18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Improving Scene Classification by Fusion of Training Data and Web Resources

Dongzhe Wang, and Kezhi Mao
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
dwang015@e.ntu.edu.sg, ekzmao@ntu.edu.sg

Gee-Wah Ng
DSO National Laboratories, Singapore
ngeewah@dso.org.sg

***Abstract*—Scene classification is often solved as a machine learning problem, where a classifier is first learned from training data, and class labels are then assigned to unlabelled testing data based on the outputs of the classifier. In this paper, we propose a novel scene classification framework that uses both training data and open resources on world wide web. This framework is inspired by human's capability to use external knowledge such as reference books or Internet when classifying something ambiguous or unknown. Specifically, we bring in the web resources in the form of text to aid visual recognition tasks. Both the classifier learned from training data and knowledge extracted from web resources are conclusive factors in the scene classification. Experimental results show that the new framework can improve scene classification accuracy by 9%.**

***Keywords*—scene classification, information fusion, web resources, heterogeneous data.**

I. Introduction

In this paper, we consider the problem of scene classification, which is an important issue in many fields such as robotics and Unmanned Aerial Vehicle (UAV). Scene classification is often solved as a machine learning problem. A machine learning-based scene classification system consists of two main components, namely feature extraction and pattern classification. In previous studies, a number of feature extraction methods based on low-level bag-of-features (BoF) [1] have been proposed (e.g., SIFT [2], GIST [3], etc.). Li et al. propose Object bank [4] as high-level image features in terms of the semantic meanings. In recent years, effective sparse coding based Spatial Pyramid Matching (SPM) [5], and its variants such as Sc$^+$SPM [6], LScSPM [7], LR-Sc$^+$SPM [8], and CCLR-Sc$^+$SPM [9] have received considerable attentions. Although substantial progress has been made in scene classification using single feature space, recent research has shown that fusion of multiple feature spaces could significantly improve classification performance. Most existing fusion models are in feature level. Transfer learning methods such as self-taught learning [10] and heterogeneous transfer learning (HTL) [11] have been proven to be effective and helpful. For instance, HTL aims to build a bridge to transfer the knowledge of heterogeneous data from the web. Nevertheless, the feature level fusion method for scene classification has some limitations. It is generally acknowledged that the scene classification performance deteriorates with the increase of semantic difficulty within images. In addition, the performance of some feature level fusion-based models is limited by the latent feature representation as the difference of image properties in terms of shape and appearance may be ambiguous in latent feature space.

Fig. 1. Two similar but different scenes, bocce and croquet.

Suppose we are given two visually similar but semantically different scenes "bocce" and "croquet", as shown in Fig. 1, to classify. The two scenes in the two images are ambiguous to machines since both scenes contain some identical objects: grass, balls, people, etc.. Moreover, the similar motion of the people in the images makes the scene classification even more difficult. Consequently, the classification accuracies of the SPM method for the two classes in a 8-class scene classification problem are as low as 42% and 48% respectively [4]. In fact, classification of similar scenes like those in Fig. 1 is challenging even for humans, especially for those who are not familiar with sport events. But humans naturally have the capability to cope with unseen pictures and recognize the scenes. One may ask: how do human brains outperform machine learning-based approaches in recognizing ambiguous or unseen scenes? This is a complicated question. The superb performance of humans in recognizing ambiguous or unseen scenes is partly due to their remarkable ability to collect and amalgamate knowledge. For example, humans rapidly think of adopting information on the Internet when they not knowing the answer to general-knowledge problems [12]. In particular, people tend to perceive the semantic meanings of new observations by the aid of web resources if possible. External knowledge not only helps reconfirm the original judgements on ambiguous scenes but also enhance the human's capability to recognize even unknown scenes.

Inspired by human cognition [12] and the work in [13], we proposed a new framework for scene classification. The central idea is to use fusion of training images and auxiliary text data extracted from resources on the Internet to improve scene classification. Humans are naturally conscious of how to properly coordinate and use the previous learned knowledge

and the external knowledge in visual recognition. We thereafter imitate the logic by the way of fusion of training data and web knowledge in high level. The new framework trains an additional combiner in addition to the image-based classifier. This combiner adaptively combines the decision scores of the training image-based and the auxiliary text-based classifiers. Thus, the web text-based classification results literally contribute to the original classification performance even after the classifier training is done.

