Coarse-to-Fine Grained Classification

Yuqi Huo*
Yao Lu*
School of Information
Renmin University of China
Beijing 100872, China
bnhony@163.com

Yulei Niu
Zhiwu Lu†
School of Information
Renmin University of China
Beijing 100872, China
zhiwu.lu@gmail.com

Ji-Rong Wen
Beijing Key Lab. BDMAM
School of Information
Renmin University of China
Beijing 100872, China
jrwen@ruc.edu.cn

ABSTRACT

Fine-grained image classification and retrieval become topical in both computer vision and information retrieval. In real-life scenarios, fine-grained tasks tend to appear along with coarse-grained tasks when the observed object is coming closer. However, in previous works, the combination of fine-grained and coarse-grained tasks was often ignored. In this paper, we define a new problem called coarse-to-fine grained classification (C2FGC) which aims to recognize the classes of objects in multiple resolutions (from low to high). To solve this problem, we propose a novel Multi-linear Pooling with Hierarchy (MLPH) model. Specifically, we first design a multi-linear pooling module to include both trilinear and bilinear pooling, and then formulate the coarse-grained and fine-grained tasks within a unified framework. Experiments on two benchmark datasets show that our model achieves state-of-the-art results.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; • **Information systems** → *Clustering and classification.*

KEYWORDS

Fine-grained classification; coarse-grained classification; bilinear pooling; deep learning

ACM Reference Format:
Yuqi Huo, Yao Lu, Yulei Niu, Zhiwu Lu, and Ji-Rong Wen. 2019. Coarse-to-Fine Grained Classification. In *42nd Int'l ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR'19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331336

1 INTRODUCTION

Even with the development of recent image recognition techniques, fine-grained image classification and retrieval are still challenging [7]. Although it is now easy to distinguish whether an object is a cat or a dog, it is still hard to recognize the difference between Boeing 737 and Boeing 747, because the subtle variance owned by those subcategories could be easily overwhelmed by factors such as pose, viewpoint, and location of the object in the image.

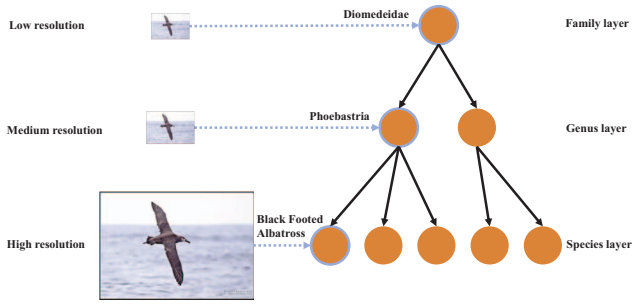

Figure 1: An illustration of the hierarchy structure in the birds dataset. When we have a low-resolution image of the bird, we can only decide which family it belongs to; but we can recognize the high-resolution image to species.

In real-life scenarios, fine-grained tasks tend to appear along with coarse-grained tasks. For example, imagine the Uber calling scenario that we are standing by the road and waiting for the called car coming. At first, the car is far away and we cannot recognize it well; maybe we can only guess it as Sedan or SUV (i.e. coarse-grained class). However, as it gets closer, we can see more details; finally, we recognize it as a Benz S-Class Sedan 2012 (i.e. fine-grained class). The fine-grained image classification task is involved in a coarse-to-fine classification process: objects are recognized as coarse-grained classes when they are in low-resolution, while they are recognized as fine-grained classes when in high-resolution.

