21st International Conference on Pattern Recognition (ICPR 2012)
November 11-15, 2012. Tsukuba, Japan

Confidence-Assisted Classification Result Refinement for Object Recognition Featuring TopN-Exemplar-SVM

Toshihiko Yamasaki[1,2,3] and Tsuhan Chen[1]
[1]*Cornell University,* [2]*The University of Tokyo,* [3]*JSPS*
ty273@cornell.edu and tsuhan@ece.cornell.edu

Abstract

This paper proposes a cascaded classifier framework for better image recognition. The proposed method is based on the confidence values given by the classifiers. By using our proposed topN-Exemplar SVM in the second stage and comparing the confidence values with those from the first stage, the classification results with less confidence are successfully updated. The validity of our algorithm has been demonstrated by the experiments using three standard image datasets.

1. Introduction

We have witnessed significant improvements in object recognition in the last decade. For instance, the classification accuracy for the Caltech-101 dataset [1] was around 40% [2][3] in the mid 2000's but it is now improved up to 73% and higher [4]-[14].

In most cases, the classifier is conventional: a multi-class support vector machine (SVM) or AdaBooost is used. The object's class is decided by picking up the class whose confidence is the highest. The confidence is a signed distance from the hyper plane in the case of SVM. However, this strategy does not necessarily mean that the classifier is always confident of its decisions. Table 1(a) shows the statistics of the confidence values for the Caltech-101 dataset using the ScSPM [4]. It shows that only 53% of the highest confidence scores are positive. When the confidence is positive, 99% of the predicted labels are correct. On the other hand, if the highest confidence is negative, only 58% of them are correct and 42% of them are wrong. Although taking the class with the highest confidence has been the best choice so far even if it is negative, it shows that there is some room for improvement.

Some works have been proposed to improve the classification results. One possible solution is limiting the candidate classes in the first stage classifier and training the second stage classifier only with the extracted candidates as proposed in [3]. A cascaded classification model (CCM) [16] is a method to train the second stage classifiers using the outputs from the first stage classifiers. Another approach is using an exemplar-SVM [17]. In this approach, an SVM is trained for each training sample so that the classifier would become sensitive only to a specific exemplar. However, as shown in Section 4, none of these approaches works properly in the framework of large-scale image datasets with a lot of different object classes.

Table 1. Classification accuracy as a function of the top confidence value to the input image: (a) Caltech-101 [1], (b) Caltech-256 [15]. ScSPM was used.

(a)

	TRUE	FALSE	Total
confidence>0	99% (3183)	1% (45)	(3228)
confidence<0	58% (1667)	42% (1189)	(2856)

(b)

	TRUE	FALSE	Total
confidence>0	97% (2434)	3% (69)	(2503)
confidence<0	35% (7169)	65% (13225)	(20394)

In this paper, we propose a confidence-assisted two-stage classification framework using our topN-exemplar-SVM classifier. The topN-exemplar-SVM classifier is a combination of the SVM-kNN [3] and the exemplar-SVM [17] which work complementarily to each other. The experiments using three standard datasets have shown promising results.

The rest of this paper is organized as follows. Section 2 demonstrates detailed analysis on the classification performance of some state-of-the-art algorithms. Section 3 describes the proposed algorithm. The experimental results are demonstrated in Section 4, followed by the concluding remarks in Section 5.

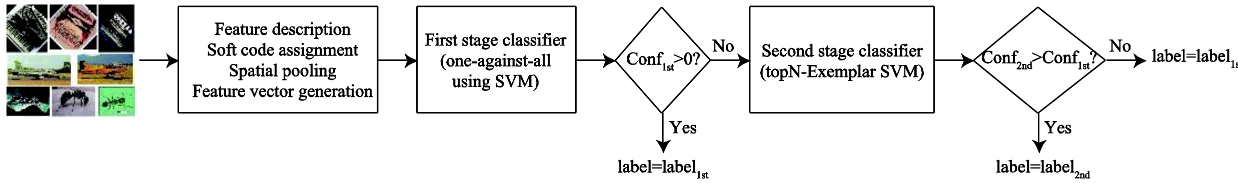

Figure 2. Flowchart of the proposed algorithm.

