Fine-Grained Visual Classification Based on Image Foreground and Sub-category Similarity

Xianjin Jiang, Xin Lin, Yi Ji, Jianyu Yang, and Chunping Liu(✉)

School of Computer Science and Technology, Soochow University,
Suzhou, Jiangsu, China
cpliu@suda.edu.cn

Abstract. We propose a fine-grained visual classification algorithm based on image foreground and sub-category similarity. In the processing of feature extracting, our model calculates the gradient of image pixels in a classification network to obtain the foreground of the image. Then input the foreground image and the original image into the bilinear convolution network to obtain the feature of the image. At the classification stage, we propose an improved SD-SVM algorithm, which takes the advantages of the similarities among sub-categories and the differences among the similarities of sub-category. Experimental results manifest that our algorithm can achieve 85.12% accuracy on the CUB-2011 dataset and 85.21% accuracy on the FGVC-aircrafts dataset even with only the category labels, which outperforms state-of-the-art fine-grained categorization methods.

1 Introduction

Fine-grained categorization, also known as sub-category image classification, has attracted wide attention in recent years. Unlike Pascal VOC's tasks for classifying boats, bicycles and cars, fine-grained visual classification algorithm distinguishes sub-categories with high similarity. Therefore, this task is more difficult than most image classification tasks.

In general, fine-grained visual classification algorithm contains two steps: feature extraction and classification. At the stage of feature extraction, the annotations of object level and part level are useful to improve the accuracy of classification. Some of existing classification algorithms [1,6,9] use manual annotation information. Because the cost of manual annotation information is expensive, some algorithms [2,11,12] only use the category label to extract the features. At the classification stage, previous fine-grained categorization algorithms are directly use Multi-classification SVM. Besides, Lin et al. [2] pointed out that the experiment using multi-classification SVM performs better than using softmax.

In this paper, a fine-grained visual classification algorithm is designed with only the category labels. In our model, we compute the gradient of CNN to obtain the foreground, refer to 3.1. Besides, we use the bilinear CNN to extract features, refer to 3.2. We found that the similarity between categories can improve the accuracy at the classification stage. The probability of classifying the image into

J. Yang et al. (Eds.): CCCV 2017, Part III, CCIS 773, pp. 143–154, 2017.
https://doi.org/10.1007/978-981-10-7305-2_13

similar category is higher than that of classifying the image into not similar category. So we modify the multi-classification SVM for that refer to Sect. 3.2.

2 Related Work

In this subsection, we introduce some researches of fine-grained visual classification in term of feature extraction and multi-classification in recent years.

2.1 Feature Extraction

Feature extraction of fine-grained visual classification mainly include two categories. One is fusing features of object and part level with manual annotation information. The other is automatic extracting feature by deep learning. Additional manual annotation information about object level and part level, can play an important role in fine-grained visual classification tasks. Zhang et al. [6] proposed the part R-CNN algorithm. The part R-CNN model uses the manual annotations of object level and part level to train R-CNN [7] network. The R-CNN network is used to detect the object and part during the test stage. Finally, we obtain the final feature combining the object features and the part feature. Branson et al. [8] also proposed a Pose Normalized CNN algorithm. The algorithm uses the key points of manual annotation to obtain the object and part, and the localization of object and part is normalized. A final feature is obtained by combining the two normalized features of object and part.

In recent years, more and more studies that tend to not use the annotation information of object and part level achieved a very good effect. To replace the effect of annotation information, Simon and Rodner [13] designed a constellation algorithm that uses convolutional network features. A Final feature is extablished by fusing the extracting features of object and part using key points. Different from the above method, Lin et al. [2] designed a novel Bilinear CNN network model, it uses the original image as the input of the classification network, and achieves 84.1% accuracy on the CUB200-2011 data set. Zhang et al. [3] construct complex features using deep filters, and achieve the highest accuracy for the database only with the category label.

Bilinear CNN has achieved great success on fine-grained problems, but this method takes the original image as input and is greatly influenced by image background. To solve this problem, our model obtains the bounding box of the image by calculating gradient of image pixels in the classification network. Then obtain features form the original image and the image cutting by bounding box.