The human logic-based new framework consists of three parts, including training image-based scene classification, web text-based scene classification, and the fusion of the decision scores of the two classifiers. Details of the new framework are described in the following section.

II. A New Framework of Scene Classification

A. Motivation and System Overview

A classifier learned from training data may mis-classify an ambiguous or unseen scene. This is just like the scenario that a man may fail to recognize such scenes if the decision is only dependent on the prior experience/knowledge. However, the man may ultimately recognize the ambiguous or unseen scene by the way of searching and using supplementary materials, such as reference books or web resources. This is because human has remarkable learning ability to collect, analyze, and amalgamate external knowledge. Human logic intentionally searches for relevant text descriptions by browsing the scene alike images in the books or on the Internet. The text descriptions explicitly or implicitly contain some helpful information (e.g., semantic categories). Combining the significant clues found from the books and his original decision according to his prior knowledge of scene fragments, the man is able to update his classification decision, which is more likely to be correct.

Inspired by human cognition, we propose a new framework for scene classification as shown in Fig. 2. In this new framework, classification of a scene is performed in three steps. First, an initial classification is performed by the pattern classifier learned from training images. Second, we search similar images on the Internet, and extract text descriptions of visually similar web images. Auxiliary scene classification will be made based on the text descriptions. Last, a decision-level fusion is performed. The fusion is actually done by applying a pre-trained weight vector to pairwise concatenate image-based and web text-based classification scores. The web text helps disambiguation, which in turn leads to more accurate and robust performance.

B. Training Image-based Scene Classification

As mentioned in Section I, a variety of feature extraction methods have been proposed in the literature. SIFT has been proven to be an effective descriptor capturing texture or appearance features. Using appearance descriptors with the image spatial layout, Bosch et al. [14] proposed a type of fast dense SIFT descriptor: Pyramid Histogram Of visual Words (PHOW) descriptor for appearance.

In this paper, we employ PHOW descriptor as the local image descriptor because of its ease of implementation

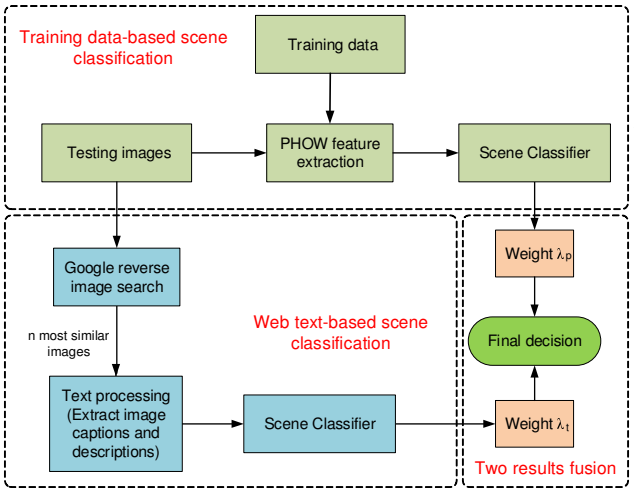

Fig. 2. A flowchart of our framework for improving scene classification. This model can be interpreted as the fusion of training data + web resources. The classification results of a testing image are obtained from training data-based classifier and web resources separately. The weighting parameter λ_p and λ_t integrate the two classification results.

and superior performance. PHOW features are extracted at multiple scales and quantized into M bins of visual words using pyramid-based k-means clustering. In order to use high dimensional discriminative features, we construct a sequence of levels of the pyramid. Suppose that the pyramid level $l = 0, \ldots, L$, and the grid at level l has 4^l cells. Multiplied by the size of visual words, level 0 has M bins of the histogram intersections, level 1 is represented by $4M$ bins and so on. The final concatenated histogram has a dimensionality of $M\sum_{l=0}^{L} 4^l$. After the spatial pyramid representation is formed, we match the image by computing a three dimensional approximated histogram map for the Chi-Square kernel. Thus, each histogram descriptor is expanded into a vector of three dimensional outcome. Consequently, we should acquire $3M\sum_{l=0}^{L} 4^l$ dimensional PHOW features in total.

