Learning Sample Specific Weights for Late Fusion

Kuan-Ting Lai, Dong Liu, Shih-Fu Chang, *Fellow, IEEE*, and Ming-Syan Chen, *Fellow, IEEE*

***Abstract*— Late fusion is one of the most effective approaches to enhance recognition accuracy through combining prediction scores of multiple classifiers, each of which is trained by a specific feature or model. The existing methods generally use a fixed fusion weight for one classifier over all samples, and ignore the fact that each classifier may perform better or worse for different subsets of samples. In order to address this issue, we propose a novel sample specific late fusion (SSLF) method. Specifically, we cast late fusion into an information propagation process that diffuses the fusion weights of labeled samples to the individual unlabeled samples, and enforce positive samples to have higher fusion scores than negative samples. Upon this process, the optimal fusion weight for each sample is identified, while positive samples are pushed toward the top at the fusion score rank list to achieve better accuracy. In this paper, two SSLF methods are presented. The first method is ranking SSLF (R-SSLF), which is based on graph Laplacian with RankSVM style constraints. We formulate and solve the problem with a fast gradient projection algorithm; the second method is infinite push SSLF (I-SSLF), which combines graph Laplacian with infinite push constraints. I-SSLF is a l_∞ norm constrained optimization problem and can be solved by an efficient alternating direction method of multipliers method. Extensive experiments on both large-scale image and video data sets demonstrate the effectiveness of our methods. In addition, in order to make our method scalable to support large data sets, the AnchorGraph model is employed to propagate information on a subset of samples (anchor points) and then reconstruct the entire graph to get the weights of all samples. To the best of our knowledge, this is the first method that supports learning of sample specific fusion weights for late fusion.**

***Index Terms*— Image recognition, video recognition, late fusion, infinite push, l_∞ norm.**

I. Introduction

THE idea of "multimodal fusion" has been advocated in the computer vision community during the past decades. The fusion strategies can be classified into early fusion (feature level fusion) or late fusion (decision score level fusion) [1]. Recently late fusion has been proven effective on various applications such as object recognition [2]–[4], biometric analysis [5], affect recognition [6], image retrieval [7] and video event detection [4], [8]–[10]. Given multiple classifiers trained with different low-level features, late fusion attempts to find optimal combination of all classifiers' prediction scores (the prediction scores of each sample are generated by classifiers to indicate the confidences of classifying the sample as positive). Such a fusion method is expected to assign positive samples higher fusion scores than the negative ones to improve the overall performance. During past studies on this topic, late fusion has shown to be capable of boosting the performance of each individual classifier and producing comparable or even better results than early fusion methods [4], [11].

An intuitive way for late fusion is to estimate a fixed weight for each classifier and then sum the weighted prediction scores as the fusion result. This approach assumes all samples share the same weight for a classifier, and hence fails to consider the differences of a classifier's prediction capability on individual sample. In fact, each classifier has different prediction capabilities on different samples. Therefore, instead of using a fixed weight for each classifier, a promising alternative is to estimate the specific fusion weights for each sample to achieve optimal fusion result.

However, discovering the sample specific fusion weights is a challenging task due to the following two issues. First, since there is no label information of the test samples, it is not clear how to derive the specific fusion weights for those unlabeled samples. Second, to get a robust late fusion result, positive samples need to have higher prediction scores (confidence level) than negative samples. Therefore, the proposed method needs to ensure the positive samples having higher final fusion scores during the learning process.

In this paper, we propose to address the above two issues by the Sample Specific Late Fusion (SSLF) method, which learns the optimal sample specific fusion weights from supervision information while directly imposing the positive samples to have the highest fusion scores in the fusion result. Figure 1 illustrates the framework of our proposed method. Suppose we have a classifier score vector $\mathbf{s}_i = [s_i^1, \ldots, s_i^m]^\top$ of a sample, where each s_i^j denotes the prediction score produced by the j-th classifier ($j = 1, \ldots, m$), and m is the total number of classifiers. Our target is to learn an optimal fusion weight vector $\mathbf{w}_i = [w_i^1, \ldots, w_i^m]^\top$ such that the fusion score $f_i(\mathbf{s}_i) = \mathbf{w}_i^\top \mathbf{s}_i$ precisely reflects the confidence of classifying the given sample as positive. Specifically, we define the fusion process as an information propagation procedure that diffuses the fusion weights learned on the individual labeled samples to the unlabeled ones. The propagation is guided by a graph built on low-level features of all samples, which enforces visually similar samples to have similar fusion scores. By incorporating the graph Laplacian in the learning objective, our method offers the capability to infer the fusion weights for the unlabeled samples.

Manuscript received July 26, 2014; revised January 7, 2015; accepted April 1, 2015. Date of publication April 15, 2015; date of current version May 22, 2015. This work was supported by the National Science Council of Taiwan under Contract NSC 101-2917-I-002-021 and Contract MOST103-2221-E-002-286-MY2. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Andrea Cavallaro.

K.-T. Lai and M.-S. Chen are with the Department of Electric Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: ktlai@arbor.ee.ntu.edu.tw; mschen@cc.ee.ntu.edu.tw).

D. Liu and S.-F. Chang are with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: dong@ee.columbia.edu; sfchang@ee.columbia.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2423560

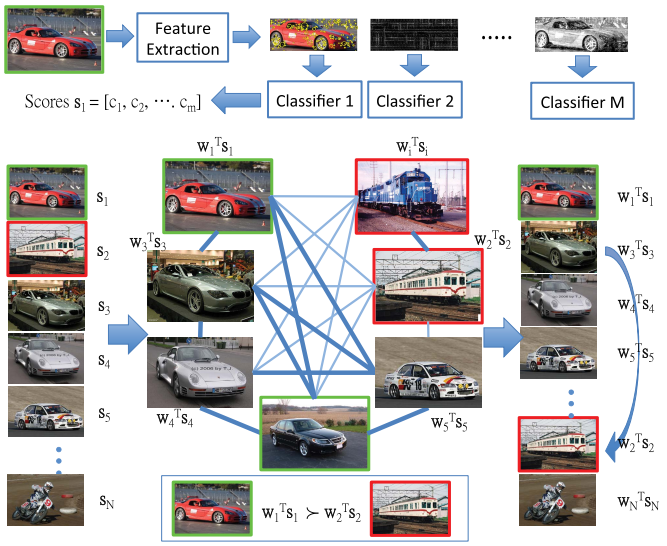


Fig. 1. Illustration of the Sample Specific Late Fusion (SSLF) method. Each image has a prediction score vector $\mathbf{s}_i$. The images with green and red borders are labeled as positives and negatives respectively, or unlabeled otherwise. Our goal is to learn a fusion weight vector $\mathbf{w}_i$ for each sample. Learning is cast into an information propagation process that the fusion weights of the labeled images are diffused to the individual unlabeled ones along a graph built on low-level features. During the propagation, the ranking constraints are applied to ensure the positive samples to have higher fusion scores ($\mathbf{w}_i^\top \mathbf{s}_i$) than the negative samples.

To achieve higher recognition accuracy, we propose to apply ranking constraints to guarantee positive samples having higher fusion scores than negative samples. We extend the work in [12] by developing new algorithm and including support of large scale dataset. In this paper, two ranking approaches are introduced and applied in our SSLF framework. The first approach is Ranking SSLF (R-SSLF), which utilizes RankSVM style constraints [13]. The objective function is formulated and solved by a fast gradient projection method. The second approach uses the ℓ_∞ norm infinite push constraint [14] to maximize the number of positive samples scored higher than the negative ones. In later experiments, we will show that the proposed SSLF methods can achieve significant performance gains over various visual recognition tasks.

Additionally, in order to apply SSLF to large-scale visual dataset, we adopted the AnchorGraph method [15] to scale up our methods linearly with the size of the dataset. The idea of AnchorGraph is to select a subset of samples, which is called anchor points, to effectively represent the entire data set. An efficient algorithm is introduced to construct a graph based on the anchor points. By employing this model, we can propagate information quickly on the much smaller AnchorGraph and learn weights of all samples in a large dataset. This approach enables us to run SSLF on a large number of visual data efficiently without sacrificing the recognition accuracy.

In summary, the main contributions of our paper are:

- Proposing a novel Sample Specific Late Fusion (SSLF) method that adaptively determines the optimal fusion weights for each sample.
- Combining ranking constraints with graph-based regularization to enhance recognition accuracy by enforcing positive samples to have higher fusion scores than negative samples during learning process.
- Adopting AnchorGraph model to efficiently scale up the proposed SSLF methods to large-scale datasets.

The rest of our paper is organized as below: in the next section we first review related works about late fusion; the notations and definitions are introduced in section III; the Ranking-SSLF (R-SSLF) is presented in section IV, while Infinite Push SSLF (I-SSLF) is elaborated in section V. In section VI, we demonstrate the approach for scaling up SSLF through AnchorGraph. The experimental results are shown in section VII, and the conclusions are made in section VIII.

II. Related Work

Many late fusion techniques have been proposed to enhance visual recognition accuracy during the past few years. Nandakumar *et al.* [5] applied the Gaussian mixture model to estimate the distributions of the classifier scores, and fused the prediction scores based on the likelihood ratio test. Terrades *et al.* [3] proposed a supervised late fusion method that aims to minimize the misclassification rates under the ℓ_1 constraints on the score combination weights. However, the above works focused on classifier-level fusion that learns a fixed weight for all prediction scores of a specific classifier. Fixed-weight fusion methods may blindly treat the prediction scores of a classifier as equally important for all samples, and hence cannot determine the optimal fusion weight for each sample.

