

Pixel Saliency based Encoding for Fine-Grained Image Classification

Chao Yin, Lei Zhang*, Ji Liu

College of Communication Engineering, Chongqing University,
No. 174 Shazheng street, Shapingba district, Chongqing 400044, China
{chaoyin, leizhang, jiliu}@cqu.edu.cn

Abstract. Fine-grained image classification concerns categorization at subordinate levels, where the distinction between inter-class objects is very subtle and highly local. Recently, Convolutional Neural Networks (CNNs) have almost yielded the best results on the basic image classification tasks. In CNN, the direct pooling operation is always used to resize the last convolutional feature maps from $n \times n \times c$ to $1 \times 1 \times c$ for feature representation. However, such pooling operation may lead to extreme saliency compression of feature map, especially in fine-grained image classification. In this paper, to more deeply explore the representation ability of the feature map, we propose a *Pixel Saliency based Encoding* method, which is called PS-CNN. First, in our PS-CNN, the saliency matrix is obtained by evaluating the saliency of each pixel in the feature map. Then, we segment the original feature maps into multiple ones with multiple generated binary masks via thresholding on the obtained saliency matrix, and subsequently squeeze those masked feature maps into the encoded ones. Finally, a fine-grained feature representation is generated by concatenating the original feature maps with the encoded ones. Experimental results show that our simple yet powerful PS-CNN outperforms state-of-the-art classification approaches. Specially, we can achieve 89.1% classification accuracy on the Aircraft, 92.3% on the Stanford Car, and 81.9% on the NABirds.

Keywords: Pixel saliency, Feature encoding, Fine-grained, Image classification

1 Introduction

Fine-grained image classification aims to recognize similar sub-categories in the same basic-level category [1–3]. More specifically, it refers to the task of assigning plenty of similar input images with specific labels from a fixed set of categories by using computer vision algorithms. Till now, for such categorization in computer vision area, Convolutional Neural Networks (CNNs) have played a vital role. The impressive representation ability of CNNs, e.g., VGG [4], GoogleNet [5], ResNet [1], and DenseNet [2], is also demonstrated in object detection [6], face recognition [7], and many other vision tasks. By using the CNN models pre-trained on the ImageNet, many image classification problems are well addressed

and their classification accuracies almost approach to their extreme performance. However, fine-grained image classification, a sub-category of basic-level category, is still a challenging task in computer vision area due to high intra-class variances caused by deformation, view angle, illumination, and occlusion of images and low inter-class variances which are tiny differences occurred only in some local regions between inter-class object and these can only be recognized by certain experts. Moreover, we are faced with several under-solved problems in fine-grained image classification. One problem is that there are limited fine-grained images with labels due to the high cost of labeling and cleaning when collecting data [8]. Another is the difficulty of acquiring better annotations and bounding boxes which are helpful in the process of classifying fine-grained images.

For image classification, CNNs are always exceptionally powerful models. First, apart from some simple image pre-processing, CNN always uses the raw images with a pre-defined size as its input. Then, CNN progressively learns the low-level (detail), middle-level, and high-level (abstract) features from bottom, intermediate to top convolutional layers without any hand-craft feature extraction policy like SIFT and HOG [9] [10]. Finally, the discriminating feature maps with a pre-defined size from top-level layers are obtained. At the same time, if the size of input images, in some case, increases, the size of output convolutional layers also increases. In general way, we can directly perform an average or max pooling to produce the last feature representation which then will be sent to classifier of network. However, such coarse pooling operation will lead to extreme saliency compression of feature map especially for fine-grained image classification that concentrates on more fine-grained structure information. In fact, saliency compression is a bottleneck for information flow of CNN.

To solve aforementioned extreme saliency compression problem when classifying fine-grained images, we propose a *Pixel Saliency based Encoding* method for CNN. The motivations for our method are presented as follows.

