applied sciences

MDPI

Article

Large-Scale Fine-Grained Bird Recognition Based on a Triplet Network and Bilinear Model

Zhicheng Zhao [1,2,*], Ze Luo [2], Jian Li [2], Kaihua Wang [1,2] and Bingying Shi [1,2]

[1] University of Chinese Academy of Sciences, Beijing 100049, China; wangkaihua@cnic.cn (K.W.); shibingying@cnic.cn (B.S.)

[2] Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China; luoze@cnic.cn (Z.L.); lijian@cnic.cn (J.L.)

* Correspondence: zhaozhicheng@cnic.cn; Tel.: +86-176-109-83286

Received: 7 September 2018 ; Accepted: 11 October 2018 ; Published: 13 October 2018

Abstract: The main purpose of fine-grained classification is to distinguish among many subcategories of a single basic category, such as birds or flowers. We propose a model based on a triple network and bilinear methods for fine-grained bird identification. Our proposed model can be trained in an end-to-end manner, which effectively increases the inter-class distance of the network extraction features and improves the accuracy of bird recognition. When experimentally tested on 1096 birds in a custom-built dataset and on Caltech-UCSD (a public bird dataset), the model achieved an accuracy of 88.91% and 85.58%, respectively. The experimental results confirm the high generalization ability of our model in fine-grained image classification. Moreover, our model requires no additional manual annotation information such as object-labeling frames and part-labeling points, which guarantees good versatility and robustness in fine-grained bird recognition.

Keywords: bird recognition; fine-grained; triplet network; bilinear model; Xception

1. Introduction

Image classification is a classical research topic in the computer vision field. Traditional image classification mainly categorizes semantic-level images or instance-level images. Semantic-level classification includes scene recognition [1,2] and object recognition [3,4], where the latter identifies different categories of objects such as cats and dogs. Meanwhile, instance-level classification distinguishes among individuals of an object, such as faces. Located between these two types, fine-grained image classification provides a more detailed class precision than coarse-grained image classification (such as object recognition), and detects subtle differences among the classes, often consisting of small local differences. For example, fine-grained classification distinguishes different types of birds [5], dogs [6], flowers [7], or any other object of interest. Fine-grained image classification is especially concerned with identifying the important distinguishing features. Such fine-grained features need to be tested against complex image backgrounds, and factors such as illumination, deformation, and occlusion interference with ambient noise should be reduced as far as possible [8]. The acquisition of fine-grained features is more complex than acquisition of coarse-grained features, and relies on image annotation to determine the complex parameters in the model while avoiding the over-fitting problem caused by small data amounts. Therefore, improving the extraction and choosing the appropriate structure of the convolutional neural network and the connection of the network are key initiatives.

Fine-grained image classification can be divided into strongly supervised and weakly supervised approaches. In the strongly supervised approach, the category labels of the images during model training are supplemented by additional manual annotation information such as object-labeling

boxes and part-labeling points. The authors of [9] detected the object levels and local areas in fine-grained images using a region-based convolutional neural network (R-CNN) algorithm. During the training phase, the R-CNN algorithm must mark the object frame and the part-labeling point. The object-labeling frame is also required in the test images. In [10], local areas were detected by a pose normalization algorithm. The images were cropped around the detected label box, and the local information at different levels was extracted for pose alignment. Finally, the convolution characteristics of the different layers were obtained. The model was constructed from two modules: one for local positioning, the other for feature learning of the global and local image blocks. However, the practicality of pose normalization is limited by the high expense of acquiring the annotation information.

Weakly supervised fine-grained image classification uses labels alone, without requiring additional annotation information. As local-area information is essential in fine-grained image classification, the detection of local areas would improve the performance of weakly supervised fine-grained image classification. The first algorithm with no reliance on annotation information was proposed in [11]. This algorithm uses two levels of features—object-level features and local-level features—and completes the fine-grained image classification by using category labels. The authors of [12] extracted local-area information based on several essential points derived from the CNN features [13]. A bilinear CNN model that performs local-area detection and feature extraction using two networks was proposed in [14].

Birds are an important part of natural ecosystem. Correct recognition of birds is very important for the protection and research of birds. However, classifying birds correctly is very difficult because the differences between the different species are very subtle. Therefore, our paper focuses on the fine-grained identification of birds. We propose a network based on the triplet network and the bilinear model. The inputs of the network consist of three images: a pre-selected image, another image of the same kind, and an image from a different category. After our experiment, we adopted the deep-learning architecture Xception as the basic network, and divided the whole architecture into two branches. One branch processes the features extracted by Xception and obtains the bilinear features. The output of the fully connected layer is then connected to an external output for category prediction. The other branch obtains a 2048-dimensional feature representation after global-average pooling and then obtains the squared distance between an identical and a different specimen. Based on this distance, two images are judged as being of the same or different species. The accuracy of this method was 88.91% on the large-scale bird dataset built by us, and 85.31% on the CUB200-201 public bird dataset. Our model has a good application prospect in bird recognition and protection.

2. Background

2.1. Dataset and Evaluation Metric

Birds-1096: Our own database (Birds-1096) includes 1096 bird categories, each containing 200–350 images (giving a total of 459,828 images). The original dataset is composed of images of different resolutions, we resize the image to (299, 299, 3) for later use. We randomly selected 10 pictures from each category for testing, thereby assigning 109,600 images to the test set. Targets in the same category have diverse postures and large illumination changes, whereas the between-category differences are very subtle, with similar shapes and colors of the target. Some images from Birds-1096 are displayed in Figure 1.

