Deep Co-occurrence Feature Learning for Visual Object Recognition

Ya-Fang Shih*[1,2], Yang-Ming Yeh*[1,3], Yen-Yu Lin[1],
Ming-Fang Weng[4], Yi-Chang Lu[3], and Yung-Yu Chuang[2]

[1]*Research Center for Information Technology Innovation, Academia Sinica*
[2]*Department of Computer Science & Information Engineering, National Taiwan University*
[3]*Graduate Institute of Electronics Engineering, National Taiwan University*
[4]*Smart Network System Institute, Institute for Information Industry*

Abstract

This paper addresses three issues in integrating part-based representations into convolutional neural networks (CNNs) for object recognition. First, most part-based models rely on a few pre-specified object parts. However, the optimal object parts for recognition often vary from category to category. Second, acquiring training data with part-level annotation is labor-intensive. Third, modeling spatial relationships between parts in CNNs often involves an exhaustive search of part templates over multiple network streams. We tackle the three issues by introducing a new network layer, called co-occurrence layer. *It can extend a convolutional layer to encode the* co-occurrence *between the visual parts detected by the numerous neurons, instead of a few pre-specified parts. To this end, the feature maps serve as both filters and images, and* mutual correlation filtering *is conducted between them. The co-occurrence layer is end-to-end trainable. The resultant co-occurrence features are rotation- and translation-invariant, and are robust to object deformation. By applying this new layer to the VGG-16 and ResNet-152, we achieve the recognition rates of* 83.6% *and* 85.8% *on the Caltech-UCSD bird benchmark, respectively. The source code is available at https://github.com/yafangshih/Deep-COOC.*

1. Introduction

Fine-grained recognition aims to identify finer-level categories in images, *e.g.* different bird species [33], dog breeds [19], and aircraft types [25]. In addition to the difficulties inherent in generic object recognition such as large intra-class variations and a large number of categories to be identified, fine-grained recognition is even more challenging due to subtle inter-class differences. *Convolutional neural networks* (CNNs) have demonstrated the effectiveness in joint visual feature extraction and nonlinear classifier learning. Recent CNN-based approaches to fine-grained recognition, *e.g.* [23, 29, 40], have been shown to significantly outperform the conventional approaches [1, 14] that work with engineered features. Despite the encouraging progress, fine-grained recognition still remains challenging, and better solutions are in demand.

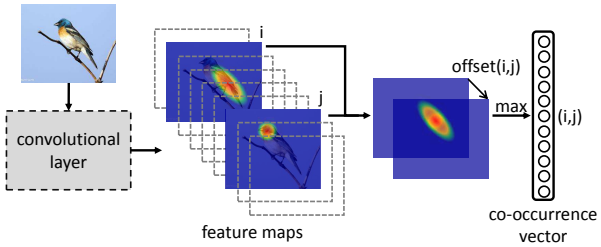


Figure 1. The proposed co-occurrence layer. An image passing a convolutional layer is represented by a set of feature maps. Mutual convolution is conducted between these feature maps. A vector recording the co-occurrence of visual parts is generated.

Part-based models, *e.g.* [6, 7], recognize objects by considering the appearances of individual object parts as well as their spatial relationships. They are robust to intra-class variations caused by different poses, object deformations, partial occlusions, and so on. Recent studies [13, 41] showed that integrating part-based models into convolutional neural networks leads to remarkable performance gains in both generic and fine-grained object recognition. Unfortunately, there exist a few issues hindering the advances of part-based models. First, most part-based models rely on a fixed number of pre-specified object parts. However, the optimal parts for recognition typically vary from category to category, and are generally unknown in ad-

*indicates equal contribution.

vance. Second, part-level labeling in training images leads to much more expensive cost than conventional image-level annotation. Third, modeling the relationships between object parts in CNNs often requires exhaustive search of part templates [34] or needs multiple network streams [35, 40].

In this work, a new network layer called *co-occurrence layer* is proposed to address the three aforementioned issues. It generalizes a convolutional layer in CNNs to discover the *co-occurrence* between visual patterns in an image. As pointed out in [39, 44], convolution filters in CNNs can be considered detectors for specific patterns, or object parts in object recognition. The feature map of such a filter records the spatial occurrence likelihood of the corresponding part in the image. Conducting correlation analysis between two feature maps reveals the degree of co-occurrence of the two corresponding parts. Figure 1 illustrates how the co-occurrence layer works. An image passing a convolutional layer is represented by N feature maps. For a pair of feature maps i and j, we treat one of them as the image, and the other as a filter. Then correlation filtering is performed, and the maximal correlation response during filtering is recorded. By repeating the procedure for every pair of the feature maps, a *co-occurrence* vector of dimension N^2 is yielded by gathering all the maximal responses. The vector can serve as the input to the last fully-connected layer, and enhance the performance of recognition.

