

Flower Classification via Convolutional Neural Network

Yuanyuan Liu

Institute of Control and Computer Engineering
North China Electric Power University
Beijing, China
yuanyuan_liu@ncepu.edu.cn

Dengwen Zhou

Institute of Control and Computer Engineering
North China Electric Power University
Beijing, China
zdw@ncepu.edu.cn

Fan Tang

LIAMA-NLPR, Institute of Automation
Chinese Academy of Sciences
Beijing, China

Yiping Meng

LIAMA-NLPR, Institute of Automation
Chinese Academy of Sciences
Beijing, China

Weiming Dong

LIAMA-NLPR, Institute of Automation
Chinese Academy of Sciences
Beijing, China
weiming.dong@ia.ac.cn

Abstract—In this paper, we address the problem of natural flower classification. It is a challenging task due to the non-rigid deformation, illumination changes, and inter-class similarity. We build a large dataset of flower images in the wide with 79 categories and propose a novel framework based on convolutional neural network (CNN) to solve this problem. Unlike other methods using hand-crafted visual features, our method utilizes convolutional neural network to automatically learn good features for flower classification. The neural network consists of five convolutional layers where small receptive fields are adopted, some of which are followed by max-pooling layers, and three fully-connected layers with a final 79-way softmax. Our approach achieves 76.54% classification accuracy on our challenging flower dataset. Moreover, test our algorithm on the Oxford 102 Flowers dataset. It outperforms the previous known methods and achieves 84.02% classification accuracy. Experimental results on a well-known dataset and our own dataset demonstrate that our method is quite effective in flower classification.

Keywords—flower classification; convolutional neural network

I. INTRODUCTION

There are about 369,000 named species of flowering plants in the world. Generally, experienced plant taxonomy experts can identify plants according to their flowers. However, it is difficult to distinguish these flowers for most people. To know the names or characteristics of the flowers, we usually consult with specialists, query with flower guidebooks or browse relevant web pages through keywords searching. An effective way to identify flower name can be done by classifying flower images, especially with the widely use of digital cameras, mobile phone, etc.

While the flower classification is appealing in its usefulness and meaningfulness, several restrictions have limited its realization. Unlike other obvious objects classification in which we need to distinguish obvious categories such as car and desk from each other, flower classification is a more difficult task because of inter-class similarity and a large intra-class variation. It is safe to say there are not any two same flowers in the world. What is more, it is tough to distinguish the difference of some kinds that are similar in appearance even for people. Figure. 1(a) shows examples of inter-class similarity, in which there are three different kinds of flowers but similar in appearance. Due to the non-rigid deformation of flowers, it is easy to be influenced by force during the process of blossom. In addition, images of flowers are often taken in a real environment where the illumination condition varies with the weather and time. Also, there is lot more variation in viewpoint, occlusions, scale of flower images. Figure. 1(b) shows intra-class variability because of illumination changes and view difference. All these problems lead to a confusion across classes and make the task of flower classification more challenging. In addition, the background also makes the flower classification task difficult. In this paper we address these limitations, providing techniques that are practical for flower classification problem.

In spite of the above limitation, we still focus on flower classification due to its significance in application of computer vision as well as botanical research. The key contributions of this paper are:

- We build a large-scale flower dataset. Previous works often focus on fastidiously clean dataset with few species. Even in the well-known Oxford 102 Flowers dataset, images are also well collected. However, classification task may be essential



(a) Inter-class similarity (Osteospermum, Pericallis, and Gazaniarigens)



(b) Intra-class variability(view difference)

Figure. 1. Diversity of flowers

just to deal more general situations. For this purpose, we build a more challenging dataset by collecting 63,442 flower images from the Internet using 79 species as key words and manually filtering 10,667 unrelated images. Apart from the dramatically larger scale of this dataset, the images exhibit considerable intra-class variability, inter-class similarities, scale variations, etc. Images from the dataset are shown in Figure. 2.