Figure 1. Images sampled from Birds-1096. Three images were randomly selected in each category (column).

CUB200-2011: This dataset is the most widely used dataset for fine-grained image classification. It contains 11,788 images of 200 bird subcategories and is divided into 5994 images for training and 5794 images for testing [15]. Each subcategory consists of 30 images for training and 1130 images for testing, and each image has detailed annotations: a subcategory label, an object bounding box, 15 part locations, and 312 binary attributes. All attributes are visual in nature, pertaining to the color, pattern, or shape of a particular part. Some images from the CUB200-2011 dataset are displayed in Figure 2.

The classification performances of our approach and another method (for comparison) were evaluated by the accuracy metric, a widely used performance index in fine-grained image classification studies [16,17].

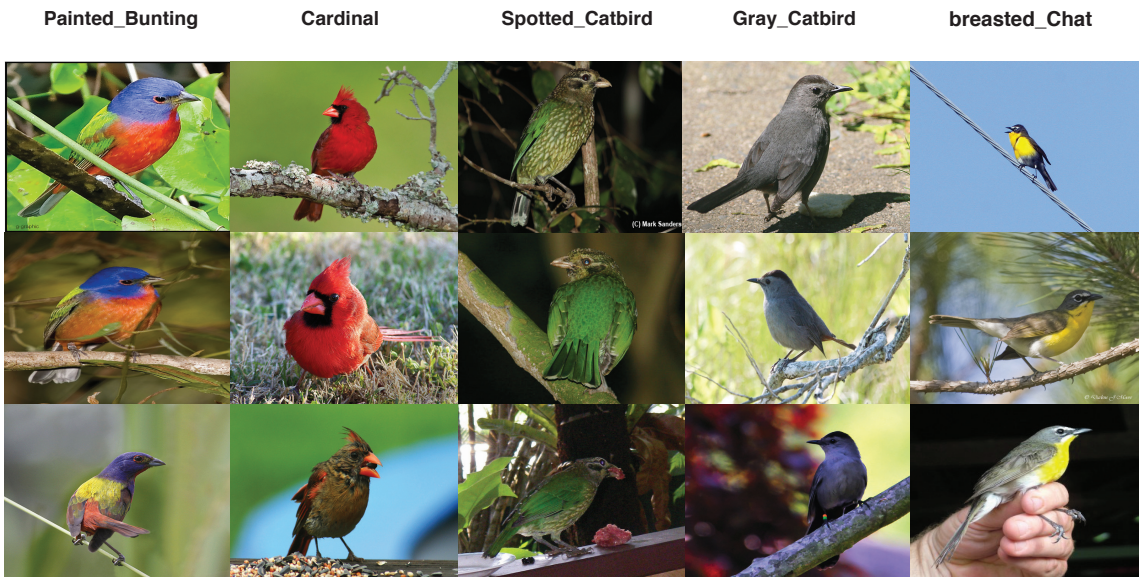

Figure 2. CUB200-2011.We selected three images from different categories.

2.2. *Xception*

Xception is an improvement over InceptionV3 [18] that uses deep separable convolution [19]. In traditional convolutional networks, the convolution kernel is deep, and the convolutional layer

looks for correlations across space and depth [20]. The basic idea of Xception is that cross-channel correlation and spatial correlation can be completely separated, and it is best not to map them jointly. Instead of splitting the input data into several compressed data blocks, the spatial correlation is mapped separately for each output channel, and a 1×1 depth convolution is then performed to obtain cross-channel correlation. The $3D$ map with a $2D + 1D$ map, including a spatial convolution that is performed separately for each channel, is replaced, followed by a 1×1 convolution per channel, which can be thought of as the first correlation across a $2D$ space, and a $1D$ space correlation is then requested. The deep separable convolution of Xception increases the network width, which not only improves the classification accuracy but also enhances the network's ability to learn subtle features. Thus, it is a feasible module for fine-grained image classification.The structure of Xception is shown in Figure 3.

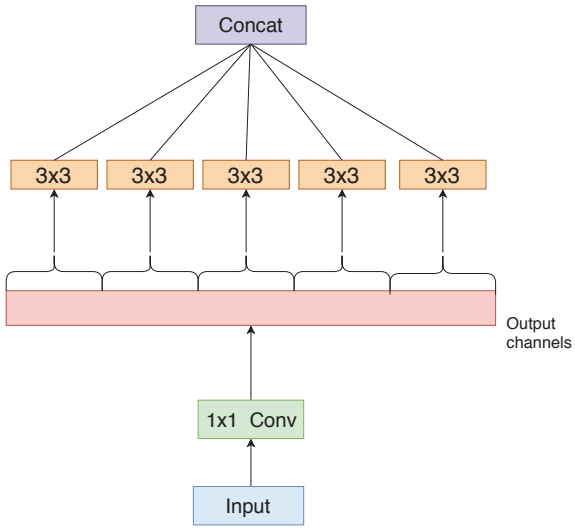

Figure 3. An "extreme" version of our Inception module, with one spatial convolution per output channel of the 1×1 convolution [19].

The image was resized to $(299, 299, 3)$ and normalized. We used Inception v3 [21], Resnet-50 [22], and Xception [19] pre-trained on ImageNet [23] and fine-tuned in Birds-1096 and CUB200. Table 1 shows the accuracies of the compared models trained on the two bird datasets, and the corresponding accuracy (Acc) curves are plotted in Figure 4.