- (1) Considering the characteristic, tiny differences only occurred in some local regions, of fine-grained images, a simple solution for fine-grained image classification is magnifying images on both training and testing phases to ‘look’ into more details [11]. The magnified input images will result in an increasing size of the last convolutional feature map. If still using the straightforward coarse Avg/Max pooling operation as usual, it will lose a lot of detailed structural information which will be help for classification. Therefore, the method of re-encoding the feature map should be adopted to explore the cryptic information for last convolutional feature map.
- (2) In image segmentation area [6, 12], it is expected that different pixels in a feature map with different range of saliency are explicitly segmented so that the interest of object is revealed and the background is hidden in a feature map, which is also ours goal here for fine-grained image classification. We argue this separation is necessary for recognizing the regions of interest and will be helpful for feature learning of the total CNN.
- (3) After all of those ‘parts’ with different saliency are segmented and then squeezed in the next layer, to involve the global information of input image,

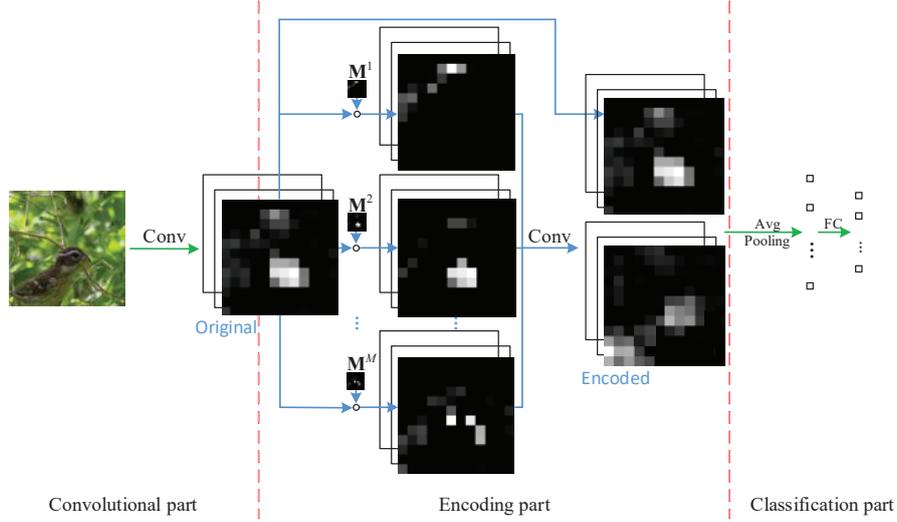


Fig. 1. Overview of the proposed encoding method. A input image passes through convolutional part of CNN to generate the (original) feature maps. First, we average those feature maps to get one saliency matrix. Then, M masks are generated based on this matrix and used to mask the original feature maps by using the *Hadamard product*. As a result, there will generate total M streamings. After that, we squeeze those M streamings into one feature representation map by using the 1×1 convolutional operation. At last, the original and encoded feature maps are concatenated to form the feature map for representation.

the original feature map reflecting the overall characteristics should also be concatenated as the final feature representation.

Our PS-CNN is a saliency based encoding method within well-known Inception-V3 framework. By encoding, PS-CNN explores the better representation ability of the last convolutional feature maps rather than the average pooling used as usual. After encoding, more details of local regions are collected separately. Then, to involve global information of object, we concatenate encoded feature maps with the original one so that new feature maps carried with global image representations are generated. Our PS-CNN is a simple yet effective method. The details of our PS-CNN are shown as in Fig. 1. As can be seen from the encoded feature map in encoding part, the region of interest is successfully picked out from the original feature map extracted by convolutional part of CNN.

The main contributions of the paper are listed in the following.

- (1) We firstly argue that pixels with different range of saliency in feature maps should be explicitly segmented. Then a simple saliency matrix calculation method is proposed to evaluate the saliency of each pixel in feature map.
- (2) Multiple binary masks are calculated based on the selected thresholds along with the calculated saliency matrix to more explicitly segment the original

feature maps. Then the information-richer feature representation is developed by concatenating the encoded feature maps with the original ones.

