Journal of Computers Vol. 29 No. 3, 2018, pp. 12-18
doi:10.3966/199115992018062903002

A Data Augment Method for Fine-grained Recognition

Yin Zhang*, Zhifeng Hu, Shenjing Tian

College of Computer Science, Zhejiang University, Hangzhou 310027, China
{yinzh, huyangc, 21621111}@zju.edu.cn

Received 21 October 2016; Revised 6 March 2017; Accepted 30 March 2017

Abstract. Fine-grained recognition is a very challenge problem, because of the similarity between different subcategories and scarce training data. Even in the same subcategories, there can be some differences due to the distinct color and pose of objects. We focus our thoughts on the details of specific object parts to settle these limitations. We propose a model for fine-grained recognition by taking advantage of deep Convolutional Neural Network (CNN) combined with bottom-up region proposals. Our method evaluates these proposals and utilize the evaluated proposals to determine the subcategory of the object. Our final result shows that our methods can increase the accuracy by 2-3% on average.

Keywords: convolutional neural network, fine-grained recognition, region proposal

1 Introduction

Object recognition is one of the major focuses of research in computer vision. Most of existing recognition tasks are on basic-level: distinguishing between table, human, computer, car and so on. On basic-level recognition, categories differ greatly from each other. On contrast, fine-grained recognition concentrates on differences between subcategory (breeds, species or product models), for example, classification of different species of birds or species of flowers, which means similarities existing across categories and subtle differences needed to be found.

Deep Convolutional Neural Networks (CNNs) achieve high success in many computer vision tasks [6, 16, 21], like Krizhevsky et al. [14] achieved an impressive result using a CNN in ILSVRC2012 [21], Girshick et al. [10] achieved breakthrough in basic-level object recognition on Pascal VOC dataset [6] by applying a set of bottom-up candidate region proposals. Deep CNN can be used to boost the accuracy not only in basic-level object detection, but in fine-grained recognition [22, 30, 32] as well. Because of the non-linear characteristic and great amounts of parameters to be trained, CNN demands strong computing power and a large number of training data. However, in fine-grained recognition, labeled training data is scarce due to the similarity between different categories, therefore, expertise knowledge is needed when labeling data. For example, can you recognize the species in Fig. 1. In Fig. 1, we can see the similarity between different subcategories (first row), and differences (color, pose) in the same subcategory (second row).

(a) windflower (b) balloon flower (c) balloon flower (d) balloon flower

Fig. 1. First row

* Corresponding Author

We consider we can boost the accuracy of fine-grained recognition by forcing the CNN to focus on subtle parts in the image. We choose selective search [24] to generate bottom-up region proposals according to the conclusion described by Zhang et al. [30] that selective search can cover most of the significant parts. We present state-of-the-art results on Oxford 102 Flowers. Source code of our experiments is available at https://github.com/huyangc/flower-classify

In this paper, we propose a training procedure to settle the limitations of fine-grained recognition:

- Combine the part-based methods and CNN by utilizing the selective search to generate bottom-up region proposals.
- Method to evaluate the region proposals without ground truth bounding boxes.

2 Related Work

2.1 Traditional Methods for Fine-grained Recognition

A variety of methods about fine-grained classification have been proposed in recent years, e.g. [1-4, 18-19, 25, 29]. As described in [1], we can divide these methods into these prominent directions: (1) Segmentation. (2) Region pooling, including pose-normalized pooling [31] and template-based pooling [29]. These methods have something in common: 1)algorithms designed to find out the significant parts in the images, (2) designing features and extracting these features using these parts, (3) classifier like SVM trained to get the recognition result. For example, Angelova et al. [2] first detects object in image and then extracts features of the segmentation using super-pixel segmentation [8], global pooling of HOG features [5], encoding by LLC method [26], then trains an SVM classifier [7] with the help of ground truth segmentation of training images.