Therefore, we define a new problem called coarse-to-fine grained classification (C2FGC) which aims to recognize the classes of objects in multiple resolutions (from low to high). For the first time, the coarse-grained and fine-grained classification are considered within a unified setting. Note that the C2FGC task actually fits itself into a hierarchy structure where higher layers represent finer-grained classes while lower layers represent coarser-grained classes. For example, the dataset of birds (Aves) [13] can be organized with a three-layer hierarchy of family, genus, and species. As shown in Figure 1, the species of a bird is black footed albatross, while it also belongs to Phoebastria in genus and Diomedeidae in family. Other datasets like Cars [6] can be fitted in a manually defined hierarchy structure with the make, model, type and year of cars. Although a number of previous works [2, 11] attempted to utilize the class hierarchy prior for fine-grained classification, they ignored the gradual information within different hierarchy layers, which is otherwise very important for solving the C2FGC task.

*Equal contribution.
†Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '19, July 21–25, 2019, Paris, France
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331336

Bilinear pooling has been shown to be effective in fine-grained recognition. As initially proposed in [7], bilinear pooling aims to model local parts of object using two-factor variations: it first learns two feature representations separately from an image; the obtained two feature representations, which are feature maps in convolutional neural network (CNN), are multiplied using the outer product at each location; finally, the outputs are averaged over all locations as the holistic representation of the image. It has many variations: [3] extends with a compact bilinear representation, and [1] defines a general kernel-based pooling framework which captures higher-order interactions of features. However, most existing bilinear pooling models only extract features from the last activation layer of CNN, which is insufficient for learning the local information of images. Although the latest work [14] extracts features from multiple layers of CNN, it only explores pairwise interactions (via bilinear pooling) among multiple layers.

In this work, we not only define a new coarse-to-fine grained classification problem, but also propose a Multi-linear Pooling (MLP) method to tackle the above drawbacks of bilinear pooling. More specifically, we first design a novel MLP module to include both bilinear (i.e. pairwise) and trilinear relationships among different layers for learning more discriminative representation of each image. We further integrate the proposed MLP module with a CNN model to solve the C2FGC problem. In this way, we formulate both coarse-grained and fine-grained tasks within a unified framework termed Multi-linear Pooling with Hierarchy (MLPH).

2 NEW PROBLEM DEFINITION

We define a new C2FGC problem to cope with the coarse-grained and fine-grained classification at the mean time. Our motivation is explained as follows. Firstly, the C2FGC task actually fits itself into a hierarchy structure where higher layers represent finer-grained classes while lower layers represent coarser-grained classes. As shown in Figure 1, the dataset of birds (Aves) [13] can be organized with a three-layer hierarchy of family, genus, and species. Secondly, if we can predict an object accurately to species, it is thus very easy to find its genus and its family. Meanwhile, if we can predict an object accurately to family, this coarse-grained information could definitely help in classifying genus and species as well. Thirdly, it is natural to combine images in a coarse-to-fine manner with an object from far away to nearby. When we are getting closer to the observed object, we can exploit more details and thus become more confident about what the object is.

Formally, given the original image dataset $\mathcal{I}$, we define three resolutions (from high to low) so that three image datasets $\mathcal{I}_{high}$, $\mathcal{I}_{medium}$, and $\mathcal{I}_{low}$ can be formed for the C2FGC task. Concretely, the three resolutions are defined as follows: we first take the resolution of the original images from $\mathcal{I}$ as the high-resolution (i.e., $\mathcal{I}_{high} = \mathcal{I}$), and classify the images from $\mathcal{I}_{high}$ to the species classes. We then gradually lower the resolution of the original image dataset $\mathcal{I}$ to obtain the other two resolutions. The accuracy of species classification would inevitably decline as the resolution goes down. When the accuracy drops below a threshold t_{med}, i.e., the classifier could not predict as accurately as in high-resolution, we define the resolution at that moment as r_{med} and the image dataset with r_{med} as $\mathcal{I}_{medium}$. After that we change the objective

