Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2014, Article ID 602763, 7 pages
http://dx.doi.org/10.1155/2014/602763

Research Article

Exploring the Best Classification from Average Feature Combination

Jian Hou,[1] Wei-Xue Liu,[1] and Hamid Reza Karimi[2]

[1] *School of Information Science and Technology, Bohai University, Jinzhou 121013, China*
[2] *Department of Engineering, Faculty of Engineering and Science, University of Agder, 4898 Grimstad, Norway*

Correspondence should be addressed to Jian Hou; dr.houjian@gmail.com

Received 22 December 2013; Revised 13 January 2014; Accepted 13 January 2014; Published 19 February 2014

Academic Editor: Shen Yin

Feature combination is a powerful approach to improve object classification performance. While various combination algorithms have been proposed, average combination is almost always selected as the baseline algorithm to be compared with. In previous work we have found that it is better to use only a sample of the most powerful features in average combination than using all. In this paper, we continue this work and further show that the behaviors of features in average combination can be integrated into the k-Nearest-Neighbor (kNN) framework. Based on the kNN framework, we then propose to use a selection based average combination algorithm to obtain the best classification performance from average combination. Our experiments on four diverse datasets indicate that this selection based average combination performs evidently better than the ordinary average combination, and thus serves as a better baseline. Comparing with this new and better baseline makes the claimed superiority of newly proposed combination algorithms more convincing. Furthermore, the kNN framework is helpful in understanding the underlying mechanism of feature combination and motivating novel feature combination algorithms.

1. Introduction

Object classification is a difficult task as there usually exists large intraclass diversity and interclass correlation, even within a small image dataset. The existing single features, for example, SIFT [1], SURF [2], and HOG [3], while being powerful with some classes, seem not enough to deal with all classes alone. In this case, feature combination is proposed to combine the strengths of multiple complementary features and produce better performance than any single one. While classifier fusion [4] can also be used to improve classification performance, in this paper we focus on the combination at the feature level. More specifically, we use SVM classifier in classification and the feature combination is translated into kernel combination [5].

Multiple kernel learning (MKL) is one popular approach to accomplish kernel combination. MKL seeks to obtain the best combination performance by jointly optimizing the weights w_i on individual kernels in $k^*(x, y) = \sum_{i=1}^{n} w_i k_i(x, y)$ together with the SVM parameters α and b [6–10]. Unlike this canonical MKL adopting a uniform weighting scheme over the whole input space, [11] presented a sample-specific MKL algorithm where kernel weights are determined based on both kernel functions and the samples. This algorithm produces some performance improvement at the cost of a large computation load and the risk of over-fitting. Between these two extremes, [10] proposed to use a group-sensitive MKL to make a trade-off between canonical and sample-specific MKL. Different from MKL algorithms optimizing weights and SVM parameters jointly, [12] presented a LPBoost algorithm where the weights and SVM parameters are trained separately in two steps.

While various MKL-like kernel combination algorithms have been published, the controversy surrounding these optimization based approaches has also become evident. On one hand, the optimization based algorithms are usually computationally expensive, and the optimization process consumes huge memory space. On the other hand, the real effectiveness of these algorithms in improving performance has been called in question. In [12] it is noticed that when all participated features are carefully designed to be powerful, the sophisticated optimization algorithms, for example,

MKL, do not show evident advantage over the baseline average combination. Only when both strong and weak features are combined, the optimization based approaches suppress the effect of weak features and perform better than average combination. In the supplement to [12] the authors further claim that the MKL-like combination algorithms seem to be overestimated in the literature, due to missing comparison with the simple yet powerful average combination. Moreover, the supplement states that there seems to be an agreement on the fact that MKL almost never improves performance.

