A Classifier Ensemble Framework for Multimedia Big Data Classification

Yilin Yan[1], Qiusha Zhu[2], Mei-Ling Shyu[1], and Shu-Ching Chen[3]
[1]Department of Electrical and Computer Engineering
University of Miami
Coral Gables, Florida 33146, USA
[2]Senzari, Inc.
601 Brickell Key Drive
Miami, Florida 33131, USA
[3]School of Computing and Information Sciences
Florida International University
Miami, Florida 33199, USA
Emails: {y.yan4, q.zhu2}@umiami.edu, shyu@miami.edu, chens@cs.fiu.edu

Abstract

Numerous classification algorithms have been developed for a variety of data types. However, it is nearly impossible for one classifier to perform the best in all kinds of datasets. Therefore, ensemble learning models which aim to take advantages of different classifiers have received a lot of attentions recently. In this paper, a scalable classifier ensemble framework assisted by a set of judgers is proposed to integrate the outputs from multiple classifiers for multimedia big data classification. Specifically, based on the confusion matrices of different classifiers, a set of "judgers" are organized into a hierarchically structured decision model. A testing instance is first input to different classifiers, and then the classification results are passed to the proposed hierarchical structured decision model to derive the final result. The ensemble system can be run on Spark, which is designed for big data processing. Experimental results on multimedia data containing different actions demonstrate that the proposed classifier ensemble framework outperforms several state-of-the-art model fusion approaches.

Keywords: *Multimedia Big Data, Classifier Ensemble Framework, Multi-classifier Fusion, Spark, Ensemble learning*

1. Introduction

With the rapid development of social media websites, we have witnessed the exponential growth of multimedia data like images and videos on the Internet. As a result, the content-based multimedia data management and retrieval become an important research area [1][2][3][4][5][6][7][8]. For example, video content analysis, in the context of automatically analyzing human actions in videos, receives a lot of attentions due to its broad applications in camera surveillance, video summarization, and video event mining. The deluge of multimedia data in the current big data era has made the data-oriented frameworks more and more popular [9][10][11][12][13][14][15][16]. In this paper, we propose a data-mining-based framework to address the issue of multimedia big data classification.

In multimedia information systems, multi-classifier fusion is an important research area because a single classifier can hardly handle heterogeneous media types from different datasets in various situations. Gradient histograms using orientation tensors for human action were developed in [17]. [18] proposed a fusion based classification system using statistical fusion such as GMM (Gaussian Mixture Model) fusion and ANN (Artificial Neural Network) fusion. Conflict results can be generated by different classifiers and the general idea to solve this is to find a way to fuse the results from different classification models. Previous results have indicated that the fusion of multiple different results can improve the performance of individual classifiers.

Another research topic in multimedia data is how to utilize multiple features from different feature extraction methods [19][20][21][22][23][24][25][26]. In [27], the authors found the complementary nature of the descriptors from different viewpoints, such as semantics, temporal and spatial resolution. They also employed a hierarchical fusion that combines static descriptors and dynamic descriptors. In [28], textual features were shown to provide high-level semantics that are sometimes difficult to be captured by visual features and a sparse linear fusion scheme was proposed to combine visual and textual features for semantic concept retrieval and data classification. Different classification frameworks can be employed for different kinds of features, which may discover different properties of the data [29][30].

In this paper, a novel idea of classifier combination is proposed. At the first step, several "judgers" are generated based on training and validation results from different classifiers and features. Next, these judgers are ranked and put together as a boosted classifier. Finally, a Spark-based classification system is developed which can be applied for multimedia big data classification. Our proposed system has three main contributions. First, the concepts of positive and negative "judgers" are defined to assemble a novel hierarchical structured decision model. Second, it considers the fusion of classifiers using the same features and the fusion of classifiers in different feature spaces simultaneously. Third, a unified ensemble fusion framework in a big data infrastructure is developed and uses action classification in videos as a proof-of-concept. Experimental results show promising results by comparing to several state-of-the-art methods.

This paper is organized as follows. In section 2, the related work in multi-classifier fusion is introduced. Section 3 presents our proposed framework. In section 4, two benchmark action datasets are used for evaluation. Finally, section 5 summarizes this paper and offers concluding remarks.

2. Related Work

The existing work on the fusion of multiple classification models generally falls into four categories, which are weighted combination strategy, statistics-based strategy, Bayesian probabilistic strategy, and regression-based strategy, respectively.