- (3) Experimental results on accuracy with different number of masks are illustrated, showing that our encoding method is efficient. In addition, the pixel saliency based encoding method proposed in our paper, can be embedded into any CNNs.

The rest of this paper is organized as follows. Section 2 describes one CNN, i.e., Inception-V3. Then the existing part/object localization and feature encoding methods are summarized. In Section 3, a *Pixel Saliency based Encoding* method for CNN (PS-CNN) is proposed. In section 4, we present experimental results to illustrate the classification accuracy improvement of the proposed PS-CNN and we also discuss the influence on classification accuracy with varying number of binary masks. Finally, Section 5 concludes this paper.

2 Related Work

Convolutional Neural Network defines an exceptionally powerful feature learning model. To better advance the image classification accuracy, one direct solution is to increase the depth and width of network. However, basic CNNs are still limited in some specific classification tasks, e.g., fine-grained image classification. The predominant approaches in fine-grained image classification domain can be categorized into two groups. One learns the critical parts of the objects, and the other one directly improves the basic CNN from the view of feature encoding.

2.1 Base Network: Inception-V3

Inception-V3 [13] is a CNN with a high performance in computer vision area and bears a relatively modest computation burden compared to those simpler and more monolithic architectures like VGG [4]. As reported in paper [13], Inception-V3 have achieved 21.2% top-1 and 5.6% top-5 error rates for single crop evaluation on the ILSVR 2012 classification task, which has set a new state of the art. Besides, it also has achieved relatively modest (2.5x) improvement in computational cost compared to the firstly proposed version, i.e., GoogleNet (Inception-V1) network described in [14].

2.2 Part and Object Localization

A common approach for fine-grained image classification is to localize various parts of the object and then model the appearance of part conditioned on their detected locations [8, 15]. The method proposed in [8] can generate parts which can be detected in novel images and learn which of those parts are useful for recognition. This method is a big step towards the goal of training fine-grained classifiers without part annotations. Recently, many attentions [16, 17] have been paid to the part and object localization method. The OPAM proposed in [17] is

aimed for weakly supervised fine-grained image classification, which jointly integrates two level attention models: object-level one localizes objects of images and part-level one selects discriminative parts of objects. The paper proposed a novel part learning approach which is named Multi-Attention Convolutional Neural Network (MA-CNN) [16]. It is interesting that two functional parts, i.e., part generation and feature learning, can reinforce each other. The core of MA-CNN is that one channel grouping sub-network is firstly taken as input feature channels from convolutional layers and then generates multiple parts by clustering, weighting, and pooling from spatially-correlated channels.

2.3 Feature Encoding

The other kind of fine-grained image classification approach is to use a robust image representation from the view of feature encoding. Traditional images representation methods always include hand-craft descriptors like VLAD Fisher vector with SIFT features. Recently, rather than using SIFT extractor, the features extracted from convolutional layers in a deep network pre-trained on ImageNet show better representation ability. Those CNN models have achieved state-of-the-art results on a number of basic-level recognition tasks.

There are many methods proposed to encode the feature maps extracted from the the last convolutional layers. The representative methods include Bilinear Convolutional Neural Networks (B-CNN) [11] and Second-order CNN [10]. In B-CNN, the output feature maps extracted by the convolutional part are combined at each location, which refers to being encoded by using the matrix outer product. The representation ability after encoding is highly effective in various fine-grained image classification. The Second-order CNN [10] makes an adequate exploration of feature distributions and presents a Matrix Power Normalized Covariance (MPN-COV) method that performs covariance pooling for the last convolutional features rather than the common pooling operation used in general (first-order) CNN. The Second-order CNN has achieved better performance than B-CNN, but needs a fully re-training on ImageNet ILSVRC2012 dataset.