- We propose a new region select method for flower classification. In order to get much better performance of flower classification, we adopt traditional method that preprocesses original flower image by combining its saliency map and luminance map. Saliency map is a successful and biologically plausible technique for modeling visual attention, which indicates interesting regions in an image based on the spatial organization of the features. And as far as luminance map to be concerned, there is a priori knowledge that flowers have high brightness.

- We utilize an end-to-end convolutional neural network model for feature extraction instead of hand-crafting. The flower classification belongs to the fine-grained category of image classification, and the main challenge of fine-grained classification is the significantly larger differences between species. CNNs are able to automatically learn multiple stages of invariant feature for the specific task. The experimental results demonstrate our deep learning algorithm outperforms those traditional methods.

The rest is organized as follows: in Section II, we briefly review the related works of flower classification; in Section III, we describe the pipeline of our method including the used framework of convolutional neural networks; in Section IV, we make experiments to demonstrate our method of performance and compare the results. Finally, Section V draws the conclusion and discusses future works.

II. RELATED WORKS

A. Flower Image Classification

Research on flower classification system is an important topic in the botanical field. Now the quantity of flowers found comes up to several hundred thousand species, which is one of

the most prosperous creatures in the world. Plant classification is the basic of botanical studies. Since the 18th century a level hierarchical plant classification system is proposed by Carl Linnaeus, and so far it has been widely used all over the world. Initially, the classification method only identifies 8,000 kind of plant, but now can recognize 369,000 species. Despite the relatively systematic classification system, the plant taxonomy experts still must take a large amount of time and energy to classify plants. In the traditional way of flower classification, botanists or taxonomists first observe the life habits of flower; and then study overall characteristics and morphological structure characteristics; eventually compare with recorded plant specimens and confirm the flower species. The flower classification method can be achieved successfully under the guidance of scientific research personnel who have rich professional knowledge and experience.

Based on the situation above, the task of flower classification requires expert and domain-specific knowledge, which very few people generally have. Therefore, developing automatic classification systems for such tasks is of much benefit to non-experts. In recent years, the classification task of flower images in many category datasets has rapidly improved. We can find a couple of works carried out in flower classification. Here we review the previous work of it.

Nilsback and Zisserman [12] designed a flower classification system by extracting visual vocabularies which represent color, shape and texture features of flower images. The entire features then combined using a multiple kernel framework with a Support Vector Machine (SVM) classifier. In order to study the effect of classification accuracy on a large



Figure. 2. Example images from each class of our challenging 79-scale flower dataset

data set, Nilasback and Zisserman [13] considered a dataset of 103 classes each containing 40 to 250 samples. The low level features such as color, histogram of gradient orientations and Scale Invariant Feature transform (SIFT) features are used. Such combinations of features can improve classification performance on a large dataset. They have achieved an accuracy of 72.8% using SVM classifier using multiple kernels. Guru and his colleagues introduced an automatic classification model for flowers using K-Nearest Neighbor (KNN) classifier [17]. In [14], SIFT-like feature descriptors and feature context method were used in coding local and spatial information, then LibLinear SVM classifier is employed for classification. Kanan and Cottrell [15] used a model that combined sequential visual attention using fixations with sparse coding, whose biologically-inspired filters are acquired using unsupervised learning applied to natural images patches. Angelova and Zhu [16] proposed a segmentation approach, followed by the extraction of Histogram of Oriented Gradient (HOG) features at 4 different levels, encoded using Locality-constrained Linear Coding (LLC). Xie et al. [30] proposed to unify image classification and retrieval algorithms into ONE (Online Nearest-Neighbor Estimation), and Yoo et al. [31] proposed the multi-scale pyramid pooling for better use of neural activations from a pre-trained CNN, both of which achieved substantially higher image classification accuracy on Oxford 102 Flowers dataset.

B. Flower Detection

The main challenge of fine-grained classification includes non-rigid deformation, illumination changes, and inter-class similarity. But beyond that, complex background also makes the flower classification task difficult. In general, images are taken in natural settings which are rich and challenging, where the background features may become prominent. Therefore, flower detection is particularly beneficial, especially when the flower takes a small area of the image, is not in the center, or when the background is shared among different classes. We propose a new region-based detection method for flower classification based on combination of saliency map and luminance map, which are good indicators of the presence of the flower and can point to the possible location of the flower. This is quite beneficial for the final classification, as shown in our experiments.