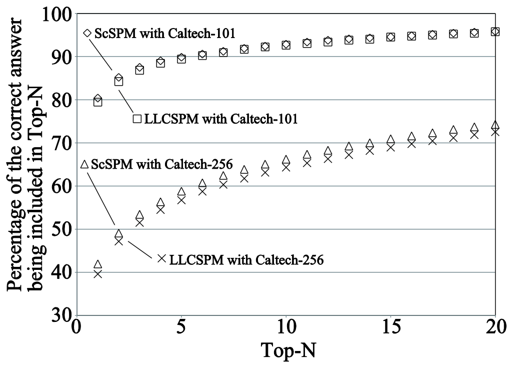

Figure 1. Percentage of the correct answers being included in Top-N classes.

2. Detailed Analysis on Recognition Results

Table 1 shows the image classification accuracy as a function the top confidence value to the input image. Here, ScSPM [4] with a linear SVM is used. When the confidence values are positive, most of the predicted labels are correct. However, such cases are not so many. In fact, 47% of the top confidence values are negative. When they are negative, the accuracy decreases drastically. This tendency becomes clearer in the case of the Caltech-256 dataset (Table 1(b)). When the top confidence is negative, the performance is miserably low.

Then, in which place are the correct answers? Fig. 1 demonstrates the percentage of the correct answers being included in the top-N classes for ScSPM and LLCSPM [5]. The Caltech-101 and the Caltech-256 datasets were used. For instance, it is observed that 87% of correct labels are within the top-3 classes and 89% of them are in the top-5 classes in the case of ScSPM for Caltech-101. This shows that about 8% of the correct answers are either in the 2nd or 3rd place even though they are not the top.

3. Proposed Algorithm

3.1. Confidence-Assisted Multi-Stage Classifier

The flowchart of our proposed method is shown in Fig. 2. The first stage classifier is a conventional one such as SVM. We assume that the first stage classifier outputs a confidence value for each class. The first stage classifier is accurate enough especially when its top confidence value is positive. Therefore, in the second stage, we only care about the case where the top confidence value is negative. And only when the confidence value in the second stage is larger than that in the first stage, the class label is updated. If not, the class label given by the first stage is employed.

There are some algorithms which use multiple (weak) classifiers such as AdaBoost and Viola-Jones cascaded classifier [18]. AdaBoost uses a weighted sum of multiple classifier outputs considering difficult-to-learn training data. A Viola-Jones cascaded classifier is a coarse-to-fine and one way approach in which most of false positives are eliminated in earlier stages while keeping true positives. On the other hand, our approach is a two-stage classifier that can decide which stage to use based on the confidence value. The first stage classifier can extract both true positives and true negatives when it is confident. The second stage classifier is used only when the first stage classifier is not confident. The final decision is made by comparing the confidence values from the first and the second stages. In this point of view, the proposed algorithm is different from the other approaches.

3.2. TopN-Exemplar-SVM

For the second stage classification, we propose a topN-exemplar-SVM, which is based on SVM-kNN [3] and exemplar-SVM [17]. Namely, top-N candidate classes are extracted and exemplar-SVMs are trained only for them.

The basic idea of the SVM-kNN [3] is to limit the training data for the second stage classifier by the k nearest neighbor search. However, simply extracting the top-N classes in the first stage SVM and retraining another SVM only with them in the second stage does not work well. Because the top-N classes are the N most confusing (i.e., probable) classes, the classifier in the second stage would yield wrong answers again as long as the same type of classifier is used. However, we borrow the idea of limiting the candidate classes for refining the classification results. Note that we use the top-N classes though Ref. [3] used top-k images.