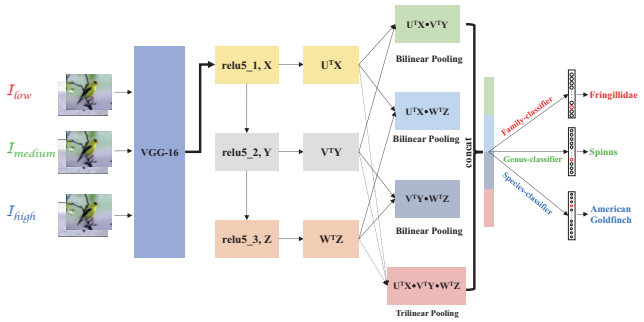

Figure 2: Overview of network architecture of Multi-linear Pooling with Hierarchy (MLPH) for coarse-to-fine grained classification. The input images in all three resolutions are resized to 448×448 ($\mathcal{I}_{low}$ and $\mathcal{I}_{medium}$ are still of lower quality than $\mathcal{I}_{high}$). Each of $[\mathcal{I}_{low}, \mathcal{I}_{medium}, \mathcal{I}_{high}]$ is processed by the same network with the only difference in the last fully-connected layers (highlighted in different colors).

to classifying images to the genera classes and repeat the same process. Finally, we can obtain r_{low} and $\mathcal{I}_{low}$ as well. In consequence, the three resolutions and their corresponding datasets could be determined by two hyperparameters: accuracy thresholds t_{med} and t_{low}. In this paper, we empirically set $t_{med} = 0.8$ and $t_{low} = 0.8$.

We further fit images in three resolutions to a taxonomy hierarchy structure. For example, the total 200 categories of species in [13] could be merged to 113 genera and to 36 families. The original classification task is assigned to $\mathcal{I}_{high}$ where images should be classified to 200 species, while $\mathcal{I}_{medium}$ and $\mathcal{I}_{low}$ are used for classification in 113 genera and 36 families, respectively.

3 THE PROPOSED FRAMEWORK

3.1 Multi-Linear Pooling

We introduce our Multi-Linear Pooling (MLP) module. In fine-grained classification, bilinear models were initially proposed by [12] to model two-factor variations, such as "style" and "content" for images. [7] adopts the generalized conception in features extracted from deep models such as VGG-16 [10]. Taking an image I as input and utilizing two feature functions f_A and f_B (usually the last layer of CNN) to extract the two features from the image, a bilinear vector output is obtained at each location using matrix outer product: the bilinear feature combination of $f_A(I) \in \mathbb{R}^{hw \times c}$ and $f_B(I) \in \mathbb{R}^{hw \times c}$ equals to $f_A(I)^{\mathrm{T}} f_B(I) \in \mathbb{R}^{c \times c}$, where c is the number of the feature maps while h and w represent the width and height of each feature map separately. Note that $h \times w$ needs to be fixed and c could vary from different choices of feature functions. In a number of existing bilinear models, f_A and f_B are usually identical feature extractors. Specifically, a deep descriptor of feature representation from the penultimate layer of CNN is defined as $\mathbf{x}_i \in \mathbb{R}^c$, where $1 < i < hw$. We then denote the descriptor matrix $X \in \mathbb{R}^{hw \times c}$, which collects all deep descriptors, i.e. $X = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{hw}]^{\mathrm{T}}$. Therefore, X can be regarded as the abstraction form of f_A and f_B.

A pooling function $\mathcal{P}$ is thus proposed to aggregate the bilinear features. Intuitively, the pooling layer could simply average all bilinear features. In mathematics, the average pooling of a product

of two identical matrices is given by:

$$G \coloneqq \frac{1}{hw}\sum_{i=1}^{hw}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}} = \frac{1}{hw}X^{\mathrm{T}}X, \tag{1}$$

where $\mathbb{R}^{c\times c}$ is the well-known Gram matrix. It can be clearly seen that the above bilinear form allows the outputs of the feature extractors f_A and f_B to be conditioned on each other by considering all their pairwise interactions in form of a quadratic kernel expansion. The resulting bilinear vector is then passed through a signed square root step, followed by an l_2 normalization layer (which leads to improvements in practice). A softmax function is finally used over all the classes for the classification task.