3 The Proposed Approach

In this Section, we provide the description of our *Pixel Saliency based Encoding* method for CNN (PS-CNN). The details of our PS-CNN architecture and some mathematical presentation of the encoding method are presented as follows.

The convolutional part of Inception-V3 network (referring to [13]) is acted as feature extractor as in our PS-CNN. In general, given the input image \mathbf{x} and the feature extractor $\Phi(\cdot)$, the output feature maps, can be written as

$$\mathbf{F}^0 = \Phi(\mathbf{x}). \quad (1)$$

Here, all the feature maps extracted by Inception-V3 are defined as $\mathbf{F}^{n,0}$, $n = 1, 2, \dots, N$. Each feature map is with size of $s \times s$. As the default setting of

Inception-V3, the s is set to 8. It is worth noting that the s is set to 1 in VGG model. In traditional way, an average pooling will be performed upon \mathbf{F}^0 to generate one feature vector. However, in our PS-CNN, we manage to encode those output information-rich feature maps \mathbf{F}^0 .

3.1 Saliency Matrix Calculation

In order to evaluate the saliency of each pixel in the feature map with size $s \times s$, we perform an element-wise average operation across N feature maps, i.e.,

$$\mathbf{M}_{i,j}^0 = \frac{1}{N} \sum_{n=0}^N \mathbf{F}_{i,j}^{n,0}, \quad (2)$$

where $i, j = 1 \cdots s$. In this *saliency matrix* \mathbf{M}^0 , the value $\mathbf{M}_{i,j}^0$ reflects the saliency of the each pixel. We then use this saliency matrix \mathbf{M}^0 to generate several binary masks \mathbf{M}^m where $m = 1, 2 \cdots, M$,

$$\mathbf{M}_{i,j}^m = \begin{cases} 0, & t_m < \mathbf{M}_{i,j}^0 < t_{m+1} \\ 1, & \text{Otherwise.} \end{cases} \quad (3)$$

where t_m is threshold. The pair of (t_m, t_{m+1}) defines the range of saliency. If the saliency lays within the range of t_m and t_{m+1} , the corresponding pixels of feature map will be masked as zero. The other pixels of feature map will remain unchanged if saliency of those pixels is outside that range. Here the selection of value t_m is flexible. Notably, value t_m should be between the minimum and maximum values of \mathbf{M}^0 . In this paper, four binary masks are utilized, i.e., $m = 1, 2, 3, 4$. Besides, the thresholds t_m and t_{m+1} shown in Eq. 3 are chosen as

$$t_m = \min(\mathbf{M}^0) + \text{percent}_m \times (\max(\mathbf{M}^0) - \min(\mathbf{M}^0)), \quad (4)$$

where the $\min(\cdot)$ and $\max(\cdot)$ find the minimum and maximum values of \mathbf{M}^0 . The percent_m are chosen as $\text{percent}_1 = 0.1$, $\text{percent}_2 = 0.3$, $\text{percent}_3 = 0.5$, $\text{percent}_4 = 0.7$. When $m = 4$, the upper bound t_{m+1} in Eq. (3), i.e., $\text{percent}_5 = 1$.

3.2 Pixel Saliency based Encoding

After obtaining the multiple binary masks, i.e., \mathbf{M}^m for $m = 1 \cdots 4$, we encode the original feature maps $\mathbf{F}^{n,0}$ as follows,

$$\mathbf{F}^{n,m} = \mathbf{F}^{n,0} \circ \mathbf{M}^m, \quad (5)$$

where operation \circ is *Hadamard product*. Thus, for masks \mathbf{M}^m , the $\mathbf{F}^{n,m}$ are the encoded feature maps of the original $\mathbf{F}^{n,0}$. Each feature map in $\mathbf{F}^{n,m}$ is encoded with all the information Implicitly carried by \mathbf{M}^m of the original feature maps $\mathbf{F}^{n,0}$. In addition, N convolutional kernels [13], each of which is with size of 1×1 , are used to squeeze the total $M \times N$ feature maps to a much smaller one,

i.e., \mathbf{G} , that has only N feature maps. The feature maps encoding process and visualization are shown as in the encoding part of Fig. 1.

