Journal of Communication and Computer 15 (2019) 14-24
doi: 10.17265/1548-7709/2019.01.002

A Flower Image Classification Algorithm Based on Saliency Map and PCANet

Yan Yangyang, and Fu Xiang
Institute of Computer Vision, Nanchang Hang Kong University, Nanchang, China

Abstract: Flower Image Classification is a Fine-Grained Classification problem. The main difficulty of Fine-Grained Classification is the large inter-class similarity and the inner-class difference. In this paper, we propose a new algorithm based on Saliency Map and PCANet to overcome the difficulty. This algorithm mainly consists of two parts: flower region selection, flower feature learning. In first part, we combine saliency map with gray-scale map to select flower region. In second part, we use the flower region as input to train the PCANet which is a simple deep learning network for learning flower feature automatically, then a 102-way softmax layer that follow the PCANet achieve classification. Our approach achieves 84.12% accuracy on Oxford 17 Flowers dataset. The results show that a combination of Saliency Map and simple deep learning network PCANet can applies to flower image classification problem.

Key words: Saliency map, PCANet, deep learning, flower image classification.

1. Introduction

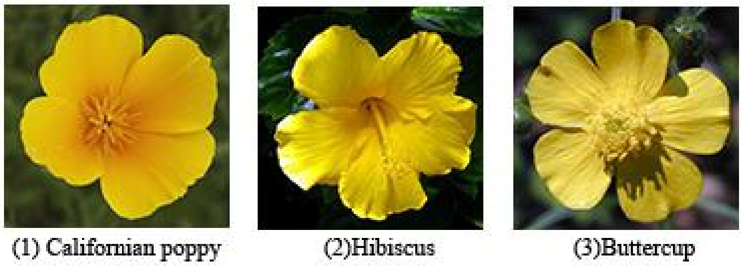
Flower classification is an important topic in the field of botany, which is the basis of botany related work. With a wide variety of flowers, the current types of flowers have reached 250,000, which is one of the most prosperous species in the world. Experienced plant taxonomists observe flowers overall characteristics such as color, texture, shape, and study their living environment, then compare with the recorded specimens, eventually determine the genera of flowers. However, this process relies on the experience and expertise knowledge of the taxonomists.

With the popularity of smartphones, people can take clear pictures of flowers easily. So it is a research idea to achieve flower classification by studying flower images. However, most of image classification research is coarse-grained image classification which need to classify faces, bicycles, cats, dogs and other unrelated categories. The largest dataset is ImageNet [1], which is used to evaluate the image classification algorithm. It contains 14,197,122 high resolution images with labels in 21841 categories. Many coarse-grained classification methods are springing up in annual ImageNet Large Scale Visual Recognition Challenge, such as AlexNet [2], ConvNet [3], GoogleNet [4], ResNet [5].

Unlike coarse-grained image classification, flower image classification is a fine-grained classification problem. Flower image classification's difficulties are inter-class similarity and inner-class diversity. There are great similarities between different types of flowers, and the same category of flowers has great diversity, like color diversity, form diversity in different time. Fig. 1(a) shows inter-class similarity, Fig. 1(b) shows the inner-class difference due to form diversity in different time. Therefore, flower image classification faces more challenges.

To deal with the above difficulties, we propose a new algorithm based on Saliency Map and *PCANet* [6] for flower image classification. This algorithm mainly consists of two parts: flower region selection, flower feature learning.

Corresponding author: Fu Xiang, Ph.D, associate professor, research field: image processing and pattern recognition, video sensor network.



(a) Inter-class similarity

(b) Inner-class difference

Fig. 1 Difference of flowers (best viewed in color).

Our contributions can be summarized as follows:

• We utilize saliency map and gray-scale map to select flower region in the original image. Because the backgrounds of different flowers are roughly the same. So the combination can remove irrelevant background and only keep the flower region's information, consequently help later feature learning exclude interference from other irrelevant part in the flower image.

• We first apply a simple deep learning network which is named as *PCANet* to flower image classification task. We adjust the parameters of network to get best classification result. Experiment results demonstrate *PCANet* can also learn flower features well.

2. Related Work

According to the analysis of current flower image classification methods, methods mainly include methods based on manual feature, methods based on deep learning.

2.1 Methods Based on Manual Feature

Those methods firstly conduct image segmentation to get the flower region in the image, then combine the hand-crafting features with traditional machine learning algorithm to implement classification. Main hand-crafting features includes color, texture, *sift* [7], *hog* [8], etc. Main traditional machine learning algorithm includes *support vector machine* [9], *kernel learning* [10], *k-nearest neighbor* algorithm [11], etc.