2.2 Convolution Neural Networks

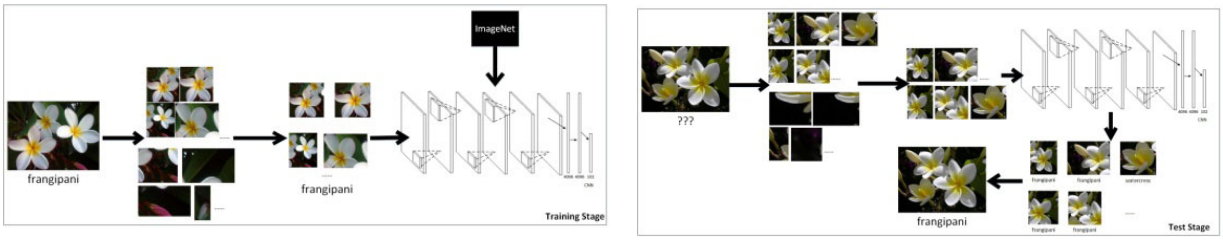
CNN was heavily used in the 1990s, first was popularized by LeCun [15] to use in digit recognition, but fell out of fashion because of the requirement for strong computing power and large amounts of training data. With the development of parallel computing and the construction of large image databases, CNN goes to front stage again and achieves high success in many computer vision tasks. For instance, Krizhevsky et al. [14] achieved an impressive result using a CNN in ILSVRC2012 [21] with two GPUs to accelerate the computation of CNN parameters.

Inspired by Krizhevsky et al., many groups proposed CNN architectures to solve the classification problems. In order to get a better performance, many CNNs [10, 22, 30] are first pre-trained on a large image set, ImageNet [21] for example, followed by domain-specific fine-tuning. Girshick et al. [10] proposed a model applied CNN to bottom-up region proposals and generalized the CNN classification results on ImageNet to Pascal VOC. N Zhang et al. [30] fine-tuned the ImageNet pre-trained CNN for the 200-way bird classify using the ground truth bounding box crops of the original images.

3 Proposed Method

Girshick et al. [9-10, 20] demonstrated the excellent result of CNN with regions on generic object detection task, N Zhang et al. [30] shows the effectiveness of the R-CNN method on Caltech-UCSD bird dataset [27]. The datasets they used have the ground truth bounding boxes which can be used to tell the foreground and background. The ground truth bounding boxes are manually labelled which leads to the scarcity of training data. Inspired by the Segmentation methods on Oxford 102 Flowers, we conjecture that we can make use of the CNN trained by specific parts in images other than only using the whole image to obtain a better result without the help of ground truth bounding boxes.

The procedure of our experiment is demonstrated in Fig. 2.



(a) Training Stage: i. Use selective search to generate part proposals. ii. Eliminate noisy proposals. iii. Label the filtered proposals the same as the original image, then using all the labelled proposals to fine-tune the pre-trained ImageNet CNN.

(b) Test Stage: i. Use selective search to generate part proposals. ii. Eliminate noisy proposals. iii. Obtain the labels of proposals through the fine-tuned CNN. iv. Get the label of the test image through the proposals.

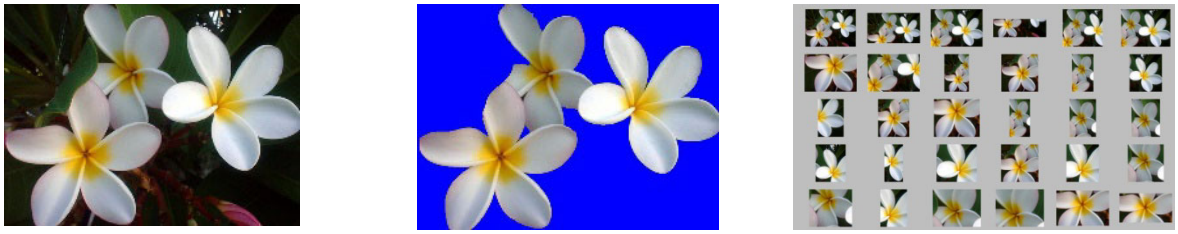
Fig. 2. Procedure of our experiments

3.1 Part Proposals

We choose to follow Girshick et al. using selective search as our part proposals method for these reasons:

(1) Selective search can generate part proposals that cover most of the interesting part we concern.

(2) Unlike segmentation methods which generate only the main part of image, selective search can generate more images from an original image, as Fig. 3 shows.



