FINE-GRAINED GIANT PANDA IDENTIFICATION

Rizhi Ding[1], *Le Wang*[1*], *Qilin Zhang*[2], *Zhenxing Niu*[3], *Nanning Zheng*[1], *Gang Hua*[4]

[1]Xi'an Jiaotong University, [2]HERE Technologies, [3]Alibaba Group, [4]Wormpex AI Research

ABSTRACT

The image-based fine-grained identification of individual giant pandas (Ailuropoda melanoleuca) is an emerging technology, and it is extraordinarily challenging due to the extremely subtle visual differences between individual giant pandas and limited annotated training data. To address these challenges, we propose the Feature-Fusion Convolutional Neural Network with Patch Detector (FFCNN-PD) algorithm, which exploits the discriminative local patches and builds a hierarchical representation generated by fusing both global and local features. Specifically, an attentional cross-channel pooling is embedded in the FFCNN-PD to improve the class-specific patch detectors. In addition, we propose a new giant panda identification dataset (iPanda-30) to establish a benchmark. Experiments on the proposed iPanda-30 dataset and other fine-grained recognition datasets demonstrate the effectiveness of the FFCNN-PD algorithm against the existing state-of-the-arts.

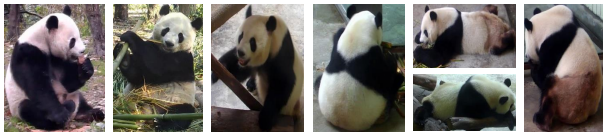
Index Terms— Fine-grained recognition, panda identification, feature fusion, patch detector

1. INTRODUCTION

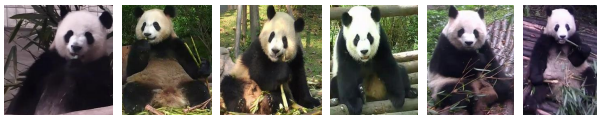
Single image-based giant panda (Ailuropoda melanoleuca) identification could be highly challenging, as illustrated in Fig. 1 (a), the instances of the same panda exhibit dramatic appearance differences (large intra-class variations) due to different illuminations, viewpoints, postures, and occlusions; while images of different individual pandas could have extremely subtle appearance differences (small inter-class distances), as illustrated in Fig. 1 (b).

Fine-grained Panda identification (FGPI) is related to fine-grained visual recognition (FGVR) [1, 2], as both aim at discovering subtle differences. However, the objectives are slightly different. The goal of generic FGVR is the prediction of the subspecies/breeds within a given species (*e.g.*, Labrador versus golden retriever); while FGPI is aim to identify individual pandas (*e.g.*, panda "wuyi" versus "qiyuan" in Fig. 1 (b)). The individual-level appearance differences make the FGPI task more challenging.

Current FGVR methods can be approximately divided into two categories: part-based methods and end-to-end methods. (1) Some part-based methods [3, 4, 5, 6] exploit discriminative part features with localization networks. Such methods rely heavily on manual part annotations, which could be time-consuming and expensive. Worse still, the quality of manual part annotations is difficult to guarantee. Alternatively, some part-based methods [7, 8, 3] embed attention mechanism into sub-networks to utilize part features without additional part annotations, at the expense of complicated network training procedure. (2) End-to-end methods [9, 10] often require less human intervention than their part-based counterparts. Typically, end-to-end methods require only image-level annotations. Some recent methods [11, 12] implement end-to-end training based on bilinear pooling frameworks, but almost all bilinear pooling-based methods only tackle features from the last convolutional layer, which is hardly beneficial for fine-grained tasks.



(a) Images of the same panda (named as "yingying").



(b) Images of different individual pandas.

Fig. 1. (a) Panda "yingying"; (b) different pandas.

To address these challenges in FGPI, we propose the Feature-Fusion Convolutional Neural Network with Patch Detector (FFCNN-PD) algorithm as illustrated in Fig. 2, which does not rely on any sub-networks and can be simply trained end-to-end without additional part annotations. Inspired by [13], we employ the "patch detector" to exploit the most discriminative local patches which are the key factors in the characteristics of giant pandas. Specifically, we fuse the global and local features in the fusion stream (F-Stream) to generate a hierarchical representation, which embodies inter-layer patch feature interactions and allows the network to further focus on more commonly discriminative features. Furthermore, to facilitate the learning of class-specific patch detectors, we specifically introduce a new attentional cross-channel pooling as the convolution filter supervision.

