera motion, background clutter and human editing (e.g., shot stitching). For illustration, snapshots of samples belonging to two different event classes are shown in Fig. 3, where a large intra-class variation can be observed. Such visual variability makes the performance of any single feature classifier limited, and motivates the use of fusion based on multiple features to boost classifier accuracy. The limited performance of base classifiers, as well as the variations in the performance of base classifiers, also brings challenges to late fusion methods.

We used 6 base classifiers, each of which predicts event probability based on a different multimedia feature. Both visual information and audio information are used in building our system. For visual information, we used high-level features from Object Bank [14] to capture the relationship between target events and objects, where it computes the response of a set of 177 object detectors, such as human and tree, etc. Object Bank is first run on each frame, then, two different clip-level representations are extracted by both max-pooling and average-pooling; this results in two different classifiers based on Object Bank. We also used built classifiers using static low-level features; in particular, color SIFT and Transformed Color Histogram [29]. To capture dynamic motion information, we used a classifier based on 3D histograms of oriented gradients (HoG3D) [11]. For audio information, we used Mel-Frequency Cepstral Coefficients (MFCCs).

For Object Bank features, a linear SVM is used to perform the classification. For all the other features, we built a bag-of-word histogram (codebook size is 4096) as the clip-level representation, and used a SVM with Negative Geodesic Distance (NGD) kernel [33] for classification. We used the NGD kernel because it showed better performance in our application compared to some other widely-used kernels, such as histogram intersection kernel.

Our experimental results (Table 1) show that our method achieves the best performance for all 10 event categories consistently. The numeric improvements in AUC by our

| Events | Fusion methods | | | | | | |
|---|---|---|---|---|---|---|---|
| | Avg | SVM | RBF | MFoM | LEF | GeoMean | Ours |
| Birthday party | .9443 | .9463 | .9420 | .9445 | .9461 | .9454 | **.9492** |
| Changing a vehicle tire | .9108 | .9113 | .8989 | .9027 | .9099 | .9146 | **.9162** |
| Flash mob gathering | .9818 | .9833 | .9825 | .9821 | .9836 | .9831 | **.9846** |
| Getting a vehicle unstuck | .9389 | .9374 | .9311 | .9370 | .9388 | .9358 | **.9429** |
| Grooming an animal | .9008 | .9002 | .8860 | .8998 | .9029 | .8950 | **.9077** |
| Making a sandwich | .9115 | .9125 | .9026 | .9117 | .9130 | .9121 | **.9163** |
| Parade | .9668 | .9679 | .9642 | .9662 | .9675 | .9683 | **.9716** |
| Parkour | .9480 | .9495 | .9442 | .9476 | .9492 | .9497 | **.9505** |
| Repairing an appliance | .9691 | .9699 | .9551 | .9664 | .9521 | .9571 | **.9712** |
| Working on a sewing project | .8853 | .8870 | .8761 | .8884 | .8865 | .8877 | **.8953** |

Table 1. Mean AUC results on MED11 test set [21]. Best results are in bold.

method is slight, although the fact that the proposed method showcases consistent performance gain across all events indicate the benefit of the proposed method. In addition, it is worth noting that fairly small quantitative difference in AUC frequently indicates meaningful difference in accuracy level, much more so than other metrics such as APs, e.g., the AUC by a random method is still as high as 0.5.

In detail, to obtain these results, for each event category, we split the training data into two halves. We trained the base classifiers with the first half of the data and trained the fusion models on the second half. We randomly generated 10 such splits, and the average AUC over 10 runs are reported in Table 1.