After extracted from an image, the features are input to a classifier. In this study, we use a linear support vector machine (SVM) [15] classifier. Suppose the labelled training images and unlabeled test images are from h classes, we thereafter train a h-class SVM classifier to derive the decision scores as expected outcomes. In particular, we implement one-vs-rest (one-vs-all) linear SVM. In this approach, h binary classifiers are employed, each of which separates class j, $j = 1, \ldots, h$ from the rest $h-1$ classes. Once the SVM classifier is carefully trained, the PHOW feature vector of a testing image is fed to the h SVM classifiers. The outcome of this testing instance is a h-dimensional vector $\mathbf{p} = [p_1, \ldots, p_h]^T$, where p_j denotes the decision score of this testing sample belonging to the j^{th} class. Here, the classification results refer to the decision scores. In general, the highest score $\max_{1 \le j \le h} \{p_j\}$ reflects the correct class the testing instance belongs to.

C. Web Text-based Scene Classification

In Fig. 1, we have shown two scene examples "bocce" and "croquet", which are visually very similar. The features extracted from the images may not be sufficient to lead to

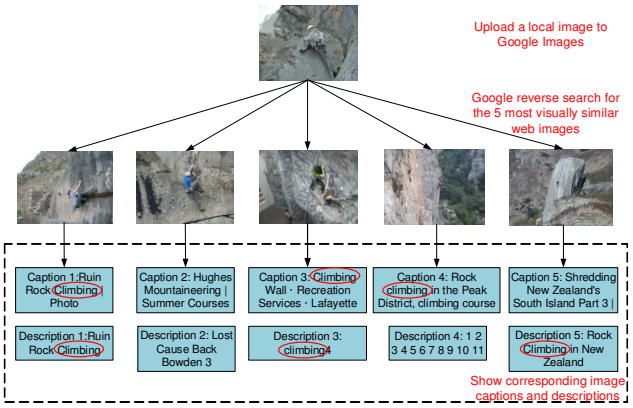

Fig. 3. An example of the procedure of Google reverse image searches results when $n = 5$. It is clear that the label information is in the image caption and description. The label information in the form of text is circled in red.

correct classification. In this study, we propose to incorporate information on the web to help, whenever the trained classifier faces some challenges to assign a class label to a scene. Web images are crawled according to local image query from Google Images (website: http://images.google.com/). Google Images is a web searching service that allows users to search web image contents by image since 2011 (officially entitled *reverse image searches*). Unlike traditional keyword-based image query, the new features involve computer vision algorithms and relieve the dependence of text information. Users search by uploading an arbitrary image as their query.

Given a testing image, we upload this image as a query to search visually similar images on the web using Google Image. Fig. 3 shows the results of the Google Image search. The returned images are sorted based on visual similarity. We therefore retrieve the first n returned images. Fortunately, all these similar images from web resources are already annotated by the web pages or personal uploading. It is observed that the semantic category, i.e. class label of each of the returned images, might be contained in the captions or descriptions of the images, as highlighted by the red cycles in Fig. 3. Thus, the class label of the query image maybe extracted from the tags or descriptions of the returned images. Next, we discuss the procedure of class label extraction.

The class labels are the most interested information that we seek. Class label information can be extracted by using natural language processing (NLP) [16] techniques. Now we take the images in Fig. 3 as examples to explain. The text information including image captions and descriptions of the returned images will be downloaded and stored as n documents. First, the raw text is tokenized into a sequence of alphabetic and non-alphabetic characters. Tokenization is followed by a process of removing morphological affixes from words, called word stemming. For example, snowboard is the word root of snowboarding and snowboarder, and both of them will be automatically transformed to snowboard after stemming. Next, according to the classes of the training images, we make a list of word roots of h image categories. Only the words within the list will be retained in the stemmed documents. Finally, we convert the collection of the documents into a matrix by detecting the presence or absence of the words in the list. We call this process as class label extraction. The extracted class label information is in the form of a matrix $\mathbf{D} = \{d_{ij}\} \in \mathbb{R}^{n \times h}$, where d_{ij} takes value 1 or 0 to denote the presence or absence of class label j in the i^{th} document. The text data processing is summarized as the first three steps in Algorithm 1.