The tiny, if any, performance improvement from MKL-like algorithms together with the huge computation and memory space consumption seems to indicate that the existing optimization based combination approaches are approaching a bottleneck, and a new framework is needed to generate further evident performance improvement. This observation motivated us to investigate the behaviors of features in average combination, in an endeavor to find out the underlying mechanism of feature combination. In fact, our work is consistent with the shift of research focus from heuristic combination algorithms to theoretical explanation of the combination mechanism [13, 14]. In [15] we have found that if we add features into average combination one by one in descending order according to their discriminative power, the classification performance of combination firstly rises, then peaks, and finally drops. In other words, average combination with a sample of most powerful features produces better classification results than with all features, and the performance gain of using a most powerful sample can be quite large in some cases. This means that it may not be convincing to claim the superiority of a new combination algorithm by comparing with the ordinary average combination. This observation further renders it necessary to explore the potential of average combination and present a better baseline combination algorithm.

While the experiments in [15] show that it is possible to improve the classification performance of average combination, some problems are left unsolved. Firstly, in [15] we tested the feature combination performance by adding features into combination in descending order, ascending order, and mixed order and found that the best classification results are obtained with a sample of most powerful features in the descending order. However, it is not clear if some other orders, for example, random ordering, can be used to produce better results than descending order. Secondly, in descending order the best sample size needs to be determined in order to obtain the best classification performance. Thirdly, we are interested to find out how the features used in combination influence the final classification results. In order to solve these problems, in this paper we firstly compare the random order with the best performing descending order. As a result, we find that the behaviors of feature combination can be elegantly integrated into the k-Nearest-Neighbor (kNN) framework. Based on this framework, we then present a selection based average combination (SBAC) algorithm to obtain the best classification results from average combination. This SBAC algorithm is simple yet powerful and can serve as a better baseline algorithm in feature combination. Furthermore, the kNN framework provides a reasonable explanation for the observations as to the relation between features combined and resulted performance gain. All these results enable us to conclude that the kNN framework sheds some light on understanding the mechanism underlying feature combination and is therefore helpful in motivating novel feature combination algorithms. Although in this paper we focus on image classification, we would like to highlight that the idea of combining features to improve classification performance is also applicable to other domains [16–18].

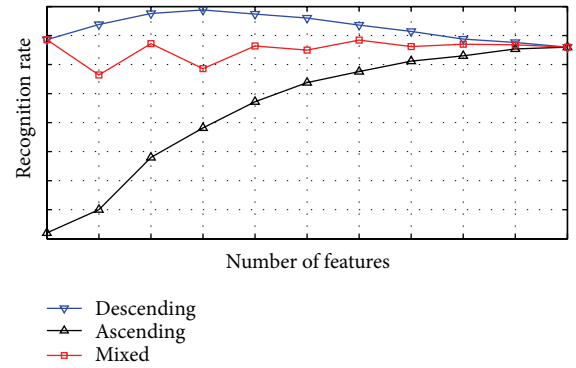

Figure 1: Illustration of recognition rate curves in descending, ascending, and mixed orders, as observed in [15].

The remainder of this paper is organized as follows. In Section 2 we compare the descending order with random order and then present the kNN framework to explain the behaviors of features in average combination. Section 3 details the SBAC algorithm and experimental results. In Section 4 we conclude the paper.

2. kNN Framework

In this section we investigate the influence of the ordering of features being added into average combination on classification performance. To begin with, we review the three orders tested in [15]. The discriminative power of a feature is evaluated by 10-fold cross-validation with its corresponding kernel matrix. In descending order, the features are sorted in descending order according to their discriminative power and added into combination one by one. In ascending order, we operate similarly in ascending order. In mixed order, the features are still sorted in descending order. Whereas in combination, we take features from the top and the bottom of the list alternatively and add them into combination one by one.

Based on the experiments in [15], the behaviors of features in the three orders can be illustrated in Figure 1. In descending order, the curve of recognition rates shows a "rise-peak-drop" shape with the addition of features into combination. In ascending order, the participation of new (and thus more discriminative) features in combination always improves the classification until all features produce the best results in this order. In mixed order, the strong and weak features push up and drag down the recognition rates alternatively. In all cases the ascending and mixed orders have no chance to outperform the descending order.