In addition to the improved accuracy, our approach provides qualitative insights into fusion classifiers, which is an additional benefit. One property is that we can use the range of each calibration function $\phi_k(s^k)$ as a good indicator of the importance of each base classifier. Since calibrated score values of each classifier start at zero, the range of calibrated scores of classifier $k$ is equal to the maximum value of calibration function $\phi_k(s^k)$, which is $\sum_{i=1}^{M} w_i^k$ according to Eq. (4). The visualization of the ranges of calibrated maximum values of the six base classifiers for two different event categories are shown in Fig 4. For the event category *Birthday Party*, which is filled with people and party objects along with birthday songs, it can be observed that both Object Bank (max) and MFCCs (audio) play the most important roles. On the other hand, for the motion-heavy event category *Parkour*, we can see that dynamic motion feature HoG3D is the most important in making the final decision. Furthermore, Fig. 5 shows the visualization of normalized non-linear calibrations $\phi(x)$ across different base classifiers, learned for the event *Birthday Party* (also in Fig. 4 (a)). The diversity in their nonlinear shape showcases the advantage that imposing minimal assumption on score distributions actually allows our model to flexibly adapt to widely different score distributions, playing a key role in improving accuracy.

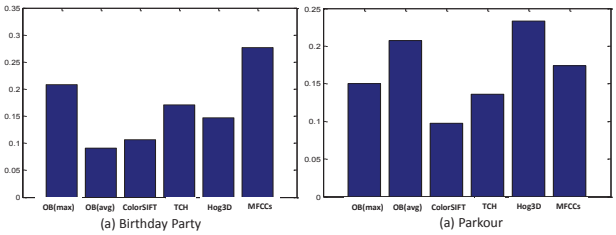



Figure 4. Examples of range of calibrated scores $\phi_k(s^k)$ for two MED11 event classes, indicating the importance of each base classifier. Bar 1 to 6 are: Object Bank (max-pooling), Object Bank(average-pooling), ColorSIFT, TCH, Hog3D, MFCCs.

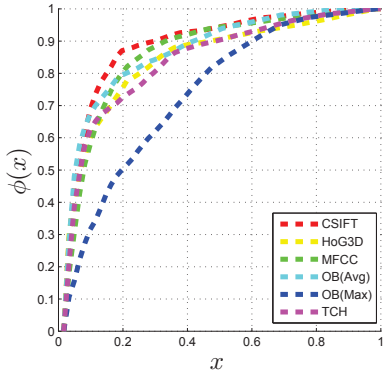



Figure 5. Learned calibration functions for SVM scores obtained for different video features. $\phi(x)$ are normalized between 0 and 1 for a better visualization when comparing their shapes.

#### 4.1.1 Robustness Test

In this section, we evaluate the robustness of fusion methods to changes in the distribution of the scores output by the base classifiers. Many fusion methods assume that the scores of the base classifiers are the posterior probabilities that the samples belong to the positive class. However, this assumption is generally not true [31], and when this assumption does not hold, the fusion performance may degrade. For instance, the scores of maximum margin methods such as SVMs or boosted trees are originally the distances to the decision boundary, and they have to be passed

| | probability | margin | rank |
|---|---|---|---|
| Avg | .93173 | .93531 | .93846 |
| SVM | .93653 | .90515 | .93353 |
| RBF | .92827 | .89859 | .84365 |
| MFoM | .93464 | .93333 | .93647 |
| LEF | .93496 | .92791 | .93177 |
| GeoMean | .93488 | .93253 | .93038 |
| Ours | **.94055** | **.94142** | **.94193** |

Table 2. Mean AUC results on MED11 test set with 3 different output profiles. Column 2: SVMs posterior probability. Column 3: SVMs margin. Column 4: Scores derived from ranking.

through a sigmoid function to obtain the posterior probability [18]. Sometimes, we can only obtain the ranking of samples, which is the case if a rank system is used as a single model and treated as black box. Therefore, it is crucial for fusion methods to have the ability to adapt automatically to such changes in the output profiles of base classifiers. This ability also eliminates the significant manual effort of categorizing the output profiles of base classifiers. We evaluate 3 different types of output profiles: (1) posterior probability, which were used in Sec. 4.1; (2) margin values output by SVMs; and (3) scores derived by the ranking of samples. For the latter, the samples are sorted from most negative to most postive, and the rank score is the ratio between its rank and the total number of samples. Therefore, the scores form a uniform distribution. The distributions of three different output profiles are shown in the first row of Fig. 1. We still use the TRECVID MED11 dataset with same base classifiers where the only changes are to the base classifier output profiles.