The exemplar-SVM [17] is a method to train a separate SVM classifier for every exemplar in the training set. Since each detector is quite specific to its exemplar,

Table 2. Classification accuracy improvement as a function of the number of training data per class: (a) Caltech-101, (b) Caltech-256, (c) PASCAL VOC2011. N is set to 3. The performance of the original ScSPM and LLCSPM is obtained by executing the source code provided by the authors.

(a)

# of training data	30	60	90
ScSPM	73.4 to 73.9 (+0.5)	-	-
LLCSPM	69.8 to 70.3 (+0.5)	-	-

(b)

# of training data	30	60	90
ScSPM	34.9 to 35.7 (+0.8)	-	-
LLCSPM	35.7 to 36.0 (+0.3)	40.6 to 41.1 (+0.5)	-

(c)

# of training data	30	60	90
LLCSPM	30.0 to 30.1 (+0.1)	33.7 to 33.8 (+0.1)	35.1 to 35.3 (+0.2)

intra-class variability of the training objects becomes less problematic. One of the significant issues of the exemplar-SVM is the processing cost. The number of classifiers to train is equal to that of the training data, which is computationally very expensive. Therefore, we propose to use it in the second stage classifier in conjunction with the SVM-kNN.

The proposed TopN-exemplar-SVM combines the advantages of the two approaches. Liming the number of training samples by top-N drastically reduces the training/testing time for exemplar-SVM. In addition, the exemplar-SVM can focus only on more "probable" classes. On the other hand, from the SVM-kNN point of view, the classifier that can respond only to specific cases is ideal because it is difficult for conventional SVMs to correctly classify such probable but confusing data.

The results given by the second stage classifier are employed only when the confidence values are larger than those in the first stage classifier. One might wonder whether we can compare the confidence values from the classifiers with different sets of training data. However, it is a common approach in one-against-all for multi-class classification. The confidence value can be regarded as a relative value to those of support vectors, which are always either +1/-1.

The processing cost for the topN-exemplar-SVM is a few seconds per image. This is not a problem in practical usage because feature extraction and code assignment are the bottleneck of the whole pipeline.

4. Experimental Results

In this section, experimental results are shown using three widely used datasets: Caltech-101 [1], Caltech-256 [15], and PASCAL VOC2011 [19]. We employed the ScSPM [4] and LLCSPM [5] for feature representation. The reason for choosing [4][5] is that the source code is available on the authors' project site. So anyone can reproduce the results in [4][5] and those in this paper. In addition, we can guarantee that all the configurations other than the classifier are the same. The source code of this paper is also available on our project site. The codebook sizes were 1,024 for Caltec-101, 4,096 for Caltech-256, and 2,048 for VOC2011, respectively. The training data were sampled randomly and the rest were used for testing. The accuracies of the original ScSPM/LLCSPM and our proposed method were calculated using the same training/test data. This procedure was repeated five times and the average accuracies were calculated.

Table 2 shows the performance improvements as a function of the number of training data per class. It is shown that the performance of the proposed algorithm is always better than the original ScSPM and LLCSPM. It is also observed that the accuracy is improved more when the number of training data is increased because the probability of a proper exemplar being included in the training set becomes larger.

Fig. 3 demonstrates the performance improvement of ScSPM for the Caltech-101 dataset and its processing time. When N is small, the probability for the correct class being included in the second stage training is smaller. The training/testing cost is also small. On the other hand, when N is large, more unrelated (negative) samples are included, resulting in lower accuracy and more computational cost, as well. We confirmed that the best N is 2-3 for ScSPM/LLCSPM regardless of the dataset.

In Table 3, the performance of our topN-exemplar-SVM is compared with other refinement methods [3][16][17]. The performances of SVM-kNN and exemplar-SVM are worse than that of the original ScSPM. In addition, CCM does not contribute very much, either.