Effective methods for object detection represent an important area of research in computer vision since many applications require the determination of the locations of objects in images. Recent works have proposed object detection for the purposes of better classification. And here we mention just a few representative object detection methods. The Histogram of oriented gradients (HOG) is a feature descriptor for the purpose of object detection [24], which counts occurrences of gradient orientation in localized portions of an image. The current state of the art for object detection is the Regions with Convolutional Neural Networks (R-CNN) method by Girshick et al. [25]. R-CNN first uses low-level cues such as color and texture to generate object location proposals in a category-agnostic fashion, and then utilizes CNN classifiers to identify object categories at those locations.

Based on the highly influential biologically inspired early

representation model introduced by Koch and Ullman [22], Itti et al. [23] define image saliency using central-surrounded differences across multi-scale image features. Saliency region detection is a topic of large practical importance and many recent works have enhanced computer vision and computer graphics applications. These methods utilize low-level processing to determine the contrast of image regions to their surroundings, and use feature attributes such as intensity, color, and edges [21].

C. Convolutional Neural Network

More recently, deep learning has become a hot topic in image processing and pattern recognition area. Convolutional neural networks (ConvNets) are a classic network structure of deep learning, which are multilayer feed-forward neural networks whose structures are biologically-inspired. Unlike other vision methods using hand-crafted features, ConvNets are able to automatically learn multiple stages of invariant feature for the specific task. Therefore, ConvNets have been substantially improving upon the state of the art in image classification and other recognition tasks [3, 4, 5, 10, 20, 29]. In the last few years, our object classification and detection capabilities have dramatically improved due to advances in deep learning and convolutional networks.

Since their introduction in the early 1990s, convolutional neural networks have consistently been competitive with other techniques for image classification and recognition. LeCun et al. [1] proposes the character recognition system LeNet-5 based on convolutional neural network, which used for bank handwritten numerals recognition. Similar to LeNet-5, convolutional neural networks have typically had a standard structure that consists of two parts: stacked convolutional layers (optionally followed by contrast normalization and max-pooling) and following one or more fully-connected layers. Variant of this basic design are prevalent in the image classification literature and have achieved the best results on MNIST [1], CIFAR [8] and most notably on the ImageNet classification challenge. The Imagenet Large Scale Visual Recognition Challenge (ILSVRC) [6] is an evaluation standard for what the current state of the art in image classification and recognition is. It is large dataset of 1.2 million images with 1,000 classes that are a subset of the Imagenet dataset [2], which is a dataset over 15 million labeled high-resolution images belonging to roughly 22,000 categories. For larger datasets such as ImageNet, the recent trend has been to increase the number of layers and layer size, while using dropout [9] to address the problem of overfitting. Krizhevsky et al. [3] proposes a large, deep convolutional neural network algorithm, and has achieved top-1 and top-5 error rates of 37.5% and 17.0% on ILSVRC2012, which is considerably better than the previous state-of-the-art. The convolutional neural network based system was made up of an ensemble of deep, eight layers network. It also incorporated important features such as pooling and normalizing layers, and dropout to avoid overfitting [19]. Sermanet et al. [10] shows that training a convolutional network to simultaneously classify, locate and detect objects in images can boost the classification accuracy and the detection and location accuracy of all tasks. It proposes a new integrated approach to object detection, recognition, and localization with a single ConvNet, which is the winner of the