*Corresponding author.

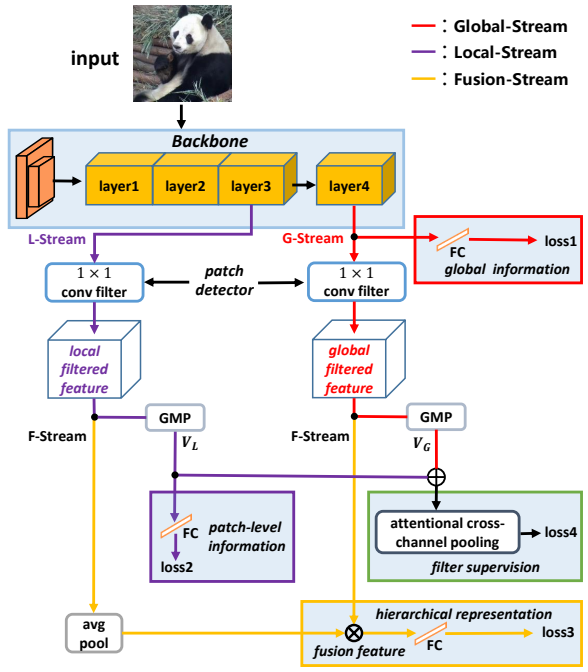


Fig. 2. Overview of the proposed FFCNN-PD algorithm.

The contributions of this paper include: (1) A new, meaningful and challenging FGPI task based on single image, with a new benchmark iPanda-30. (2) Leveraging "patch detectors" and embedding them cross layers to generate more significant representations. (3) A new hierarchical representation to capture inter-layer patch feature interactions and a new attentional cross-channel pooling served as the convolution filter supervision to enhance class-specific patch detectors. (4) An end-to-end FFCNN-PD algorithm with multi-representations that achieves state-of-the-art both on iPanda-30 and other FGVR datasets.

2. APPROACH

As illustrated in Fig. 2, we adopt an asymmetric multi-stream structure inspired by [13], which consists of a local stream (L-Stream) for patch-level supervision ($loss_2$) and a global stream (G-Stream[1]) for global supervision ($loss_1$), respectively. Specifically, this FFCNN-PD network design is quite different from [13] in two additional loss functions (*i.e.*, $loss_3$ and $loss_4$) which are discussed in the following sections.

2.1. Patch Detector and loss_2

We design a 1×1 convolution filter as the patch detector which could select local "patches" with high responses representing subtle discriminative characteristics of one category. After training, this patch detector will respond to different discriminative patches in one different specific category, rather than limited to manually prescribed fixed types of parts for each category. Thanks to this design, our algorithm does not require additional manual part annotations, and the local patches of each category are self-excavated by the network.

Given an image/video frame $\mathbf{X}$, a $C \times H \times W$ feature map can be obtained after a certain convolution layer. Therefore, 1×1 convolution filter operates on the feature map and selects "patches" with high responses, representing the most discriminative local patches. As illustrated in Fig. 2, we embed such patch detectors in both the L-Stream and G-Stream, which is different from [13]. Considering that fine-grained characteristics are usually located on subtle parts, we choose the local "patches" from the L-Stream, which have smaller receptive fields than the ones from the G-Stream. Then we pick the maximal values across channels of the filtered feature map with the Global Max Pooling (GMP), and feed them into a fully-connected layer with a softmax layer to get $loss_2$, which accounts for patch-level information.

2.2. Feature Fusion and loss_3

After passing through the patch detectors of G-Stream and L-Stream, we obtain the global filtered feature map $\mathbf{G} = \mathcal{G}(\mathbf{X})$ and local filtered feature map $\mathbf{L} = \mathcal{L}(\mathbf{X})$, respectively. Suppose $\mathbf{G} \in \mathcal{R}^{C_G \times H_G \times W_G}$, $\mathbf{L} \in \mathcal{R}^{C_L \times H_L \times W_L}$, where C, H and W denote channel, height and width, respectively. Typically, $C_G = C_L$, $H_G < H_L$, $W_G < W_L$. An additional average pooling layer is included prior to computing $loss_3$, which reduces its spatial dimension, $\bar{\mathbf{L}} = \text{Avg-Pool}(\mathbf{L})$, so that $\bar{\mathbf{L}} \in \mathcal{R}^{C_G \times H_G \times W_G}$. Subsequently, feature fusion is implemented with element-wise multiplication, $\mathbf{F} = \mathbf{G} \odot \bar{\mathbf{L}}$, where $\odot$ denotes the Hadamard/element-wise product and $\mathbf{F} \in \mathcal{R}^{C_G \times H_G \times W_G}$, followed by averaging feature map at each channel, which yields $\bar{\mathbf{F}} \in \mathcal{R}^{C_G}$, where each element