Fig. 1 shows that, when the distributions of scores change, our method adaptively learns different shapes of calibration functions. For the distribution in Fig. 1(a), our calibration function in (d) has a steep slope in the small value range. For the distribution in Fig. 1(b), the steep slope occurs in the higher value range, Fig. 1(e). Fig. 1(f) shows a more linear shape suitable for uniform distribution.

For quantitative evaluation, we used the same training and testing splitting as in Sec. 4.1, but generated the three different output profiles for each single base classifier. To evaluate the overall performance, we computed the mean AUC over all event categories. The performance is reported in Table 2. The proposed method consistently yields the highest AUC in all of three settings with small variance across different profile settings, which demonstrates its robustness to changes in base classifier score distributions.

### 4.2. Image Retrieval on Oxford Flower 17

In this section, we report the performance of the proposed approach on Oxford Flower 17 dataset. The Oxford Flower 17 dataset contains 17 different types of flowers with 80 images per category. There are 680 training images ($17 \times 40$ images), 340 validation images ($17 \times 20$ images) and 340 test images ($17 \times 20$ images). Seven different types of features including shape, color, texture, HSV, HoG, SIFT internal, and SIFT boundary, are extracted in [20]. The author provides the pre-computed distance matrices for the three splits. More details about the features and kernels can be found in [20]. We follow the experiment settings in [17]. 5-fold cross-validation (CV) in the training set is used to select the best classifier for each feature, and the CV outputs are used for the fusion training.

| Single Feature | | Fused | |
|---|---|---|---|
| feature | mean AUC | method | mean AUC |
| Color | .931±.056 | Avg | .983±.016 |
| Shape | .955±.041 | GeoMean | .987±.014 |
| Texture | .918±.063 | SVM | .982±.020 |
| HSV | .931±.053 | RBF | .978±.021 |
| HoG | .919±.061 | MFoM | .982±.018 |
| SIFTint | .956±.049 | LEF | .981±.019 |
| SIFTbdy | .917±.067 | Ours | **.988±.015** |

Table 3. Mean AUC and 3×standard deviation on Oxford Flowers.

The fusion results are shown in Table. 3. The performance of the single feature classifiers is shown on the left, and the performance of seven different fusion methods is reported on the right. It is clear that all fusion methods show some improvement compared to single feature classifiers. This demonstrates that late fusion techniques, which combine the output of multiple singe feature classifiers, can be very effective in building a highly accurate fusion classifier. All methods, including ours, report accuracy higher than 97%, which indicates that the performance on this dataset saturated. In this case, it is difficult to draw statistically meaningful conclusions about the benefits of certain methods in comparison to others. Nonetheless, detailed observations are described below, which highlights that the proposed approach can be considered to be in the group of the state-of-the-arts.

The proposed method achieves comparable performance to all learning-based approaches, including the latest state-of-the-arts [16, 27]. Additionally, we include brief comparisons to two recent fusion methods [31] and [17], which reported performance on the same dataset using different metrics. The method in [31] is based on low rank minimization, where their goal is finding a common ranking from several base classifier rankings and reported mAP on Oxford Flower Dataset. Although our approach optimizes the AUC metric, we achieve mAP of 0.910, compared to 0.898 and 0.917 obtained by RLF and GRLF from [31] respectively, which indicates comparable performance. In [17], they evaluated the performance of a multi-class classification task, in term of classification accuracy. Since our approach is designed for binary classification task, to extend it to multi-class task, we simply normalize the calibrated scores generated by our one-vs-all model between 0

and 1 for each class, and assign an image to the class with the highest score. We obtained a classification accuracy of 86.4%, which is slightly higher than [17] (86.3%).

## 5. Conclusion

In this work, we presented our novel score fusion approach which learns non-linear score calibrations for multiple base classifier scores. Our approach provides a unified solution to jointly solve score normalization and fusion classifier learning. Our extensive experiments demonstrate the strength and robustness of the proposed approach. We believe that the proposed approach will allow fusion systems to scale up further in an automated manner to incorporate large number of features and additional classifiers.