2.2 Multi-classification SVM

SVM was originally designed for binary classification problem. When dealing with multiple classification problem, we need to construct the appropriate multi-class classifier. At present, there are two methods to construct SVM classifier:

(1) Direct method: the kind of methods directly modify the objective function, and merge the parameters of multiple classifications into an optimization problem. This method has high computational complexity, and is only suitable for small problems. (2) Indirect method: This kind of methods achieve the framework of multi-classifier through the combination of multiple two classifier. These methods can be classified two categories: one-to-rest and one-to-one.

One-to-rest [17] is one of the earliest and most widely used methods. This method firstly constructs k classifiers (k is the total category), then a object is classified to the kth category or the remaining categories using the kth classifier. One-to-one [16] method trains a classifier between two sub-categories, so there will be $k(k-1)/2$ classifiers for a k-type problem.

In the fine-grained classification problem, there is a stronger similarity between sub-categories. Tradition classification models not use the similarity between sub-category and the difference between the similarities of sub-category. This paper establishes SD-SVM classification model that learns these information at the training stage to advance the accuracy of fine-grained classification.

3 Methods

In this subsection, we will introduce proposed fine-grained visual recognition algorithm. We first introduce how to get the bounding box of object in the image. Then we introduce the feature extraction using the bilinear CNN. Finally, we introduce improved SD-SVM multi-classifier.

3.1 Generating Bounding Box

We see a classification network as a mapping $y = f(x)$, where y is the final score vector and x is the input image. The mapping of the input layer to the conv1 layer is expressed as $f^{(1)}$, the mapping from the nth layer to the (n+1)th layer is $f^{(n)}$. So the whole network can be expressed as $f(x) = f^{(n)} f^{(n-1)} \cdots f^{(1)}(x)$. We defined $g^{(k)} = f^{(k)} f^{(k-1)} \cdots f^{(1)}(x)$. The output of the kth layer is expressed as $x^{(k)} = g^{(k-1)}(x)$.We can compute the gradient of y for input layer x:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial g^{(n-1)}} \frac{\partial g^{(n-1)}}{\partial g^{(n-2)}} \cdots \frac{\partial g^{(1)}}{\partial x} = \frac{\partial g_i^{(n)}}{\partial x^{(n)}} \frac{\partial g^{(n-1)}}{\partial x^{(n-1)}} \cdots \frac{\partial g^{(1)}}{\partial x} \tag{1}$$

The input image consists of three channels. For each pixel of the input image, we calculate the gradient of the three channels of y, and take the average of the three gradients as the gradient value of the pixel. We got a gradient image of the same size as the original image. As shown in Fig. 1(b). Since the gradient map obtains the most relevant part of the object, it may not have the information of the whole object. So we use the GraphCut [18] algorithm to obtain the mask of object segmentation. The advantage of the algorithm is taking advantage of some of the foreground information and the continuity of color. Then we obtain the enclose mask with the smallest rectangular border, which is the

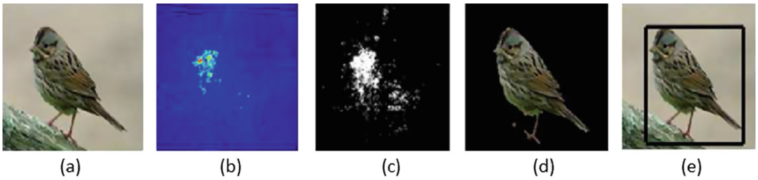


Fig. 1. An example of calculating gradient of pixel. (a) is the input image, (b) is the gradient maps, (c) is the foreground image, (d) is the result of GrapCut, (e) is the result of bounding box.

bounding box we need. We acquire the foreground information of image using the threshold gradient map. In our experiments, we use the region which gradient is greater than 95% as foreground. The foreground information is shown in Fig. 1(c). The mask of image and bounding boxes are showing in Fig. 1(d) and Fig. 1(e) respectively.

3.2 Feature Extraction

We extract the features by bilinear CNN model. A bilinear model M consists of a quaternion: $M = (f_A, f_B, P, C)$. f_A and f_B represents feature function that extract from bilinear CNN network A and B respectively. P is a pooling function and C is a classification function. The output of two networks is transformed into the final feature by bilinear operation and pooling function. For more details about bilinear CNN, please refer this paper [2].