Factorized bilinear pooling (FBP) [5] aims to introduce an efficient attention mechanism of multimodal learning into the original bilinear pooling based on the Hadamard product. The full FBP model is defined as follows:

$$f = X^{\mathrm{T}}FX, \tag{2}$$

where $F \in \mathbb{R}^{hw\times hw}$ is a projection matrix, and f is the output of the bilinear model. According to the matrix factorization technique proposed in [8], the projection matrix F can be factorized into two one-rank vectors:

$$f = X^{\mathrm{T}}FX = P^{\mathrm{T}}(U^{\mathrm{T}}X \circ V^{\mathrm{T}}X), \tag{3}$$

where $U \in \mathbb{R}^{hw\times d}$ and $V \in \mathbb{R}^{hw\times d}$ are projection matrices, $P \in \mathbb{R}^{d\times cc}$ is the classification matrix, $\circ$ is the Hadamard product, and d denotes the dimension of joint embeddings.

Most existing bilinear models only extract features from the last activation layer of CNN (using two identical extractors), which is insufficient for learning the local/fine-grained information of images. Intuitively, a cross-layer bilinear pooling approach could capture finer-grained information of images. Therefore, we formulate the cross-layer factorized bilinear pooling as follows:

$$f = P^{\mathrm{T}}(U^{\mathrm{T}}X \circ V^{\mathrm{T}}Y), \tag{4}$$

where X represents one layer and Y represents another layer. Inspired by Eq. (4), we propose the trilinear pooling method, which comprises three features extracted from three different layers: X, Y, and Z. Instead of using the Hadamard product to combine only two layers, the Trilinear Pooling method is defined as:

$$f = P^{\mathrm{T}}(U^{\mathrm{T}}X \circ V^{\mathrm{T}}Y \circ W^{\mathrm{T}}Z), \tag{5}$$

where W is a projection matrix $\in \mathbb{R}^{hw\times d}$, and f combines the three separate layers (i.e. features).

Although [14] also extracts multiple bilinear features from multiple layers of a CNN, it only explores pairwise interaction over multiple layers. In this paper, we thus propose a Multi-Linear Pooling (MLP) model by concatenating multiple cross-layer bilinear pooling modules, together with a trilinear pooling module. The final output of the proposed MLP model is derived as:

$$\begin{aligned} f_{MLP} = P^{\mathrm{T}}concat(&U^{\mathrm{T}}X \circ V^{\mathrm{T}}Y, U^{\mathrm{T}}X \circ W^{\mathrm{T}}Z,\\ &V^{\mathrm{T}}Y \circ W^{\mathrm{T}}Z, U^{\mathrm{T}}X \circ V^{\mathrm{T}}Y \circ W^{\mathrm{T}}Z). \end{aligned} \tag{6}$$

Table 1: Statistics of the two benchmark datasets.

Datasets	Category	Training	Test
CUB-200-2011	200	5,994	5,794
Stanford Cars	196	8,144	8,041

Table 2: Hierarchy structure of the two datasets.

Datasets	Species	Genera	Family
CUB-200-2011	200	113	36
Stanford Cars	196	113	14

3.2 Network Architecture

By adopting MLP as the classifier, the whole coarse-to-fine classification model called Multi-linear Pooling with Hierarchy (MLPH) is illustrated in Figure 2. It can be observed that three image datasets in different resolutions are related to three different classification tasks, and VGG-16 is used as the deep feature extractor. Note that *relu5_1*, *relu5_2*, and *relu5_3* are utilized to define the three feature layers (i.e. X, Y, Z) because they can convey more part semantic information as compared with shallower layers.