At last, to involve global information of image/object, the original feature maps are concatenated with the feature map \mathbf{G} of subsequent layer by channel, which forms the last feature representation as

$$\mathbf{H} = [\mathbf{F}^{n,0}; \mathbf{G}]. \quad (6)$$

The classification part as shown in Fig. 1 is the same as the original Inception-V3. Our encoding method is transplantable and simple enough so that it can be embedded into any other CNN framework.

Remarks: Considering the number of feature maps of new representation, i.e., \mathbf{H} in Eq. (6), is twice than the original one in Eq. (1) which is only with the number of N , we could reduce the size of representation by using $N/2$ convolutional kernel, each with the size of 1×1 .

4 Experiments

We use AutoBD [18], B-CNN [11], M-CNN [19], and Inception-V3 [13] as compared methods. The model of Inception-V3 is fine-tuned by ourselves. We extend this baseline CNN to include our proposed pixel saliency encoding method and the parameters of our PS-CNN are directly adopted from the Inception-V3 without any sophisticated adjustment.

4.1 Fine-grained Datasets

There are three datasets chosen in our experiments. The total number, total species, and default train/test split of Aircraft [20], Stanford Car [21], and NABirds [22] datasets are summarized as Table 1. All the image number of those three datasets are much smaller comparing to the basic image classification datasets, e.g., ImageNet, WebVision. The three datasets are also analyzed.

Table 1. Comparison about number, species, and train/test split of Aircraft, Car, and NABirds datasets.

	Aircraft	Car	NABirds
total number	10,000	16,185	48,562
total species	90	196	555
train	6,667	8,144	23,929
test	3,333	8,041	24,633

Aircraft [20] is a benchmark dataset for the fine-grained visual categorization of aircraft introduced in well-known FGComp 2013 challenge. It consists of 10,000

images of 100 aircraft variants. The airplanes tend to occupy a significantly large portion of the image and appear in relatively clear background. Airplanes also have a smaller representation in the ImageNet dataset on which the most CNN models are trained, compared to some other common objects.

Stanford Car [21] contains 16,185 images of 196 classes as part of the FGComp 2013 challenge as well. Categories are typically at the level of Year, Make, Model, e.g., “2012 Tesla Model S” or “2012 BMW M3 coupe”. It is special because cars are smaller and appear in a more cluttered background compared to Aircraft. Thus object and part localization may play a more significant role here.

NABirds [22] is a pretty large-scale dataset which consists of 48,562 birds images of North America. It has total 555 species. This dataset provides not only label of each bird image, but also additional valuable parts and bounding-box annotations. However, we do not use those information in both of our training and testing. It means when training our models, only the raw birds images and corresponding category labels are used.

4.2 Implementation Details

We fine-tune the network with initial weight pre-trained on ImageNet ILSVRC2012 published by Google in TensorFlow model zone. Some implementation details in image pre-processing, training, and policy are as follows.

Image pre-processing: We adopt almost the same way as Google Inception [13] for image pre-processing and augmentation, with several differences. Random crop rate is set to 0.2 rather than 0.1 in default. For network evaluation, the center crop is adopted and the corresponding crop rate is set as 0.8. To keep more details of the input image, following the experimental setup as [11], the inputs for both model training and testing are resized before sent to network to 448×448 rather than the default 229×229 .

Training policy: On the training phase, the batch size for Aircraft and Car are both set as 32 with single GPU. For NABirds, 4 GPUs are used to parallelly train the network where the batch size is also set as 32. Learning rate starts from 0.01 and exponentially decays with a decrease factor 0.9 every 2 epochs. RMSProp with momentum 0.9 is chosen as optimizer and decay 0.9, similar with Inception-V3. For Aircraft, two-stage fine-tune is adopted following from [11]. First, we train only the last fully connected layer for several epochs. After that, we train all the network until convergence. For all the networks training on all the datasets, dropout rates in network are set as 0.5.