Liu *et al.* [9] recently proposed a local expert forest model for late fusion, which partitions the score space into local regions and learns the local fusion weights in each region. Nonetheless, the learning can only be performed on the training samples whose label information is provided, and cannot be used to learn the fusion weights on the test samples. Moreover, the partition requires a threshold, which is difficult to predefine. One promising work that attempts to learn sample specific fusion scores is the low rank late fusion proposed by Ye *et al.* [4]. The authors converted the prediction score vectors of multiple classifiers into several pairwise relation matrices, and extracted a shared rank-2 matrix by decomposing each original matrix into a common rank-2 matrix and a sparse residual matrix. Finally, a score vector is extracted from the rank-2 matrix as the late fusion result. Despite its advantages, the supervision information is not present in the practice of seeking shared score patterns across the classifiers. As a result, the proposed method depends entirely on the agreement of different classifiers, which may blindly bring the common prediction errors shared across different classifiers into the final fusion results. On the contrary, our method focuses on learning the optimal fusion weights for the individual samples by exploiting the supervision information, and hence can achieve robust fusion results by considering the different prediction abilities of all classifiers on the each sample.

Methodologically, our work is also inspired by the recent success of optimizing ranking results at the top methods in machine learning [14], [16], [17]. One representative work is the support vector infinite push method [14],

which introduces the ℓ_∞ push loss function into the learning-to-rank problem with the goal of maximizing the number of positive samples on the absolute top of the list. Later on, Rakotomamonjy *et al.* [18] further developed a sparse support vector infinite push method, which incorporates the feature selection into the support vector infinite push method. However, these methods can only learn a uniform ranking function for all the test samples, and cannot be applied to the sample specific fusion weight learning. Related work can also be found in graph-based semi-supervised learning [19], [20], but they are restricted to estimating the classification or ranking score of each node, and thus cannot be used to learn the weights for fusion purpose.

Since our method is based on graph-based semi-supervised learning [21], [22], the learning process could be difficult and slow on large-scale datasets. Therefore, we need to find an efficient way to scale up our method. The complexity of typical graph-based semi-supervised learning is $O(n^3)$ due to the inverse step of $n \times n$ graph Laplacian. Although many methods have been proposed to reduce the complexity, most methods are still not satisfactory. For example, the complexity of classical Transductive SVM (T-SVM) [23] grows exponentially with n, while the latest large-scale T-SVM based on CCCP [24] still has $O(n^2)$ complexity. To tackle this problem, we leverage the AnchorGraph approach [15] to scale up our SSLF method linearly. The model adopts the manifold assumption and utilizes a selected subset of samples (anchor points) to cover the entire point cloud. The anchor-based label prediction and adjacency matrix are simultaneously learnt by the formulation proposed in [15]. Therefore, we can accelerate the information propagation process by diffusing weights on anchor points first, then reconstruct the adjacency matrix to assign fusion weights to all samples. This approach enables the proposed SSLF method to be efficiently applied on large-scale dataset.

III. Learning Sample Specific Fusion Weight

In this section, we will introduce the two proposed Sample Specific Late Fusion (SSLF) methods in details. The notations and definitions are first presented, and optimization formulations are then elaborated.

A. Notation and Definition

The proposed method works in a transductive setting. Suppose we have a set of m classifier C_i $(i = 1, \ldots, m)$, each of which is learned based on one type of feature. There are l labeled samples $\{x_i, y_i\}_{i=1}^{l}$ and u unlabeled samples $\{x_i\}_{i=l+1}^{l+u}$ available, where $y_i \in \{0, 1\}$ is the label of sample x_i. Specifically, the labeled samples are "fusion weight training set", which are responsible for providing supervision information. Since our method works on the prediction scores of the classifiers, it is important that the labeled samples in fusion weight training set should be disjoint from the samples in classifier training set. This is because that the ground-truth labels of the classifier training set have been utilized by the classifiers, making the prediction scores on classifier training samples bias towards the ground-truth labels. Such prediction scores cannot reflect the classifier's prediction capabilities on the unseen samples, mitigating the value of any deliberate fusion method. In the real-world visual classification tasks, the fusion weight training set can be easily obtained. For example, besides training and test set, many visual classification tasks also provide the validation set for parameter selection. In this case, we can directly select it as our fusion weight training set. Even if the validation set is not available, we can still retrieve such sample set by splitting from the classifier training samples before training, or even crawling additional labeled samples from the online resources.

TABLE I
The Table of Notations

Symbol	Explanation
x_i	The i-th data sample
y_i	The label of x_i
l	Total number of labeled data
u	Total number of unlabeled data
m	Total number of classifiers
C_i	The i-th classifier trained by one type of feature
s_i^j	Prediction score of j-th classifier C_j for sample x_i
s_i^+	Prediction score of i-th positive sample x_i
s_j^-	Prediction score of j-th negative sample x_j
w_i^j	The j-th fusion weight for sample x_i
$\mathbf{w}_i$	Fusion weight vector for sample x_i, $\mathbf{w} \in \mathbb{R}^m$
p	Total number of positive samples
n	Total number of negative samples
$\mathcal{P}$	Subset of positive samples $\mathcal{P} = \{\mathbf{s}_i^+\}_{i=1}^p$
$\mathcal{N}$	Subset of negative samples $\mathcal{N} = \{\mathbf{s}_i^-\}_{i=1}^n$
$f_i(s_i)$	Sample specific fusion function $f_i(s_i) = \mathbf{w}_i^\top \mathbf{s}_i$
$\mathbf{G}$	Weight matrix of kNN graph for samples

By applying the classifiers on the labeled samples and unlabeled samples, we obtain a labeled score vector set $\{\mathbf{s}_i, y_i\}_{i=1}^{l}$ and unlabeled score vector set $\{\mathbf{s}_i\}_{i=l+1}^{l+u}$, where $\mathbf{s}_i = [s_i^1, \ldots, s_i^m]^\top$ denotes the prediction score vector of sample x_i $(i = 1, \ldots, l+u)$ with s_i^j being the prediction score of the j-th classifier C_j. To ease the following discussion, we divide the fusion weight training set into a positive subset $\mathcal{P} = \{\mathbf{s}_i^+\}_{i=1}^p$ and a negative subset $\mathcal{N} = \{\mathbf{s}_i^-\}_{i=1}^n$, where $\mathbf{s}_i^+$ and $\mathbf{s}_i^-$ respectively denote the score vector of a positive sample and a negative sample, p and n are the total number of positive and negative samples. Finally, we stack all score vectors into a matrix $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_{l+u}]$, which includes the score vectors of the labeled and unlabeled samples. The notations used in this paper are summarized in Table I.

B. Problem Formulation

Our objective is to learn a sample specific fusion function $f_i(\mathbf{s}_i) = \mathbf{w}_i^\top \mathbf{s}_i$ for each sample $(i = 1, \ldots, l+u)$, where $\mathbf{w}_i = [w_i^1, \ldots, w_i^m]^\top$ is a non-negative fusion weight vector with w_i^j being the fusion weight of s_i^j. While one can easily learn the fusion weights for the labeled samples based on the given label information, learning the fusion weights for the unlabeled samples is not trivial since there is no direct supervision available.

As visually similar samples share similar labels, the fusion score distribution should be locally smooth. To leverage

this property, we build a nearest neighbor graph based on the low level features. For each sample x_i, we find its K nearest neighbors and connect edges between x_i and its neighbors. The entry G_{ij} in the weight matrix $\mathbf{G}$ associated with the graph is defined as

$$G_{ij} = \begin{cases} \exp(-\frac{\bar{d}(x_i,x_j)}{\sigma}), & \text{if } i \in \mathcal{N}_K(j) \text{ or } j \in \mathcal{N}_K(i), \\ 0, & \text{otherwise}, \end{cases} \quad (1)$$

where $\mathcal{N}_K(i)$ denotes the index set of the K nearest neighbors of samples x_i ($K = 6$ in this work), $\bar{d}(x_i, x_j) = \frac{1}{m}\sum_{k=1}^{m} d^k(x_i, x_j)$ is the average distance between two samples, in which $d^k(x_i, x_j)$ denotes the distance calculated based on the k-th feature type (In our experiment, we use $L1$, $L2$ and χ^2 distance functions for different features). σ is the radius parameter of the Gaussian function, which is set as the mean value of all pairwise average distances among the samples. In practice, the graph can be constructed offline.

Then our late fusion method is formulated as follows:

$$\begin{aligned} &\min_{\mathbf{W}} \quad \Omega(\mathbf{W}) + \lambda \ell(\{f_i\}_{i=1}^{l}; \mathcal{P}, \mathcal{N}), \\ &\text{s.t.} \quad \mathbf{w}_i \geq 0, \quad i = 1, \ldots, l+u, \end{aligned} \quad (2)$$

where $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_{l+u}]$ consists of $l+u$ fusion weight vectors to be learned for both labeled and unlabeled samples, and λ is a trade-off parameter among the two competing terms. The first term is a regularization term responsible for the implicit fusion weight propagation:

$$\begin{aligned} \Omega(\mathbf{W}) &= \sum_{i,j=1}^{l+u} E_{ij}(\mathbf{w}_i^\top \mathbf{s}_i - \mathbf{w}_j^\top \mathbf{s}_j)^2 \\ &= \big(\pi(\mathbf{W})\big)^\top \mathbf{L}\big(\pi(\mathbf{W})\big), \end{aligned} \quad (3)$$

where $\mathbf{E} = \mathbf{U}^{-\frac{1}{2}}\mathbf{G}\mathbf{U}^{-\frac{1}{2}}$ is a normalized weight matrix of $\mathbf{G}$. $\mathbf{U}$ is a diagonal matrix whose (i,i)-entry is the i-th row/column sum of $\mathbf{G}$. $\mathbf{L} = (\mathbf{I} - \mathbf{E})$ is the graph laplacian. $\pi(\mathbf{W})$ is a vector defined as $\pi(\mathbf{W}) = \big((\mathbf{W}^\top \mathbf{S}) \circ \mathbf{I}\big)\mathbf{1}$, in which $\circ$ is the Hadamard matrix product. Intuitively, the minimization of Eq. (3) enforces a smooth fusion score propagation over the graph structure, making visually similar samples have similar fusion scores.