3.3 Improved SD-SVM Multi-classifier

At the training stage, we extract the features of the training image set using B-CNN, and use the one-to-rest strategy to obtain 200 SVM classifiers. At the testing stage, the feature of testing image I_i is expressed as f_i. Using the return values of 200 trained SVM classifiers, we can link two hundred return values into a new feature vector f_s^i for image$i$, s means the feature come from SVM, and $f_s^i(u)$ is the return value that the image is classified into u class. The forms of our feature vector is different from the previous classification methods that select the maximum value of 200 return values. The category of image is assigned according to the loss function in traditional SVM classifier. The objective function is as follows:

$$\begin{aligned} & v = \arg\min_u \; loss(f_s^i, u) \\ & s.t. \\ & loss(f_s^i, u) = -f_s^i(u) \end{aligned} \tag{2}$$

The return value of SVM represents the distance between the vector and the optimal decision surface in the vector space. This method receives good results when the maximum return value is large. However, with the decrease of the maximum return value, the accuracy of classification becomes worse, as shown in Table 1. The test set has 5794 pictures, and we count the accuracy

Table 1. The performance of different maximum return value

$\max(f_s^i)$	Correct number	Total	Accuracy
$<inf$	4872	5794	84.09%
<0.8	2601	3489	74.55%
<0.6	921	1632	56.43%
<0.4	87	291	29.9%

with pictures whose $\max(f_s^i)$ are respectively lower than inf, 0.8, 0.6, 0.4. Total represents the images within range respectively, and we give the corresponding accuracy.

For this problem, we set a threshold ε_1 for the maximum return value. We proposed SD-SVM model which is a combination of two correct methods S-SVM (SVM Modified by similarity of categories) and D-SVM (SVM Modified by difference between the similarities of sub-categories) to correct the classification model when $\max(f_s^i) < \varepsilon_1$. The threshold is derived from the distribution of the return value of the training set. We let ε_1 fit: $p(f_s^i(u) > \varepsilon_1 | I_i \notin u) = 0.03$. The formula means the probability of the u-th return value of image i is greater than ε_1 equal 0.3 when image i is not belong to u. We estimate the probability as follow:

$$\widehat{p}(f_s^i(u) > \varepsilon_1 | I_i \notin u) = \frac{\sum_{f_s^i(u)>\varepsilon_1, I_i \notin u} 1}{\sum_{I_i \notin u} 1} \tag{3}$$

Modified by similarity of categories. In fine-grained classification, the distinct between different categories is smaller than other classification problems, many sub-categories shares strong similarity. Using the SVM return value of the training set, we can create a category similarity matrix: $M_{uv} = E(f_s^i(u)), I_i \in v$. $E(\cdot)$ is the expect function. Let $f_p^v = [M_{1v}, M_{2v}, \cdots M_{200v}]^T$ represent the priori feature of category v. When the category u and v are similar, the value of M_{uv} is relatively large. So we build the S-SVM model:

$$\begin{array}{l} v = \arg\min_v \ loss(f_s^i, f_p^v, v) \\ s.t. \\ loss(f_s^i, f_p^v, v) = -f_s^i(v) + 1/n * \sum_{f_p^v(u)>\varepsilon_2} (f_s^i(u) - f_p^v(u))^2 \end{array} \tag{4}$$

where n is the number satisfied $f_p^v(u) > \varepsilon_2$. We judge the condition of $f_p^v(u) > \varepsilon_2$ to determine whether the two categories are similar enough. ε_2 is decided from the distribution of the return value of the training set. For example, we assume that there are three similar categories for each category on average, so we need to get a similar relationship of 1.5% for two hundred categories of bird datasets. We have to calculate ε_2 satisfying: $p(f_p^v(u) > \varepsilon_2 | u \neq v) = 0.015$. We estimate the probability as follow:

$$\widehat{p}(f_p^v(u) > \varepsilon_2 | u \neq v) = \frac{\sum_{f_p^v(u)>\varepsilon_2, u \neq v} 1}{\sum_{u \neq v} 1} \tag{5}$$

In order to showing the results of classification using our first kind of correction model(S-SVM). Figure 2 shows an example of the successful classification. Figure 2(e) is a test image which belongs to category 1 (Black footed Albatross). Figure 2(a) is an example of category 1, and Fig. 2(b) to (d) is a image that is similar to the category 1. Figure 2(f) is an example of category 45 (Northern Fulmar). Figure 2(g) to (i) is a image that is similar to the category 45. The return value of Fig. 2(e) for category 1 is 0.3923, and the return value of Fig. 2(e) for category 45 is 0.4462. Using the traditional classification method, this image is classified as category 45 incorrectly. Since the overall similarity of the image in the first line is higher than that of the third line, the image is classified correctly in our model.