Based on the experimental results, we selected Xception as our basic network. The Xception model yielded higher accuracy than Resnet-50 and Inception-v3 on both datasets, indicating that the Xception model learns the subtle features in fine-grained image classification. The lower accuracy on the CUB200-2011 dataset than on Birds-1096 is due to the lack of training samples in each category.

Table 1. Accuracies of the evaluated models trained on Birds-1096 and CUB200-2011.

Model	Dataset	Acc
Inception-v3	Birds-1096	76.73%
Resnet-50	Birds-1096	77.25%
Xception	Birds-1096	79.68%
Inception-v3	CUB200-2011	67.93%
Resnet-50	CUB200-2011	67.03%
Xception	CUB200-2011	71.38%

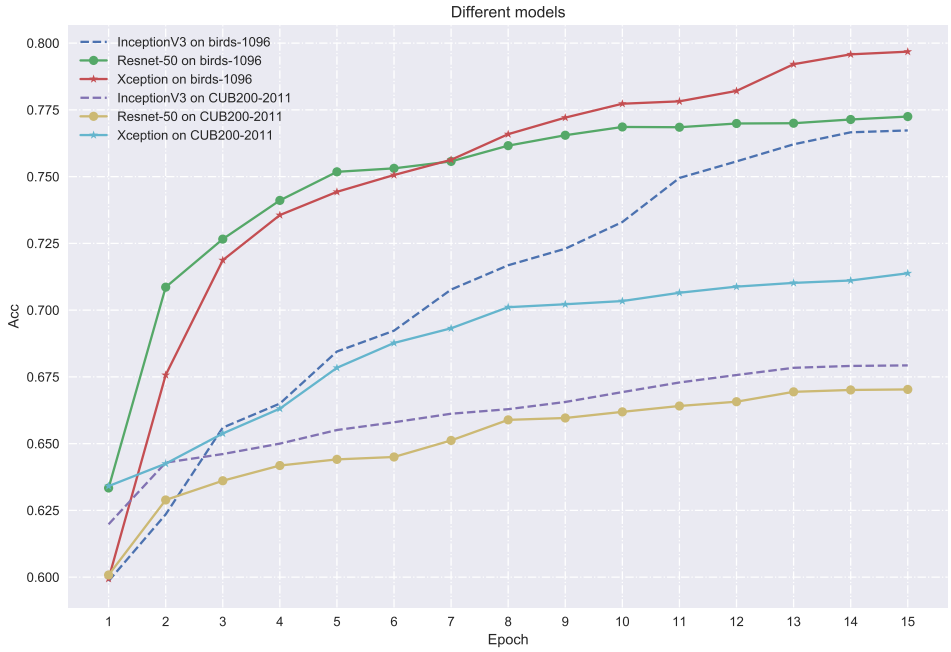

Figure 4. Acc curves of the evaluated basic networks trained on Birds-1096 and CUB200-2011.

2.3. Fully Shared Bilinear Model

Deep learning is successful, largely because it integrates the original decentralized processing (feature extraction and model training) into a complete system, enabling overall end-to-end optimization training. Lin et al. [24] designed an end-to-end network model called bilinear CNN (B-CNN), which achieved a very high accuracy on the CUB200-2011 dataset with a weakly supervised fine-grained classification model.

A B-CNN for image classification (see Figure 5) consists of a quadruple $B = (f_a, f_b, P, C)$. Here, f_A and f_B are feature functions based on CNN A and CNN B, respectively, P is a pooling function, and C is a classification function. Each feature function is a mapping $f : \mathcal{L} \times \mathcal{I} \rightarrow \mathbb{R}^{K \times D}$ that takes an image $I \in \mathcal{I}$ and a location $l \in \mathcal{L}$ and outputs a feature of size $K \times D$. Outputs are combined at each location using the matrix outer product; i.e., the bilinear combination of f_A and f_B at the location l is given by

$$bilinear(l, I, f_A, f_B) = f_A(l, I)^T f_B(l, I). \tag{1}$$

Both f_A and f_B must have the same feature dimension K to be compatible. The value of K depends on the particular model. The pooling function P aggregates the bilinear combination of features across all locations in the image to obtain a global image representation $\Phi(I)$. We use sum pooling in all our experiments, i.e.,

$$\Phi(I) = \sum_{l \in \mathcal{L}} bilinear(l, I, f_A, f_B). \tag{2}$$

Note that pooling ignores the feature locations, so the bilinear feature $\Phi(I)$ is an orderless representation. If f_A and f_B extract features of size and $K \times N$ respectively, then $\Phi(I)$ is the size of $M \times N$. The bilinear feature is a general purpose image representation that can be used with a classifier C. Intuitively, the outer product conditions the outputs of features f_A and f_B on each other by considering their pairwise interactions, similar to the feature expansion in a quadratic kernel.

The bilinear model can be divided into (a) non-shared, (b) partially shared, and (c) fully shared [25]. Here, we adopt a fully shared approach based on Xception (see Figure 6).