The second term $\ell(\{f_i\}_{i=1}^{l}; \mathcal{P}, \mathcal{N})$ is a loss function. In this paper we use two kinds of constraints and derive two algorithms: Ranking SSLF (R-SSLF) and Infinite Push SSLF (I-SSLF). The first one utilizes the RankSVM style constraints while the second one adopts recent developed infinite push loss function. The formulations of the two algorithms are elaborated in the following sections.

IV. Ranking Sample Specific Late Fusion (R-SSLF)

The first proposed algorithm is Ranking SSLF (R-SSLF). In the algorithm we adopted RankSVM style constraints in a large-margin framework, which enforces each positive sample to have higher fusion scores than all negative samples. We define the loss function as:

$$\ell(\{f_i\}_{i=1}^{l}; \mathcal{P}, \mathcal{N}) = \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} I_{f_i(\mathbf{s}_i^+) < f_j(\mathbf{s}_j^-)}, \quad (4)$$

where $I_{(\cdot)}$ is the indicator function whose value is 1 if the argument is true and 0 otherwise. The loss function compares each positive fusion score $f_i(\mathbf{s}_i^+)$ to all negative fusion scores $\sum_{j \in \mathcal{N}} f_j(\mathbf{s}_j^-)$ and sum up the loss where negative samples have higher scores. As the indicator function $I_{(\cdot)}$ is a discrete function and not differentiable, we choose to minimize on the hinge ranking loss, which is the convex upper bound of $I_{(\cdot)}$:

$$\ell(\{f_i\}_{i=1}^{l}; \mathcal{P}, \mathcal{N}) = \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \max(0, 1 - (f_i(\mathbf{s}_i^+) - f_j(\mathbf{s}_j^-)), \quad (5)$$

where $(w_i \neq w_j, i \neq j)$. As can be seen, the hinge loss of all positive-negative sample pairs are considered. The objective function of Ranking SSLF is defined as:

$$\begin{aligned} &\min_{\mathbf{w}} . \quad \Omega(\mathbf{W}) + \lambda \sum_{i=1}^{P} \sum_{j=1}^{N} \xi_{ij} \\ &\text{s.t.} \quad 1 - (f_i(\mathbf{s}_i^+) - f_j(\mathbf{s}_j^-)) \leq \xi_{ij} \end{aligned} \quad (6)$$

To solve this objective function, we can use the alternating descent method to solve the weights one by one [25]. First we rewrite the objective function as below:

$$\begin{aligned} &\min_{\mathbf{w}} . \quad \sum_{i,j=1}^{l+u} E_{ij}(\mathbf{w}_i^\top \mathbf{s}_i - \mathbf{w}_j^\top \mathbf{s}_j)^2 + \lambda \sum_{i=1}^{P} \sum_{j=1}^{N} \xi_{ij} \\ &\text{s.t.} \quad 1 - (\mathbf{w}_i^\top \mathbf{s}_i^+ - \mathbf{w}_j^\top \mathbf{s}_j^-) \leq \xi_{ij}, \quad i \in \mathcal{P}, \quad j \in \mathcal{N} \\ &\qquad \xi \geq 0, \ \mathbf{w}_i \geq 0, \quad i = 1, \ldots, l+u. \end{aligned} \quad (7)$$

To solve all weights, we need to fix $\mathbf{w}_j$ and solve $\mathbf{w}_i (i \neq j)$ in each iteration. Furthermore, we need to solve weights of positive and negative samples respectively. Let us solve the weights of positive samples first. The objective function of positive weight $\mathbf{w}_i$ can be written as a standard QP problem:

$$\begin{aligned} &\min_{\mathbf{w}} . \quad \sum_{i,j=1}^{l+u} E_{ij}(\mathbf{w}_i^\top \mathbf{s}_i \mathbf{s}_i^\top \mathbf{w}_i - 2\mathbf{w}_i^\top \mathbf{s}_i \mathbf{s}_j^\top \mathbf{w}_j) + \lambda \sum_{j=1}^{N} \xi_{ij} \\ &\text{s.t.} \quad 1 - (\mathbf{w}_i^\top \mathbf{s}_i^+ - \mathbf{w}_j^\top \mathbf{s}_j^-) \leq \xi_{ij}, \quad j \in \mathcal{N} \\ &\qquad \xi \geq 0, \quad \mathbf{w}_i \geq 0 \end{aligned} \quad (8)$$

where $\mathcal{N}$ is the set of negative samples. By introducing Lagrange multipliers, we get the following equation:

$$\begin{aligned} &\mathcal{L}(\mathbf{w}_i, \alpha, \gamma, \xi, \beta) \\ &\quad = \sum_{i,j=1}^{l+u} E_{ij}(\mathbf{w}_i^\top \mathbf{s}_i \mathbf{s}_i^\top \mathbf{w}_i - 2\mathbf{w}_i^\top \mathbf{s}_i \mathbf{s}_j^\top \mathbf{w}_j) \\ &\quad + \sum_{j=1}^{N} \alpha_{ij}(1 - ((\mathbf{s}_i^+)^\top \mathbf{w}_i - (\mathbf{s}_j^-)^\top \mathbf{w}_j) - \xi_{ij}) \\ &\quad + \lambda 1^\top \xi - \beta^\top \xi - \gamma^\top \mathbf{w}_i \end{aligned} \quad (9)$$

The derivative of the Lagrangian equation is shown as:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}_i} &= 2(\sum_{j=1}^{l+u} E_{ij}\mathbf{s}_i\mathbf{s}_i^\top)\mathbf{w}_i - 2\sum_{j=1}^{l+u} E_{ij}\mathbf{s}_i\mathbf{s}_j^\top \mathbf{w}_j \\ &\quad - \sum_{j=1}^{N} \mathbf{s}_i^+ \alpha_{ij} - \gamma = 0 \end{aligned} \quad (10)$$

Thus,

$$\mathbf{w}_i = \frac{1}{2} B^{-1}(s_i^+ 1^\top \alpha + \gamma + 2C_i) \quad (11)$$

Algorithm 1 Gradient-Projection for Ranking SSLF

Input: $\mathbf{S} \in \mathbb{R}^{m\times(l+u)}$, $\mathbf{L} \in \mathbb{R}^{(l+u)\times(l+u)}$, $\{y_i\}_{i=1}^{l}$, λ.
Initialization: $\mathbf{a}^0 \leftarrow 10^{-4}$ (initialize to small values)
for t=0 to t_{max} **do**
  [Gradient step] $\alpha^{(t+1/2)} \leftarrow \alpha^{(t)} - \frac{\eta_0}{\sqrt{t}} \bigtriangledown Q(\alpha^{(t)})$
  [Projection step]
  if $\alpha_{ij}^{(t+1/2)} < 0$ **then**
    $\alpha_{ij}^{(t+1)} \leftarrow 0$
  else if $\alpha_{ij}^{(t+1/2)} > \frac{\lambda}{\mathcal{PN}}$ **then**
    $\alpha_{ij}^{(t+1)} \leftarrow \frac{\lambda}{\mathcal{PN}}$
  else
    $\alpha_{ij}^{(t+1)} \leftarrow \alpha_{ij}^{(t+1/2)}$
  end if
end for
Output: $f_i(\mathbf{s}_i) = (\mathbf{w}_i^*)^\top \mathbf{s}_i$, $i = l+1, \ldots, l+u$.

where $B = \sum_{j=1}^{l+u} E_{ij}\mathbf{s}_i\mathbf{s}_i^\top$, $C_i = \sum_{j=1}^{l+u} E_{ij}\mathbf{s}_i\mathbf{s}_j^\top\mathbf{w}_j$. Substituting (11) back into (9) yields the following dual problem:

$$\begin{aligned}\min_{\alpha}\ & ((s_i^+)^\top\alpha + \gamma + 2C_i)^\top B^{-1}((s_i^+)^\top\alpha + \gamma + 2C_i) \\ & + 4\sum\nolimits_{j=1}^{N}((s_j^-)^\top\mathbf{w}_j - 1)\alpha_{ij}) \\ \text{s.t.}\ & 0 \le \alpha \le \lambda, \quad \gamma \ge 0 \end{aligned} \tag{12}$$

For calculation, we can expand and rewritten dual problem formula as below:

$$\begin{aligned}\min_{\alpha}\ & ((s_i^+)^\top\alpha + \gamma)^\top B^{-1}((s_i^+)^\top\alpha + \gamma) \\ & + 4C_i^\top B^{-1}((s_i^+)^\top\alpha + \gamma) + 4(\sum\nolimits_{j=1}^{N}(s_j^-)^\top\mathbf{w}_j - 1)\alpha \\ \text{s.t.}\ & 0 \le \alpha \le \lambda, \quad \gamma \ge 0 \end{aligned} \tag{13}$$

In terms of weights of negative samples $\mathbf{w}_j$, the inference steps are the same and the objective functions are similar:

$$\mathbf{w}_j = \frac{1}{2}B^{-1}(-(s_j^-)^\top\alpha + \gamma + 2C_j) \tag{14}$$

where $B = \sum_{i=1}^{l+u} E_{ij}\mathbf{s}_j\mathbf{s}_j^\top$, $C_j = \sum_{i=1}^{l+u} E_{ij}\mathbf{s}_i\mathbf{s}_j^\top\mathbf{w}_i$. Substituting (14) back into the Lagragian equation we will get the dual problem:

$$\begin{aligned}\min_{\alpha}\ & ((s_j^-)^\top\alpha - \gamma)^\top B^{-1}((s_j^-)^\top\alpha - \gamma) \\ & - 4C_j^\top B^{-1}((s_j^-)^\top\alpha - \gamma) - (4\sum\nolimits_{i=1}^{l}(s_i^+)^\top\mathbf{w}_i + 4)\alpha \\ \text{s.t.}\ & 0 \le \alpha \le \lambda, \quad \gamma \ge 0 \end{aligned} \tag{15}$$

The equation (13) and (15) are standard Quadratic Programming problems that global optimal of α can be found. However, the typical QP solvers require $O((\mathcal{PN})^3)$ time. To accelerate the process, we propose to employ the gradient projection method in [14] with $O((\mathcal{P}+\mathcal{N})^2)$ time. Let $Q(\alpha)$ denote the quadratic objective function and $\bigtriangledown Q$ denotes the gradient of Q, the optimization procedure is listed in Algorithm 1.