Nilsback and Zisserman [12] develop a visual vocabulary that represent various aspects including colour, shape and texture to distinguish categories which have significant visual similarity. They also produce a dataset consisting of 17 species of flowers with 80 images of each, and conducts experiments on it, the classification accuracy is 71.76%.

Manik Varma and Debajyoti Ray [13] investigate the problem of learning optimal descriptors for classification task. They developed an approach for learning the discriminative power-invariance trade-off that combine base descriptors like shape, colour and texture descriptors in flower image optimally in a kernel learning framework.

Hossam and his colleagues [14] conducted segmentation using color information, then used Scale Invariant Feature Transform *(sift) and* Segmentation based Fractal Texture Analysis *(SFTA)* algorithms to extract flowers' features, lastly applied Support

Vector Machine (*SVM*) and Random Forests (*RF*) algorithms to classify different kinds of flowers.

Although the above classification method gained good classification effect under certain conditions, the results depend on segmentation and manual features. And segmentation method and manual features need to be designed by experienced researchers.

2.2 Methods Based on Deep Learning

Deep learning was put forward by Hinton in 2006. Deep learning mainly simulate the high-level abstract characteristics of the data through the multi-layer nonlinear processing unit [15]. Unlike other vision methods using hand-crafted features, deep learning is able to automatically learn multiple stages of invariant feature for the specific task.

Methods based on deep learning mainly utilize deep convolution neural network [16]. Some methods are as following:

Liu Yuanyuan [17] firstly established a larger dataset of 52,775 flower images in 79 categories. And a new model based on convolution neural network is proposed, which consists of five convolution layers, each convolution layer is followed by the largest pool layer, and then connects with three fully connected layers and the *softmax* layer. On the newly dataset, 76.54% of the accuracy was obtained, while the accuracy rate was 84.02% on Oxford 102 Flowers dataset.

Xiaoling Xia et al. [18], they use the transfer learning technology to retrain the flower category datasets based on Inception-v3 model of TensorFlow platform, which can greatly improve the accuracy of flower classification.

However, above deep learning network is non-linear and have millions of parameters to be estimated, like the simplest convolution neural network lenet-5 [16], only the F6 layer contains 10164 train parameters. And it requires strong computing power for optimization and large training data to be generalized well.

So some researchers started to study the structure of deep network for reducing the complexity of the network under the condition that certain feature learning ability is guaranteed.

Tsung-han Chan et al. [19] proposed a new deep learning baseline, namely main-principal component analysis network (PCANet). The network contains only very basic processing units: cascade principal component analysis (PCA), hash coding, block histogram. This network can effectively and easily learn features.

In order to simply feature learning part of flower image classification, this paper utilizes PCANet to realize flowers' feature learning. However, to adapt to flower image classification problem, we combine PCANet with saliency map. The combination can deal with inter-class similarity and inner-class difference in flower classification.

3. Framework

Our method is divided into two parts: flower region selection, flower feature learning.

In the first part, we combine saliency map with gray-scale map to produce plain saliency region image, then calculate its maximum connection area and crop the same area in the original flower image. The cropped area is the flower region which includes the floral part we need to identify in addition to other irrelevant background.

In the second part, we input flower region to train PCANet [6] for flower feature learning. Then PCANet is followed by a softmax layer rather than SVM to complete classification. The framework is shown in Fig. 2.

3.1 Flower Region Selection

In the field of flower image classification, the complex background makes it more difficult to classify the flowers. In order to eliminate the interference caused by the complex background， we propose a method to extract flower region in original flower image.