From our description and Figure 1 we see that in the three orders, the best classification results come from the descending order, when a sample of most discriminative features is combined. Therefore we select the descending order as the best performing one out of the three orders.

Besides the three orders tested above, there is another possible ordering method, that is, random sampling. In this order we randomly sample a number of features from all and use them in average combination. Although we have no theoretical or intuitive ground to support random sampling as a better order than the descending one, we cannot trivially discard this order without extensive experiments. Therefore we compare the random sampling with descending order as follows. With each number N of features in combination, that is, from 2 to the number of all features, we randomly sample N features and use them in combination. Note that the sampled N features must not be exactly the same as the ones used in descending mode. We repeat the random sampling 50 times and use the best result to represent this order.

In the experiments we use the same four datasets as in [15] and the experimental setups are briefly listed as follows:

Event-8 [19]: randomly selected 70 images as training and 60 images as testing per class.

Scene-15 [20]: randomly selected 100 images as training and all the others as testing per class.

Oxford Flower-17 [21]: randomly selected 20 images as training and 20 images as testing per class.

Caltech-101 [22]: randomly selected 30 images as training and 15 images as testing per class.

The features adopted in our experiments include 500-bin Gabor filters, Bag-of-SIFT descriptors in gray, HSV and CIE-Lab space, 20-bin oriented and 40-bin unoriented PHOG [23], and 64-bin gray value histogram. In building Bag-of-SIFT descriptors, SIFT descriptors [1] are extracted on regular grids with spacing of 10 pixels and with patches of radii $r = 4, 8, 12, 16$ to allow for scalability and then quantized into a 500-bin vocabulary. Altogether we used 7 types of features for Caltech-101, Event-8, and Flower-17 and 5 types for Scene-15 (only containing gray images). For each feature, we build the descriptors in spatial pyramid from level 0 to level 2. In total, we have 21 descriptors for the three color datasets and 15 for Scene-15.

For each descriptor, the kernel matrix is built with each entry in the form of $k(x, y) = \exp(-d_0^{-1} d(x, y))$, where d is the pairwise χ^2 distances and d_0 is the mean of pairwise distances. We adopt χ^2 distance to build kernels as it performs the best among several other commonly used kernels [15, 24]. In all our experiments the multiclass SVM is trained in a one-versus-all manner and the regulation parameter C is fixed to be 1000. The performance measure is reported as the mean recognition rate per class. For each dataset, we test with 10 different training-testing splits and report the mean of classification results in Figure 2.

Although randomly sampling 50 times is not a exhaustive search, we can see in Figure 2 that random sampling is barely able to outperform descending mode. At the same time, we note that there do exist some cases where the best of random order performs a little better than descending order. This observation can be attributed to the ordering criterion, that is, cross-validation accuracy. Since cross-validation accuracy is only an approximate estimation of the discriminative power, but not a precise measure, the top N features in descending order may not be exactly the N most discriminative features. In this case, it is likely that a random sample captures the N most discriminative features *accidentally,* while descending mode does not. This also explains why the recognition rate curves of descending order do not follow the "rise-peak-drop" shape strictly. Since in the random order we only report the best results, this observation does not influence our conclusion that the descending order performs the best in the four orders. To our best knowledge, we have listed all the possible orderings in average combination and we conclude that descending order performs the best and can be used to produce better performance than the ordinary average combination.