V. Infinite Push Sample Specific Late Fusion (I-SSLF)

The second proposed algorithm is to employ an infinite push loss function [26], which tries to maximize the number of positive samples scored above the highest-scored negative sample. Actually, the number of positives scored above the highest-scored negative is exactly the largest number of positives scored above any negative, which as a fraction of the total number of positives p, is defined as:

$$\ell(\{f_i\}_{i=1}^{l}; \mathcal{P}, \mathcal{N}) = \max_{1\le j\le n}\big(\frac{1}{p}\sum_{i=1}^{p} I_{f_i(\mathbf{s}_i^+) < f_j(\mathbf{s}_j^-)}\big), \tag{16}$$

where I is the indicator function. The maximum over j corresponds to taking the ℓ_∞ norm of the vector containing n terms in the parentheses of (16). This essentially ensures the positive samples have higher fusion scores than the negative, leading to more accurate fusion results.

While directly minimizing (16) is difficult due to its discrete nature, we minimize instead a convex upper bound:

$$\ell(\{f_i\}_{i=1}^{l}; \mathcal{P}, \mathcal{N}) = \max_{1\le j\le n}\left(\frac{1}{p}\sum_{i=1}^{p}\big(1 - (\mathbf{w}_i^\top\mathbf{s}_i^+ - \mathbf{w}_j^\top\mathbf{s}_j^-)\big)_+\right), \tag{17}$$

where $(u)_+ = u$ if $u > 0$ and 0 otherwise.

Finally, the objective function can be written as:

$$\begin{aligned}\min_{\mathbf{W}}\ & \big(\pi(\mathbf{W})\big)^\top\mathbf{L}\big(\pi(\mathbf{W})\big) \\ & + \lambda\max_{1\le j\le n}\left(\frac{1}{p}\sum_{i=1}^{p}\big(1 - (\mathbf{w}_i^\top\mathbf{s}_i^+ - \mathbf{w}_j^\top\mathbf{s}_j^-)\big)_+\right), \\ \text{s.t.}\ & \mathbf{w}_i \ge 0,\ i = 1, \ldots, l+u. \end{aligned} \tag{18}$$

The above objective function is convex, and thus can achieve the global optimum. The main difficulty in optimizing (18) arises from the non-smoothness of the ℓ_∞ norm loss function. In this section, we will derive an Alternating Direction Method of Multipliers (ADMM) method [16] for the optimization. To this end, we first drop the $\mathbf{w}_i \ge 0$ constraint, so that the ADMM method can be applied to solve (18), and then we project the solution back to the feasible region.

A. Deriving ADMM Formulation

We rewrite the optimization problem in (18) as the following linearly-constrained problem:

$$\begin{aligned}\min_{\mathbf{W}, a_{ij}}\ & \Omega(\mathbf{W}) + \lambda\max_{1\le j\le n}\big(\frac{1}{p}\sum_{i=1}^{p}(a_{ij})_+\big), \\ \text{s.t.}\ & a_{ij} = 1 - (\mathbf{w}_i^\top\mathbf{s}_i^+ - \mathbf{w}_j^\top\mathbf{s}_j^-). \end{aligned} \tag{19}$$

Then, by defining the matrix $\mathbf{J} = [(\mathbf{A}\bigotimes\mathbf{B})^\top, (\mathbf{C}\bigotimes\mathbf{D})^\top, (\mathbf{F}\bigotimes\mathbf{B})^\top]^\top$, where $\mathbf{A} = \mathbf{I}_p$, $\mathbf{B} = \mathbf{1}_{n\times 1}^\top$, $\mathbf{C} = \mathbf{I}_n$, $\mathbf{D} = -\mathbf{1}_{p\times 1}^\top$, $\mathbf{F} = \mathbf{0}_{u\times p}$, $\bigotimes$ denotes the Kronecker product, $\mathbf{X} = (\mathbf{W}^\top\mathbf{S}) \circ \mathbf{I}$, the vector $\mathbf{a}$ composing of all a_{ij}'s and the

function $g(\mathbf{a}) = \lambda \max_{1\le j\le n}\left(\frac{1}{p}\sum_{i=1}^{p}\max(a_{ij},0)\right)$, we arrive at the following formulation:

$$\begin{aligned}\min_{\mathbf{W},\mathbf{a}}\ & \Omega(\mathbf{W}) + g(\mathbf{a}),\\ \text{s.t.}\ & \mathbf{J}^\top\mathbf{X}\mathbf{1} + \mathbf{a} - \mathbf{1} = 0.\end{aligned} \tag{20}$$

The augmented Lagrangian of the above problem is

$$\begin{aligned}\mathcal{L}(\mathbf{W},\mathbf{a},\gamma,\mu) = \Omega(\mathbf{W}) + g(\mathbf{a}) + \gamma^\top(\mathbf{J}^\top\mathbf{X}\mathbf{1} + \mathbf{a} - \mathbf{1})\\ + \frac{\mu}{2}\|\mathbf{J}^\top\mathbf{X}\mathbf{1} + \mathbf{a} - \mathbf{1}\|_2^2,\end{aligned} \tag{21}$$

where γ is the vector of Lagrangian multipliers of the linear constraints and μ is a weighting parameter of the quadratic penalty. Following the experimental practices of ADMM [16], we set μ to be 10^{-4}. The above formulation can be equally rewritten as

$$\mathcal{L}(\mathbf{W},\mathbf{a},\beta) = \Omega(\mathbf{W}) + g(\mathbf{a}) + \frac{\mu}{2}\|\mathbf{J}^\top\mathbf{X}\mathbf{1} + \mathbf{a} - \mathbf{1} + \beta\|_2^2, \tag{22}$$

where $\beta = \gamma/\mu$. Finally, the optimization becomes iteratively solving the saddle point of the augmented Lagrangian. At iteration k, we need to solve the following three sub-problems:

$$\mathbf{W}^{k+1} = \arg\min_{\mathbf{W}} \mathcal{L}(\mathbf{W},\mathbf{a}^k,\beta^k), \tag{23}$$

$$\mathbf{a}^{k+1} = \arg\min_{\mathbf{a}} \mathcal{L}(\mathbf{W}^{k+1},\mathbf{a},\beta^k), \tag{24}$$

$$\beta^{k+1} = \beta^k + \mathbf{J}^\top\mathbf{X}^{k+1}\mathbf{1} + \mathbf{a}^{k+1} - \mathbf{1}, \tag{25}$$

where $\mathbf{X}^{k+1} = \left((\mathbf{W}^{k+1})^\top\mathbf{S}\right)\circ\mathbf{I}$. In the next subsection, we will show how to solve these sub-problems.

B. Alternating Optimization

The optimization of Eq. (23) can be stated as

$$\min_{\mathbf{W}} \Xi(\mathbf{W}) \equiv \frac{\mu}{2}\|\mathbf{J}^\top\mathbf{X}\mathbf{1} - \mathbf{t}\|_2^2 + \Omega(\mathbf{W}), \tag{26}$$

where $\mathbf{t} = 1 - \mathbf{a}^k - \beta^k$ and its gradient can be calculated as

$$\frac{\nabla\Xi(\mathbf{W})}{W_{ij}} = \mathrm{tr}\left[\left(\mu\mathbf{J}(\mathbf{J}^\top\mathbf{X}\mathbf{1} - \mathbf{t})\mathbf{1}^\top + 2\mathbf{L}\mathbf{X}\mathbf{1}\mathbf{1}^\top\right)^\top\frac{\partial\mathbf{X}}{\partial W_{ij}}\right], \tag{27}$$

through which the optimization problem can be solved by a conjugate gradient descent method.

The optimization problem in Eq. (24) boils down to be

$$\min_{\mathbf{a}} g(\mathbf{a}) + \frac{\mu}{2}\|\mathbf{a} - \mathbf{t}\|_2^2, \tag{28}$$

where $\mathbf{t} = \mathbf{1} - \beta^k - \mathbf{J}^\top\mathbf{X}^{k+1}\mathbf{1}$. To solve the two nested max operators in $g(\mathbf{a})$, the double trick can be used to convert the problem as:

$$\begin{aligned}\min_{\mathbf{a}^+,\mathbf{a}^-}\ & \frac{1}{2}\|\mathbf{a}^+ - \mathbf{a}^- - \mathbf{t}\|_2^2 + \max_{1\le j\le n}\left(\frac{\lambda}{\mu p}\sum_{i\in\mathcal{G}_j} a_i^+\right)\\ \text{s.t.}\ & \mathbf{a}^+ \ge 0,\quad \mathbf{a}^- \ge 0,\end{aligned} \tag{29}$$

where $\mathbf{a} = \mathbf{a}^+ - \mathbf{a}^-$ and $\mathcal{G}_j$ denotes the indices of the positive samples in vector $\mathbf{a}$ that are coupled with the negative sample $\mathbf{s}_j$. This problem can be solved by iterative optimization by employing the optimization method in [18], where $\mathbf{a}^-$ has a closed-form solution while $\mathbf{a}^+$ can be solved by Douglas-Rachford method [27], which alternately performs two proximal operators on the positive quadrant and the $\ell_{1,\infty}$ mixed norm until convergence [28]. The optimization procedure is shown in Algorithm 2.