The resultant co-occurrence vector encodes the co-occurrence between the numerous parts detected by the neurons. Thus, the co-occurrence layer integrates part-based information into CNNs. It requires neither pre-defined object parts nor part-level annotation. It involves only filtering and max operations, so it is simple to implement. The co-occurrence layer is rotation- and translation-invariant, and is robust to deformation. Unlike the nonlinearity given by an activation function in a neuron of CNNs, the resultant co-occurrence vector captures the nonlinear properties across neurons of the same convolutional layer.

The proposed co-occurrence layer is differentiable and therefore supports end-to-end training. Previous studies, *e.g*. [44], have shown that the neurons in different convolutional layers extract object parts in a coarse-to-fine manner. The co-occurrence layer is general in the sense that it can serve as a building block, and generalize an arbitrary subset of the convolutional layers in a CNN model. The generalized network is a *directed acyclic graph* (DAG), so conventional back-propagation algorithms can be applied to network optimization. In the experiments, we illustrate the proposed co-occurrence layer by applying it to the last three convolutional layers of two widely-used CNN frameworks, VGG-16 [28] and ResNet-152 [17]. The recognition rates on the *Caltech-UCSD bird benchmark* (CUB200-2011) [33] are significantly improved from 70.4% to 83.6% with VGG-16 and from 73.3% to 85.8% with ResNet-152, respectively.

2. Related Work

We review a few relevant topics in this section.

2.1. Part-based methods for object recognition

One major difficulty hindering the advance of accurate object recognition is large intra-class variations, which result from both intrinsic and extrinsic issues, such as different object poses, instance diversity, partial occlusions and cluttered background. Part-based methods are introduced to address these variations by representing an object as an assembly of local parts and their spatial layout.

The *constellation models* [9, 36] represent an object as a fully connected constellation. Namely, the pairwise locations of object local parts are geometrically constrained. The shape, appearance, and scale of parts are jointly formulated by probabilistic density functions. Despite the effectiveness, the computation cost of inferring objects in an image grows exponentially with respect to the number of parts. The *pictorial models* [7, 11] instead represent objects as a collection of parts in a tree-structured deformable configuration. Object part inference is carried out by energy minimization in polynomial time with the Viterbi algorithm.

Inspired by the pictorial model, the *deformable part model* (DPM) [8] uses the *histogram of oriented gradients* (HOG) [4] for describing the object *root* and *parts*. It localizes objects by computing an appearance cost based the detected root and parts as well as a deformation cost based on the part locations relative to the root. In [6], a mixture model was later introduced to cover objects of multiple scales and viewpoints. DPM is widely adopted in many applications, such as pose estimation [38, 45] and object detection [13, 34]. For fine-grained recognition, *deformable part descriptor* (DPD) [42] leverages DPM to obtain pose-normalized features, which enhance fine-grained recognition. However, these approaches rely on handcrafted features. Their performance can be considerably improved by using learned features via CNNs.

2.2. CNNs with part-based representations

Recent studies *e.g*. [13, 34] consider DPM and CNNs complementary to each other in the sense that the former employs a graph structure for modeling the spatial layout of parts, while the latter learns more discriminative features for part description. Girshick *et al*. [13] introduced a new network layer called *distance transform pooling*, and showed that DPM can be formulated as an equivalent CNN model. Wan *et al*. [34] presented a system that integrates CNNs, DPM, and non-maximum suppression, and applied it to part-based object detection. The tolerable shift between parts is pre-specified in their work, which might makes these models less robust to large deformation. In light of that part-based representations are robust to partial occlusions, Tian *et al*. [31] proposed *DeepParts*, consisting of

multiple ConvNet part detectors, and alleviated the problem caused by partial occlusions in pedestrian detection. These methods, however, rely on an exhaustive search of multiple part or viewpoint templates, or require multiple CNN models. In addition, the number of parts are manually given, but the optimal number is typically unknown in advance.