When we look at the recognition rate curves of the four orders in Figures 1 and 2, we find that the behavior of features in average combination can be elegantly explained in the framework of k-Nearest-Neighbor (kNN) classification. Regarding the most discriminative features as the closest training examples and the least discriminative features as the furthest ones, we readily understand why the recognition rate curves of the three orders are of the shapes illustrated in Figure 1 and why the descending order is able to produce better results than the ordinary average combination. Furthermore, we arrive at some interesting conclusions from this kNN framework. In the case that the discriminative powers of individual features vary widely, the weak features added later shall drag down the performance curves significantly and make the average combination with all features much inferior to the one with only a sample of most discriminative features. This is where the optimization based algorithms show their advantage over ordinary average combination by suppressing the effect of weak features. On the other hand, if all features are of similar discriminative power, the recognition rate curves shall only be dragged down marginally as all training examples are of similar distance. Therefore the average combination with all features performs similar to the one with a sample of most discriminative features. This leaves little space for optimization based algorithms to improve and explains why, with a set of carefully designed features, the optimization based combination algorithms perform no better than the baseline average combination, as observed in [12]. We shall see, in Section 3, that experiments validate these two arguments.

3. Selection Based Average Combination

Now we are able to present the selection based average combination (SBAC) algorithm as a better baseline algorithm for feature combination. Since in the descending order the recognition rate curves follow the "rise-peak-drop" trend, what is left for us to do is to determine where the peak is reached, that is, the appropriate candidate of k in kNN. Cross-validation is an effective measure to evaluate