Algorithm 2 ADMM for Infinite-Push SSLF

Input: $\mathbf{S}\in\mathbb{R}^{m\times(l+u)}$, $\mathbf{L}\in\mathbb{R}^{(l+u)\times(l+u)}$, $\{y_i\}_{i=1}^{l}$, λ.
Initialization: $k=0$, $\mathbf{a}^0$, $\beta^0=0$, $\mathbf{W}^0>0$, $\mu=10^{-4}$.
Calculate $\mathbf{J}\in\{-1,0,1\}^{(l+u)\times pn}$ based on the label information.
repeat
  $\mathbf{t} = \mathbf{1} - \mathbf{a}^k - \beta^k$.
  $\mathbf{W}^{k+1} = \arg\min_{\mathbf{W}} \frac{\mu}{2}\|\mathbf{J}^\top\mathbf{X}\mathbf{1} - \mathbf{t}\|_2^2 + \Omega(\mathbf{W})$.
  Force the negative values in $\mathbf{W}^{k+1}$ to 0.
  $\mathbf{X}^{k+1} = \left((\mathbf{W}^{k+1})^\top\mathbf{S}\right)\circ\mathbf{I}$.
  $\mathbf{t} = \mathbf{1} - \beta^k - \mathbf{J}^\top\mathbf{X}^{k+1}\mathbf{1}$.
  Obtain $\mathbf{a}^-$ and $\mathbf{a}^+$ by solving Eq. (**??**) based on [**?**].
  $\mathbf{a}^{k+1} = \mathbf{a}^+ - \mathbf{a}^-$.
  $\beta^{k+1} = \beta^k + \mathbf{J}^\top\mathbf{X}^{k+1}\mathbf{1} + \mathbf{a}^{k+1} - \mathbf{1}$.
  $k = k+1$.
until convergence.
Output: $f_i(\mathbf{s}_i) = (\mathbf{w}_i^*)^\top\mathbf{s}_i$, $i = l+1,\ldots,l+u$.

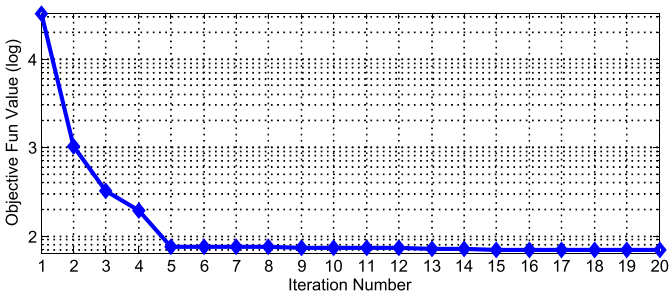


Fig. 2. The convergence curve of I-SSLF on the 1st category of Oxford Flower 17 dataset.

C. Algorithmic Analysis

Algorithm 2 is built upon ADMM and Douglas-Rachford procedure, each of which has shown very good convergence property. Since the objective function is convex, the algorithm will approach the global optimum. Figure 2 shows the convergence process of the iterative optimization which is captured in our later experiment. As can be seen, the objective function value converges to the minimum after about 6 iterations, which is rather fast. For example, in the experiment on Oxford Flower 17 dataset (see Section VII-A) implemented on the MATLAB platform on an Intel Xeon X5660 workstation with 3.2 GHz CPU and 18 GB memory, Algorithm 2 can be finished within 3.56 seconds on average for each category, which verifies its efficiency. Note that the scalability of our algorithm is dominated by the total number of samples involved in the optimization. We will introduce a technique named Anchor Graph to support large-scale datasets.

D. Learning Fusion Weight for a New Test Sample

Note that we can adopt the classical out-of-sample extension method in transductive learning to estimate the fusion score of a new sample [29], [30]. For a new test sample z, we can

use the low-level feature to search a set of nearest neighbors $\{x_i\}_{i=1}^{q}$ from all samples in the original dataset, where x_i is a neighbor of z and q is the total number of neighbors. Based on the neighborhood set, the late fusion score can be determined as $f(z) = \sum_{i=1}^{q} \frac{G(z,x_i)}{\sum_{i=1}^{q} G(z,x_i)} (\mathbf{w}_i^*)^\top \mathbf{s}_i$, where $G(z, x_i)$ is the similarity between z and x_i, and $(\mathbf{w}_i^*)^\top \mathbf{s}_i$ is the fusion score of x_i obtained on the original dataset. In this way, we obtain the fusion score for the unseen sample.

VI. Scaling Up SSLF With Anchor Graph

As mentioned in previous sections, the running time of proposed SSLF framework grows quadratically with sample size. In order to apply SSLF to large scale dataset efficiently, we propose to utilize the AnchorGraph model [15] to reduce the size of the graph. The AnchorGraph is defined as

$$\mathbf{x} = \mathbf{Z}\mathbf{a}, \quad \mathbf{Z} \in \mathbb{R}^{n \times n_a}, \quad n_a \ll n, \tag{30}$$

where $\mathbf{a}$ is the anchor points vector, n_a is the number of anchor points, and $\mathbf{Z}$ is the weights. To design good estimation of the graph adjacency matrix $\mathbf{G}$ with $\mathbf{Z}$, the authors propose three design principles:

- $\sum_{k=1}^{m} Z_{ik} = 1$ and $Z_{ik} \geq 0$. The *manifold assumption* suggests the contiguous data points should have similar labels and distant data points are very unlikely to take similar labels. Therefore, we can impose the nonnegative normalization constraints $\sum_{k=1}^{m} Z_{ik} = 1$ and $Z_{ik} \geq 0$.
- The weighted adjacency matrix $\mathbf{G} \geq 0$. The nonnegative adjacency matrix is sufficient to make the resulting graph Lapalcian L positive semidefinite, which can guarantee global optimum of many graph-based semi-supervised learning.
- The weighted adjacency matrix $\mathbf{G}$ should be sparse. A sparse adjacency matrix has less spurious connections between dissimilar points and can lead to better quality. Empirically sparse graphs have better performances than fully connected dense graphs [22].

Following the three principles, the authors [15] proposed to design the matrix $\mathbf{Z}$ based Local Linear Embedding (LLE) [31], which is named Local Anchor Embedding (LAE):

$$\begin{aligned} \min_{\mathbf{z}_i} \quad & \frac{1}{2}||\mathbf{x}_i - \mathbf{U}_{<i>}\mathbf{z}_i||^2 \\ \text{s.t.} \quad & \mathbf{1}^T\mathbf{z} = 1, \quad \mathbf{z}_i \geq 0, \end{aligned} \tag{31}$$

where $\mathbf{U}$ is a sub-matrix composed of k nearest anchor points of $\mathbf{x}_i$. In practice it turns out that k-means is the most effective way to find anchor points. Beyond LLE, the LAE applies the nonnegative constraint and turn (31) into a multinomal simplex:

$$\mathbb{S} = \{\mathbf{z} \in \mathbb{R}^s : \mathbf{1}^T\mathbf{z} = 1, \mathbf{z} \geq 0\}, \tag{32}$$

where s is the number of anchors used to reconstruct a data point. The standard QP solver needs to calculate the approximation of Hessian matrix and therefore quite inefficient. The authors proposed to use projected gradient method to accelerate the process. As the result, the complexity of designing $\mathbf{Z}$ is $O(sn_ab + s^2Tn)$, where T is the iterations during optimization. To apply AnchorGraph in our framework, we first apply k-means to learn anchor points, and learn sample specific fusion weights for each anchor. Once the fusion weights of anchors are learnt, we reconstruct the fusion weights of all samples by using sparse weight matrix $\mathbf{Z}$:

$$\mathbf{W} = \mathbf{Z}\mathbf{W}_a, \quad \mathbf{W} \in \mathbb{R}^{n \times m}, \quad \mathbf{Z} \in \mathbb{R}^{n \times h}, \quad \mathbf{W}_a \in \mathbb{R}^{n_a \times m}. \tag{33}$$

where $\mathbf{W}_a$ is the fusion weight matrix of anchor points.

VII. Experiments

In this section, we will evaluate the proposed late fusion method by applying it to various visual recognition tasks including object classification and video event detection. Six different methods are run on each dataset to compare the performances: (1) Kernel Averaging (KA). The kernel matrices of different features are averaged to obtain a fused kernel matrix. This is actually the most common way for early fusion of multiple features and is proved to achieve highly comparative results as multiple kernel learning [32]. (2) Average Late Fusion (ALF). After getting the prediction scores from all the classifiers, we simply average the scores as the fusion result. (3) Low Rank Late Fusion (LRLF). In this method, the prediction scores of each classifier are first converted into a binary comparative relationship matrix and a shared rank-2 matrix is then discovered across all matrices. The final fusion score vector can be extracted from the rank-2 matrix by matrix decomposition. (4) Uniform Weight Late Fusion (UWLF). Instead of learning sample specific fusion functions, we learn a uniform fusion function $f(\mathbf{s}) = \mathbf{w}^\top\mathbf{s}$ for all the samples. This essentially applies the same weight w_i to all the scores of the i-th classifier. To achieve this, we replace the fusion function $f_i(\mathbf{s}_i) = \mathbf{w}_i^\top\mathbf{s}_i$ in our objective function with $f(\mathbf{s}_i) = \mathbf{w}^\top\mathbf{s}_i$ $(i = 1, \ldots, l + u)$. (5) Our proposed Ranking Sample Specific Late Fusion (R-SSLF) and (6) Infinite-Push Sample Specific Late Fusion (I-SSLF) method.

Following previous work on late fusion [4], we employ the probabilistic outputs of the one-vs-all SVM classifier as the prediction scores, in which each value measures the possibility of classifying a sample as positive. To evaluate the performance of each method, the Average Precision (AP) is employed as the evaluation metric. The AP for each visual category is calculated and the mean Average Precision (mAP) across all categories of the entire dataset are reported as the final evaluation metric. To search the appropriate parameter for our method and UWLF, we vary the value of λ in the grid of $\{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$, and run 2-fold cross validation on the fusion weight training set to select the best parameter value based on validation performance. Regarding with the parameter setting of LRLF, we follow the suggested parameter setting strategy as in [4] and choose the best parameter values based on 2-fold cross-validation. The tradeoff parameter for SVM is selected from $\{10^{-1}, \ldots, 10^{3}\}$ through 5-fold cross-validation on the training set.