In contrast to the training image-based scene classification, the web text-based scene recognition is performed in a straightforward way. Since the web image-to-label matrix $\mathbf{D}$ contains important class label information, the classification decision scores can be explicitly derived from it. By taking the average of every column of $\mathbf{D}$, we obtain image-to-label vector $\mathbf{t} = [t_1, t_2, \ldots, t_h]^T$ of the query image, where t_j denotes the decision score of the testing (query) image belonging to the j^{th} class. This corresponds to the last step in Algorithm 1. The highest score $\max_{1 \le j \le h} \{t_j\}$ indexes to the class that the testing image belongs to.

Algorithm 1 Web Text Data Processing and Classification

Input: n returned documents of a testing image.
Output: Decision scores vector $\mathbf{t}$ of a testing image.

1: Tokenize raw text into a sequence and then stem the words.
2: Retain the stemmed words in the list of word radicals of h image categories.
3: Detect the occurrence of the list words from the retained documents and save it in the matrix $\mathbf{D} \in \mathbb{R}^{n \times h}$.
4: Compute the decision scores $\mathbf{t}$ of the testing image by averaging the nonzero rows in $\mathbf{D}$

D. Fusion of Two Heterogeneous Components

In the proposed framework shown in Fig. 2, the training image-based classification and web text-based classification are fused at the decision level. The decision scores from the h SVM classifiers and the decision scores extracted from web resources are linearly combined. The linear combination weight vector is learned through K-fold cross validation in the present study.

Assuming there are N labelled training images from h classes. In K-fold cross validation [17], the N training images are partitioned into K parts. In each of the K repeats, one part is used for validation, and $k-1$ parts are used for training. After completion of the K repeats, each of the N training images has been used once as a validation image.

Assuming that the weight vectors for training image-based decision scores and web text-based decision scores are denoted by $\lambda_p = [\lambda_{p1}, \ldots, \lambda_{ph}]^T$ and $\lambda_t = [\lambda_{t1}, \ldots, \lambda_{th}]^T$ respectively, where λ_{pi} and λ_{ti} denote the two weights of class i. Different classes use different weights because the reliability of training data and web resources of different classes might be different.

As mentioned above, in K-fold cross validation, each of the training data is used once as a validation data. When used as a validation, the decision scores of image i are denoted by $\mathbf{p}_i = [p_{i1}, \ldots, p_{ih}]^T$ and $\mathbf{t}_i = [t_{i1}, \ldots, t_{ih}]^T$ respectively. The fusion result is a linear combination of $\mathbf{t}_i$ and $\mathbf{p}_i$:

$$\mathbf{q}_i = [q_{i1}, \ldots, q_{ih}]^T, \quad (1)$$

where

$$q_{ij} = p_{ij}\lambda_{pj} + t_{ij}\lambda_{tj}, \qquad (1 \le i \le N). \tag{2}$$

Weight vectors λ_p and λ_t should make $\mathbf{q_i}$ be close to its target $\mathbf{y}_i \in \mathbb{R}^h$, which is a column vector with value 1 at the c^{th} position and 0 at other positions, and c is the class label of image i. λ_p and λ_t can be obtained by minimizing the following cost function:

$$J(\lambda_p, \lambda_t) = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{q}_i - \mathbf{y}_i\|^2. \tag{3}$$

By Defining

$$\mathbf{X} = \begin{bmatrix} p_{11} & t_{11} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & p_{12} & t_{12} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & p_{Nh} & t_{Nh} \end{bmatrix} \tag{4}$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_N \end{bmatrix} \tag{5}$$

and parameter vector $\mathbf{w} = [\lambda_{p1}, \lambda_{t1}, \dots, \lambda_{ph}, \lambda_{th}]^T$, we can rewrite Eqn (3) as:

$$J(\mathbf{w}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 \tag{6}$$

Setting the gradient of (6) with respect to $\mathbf{w}$ to zero, yields:

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \tag{7}$$

Since $\mathbf{w}$ is the combined column vector of λ_p and λ_t parameters alternating with each other, the weight vectors λ_p and λ_t are retrieved respectively by simply reshaping $\mathbf{w}$.