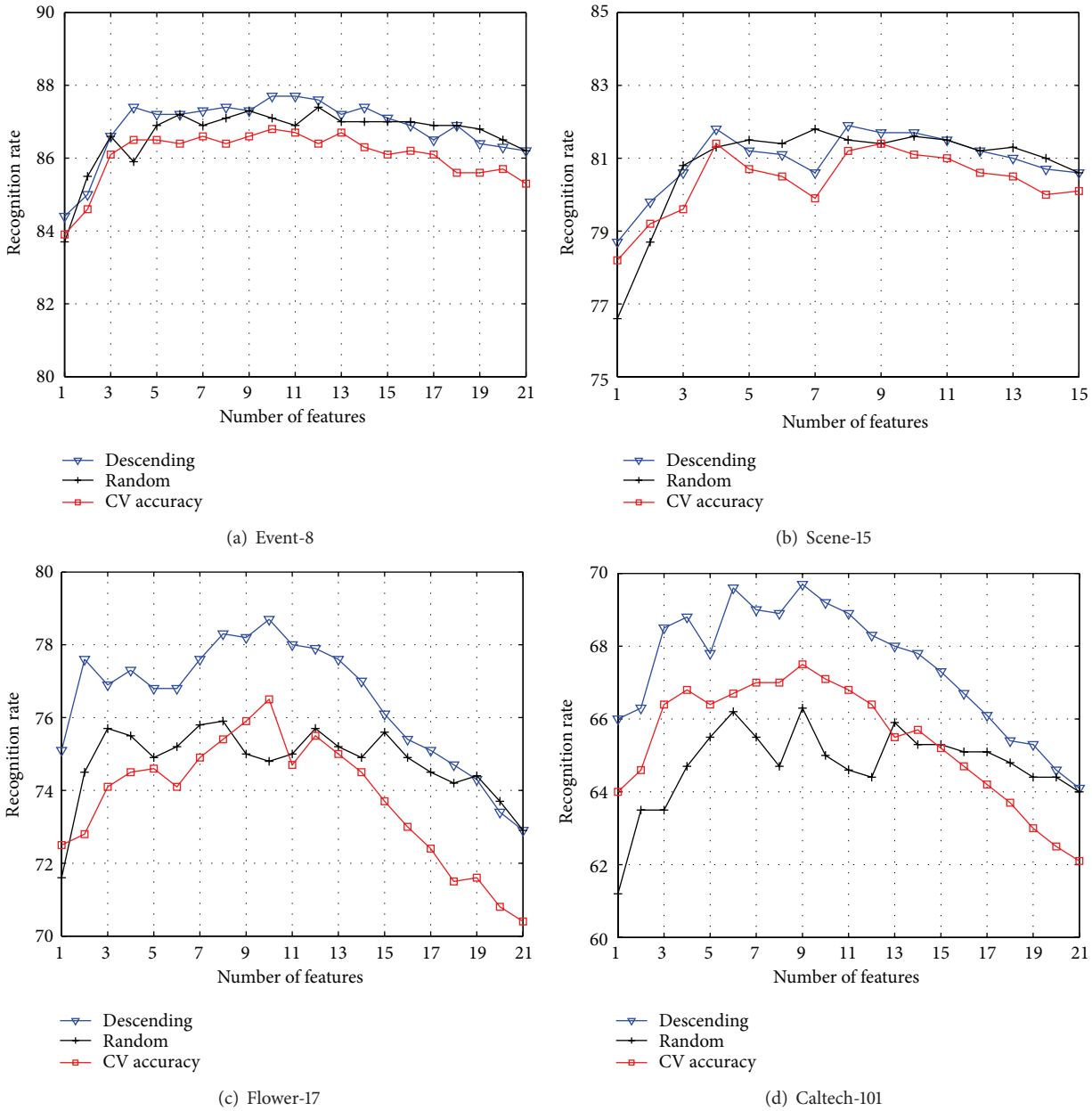

FIGURE 2: Comparison of average combination in descending order and in random order. The cross-validation accuracy in descending order is also illustrated.

the discriminative powers of features and we have used it in the ordering of features. Here we can also use 10-fold cross-validation to determine the best sample size in combination. Specifically, we sort the features in descending order and add them into combination one by one. When a feature is added into combination, we use 10-fold cross-validation to assess and record the discriminative power of the combined kernel matrix. When the cross-validation accuracy peak is reached, the peak of the recognition rate curve is also reached. In order to support this method, we compare the actual recognition rates and the cross-validation accuracy in Figure 2. Although the curves of recognition rate and cross-validation accuracy do not have exactly the same shapes, they do have very similar trends, and the peaks of cross-validation accuracy curves indicate the location of recognition rate peak correctly.

It is evident from Figure 2 that SBAC performs better than the ordinary average combination with all features. In our experiments, the performance gains are 1.5, 1.1, 5.8, and 5.6 percent for Event-8, Scene-15, Flower-17, and Caltech-101, respectively. Considering that MKL algorithms, for example, in [12], usually outperform the ordinary average combination by only several percent, which are in the same order of magnitude as our performance gains, we believe these results highlight the importance of exploring the best results from average combination and presenting a better baseline combination algorithm.