The weighted combination strategy is commonly used in multi-classifier fusion. The sum and product rules are two popular weighted combination approaches, which treat the sum and product as the arithmetic mean and geometric mean, respectively. It was suggested that the product rule is good when the individual classifiers are independent, and the sum rule is equivalent to the product rule for small deviations in the classifier outcomes under the same assumption [31]. The sum and product rules can be generalized to the weighted combination rules. The general rule of weighted combination for different scores is to find a set of weights for different scores generated by different classifiers. Therefore, the key to this strategy is to determine the weights. Many different strategies were proposed in the literature. For example, an information gain method for assigning weights is explained in [32]. Meng et al. [33] utilized the normalized accuracy to compute the weights for each model built on a specific image patch. In a recent study proposed in [34], the researchers further extended this method by first sorting all the models according to interpolated the average precision and then selecting the models with top performance. The number of models to retain in the final list is determined via an empirical study. Experimental results indicate that this strategy gives a relatively good performance. However, the success of this kind of approaches is determined by the proper choice of weights. Therefore, it relies on specific knowledge from domain experts or experience from data mining researchers to provide a good estimation of weights.

Four commonly used approaches in statistics-based approaches are "min", "max", "sum", and "median" rules. The "min" fusion approach is a relatively conservative estimation, where the lowest score of all the models is chosen. On the other hand, the "max" fusion strategy picks the highest value. The "sum" strategy actually gives an estimation of the final score based on the majority-voting theory. It is actually equivalent to the weight combination strategy by setting all the weights to the same value. The "median" rule gets the median value of all the scores. Kuncheva [35] evaluated the advantages and disadvantages of these strategies from a theoretical point of view [36]. The main advantage of the statistics-based approach is the low time complexity, while the main issue is that the performance of these models is not quite stable under the condition that the underlying models are not accurate.

The Bayesian theory is also widely used in multi-classifier fusion, and sometimes is combined with other strategies. The final score is computed using the Bayesian rule based on all the scores from the models. This method relies on a strong assumption that the scores are conditionally independent with each other [37]. However, this assumption does not hold under most circumstances. Dempster–Shafer theory is an improved method of the Bayesian theory for a subjective probability. It is also a powerful method for combining measures of evidence from different classifiers [38]. In practice, this kind of approaches may give relatively bad performance because of the severe deviation from the independence assumption.

Recently, the regression-based strategy receives a lot of attentions. In this research direction, the logistic regression based model is commonly utilized. Parameters are estimated using the gradient descent approach in the training stage. After the parameters are learned, the score of a testing data instance can be computed. In [39], the logistic regression model is trained to integrate the output scores for different classification models to get a final probabilistic score for the target concept. In practice, the logistic regression model gives relatively robust performances while sometimes suffering from the overfitting issue.

3. The Proposed Framework

3.1. Feature extraction

In this paper, the feature extraction approach proposed in [20] is utilized. Different from other popular frameworks that extract features from the whole image frame, features are extracted from the Region Of Action (ROA) in order to capture the action related information. In addition, it analyzes and integrates the motion information of actions in both spatial and temporal domains. The steps of this approach are summarized as follows.

First, the ROA is driven from two spatiotemporal methods, namely optical flow and Harris3D corners. Next, the idea of integral image in [40] is utilized for its fast implementation of the box type convolution filters. Then the Gaussian Mixture Models (GMM) are applied sequentially in this paper. Two popular features, SIFT [41] and STIP [42], are used in our work to describe the action video sequences in action recognition.

3.2. Classification

Two classification models are used in this paper, which are Sparse Representation Classification (SRC) and Hamming Distance Classification (HDC). Although these two classifiers are chosen for this paper, the proposed classifier ensemble framework accepts all kinds of classifiers.

Sparse representation [43] is a technique to build an overcomplete dictionary to represent the target. With a learned dictionary, signals can be decomposed as sparse linear combinations of these atoms. It has been applied to various applications including compression, regularization in inverse problems, feature extraction, object detection, denoising, etc. In this paper, we utilize the sparse representation and dictionary learning techniques to design a framework to analyze action events between multiple people. For the task of dictionary learning, the widely-used *K-SVD* algorithm is adopted, which aims to derive the dictionary of sparse representation using the Singular Value Decomposition (SVD). After the dictionary is trained, the class label of a testing sample can be determined by finding the minimum reconstruction error of the testing sample represented by the trained dictionary.