Given a testing image, the classification outcomes from the training image-based classifiers and web resources are first obtained: $\mathbf{p}' = [p'_1, \dots, p'_h]^T$, $\mathbf{t}' = [t'_1, \dots, t'_h]^T$. The decision scores are then combined using the learned weight vectors λ_p and λ_t:

$$\mathbf{q}' = \left[\lambda_{p1}p'_1 + \lambda_{t1}t'_1, \dots, \lambda_{ph}p'_h + \lambda_{ph}t'_h\right]^T \tag{8}$$

The classification decision is finally made by the highest decision score in $\mathbf{q}'$

$$c' = \arg\max_{1 \le j \le h} q'_j \tag{9}$$

The above fusion procedure is summarized in Algorithm 2.

III. Experiments and Results

In this section, we assess our image data and web resources fusion model for scene classification using the benchmark UIUC-Sport events dataset [18]. In order to verify the effectiveness of our framework, we measure the experimental results and compare our classification performance with other state-of-the-art methods.

Algorithm 2 Fusion System Formulation

Input: N labelled training image data, decision scores matrix $\mathbf{T}$ based on the web text of training data, a testing image, decision scores vector $\mathbf{t}'$ based on the web text of testing data.
Output: The classification decision c' of a testing image.

1. Conduct a K-fold cross validation on N training data.

2. **while** k **not exceed** K

   1: Perform a classifier on N/K training images and obtain the decision scores matrix $\mathbf{P}^{(k)}$.

   2: Repeat computing for other $K-1$ trails.

3. **end while**

4. Construct (4) using $\mathbf{P}$ and $\mathbf{T}$ and evaluate (6).

5. Solve the least squares estimate and obtain the weight vector $\mathbf{w}$ denoted by (7).

6. Perform a classifier on the testing images and obtain the decision scores vector $\mathbf{p}'$.

7. Fuse $\mathbf{p}'$ and $\mathbf{t}'$ and retrieve the decision scores vector $\mathbf{q}'$ by applying λ_p and λ_t.

8. The final classification decision c' is made by the highest score in $\mathbf{q}'$.

A. Dataset and Experiment Setup

The UIUC-Sport dataset contains 8 sports event categories: rock climbing (194 images), badminton (200 images), bocce (137 images), croquet (236 images), polo (182 images), rowing (250 images), sailing (190 images), and snowboarding (190 images). The image number in each class ranges from 137 to 250, and there are 1579 images in total. It is noted that the difficulty levels of classification within a category are varying with the distance of the foreground objects. Fig. 4 shows some example images in the dataset.

The auxiliary unlabelled images are searched through Google Images by using the images as the query. We upload every local image and search for the visually similar web images. Note that these images are found and ranked according to the similarity to query images. In addition, we filter out images whose sizes exceed 800×600 pixels. In this work, we extract text descriptions of the first 5 returned web images for each training and testing image, and this results in a total collection of 7895 auxiliary text documents.

As discussed in Section II-B, we utilize PHOW features to describe the images. To begin with, an image is downsized to 640×480 pixels. Next, SIFT descriptors of 6×6 pixels are computed over regular grids spacing of 4 6 8 10. The pyramid level number is set to 2. We quantize the pyramid histogram vectors into a typical 200 visual words in k-means clustering. After matching procedure, we obtain 12600 dimensional PHOW features for a single image. Adding image label vector given by the UIUC-Sport dataset, the image dataset is now represented by a matrix $\mathbf{I} \in \mathbb{R}^{1579 \times 12601}$. We implement PHOW descriptor extraction via VLFeat toolbox [19]. On the other hand, we now have 7895 unlabelled web image descriptions. Using the text information retrieval method

Fig. 4. Example images for the UIUC-Sport dataset.

discussed in Section II-C, we derive the target image-to-label matrix $\mathbf{T} \in \mathbb{R}^{1579\times 8}$. The text analytics is implemented by the natural language toolkit (NLTK) [20] and Scikit machine learning toolkit [21].