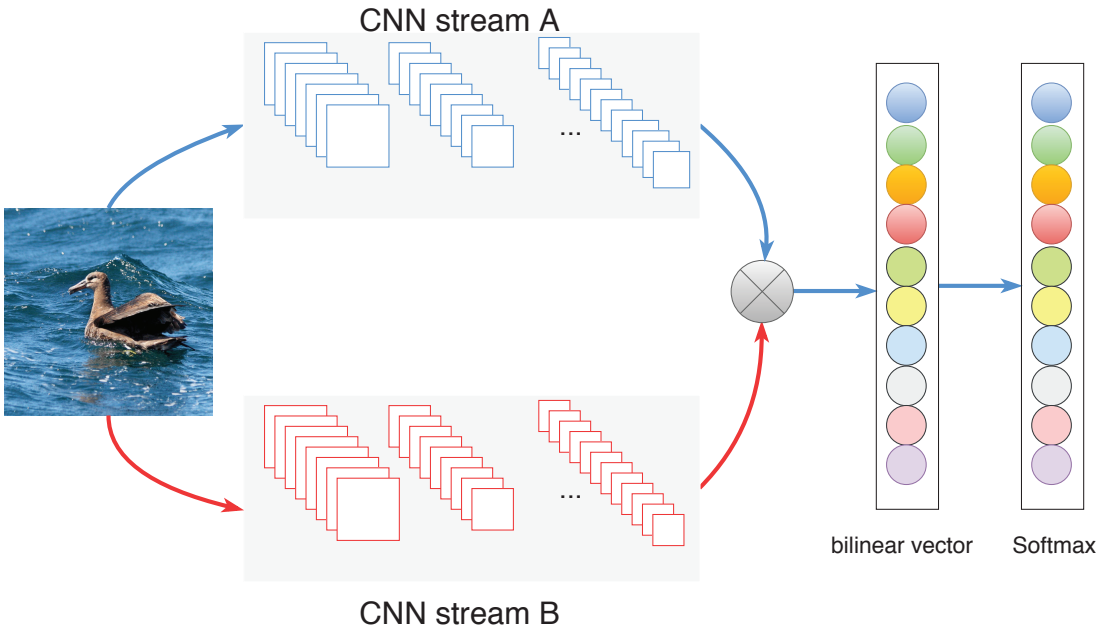

Figure 5. Image classification by a bilinear convolutional neural network (B-CNN). An image is passed through CNNs A and B. The CNN outputs at each location are combined using the matrix outer product and then average-pooled to obtain the bilinear feature representation. After passing the feature representation through a linear softmax layer, the class prediction is obtained [24].

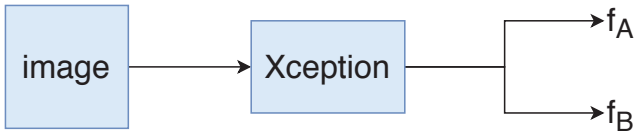

Figure 6. Fully shared bilinear model based on Xception.

2.4. Triplet Networks

The triplet network (inspired by the Siamese network) [26,27] is composed of three instances of the same feedforward network (with shared parameters). When fed with these three samples, the network outputs two intermediate values, namely, the L2 distances between the embedded representations of two of its inputs and the representation of the third input [28]. The three inputs are denoted as x, x^+, and x^-, and the embedded representation of the network is denoted as $Net(x)$. The penultimate layer is the following vector:

$$TripletNet(x, x^+, x^-) = \begin{bmatrix} ||Net(x) - Net(x^-||_2) \\ ||Net(x) - Net(x^+||_2) \end{bmatrix}. \quad (3)$$

Equation (3) encodes the pair of distances between the x^+ and x^- inputs and the reference input x. The training process make the distances between different categories larger than the distance between images of the same class [29]. Figure 7 shows the structure of a triplet network.

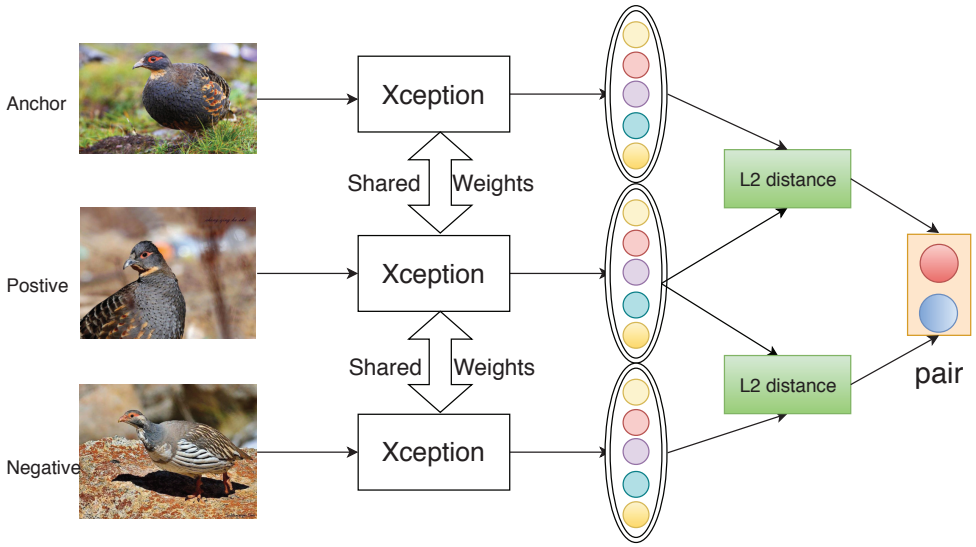

Figure 7. Structure of a triplet network.