2.3. Fine-grained recognition

Compared to generic object recognition, fine-grained recognition highly relies on discriminative parts, which capture subtle inter-class differences. Thus, many approaches to fine-grained recognition, *e.g*. [1, 2, 3, 14, 22, 41], work with training data with part-level annotation. Part localization becomes an inherent component of these approaches in the testing stage. For example, part-based R-CNN [41] adopts R-CNN [12] to detect parts, and constrains the part locations within a distance to the object root. In [40], an extra detection network for part localization is combined with the recognition network for fine-grained categorization. However, labeling object parts in training data collection is labor-intensive in these approaches.

Recent success in fine-grained recognition often makes use of CNN models with multiple network streams. In [37], an extra network is employed to filter out the non-object patches so that the other network can focus on the regions of objects for bird species recognition. A framework with multiple CNN models is trained in [35] to learn the multiple granularity descriptors. Lin *et al*. [23] extracted bilinear, orderless features by using two-stream CNNs. In [18], the ST-CNN framework containing a localisation network and two Inception models [30] achieved the state-of-the-art result on CUB200-2011. However, the models in these approaches are complicated. More training data and longer training time are required.

In FlowNet [10], patch-wise correlation over *all* feature maps is conducted for flow prediction. In contrast, we take out every pair of feature maps, and evaluate the co-occurrence of the two corresponding patterns over the whole maps.

Our approach generates co-occurrence features between object parts based on CNNs. It distinguishes itself from previous work by having the following three appealing properties simultaneously. First, it encodes the relationships between numerous parts detected by neurons, instead of a small number of pre-specified parts. Second, it does not require any part-level annotation. Third, our co-occurrence layer almost introduces no extra parameters. It produces co-occurrence features based on a single-stream network.

3. The proposed approach

Our approach is introduced in this section. Given a target CNN architecture, our goal is to associate a subset of or even the whole convolutional layers with the *co-occurrence layers* so that the co-occurrence properties between object parts can be leveraged to enhance the performance of fine-grained recognition. For a quick overview, how the co-occurrence layer works with a convolutional layer has been illustrated in Figure 1, and the resultant network architecture is shown in Figure 3. In the following, the forward pass and backward propagation of the proposed co-occurrence layer are firstly described. Then, we show the coupled convolutional and co-occurrence layers can serve as the building block for network construction. Finally, the implementation details of our approach are given.

3.1. Co-occurrence layer: Forward pass

Consider a convolutional layer that employs N convolution filters and maps the input to a set of N feature maps. It can be formulated by

$$A^i = \sigma\left(W_i * X + b_i\right), \text{ for } i = 1, ..., N \tag{1}$$

where X is the input, W_i and b_i are the learnable weight matrix and bias of the ith filter respectively, and $*$ is the convolution operator. σ denotes the activation function. ReLU [21] is adopted in this work, *i.e*. $\sigma(Z) = \max(\mathbf{0}, Z)$. $A^i \in \mathbb{R}^{m \times m}$ is the ith feature map of size $m \times m$.

The idea behind the co-occurrence layer is simple: Feature map A^i in Eq. (1) records the spatial occurrence likelihood of the visual part detected by the ith filter. Conducting spatial correlation between a pair of feature maps reveals the extent that the corresponding parts jointly occur. Specifically for a pair of feature maps $A^i \in \mathbb{R}^{m \times m}$ and $A^j \in \mathbb{R}^{m \times m}$, we respectively treat them as a filter and an image, and perform correlation filtering. We seek the maximal correlation response c_{ij} as well as the spatial offset $\mathbf{o}_{ij} = [o_{ij,x}, o_{ij,y}]^\top \in \mathbb{R}^2$, *i.e*.

$$c_{ij} = \max_{\mathbf{o}_{ij}} \sum_{\mathbf{p} \in [1,m] \times [1,m]} A^i_{\mathbf{p}} A^j_{\mathbf{p}+\mathbf{o}_{ij}}, \tag{2}$$

where $A^i_{\mathbf{p}}$ is the element of A^i at location $\mathbf{p}$. $A^j_{\mathbf{p}+\mathbf{o}_{ij}}$ is similarly defined. Note that zero-padding is performed before filtering. The maximal response c_{ij} can be interpreted as the *degree of co-occurrence* between the object parts detected respectively by the ith and the jth filters. After repeating the procedure of mutual correlation for each feature map pair, and the *co-occurrence vector* $\mathbf{c} = [c_{ij}] \in \mathbb{R}^{N^2}$ is generated, which will be used for recognition via a classifier, such as a fully-connected layer in this work.