The event recognition task is an 8-class classification problem. Following the experiment setting of [18], 70 local images are randomly selected for training and we test on 60 images. Accordingly, we split every web instances of $\mathbf{T}$ into the corresponding training and test subsets. In order to achieve statistically significant experimental results, we repeat 50 times of the training/ test data random split process and present the averaged results. In this paper, we conduct three experiments and report their results. In the first experiment, we implement linear SVM as the multi-class classifier on $\mathbf{I}$ alone using LIBSVM [22], where the hyperparameter C is set to default value, and b is set to 1. In the second experiment, we use the proposed fusion framework to combine classification results from training data and web resources. In the third experiment, we extend the proposed framework to other types image feature extraction methods to test the effectiveness of the proposed framework.

B. Results

In this section, we show the experimental results. We report the averaged overall accuracies of 50 trials. Table I and Table II show the confusion tables of the first and the second experiments. In the confusion matrix, the rows represent the instances of actual scene categories, while each column denotes the instances of predicted scene categories.

For the training data-based approach using PHOW features only, we obtain an overall accuracy of $83.95 \pm 1.11\%$. In contrast, a remarkable improvement has been achieved with an overall accuracy of $88.19 \pm 1.25\%$ by using our fusion framework. It is observed that categories bocce and croquet confuse the most in both confusion tables. This phenomenon is in line with all previous reported works on UIUC-Sport event dataset. The similarity of the foreground objects shared in this two categories causes substantial difficulties in the scene discrimination. Moreover, the performance of category badminton

TABLE I. CONFUSION MATRIX FOR THE UIUC-SPORT EVENT RECOGNITION EXPERIMENT WITH LOCAL IMAGE DESCRIPTOR ALONE. THE AVERAGE ACCURACY AND THE STANDARD DEVIATION ARE 83.95% AND 1.11%, RESPECTIVELY.

	rockclimbing	*badminton*	*bocce*	*croquet*	*polo*	*rowing*	*sailing*	*snowboarding*
rockcliming	.93	0	.02	.01	0	.01	0	.3
badminton	0	.92	.02	.02	.01	.01	.01	.02
bocce	.04	.04	.64	.15	.06	.02	0	.05
croquet	.02	0	.15	.76	.03	.01	.01	.01
polo	.01	.02	.03	.03	.85	.02	.01	.03
rowing	.01	.01	.02	.01	.02	.88	.02	.02
sailing	0	0	.01	.02	.01	.04	.91	.01
snowboarding	.05	.01	.05	.01	.02	.03	.01	.81

TABLE II. CONFUSION MATRIX FOR THE UIUC-SPORT EVENT RECOGNITION EXPERIMENT WITH FUSION FRAMEWORK. THE AVERAGE ACCURACY AND THE STANDARD DEVIATION ARE 88.19% AND 1.25%, RESPECTIVELY.

	rockclimbing	*badminton*	*bocce*	*croquet*	*polo*	*rowing*	*sailing*	*snowboarding*
rockcliming	.97	0	0	.01	0	.01	0	.01
badminton	0.01	.88	.07	0	.01	.01	.01	.02
bocce	.03	.03	.66	.13	.07	.02	0	.05
croquet	.02	0.01	.14	.77	.03	0	.01	.02
polo	.01	.01	.02	.02	.91	.01	0	.02
rowing	0	0	0	0	0	.96	.02	0
sailing	0	0	0	0.01	0	.01	.98	0
snowboarding	.02	.01	.02	0	.01	.01	.01	.90

TABLE III. PERFORMANCE COMPARISON OF THE STATE-OF-THE-ART SCENE RECOGNITION ALGORITHMS WITH/ WITHOUT OUR FUSION FRAMEWORK