A novel HDC scheme is also employed in classification [44][45], which is an efficient method for real-time applications. For each class, a threshold will be calculated by the median value of the inner products of each pair of features in the training set. Given a testing sample, a binary string can be coded based on the inner product between the testing sample and each training sample. If the inner product of the testing sample and a training sample is greater than the trained threshold, it would be coded as *1*. Otherwise, it would be assigned to *0*. Then, the class of the testing sample can be decided by the mean value of the hamming distance between the coded testing sample and each class.

3.3. Multi-classifier

Assume we have a sample **x**, where **x** is a *d*-dimensional feature vector. Let $\omega_1, \omega_2, \cdots, \omega_M$ be *M* categories, and $\alpha_1, \alpha_2, \cdots, \alpha_M$ be a finite set of possible actions. Suppose we have totally *N* classifiers, namely $c_1, c_2, \cdots, c_N$. Each classifier will generate a posterior probability $P_{c_n}(\omega_j | \mathrm{x})$ for **x**. Here, we define a loss function $\lambda(\alpha_i | \omega_j)$ which describes the loss occurred for taking action α_i when the state of nature is ω_j. Obviously, we can get a set of posterior probabilities used for classification generated by different classifiers $P_{c_1}(\omega_j | \mathrm{x}), P_{c_2}(\omega_j | \mathrm{x}), \cdots, P_{c_n}(\omega_j | \mathrm{x})$.

For each probability function, the expected loss associated with taking action α_i is defined in Equation (1).

$$R_{c_n}(\alpha_i | x) = \sum_{j=1}^{M} \lambda(\alpha_i | \omega_j) P_{c_n}(\omega_j | x) \tag{1}$$

As a classification problem, a zero-one loss function is defined in Equation (2). Thus, for each classifier, the condition risk for category ω_j is defined in Equation (3). *R* needs to be minimized to achieve the best performance for a certain classifier c_n using Equation (4).

$$\lambda(\alpha_i | \omega_j) \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases}, \quad i, j = 1, 2, \cdots, M \tag{2}$$

$$R_{c_n}(\alpha_i | x) = \sum_{j \neq i} P_{c_n}(\omega_j | x) = 1 - P_{c_n}(\omega_i | x) \tag{3}$$

$$\begin{aligned} R_{c_n} &= \int_{x \in \Omega} R_{c_n}(\alpha | x) p(x) dx \\ &= \int_{x \in \Omega} \left[1 - P_{c_n}(\omega | x) \right] p(x) dx \end{aligned} \tag{4}$$

Considering *N* different classifiers, most previous fusion methods use a certain algorithm to fuse different $P_{c_n}(\omega_j | x)$ for *M* categories. For example, using the weighted combination rules, we can generate a combined posterior and a new conditional risk *R* using Equations (5) and (6).

$$P_{fusion}(\omega_j | x) = \sum_{n=1}^{N} w_n P_{C_n}(\omega_j | x); \tag{5}$$

$$\begin{aligned} & R_{fusion} = \int_{x \in \Omega} \left[1 - P_{fusion}(\omega | x) \right] p(x) dx \\ & = \int_{x \in \Omega_1} \left[1 - P_{fusion}(\omega_1 | x) \right] p(x) dx \\ & + \int_{x \in \Omega_2} \left[1 - P_{fusion}(\omega_2 | x) \right] p(x) dx \\ & + \cdots \\ & + \int_{x \in \Omega_M} \left[1 - P_{fusion}(\omega_M | x) \right] p(x) dx \\ & \textit{where } \Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_M = \Omega \\ & \textit{and } \Omega_i \cap \Omega_j = \phi \ (i \neq j,\ i, j = 1, 2, \cdots, M) \end{aligned} \tag{6}$$

As mentioned earlier, the problem of such fusion approaches is that we can only fuse the “good” classifiers that perform well in all categories. Using a “bad” classifier may not lead to better results and eventually reduces the performance in most of the time. However, a “bad” classifier may perform well for a certain class. Even if it is not a good classifier for all the other classes, our proposed framework can still use it to enhance the result. The main reason is that a classifier is split into different “judgers”, with each judger working independently to determine the label of a testing instance. Therefore, the conditional risk can be reduced by using different posteriors for different classes, as shown in Equation (7).