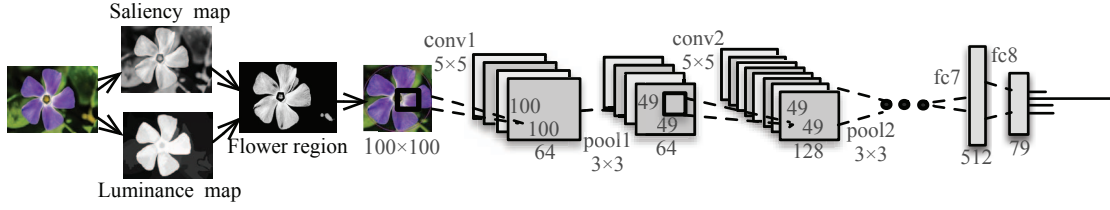


Figure 3. Pipeline of our framework

localization with a single ConvNet, which is the winner of the localization task of the ILSVRC2013 and obtained very competitive results for the detection and classification tasks. Szegedy et al. [11] proposes a deep convolutional neural network architecture codenamed Inception that achieves the new state of the art for classification and detection on ILSVRC2014. Szegedy et al. [7] demonstrates that the representation depth is beneficial for the classification, and that state-of-the-art performance on the ImageNet challenge dataset can be achieved using a conventional ConvNet architecture with substantially increased depth.

III. FRAMEWORK & IMPLEMENTATION

We first provide the pipeline of our framework, with the implementation details given in the remainder of this section. For each image presented to the system, preprocessing is applied to transform the image into the template used in database. Then the system extracts visual feature by convolutional neural network. Figure 3 shows the pipeline of our framework.

A. Flower Region Selection

Our challenging dataset consists of variable-resolution images, while our system requires a constant input dimensionality. Therefore, we down-sampled the images to a fixed resolution of 100. In order to get much better performance on flower classification, we first crop out a maximum square patch from the resulting image generated by combining saliency map and luminance map from original image, and then resize it to 100×100 as the input of the convolutional neural network. The final pre-processing we do is subtracting the mean RGB value, computed on the training set from each pixel.

The priori knowledge, flowers have high brightness, is one of the reason why the combination of saliency map and luminance map can enhance performance on flower classification. We convert each pixel from the RGB color space to YUV color space to obtain the luminance and the Y component determines the brightness of the color. We have also tried to choose other color space, but the performance does not further improve. Let $I_R(x, y)$, $I_G(x, y)$ and $I_B(x, y)$ denote respectively the R , G , and B color values of a pixel locating at (x, y) . Luminance $L(x, y)$ in YUV color space is given by the expression (see Figure 4(b))

$$L(x, y) = 0.3 \times I_R(x, y) + 0.6 \times I_G(x, y) + 0.1 \times I_B(x, y) \quad (1)$$

There are many computational models designed to produce saliency maps. Typically these algorithms produce maps that assign high saliency to regions with rare features or features

that differ from their surroundings. We use a regional contrast based salient object detection algorithm to compute bottom-up saliency map (see Figure 4(c)), which simultaneously evaluates global contrast difference and spatial weighted coherence scores [18]. The algorithm is simple, efficient, naturally multi-scale, and produces full-resolution, high-quality saliency maps.

B. Deep Architecture

Here we describe the overall architecture of our deep CNN. As depicted in Figure 5, the net contains eight layers with weights; the first five are convolutional and the remaining three are fully-connected. The output of the last fully-connected layer is fed to a 79-way softmax, which produces a distribution over the 79 class labels. The following is the structure of convolutional neural network used in this article.

The image is passed through a stack of convolutional layers. The first convolutional layer filters the $100 \times 100 \times 3$ input image with 64 kernels of size $5 \times 5 \times 3$. The second, third convolutional layers have 128, 256 kernels of size $5 \times 5 \times 64$, $3 \times 3 \times 128$ connected to the (normalized, pooled) outputs of the previous convolutional layer respectively. The fourth convolutional layer takes as input the pooled of the third convolutional layer and filters it with 512 kernels of size $3 \times 3 \times 256$, and the fifth convolutional layer has 512 kernels of size $3 \times 3 \times 512$. The convolution stride is fixed to 1 pixel (this is the distance between the receptive field centers of neighboring neurons in a kernel map); the padding is 2 pixels for 5×5 convolutional layers and 1 pixel for 3×3 convolutional layers. The stack of convolutional layers is followed by three fully-connected layers (the neurons in the fully-connected layers are connected to all neurons in the previous layer): the first two have 1024, 512 channels respectively, the last performs 79-way classification thus contains 79 channels (one for each class). Response-normalization layers follow the first and second convolutional layers. Spatial pooling is carried out by four max-pooling layers, which follow all convolutional layers except the fourth one and are performed over a 3×3 pixel window, with stride 2.