Discussion. Part-based representations are valuable for fine-grained recognition [35, 43], since discriminative features for subordinate categorization are often enveloped in object parts, instead of the whole objects. Despite the effectiveness, most part-based models work with only a few pre-specified parts. Besides, the optimal parts for recognition

are generally unknown in advance. The co-occurrence layer addresses these unfavorable issues simultaneously. It discovers the co-occurrence between the visual parts detected by numerous neurons, instead of those pre-specified parts. Therefore, any part-level annotation is not required. On the other hand, these neurons will be optimized for object part detection, since the loss function is a function of the generated co-occurrence features.

The co-occurrence vector $\mathbf{c}$ is generated by seeking the maximal correlation via Eq. (2). It can be observed that the vector $\mathbf{c}$ is rotation-, mirror-reflection, and translation-invariant, and is robust to object deformation. These properties are desirable for object recognition. In Eq. (2), each element in the co-occurrence vector is in form of the inner product between two overlapping feature maps produced their own neurons. Hence, the co-occurrence vector captures the nonlinear properties across neurons. It is complementary to the nonlinearity given by an activation function in an individual neuron of CNNs.

3.2. Co-occurrence layer: Backward Propagation

The co-occurrence layer does not introduce additional learnable parameters. It takes the feature maps $A = \{A^i\}_{i=1}^N$ as inputs, and compiles the co-occurrence vector $\mathbf{c}$. While the maps A are parametrized by the learnable matrices and biases $W = \{W_i, b_i\}_{i=1}^N$ of the convolutional layer, the vector $\mathbf{c}$ is an input to the objective function l as shown in Figure 3. The dependence relationships among W, A, $\mathbf{c}$, and l are summarized in Figure 2. The objective l for network learning is set to maximize the multinomial logistic regression in this work.

In the following, we show that the network with the integration of the co-occurrence layers can be still learned by stochastic gradient descent. To this end, the gradient of the objective function with respect to the parameters, *i.e.* $\frac{\partial l}{\partial W}$, is required. By applying the chain rule, we have $\frac{\partial l}{\partial W} = \frac{\partial l}{\partial \mathbf{c}} \frac{\partial \mathbf{c}}{\partial A} \frac{\partial A}{\partial W}$. The last term $\frac{\partial A}{\partial W}$ has been derived in the literature of CNNs, so we focus on the derivation of $\frac{\partial l}{\partial A} = \frac{\partial l}{\partial \mathbf{c}} \frac{\partial \mathbf{c}}{\partial A}$. Its element-wise calculation is given below:

$$\frac{\partial l}{\partial A^k_{\mathbf{p}}} = \sum_{i,j=1}^{N} \frac{\partial l}{\partial c_{ij}} \frac{\partial c_{ij}}{\partial A^k_{\mathbf{p}}} \tag{3}$$

$$= \sum_{i,j=1}^{N} \frac{\partial l}{\partial c_{ij}} \frac{\partial \sum_{\mathbf{q}\in[1,m]\times[1,m]} A^i_{\mathbf{q}} A^j_{\mathbf{q}+\mathbf{o}_{ij}}}{\partial A^k_{\mathbf{p}}} \tag{4}$$

$$= \sum_{j=1}^{N} \frac{\partial l}{\partial c_{kj}} A^j_{\mathbf{p}+\mathbf{o}_{kj}} + \sum_{i=1}^{N} \frac{\partial l}{\partial c_{ik}} A^i_{\mathbf{p}-\mathbf{o}_{ik}}, \tag{5}$$

where the partial derivatives, $\frac{\partial l}{\partial c_{kj}}$ and $\frac{\partial l}{\partial c_{ik}}$, can be computed via back-propagation.