$$\begin{aligned} R_{\min} &= \int_{x\in\Omega_1} [1-P_{\max}(\omega_1 \mid x)]\, p(x)dx \\ &+ \int_{x\in\Omega_2} [1-P_{\max}(\omega_2 \mid x)]\, p(x)dx \\ &+ \cdots \\ &+ \int_{x\in\Omega_M} [1-P_{\max}(\omega_M \mid x)]\, p(x)dx \end{aligned} \quad (7)$$

3.4. Judgers generation

The ensemble model will split a dataset into three parts, namely a training dataset, a validation dataset, and a testing dataset. After the classification models are trained using the training dataset, we can calculate the precision and recall on the corresponding validation dataset. In our proposed framework, we define precision as a positive judger and recall as a negative judger, where *TP* stands for the true positive value, and *FP* and *FN* are the false positive and false negative values, respectively.

$$J_{pos} = precision = \frac{TP}{TP+FP}$$

$$J_{neg} = recall = \frac{TP}{TP+FN}$$

Formally, suppose there are M classes in a certain dataset. For one type of features f_l ($l \in [1, L]$, L is the number of feature descriptors, which is two in this paper), based on the classification results on the validation dataset, $2\times M$ judgers will be generated for a certain classifier c_n ($n \in [1, N]$, N is the total number of the classification models built on one type of features) as follows:

$$J^{n,l}_{pos_1}, J^{n,l}_{pos_2}, J^{n,l}_{pos_3}, \cdots J^{n,l}_{pos_m} \cdots, J^{n,l}_{pos_M}$$

$$J^{n,l}_{neg_1}, J^{n,l}_{neg_2}, J^{n,l}_{neg_3}, \cdots J^{n,l}_{neg_m} \cdots, J^{n,l}_{neg_M}$$

where $J^{n,l}_{pos_m}$ is a positive judger generated by classifier c_n using feature f_l for the class ω_m ($m \in [1, M]$, M is the total number of categories). Correspondingly, $J^{n,l}_{neg_m}$ is a negative judger. If $J^{n,l}_{pos_m}$ is high, it indicates that this judger is relatively accurate. Accordingly, if it judges a testing instance as in class ω_m, it is highly likely that this judgment is correct. On the other hand, if $J^{n,l}_{neg_m}$ is high, it can be considered as a good negative judger since if classifier c_n does not label a testing instance as class ω_m, the ground truth of the instance is not likely to be ω_m. These judgers form the committee to give the final classification results.

As there are L types of features and N classification models built on each type of features, the total number of judgers is $2\times M\times L\times N$. For example, for the KTH dataset [30] used in this study, there are 6 classes. Therefore, the total number of judgers is $2\times 6\times 2\times 2 = 36$. The next question is how to combine the outputs from different judgers to draw the final conclusion. To solve this problem, a novel classifier ensemble framework is proposed in the following section.

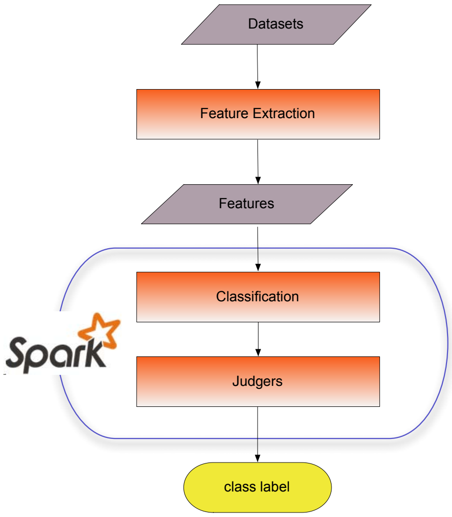

Figure 1. The proposed classifier ensemble framework

3.5. The classifier ensemble framework

Based on all these judgers, a novel efficient classifier ensemble framework is proposed as shown in Figure 1. Given a testing instance **x**, each classifier will assign **x** a set of positive and negative judgers on each feature space. Suppose we have the following list of judgers:

$$J^{11}_{pos_1}, J^{11}_{neg_1}, J^{11}_{pos_2}, J^{11}_{neg_2}, \cdots J^{21}_{pos_1}, \cdots J^{12}_{pos_1}, \cdots J^{22}_{pos_M}, J^{22}_{neg_M}$$

These judgers will be re-ranked by their accuracies. If the highest positive judger ranks first and assigns **x** as class ω_m , **x** will be determined as class ω_m . If the highest negative judger ranks first and decides that **x** cannot be class ω_m, then the next judger will be considered. Even if the next positive judger assigns **x** to class ω_m , it will not be labeled as class ω_m because there exists a negative judger that ranks higher previously. The next judger will be considered until a positive judger assigns **x** to a class without a higher-ranked negative judger rejecting it. The results of this particular testing instance are shown as below:

$$J^{22}_{neg_1}, J^{11}_{pos_1}, J^{12}_{pos_4}, J^{11}_{neg_2}, J^{21}_{pos_3}, J^{21}_{neg_2}, \cdots\cdots$$

In this example, our model assigns **x** to class 4. The same process will be done on all the testing instances.