$$\bar{\mathbf{F}}(c) = \frac{1}{H_G \cdot W_G} \sum_{i=1}^{H_G} \sum_{j=1}^{W_G} \mathbf{F}(c, i, j), \tag{1}$$

where $c = 1, \cdots, C_G$; $i = 1, \cdots, H_G$; and $j = 1, \cdots, W_G$. Further ℓ_2 normalization is carried out and the final "hierarchical representation" is obtained as $\tilde{\mathbf{F}} = \bar{\mathbf{F}} / \|\bar{\mathbf{F}}\|_2$. $\tilde{\mathbf{F}}$ is subsequently fed into a fully connected layer and a softmax layer, where the $loss_3$ is computed, as illustrated in Fig. 2.

2.3. Attentional Convolution Filter Supervision: loss_4

The fully connected layer used in computing $loss_2$ inevitably scrambles all discriminative patches together, with no specific guarantee of the "patch detector" emphasizing specific discriminative patches of a certain category. Therefore, we need a different loss function to encourage convolution filters in the patch detectors to emphasize class-specific discriminative patches. While doing so, we propose an attentional cross-channel pooling module as illustrated in Fig. 3.

Specifically, let the convolution filters (*i.e.*, patch detectors) be of size $(k \cdot n) \times 1 \times 1$, where n is the total number of of category and k is a pre-defined number of the top-k most discriminative local patches per panda/category. After

[1]The single G-Stream network with only $loss_1$ is later utilized as one of the baselines, denoting generic classification network without special treatment of patches or fine-grained feature representation.

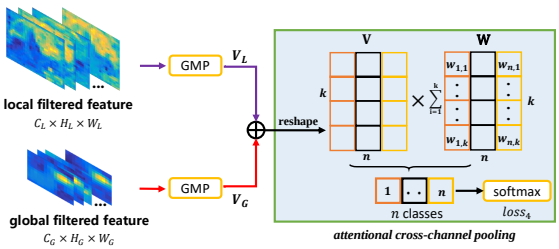


Fig. 3. The proposed attentional cross-channel pooling module is detailedly illustrated in the green rectangle.

passing through the convolution filter and the GMP, a $(k \cdot n)$-dimensional feature vector is obtained. We conduct an element-wise addition as $\mathbf{V} = \mathbf{V_L} + \mathbf{V_G}$, where $\mathbf{V_L}$ is from the L-Stream and $\mathbf{V_G}$ is from the G-Stream in Fig. 2. For notational convenience, we reshape the $(k \cdot n)$-dimensional feature vector as a matrix $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_n] \in \mathcal{R}^{k \times n}$. In [13], this combined $\mathbf{V}$ is simply average-pooled to generate a n-dimensional vector $\mathbf{a} \in \mathcal{R}^{n \times 1}$,

$$\mathbf{a} = \frac{1}{k}\mathbf{V}\mathbf{1}_{k \times 1}, \tag{2}$$

where $\mathbf{1}_{k \times 1}$ denotes an all-ones vector of size $k \times 1$.

Simple averaging strategies such as this in Eq. (2) could induce balanced responses in these k patches during back-propagation. We are concerned that Eq. (2) might assign equal weights to patches with different semantic significance, and incur a performance penalty. Alternatively, we propose a new attention mechanism which automatically learns the weights assigned to the k local patches. Let $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_n] \in \mathcal{R}^{k \times n}$ denote the attentional weights, with each column $\mathbf{w}_i = [w_{i,1}, \cdots, w_{i,k}]^T \in \mathcal{R}^{k \times 1}$, $i = 1, \cdots, n$. All elements in $\mathbf{W}$ are initialized with the value of $1/k$, and $\mathbf{W}$ is automatically updated during the training process via back-propagation. Specifically,

$$\tilde{\mathbf{a}} = (\mathbf{V} \odot \mathbf{W})\, \mathbf{1}_{k \times 1}, \tag{3}$$

where $\odot$ denotes the element-wise product. Subsequently, $\tilde{\mathbf{a}}$ is fed into a softmax layer to generate $loss_4$.