Let the loss in each resolution be defined as: $\mathcal{L}_{high} = loss(\mathcal{I}_{high})$, $\mathcal{L}_{medium} = loss(\mathcal{I}_{medium})$, and $\mathcal{L}_{low} = loss(\mathcal{I}_{low})$. The full loss of the proposed MLPH model is formulated as:

$$\mathcal{L}_{full} = \mathcal{L}_{high} + \mathcal{L}_{medium} + \mathcal{L}_{low}. \tag{7}$$

4 EXPERIMENTS

4.1 Datasets and Settings

Datasets. We report results on two widely-used benchmark datasets, including Caltech-UCSD Birds (CUB-200-2011) [13] and Stanford Cars [6]. As shown in Table 1, the birds dataset contains 11,788 images of 200 bird species (i.e. fine-grained classes), while the cars dataset contains 16,185 images of 196 types of cars where fine-grained classes are defined at the level of (Make, Model, Year). Note that we only use fine-grained class labels in our experiments.

Hierarchy Structure. As shown in Table 2, the birds dataset is fitted in a hierarchy structure which consists of 200 species, 113 genera, and 36 families. Because the taxonomy does not exist within the cars dataset as it does in the birds dataset, the family and genus should be manually defined. Specifically, in the cars dataset, with the species being fine-grained class at the level of (Make, Model, Year), we define the family as the model of the species (e.g. a SUV or convertible), and then define the genus as the model combined with its brand (e.g. a Chrysler SUV or a BMW convertible). In consequence, there are totally 196 species, 113 genera, and 13 families in the hierarchy structure of the cars dataset.

Implementation Details. For fair comparison with other state-of-the-art approaches, our MLPH model adopts VGG-16 [10] pre-trained on the ImageNet dataset [9] as the backbone network. The size of input image is 448×448, which is common in other bilinear pooling methods. The parameters of the projection layers and the softmax layer are initialized randomly. We first train only the softmax layer while keep parameters of other layers fixed, and then fine-tune the whole network by using stochastic gradient descent with a batch size of 8, momentum of 0.9, weight decay of 5×10^{-4}, and a learning rate of 1×10^{-3} periodically annealed by 0.5. The dimension of projection layers is empirically set to 8,192. Note that

Table 3: Comparative accuracies (%) for coarse-to-fine grained classification on the two datasets. Note that the last two rows report the results of coarse-grained tasks.

Models	Anno.	CUB	Cars
Part-based R-CNNs [15]	Y	76.4	–
B-CNN [7]	Y	85.1	–
CB-CNN [3]	Y	84.6	–
Part-based R-CNNs [15]	N	73.9	–
B-CNN [7]	N	84.1	91.3
KP [1]	N	86.2	92.4
HBP [14]	N	86.8	93.6
MLP (ours, only $\mathcal{I}_{high}$)	N	87.0	93.8
MLPH (ours, high-reso.)	N	**87.3**	**94.0**
MLPH (ours, med-reso.)	N	91.7	94.1
MLPH (ours, low-reso.)	N	94.5	94.7

the three hierarchy layers are trained recurrently, e.g., first fine-tune the parameters in a 200-way softmax layer using a batch of images in $\mathcal{I}_{high}$, followed by a 113-way softmax layer using $\mathcal{I}_{medium}$ and a 36-way classifier in $\mathcal{I}_{low}$, and finally back to high again. The standard data augmentation methods are used, e.g., raw images are first resized to $512 \times S$ where S is the larger side, and then random sampling and horizontal flipping are performed during training (only center cropping is involved in test). The full model is trained in an end-to-end manner. All experiments are implemented with Caffe [4] on a PC platform with Titan Xp GPUs.

Evaluation Metrics. In the coarse-to-fine grained classification task, only the finest-grained classification accuracy is accounted for comparison with other fine-grained classification methods. Since $\mathcal{I}_{medium}$ and $\mathcal{I}_{low}$ only exist in our setting, we just report the results without comparison to the state-of-the-art.