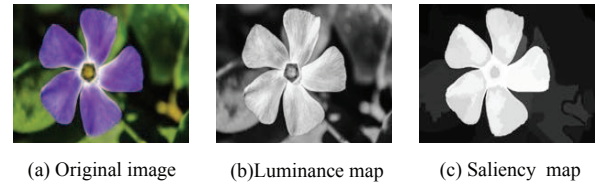


Figure 4. Original image, its corresponding luminance map and bottom-up saliency map, which indicates interesting features in an image.

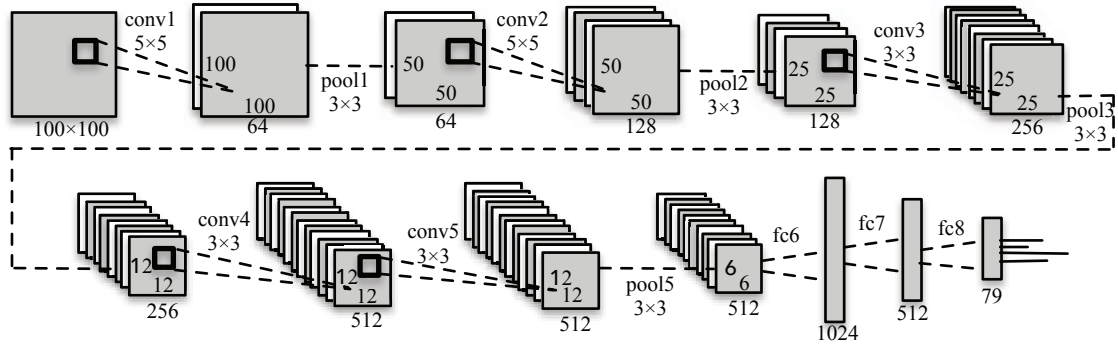


Figure. 5. The architecture of our deep CNN

The ReLU non-linearity is applied to the output of every convolutional and fully-connected layer.

Above is the overall architecture of our CNN, and next we specifically introduce the crucial methods used in the structure, which improve flower classification performance of the model in some extend.

1) Local Response Normalization

In general, the neuron output function of neural network model is saturating nonlinearity $f(x) = \tanh(x)$ or $f(x) = (1 + e^x)^{-1}$. Following Nair and Hinton [28], Krizhevsky et al. [3] proposed a non-saturating nonlinear activation function $f(x) = \max(0, x)$, named Rectified Linear Units (ReLU). First and foremost, ReLUs are much faster than these saturating nonlinearities in terms of training speed. Faster learning has an excellent influence on the performance of large models trained on large datasets. And furthermore, these saturating nonlinearities above change slowly near saturation zone, which easily leads to gradient diffusion in back-propagation. In addition, ReLUs have good performance because they do not require input normalization to prevent them from saturating. We adopt local normalization scheme to prevent saturation. We use $a_{x,y}^i$ to denote the activity of a neuron computed by applying kernel i at position (x, y) , and then apply the ReLU nonlinearity to this activity. The response normalization $b_{x,y}^i$ is given by the expression

$$b_{x,y}^i = a_{x,y}^i / \left(k + \left(\sum_{j=\max(0, i-n/2)}^{\min(N-1, i+n/2)} (a_{x,y}^j)^2 \right)^\beta \right) \quad (2)$$

The sum is processed at the same position in the field of convolution kernels n , and N is the total number of convolution kernels in the layer. The order of the kernel maps is arbitrary and determined before the training. The response normalization is a lateral inhibition. The constants k , n , α and β are hyper-parameters which are determined by the validation set; we used $k = 2$, $n = 5$, $\alpha = 10^{-4}$ and $\beta = 0.75$. We applied this normalization after applying the ReLU nonlinearity in certain layers.