(a) Top row: origin image

(b) segmentation result

(c) selective search's proposals (after filtration)

Fig.3. Selective search's proposals and segmentation result

We set the parameters of selective search as: $k = 200$, $minSize = 200$, $\sigma = 0.8$. After selective search, we get $X = \{x_{i0},x_{i1},...,x_{in}\}(i \in [1,N])$ where N is the size of training set. n denotes number of images generated by selective search on one image, x_{ik} means the k-th proposal of i-th image on training set.

Considering the prior of Oxford 102 Flowers dataset, which is only consisted of flower images, we first fine-tune the CNN in Krizhevsky et al. [14] (AlexNet) to obtain a binary classifier (flower or not). In particular, we replace the last 1000-way full-connected layer (fc8) with a new 2-way full-connected layer whose weights are randomly initialized using a Gaussian with $\mu = 0$ and $\sigma = 0.01$. Then we use the classifier to filter the part proposals. As Fig. 3 shows, after filtration, we obtain proposals that all about flowers.

3.2 Fine-grained Recognition

We label x_{ik} the same as x_i at training. After that, we use all the labelled proposals as training set to enlarge the training data. On test step, for a test image, we utilize the recognition results of proposals to determine the recognition result of the test image. Representation of proposals recognition results is $[\phi(x_{i0})...\phi(x_{in})]$ and $[score(x_{i0})...score(x_{in})]$, where $\phi(x_{ik})/score(x_{ik})$ represents the CNN recognition result/score of k-th part in x_i. We first calculate the average score of x_i on each label θ.

$$p_i(\theta) = \begin{cases} \frac{1}{n_\theta} \cdot \sum score(x_{ik}), & (x_{ik} \mid \phi(x_{ik}) = \theta, k \in [1, n]) \\ 0 & \text{otherwise} \end{cases}$$

n_θ denotes the number of proposals whose label is $\theta, \theta \in [1,102]$. Then we label this test image the same as the label of maximum average score.

$$\text{label}(x_i) = \text{argmax}(p_i(\theta)), \theta \in [1,102]$$

In our experiments, we use the CNNs (AlexNet [14], GoogLeNet [23]) pre-trained on ImageNet. Moreover, we fine-tune the pre-trained CNNs to fit the 102-way Oxford 102 Flowers dataset. Specifically:

(1) AlexNet experiment: we replace the last 1000-way full-connected layer (fc8) with a new 102-way full-connected layer whose weights are randomly initialized using a Gaussian with $\mu = 0$ and $\sigma = 0.01$. We set the base learning rate to a tenth of AlexNet and drop it by a factor of 10 throughout training (per 10000 iterations), with a momentum 0.9 to avoid over fitting (Learning rate: $\alpha = 0.001$, Momentum: $\mu = 0.9$).

(2) GoogLeNet experiment: we set the fine-tune parameters just the same as AlexNet experiment letting $\alpha = 0.001$, $\mu = 0.9$. We replace each of the three 1000-way loss/classifier layers with a new full-connected 102-way layer whose weights are initialized by Xavier algorithm [11].

We warp each proposal to 227 × 227 to fit the network input size, and use softmax to generate every label score for each proposal. We use Caffe [13] to train the filter of proposals and fine-tune the pre-trained CNNs.

4 Evaluation

4.1 Dataset

We evaluate our method on fine-grained recognition dataset 102 Flowers. Oxford 102 flowers dataset [17] contains 102 categories of all 8,189 images. Each category contains 40 to 258 of images. The dataset is split into training, test, and validation sets. In training and validation sets, each category contains 10 images. The test set consists of the remaining 6,149 images.

We use the provided segmentations of all the 102 Flowers images as positive examples (flower) together with the leaves image in Flavia [28] and 2013 ImageCLEF's plant identification task [12] as negative examples. We manually remove the images in which leaf or flower is not the main part. Fig. 4 shows one of the positive examples and negative examples as well as the preferred image and noisy image. We get 14000 images of leaves and flowers in the end. These images are divided into training set (4900 flower images and 4900 leaf images) and test set (2,100 flower images and 2,100 leaf images).