3.6. The classifier ensemble system on Spark

Apache Spark$^{\text{TM}}$ is a fast and general engine for large-scale data processing and has been deployed for many popular big data systems. Using Spark, an efficient system for classifier ensemble in the scope of big data is built. Spark Core is the foundation of the overall project. It provides distributed task dispatching, scheduling, and basic I/O functionalities. The fundamental programming abstraction is called Resilient Distributed Datasets (RDDs), a logical collection of data partitioned across machines. RDDs can be created by referencing the datasets in external storage systems, or by applying coarse-grained transformations (e.g., map, filter, reduce, and join) on the existing RDDs. The RDD abstraction is exposed through a language-integrated API in Java, Python, Scala, and R similar to local, in-process collections. This simplifies the programming complexity because the way how the applications manipulate RDDs is similar to how they manipulate the local collections of data.

In order to build a better multimedia big data classification framework using Spark, we first read the keys (sample ID) and values (scores from different classifiers for different features) as follows. The meanings and notations are the same as introduced in the previous sections.

$$Key = sample_1,\ Value = (s_1^{c_1,f_1,\omega_1}, s_1^{c_1,f_1,\omega_2} \cdots s_1^{c_2,f_1,\omega_1} \cdots s_1^{c_N,f_L,\omega_M})$$
$$Key = sample_2,\ Value = (s_2^{c_1,f_1,\omega_1}, s_2^{c_1,f_1,\omega_2} \cdots s_2^{c_2,f_1,\omega_1} \cdots s_2^{c_N,f_L,\omega_M})$$
$$Key = sample_3,\ Value = (s_3^{c_1,f_1,\omega_1}, s_3^{c_1,f_1,\omega_2} \cdots s_3^{c_2,f_1,\omega_1} \cdots s_3^{c_N,f_L,\omega_M})$$
$$\cdots$$
$$Key = sample_P, Value = (s_P^{c_1,f_1,\omega_1}, s_P^{c_1,f_1,\omega_2} \cdots s_P^{c_2,f_1,\omega_1} \cdots s_P^{c_N,f_L,\omega_M})$$

The output values are the classification results. Though the dataset used in the experiment is a medium-sized one, the proposed Spark-based system can handle larger datasets. For a larger dataset, more key-value pairs will be created and thus the system can help more in terms of efficiency compared to the traditional classifier ensemble frameworks.

4. Experimental Results

The proposed classifier ensemble framework was tested on a widely accepted benchmark action dataset called KTH dataset [46]. The experimental results are shown as follows.

The KTH dataset contains six kinds of human actions (walking, jogging, running, boxing, hand waving, and hand clapping). Each type of actions was performed several times by 25 subjects in four different scenarios, resulting in 600 video sequences in total. The video sequences were recorded in a controlled setting with slight camera motion and a simple background. All sequences are in the “avi” format and available online.

Table 1. The confusion matrix of six action categories in the KTH data set

	Box	Clap	Wave	Jog	Run	Walk
Box	99	0	0	0	0	1
Clap	1	98	0	0	0	1
Wave	2	3	95	0	0	0
Jog	0	0	0	88	11	1
Run	0	0	0	7	93	0
Walk	1	0	0	0	1	98

In the experiment, we utilized 25-fold cross validation. Two classifiers, namely SRC and HDC were adopted, and the popular SIFT and STIP features were used. Table 1 shows the confusion matrix of our results. In order to fully evaluate the proposed framework, it is compared to 5 state-of-the-art frameworks. Multi-classifier on SIFT and Multi-classifier on STIP apply the multi-classifier as described in Section 3.4 on SIFT and STIP features, respectively. The comparisons of accuracies are shown in Table 2.