2) Overlapping Pooling

Pooling layers in CNNs summarize the outputs of adjacent groups of neurons in the same convolution kernels, which progressively reduce the amount of parameters and

computation in the network, and hence to also control overfitting. Traditional pooling operation is processed without overlap. Namely we put the $z \times z$ as pooling unit, and if we set $s = z$, it represents the traditional process of pooling. In other case, if we set $s < z$, we obtain overlapping pooling. Among them, s expresses movement step length of pooling unit. This is we use in our CNNs, with $s = 2$ and $z = 3$.

3) Dropout

To avoid substantial overfitting in the fully-connected layers, we employ an important regularization method, called “dropout”. Dropout designs to suppress the neurons of one hidden layer with certain probability, which sets to zero the output of each hidden neuron. That is, the neurons that are suppressed in this way do not contribute to the forward pass and do not participate in back-propagation. At test time, we use all the neurons but multiply their outputs by certain probability above. Dropout with a rate of 0.5 is utilized in the first two fully-connected layers of Figure. 5.

C. Convolutional Activation Features

As shown in Figure. 5, the original image is adjusted to the size of 100×100 as input layer. The size of the first two layer feature map is 100×100 , 50×50 respectively, and the number of each layer feature map is 64, 128. The third layer feature map has the size of 25×25 , which contains 256 feature maps. The feature of the fourth and fifth layer is all 12×12 , and the number of each layer feature map is 512. The sixth and seventh layer are fully-connected layers and each layer is made up of 1024, 512 neurons. The last layer is fully-connected layer as well as policy maker in this architecture of convolutional neural network and it is no reference in this paper, so we only consider the first seven layers.

IV. EXPERIMENTS

In this part, we show experimental results of our proposed algorithm on well-known Oxford 102 flowers and our challenging flower dataset. In each dataset we report the performance of our classification algorithm and previous known algorithms in the same settings.

A. Oxford 102 Flower Species Dataset

Oxford 102 flowers dataset is a well-known dataset for fine-grained recognition, which contains 102 species of flowers and a total of 8,189 images, each category containing between 40

and 200 images (See Figure. 6(a)).

For fair consideration, we compare the performance of our algorithm on Oxford 102 flowers dataset with a lot of previous methods, including some segmentation-based. Some of the segmentation methods are applied to be very specific to the appearance of flowers, with the assumption that a single flower is in the center of the image and takes most of the image, while our approach can be designed to more general flower images.

The comparison results are shown in Table. 1 where the results of other methods are available from published papers conveniently. The comparison results show that our approach achieves 84.02% classification accuracy on Oxford 102 flowers dataset, which outperforms the previous known methods in the literature. The underlying reason is that the convolutional neural network we use is able to learn discriminative and reliable features for flower type classification.

B. Large-scale 79 Flower Species Dataset

Considering that classification task may be essential just to deal with more general situations, we collected a complex and challenging flower dataset to test our method, which contains 79 different species of flowers and 52,775 images. Our dataset is in nature very different from the collaboratively revised and filtered data used in Oxford 102 flowers dataset. Figure. 6(b) shows the example images of dataset. We can see that these images contain changes in illumination condition, scale variation and viewpoint, etc. In all, there are roughly 47,500 training images and the rest are testing images.

We test our method on the challenging flower dataset and report its performance. Our approach achieves 76.54% accuracy. As we expected, our model can precisely classify flower category in some challenging situations, such as different lighting conditions, inter-class similarities and viewpoint changes. The primary reason is that our convolutional neural network is able to learn discriminative features for flower classification. We further investigate the contribution of network depth. And we compare the performance of our algorithm on the challenging dataset with a simple, straight-forward CNN method. Their classification accuracies can be found in Table. 2.