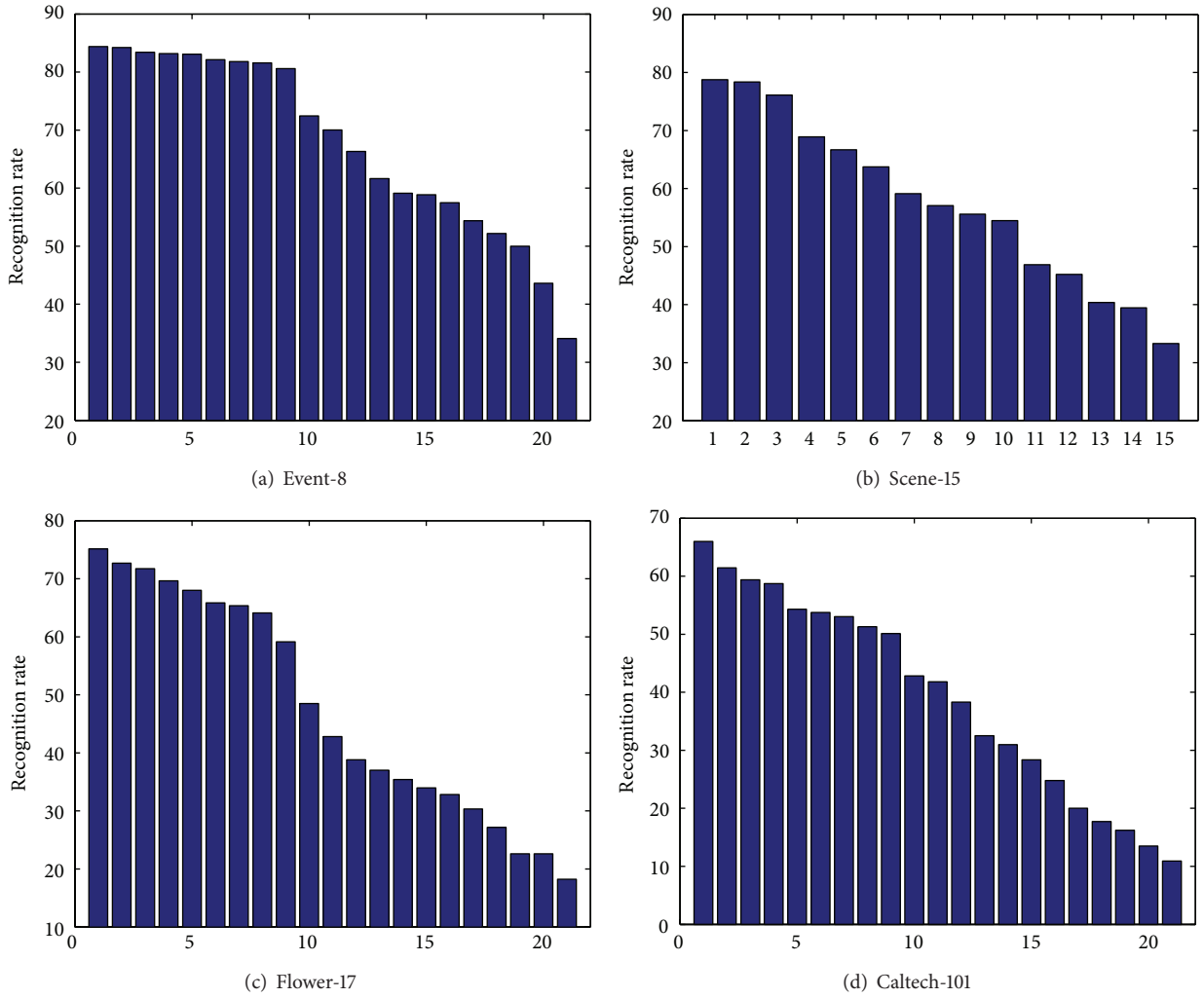

Figure 3: The cross-validation accuracy of individual features used in average combination.

An interesting observation here is that the performance gains vary from dataset to dataset, that is, quite large with Flower-17 and Catech-101 and fairly small with Event-8 and Scene-15. In Section 2 we attribute this observation to the variance of the discriminative power of individual features. In order to validate this explanation from the kNN framework, we list the 10-fold cross-validation accuracy of individual features in descending order in Figure 3. Comparing Figure 3 to Figure 2, we observe a definite correlation between the discriminative power variances and the performance gains. In fact, in our experiments the standard deviation of the cross-validation accuracy of individual features is 15.49, 14.56, 19.39, and 17.56 for Event-8, Scene-15, Flower-17, and Caltech-101, respectively. This explains why the performance gains are large for Flower-17 and Caltech-101 and small for Event-8 and Scene-15. These observations further confirm the effectiveness of the kNN framework in explaining the behaviors of features in average combination.

Although in this paper we focus on image classification, the idea of combining multiple features and classifiers to obtain better classification performance is also applicable to other related domains, for example, document classification, speech recognition, fault diagnosis, and others [25–27]. Therefore in the next step we plan to continue our work in two aspects. Firstly, we shall investigate the behaviors of features in average combination in these domains and check if the kNN framework is still valid. This investigation shall deepen our understanding of the feature combination mechanism and help motivate novel and more powerful feature combination algorithms. Secondly, we plan to make an in-depth study of the existing feature combination algorithms in these domains to see if it is possible to apply them to image classification. Altogether, we aim for a more precise and universal understanding of the feature combination mechanism and the best classification performance from average combination.