To have a compact representation, we create two auxiliary variables $\mathbf{u}^k_{\mathbf{p}} \in \mathbb{R}^N$ and $\mathbf{v}^k_{\mathbf{p}} \in \mathbb{R}^N$, and define them as follows:

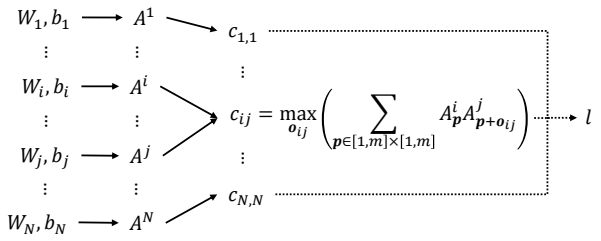


Figure 2. The dependence relationships among the parameters W of the convolutional layer, the feature maps A, the co-occurrence vector $\mathbf{c}$, and the objective function l.

$$\mathbf{u}^k_{\mathbf{p}} = \begin{bmatrix} A^1_{\mathbf{p}+\mathbf{o}_{k1}} \\ A^2_{\mathbf{p}+\mathbf{o}_{k2}} \\ \vdots \\ A^N_{\mathbf{p}+\mathbf{o}_{kN}} \end{bmatrix} \text{ and } \mathbf{v}^k_{\mathbf{p}} = \begin{bmatrix} A^1_{\mathbf{p}-\mathbf{o}_{1k}} \\ A^2_{\mathbf{p}-\mathbf{o}_{2k}} \\ \vdots \\ A^N_{\mathbf{p}-\mathbf{o}_{Nk}} \end{bmatrix}. \tag{6}$$

A third-order tensor $U^k \in \mathbb{R}^{m\times m\times N}$ is constructed by concatenating $\mathbf{u}^k_{\mathbf{p}}$ for every location $\mathbf{p}$ along its first two dimensions. Similarly, we have $V^k \in \mathbb{R}^{m\times m\times N}$ by concatenating $\mathbf{v}^k_{\mathbf{p}}$ for every location $\mathbf{p}$.

The matrix form of Eq. (5) can be expressed as

$$\frac{\partial l}{\partial A^k} = U^k \times_3 \tilde{\mathbf{c}}_k + V^k \times_3 \hat{\mathbf{c}}_k, \tag{7}$$

where operator $\times_3$ is the 3-mode product, and $\tilde{\mathbf{c}}_k = [\frac{\partial l}{\partial c_{k1}}, \ldots, \frac{\partial l}{\partial c_{kN}}]$ and $\hat{\mathbf{c}}_k = [\frac{\partial l}{\partial c_{1k}}, \ldots, \frac{\partial l}{\partial c_{Nk}}]$ are two row vectors.

The gradient $\frac{\partial l}{\partial A}$ is attainable by computing $\{\frac{\partial l}{\partial A^k}\}_{k=1}^N$ via Eq. (7). It follows that the network with the integration of the co-occurrence layers remains end-to-end trainable. We train the network by using stochastic gradient descent with momentum in the experiments.

3.3. Generalization

The philosophy of *deeper-is-better* has been adopted in many powerful CNN frameworks, such as VGG-16 [28] and ResNet [15]. Recent studies, *e.g.* [44], showed that the earlier convolutional layers in deep CNN models tend to detect low-level patterns, such as edge-like and blob-like features, while the later layers tend to detect high-level patterns, such as object parts.

The proposed co-occurrence layer encodes the co-occurrence properties between patterns detected in a particular layer. To generalize it, we use the coupled convolutional and co-occurrence layers as a *building block* to construct the network. In this way, co-occurrence features between not only high-level but also low-level visual patterns can be extracted, and further facilitate recognition. As

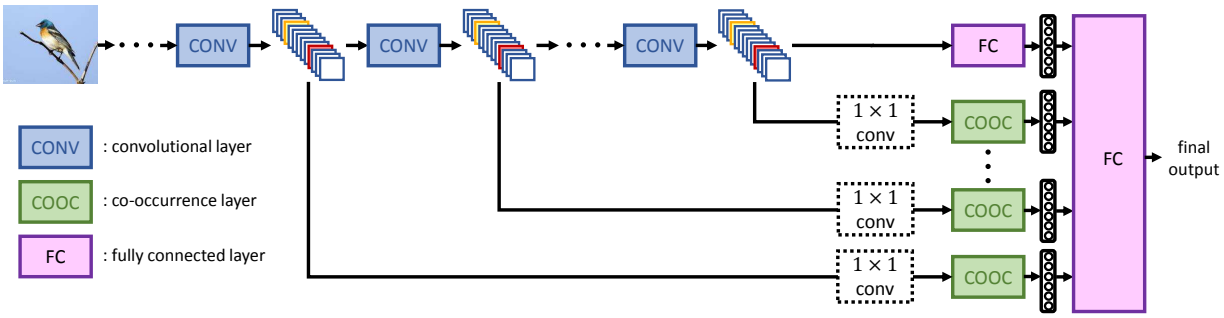


Figure 3. The network architecture. The coupled convolutional and co-occurrence layer serve as the building block for network construction. The co-occurrence vectors coming form different layers are fused via a fully connected layer for prediction making.

shown in Figure 3, the resultant network concatenates co-occurrence vectors by using a fully-connected layer. The architecture is a directed acyclic graph. Therefore, the back-propagation algorithm is applicable to the computation of gradients with respect to all the parameters.