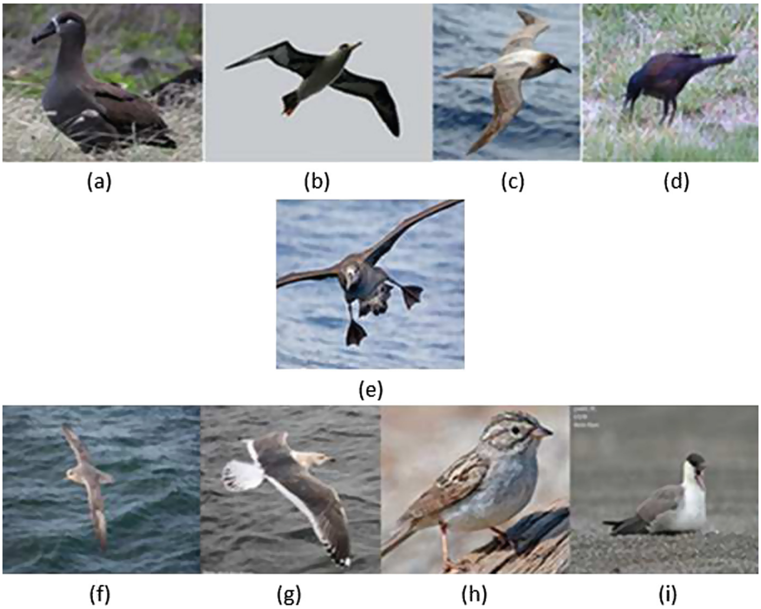


Fig. 2. A successful example for classification with S-SVM. (e) is the image to be classified. (a) and (f) are the two categories which own the highest score. (b), (c), (d) are the closest three categories to (a). (g), (h), (i) are the closest three categories to (f). Since the similarity between (e) and (a)–(d) is greater than that of (f)–(i), we correctly classify (e) as category (a).

Modified by difference between the similarities of sub-categories. Since the similarity between categories is affected by many factors, the similarity is not transitive. For example, category a is similar to category b as black mouth, and category b is similar to category c as pointed mouth, however, category a is not similar to category c. When it is difficult to distinguish between category a and category b for given image, we can judge by the similarity information between the image and category c.

Based on this idea, we create a strongly discriminant category set $\omega(v)$ for each category v. We use $S(u, v)$ to express the ability of category u to distinguish category v with other categories similar to category v. $f_p^v(w) > \varepsilon_2$ means category w is similar to category v. We define $max3(\cdot)$ as a function return three of the maximum in the input. We selected the largest of the three in $S(u,v)$ into $\omega(v)$. So we build the D-SVM model:

$$\begin{array}{l} v = \arg\min_v \ loss(f_s^i, f_p^v, v) \\ s.t. \\ loss(f_s^i, f_p^v, v) = -f_s^i(v) + 1/n * \sum_{u\in\omega(v)} (f_s^i(u) - f_p^v(u))^2 \\ \omega(v) = max3(S(u,v)) \\ S(u,v) = \sum_{f_p^v(w)>\varepsilon_2} (f_p^v(u) - f_p^w(u))^2 \end{array} \tag{6}$$

In order to showing the results of classification using our second kind of correction model(D-SVM). Figure 3 shows an example. Figure 3(d) is an image in the test set, and it's category is category 41 (Scissor tailed Flycatcher). Figure 3(a) to (c) are sample images of categories 41, category 189 (Red bellied Woodpecker), category 36 (Northern Flicker), category 189 is similar to category 36 and category 41, while category 41 is not similar to category 36. The return value of Fig. 3(d) for category 41 is 0.4254, and the return value for category 189 is 0.4649. In the traditional classification method, this image is misclassified as category 189. Since Fig. 3(d) is not similar to category 36, the image is classified as category 41 correctly in our model.