3. EXPERIMENTS

3.1. Implementation Details and the iPanda-30 Dataset

The proposed FFCNN-PD algorithm is implemented with PyTorch and trained with four Nvidia 1080Ti GPUs. Our backbone is Resnet50 [14] pre-trained on ImageNet [15]. Regular stochastic gradient descent optimizer is used with a momentum of 0.9, a weight decay of 5×10^{-4} and a batch size of 32. and the learning rate decay is set as a factor of 0.1 for every 20 epochs. The number of the most discriminative patches per category, *i.e.*, k is empirically fixed at 10. The overall loss is a summation of $loss_1$, $loss_2$, $loss_3$, and $loss_4$.

We collect giant panda streaming videos from the Panda Channel[2] which monitors the daily routines of pandas.We recruit professional zookeepers and breeders to provide identity annotations. Subsequently, we use the structural similarity index measure (SSIM) [16] to compute the similarities between adjacent video frames, and only retain the "key" frames with small similarities with their previous ones. We further manually select the images with various illuminations, viewpoints, postures, and occlusions. In addition, we manually crop out each individual panda with a tight bounding box without aspect ratio requirements. The iPanda-30 dataset consists of $3,552$ images of 30 individual giant pandas with $54 \sim 220$ images per panda. This iPanda-30 dataset is available online[3], and we are in the process of further expanding the dataset.

Method	Max.	Mean	δ	P-value	Conf.
Baseline	80.8	80.0	1.43	1.1e-5	>99%
MFC	86.3	85.6	0.93	5.1e-2	>94%
HBP[10]	78.4	77.6	0.66	2.7e-8	>99%
BCNN[17]	81.5	80.7	1.00	4.6e-6	>99%
DFL-CNN[13]	86.7	85.9	0.52	3.7e-2	>96%
FFCNN-PD	**87.7**	**86.7**	0.56	-	-

Table 1. Performance comparison on the iPanda-30 dataset via 5 random trials.

3.2. Experimental Result

Comparison on iPanda-30. A performance comparison between the proposed FFCNN-PD algorithm and competing ones on iPanda-30 is summarized in Table 1. All statistics in Table 1 are obtained with 5 independent random training/testing splits ($2,120$ and $1,432$ images in training and testing splits, respectively). The maximum (Max.), mean (Mean) and standard deviation (δ) of the accuracy (%) over these 5 trials are reported. To account for coincidental fluctuations and reveal statistical significance, a series of 5 one-tailed student's t-tests are carried out, with null hypothesis H_0 being there is no effective advantage of FFCNN-PD over other competing algorithms. P-values and the confidence intervals (Conf.) of the alternative hypothesis H_1 being true are also reported with each competing algorithm.

We also implement a multi-feature concatenation ("MFC") method, which directly concatenates features from multiple convolution layers, and it outperforms "Baseline", indicating the value of incorporating cross-layer features. More importantly, we re-implement three FGVR methods, *i.e.*, **(1)** the classical bilinear pooling method BCNN [17]; **(2)** a hierarchical bilinear pooling framework HBP [10], which concatenates multiple cross-layer bilinear features; **(3)** DFL-CNN [13], which learns a mid-level representation to capture class-specific discriminative patches. Our proposed FFCNN-PD outperforms all these competing ones with confidence intervals of at least $> 94\%$.

Ablation Studies. To isolate the effects of the four losses in our proposed FFCNN-PD, we conduct experiments on

[2]Video streaming website in Chinese, `http://live.ipanda.com`.

[3]`https://github.com/iPandaDateset/ipanda30.git`

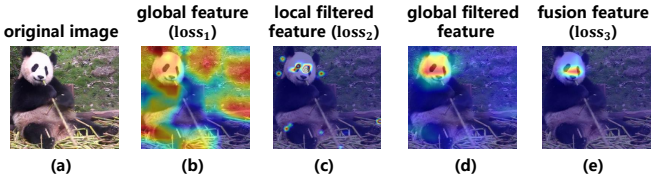


Fig. 4. Grad-CAM [18] visualization of "global filtered feature" and the convolutional feature maps at the end of different branch that produces $loss_1$, $loss_2$,$loss_3$, respectively.