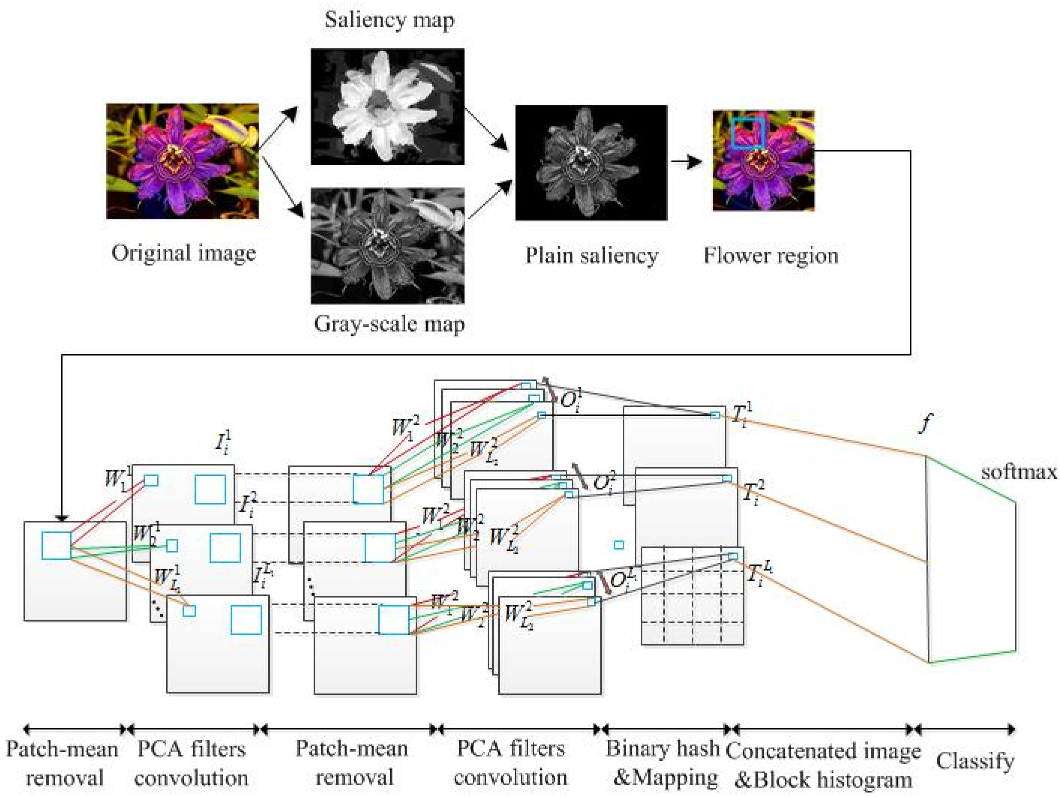


Fig. 2 Flower image classification framework base on saliency map and PCANet (best viewed in color).

Firstly, we choose salient region detection method based on regional contrast that is proposed by Ming-Ming Cheng [20] to generate saliency map. The salient region detection method considers both global contrast and spatial coherence, the method is simple and efficient, and can produce a saliency map of full resolution. Some saliency maps generated by this mehod are shown in Fig. 3.

The saliency map contains the significant region of the image that is the outline of the region. But it miss the details of the floral area. So we use the location information of the saliency region in saliency map to keep the flower region in the gray-scale map while others are set as background. The combination of the saliency map and the gray-scale map can produce clear flower region map. Then we compute the Maximum connected region of flower region map and get the location of the region. Lastly we crop the same area in the original image so as to complete the flower region selection.

However, the later feature learning framework need input of fixed size. So we need to resize the flower region to uniform size. The comparison of partial original flower image, gray-scale map, saliency map, flower region map, and cropped flower map is shown in Fig. 4.

3.2 Flower Feature Learning

We obtained the flower region in the original flower image by using the method of flower region selection proposed in the previous section. Therefore, this part completes the feature learning of the flower region.

Firstly, PCANet is selected as the feature learning framework to learn flowers' feature. Secondly, we change SVM classifier to softmax [21] for the transformation from feature to category probability. Because the flower image classification is multiple classification problem, softmax can handle multiple classification problems better with small computation and high efficiency. The flower feature learning framework is shown in Fig. 5.

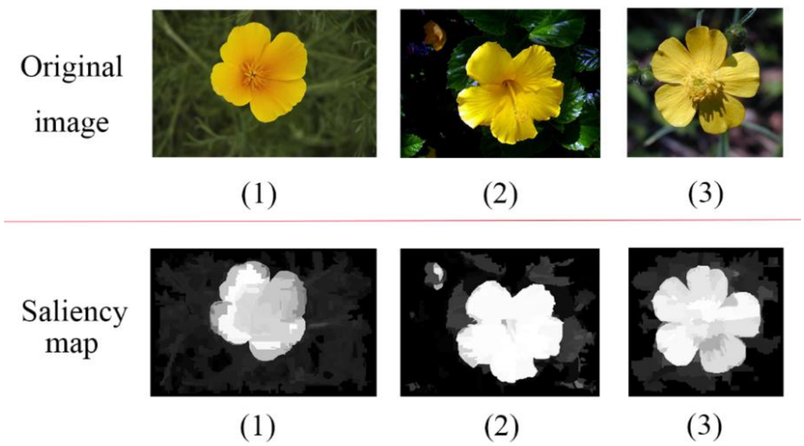


Fig. 3 Partial saliency map (best viewed in color).