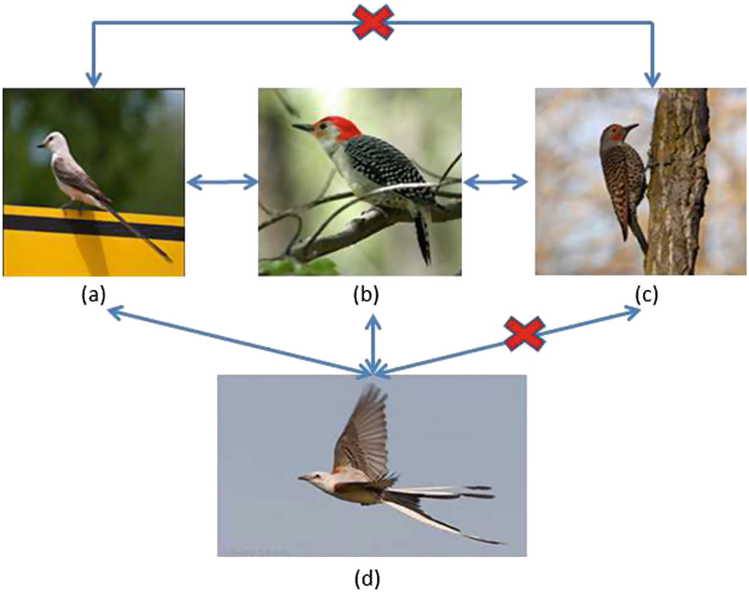


Fig. 3. A successful example for classification with D-SVM. (d) is the image to be classified. (a) and (b) are the two categories which own the highest score. (a) is not similar with (c). Meanwhile (b) is similar with (c). Since result shows (d) is not similar with (c). So we correctly classify (d) as category (a).

4 Experimental Results and Analysis

In order to verify that the proposed method is effective to improve the accuracy of fine-grained visual recognition, experiments are done in two fine-grained visual identification datasets (Caltech-UCSD Birds-200-2011 and FGVC-aircraft). At the stage of getting the bounding box, we use the DeCAF using the CNN framework that is provided by [1]. The DeCAF is trained on the ILSVRC 2012 dataset.

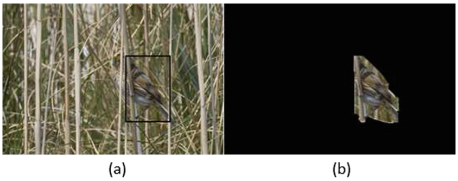
4.1 CUB-2011

The CUB-2011 dataset consists of 11,788 bird images belonging to 200 subcategories. We train the B-CNN [D, M] network based on the training set as [2]. We get $\varepsilon_1 = 0.58$, $\varepsilon_2 = 0.18$ in the training stage based on the method in Sect. 3.3.

Use Bounding box. We separately use the original image and the image intercepted by bounding box as the input. The experimental results are shown in Table 2. The method using the original image achieves 84.09% accuracy. The method using Bounding box achieves 83.74% accuracy. The accuracy is slightly lower than the accuracy of the original image. While the correct Bounding box is ful for removing the background of the image, the method of acquired Bounding box can not achieve the effect of manual calibration. Some images also lose important information of the foreground while removing the background, as is show in Fig. 4. The third method combining original image and bounding box is to select the maximum return value between two return values, which effectively avoids the risk of loss foreground information. This method achieves 84.62% accuracy. Table 2 is show the results of three method.

Table 2. The performance of different input image

Input	Accuracy
Original	84.09%
Bbox	83.74%
Original + Bbox	84.62%


(a)

(b)

Fig. 4. An example of incorrect bounding box. (a) shows the wrong bounding box, (b) shows the defective foreground image obtained by wrong bounding box.

We also tried to calculate the bounding box for the training image during the training stage, but the result is so bad because the training stage could not avoid the wrong bounding box as the test stage. The network will train the wrong image as the correct category.