Algorithm	Image feature only (%)	Fusion (%)
GIST [3]	64.15 ± 1.95	$\mathbf{77.44 \pm 1.94}$
Sc$^+$SPM [6]	80.28 ± 0.93	$\mathbf{85.91 \pm 0.92}$
Object bank [4]	77.87 ± 0.91	$\mathbf{86.98 \pm 1.01}$
PHOW	83.95 ± 1.11	$\mathbf{88.19 \pm 1.25}$

degrades to 88.13% as shown in Table II, and it is the only category that is not improved in our fusion experiment. We found that most badminton images in the UIUC-Sport dataset have the background of multipurpose indoor courts, while the backgrounds of the Google images badminton collection are mostly exclusive badminton courts. This mismatch may cause inaccurate web query results, and this in turn leads to deterioration of fusion performance. On the other hand, we observe considerable improvements of our model in other event categories as shown in Table I and II. Among these categories, the accuracy of rockclimbing and sailing are the highest, at 97.11% and 97.94% respectively. Furthermore, the results of rowing and snowboarding categories are improved by 8% and 9% respectively. These results indicate that our method has good noise immunity to learn the contextual semantic meaning within the images.

In Table III, we show the classification results of other state-of-the-art feature descriptors when they are used alone or combined with web resources under the proposed fusion framework. The boldfaced numbers denote the performance with the web resources aid. Obviously, significant improvements are achieved in all the 4 feature extraction methods. In addition, it is noted that the most recent and best result on the benchmark dataset reported in the literature is CCLR-Sc$^+$SPM [9] with an overall accuracy of $87.75 \pm 1.29\%$. Our training data and web resources fusion framework produces even better results with less computational complexity in terms of the dimensionality of the feature space. The experiment results indicate that the proposed framework is a general architecture, any image feature extraction methods can be used.

IV. CONCLUSION

In this paper, we have proposed a novel framework that uses both training data and web resources for scene classification. Experimental results on the benchmark dataset show that the proposed fusion framework could significantly improve classification performance. This fusion framework imitates human way of learning by using external knowledge, and hence has a sound cognitive basis. Experimental results also show that the proposed framework is a general architecture, under which any feature extraction method can be used to combine with web resources to improve performance. Exploration of the proposed framework in other applications is undergoing, and results will be reported in our future publications.

ACKNOWLEDGMENT

This work is based on research supported by the Asian Office of Aerospace Research and Development (AOARD) under the award number FA2386-13-1-4083.

REFERENCES

[1] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, no. 1-22, 2004, pp. 1–2.

[2] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[3] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[4] L.-J. Li, H. Su, Y. Lim, and L. Fei-Fei, "Objects as attributes for scene classification," in *Trends and Topics in Computer Vision*. Springer, 2012, pp. 57–69.

[5] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2. IEEE, 2006, pp. 2169–2178.

[6] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1794–1801.

[7] S. Gao, I. W. Tsang, L.-T. Chia, and P. Zhao, "Local features are not lonely–laplacian sparse coding for image classification," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3555–3561.

[8] C. Zhang, J. Liu, Q. Tian, C. Xu, H. Lu, and S. Ma, "Image classification by non-negative sparse coding, low-rank and sparse decomposition," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1673–1680.

[9] C. Zhang, J. Liu, C. Liang, Z. Xue, J. Pang, and Q. Huang, "Image classification by non-negative sparse coding, correlation constrained low-rank and sparse decomposition," *Computer Vision and Image Understanding*, vol. 123, pp. 14–22, 2014.

[10] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: transfer learning from unlabeled data," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 759–766.

[11] X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu, "Transfer learning on heterogenous feature spaces via spectral transformation," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 1049–1054.

[12] B. Sparrow, J. Liu, and D. M. Wegner, "Google effects on memory: Cognitive consequences of having information at our fingertips," *science*, vol. 333, no. 6043, pp. 776–778, 2011.

[13] S. Gao, Z. Wang, L.-T. Chia, and I. W.-H. Tsang, "Automatic image tagging via category label and web data," in *Proceedings of the international conference on Multimedia*. ACM, 2010, pp. 1115–1118.

[14] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," *In Proc. ICCV*, 2007.

[15] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[16] P. Spyns, "Natural language processing," *Methods of information in medicine*, vol. 35, no. 4, pp. 285–301, 1996.

[17] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *IJCAI*, vol. 14, no. 2, 1995, pp. 1137–1145.

[18] L.-J. Li and L. Fei-Fei, "What, where and who? classifying events by scene and object recognition," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[19] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[20] S. Bird, "Nltk: the natural language toolkit," in *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, 2006, pp. 69–72.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[22] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.