3. Methods

3.1. The Architecture

Our architecture combines the triplet network with a bilinear model. As an example, we assume a $(299 \times 299 \times 3)$-sized image. After processed by Xception, we retrieve a $(14 \times 14 \times 2048)$-sized feature map. As mentioned above, one of the Xception branches is based on the characteristics extracted by Xception. Adding a global-average pooling layer [30] remarkably improves the localization ability of the CNN, despite its training on image-level labels [31]. The pooled averaging outputs a 2048-dimensional vector. The distance representation between this vector and another 2048-dimensional vector is obtained, and whether this distance represents a positive or negative sample pair is predicted by two neurons selected for that purpose. The other branch of Xception reduces the problem dimension through a $(1 \times 1 \times 128)$ sized filter, obtains the bilinear vector, fully connects the layer outputs.

The bilinear pooling mainly includes the following processes: (1) obtain the bilinear feature $\Phi(x)$; (2) flatten the feature and use the signed square-root function $(y \leftarrow sign(\mathbf{x})\sqrt{|\mathbf{x}|})$; (3) make l_2 normalization l_2 $(z \leftarrow y/ \parallel y \parallel_2)$ [32]. This improves performance in practice.

3.2. Train and Test

The training and testing steps are given below.

Step 1: Image data normalization. First, the image was scaled to (299×299) pixels, and each pixel data type of the image was converted to a floating point type and normalized to $[-1, +1]$ by the following formula:

$$J = (I/255.0 - 0.5) \times 2 \tag{4}$$

where I is the image pixel matrix, and J is the result of data type conversion and normalization;

Step 2: Model parameter initialization. Neural networks are commonly trained by fine-tuning, which extracts the features by a publicly pre-trained model and uses them in the targeted classification. Fine-tuning does not require a complete retraining of the model, which improves the efficiency and achieves a good result with fewer iterations. We initialized the convolutional layer and the softmax layer using the pre-training model parameters of the ImageNet classification [33] and Xavier's [34] method, respectively.

Step 3: Model training. We first trained the branch of the bilinear model using the Adam optimizer [35] and then trained the two branches together using stochastic gradient descent (SGD) [36] with a learning rate of 0.001. Under this training approach, the network converged to a good result.

To improve the generalization ability and reduce the over-fitting of the network, we adopted the Dropout regularization technique with a value of 0.5.

The model is trained in an end-to-end manner. As shown in Figure 8, the first half of the model constitutes the convolutional layer and the pooling layer. Therefore, the whole model can be trained using the gradient value of the latter half of the model. Suppose that, at each position l, the feature extraction function f_A outputs A and A^T. The pooled bilinear feature is $A^T \times A$, so dl/dx represents the gradient of the loss function of feature x. The gradient of the loss function at the network output is then obtained by the chain rule, thereby completing the end-to-end training of the model.

The bilinear branch makes a prediction from an entered image, whereas the triplet network verifies a pair of images.

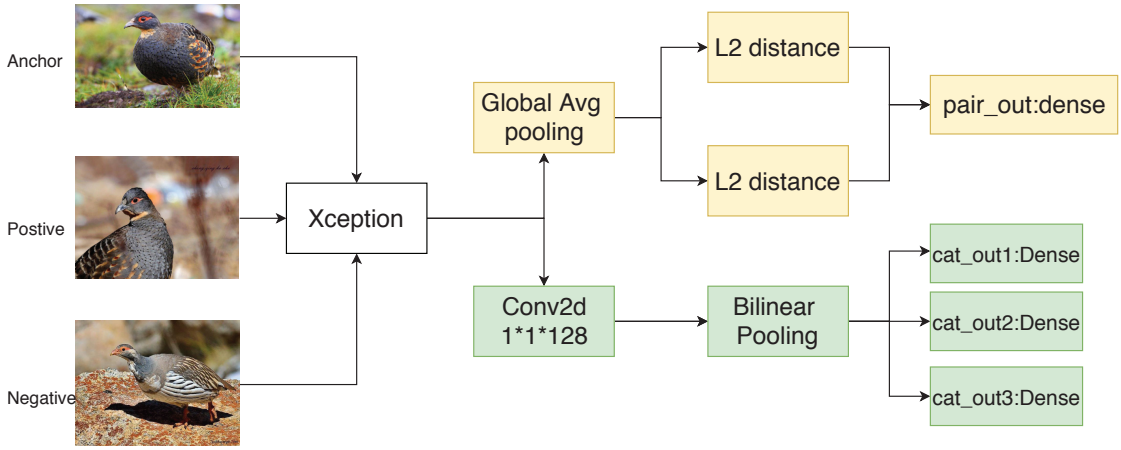

Figure 8. Architecture of our model.

3.3. Loss Function

Our entire network is weakly supervised by label learning. As the loss function in the branches of our network, we adopted the weighted cross entropy [37]. The loss in the whole network is the sum of the weighted cross entropies of the different branches. By trial-and-error experiment, we found that the network achieved superior results when the bilinear and triplet-network branches were weighted by 1.0 and 0.5, respectively.

$$C = -\sum_{i=1}^{n} y_i * log(\hat{y}_i). \tag{5}$$

$$Loss = \sum_{j} W_i C_j. \tag{6}$$

In the above expressions, y_i denotes the label, $\hat{y}_i$ is the predictive probability, j is the branch number, and W is the loss weight.