Correct SVM multi-classification. Table 3 shows the results of classification using our proposed S-SVM, D-SVM, SD-SVM. S-SVM classification method improved the accuracy by 0.3%. D-SVM classification method also increased the accuracy by 0.3%. While we adopt loss function using both correction methods to modify loss function, the accuracy of our model achieves 84.55%. Combining with the bounding box and SD-SVM classification algorithm, our proposed method can further improve the accuracy of fine-grained classification. Furthermore, we firstly extract the features of bounding box image and the original image, and then calculate the loss of classification using the extracted feature and SD-SVM algorithm. The final category is determined by the minimum loss function value. The final result is up to 85.12%.

Table 3. The performance on variants of our method

Algorithm	Accuracy
B-CNN	84.09%
B-CNN + S-SVM	84.40%
B-CNN + D-SVM	84.40%
B-CNN + SD-SVM	84.55%
B-CNN + Bbox + SD-SVM	85.12%

Comparison of threshold ε_1. The threshold ε_1 is a parameter that is used to determine the dividing line of the improved algorithm. This paper gets ε_1 value from the distribution of training set. Figure 5 shows the experimental results through manual changing ε_1. We can see that all three methods get higher accuracy opposite to traditional methods. When the threshold equals 0, our method is equivalent to the traditional method. The accuracy rise along with the threshold, and it falls until coming to the point about 0.5 or 0.6. The bigger of the return value, the more reliable the value is. So, our method get little improvement in this case. Besides, we found that SD-SVM usually outperforms S-SVM and D-SVM. We can see that when the maximum of SVM return value is less than a certain value, our method is always better than the unmodified method. While using the distributed of training set to obtain the threshold can also achieves a good result.

Comparison with previous works. Table 4 shows the results of existing best algorithms and our proposed method. Our approach is superior to all current fine-grained recognition algorithm with only category labels, and is better than

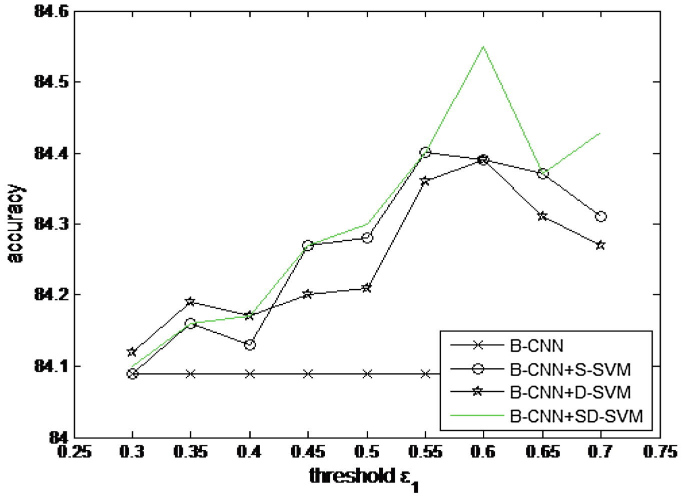


Fig. 5. The performance of different values of ε_1.

Table 4. Comparison of different methods on CUB-2011

Method	Train label	Test label	Accuracy
ours	n/a	n/a	85.12%
Two attention(2015)[12]	n/a	n/a	77.9%
STN(2015)[5]	n/a	n/a	84.1%
B-CNN(2015)[2]	n/a	n/a	84.09%
Pick filter(2016)[3]	n/a	n/a	84.54%
No part(2015)[9]	Bbox	n/a	82.0%
SPDA(2016)[4]	Bbox+Parts	Bbox	85.14%
FOAF(2014)[15]	Bbox+Parts	Bbox+Parts	81.2%
PN-CNN(2014)[8]	Bbox+Parts	Bbox+parts	85.4%

most algorithms of the fine-grained classification with the expensive manual annotation. The accuracy of our method is not as good as PN-CNN [8] and SPDA [4], which uses manual annotation of the Bbox level and part level during the training and testing stages.

4.2 FGVC-Aircraft

The FGVC-aircraft dataset consists of 10,000 aircraft images belonging to 100 subcategories. We use the B-CNN [D, D] network to train the training set as [2]. We get $\varepsilon_1 = 0.49$, $\varepsilon_2 = 0.20$ in the training stage based on the method in Sect. 3.3.