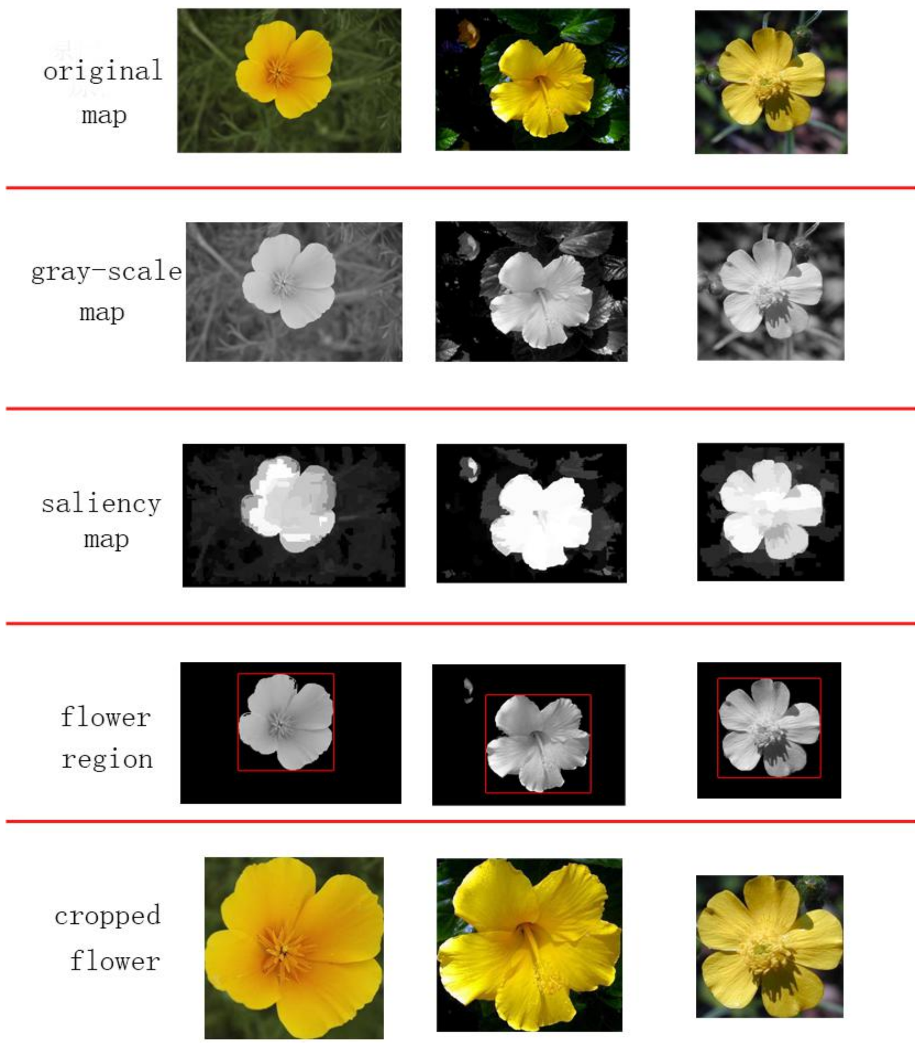


Fig. 4 Comparison of different map (best viewed in color).

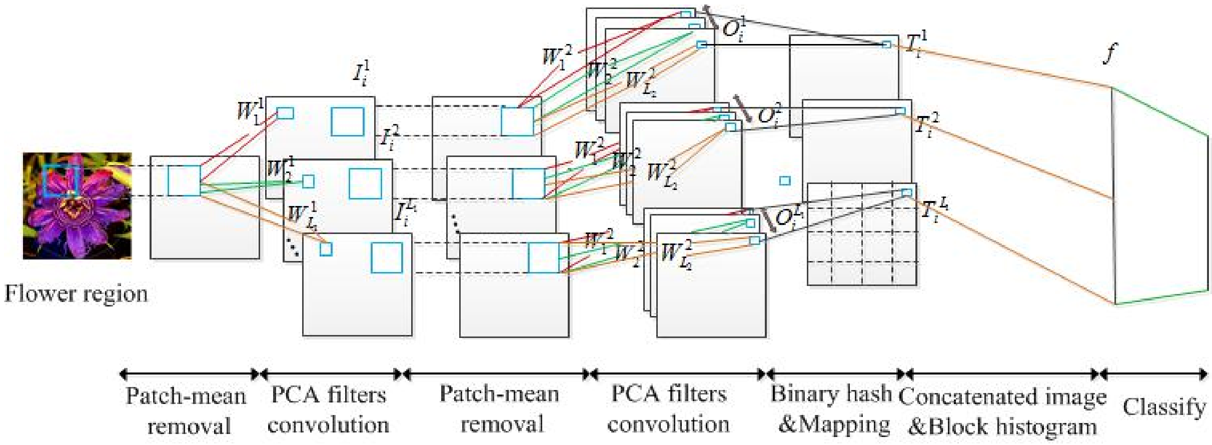


Fig. 5 Flower feature learning framework of PCANet with softmax (best viewed in color).