## References


[1] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML*, 2004. 1

[2] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, May 2011. 5

[3] C. Cortes and M. Mohri. Auc optimization vs. error rate minimization. In *NIPS*, 2004. 2

[4] P. V. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009. 1

[5] A. Jain, K. Nandakumar, and A. Ross. Score normalization in multimodal biometric systems. *Pattern Recogn.*, 38(12):2270–2285, Dec. 2005. 1, 2, 3

[6] Y.-G. Jiang, X. Zeng, G. Ye, S. Bhattacharya, D. Ellis, M. Shah, and S.-F. Chang. Combining multiple modalities, contextual concepts, and temporal matching. In *NIST TRECVID Workshop*, 2010. 1

[7] T. Joachims. A support vector method for multivariate performance measures. In *ICML*, pages 377–384, 2005. 3

[8] S. S. Keerthi and D. DeCoste. A modified finite newton method for fast solution of large scale linear svms. *Journal of Machine Learning Research*, 6:341–361, 2005. 4

[9] 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 (3)*, 2012. 1, 2, 3, 5

[10] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *PAMI*, 20:226–239, 1998. 1, 2

[11] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008. 5

[12] A. Kumar, A. Niculescu-Mizil, K. Kavukcuoglu, and H. D. III. A binary classification framework for two stage multiple kernel learning. In *ICML*, 2012. 1, 4

[13] Z.-Z. Lan, L. Bao, S.-I. Yu, W. Liu, and A. G. Hauptmann. Double fusion for multimedia event detection. In *ICME*, 2012. 3

[14] 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 *NIPS*, 2010. 5

[15] C.-J. Lin, R. C. Weng, and S. S. Keerthi. Trust region newton method for logistic regression. *Journal of Machine Learning Research*, 9:627–650, 2008. 4

[16] J. Liu, S. McCloskey, and Y. Liu. Local expert forest of score fusion for video event classification. In *ECCV*, 2012. 1, 2, 3, 5, 7

[17] A. J. Ma and P. C. Yuen. Linear dependency modeling for feature fusion. In *ICCV*, pages 2041–2048, 2011. 3, 7, 8

[18] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *ICML*, 2005. 1, 2, 3, 7

[19] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *CVPR*, 2006. 5

[20] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, pages 722–729, 2008. 7

[21] P. Over, G. Awad, M. Michel, J. Fiscus, B. Antonishek, A. F. Smeaton, W. Kraaij, and G. Quéenot. TRECVID 2011 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID 2011*. NIST, USA, 2011. 5, 6

[22] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999. 2

[23] J. R. Quinlan. Bagging, boosting, and c4.5. In *AAAI*, 1996. 5

[24] W. Scheirer, A. Rocha, R. Micheals, and T. Boult. Robust fusion: extreme value theory for recognition score normalization. In *ECCV*, pages 481–495, 2010. 1, 2, 3

[25] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: primal estimated sub-gradient solver for svm. *Math. Program.*, 127(1):3–30, 2011. 4

[26] J. Smith, M. Naphade, and A. Natsev. Multimedia semantic indexing using model vectors. In *ICME*, 2003. 2, 3

[27] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. S. Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, 2012. 1, 3, 7

[28] O. R. Terrades, E. Valveny, and S. Tabbone. Optimal classifier fusion in a non-bayesian probabilistic framework. *PAMI*, 31(9), Sept. 2009. 1, 2, 3

[29] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *PAMI*, 32(9):1582–1596, 2010. 5

[30] G. Ye, I.-H. Jhuo, D. Liu, Y.-G. Jiang, D. T. Lee, and S.-F. Chang. Joint audio-visual bi-modal codewords for video event detection. In *ICMR*, 2012. 1

[31] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang. Robust late fusion with rank minimization. In *CVPR*, 2012. 6, 7

[32] Z. Yin, F. Porikli, and R. T. Collins. Likelihood map fusion for visual object tracking. In *WACV*, pages 1–7, 2008. 5

[33] D. Zhang, X. Chen, and W. S. Lee. Text classification with kernels on the multinomial manifold. In *SIGIR*, 2005. 5