Comparison with previous work. Table 5 shows the results of different methods. Since the FGVC-aircraft data set does not provide Bbox-level annotation, many existing fine-grained classification algorithms can not be realized. As we can see from the table, our method achieves the highest accuracy.

Table 5. Comparison of different methods on FGVC-aircraft

Algorithm	Accuracy
Symbiotic segmentation(2013)[10]	72.5%
FV-SIFT(2014)[14]	80.7%
B-CNN(2015)[2]	84.1%
ours	85.21%

5 Conclusion

In this paper We propose a fine-grained visual classification algorithm based on image foreground and sub-category Similarity. Our algorithm combines the advantages of unsupervised object detection algorithm, feature extraction of bilinear CNN, as well as the similarity of sub-category. Experimental results show that our method outperforms state-of-the-art fine-grained categorization methods with only the category labels.

Acknowledgement. This work is supported by Provincial Natural Science Foundation of Jiangsu (Grant Nos. BK20151260, BK20151254), Six talent peaks project in Jiangsu Province (DZXX-027), National Natural Science Foundation of China (NSFC Grant Nos. 61272258, 61170124, 61301299, 61272005), Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University (Grant No. 93K172016K08), and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

1. Donahue, J., Jia, Y., Vinyals, O., et al.: DeCAF: a deep convolutional activation feature for generic visual recognition. In: International Conference on Machine Learning, pp. 647–655 (2014)
2. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear cnn models for fine-grained visual recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1449–1457 (2015)
3. Zhang, X., Xiong, H., Zhou, W., et al.: Picking deep filter responses for fine-grained image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1134–1142 (2016)
4. Zhang, H., Xu, T., Elhoseiny, M., et al.: SPDA-CNN: unifying semantic part detection and abstraction for fine-grained recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1143–1152 (2016)

5. Jaderberg, M., Simonyan, K., Zisserman, A.: Spatial transformer networks. In: Advances in Neural Information Processing Systems, pp. 2017–2025 (2015)
6. Zhang, N., Donahue, J., Girshick, R., Darrell, T.: Part-based R-CNNs for fine-grained category detection. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 834–849. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_54
7. Girshick, R., Donahue, J., Darrell, T., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
8. Branson, S., Van Horn, G., Belongie, S., et al.: Bird species categorization using pose normalized deep convolutional nets. arXiv preprint arXiv. 1406.2952 (2014)
9. Krause, J., Jin, H., Yang, J., et al.: Fine-grained recognition without part annotations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5546–5555 (2015)
10. Chai, Y., Lempitsky, V., Zisserman, A.: Symbiotic segmentation and part localization for fine-grained categorization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 321–328 (2013)
11. Jaderberg, M., Simonyan, K., Zisserman, A.: Spatial transformer networks. In: Advances in Neural Information Processing Systems, pp. 2017–2025 (2015)
12. Xiao, T., Xu, Y., Yang, K., et al.: The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 842–850 (2015)
13. Simon, M., Rodner, E.: Neural activation constellations: unsupervised part model discovery with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1143–1151 (2015)
14. Gosselin, P.H., Murray, N., Jgou, H., et al.: Revisiting the fisher vector for fine-grained classification. Pattern Recognit. Lett. **49**, 92–98 (2014)
15. Zhang, X., Xiong, H., Zhou, W., et al.: Fused one-vs-all mid-level features for fine-grained visual categorization. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 287–296. ACM (2014)
16. Li, H., Qi, F., Wang, S.: A comparison of model selection methods for multi-class support vector machines. In: Gervasi, O., Gavrilova, M.L., Kumar, V., Laganá, A., Lee, H.P., Mun, Y., Taniar, D., Tan, C.J.K. (eds.) ICCSA 2005. LNCS, vol. 3483, pp. 1140–1148. Springer, Heidelberg (2005). https://doi.org/10.1007/11424925_119
17. Kreßel, U.H.G.: Pairwise classification and support vector machines. In: Advances in Kernel Methods, pp. 255–268. MIT Press (1999)
18. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In: Proceedings of the Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol. 1, pp. 105–112. IEEE (2001)