3.2.1 PCANet

PCANet is a simple framework for deep learning, which includes three parts: cascaded principal component analysis, binary hashing, block-wise histograms. In the framework, the main component analysis unit is mainly used to learn the multi-stage filtering parameters. Binary hash and block histogram is used mainly for indexing and pooling.

We suppose there are N different flower images $\{I_i\}_{i=1}^{N}$ for training the network, their size is $m\times n$, the patch size in all stages is $k_1\times k_2$.

(1) Input Processing

For flower image I_i, we take $k_1\times k_2$ overlapping patch in sliding way to get $m\times n$ patches, i.e., $x_{i,1},x_{i,2},...,x_{i,mn}\in R^{k_1k_2}$,where $x_{i,j}$ denotes the *j*th vectorized patch in I_i .Then we subtract patch mean from each patch and obtain $\overline{X}_i=\left\{\overline{x_{i,1}},\overline{x_{i,2}},...,\overline{x_{i,mn}}\right\}$.

By doing the same for all flower images and putting them together, we get a matrix X :

$$X=\left[\overline{X}_1,\overline{X}_2,...,\overline{X}_N\right] \tag{1}$$

(2) Cascaded PCA

We suppose the number of PCA filters in each stage is L_1, L_2.

1) First Stage of PCA

PCA algorithm is utilized to minimize the reconstruction error within a family of orthonormal filters, i.e.,

$$\min_{V\in R^{k_1k_2\times L_1}}\left\|X-VV^TX\right\|_F^2,\ \ s.t.\ V^TV=I_{L_1} \tag{2}$$

Where I_{L_1} is identity matrix of size $L_1\times L_1$. The solution of minimization is the L_1

Principal eigenvectors of XX^T. The PCA filters are therefore expressed as

$$W_l^1=mat_{k_1,k_2}\left(q_l\left(XX^T\right)\right)\in R^{k_1\times k_2},\ \ l=1,2,...,L_1 \tag{3}$$

Where $mat_{k_1,k_2}(V)$ is a function that maps $V\in R^{k_1k_2\times L_1}$ to a matrix $W\in R^{k_1\times k_2}$, and $q_l\left(XX^T\right)$ denotes the *l*th principal eigenvector of XX^T. The leading principal eigenvectors capture the main feature of all the training flower patches.

Getting L_1 PCA filters of size $k_1\times k_2$, we convolve each filter with the training flower image, i.e. the *l*th filter output of the first stage be

$$I_i^l=I_i*W_l^1,\ \ i=1,2,...,N\quad l=1,2,...,L_1 \tag{4}$$

Where $*$ denotes 2D convolution. I_i^l is the *l*th feature of flower image I_i in first stage .

2) Second Stage of PCA

Like input processing stage, we take $k_1\times k_2$ overlapping patch in sliding way to get patches of each filter output I_i^l in the first stage. We subtract patch mean from each patch and obtain

$$Y=\left[Y^1,Y^2,...,Y^{L_1}\right]\in R^{k_1k_2\times L_1Nmn} \tag{5}$$

So in second stage of PCA, the filter parameters of PCA will be obtained by

$$W_l^2 = mat_{k_1,k_2}\left(q_\ell\left(YY^T\right)\right) \in R^{k_1 \times k_2}, \ell = 1,2,...,L_2 \quad (6)$$

For each input in second stage, we will get L_2 outputs, and each output is derived from the convolution operation of I_i^l and W_l^2 .The calculation formula is as follows:

$$O_i^l = \left\{I_i^l * W_\ell^2\right\}_{\ell=1}^{L_2} \quad (7)$$

Therefore, in second stage, we obtain L_1L_2 outputs. If a deeper network structure is expected to be constructed, you can repeat the process to build more PCA stages.

3) Output stage: hashing and histogram

In this stage, it mainly reduce the dimension of learned feature like pooling layer in convolution neural network. We deal with the L_1L_2 outputs in last stage using Binary hash algorithm.

For L_2 outputs of each input image I_i^l , $i = 1,2,...,N \quad l = 1,2,...,L_1$ in second stage of PCA, we deal with them by binary hash algorithm, so obtain $\left\{H\left(I_i^l * W_\ell^2\right)\right\}_{\ell=1}^{L_2}$, where H() is the heaviside step function:

$$H(n) = \begin{cases} 0, & \text{n} < 0 \\ 1, & \text{n} \geq 0 \end{cases} \quad (8)$$

Then we convert those L_2 output into decimal number around each pixel.so we convert L_2 outputs of each input image I_i^l , $i = 1,2,...,N \quad l = 1,2,...,L_1$ into a simple image with a pixel value of an integer:

$$T_i^l = \sum_{\ell=1}^{L_2} 2^{\ell-1} H\left(I_i^l * W_\ell^2\right) \quad (9)$$