4. Results

4.1. Comparisons with Other Models

Gradient-weighted class activation mapping (Grad-CAM) improves the transparency of convolutional neural network (CNN)-based models by visualizing the important regions of the input from a predictive perspective [38]. The Saliency Maps are feature maps that tell us how the pixels in the image affect the image classification results [39]. We generated guided Grad-CAM visual explanations and Saliency maps to better understand the focus in our deep networks [40] through gradient-based localization, and why our architecture improves the classification results.

To identify the maximum stimulation corresponding to the species on the original image, we visualized the dense layer of the network. By combining the triplet network, our model captures the details in the original images better than the Xception+Bilinear model. Representative results of the two methods are compared in Figure 9. The toes, torso, limbs, and other details of the birds confirm that our network better extracts features that distinguish among the different species. Our network pays more attention to detail and achieves a better overall effect.

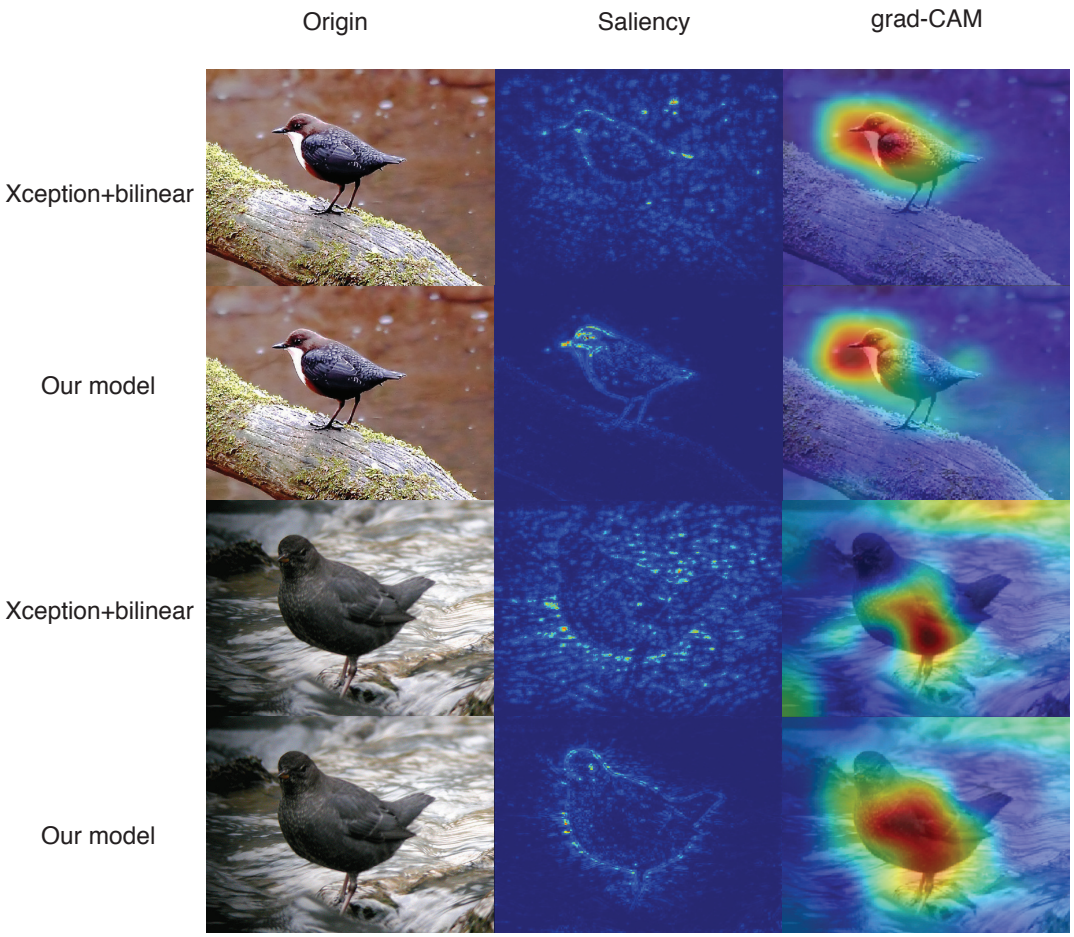

Figure 9. Representative results of our model and the Xception+Bilinear model.

4.2. System Accuracy

We compared our model with other state-of-the-art weakly supervised algorithms on the CUB200-20011. Compared with recurrent attention convolutional neural network (RA-CNN) [41], spatial transformer convolutional neural network (ST-CNN) [42], and picking deep filter responses (PDFR) [16], our model achieves a better performance. Besides achieving high classification accuracy on small-scale datasets such as CUB200-2011, our model classified the large-scale dataset Birds-1096 with an accuracy of 88.91%. Therefore, our model is both robust and generalizable, applicable to both small and large datasets. The accuracies of our model (Xception+Bilinear+Triplet), Xception+Bilinear, and Xception are shown in Table 2 and in the Acc plots below (Figure 10).