3.4. Implementation details

The co-occurrence layer works on the feature maps generated by a convolutional layer. In our implementation, a 1×1 convolutional layer and Gaussian filtering are applied to pre-process the feature maps. The former reduces the number of feature maps, while the latter suppresses noise. The resultant co-occurrence vector is further post-processed by passing it through signed square root and ℓ_2 normalization for a more stable and better performance.

Feature maps reduction. The co-occurrence layer estimates the degree of co-occurrence between every feature map pair. In practice, most feature map pairs have very weak correlation. Most elements of the co-occurrence vectors have no contribution to information. As a result, we use 1×1 convolution filters [30] to reduce the number of feature maps. This operation adds the discriminative power to the feature maps after filtering, since less correlated feature maps are more likely to be removed by assigning lower weights. More importantly, it greatly reduces the computational cost of the co-occurrence layers, because the dimension of the co-occurrence vector is quadratic to the number of feature maps.

Noise suppression. In practice, noisy responses are often present in the feature maps. The proposed co-occurrence layers take the feature maps as inputs, and may suffer from the noisy responses when performing mutual correlation filtering between feature maps. To address this issue, Gaussian filtering is applied to feature maps before performing mutual correlation filtering between them. Gaussian filtering keeps the strong responses which tend to encode object parts, while suppressing weak responses which are more likely to be caused by noise. This process is simple but effective because it remarkably improves the quality of the generated co-occurrence vectors in recognition.

4. Experimental results

The performance of the proposed co-occurrence layer is evaluated in this section. We first describe the adopted dataset, CUB200-2011 [33], for performance measure, and give the details about the construction and initialization of our network. Then, the quantitative results are reported and analyzed, including the performances of applying the co-occurrence layer to the last few convolutional layers individually or jointly, and the comparison of our approach to the existing approaches. Finally, we present a scheme for visualizing the learned co-occurrence features, and show that these features tend to detect object parts that are distinctive and occur jointly.

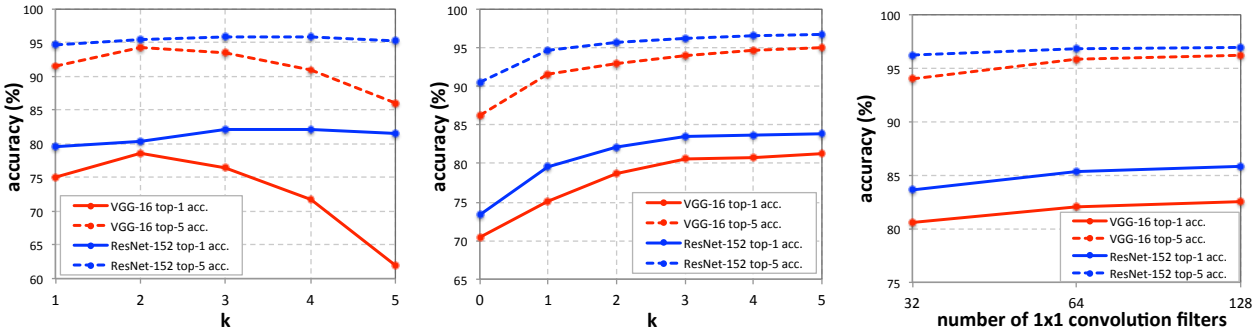
4.1. The CUB200-2011 dataset

The dataset is composed of images from 200 species of birds, and is considered quite challenging due to its large intra-class variations and small inter-class differences. It contains about 60 images for each of the 200 species of birds, including 30 for training and 30 for testing. The number of the images is 11,788 in total. As the sizes of these images are different, we resize them to a resolution of 448×448 before using them to train or test the network.

4.2. Experimental setup

In our experiments, we take VGG-16 [28] and ResNet-152 [17] models pretrained on ImageNet [5], and replace the last fully-connected layer of each model by another fully-connected layer with 200 output units for classifying data in the CUB200-2011 dataset. The abundant data in ImageNet help a lot in initializing deep CNN models especially when the domain specific fine-grained datasets have no sufficient training images.