4.2 Comparative Results

The comparative accuracies for coarse-to-fine classification on the two datasets are reported in Tables 3. Although the CUB dataset provides ground-truth annotations of bounding boxes and parts of birds, we only use the class labels in image level. Moreover, only when the fine-grained classification task is concerned, our MLPH model is compared to the state-of-the-art models.

We have the following observations: (1) Our full model achieves the best fine-grained classification results on both datasets, showing that the proposed Multi-linear Pooling with Hierarchy (MLPH) framework is indeed effective for solving the fine-grained classification task. (2) We reproduced the latest work [14] by directly using the released code. Our MLPH model is shown to outperform HBP [14] on both datasets. Given that HBP also includes three cross-layer bilinear models for fine-grained classification, this observation directly validates the effectiveness of the proposed Multi-linear Pooling with Hierarchy framework.

4.3 Ablation Study

Our full MLPH model has two simplified versions: (1) Without using the hierarchy structure, our full MLPH model degrades to the MLP model. (2) By discarding the trilinear pooling module, our MLP model degrades to the HBP model [14]. We compare the three models on both datasets in Tables 3. It can be seen that: (1) Our MLPH model leads to improvements over MLP, indicating that the coarse-grained and fine-grained tasks need to be coped with at the mean time. (2) Our MLP model outperforms HBP reproduced by us, which means that trilinear pooling can explore more part semantic information than bilinear pooling. Although the improvements achieved by our model are relatively small, this is still remarkable given that HBP has reported the best results so far.

5 CONCLUSION

In this work, we define a new problem called coarse-to-fine grained classification (C2FGC) which aims to recognize the classes of objects in multiple resolutions. To solve this new problem, we then propose a novel deep learning model termed Multi-linear Pooling with Hierarchy (MLPH). Experiments on two benchmark datasets show that our model achieves state-of-the-art results. In the ongoing work, we will try other bilinear models to obtain better results.

ACKNOWLEDGMENTS

This work was supported in part by NSFC (61573363 and 61832017), and the Outstanding Innovative Talents Cultivation Funded Programs 2018 of Renmin University of China.

REFERENCES

[1] Yin Cui, Feng Zhou, Jiang Wang, Xiao Liu, Yuanqing Lin, and Serge Belongie. 2017. Kernel pooling for convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 3049–3058.
[2] Jia Deng, Nan Ding, Yangqing Jia, Andrea Frome, Kevin Murphy, Samy Bengio, Yuan Li, Hartmut Neven, and Hartwig Adam. 2014. Large-scale object classification using label relation graphs. In *European Conference on Computer Vision*. Springer, 48–64.
[3] Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. 2016. Compact bilinear pooling. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 317–326.
[4] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*. ACM, 675–678.
[5] Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2016. Hadamard product for low-rank bilinear pooling. *arXiv preprint arXiv:1610.04325* (2016).
[6] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshops*. IEEE, 554–561.
[7] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. 2015. Bilinear CNN Models for Fine-grained Visual Recognition. In *International Conference on Computer Vision*. IEEE, 1449–1457.
[8] Steffen Rendle. 2010. Factorization machines. In *IEEE International Conference on Data Mining*. IEEE, 995–1000.
[9] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.
[10] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[11] Nitish Srivastava and Ruslan R Salakhutdinov. 2013. Discriminative transfer learning with tree-based priors. In *Advances in Neural Information Processing Systems*. 2094–2102.
[12] Joshua B Tenenbaum and William T Freeman. 2000. Separating style and content with bilinear models. *Neural computation* 12, 6 (2000), 1247–1283.
[13] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset. (2011).
[14] Chaojian Yu, Xinyi Zhao, Qi Zheng, Peng Zhang, and Xinge You. 2018. Hierarchical bilinear pooling for fine-grained visual recognition. In *European Conference on Computer Vision*. Springer, 595–610.
[15] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. 2014. Part-based R-CNNs for fine-grained category detection. In *European Conference on Computer Vision*. Springer, 834–849.