As shown in Table 4, the accuracy of our proposed method is comparable to state-of-the-art image recogni-

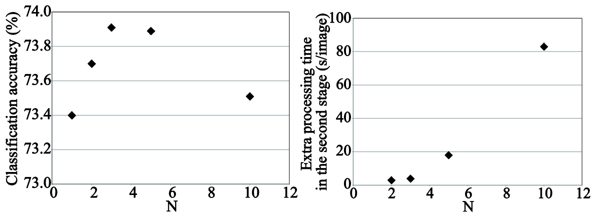

Figure 3. (left) classification performance, (right) processing time for the second stage classification. ScSPM with Caltech-101 is used.

Table 3. Comparison with SVM-kNN, CCM, and Exemplar-SVM using ScSPM for Caltech-101.

Algorithm	Accuracy
ScSPM [4]	73.2%
Modified SVM-kNN [3], N=3	69.4%
CCM [16]	72.1%
Exemplar-SVM [17]	69.1%
TopN-Exemplar-SVM (proposed)	73.9%

tion algorithms [4]-[9]. They are focused on feature representation or spatial pooling and they use conventional SVMs in the classification stage. Therefore, our method can be incorporated into such algorithms.

5. Conclusions

In conventional multi-class object recognition, the class with the highest confidence was taken. Although this has been the best and the only choice, the classification accuracy was low when the top confidence value was negative. In this paper, we have investigated the statistics on the confidence values in detail and demonstrated that the classification accuracy can be improved by the two-stage classifier based on the confidence values. For the second stage classifier, we have developed a topN-exemplar SVM classifier. The experimental results using three standard datasets demonstrated that the proposed work can improve the object classification accuracy of state-of-the-art algorithms.

Table 4. Classification accuracy comparison with recent approaches using ScSPM for Caltech-101.

Algorithm	Accuracy
ScSPM [4]	73.2%
LLCSPM [5]	73.4%
D-SP [6]	67.2%
LC-KSVD2 [7]	73.6%
RLDA [8]	73.7%
Hie Sc [9]	74.0%
Code Relation [10]	74.3%
Proposed (using ScSPM)	73.9%

References

[1] F.-F. Li and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In CVPR WS on Generative-Model Based Vision, 2004.

[2] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In ICCV, 2005.

[3] H. Zhang, A. Berg, M. Maire, J. Malik. SVM-KNN: discriminative nearest neighbor classification for visual category recognition. In CVPR, 2006.

[4] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In CVPR, 2009.

[5] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality constrained linear coding for image classification. In CVPR, 2010.

[6] T. Harada, Y. Ushiku, Y. Yamashita, and Y. Kuniyoshi. Discriminative spatial pyramid. In CVPR, 2011.

[7] Z. Jiang, Z. Lin, and L. S. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-SVD. In CVPR, 2011.

[8] S. Karayev, M. Fritz, S. Fidler, and T. Darrell. A probabilistic model for recursive factorized image features. In CVPR, 2011.

[9] K. Yu, Y. Lin, and J. Lafferty. Learning image representations from the pixel level via hierarchical sparse coding. In CVPR, 2011.

[10] Y. Huang, K. Huang, C. Wang, and T. Tan. Exploring relations of visual codes for image classification. In CVPR, 2011.

[11] 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 CVPR, 2011.

[12] X. Wang, X. Bai, W. Liu, and L. J. Latecki. Feature context for image classification and object detection. In CVPR, 2011.

[13] J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric lp-norm feature pooling for image classification. In CVPR, 2011.

[14] N. Kulkarni and B. Li. Discriminative affine sparse codes for image classification. In CVPR, 2011.

[15] Griffin, G. Holub, and P. A. Perona. Caltech-256 object category dataset. In Tech. Report 7694, Caltech, 2007.

[16] G. Heitz, S. Gould, A. Saxena, and D. Koller. Cascaded classification models: combining models for holistic scene understanding. In NIPS, 2008.

[17] T. Malisiewicz, A. Gupta, A. A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In ICCV, 2011.

[18] P. Viola and M. Jones. Robust real time object detection. IJCV, Vol. 57, Issue 2, pp. 137-154, 2004.

[19] M. Everingham, L. VanGool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge 2009 results.