(a) Negative example, preferred image

(b) Noisy image

(c) Positive example, preferred image

(d) Noisy image

Fig. 4. Examples of noise data

4.2 Results

Proposals filtration. We train two classifiers to identify whether the segmentations of dataset are important or not. One uses the Oxford 102 Flowers segmentations as the positive examples and another just uses the original Oxford 102 Flowers dataset image. Both of the classifiers trained for the proposals

filtration achieve 99% of binary classification accuracy on our test set.

Fine-grained recognition. We present the result of our experiments in Table 1. We first present the result of improvement by using selective search to enlarge the training set. Then we investigate the influence of proposals size on training stage and test stage separately.Without filtering the noisy proposals, we can not make the pre-trained CNN converge. So all of our experiments have the procedure of proposals filtration. The results show that we can get a higher accuracy when adding a size constraint to the proposals on both training stage and test stage. Because the input size of both CNNs is 227×227, when proposals are warped to the size, the smaller proposals will be added on some noise. Therefore, we get a lower accuracy no matter in 'train with all size of proposals' or 'test with all size of proposals' experiments. It is very interesting that we get a higher accuracy when just using the whole image to fine-tune the AlexNet than GoogLeNet. And we get a more growth when using the data augment method in finetuning GoogLeNet than AlexNet. We think it is because of the structure differences that GoogLeNet needs more training data to be more discriminative for fine-grained recognition task. In the end, it shows that it is almost nothing different whether to use the segmentations or not in proposal filtration.

Table 1. The result of our experiments

Training mode	AlexNet	GleNet
Train: whole image; Test: whole image	83.35%	75.85%
Train: whole image; Test: whole image, seg.	83.85%	76.15%
Train: all size proposals; Test: all size proposals	79.05%	77.79%
Train: all size proposals; Test: all size proposals, seg.	79.35%	77.89%
Train: whole image; Test: whole image	79.93%	78.21%
Train: whole image; Test: whole image	80.12%	78.63%
Train: whole image; Test: whole image	75.72%	72.27%
Train: whole image; Test: whole image	75.83%	72.44%
Train: whole image; Test: whole image	79.05%	81.12%
Train: whole image; Test: whole image	79.65%	81.42%
Train: whole image; Test: whole image	85.95%	88.24%
Train: min (size) ≥227proposals; Test: min (size) ≥227proposals, seg.	86.65%	88.40%
Train: whole image; Test: whole image	83.72%	85.79%
Train: whole image; Test: whole image	83.91%	85.86%

Note. "GLe Net" means Goog Le Net, "Train" means on training stage, "Test" means on test stage, "whole" means we use the whole image as input of CNN without data augment, "min (size)" means the smaller one between heigh tand width of proposal's bounding box. "seg." means using segmentations as the positive example when training proposal filter.

We compare our method with the other start-of-the-art methods. Table 2 shows our state-of-the-art accuracy.

Table 2. Comparison to other methods

Method	Accuracy
Nilsback and Zisserman [17]	72.80%
Nilsback [18]	76.30%
Chai et al. [3]	80.00%
Angelova et al. [2]	80.66%
Chai et al. [4]	85.20%
Razavian et al. [22]	86.80%
Ours (from Table 1)	88.40%

5 Conclusion

We have proposed a training procedure for training set augment in fine-grained recognition to settle the limitations, which is capable of state-of-the-art methods. Our method utilizes the bottom-up region proposals and pre-trained CNN to boost the accuracy of fine-grained recognition. Our experiments show that it is highly beneficial to force CNN to focus on significant parts of object other than the whole image

by bottom-up region proposals. In the meantime, proposals can also enlarge the training data for CNN to settle the scarce training data limitation of fine-grained recognition. It is also important to point out that we obtain almost the same accuracy without the help of segmentations on proposal filtration. In future extensions of this work, we will consider using CNN features of proposals to classify the object other than the CNN classify results. We also plan to investigate the influence of features from different CNN layer, because we think that we should always focus on significant subtle parts rather than the whole image in fine-grained recognition. Finally, we will explore the way to accelerate the recognition procedure to obtain both accuracy and speed.

Acknowledgments

This work is supported by China Knowledge Centre for Engineering Sciences and Technology (No. CKCEST-2017-1-3) and Zhejiang Provincial Natural Science Foundation of China (No. LY14F020027).