After getting the above L_1 images, we divide each image into blocks that are over-

Lapping and calculate the histogram of each block whose numbers of bins is set to 2^{L2}. Lastly, all the histograms of all blocks in each image are united to one vector $Bhist\left(T_i^l\right)$, so the flower image I_i is defined as a series of block histograms:

$$f_i = [Bhist(T_i^1), Bhist(T_i^2), ..., Bhist(T_i^{L_1})] \in R^{\left(2^{L_2}\right)L_1B} \quad (10)$$

The model parameters of the whole PCANet include the filter size $k_1 \times k_2$, the number of filters per PCA stage L_1, L_2, the blocksize in the output layer *HistBlockSize*, the overlapping rate of blocks *BlkOverLapRatio*. In addition, PCA requires parameters to satisfy $k_1k_2 \geq L_1, \ldots, k_1k_2 \geq L_s$.

3.2.2 Softmax

Softmax [21] is the regression model that is mainly used to solve multiple classification problems. In this paper, Softmax regression model is selected as the classification layer, mainly because it can handle the multi-classification problems better, and the calculation quantity is small and the efficiency is high.

Softmax regression model is Supervised. Suppose the training flower image is $\left\{\left(x^{(1)}, y^{(1)}\right), ..., \left(x^{(m)}, y^{(m)}\right)\right\}$, $y^{(i)} \in \{1,2,...,k\}$. $x^{(i)}$ i s the feature of flower while $y^{(i)}$ is the label of this flower.

For a test data, softmax use the hypothesis function to estimate the probability value $p\left(y = i \middle| x\right)$ for each class i . The function is going to print out a vector of dimensions (the sum of the vector elements is 1) to represent a probability value. The hypothesis function $h_\theta\left(x\right)$ is followed:

$$h_\theta\left(x^{(i)}\right) = \begin{bmatrix} p\left(y^{(i)} = 1 \middle| x^{(i)};\theta\right) \\ p\left(y^{(i)} = 2 \middle| x^{(i)};\theta\right) \\ \vdots \\ p\left(y^{(i)} = k \middle| x^{(i)};\theta\right) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix} \quad (11)$$

Where $\theta_1, \theta_2, \cdots, \theta_k \in R^{n+1}$ are training parameters.

For a train data, softmax minimize the cost function to make the probability value of true flower category max. The cost function $J\left(\theta\right)$ is followed:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} I\left\{y^{(i)} = j\right\}\log\frac{e^{\theta_l^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right] + \frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^{\ 2} \quad (12)$$

Where $I\{x\}$ is display function, when x is true, $I(x)=1$; otherwise $I(x)=0$. $\frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^{\ 2}$ is weight attenuation term that is to punish too many parameter items and make the cost function strict convex function for guaranting the unique solution.

To minimize $J(\theta)$, gradient descent method is used, each iteration is updated as follows:

$$\theta_j = \theta_j - \alpha\nabla_{\theta_j} J(\theta) \quad (j = 1,2,...,k)$$

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[x^{(i)}\left(I\left\{y^{(i)} = j\right\} - p\left(y^{(i)} = j \middle| x^{(i)};\theta\right)\right)\right] + \lambda\theta_j \quad (13)$$

Through continuous updating, the cost function is minimized to complete the training of the softmax regression model.

4. Experiments

4.1 Experimental Environment

We conduct experiments on Intel(R) Xeon(R) CPU L5640 @2.27GHz, memory is 80G. The programming environment is Matlab R2014a.

4.2 Experimental Dataset

We evaluate our method on Oxford 17 flowers [12]. The dataset with 80 images for each class include 1360 flowers images consisting of 17 categories. The images have large scale, pose and light variations and there are classes with large varations of images within the class and close similarity to other classes. And the image is divided into three sets: train data (680 images), validation data (340 images), test data (340 images). Each image has a label of its own ranging from 1 to 17. For better training networks, we also choose validation set as training data.

4.3 Experimental Procedure

4.3.1 Training Phase

First, we use the method in part 3.1 to process the training set for flower region selection. Because the size of the resulting flower area varies, so we resize all flower region image to 224×224.

Second, we will get the flowers area into PCANet and start training the network, mainly training the PCA filter parameters.

Finally, the obtained features and labels are input into the softmax layer for training.

However, the parameters of the network affect the experimental results, so the network parameters need to be determined by experiment. According to experience, in this paper we sets the output layer block size to *HistBlockSize* = 16×16, Blocks overlapping rate to BlkOverLapRatio = 0.5. As for parameter of filtering size $k_1 \times k_2$ and filtering numbers L_1, L_2 in two stage, they can be optimized by grid searching method [22], and we set $L_1 = L_2$, $k_1 = k_2$ to reduce the complexity. The optimal parameters are determined: $L_1 = L_2 = 20$, $k_1 = k_2 = 5$.