4. Conclusion

In this paper we investigated the behaviors of features in average combination through extensive experiments on four diverse datasets. As a result, we found that the average

feature combination can be integrated into the k-Nearest-Neighbor framework where the most discriminative features are regarded as the closest training examples and the least discriminative features as the furthest ones. Based on this framework, we present a selection based average combination algorithm which performs evidently better than the ordinary average combination and thus serves as a better baseline combination algorithm. Since the kNN framework can be used to explain all the behaviors we observed in average feature combination, we believe it is helpful in understanding the feature combination mechanism and motivating novel feature combination algorithms.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant no. 61304102), Natural Science Foundation of Liaoning Province of China (Grant no. 2013020002), and Scientific Research Fund of Liaoning Provincial Education Department under Grant no. L2012400.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "Surf: speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 886–893, June 2005.

[4] L. Sun, C.-Z. Han, J.-J. Shen, and N. Dai, "Generalized rough set method for ensemble feature selection and multiple classifier fusion," *Acta Automatica Sinica*, vol. 34, no. 3, pp. 298–304, 2008.

[5] J. Hou and M. Pelillo, "A simple feature combination method based on dominant sets," *Pattern Recognition*, vol. 46, pp. 3129–3139, 2013.

[6] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[7] A. Kumar and C. Sminchisescu, "Support Kernel machines for object recognition," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, October 2007.

[8] Y. Y. Lin, T. L. Liu, and C. S. Fuh, "Local ensemble kernel learning for object category recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, pp. 1–8, June 2007.

[9] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV '07)*, pp. 1–8, October 2007.

[10] J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao, "Group-sensitive multiple kernel learning for object categorization," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 436–443, October 2009.

[11] M. Gönen and E. Alpaydin, "Localized multiple kernel learning," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 642–651, July 2008.

[12] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 221–228, 2009.

[13] L. I. Kuncheva, "A theoretical study on six classifier fusion strategies," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 281–286, 2002.

[14] S. Tulyakov, S. Jaeger, V. Govindaraju, and D. Doermann, *Review of Classifier Combination Methods*, Springer, New York, NY, USA, 2008.

[15] J. Hou, B. P. Zhang, N. M. Qi, and Y. Yang, "Evaluating feature combination in object classification," in *Proceedings of the International Symposium on Visual Computing*, pp. 597–606, 2011.

[16] S. Yin, S. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic datadriven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process," *Journal of Process Control*, vol. 22, pp. 1567–1581, 2012.

[17] S. Yin, X. Yang, and H. R. Karimi, "Data-driven adaptive observer for fault diagnosis," *Mathematical Problems in Engineering*, vol. 2012, Article ID 832836, 21 pages, 2012.

[18] S. Yin, G. Wang, and H. Karimi, "Data-driven design of robust fault detection system for wind turbines," *Mechatronics*, 2013.

[19] L. L. Jia and L. Fei-Fei, "What, where and who? classifying event by scene and object recognition," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1–8, 2007.

[20] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, vol. 2, pp. 2169–2178, June 2006.

[21] M. E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 1447–1454, June 2006.

[22] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories," in *Proceedings of the Computer Vision and Pattern Recognition Workshop (CVPRW '04)*, p. 178, 2004.

[23] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR '07)*, pp. 401–408, July 2007.

[24] J. Hou, W. X. Liu, E. Xu, Q. Xia, and N. M. Qi, "An experimental study on the universality of visual vocabularies," *Journal of Visual Communication and Image Representation*, vol. 24, pp. 1204–1211, 2013.

[25] S. Yin, H. Luo, and S. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE*

Transactions on Industrial Electronics, vol. 61, pp. 2402–2411, 2014.

[26] A. Baak, M. Muller, G. Bharaj, H.-P. Seidel, and C. Theobalt, "A data-driven approach for real-time full body pose reconstruction from a depth camera," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1092–1099, November 2011.

[27] S. Yin, S. Ding, A. Haghani, and H. Hao, "Data-driven monitoring for stochastic systems and its application on batch process," *International Journal of Systems Science*, vol. 44, pp. 1366–1376, 2013.