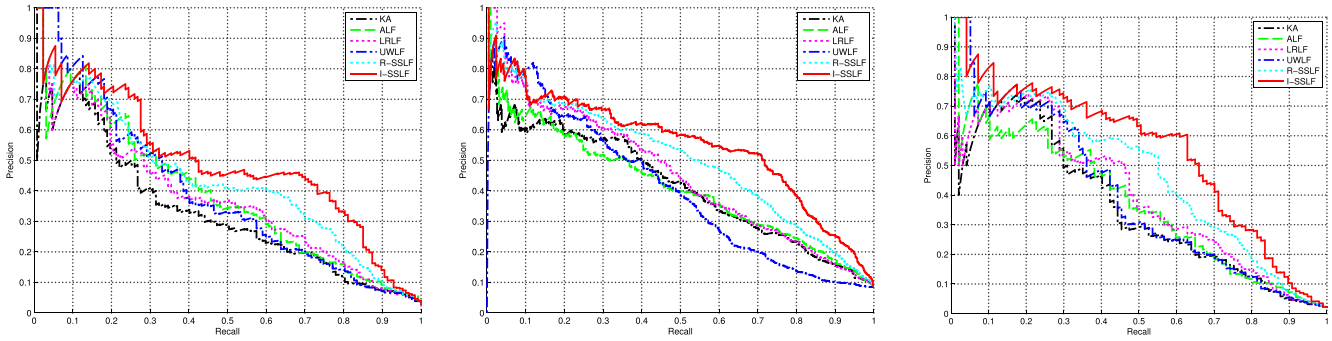
A. Results for Object Classification

In this subsection, we evaluate our proposed method on the object classification task. The following two benchmark

TABLE II

PER-CATEGORY PERFORMANCE COMPARISON (AP %) OF DIFFERENT METHODS ON PASCAL VOC'07 DATASET

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	moto	person	plant	sheep	sofa	train	tv	mAP
KA	73.9	54.0	48.5	68.3	19.8	55.2	71.5	50.2	44.1	33.5	44.4	41.6	77.4	60.5	84.9	37.5	36.1	39.1	77.4	42.0	53.0
ALF	70.0	55.4	47.2	62.8	20.4	56.0	69.6	51.9	43.2	39.1	43.5	41.7	76.7	58.2	82.3	33.1	37.7	37.0	76.4	41.1	52.2
LRLF	76.0	57.7	52.8	**73.2**	24.5	63.5	73.0	53.4	49.3	39.5	48.1	45.7	**80.9**	62.2	**88.1**	40.6	40.1	45.5	72.3	42.7	56.5
UWLF	74.6	55.2	50.7	68.9	22.8	57.1	72.5	51.9	46.7	37.1	47.0	43.5	77.8	61.4	85.5	39.4	39.0	42.0	78.1	44.2	54.8
SVM [?]	72.7	53.0	49.1	66.8	25.6	52.4	69.9	50.0	46.0	36.4	43.3	43.9	74.7	59.5	83.4	39.0	39.5	39.9	74.3	42.8	53.1
DPM v5 [?]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
R-CNN [?]	64.2	**69.7**	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	**56.1**	60.6	**66.8**	54.2	31.5	**52.8**	48.9	57.9	**64.7**	54.2
R-SSLF	74.3	60.5	**59.0**	73.1	30.6	61.2	73.8	59.3	**58.3**	41.3	54.7	49.1	78.9	62.8	86.5	**47.2**	44.1	46.5	79.8	51.6	59.63
I-SSLF	**78.0**	64.9	58.0	73.1	**32.2**	**64.0**	**76.4**	**62.4**	57.3	**44.6**	**56.7**	51.0	80.4	65.9	87.5	46.5	46.3	**49.7**	**82.9**	55.3	**61.7**



(a) cow (b) dog (c) sheep

Fig. 3. Precision/recall curves of different methods on three categories (cow, dog, sheep) of PASCAL VOC'07 dataset.

datasets are utilized in our experiment: PASCAL VOC'07 and Oxford Flower 17.

1) PASCAL VOC'07: This dataset consists of 9,963 images which were crawled by querying for images of 20 different object categories from Flickr website. For feature representations, we directly downloaded the 15 features provided by [33], including 4 kinds of SIFT Bag-Of-Words (BoW) histograms [36], 4 kinds of Hue BoW histograms [37], 2 kinds of RGB color histograms, 2 kinds of HSV histograms, 2 kinds of LAB color histograms and 1 GIST feature [38]. The details on the features can be found in [33]. Following [33], we use $L1$ distance for the color histograms, $L2$ for GIST, and χ^2 for the visual word histograms. For a given distance matrix, the kernel matrix of SVM classifier is calculated as $\exp(-d(x,y)/\sigma)$ where $d(x,y)$ is the distance between x and y and σ is the mean value of all the pairwise distances on the training set.

In our experiment, we follow the standard training (5,011 images) and test (4,952 images) data split provided by this dataset. To generate the fusion weight training set for late fusion, we uniformly divide the training samples of each category into 5 folds, and select 4 folds as the training data for SVM training while using the remaining 1 fold as the fusion weight training set for late fusion.[1] The experiments are repeated 5 times so that each fold can be used as the fusion weight training set, and the average result is reported. Note that such splits are only applied to the UWLF and two SSLF methods which need supervision information for late fusion. For other methods including KA, ALF, LRLF, we still use the original data splits.

Table II shows the per-category performances of all the methods in comparison. From the results, we have the following observations: (1) The proposed two SSLF methods consistently beat all the other baseline methods by a large margin, which demonstrates its effectiveness in determining the optimal fusion weights for each sample. (2) The LRLF, UWLF and two SSLF late fusion methods all outperform the ALF method. This is due to the fact that the former methods take advantages of additional knowledge (either consistent score patterns across the classifiers or supervision information) while the latter only blindly averages the scores from different classifiers without accounting their difference. (3) The sample level late fusion methods including LRLF and SSLF methods outperform the UWLF. The reason may be that UWLF only tries to learn uniform fusion weights for all the samples and hence cannot discover the optimal fusion weights for each sample. (4) Our SSLF methods perform better than LRLF method, since LRLF does not exploit the supervision information. In Figure 3, we show the precision-recall curves of different methods for some representative categories. As can be seen, the precisions of our methods are higher than the other methods when the recall varies from 0 to 1. This clearly demonstrates that our method is able to assign higher fusion scores to the positive samples. Figure 4 shows the rank positions of some example images after ranking the 4,952 test images based on fusion scores of different methods. The SSLF method successfully ranks the positive images at higher positions in the fusion score rank list.

2) Oxford Flower 17: The Oxford Flower 17 dataset is a benchmark dataset for multi-feature object classification [39]. This dataset contains 1,360 images falling into 17 different species of flowers, and each class contains 80 images. In our experiment, the predefined training (17×40), validation (17×20) and test (17×20) data splits are used along with the χ^2 distance matrices calculated from different features. There are seven features provided by this dataset,

[1]For the sake of simplicity, we set the ratio of fusion weight training set to be 1/5 of the training set. Studying the effect of varying the ratio is a legitimate topic but not the main focus of this current work.

Fig. 4. Example images and their rank positions in the fusion score rank list obtained from different fusion methods. For each method, the rank list is obtained by ranking all 4, 952 test images in descending order based on the fusion scores.

TABLE III

PERFORMANCE COMPARISONS ON OXFORD FLOWER 17

methods	mAP (%)
KA	86.0 ± 1.7
ALF	86.9 ± 2.1
LRLF	91.7 ± 1.7
UWLF	91.2 ± 1.5
Our Method (R-SSLF)	**92.90 ± 1.1**
Our Method (I-SSLF)	**93.40 ± 1.3**

Fig. 5. Top 15 images ranked with the fusion scores of different methods. Images with red borders are incorrect.

including color, shape, texture, HOG [40], clustered HSV values, SIFT feature on the foreground boundary (SIFTbdy) and SIFT feature on the foreground internal region (SIFTint). For SVM classifier, we use the χ^2 kernel and the best parameter C is selected via validation performance on the validation set. We use the validation set as fusion weight training set for UWLF and SSLF.

The results of our proposed method and all other baseline methods are shown in Table III. As can be seen, our proposed method outperforms all the baseline methods. Again, the experiment results demonstrate the superiority of the proposed method. Figure 5 shows the image ranking results of different fusion methods.

B. Results for Video Event Detection

We also test our method on the task of video event detection, in which the Columbia Consumer Video (CCV) and TRECVID 2011 Multimedia Event Detection (MED) datasets and are utilized as the testbed.

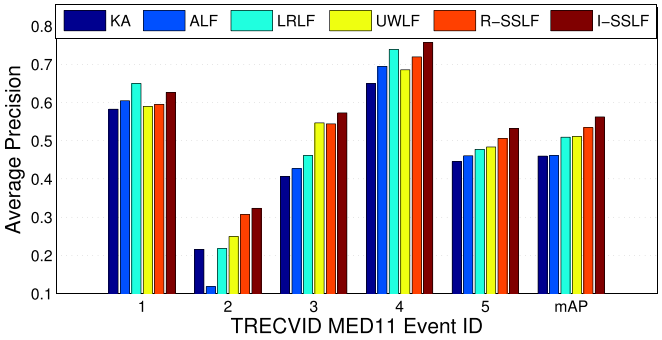


Fig. 6. Per-category performance comparison on TRECVID MED 2011 development (DEVT) dataset, which contains first five events.