Test policy: On the testing phase, the image pre-processing and other hyper-parameters are same as the training phase. Because forward calculation of CNN is more GPU-memory-efficient than gradient backward propagation, the batch size setting is bigger than training phase and set as 100 so that the computation efficiency is more thoroughly advanced.

In addition, all experiments are performed on machine with 4 NVIDIA 1080Ti GPUs and Intel(R) Core(TM) i9-7900X CPU @ 3.30GHz.

4.3 Experimental Results

Table 2. Comparison of classification accuracy on the Aircraft [20], Cars [21] and NABirds [22] dataset with state-of-the-art methods. The Inception-V3 network is fine-tuned and evaluated by us. The PS-CNN is evaluated using the same hyper-parameters as Inception-V3. Dashed line means the absents of accuracy of the original paper.

	Aircraft	Car	NABirds
AutoBD	--	88.9%	68.0%
B-CNN	84.5%	91.3%	79.4%
M-CNN	--	--	80.2%
Inception-V3	88.2%	90.3%	80.8%
PS-CNN	89.1%	92.8%	81.9%

As can be seen from the Table 2, our method is with the best performance in accuracy, compared with several state-of-the-art methods. Specially, for Aircraft classification problem, our PS-CNN is 1% higher than the best compared method Inception-V3. For Car classification, our proposed method achieves the best accuracy which is 2% higher compared to Inception-V3. For the larger NABirds dataset, our PS-CNN also achieves best classification rate. We choose 4 binary masks herein to perform a feature map segmentation. We find our proposed method works for all the three datasets. The influence on classification accuracy with varying number of masks thus multiple streams will be discussed then.

4.4 Discussion

Number of Masks: In some degree, the increase in the number of masks will also result in increase in the number of streamings, just like each Inception block in Google Inception family [5, 13]. The classification accuracies of networks with 4 blocks are shown in Table 2. We will discuss the influence of different number of masks on image classification performance herein. We choose 2 (percent_m setting as percent₁ = 0.1, percent₂ = 0.5, percent₃ = 1.0) and 3 (percent_m setting as percent₁ = 0.1, percent₂ = 0.3, percent₃ = 0.5, percent₄ = 1) masks to evaluate the influence. The corresponding classification accuracies upon the three datasets are list in row 2 and 3 respectively in Tab 3.

As can be seen from the Table 3, when we choose 4 masks, the classification accuracies are highest. In cases of 2 and 3 masks, the performance on both Aircraft and NABirds will decrease because the feature maps are not explicitly enough separated. However, for Car dataset, the classification performance is still better than the basic network, i.e., Inception-V3.

Visualization: In the case of 4 masks, as can be seen from the Table 3, the error rate of Stanford Car dataset is about 7.2%, which means that 1165 cars images on Standard dataset are mis-classified. To explore the reason why those

Table 3. Influence of number of masks on the accuracy performance. The Inception-V3 is chosen as our base network.

	Aircraft	Car	NABirds
Inception-V3	88.2%	90.3%	80.8%
2 masks	88.0%	92.2%	80.5%
3 masks	87.8%	92.7%	79.3%
4 masks	89.1%	92.8%	81.9%

**Fig. 2.** Some images which are mis-classified in our experiments. The most of them are mis-classified because they bear a big view angle and ‘strange’ illumination. These problems should be addressed if we want to perform better in this fine-grained image classification problem. (Best viewed in color.)

images are mis-classified, we pick out the mis-classified images in the test set. For simplicity, only 32 (forming as 4×8) of those total 1165 mis-classified images are selected as shown in Fig. 2.