As mentioned in [44], the later convolutional layers tend to extract high-level concepts of objects. We apply our co-occurrence layers to the last K convolutional layers of VGG-16 and the last K building blocks of ResNet-152. In the experiments, we will show that only the co-occurrence vectors from the last few layers are helpful in improving the



(a) (b) (c)

Figure 4. (a) Recognition rates when applying the co-occurrence layer to the kth to the last convolutional layer. (b) Recognition rates when applying the co-occurrence layers to the last k convolutional layers. (c) Recognition rates with different numbers of 1×1 filters.

recognition rate. Hence, K is set up to 5 here.

As shown in Figure 3, one additional 1×1 convolutional layer is connected with each of the chosen convolutional layers in order to remove insignificant feature maps and also reduce the number of feature maps from 512 to M where M is set to 32, 64, and 128, respectively. We adopt a two-step training procedure introduced in [2].

We initialize the 1×1 convolutional layers and the last fully-connected layer by the way described in [16], and train the network by using multinomial logistic regression objective. After the training procedure converges, we fine-tune all layers after the last K convolutional layers in an end-to-end fashion. This work is implemented based on the open source framework `MatConvNet` [32].

Once fine-tuning is accomplished, the extracted features, *i.e*. the input to the last fully-connected layer, are power-normalized [26], and used to train one-vs-all linear SVMs. On this dataset, the SVMs classifier gives slightly higher recognition rates for VGG-16, while the softmax layer is better for ResNet-152. Thus, we adopt this setting to report the recognition rates.

4.3. Comparison to the baseline

Although the co-occurrence layers can be applied to an arbitrary subset of convolutional layers, how to select a proper subset is unclear. The top-1 accuracy on CUB200-2011 dataset is 70.4% for VGG-16 and 73.3% for ResNet-152. We consider them the baselines, and conduct a set of experiments to find an appropriate set of convolutional layers to be generalized by the co-occurrence layers.

We set the number of filters in each 1×1 convolutional layer as 32, which is the number of feature maps taken by a co-occurrence layer. We apply the co-occurrence layer to the kth to the last convolutional layer for VGG-16 and to the last building block for ResNet-152. The resultant co-occurrence vector is used for classification. By varying the value of k from 1 to 5, the recognition rates are reported in Figure 4(a).

By using the VGG-16 model, the co-occurrence vectors from the last three convolutional layers respectively give the recognition rates of 75.1%, 78.7%, and 76.4%, which are remarkably higher than the baseline. It is also worth mentioning that the dimension of each co-occurrence vector is 1,024, which is much lower than 4,096, the feature dimension in VGG-16. However, the accuracy of using co-occurrence vectors from the fourth and the fifth to last layers drops dramatically. The results imply that compared to the VGG-16 model, only co-occurrence vectors from last few layers are effective. When using ResNet-152, the co-occurrence vectors from the last five building blocks achieve similar accuracy ranging from 79.5% to 82.2%. It implies that all these building blocks capture high-level concepts, which is not very surprising in such a deep network.

We investigate if the co-occurrence vectors from different layers are complementary. To this end, we apply co-occurrence layers to the last k convolutional layers or building blocks jointly, and represent data by concatenating the learned co-occurrence vectors. By varying the value of k from 1 to 5, the recognition rates are shown in Figure 4(b). As can be seen, the accuracy converges rapidly when $k = 3$ for both VGG-16 and ResNet-152. The results confirm that these co-occurrence vectors are complementary, and combining them leads to a remarkable performance gain.

The number of filters in the 1×1 convolutional layer is set with the trade-off between accuracy and efficiency. Figure 4(c) reports the recognition accuracy, when the number of 1×1 convolution filters is set to 32, 64, and 128, respectively. The results show that using more 1×1 filters gives better performance, but the improvement is minor by doubling the filter number from 64 to 128.