TABLE IV

20 EVENTS OF COLUMBIA CONSUMER VIDEO (CCV) DATASET

ID	Event Name	ID	Event Name
1	Wedding ceremony	11	Basketball
2	Wedding reception	12	Beach
3	Biking	13	Ice skating
4	Graduation	14	Cat
5	Baseball	15	Parade
6	Birthday	16	Skiing
7	Soccer	17	Swimming
8	Playground	18	Dog
9	Bird	19	Non-music performance
10	Wedding dance	20	Music performance

1) Columbia Consumer Video (CCV): This dataset contains 9, 317 YouTube videos annotated over 20 semantic categories, where 4, 659 videos are used for training and 4, 658 videos are used for testing [41]. The event names are listed in Table IV. Three kinds of low-level features provided by this dataset, which include 5, 000-dimension SIFT BoW, 5, 000-dimension Spatial-Temporal Interest Points (STIP) BoW feature [42] and 4, 000-dimension Mel-Frequency Cepstral Coefficients (MFCC) BoW feature, are downloaded as underlying feature representation. We follow the same setting as in the TRECVID MED dataset and the per-category results are shown in Figure 7. From the results, we can see that the proposed I-SSLF method achieves the best performance in terms of mean AP, where it outperforms KA, ALF, LRLF and UWLF by 8.7%, 9.3%, 5.4% and 4.9% respectively.

2) TRECVID 2011 Multimedia Event Detection (MED): This official TRECVID MED 2011 dataset contains three dataset: Event Collection (EC), the development collection (DEVT) and test collection (DEVO). There are 2,680 videos in EC set, 10, 803 videos videos in DEVT set, and 32,061 videos in DEVO set. There are 15 events defined in MED11, which are listed in Table V. Some video clips of training data are shown in Figure 9.

We first evaluate our methods on the DEVT set. The videos in DEVT set falling into five event classes and the background class. The five events are *attempting a board trick*, *feeding an animal*, *landing a fish*, *wedding ceremony*, and *working on a woodworking project* respectively. The dataset is partitioned into the training set (8, 783 videos) and test set (2, 021 videos). The training set contains 8, 273 background videos that do not belong to any of the event classes, making the detection task challenging.

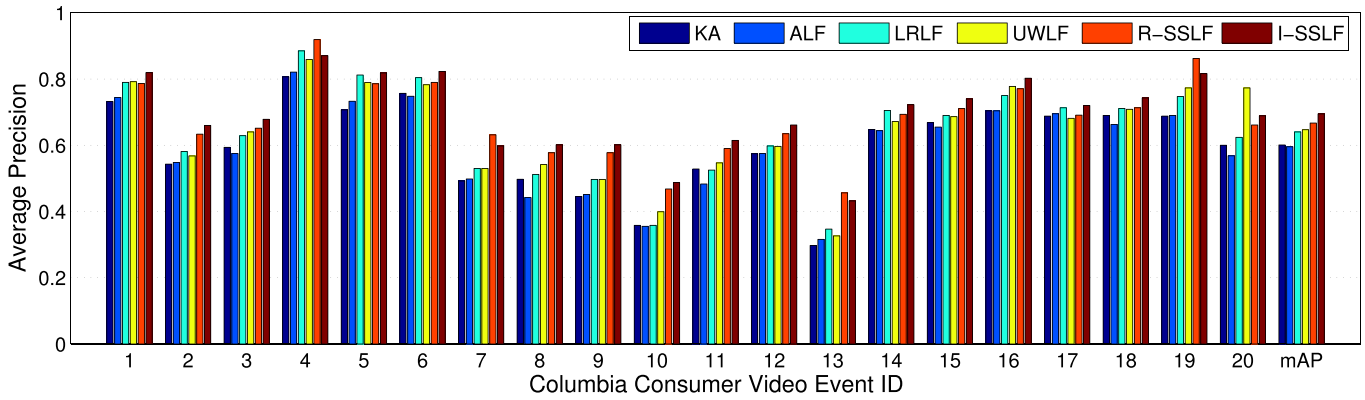


Fig. 7. Per-category performance comparison on CCV dataset. The standard deviations of mAP for R-SSLF and I-SSLF are respectively 0.36% and 0.41%.

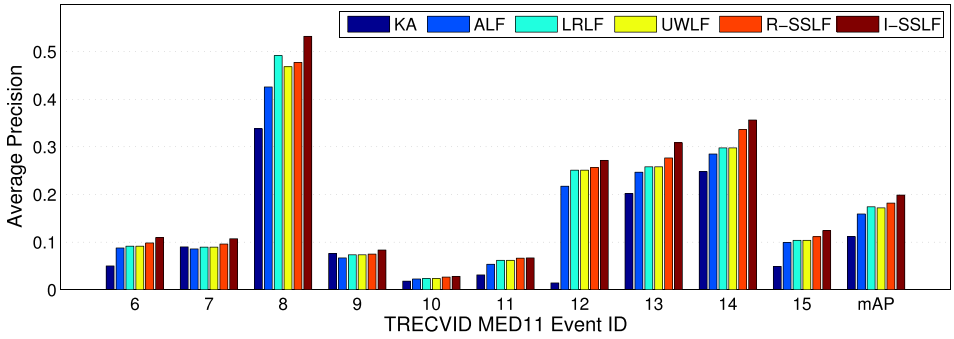


Fig. 8. Per-category performance comparison on TRECVID MED 2011 DEVT dataset. The mAP of the methods are 0.111 (KA), 0.159 (ALF), 0.174 (LRLF), 0.189 (R-SSLF) and 0.198 (I-SSLF).

TABLE V
THE 15 EVENTS DEFINED IN TRECVID MED 2011 DATASET

TRECVID MED 2011 Events			
ID	Event Name	ID	Event Name
1	Attempting board trick	9	Getting vehicle unstuck
2	Feeding animals	10	Grooming animal
3	Landing a fish	11	Making sandwich
4	Wedding ceremony	12	Parade
5	Woodworking project	13	Parkour
6	Birthday party	14	Repairing appliance
7	Changing a tire	15	Sewing project
8	Flash mob gathering		

Given a video clip, we extract three different low-level features including $5,000$-dimension SIFT BoW, $5,000$-dimension STIP BoW and $4,000$-dimension MFCC BoW. We use $L2$ distance to calculate the distance matrix of each feature and then train SVM classifiers with χ^2 kernel. Following the experiment setting on PASCAL VOC'07, we uniformly split the training samples into 5 folds and use 4 folds for SVM training and 1 fold for learning fusion weight. The experiments are repeated 5 times and the averaged result is reported.

Figure 6 shows the per-event performance of all the methods. As can be seen, our method achieves the best performance on four out of the five events. Specifically, our method outperforms the KA, ALF, LRLF and UWLF by 10.3%, 10.1%, 5.3% and 5.1% respectively in terms of mAP. Moreover, it achieves the best performances on most of the event categories. For instance, on the event "feeding an animal," our method outperforms the best baseline UWLF by 7.4%.

TABLE VI
THE MEAN APS AND RUNNING TIME ON TRECVID MED11 DEVO DATASET WITH 10 EVENTS. WE SELECT 150 POSITIVE SAMPLES AND 1000 NEGATIVE SAMPLES FOR EACH CATEGORY

Method	mAP (%)	Running Time (s)
R-SSLF	18.91	869.33
I-SSLF	19.83	935.23
R-SSLF with AnchorGraph	18.07	217.55
I-SSLF with AnchorGraph	19.09	246.33

TABLE VII
SOME STATE-OF-THE-ART RESULTS ON TRECVID MED 2011 DATASET

Authors	Description of Underlying Method	mAP (%)
Tang *et al.* [44]	AND/OR graph with 15 features	21.78
Tamrakar *et al.* [45]	Evaluation of different fusion methods with 7 features	24.42
Lai *et al.* [46]	Using static-dynamic instances for recognition with 2 features	16.02
Vahdat *et al.* [47]	A multiple kernel learning latent variable approach with 6 features	15.69

We conducted the second experiment on MED11 by following the official data splits, *i.e.* using EC and DEVT as training set and DEVO as testing set. Five kinds of features are extracted: SIFT, STIP, MFCC, Motion Boundary Histogram (MBH) [43], and GIST. All features are quantized into 5000-dimension BoW feature vector except MFCC is quantized into 4000-dimension. The $L2$ distance is used to calculate the distance matrix and χ^2 kernels are computed to train SVM classifiers through 3 fold cross validation.

Fig. 9. Exemplar video clips from some of the events defined in TRECVID MED 2011. (a) Event 5: woodworking project. (b) Event 13: parkour.

The experimental results are shown in Figure 8. Again the proposed R-SSLF and I-SSLF methods show better performance than other existing methods. Some state-of-the-art results are shown in Table VII.

In order to compare the running time of different algorithms, especially with and without AnchorGraph, we list the fusion time for each experiment in Table VI. All experiments are conducted on servers with Intel Xeon X5650 2.66 GHz CPU and 60 GB memory. For each event, we select 150 positive samples of the event in the Event Collection, and randomly choose 50 negative samples from remaining 14 events, and 300 from the DEVT set, which are 1000 negatives in total. In terms of AnchorGraph, the number of anchors are set to 200, with 20 positive and 180 negative anchors. The results show that AnchorGraph can achieve similar performance with much faster processing time. In summary, the Infinite-Push method can indeed optimize the ranking list by maximizing the number of positive samples at the top, and hence I-SSLF can achieve better results than R-SSLF on most datasets.

VIII. Conclusions

We have introduced two sample specific late fusion methods to learn the optimal fusion weights for each sample. The proposed methods work in a transductive setting that propagates the fusion weights of the labeled samples to the individual unlabeled samples, while leveraging the ranking constraints to enhance recognition accuracy. Two variants of SSLF were presented. The first algorithm is Ranking SSLF (R-SSLF) that employs the traditional ranking-SVM constraints. The second algorithm is Infinite Push SSLF (I-SSLF), which adopts latest Infinite Push constraints. The I-SSLF can be solved by the ADMM method. Additionally, we also present an efficient method to scale up our algorithms by using AnchorGraph. Extensive experiments on large-scale dataset have demonstrated the effectiveness of the proposed method on various visual category recognition tasks including object categorization and video event detection. For future work, we will pursuit the sample specific late fusion for multi-class and multi-label visual recognition tasks.