References

[1] A. Angelova, P.M. Long, Benchmarking large-scale fine-grained categorization, in: Proc. IEEE Winter Conference on Applications of Computer Vision, 2014.

[2] A. Angelova, S. Zhu, Efficient object detection and segmentation for fine-grained recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2013.

[3] Y. Chai, V.S. Lempitsky, A. Zisserman, Bicos: a bi-level co-segmentation method for image classification, in: Proc. ICCV, 2011.

[4] Y. Chai, E. Rahtu, V. Lempitsky, L. Van Gool, A. Zisserman, Tricos: a tri-level classdiscriminative co-segmentation method for image classification, in: Proc. European Conference on Computer Vision, 2012.

[5] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection. in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005.

[6] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International Journal of Computer Vision 88(2)(2010) 303-338.

[7] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, C.J. Lin, Liblinear: a library for large linear classification, Journal of Machine Learning Research 9(2008) 1871-1874.

[8] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient graph-based image segmentation, International Journal of Computer Vision 59(2)(2004) 167-181.

[9] R. Girshick, Fast r-cnn, in: Proc. IEEE International Conference on Computer Vision, 2015.

[10] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[11] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proc. the 13th International Conference on Artificial Intelligence and Statistics 2010, 2010.

[12] H. Goēau, P. Bonnet, A. Joly, V. Bakic, D. Barthélémy, N. Boujemaa, J.F. Molino, Thé imageclef 2013 plant identification task, in: Proc. the 2nd ACM International Workshop on Multimedia Analysis for Ecological Data, 2013.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, arXiv:1408.5093, 2014.

[14] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Proc. Advances in Neural Information Processing Systems, 2012.

[15] Y. LeCun, L., Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, IEEE 86(11)(1998) 2278-2324.

[16] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, C.L. Zitnicḱ, Microsoft coco: common objects in context, in Proc. European Conference on Computer Vision, 2014.

[17] M.E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: Proc. the Indian Conference on Computer Vision, Graphics and Image Processing, 2008.

[18] M.E. Nilsback, A. Zisserman, An automatic visual flora-segmentation and classification of flower images, Oxford University, 2009.

[19] O.M. Parkhi, A. Vedaldi, A. Zisserman, C. Jawahar, Cats and dogs, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[20] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, in: Proc. Advances in Neural Information Processing Systems, 2015.

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, C.B. Alexander, F.-F. Li, Imagenet large scale visual recognition challenge, International Journal of Computer Vision 115(3)(2015) 211-252.

[22] A. Sharif Razavian, H. Azizpour, J. Sullivan, S. Carlsson, Cnn features off-the-shelf: an astounding baseline for recognition, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2015.

[24] J.R. Uijlings, K.E. van de Sande, T. Gevers, A.W. Smeulders, Selective search for object recognition, International Journal of Computer Vision 104(2)(2013) 154-171.

[25] C. Wah, S. Branson, P. Perona, S. Belongie, Multiclass recognition and part localization with humans in the loop, in Proc. 2011 International Conference on Computer Vision, 2011.

[26] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[27] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, P. Perona, Caltechucsd birds 200. <http://www.vision.caltech.edu/visipedia/CUB-200.html>, 2010.

[28] S.G. Wu, F.S. Bao, E.Y. Xu, Y.X. Wang, Y.F. Chang, Q.L. Xiang, A leaf recognition algorithm for plant classification using probabilistic neural network, in Proc. IEEE International Symposium on Signal Processing and Information Technology, 2007.

[29] S. Yang, L. Bo, J. Wang, L.G. Shapiro, Unsupervised template learning for fine-grained object recognition, in Proc. Advances in Neural Information Processing Systems, 2012.

[30] N. Zhang, J. Donahue, R. Girshick, T. Darrell, Part-based r-cnns for fine-grained category detection, in: Proc. European Conference on Computer Vision, 2014.

[31] N. Zhang, R. Farrell, T. Darrell, Pose pooling kernels for sub-category recognition, in: Proc. IEEE Computer Vision and Pattern Recognition (CVPR), 2012.

[32] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, L. Bourdev, Panda: pose aligned networks for deep attribute modeling, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2014.