Table 2. Comparison of overall average precision of our method and state-of-the-art methods on the KTH dataset

Method	Average precision
Schuldt et al. [46]	71.5%
Dollar et al. [47]	80.7%
Yin et al. [48]	82.0%
Niebles et al. [49]	91.3%
Our work on SIFT	**92.7%**
Our work on STIP	**94.3%**
Our work on both features	**95.2%**

It can be seen that the proposed framework outperforms the other ones. Although the result by only one kind of feature descriptors may not perform better than all other methods, our proposed framework can fuse different kinds of features to achieve a better performance than the other methods. In addition, the proposed fusion strategy is also compared with 6 other fusion strategies in terms of accuracy and the results are given in Table 3. The arithmetic mean and geometric mean represent the sum and product results of the scores. There are two kinds of hybrid means. One way is first to calculate the arithmetic mean scores among different classifiers based on the same kind of features and then to compute the geometric mean scores between different kinds of features. The other way is to do the opposite.

Table 3. Comparison of our classifier ensemble framework and other fusion algorithms on the KTH dataset

Algorithm	Average precision
Arithmetic mean	90.7%
Geometric mean	90.0%
Hybrid mean (Arithmetic for different features)	90.7%
Hybrid mean (Geometric for different features)	90.3%
Linear regression on SIFT	90.5%
Linear regression on STIP	91.2%
The proposed framework	**95.2%**

5. Conclusions and Future Work

In the paper, we propose a novel classifier ensemble framework to fuse classification results generated from different classifiers using different features. As a proof-of-concept, the proposed framework is applied on categorizing human actions in videos. Experimental results show that the proposed framework is capable of taking advantages of different classifiers and outperforms some existing state-of-the-art approaches. Although the experimental dataset is a medium-sized one, the proposed framework can handle big datasets as it is integrated with Spark.

It is also important to note that the proposed framework can be easily extended for other multi-class classification problems with big datasets. Hence, it can be applied to other research areas. If more classifiers are included, more judgers will be generated correspondingly, which can potentially lead to an even better performance.

6. Acknowledgment

For Shu-Ching Chen, this research is partially supported by DHS's VACCINE Center under Award Number 2009-ST-061-CI0001 and NSF HRD-0833093, HRD-1547798, CNS-1126619, and CNS-1461926.

REFERENCES

[1] L. Lin and M.-L. Shyu, “Weighted association rule mining for video semantic detection,” *International Journal of Multimedia Data Engineering and Management*, vol. 1, no. 1, pp. 37-54, 2010.

[2] Q. Zhu, L. Lin, M.-L. Shyu, and S.-C. Chen, “Feature selection using correlation and reliability based scoring metric for video semantic detection,” in *Proceedings of the Fourth IEEE International Conference on Semantic Computing*, 2010, pp. 462-469.

[3] M.-L. Shyu, T. Quirino, Z. Xie, S.-C. Chen, and L. Chang, “Network intrusion detection through adaptive sub-eigenspace modeling in multiagent systems,” *ACM Transactions on Autonomous and Adaptive Systems*, vol. 2, no. 3, Sep. 2007.

[4] M.-L. Shyu, S.-C. Chen, M. Chen, and C. Zhang, “A unified framework for image database clustering and content-based retrieval,” in *Proceedings of the 2nd ACM International Workshop on Multimedia Databases*, ser. MMDB'04. New York, NY, USA: ACM, 2004, pp. 19-27.

[5] S.-C. Chen, M.-L. Shyu, and C. Zhang, “An intelligent framework for spatio-temporal vehicle tracking,” in *Proceedings of the 4th IEEE International Conference on Intelligent Transportation Systems*, August 2001, pp. 213-218.

[6] S.-C. Chen, M.-L. Shyu, C. Zhang, and R. L. Kashyap, “Identifying overlapped objects for video indexing and modeling in multimedia database systems,” *International Journal on Artificial Intelligence Tools*, vol. 10, no. 4, pp. 715-734, 2001.

[7] S.-C. Chen, M.-L. Shyu, and C. Zhang, “Innovative shot boundary detection for video indexing,” in *Video Data Management and Information Retrieval*, S. Deb, Ed. Idea Group Publishing, 2005, pp. 217-236.

[8] M.-L. Shyu, C. Haruechaiyasak, S.-C. Chen, and N. Zhao, “Collaborative filtering by mining association rules from user access sequences,” in *Proceedings of the International Workshop on Challenges in Web Information Retrieval and Integration*, April 2005, pp. 128-135.