References

[1] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: A survey," *Multimedia Syst.*, vol. 16, no. 6, pp. 345–379, 2010.

[2] B. L. Tseng, C.-Y. Lin, M. Naphade, A. Natsev, and J. R. Smith, "Normalized classifier fusion for semantic visual concept detection," in *Proc. IEEE ICIP*, vol. 2. Sep. 2003, pp. II-535–II-538.

[3] O. R. Terrades, E. Valveny, and S. Tabbone, "Optimal classifier fusion in a non-Bayesian probabilistic framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 9, pp. 1630–1644, Sep. 2009.

[4] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang, "Robust late fusion with rank minimization," in *Proc. IEEE CVPR*, Jun. 2012, pp. 3021–3028.

[5] K. Nandakumar, Y. Chen, S. C. Dass, and A. K. Jain, "Likelihood ratio-based biometric score fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 342–347, Feb. 2008.

[6] Z. Zeng, M. Pantic, G. I. Roisman, and T. S. Huang, "A survey of affect recognition methods: Audio, visual, and spontaneous expressions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 1, pp. 39–58, Jan. 2009.

[7] S. Ayache, G. Quénot, and J. Gensel, "Classifier fusion for SVM-based multimedia semantic indexing," in *Advances in Information Retrieval*. Berlin, Germany: Springer-Verlag, 2007.

[8] P. Natarajan *et al.*, "Multimodal feature fusion for robust event detection in Web videos," in *Proc. IEEE CVPR*, Jun. 2012, pp. 1298–1305.

[9] J. Liu, S. McCloskey, and Y. Liu, "Local expert forest of score fusion for video event classification," in *Proc. 12th ECCV*, 2012, pp. 397–410.

[10] S. Oh *et al.*, "Multimedia event detection with multimodal feature fusion and temporal concept localization," *Mach. Vis. Appl.*, vol. 25, no. 1, pp. 49–69, 2014.

[11] C. G. M. Snoek, M. Worring, and A. W. M. Smeulders, "Early versus late fusion in semantic video analysis," in *Proc. 13th Annu. ACM Int. Conf. Multimedia*, 2005, pp. 399–402.

[12] D. Liu, K.-T. Lai, G. Ye, M.-S. Chen, and S.-F. Chang, "Sample-specific late fusion for visual category recognition," in *Proc. IEEE CVPR*, Jun. 2013, pp. 803–810.

[13] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2002, pp. 133–142.

[14] S. Agarwal, "The infinite push: A new support vector ranking algorithm that directly optimizes accuracy at the absolute top of the list," in *Proc. SDM*, 2011, pp. 839–850.

[15] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. 27th ICML*, 2010, pp. 679–686.

[16] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[17] S. Boyd, C. Cortes, M. Mohri, and A. Radovanovic, "Accuracy at the top," in *Proc. NIPS*, 2012, pp. pp. 953–961.

[18] A. Rakotomamonjy, "Sparse support vector infinite push," in *Proc. 29th ICML*, 2012, pp. 1335–1342.

[19] D. Zhou and B. Schölkopf, "Learning from labeled and unlabeled data using random walks," in *Pattern Recognition*. Berlin, Germany: Springer-Verlag, 2004, pp. 237–244.
[20] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," in *Proc. NIPS*, vol. 3. 2004, pp. 169–176.
[21] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, vol. 2. Cambridge, MA, USA: MIT Press, 2006.
[22] X. Zhu, "Semi-supervised learning literature survey," Ph.D. dissertation, Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, 2006.
[23] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. 16th ICML*, 1999, pp. 200–209.
[24] R. Collobert, F. Sinz, J. Weston, and L. Bottou, "Large scale transductive SVMs," *J. Mach. Learn. Res.*, vol. 7, pp. 1687–1712, Dec. 2006.
[25] D.-C. Zhan, M. Li, Y.-F. Li, and Z.-H. Zhou, "Learning instance specific distances using metric propagation," in *Proc. 26th Annu. ICML*, 2009, pp. 1225–1232.
[26] C. Rudin, "The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list," *J. Mach. Learn. Res.*, vol. 10, pp. 2233–2271, Dec. 2009.
[27] J. Eckstein and D. P. Bertsekas, "On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, nos. 1–3, pp. 293–318, 1992.
[28] A. Quattoni, X. Carreras, M. Collins, and T. Darrell, "An efficient projection for $l_{1,\infty}$ regularization," in *Proc. 26th Annu. ICML*, 2009, pp. 857–864.
[29] Y. Bengio, J.-F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet, "Out-of-sample extensions for LLE, Isomap, MDS, eigenmaps, and spectral clustering," in *Advances in Neural Information Processing Systems*, vol. 16. Cambridge, MA, USA: MIT Press, 2004, pp. 177–184.
[30] C. Hou, F. Nie, F. Wang, C. Zhang, and Y. Wu, "Semisupervised learning using negative labels," *IEEE Trans. Neural Netw.*, vol. 22, no. 3, pp. 420–432, Mar. 2011.
[31] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
[32] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proc. IEEE 12th ICCV*, Sep./Oct. 2009, pp. 221–228.
[33] M. Guillaumin, J. Verbeek, and C. Schmid, "Multimodal semi-supervised learning for image classification," in *Proc. IEEE CVPR*, Jun. 2010, pp. 902–909.
[34] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. *Discriminatively Trained Deformable Part Models, Release 5*. http://people.cs.uchicago.edu/~rbg/latent-release5/
[35] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE CVPR*, Jun. 2014, pp. 580–587.
[36] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[37] J. van de Weijer and C. Schmid, "Coloring local feature extraction," in *Proc. 9th ECCV*, 2006, pp. 334–348.
[38] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
[39] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. 6th ICVGIP*, Dec. 2008, pp. 722–729.
[40] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. CVPR*, Jun. 2005, pp. 886–893.
[41] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui, "Consumer video understanding: A benchmark database and an evaluation of human and machine performance," in *Proc. 1st ACM ICMR*, 2011, Art. ID 29.
[42] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, nos. 2–3, pp. 107–123, 2005.
[43] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, 2013.
[44] K. Tang, B. Yao, L. Fei-Fei, and D. Koller, "Combining the right features for complex event recognition," in *Proc. IEEE ICCV*, Dec. 2013, pp. 2696–2703.
[45] A. Tamrakar *et al.*, "Evaluation of low-level features and their combinations for complex event detection in open source videos," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 3681–3688.
[46] K.-T. Lai, D. Liu, M.-S. Chen, and S.-F. Chang, "Recognizing complex events in videos by learning key static-dynamic evidences," in *Proc. 13th ECCV*, 2014, pp. 675–688.
[47] A. Vahdat, K. Cannons, G. Mori, S. Oh, and I. Kim, "Compositional models for video event detection: A multiple kernel learning latent variable approach," in *Proc. IEEE ICCV*, Dec. 2013, pp. 1185–1192.

Kuan-Ting Lai received the B.Eng. degree in electrical engineering, the M.Sc. degree in computer science, and the Ph.D. degree from National Taiwan University, in 2003, 2005, and 2009, respectively. He was a SoC Engineer with Quanta Computer, and a Product Manager with Chinatech. From 2012 to 2013, he was a Visiting Scholar with the DVMM Laboratory, Columbia University, New York City, NY, USA. He is currently the Vice President of Technology with the Arkados Group. His research interests include computer vision, machine learning, and Internet of Things.

Dong Liu received the Ph.D. degree from the School of Computer Science and Technology, Harbin Institute of Technology, advised by Dr. H.-J. Zhang. He is currently a Post-Doctoral Research Scientist with the DVMM Laboratory, Columbia University, supervised by Prof. S.-F. Chang. His research interests fall in large-scale multimedia retrieval, computer vision, and novel machine learning techniques for enabling Next-Generation Visual Search Engine. Over the past years, he has worked by investigating new machine learning and information retrieval theoretical approaches, in particular, three major research domains: content-based tag processing for Internet social images, multimedia event detection in consumer videos, and novel machine learning techniques for visual understanding.

Shih-Fu Chang (S'89–M'90–SM'01–F'04) has made significant contributions to content-based image search, video recognition, image authentication, large-scale hashing for image search, and novel application of visual search in brain machine interface and mobile communication. Impact of his work can be seen in more than 300 peer-reviewed publications, best paper awards, 25 patents, and technologies licensed to companies. He received the IEEE Signal Processing Society Technical Achievement Award, the ACM Multimedia SIG Technical Achievement Award, the IEEE Kiyo Tomiyasu Award, and the IBM Faculty Award. For his accomplishments in education, he received the Great Teacher Award from the Society of Columbia Graduates. He served as the Editor-in-Chief of the *IEEE Signal Processing Magazine* from 2006 to 2008. In his current capacity as the Senior Vice Dean of the Columbia Engineering School, he plays a key role in strategic planning, special research initiatives, and faculty development. He is a fellow of the American Association for the Advancement of Science.

Ming-Syan Chen (F'04) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, and the M.S. and Ph.D. degrees in computer, information, and control engineering from the University of Michigan, Ann Arbor, in 1985 and 1988, respectively. He is currently a Distinguished Professor with the Department of Electrical Engineering, National Taiwan University. He was a Research Staff Member with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA, the President/CEO of the Institute for Information Industry, and also the Director of Research Center of Information Technology Innovation with Academia Sinica. His research interests include databases, data mining, cloud computing, and multimedia networking. He was a recipient of the Academic Award of the Ministry of Education, the National Science Council Distinguished Research Award, the Pan Wen Yuan Distinguished Research Award, the Teco Award, the Honorary Medal of Information, and the K.-T. Li Research Breakthrough Award for his research work, and also the Outstanding Innovation Award from IBM Corporate for his contribution to a major database product. He is a fellow of the Association for Computing Machinery.