We can see from this overall picture that all the mis-classified cars bear the same characteristics such as various view angle, strong illumination changing, and big occlusion. Those factors may have little influence on basic image classification. However, in this fine-grained tasks, there will be serious impact.

We have magnified the input images and then encode the enlarged feature maps exquisitely in order to make sure that more details of fine-grained image can be ‘observed’. This is a solution to handle the small inter-class problem. But when big intra-class problem is encountered, e.g., view angle, the performance becomes embarrassing. Thus the solutions like pose normalization [23] or Spatial transformer [24] should be considered.

5 Conclusion

In this paper, to avoid the extreme information compression brought by the straightforward coarse Avg/Max pooling upon last convolutional feature maps in general CNN, one *Pixel Saliency based Encoding* method for CNN (PS-CNN) is proposed for fine-grained image classification. First, we provide a saliency matrix to evaluate the saliency of each pixel in feature map. Then, we segment the original feature maps into multiple ones with multiple thresholded saliency matrices, and subsequently squeeze those multiple feature maps into encoded one by using the 1×1 convolution kernel. At last, the encoded feature maps are concatenated with the original one as the last feature representation. By embedding such novel encoding method into the Inception-V3 framework, we achieve perfect performance on the three fine-grained datasets, i.e., Aircraft, Stanford Car, and NABirds. Especially, with this simple yet efficient method, we have achieved the best classification accuracy (81.9%) of large scale dataset NABirds, which demonstrates the efficiency of our PS-CNN. What's more, our pixel saliency based encoding method can be embedded into other convolutional neural networks frameworks as one simple net block.

References

1. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
2. Gao Huang, Zhuang Liu, van der Maaten, Laurens, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017.
3. Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.
4. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
5. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.
6. Bharath Hariharan, Pablo Arbeliz, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 447–456, 2015.
7. Qingyan Duan, Lei Zhang, and Wangmeng Zuo. From face recognition to kinship verification: An adaptation approach. In *2017 IEEE International Conference on Computer Vision Workshop (ICCVW)*, pages 1590–1598. IEEE, 2017.
8. Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei. Fine-grained recognition without part annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5546–5555, 2015.
9. Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

10. Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo. Is second-order information helpful for large-scale visual recognition? *arXiv preprint arXiv:1703.08050*, 2017.
11. Tsung-Yu Lin, Aruni RoyChowdhury, and Subhansu Maji. Bilinear convolutional neural networks for fine-grained visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.
12. Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3640–3649, 2016.
13. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.
14. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 448–456, 2015.
15. X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian. Picking deep filter responses for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1134–1142, 2016.
16. Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5209–5217, 2017.
17. Y. Peng, X. He, and J. Zhao. Object-part attention model for fine-grained image classification. *IEEE Trans. Image Process.*, PP(99):1–1, 2017.
18. Hantao Yao, Shiliang Zhang, Chenggang Yan, Yongdong Zhang, Jintao Li, and Qi Tian. AutoBD: Automated bi-level description for scalable fine-grained visual categorization. *IEEE Trans. Image Process.*, 27(1):10–23, 2018.
19. Xiu-Shen Wei, Chen-Wei Xie, Jianxin Wu, and Chunhua Shen. Mask-CNN: Localizing parts and selecting descriptors for fine-grained bird species categorization. *Pattern Recognit.*, 2017.
20. Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, pages 2204–2212, 2014.
21. Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the International Conference on Computer Vision Workshops (ICCVW)*, pages 554–561. IEEE, 2013.
22. Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L Alexander, David W Jacobs, and Peter N Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2019–2026. IEEE, 2014.
23. Steve Branson, Grant Van Horn, Serge Belongie, Pietro Perona, and Cornell Tech. Bird species categorization using pose normalized deep convolutional nets. *arXiv preprint arXiv:1406.2952*, 2014.
24. Max Jaderberg, Karen Simonyan, and Andrew Zisserman. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.