4.3.2 Testing Phase

Through training phase, we obtained the model that has been trained well. Therefore, in testing phase, we just need to extract the flower region of test data in turn, input the flower area to PCANet for learning features, and input the feature to the softmax layer for classification.

4.4 Experimental Results and Analysis

4.4.1 Impact of Flower Region Selection

In order to prove that the method of extraction of flower area used in this paper can improve the classification accuracy of flowers, the original map, the significant graph and the cropped flower region by our method in same size of 224×224 are respectively input into PCANet for training. The experimental results are shown in Table 1.

By analysis of Table 1, when the input is saliency map, the classification accuracy is lowest. The reason may be that saliency map is a gray image, does not include the color information and loses flower detail information of significant areas. However when the Our cropped flower region is regarded as the input of the PCANet, classification accuracy is significantly higher than the others, because cropped flower region in this paper contains the flower area of complete information and remove other irrelevant parts of original image that increase classification difficulty.

Some flower image's feature formed in mapping stage are shown in Fig. 6, we can see from the figure that PCANet can learned the features of flower images, like Contour and texture features.

4.4.2 Comprehensive Comparison

We makes experiments to contrast some methods on Oxford 17 flower dataset. Methods for comparison include some traditional classification method and two deep learning network without pretrained by Imagenet. The results are shown in Table 2.

You can see from the above table, our method in this paper is superior to traditional classification based on manual feature extraction. In addition, Alexnet and Googlenet without pretrained by Imagenet attain low accuracy, mainly because the training data of Oxford 17 flower dataset is not enough to get the optimal network parameters, so the deep network has been

Table 1 Comparison of classification effects with different input on Oxford 17 flower dataset.

Input	Feature learning framework	Accuracy (%,Top 1)
Original map		70.30
Saliency map	PCANet + Softmax	64.72
Our flower region		84.12

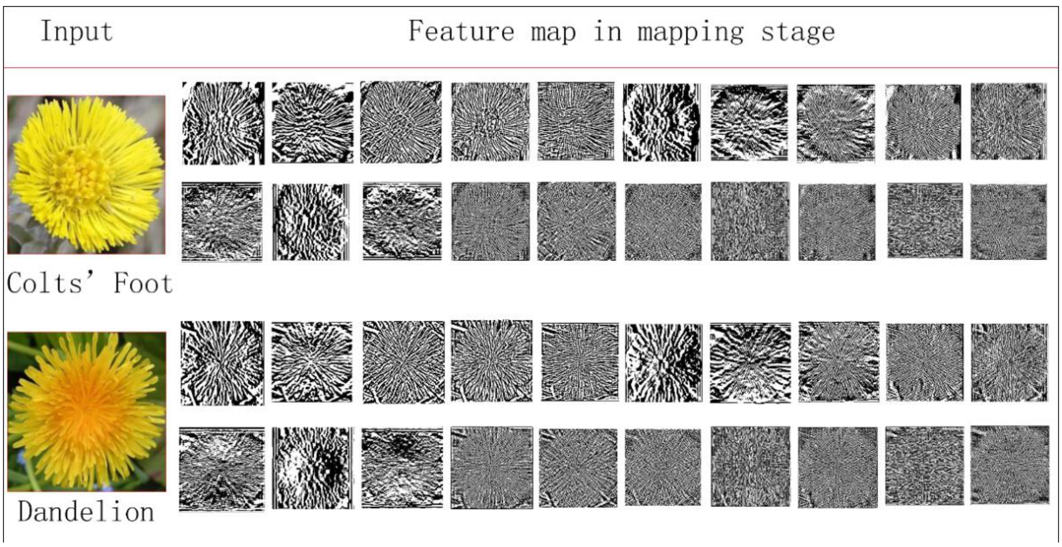


Fig. 6 Feature map formed in mapping stage (best viewed in color).

Table 2 Comparison of different method on Oxford 17 flower dataset.

Method	Classification Accuracy (%,Top 1)
Nilsback, Zisserman[06]	71.76
Varma and D-Ray[08]	82.55
Alexnet base (fc6)[2]	73.53
Googlenet base[15]	70.59
ours	84.12

overfitted. However, In terms of a small amount of training data, good result in our method owes the Simple network structure and flower region selection.