V. CONCLUSION

In this paper, we propose a flower classification approach to automatically identify flower category, by using convolutional neural network. We extract the features using the convolutional neural network, which are effective for a variety of object classification tasks. Experiments conducted on two



Figure. 6. Images in Oxford 102 and our dataset

Table 1. Classification accuracy on Oxford 102

Method	Accuracy (in %)
Nilsback and Zisserman [13]	72.8
C. Kanan and G. Cottrell [15]	75.2
Nilsback and Zisserman [26]	76.3
Chai, BicosMT method [27]	80.0
Angelova A and Zhu S [16]	80.66
Ours	84.02

Table 2. Classification accuracy on our dataset

Method	Accuracy (in %)
Simple CNN	70.12
Ours	76.54

different flower image databases consisting of 79 species and 102 species have shown that our proposed approach achieves higher classification accuracy than the previous methods. In future, we will apply the support vector machine (SVM) classifier to perform multi-label classification, which improves the accuracy on the current training dataset, even much larger and more challenging flower dataset, and better applies to real life.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their constructive comments. This work was supported by National Natural Science Foundation of China under nos. 61372184, 61331018 and 61672520, and by Beijing Natural Science Foundation under No. 4162056. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X GPU used for this research.

REFERENCES

- [1] Y. LeCun, L. Bottou, and et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278-2324, 1998.
- [2] J. Deng, W. Dong, and et al. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248-255, 2009.
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106-1114, 2012.
- [4] D.C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, pages 3642-3649, 2012.
- [5] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *ECCV*, volume 8689 of *Lecture Notes in Computer Science*, pages 818-833. Springer, 2014.
- [6] O. Russakovsky, J. Deng, J. Krause, A. Berg, F. Li. ILSVRC-2013, 2013, URL <http://www.image-net.org/challenges/LSVRC/2013/>.
- [7] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.
- [8] A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto, 2009.

- [9] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors, CoRR, abs/1207.0580, 2012.
- [10] P. Sermanet, D. Eigen, and et al. OverFeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229, 2013.
- [11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
- [12] M.E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In CVPR, volume 2, pages 1447-1454, New York, 2006.
- [13] M.E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In Proc. Indian Conference on Computer Vision, Graphics and Image Processing, pp. 1-8, 2008.
- [14] Wenjing Qi, Xue Liu, and Jing zhao. Flower classification based on local and spatial visual cues. In CSAE, pages 670-674, 2012.
- [15] C. Kanan, G. Cottrell. Robust classification of objects, faces, and flowers using natural image statistics. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pages 2472-2479, 2010.
- [16] Angelova A, Zhu S. Efficient object detection and segmentation for fine-grained recognition. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pages 811-818, 2013.
- [17] D.S.Guru, Y.H.Sharath, and S. Manjunath. " Texture features and KNN in classification of flower images", IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition (PTTIPPR), Vol. 1, pp. 21-29, 2010.
- [18] M-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 3, pp. 569-582, Mar. 2015.
- [19] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012.
- [20] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In International Conference on Computer Vision, pages 2146-2153. IEEE, 2009.
- [21] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk. Frequency-tuned salient region detection. In Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 1597-1604, 2009.
- [22] C. Korch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. Human Neurobiol., vol. 4, pp. 219-227, 1985.
- [23] L. Itti, C. Korch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 11, pp. 1254-1259, Nov. 1998.
- [24] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, volume 2, pages 886-893, 2005.
- [25] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In In CVPR, pages 580-587, 2014.
- [26] M.-E. Nilsback and A. Zisserman. An automatic visual flora – segmentation and classification of flower images. Dphil Thesis, University of Oxford, UK, 2009.
- [27] Y. Chai, V. Lempitsky, and A. Zisserman. Bicos: A bi-level co-segmentation method for classification. In ICCV, pages 2579-2586, 2011.
- [28] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In Proc. 27th International Conference on Machine Learning, 2010.
- [29] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In Proc. IEEE Int. Conf. Comput. Vis. Dec. 2015, pp. 1143-1151.
- [30] L. Xie, Q. Tian, R. Hong, and B. Zhang. Image classification and retrieval are one. In International Conference on Multimedia Retrieval (ICMR), pages 3-10, 2015.
- [31] D. Yoo, S. Park, J.-Y. Lee, and I. S. Kweon. Multi-scale pyramid pooling for deep convolutional representation. In CVPR, pp. 71-80, 2015.