[9] S.-C. Chen, S. Rubin, M.-L. Shyu, and C. Zhang, “A dynamic user concept pattern learning framework for content-based image retrieval,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 36, no. 6, pp. 772-783, Nov 2006.

[10] M. L. Shyu, Z. Xie, M. Chen, and S. C. Chen, “Video semantic event/concept detection using a subspace-based multimedia data mining framework,” *IEEE*

Transactions on Multimedia, vol. 10, no. 2, pp. 252-259, Feb 2008.

[11] L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen, "Effective feature space reduction with imbalanced data for semantic concept detection," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2008, pp. 262-269.

[12] M.-L. Shyu, S.-C. Chen, and R. Kashyap, "Generalized affinity-based association rule mining for multimedia database queries," *Knowledge and Information Systems (KAIS): An International Journal*, vol. 3, no. 3, pp. 319-337, August 2001.

[13] X. Huang, S.-C. Chen, M.-L. Shyu, and C. Zhang, "User concept pattern discovery using relevance feedback and multiple instance learning for content-based image retrieval," in *Proceedings of the Third International Workshop on Multimedia Data Mining, in conjunction with the 8th ACM International Conference on Knowledge Discovery & Data Mining*, July 2002, pp. 100-108.

[14] X. Li, S.-C. Chen, M.-L. Shyu, and B. Furht, "Image retrieval by color, texture, and spatial information," in *Proceedings of the 8th International Conference on Distributed Multimedia Systems*, September 2002, pp. 152-159.

[15] L. Lin, G. Ravitz, M.-L. Shyu, and S.-C. Chen, "Video semantic concept discovery using multimodal-based association classification," in *Proceedings of the IEEE International Conference on Multimedia & Expo*, July 2007, pp. 859-862.

[16] M.-L. Shyu, S.-C. Chen, M. Chen, C. Zhang, and K. Sarinnapakorn, "Image database retrieval utilizing affinity relationships," in *Proceedings of the 1st ACM International Workshop on Multimedia Databases*, New York, NY, USA: ACM, 2003, pp. 78-85.

[17] E. A. Perez, V. F. Mota, L. M. Maciel, D. Sad, and M. B. Vieira, "Combining gradient histograms using orientation tensors for human action recognition," in *Proceedings of the International Conference on Pattern Recognition*, pp. 3460–3463, 2012.

[18] B. Yin, N. Ruiz, F. Chen, and E. Ambikairajah, "Investigating speech features and automatic measurement of cognitive load," in *Proceedings of the IEEE 10th Workshop on Multimedia Signal Processing*, pp. 988-993, Oct. 2008.

[19] S.-C. Chen and R. Kashyap, "Temporal and spatial semantic models for multimedia presentations," in *Proceedings of the 1997 International Symposium on Multimedia Information Processing*, 1997, pp. 441-446.

[20] M.-L. Shyu, C. Haruechaiyasak, and S.-C. Chen, "Category cluster discovery from distributed www directories," *Information Sciences*, vol. 155, no. 3, pp. 181-197, 2003.

[21] D. Liu, Y. Yan, M.-L. Shyu, G. Zhao, and M. Chen, "Spatio-temporal analysis for human action detection and recognition in uncontrolled environments," *International Journal of Multimedia Data Engineering and Management*, vol. 6, no. 1, pp. 1-18, Jan. 2015.

[22] S.-C. Chen, M.-L. Shyu, and R. Kashyap, "Augmented transition network as a semantic model for video data," *International Journal of Networking and Information Systems*, vol. 3, no. 1, pp. 9-25, 2000.

[23] S.-C. Chen, S. Sista, M.-L. Shyu, and R. Kashyap, "Augmented transition networks as video browsing models for multimedia databases and multimedia information systems," in *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, 1999, pp. 175-182.

[24] X. Li, S.-C. Chen, M.-L. Shyu, and B. Furht, "An effective content-based visual image retrieval system," in *Proceedings of the Computer Software and Applications Conference*, 2002, pp. 914{919.

[25] Y. Yan, M. Chen, M.-L. Shyu, and S.-C. Chen, "Deep Learning for Imbalanced Multimedia Data Classification," in *Proceedings of the 2015 IEEE International Symposium on Multimedia (ISM)*, Miami, FL, 2015, pp. 483-488.

[26] Y. Yan, M.-L. Shyu, and Q. Zhu, "Negative Correlation Discovery for Big Multimedia Data Semantic Concept Mining and Retrieval," in *Proceedings of the 2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, Laguna Hills, CA, 2016, pp. 55-62.