5. Conclusion

This paper presents a method of flower image classification based on Saliency Map and principal component analysis network (PCANet). The method is characterized by combining the saliency map with the gray-scale map to extract the flower region in the original image, and using PCANet to learn flowers' features, finally using the softmax regression model to implement classification. Experimental results show that the method is suitable for the flower image classification task. However, when the training data is sufficient, compared with method based on other deep learning network ,like alexnet pretrained by imagenet, Our method can't get good result because of the simple network structure in feature learning. Therefore, it is further research work in this paper whether the multi-scale principal component analysis network can be combined to improve the feature learning ability.

References

[1] Deng, J., Dong, W., and Socher, R. et al. 2009. "Imagenet: A Large-scale Hierarchical Image Database." *Computer Vision and Pattern Recognition (CVPR)*, IEEE Conference on Miami, USA: IEEE, 248-255.

[2] Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012 "Imagenet Classification with Deep Convolutional Neural Networks." In: *Advances in Neural Information Processing Systems*, Cambrige: MIT Press, 1097-1105.

[3] Zeiler, M. D., and Fergus, R. 2013. "Visualizing and Understanding Convolutional Networks." In: *European Conference on Computer Vision*. Springer, Cham, 818-833.

[4] Szegedy, C., Liu, W., and Jia, Y. et al. 2015. "Going Deeper with Convolutions." *Computer Vision and Pattern Recognition*, IEEE, 1-9.

[5] He, K., Zhang, X., and Ren, S. et al. 2016. "Deep Residual Learning for Image Recognition." *Computer Vision and Pattern Recognition*, IEEE, 770-778.

[6] Chan, T. H., Jia, K., and Gao, S. et al. 2015. "PCANet: A Simple Deep Learning Baseline for Image Classification?" *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society* 24 (12): 5017-5032.

[7] Lowe, D. G. 2004. "Distinctive Image Features From Scale-Invariant Keypoints." *International Journal of Computer Vision* 60 (2): 91-100.

[8] Dalal, N., and Triggs, B. 2005. "Histograms of Oriented Gradients for Human Detection." In: *San Diego: IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, 886-893.

[9] Burges, C. J. C. 1998. "A Tutorial on Support Vector Machines for Pattern Recognition." *Data Mining & Knowledge Discovery* 2 (2): 121-167.

[10] Bach, F. R., Lanckriet, G. R. G., and Jordan, M. I. 2004. "Multiple Kernel Learning, Conic Duality, and the SMO Algorithm." In: *NIPS*.

[11] Hastie, T., and Tibshirani, R. 2002. "Discriminant Adaptive Nearest Neighbor Classification." *IEEE Transactions on Pattern Analysis & Machine Intelligence* 18 (6) 607-616.

[12] Nilsback, M. E., and Zisserman, A. 2006. "A Visual Vocabulary for Flower Classification." In: *Proc. Comput. Vis. Pattern Recognit.*, 1447-1454.

[13] Varma, M., and Ray, D. Oct. 2007. "Learning the Discriminative Power-Invariance Trade-Off." In: *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 1-8.

[14] Zawbaa, H. M., Abbass, M., and Basha, S. H. et al. 2014. "An Automatic Flower Classification Approach Using Machine Learning Algorithms." In: *International Conference on Advances in Computing, Communications and Informatics*, IEEE, 895-901.

[15] Hinton, G. E., and Salakhutdinov, R. R. 2006. "Reducing the Dimensionality of Data with Neural Networks." *Science* 313 (5786): 504.

[16] Lécun, Y., Bottou, L., and Bengio, Y. et al. 1998. "Gradient-based Learning Applied to Document Recognition." *Proceedings of the IEEE* 86 (11): 2278-2324.

[17] Liu, Y., Tang, F., and Zhou, D. et al. 2016. "Flower Classification via Convolutional Neural Network." In: *International Conference on Functional-Structural Plant Growth Modeling, Simulation, Visualization and Applications*, IEEE, 110-116.

[18] Xia, X., Xu, C., and Nan, B. 2017. "Inception-v3 for Flower Classification." In: *International Conference on Image, Vision and Computing*, IEEE, 783-787.

[19] Chan, T. H., Jia, K., and Gao, S. et al. 2015. "PCANet: A Simple Deep Learning Baseline for Image Classification?" *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society* 24 (12): 5017-5032.

[20] Ming-Ming Cheng, Niloy J. Mitra, and Xiaolei Huang 2011. "Global Contrast Based Salient Region Detection."

IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 409-410.

[21] Liu, W., Wen, Y., and Yu, Z. et al. 2016. "Large-Margin Softmax Loss for Convolutional Neural Networks." *ICML*, 507-516.

[22] Zeng, R., Wu, J., and Shao, Z. et al. 2014. "Multilinear Principal Component Analysis Network for Tensor Object Classification." *Xiv Preprint ArXiv* 1411: 1171.