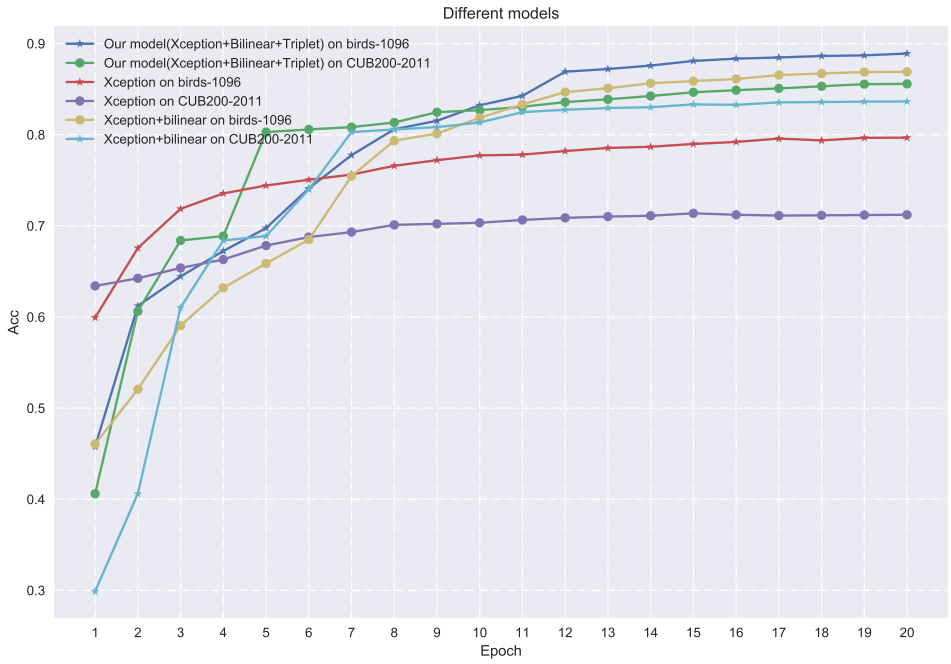

Figure 10. Acc curves of the evaluated different models trained on Birds-1096 and CUB200-2011.

Table 2. Accuracy comparisons of our model and other models. CNN: convolutional neural network.

Model	Dataset	Acc
Our model (Xception+Bilinear+Triplet)	Birds-1096	88.91%
Our model (Xception+Bilinear+Triplet)	CUB200-2011	85.58%
Xception+Bilinear	Birds-1096	86.91%
Xception+Bilinear	CUB200-2011	83.65%
Xception+Bilinear	CUB200-2011	83.65%
Xception	Birds-1096	79.68%
Xception	CUB200-2011	71.38%
RA-CNN (scale 1+2+3)	CUB200-2011	85.30%
ST-CNN (Incaption net)	CUB200-2011	84.10%
FDFR	CUB200-2011	82.60%

5. Discussion

In this paper, we propose a model based on a triplet network and bilinear methods for fine-grained bird recognition. Our proposed model was easily trained, which effectively increased the inter-class distance of the network extraction features and improved the accuracy of bird recognition. Compared with other weakly supervised methods such as RA-CNN, ST-CNN, and PDFR, our model achieved a better accuracy of 85.58% on the CUB200-2011. The proposed model not only used the label information of the species, but also constructed positive and negative sample pairs in the bird datasets, which exploited the unsupervised information. Experimental comparisons between our method and other existing approaches on two classification databases confirmed the superior classification accuracy of our approach. Moreover, our method also performed well on large-scale datasets. The model is generalizable, robust, and available for fine-grained image classification. In our future work, we will conduct the research on two directions. Firstly, we will find the solution on how to combine the the losses in different branches. Secondly, we will figure out how to integrate an attention mechanism with our model for more complex fine-grained categories.

Supplementary Materials: The following are available at http://www.mdpi.com/2076-3417/8/10/1906/s1, File S1.

Author Contributions: Conceptualization, Z.Z.; data curation, Z.Z.; formal analysis, Z.Z.; funding acquisition, Z.L.; investigation, J.L., K.W.; writing original draft, K.W.; review and editing, B.S.

Funding: This research was supported by the IT integrated service platform of Sichuan Wolong Natural Reserve(STS-Y82E01), The National R&D Infrastructure and Facility Development Program of China, "Fundamental Science Data Sharing Platform" (DKA2018-12-02-XX), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDA19060205), the Special Project of Informatization of Chinese Academy of Sciences (XXH13505-03-205), the Special Project of Informatization of Chinese Academy of Sciences (XXH13506-305), the Special Project of Informatization of Chinese Academy of Sciences (XXH13506-303), and Around Five Top Priorities of "One-Three-Five" Strategic Planning, CNIC(Grant No. CNIC_PY-1408 and Grant No. CNIC_PY-1409).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bosch, A.; Zisserman, A.; Muñoz, X. Scene Classification Using a Hybrid Generative/Discriminative Approach. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 712–727. [CrossRef] [PubMed]
2. Wu, J.; Rehg, J.M. CENTRIST: A Visual Descriptor for Scene Categorization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1489–1501. [CrossRef] [PubMed]
3. Gehler, P.; Nowozin, S. On feature combination for multiclass object classification. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 221–228. [CrossRef]
4. Jarrett, K.; Kavukcuoglu, K.; Ranzato, M.; LeCun, Y. What is the best multi-stage architecture for object recognition? In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 2146–2153. [CrossRef]
5. Wah, C.; Branson, S.; Welinder, P.; Perona, P.; Belongie, S. *The Caltech-UCSD Birds200-2011 Dataset*; California Institute of Technology: Pasadena, CA, USA, 2011.
6. Khosla, A.; Jayadevaprakash, N.; Yao, B.; Li, F.-F. Novel Dataset for Fine-Grained Image Categorization. In Proceedings of the First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, CO, USA, 20–25 June 2011.
7. Nilsback, M.; Zisserman, A. Automated Flower Classification over a Large Number of Classes. In Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing, Bhubaneswar, India, 16–19 December 2008; pp. 722–729. [CrossRef]
8. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436. [CrossRef] [PubMed]
9. Zhang, N.; Donahue, J.; Girshick, R.; Darrell, T. Part-Based R-CNNs for Fine-Grained Category Detection. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 834–849.
10. Branson, S.; Horn, G.V.; Belongie, S.J.; Perona, P. Bird Species Categorization Using Pose Normalized Deep Convolutional Nets. *arXiv* **2014**, arXiv:1406.2952.
11. Xiao, T.; Xu, Y.; Yang, K.; Zhang, J.; Peng, Y.; Zhang, Z. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, 7–12 June 2015; pp. 842–850. [CrossRef]
12. Simon, M.; Rodner, E. Neural Activation Constellations: Unsupervised Part Model Discovery with Convolutional Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, 7–13 December 2015; pp. 1143–1151. [CrossRef]
13. Chen, Y.; Li, J.; Xiao, H.; Jin, X.; Yan, S.; Feng, J. Dual Path Networks. *arXiv* **2017**, arXiv:1707.01629.
14. Lin, T.; Roy Chowdhury, A.; Maji, S. Bilinear CNN Models for Fine-Grained Visual Recognition. In Proceedings of the 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, 7–13 December 2015; pp. 1449–1457. [CrossRef]
15. Zhao, B.; Wu, X.; Feng, J.; Peng, Q.; Yan, S. Diversified Visual Attention Networks for Fine-Grained Object Classification. *IEEE Trans. Multimedia* **2017**, *19*, 1245–1256. [CrossRef]