$loss1$	$loss2$	$loss3$	$Avera$	$Atten$	Max.	Mean	δ
✓					80.8	80.5	2.00
	✓				75.3	74.5	0.69
		✓			69.4	68.3	0.79
	✓	✓			76.5	76.0	0.41
✓	✓				86.0	85.8	0.23
✓		✓			86.3	85.9	0.42
	✓	✓		✓	77.1	76.7	0.41
✓	✓			✓	87.3	86.3	0.78
✓	✓	✓			86.8	86.1	0.54
✓	✓	✓	✓		87.2	86.1	0.82
✓	✓	✓		✓	**87.7**	**86.7**	0.56

Table 2. Classification accuracy with different loss combinations on the iPanda-30 dataset via 5 random trials.

iPanda-30 dataset with different loss combinations. The statistics are summarized in Table 2. With a single loss (*e.g.*, $loss_1$ or $loss_2$ or $loss_3$), the performance deteriorates evidently. The combination of $loss_1$ with any other loss terms boosts their performance evidently, indicating the necessity of including the global image features as visualized in Fig. 4 (b). Loss combinations with $loss_3$ terms always perform better than those without it (*e.g.*, $loss_2 + loss_3$ versus $loss_2$), which agrees with our speculation that the hierarchical representation is beneficial. Specially, the fusion feature ($loss_3$ in Fig. 4 (e)) is fused by the local filtered feature ($loss_2$ in Fig. 4 (c)) and the global filtered feature (in Fig. 4 (d)), which further activates commonly regions of high responses and assists the network to emphasize important regions and suppress background noise. It can also be observed that such discriminative regions (in Fig.4 (c)) are concentrated near the panda faces, especially around their unique black patches around their eyes. We are surprised that these visualizations agree well with a popular article [19] in the Science magazine, which claims that such back eye patches "may help pandas recognize one another".

Note that $loss_4$ serves only as the convolution filter (*i.e.,* "patch detector") supervision, thus it is unfair to evaluate it independently (*i.e.*, no pure $loss_4$ row in Table 2) and it must be verified in conjunction with $loss_2$. The last three rows of Table 2 present the classification accuracies of our method with different convolution supervisions, which are characterized by columns "Avera" (as in Eq. (2), described in [13]) and "Atten" (proposed by us in Eq. (3)). They are all combined with $loss_1$+$loss_2$+$loss_3$. The results show that "average pooling" and "attentional pooling" both outperform "None", with "attentional pooling" slightly better than "average pooling", which indicates the value of our attentional pooling. Ultimately, the loss combination of $loss_1 + loss_2 + loss_3$+"attentional pooling" achieves the best result, which supports our claim that the patch-level information, the global information and the hierarchical representation jointly contribute to the overall performance.

	Method	CUB	Cars	Aircraft
part-based	PN-CNN [1]	85.4	-	-
	PC-DenseNet [4]	86.9	92.9	89.2
	RA-CNN [22]	85.3	92.5	88.2
	MA-CNN [7]	86.3	92.8	89.9
	OSME [23]	86.5	93.0	-
end-to-end	B-CNN [17]	84.1	91.3	84.1
	CBP [24]	84.0	-	-
	LRBP [25]	84.2	90.9	87.3
	KP [9]	86.2	92.4	86.9
	HBP [10]	87.1	93.7	90.3
	DFL-CNN [13]	87.4	93.8	92.0
	FFCNN-PD($ours$)	**87.9**	**94.7**	**93.2**

Table 3. Classification accuracy on the larger FGVR datasets.

Comparison on other larger Fine-grained datasets. To further demonstrate the effectiveness of our method, we compare FFCNN-PD with 11 existing FGVR methods on other three larger FGVR datasets (*i.e.*, CUB-200-2011 [1], Stanford Cars [20] and FGVC-Aircraft [21]). As the results in Table 3 shown, our end-to-end method, without additional manual part annotations, achieves state-of-the-art performance on all the three widely used FGVR datasets.

4. CONCLUSION

We propose the FFCNN-PD method to address the interesting yet challenging fine-grained giant panda identification (FGPI) task. It exploits discriminative local image patches and fuses both global and local features to generate a hierarchical representation. Specifically, a new attentional cross-channel pooling module is proposed to provide more effective training supervision of the patch detectors. Moreover, we propose a new iPanda-30 dataset to evaluate our proposed FFCNN-PD algorithm and existing FGVR algorithms with the FGPI task.