[27] M. Merler, B. Huang, L Xie, H. Gang, and A. Natsev, "Semantic model vectors for complex video event recognition," *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 88-101, Feb. 2012.

[28] Q. Zhu and M.-L. Shyu, "Sparse linear integration of content and context modalities for semantic concept retrieval," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 2, pp. 152-160, Jun. 2015.

[29] R. P. W. Duin and D. M. J. Tax, "Experiments with classifier combining rules," in *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, Josef Kittler and Fabio Roli (Eds.), 2000.

[30] Y. Yan, Y. Liu, M.-L. Shyu, and M. Chen, "Utilizing concept correlations for effective imbalanced data classification," in *Proceedings of the 15th IEEE Interna-*

tional Conference on Information Reuse and Integration, pp. 561-568, Aug. 13-15, 2014.

[31] R. P. W. Duin, “The combining classifier: to train or not to train?,” in *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 2, pp. 765-770, 2002.

[32] A. B. Ashfaq, M. Javed, S. A. Khayam, and H. Radha, “An information-theoretic combining method for multi-classifier anomaly detection systems,” in *Proceedings of the IEEE International Conference on Communications*, pp. 1-5, May 2010.

[33] T. Meng, M.-L. Shyu, and L. Lin, “Multimodal information integration and fusion for histology image classification,” *International Journal of Multimedia Data Engineering and Management*, vol. 2, no. 2, pp. 54-70, April-June 2011.

[34] N. Liu, E. Dellandréa, L. Chen, C. Zhu, Y. Zhang, C.-E. Bichot, S. Bres, and B. Tellez, “Multimodal recognition of visual concepts using histograms of textual concepts and selective weighted late fusion scheme,” *Computer Vision and Image Understanding,* vol. 117, no. 5, pp. 493-512, May 2013.

[35] L. I. Kuncheva, “Combining Pattern Classifiers: Methods and Algorithms,” *Wiley-Interscience*, pp. 157-163, 2004.

[36] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, “On combining classifiers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226-239, Mar 1998.

[37] L. Xu, A. Krzyzak, and C. Y. Suen, “Methods of combining multiple classifiers and their applications to handwriting recognition,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 3, pp. 418-435, May/Jun 1992.

[38] A. Al-Ani and M. Deriche, “A new technique for combining multiple classifiers using the Dempster-Shafer theory of evidence,” *Journal of Artificial Intelligence Research*, vol. 17, pp. 333-361, July 2002.

[39] T. Meng and M.-L. Shyu, “Leveraging concept association network for multimedia rare concept mining and retrieval,” in *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo,* pp. 860-865, 2012.

[40] D. Liu and M.-L. Shyu, “Effective moving object detection and retrieval via integrating spatial-temporal multimedia information,” in *Proceedings of the IEEE International Symposium on Multimedia*, pp. 364-371, 2012.

[41] I. Laptev, “On space-time interest points,” *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[42] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, 60, 2, pp. 91-110, 2004.

[43] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online learning for matrix factorization and sparse coding,” *The Journal of Machine Learning Research*, vol. 11, pp.19–60, 2010.

[44] L.-C. Chen, J.-W. Hsieh, Y. Yan, and D.-Y. Chen, “Vehicle make and model recognition using sparse representation and symmetrical SURFs,” *Pattern Recognition*, vol. 48, no. 6, pp. 1979-1998, 2015.

[45] Y. Yan, J.-W. Hsieh, H.-F. Chiang, S.-C. Cheng, and D.-Y. Chen, “PLSA-based sparse representation for object classification,” in *Proceedings of the 22nd International Conference on Pattern Recognition*, pp. 1295-1300, Aug. 24-28, 2014.

[46] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local SVM approach,” in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 3, pp. 32-36, Aug. 2004.

[47] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatio-temporal features,” in *Proceedings of the International Conference on Computer Vision Workshop Visual Surveillance Performance Evaluation Tracking Surveillance*, Oct. 2005, pp. 65-72.

[48] J. Yin and Y. Meng, “Human activity recognition in video using a hierarchical probabilistic latent model,” in *Proceedings of the 17th International Conference on Pattern Recognition*, pp. 15-20, 2010.

[49] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, “Modeling temporal structure of decomposable motion segments for activity classification,” in *Proceedings of the European Conference on Computer Vision*, pp. 1-14, 2010.