16. Zhang, X.; Xiong, H.; Zhou, W.; Lin, W.; Tian, Q. Picking Deep Filter Responses for Fine-Grained Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 1134–1142. [CrossRef]
17. Zhang, Y.; Wei, X.; Wu, J.; Cai, J.; Lu, J.; Nguyen, V.A.; Do, M.N. Weakly Supervised Fine-Grained Categorization With Part-Based Image Representation. *IEEE Trans. Image Process.* **2016**, *25*, 1713–1725. [CrossRef] [PubMed]
18. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [CrossRef]
19. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807. [CrossRef]
20. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [CrossRef]
21. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. *arXiv* **2015**, arXiv:1512.00567.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. *arXiv* **2015**, arXiv:1512.03385.
23. Bengio, Y. Deep Learning of Representations for Unsupervised and Transfer Learning. In Proceedings of the ICML Workshop on Unsupervised and Transfer Learning, Bellevue, WA, USA, 27 June 2012; pp. 17–36.
24. Lin, T.; Roy Chowdhury, A.; Maji, S. Bilinear CNN Models for Fine-grained Visual Recognition. *arXiv* **2015**, arXiv:1504.07889.
25. Zheng, G.; Tan, M.; Yu, J.; Wu, Q.; Fan, J. Fine-grained image recognition via weakly supervised click data guided bilinear CNN model. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo, ICME 2017, Hong Kong, China, 10–14 July 2017; pp. 661–666. [CrossRef]
26. Hadsell, R.; Chopra, S.; LeCun, Y. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 539–546. [CrossRef]
27. Norouzi, M.; Fleet, D.J.; Salakhutdinov, R. Hamming Distance Metric Learning. In *Advances in Neural Information Processing Systems 25, Proceedings of the 26th Annual Conference on Neural Information Processing Systems 2012, Lake Tahoe, NV, USA, 3–6 December 2012*; MIT Press Ltd.: Cambridge, MA, USA, 2012; pp. 1070–1078.
28. Hoffer, E.; Ailon, N. Deep metric learning using Triplet network. *arXiv* **2014**, arXiv:1412.6622.
29. Lu, R.; Wu, K.; Duan, Z.; Zhang, C. Deep ranking: Triplet MatchNet for music metric learning. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, 5–9 March 2017; pp. 121–125. [CrossRef]
30. Lin, M.; Chen, Q.; Yan, S. Network In Network. *arXiv* **2013**, arXiv:1312.4400.
31. Zhou, B.; Khosla, A.; Lapedriza, À.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; pp. 2921–2929. [CrossRef]
32. Lin, T.; Maji, S. Improved Bilinear Pooling with CNNs. *arXiv* **2017**, arXiv:1707.06772.
33. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.S.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
34. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
35. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
36. Mandt, S.; Hoffman, M.D.; Blei, D.M. Stochastic Gradient Descent as Approximate Bayesian Inference. *arXiv* **2017**, arXiv:1704.04289.
37. Shore, J.E.; Johnson, R.W. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Trans. Inf. Theory* **1980**, *26*, 26–37. [CrossRef]

38. Selvaraju, R.R.; Das, A.; Vedantam, R.; Cogswell, M.; Parikh, D.; Batra, D. Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization. *arXiv* **2016**, arXiv:1610.02391.
39. Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv* **2013**, arXiv:1312.6034.
40. Chattopadhyay, A.; Sarkar, A.; Howlader, P.; Balasubramanian, V.N. Grad-CAM++: Generalized Gradient-based Visual Explanations for Deep Convolutional Networks. *arXiv* **2017**, arXiv:1710.11063.
41. Fu, J.; Zheng, H.; Mei, T. Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-Grained Image Recognition. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4476–4484. [CrossRef]
42. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial Transformer Networks. *arXiv* **2015**, arXiv:1506.02025.