Acknowledgements. This work was supported partly by National Key R&D Program of China Grant 2018AAA0101400, NSFC Grants 61629301, 61773312, and 61976171, China Postdoctoral Science Foundation Grant 2019M653642, and Young Elite Scientists Sponsorship Program by CAST Grant 2018QNRC001.

5. REFERENCES

[1] Steve Branson, Grant Van Horn, Serge Belongie, and Pietro Perona, "Bird species categorization using pose normalized deep convolutional nets," in *BMVC*, 2014.

[2] Yue Chen, Yalong Bai, Wei Zhang, and Tao Mei, "Destruction and construction learning for fine-grained image recognition," in *CVPR*, 2019, pp. 5157–5166.

[3] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al., "Spatial transformer networks," in *NIPS*, 2015, pp. 2017–2025.

[4] Abhimanyu Dubey, Otkrist Gupta, Pei Guo, Ramesh Raskar, Ryan Farrell, and Nikhil Naik, "Pairwise confusion for fine-grained visual classification," in *ECCV*, 2018, pp. 70–86.

[5] Shaoli Huang, Zhe Xu, Dacheng Tao, and Ya Zhang, "Part-stacked cnn for fine-grained visual categorization," in *CVPR*, 2016, pp. 1173–1182.

[6] Xiu-Shen Wei, Chen-Wei Xie, Jianxin Wu, and Chunhua Shen, "Mask-cnn: Localizing parts and selecting descriptors for fine-grained bird species categorization," *PR*, vol. 76, pp. 704–714, 2018.

[7] Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo, "Learning multi-attention convolutional neural network for fine-grained image recognition," in *ICCV*, 2017, pp. 5209–5217.

[8] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo, "Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition," in *CVPR*, 2019, pp. 5012–5021.

[9] Yin Cui, Feng Zhou, Jiang Wang, Xiao Liu, Yuanqing Lin, and Serge Belongie, "Kernel pooling for convolutional neural networks," in *CVPR*, 2017, pp. 2921–2930.

[10] Chaojian Yu, Xinyi Zhao, Qi Zheng, Peng Zhang, and Xinge You, "Hierarchical bilinear pooling for fine-grained visual recognition," in *ECCV*, 2018, pp. 574–589.

[11] Xing Wei, Yue Zhang, Yihong Gong, Jiawei Zhang, and Nanning Zheng, "Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification," in *ECCV*, 2018, pp. 355–370.

[12] Sijia Cai, Wangmeng Zuo, and Lei Zhang, "Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization," in *ICCV*, 2017, pp. 511–520.

[13] Yaming Wang, Vlad I Morariu, and Larry S Davis, "Learning a discriminative filter bank within a cnn for fine-grained recognition," in *CVPR*, 2018, pp. 4148–4157.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[16] Zhou Wang, Alan C Bovik, Hamid R Sheikh, Eero P Simoncelli, et al., "Image quality assessment: from error visibility to structural similarity," *IEEE T-IP*, vol. 13, no. 4, pp. 600–612, 2004.

[17] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji, "Bilinear cnn models for fine-grained visual recognition," in *ICCV*, 2015, pp. 1449–1457.

[18] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *ICCV*, 2017, pp. 618–626.

[19] Virginia Morell, "How pandas got their patches," *Science, doi:10.1126/science.aal0840*, 2017.

[20] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei, "3d object representations for fine-grained categorization," in *CVPR*, 2013, pp. 554–561.

[21] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi, "Fine-grained visual classification of aircraft," *arXiv preprint arXiv:1306.5151*, 2013.

[22] Jianlong Fu, Heliang Zheng, and Tao Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *CVPR*, 2017, pp. 4438–4446.

[23] Ming Sun, Yuchen Yuan, Feng Zhou, and Errui Ding, "Multi-attention multi-class constraint for fine-grained image recognition," in *ECCV*, 2018, pp. 805–821.

[24] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell, "Compact bilinear pooling," in *CVPR*, 2016, pp. 317–326.

[25] Shu Kong and Charless Fowlkes, "Low-rank bilinear pooling for fine-grained classification," in *CVPR*, 2017, pp. 365–374.