4.4. Comparison to previous work

Inspired by the results shown in Figure 4, we apply the co-occurrence layers to the last three convolutional layers of VGG-16 and the last three building blocks of ResNet-152, and set the filter number in each 1×1 convolutional layer to 128. The resultant networks reach the recognition rates of 82.6% and 85.8%, respectively. By concatenating

method	network	part annotation	acc. (%)
Berg *et al*. [1]	-	✓	56.9
Göering *et al*. [14]	-	✓	57.8
Chai *et al*. [3]	-	✓	59.4
Zhang *et al*. [42]	-	✓	64.9
Liu *et al*. [24]	Caffe		73.5
Zhang *et al*. [41]	Caffe	✓	73.9
Branson *et al*. [2]	Caffe	✓	75.7
Simon *et al*. [27]	VGG		81.0
Krause *et al*. [20]	VGG		82.0
Ours	VGG		83.6
Ours	ResNet-152		**85.8**
Xiao *et al*. [37]	AlexNet+VGG		77.9
Wang *et al*. [35]	VGG×3		81.7
Lin *et al*. [23]	AlexNet+VGG		84.1
Jaderberg *et al*. [18]	Inception×4		84.1

Table 1. Accuracy of various approaches on CUB200-2011.

the co-occurrence vectors with the 4096-d feature vector learned by VGG-16, the accuracy of VGG-16 model is further boosted to 83.6%.

Table 1 reports the recognition rates of our approach and the competing approaches on the CUB200-2011 dataset. The competing approaches, including [1, 14, 3, 42], perform fine-grained recognition based on hand-crafted features. Though these approaches were developed with the theoretic merit, they can be surpassed by modern CNN-based methods where feature learning and the classifier training are carried out jointly.

Some CNN-based competing methods [37, 35, 23, 18] were developed based on multi-stream networks. The sizes of their models are larger. More training data are required to tune the parameters. Some competing methods [41, 2] rely on training data with part-level annotation, leading to an expensive cost in training data collection. In contrast, our approach can generate co-occurrence features between object parts on a single-stream network. It does not require any part-level annotation, and can greatly improve the performance of fine-grained recognition.

Our approach, with the recognition rate of 85.8% based on ResNet-152, outperforms all the competing approaches. Its accuracy based on VGG-16, 83.6%, is comparable to the state-of-the-art approaches [23, 18], and our method is with the advantage of having a small model, since the proposed co-occurrence layer itself does not introduce extra parameters. The approach in [23] employs a two-stream network, where the number of learnable parameters is also doubled. The approach in [18], established on CNNs with four *Inception models*, is more complicated than our approach.

The total dimension of the co-occurrence vectors is LM^2 where $L = 3$ is the number of the co-occurrence layers and $M = \{32, 64, 128\}$ is the number of 1×1 convolution filters for feature map reduction. Table 2 reports the feature dimensions and the accuracy of VGG-16, *bilinear CNN* (BCNN) [23], and ours with three different numbers of 1×1 filters. Note that both the accuracy rates of applying the co-occurrence layers to VGG-16 and ResNet-152 are given, and the accuracy rates are reported without concatenating the feature vectors by VGG-16 and ResNet-152. In Table 2, our method with 32 1×1 filters achieves much better performance than VGG-16 with even lower feature dimensions. Compared to BCNN, our method based on ResNet-152 with 128 1×1 filters gives better recognition rate and with much lower dimensions.

method	Ours			VGG-16	BCNN
# of 1×1 filters	32	64	128		
dimension	3k	12.3k	49.2k	4.1k	262.1k
accuracy (%)	80.6	82.0	82.6	70.4	84.1
	83.6	85.3	**85.8**		

Table 2. Feature dimensions and accuracy of three methods. Two accuracy rates are reported for our approach: the top one is based on VGG-16 and the bottom one is based on ResNet-152.

4.5. Visualization

To gain insight into the quantitative results, we investigate how the co-occurrence layer guides fine-grained recognition by highlighting the co-occurring regions detected by the co-occurrence features.

As shown in Figure 3, the last fully connected layer in the network maps all the co-occurrence vectors to the final output. This layer maintains a weight for every co-occurrence feature and category combination. For a target category, we focus on the co-occurrence feature with the highest weight for that category, since it is the most influential feature for images predicted as that category. The visualization of this co-occurrence feature can be carried out through an image of that category. For the image, we take its two feature maps corresponding to this co-occurrence feature, and perform hard thresholding to exclude the regions with low activation responses. By up-sampling the two resultant *heatmaps* to the size of the input image, the co-occurring regions are highlighted.

Figure 5 displays the most *influential* co-occurrence features for five bird species, including `parakeet auklet`, `painted bunting`, `bronzed cowbird`, `least flycatcher`, and `ovenbird`. Each co-occurrence feature is visualized through three examples in the form of heatmap pairs shown in a row of the figure. It can be observed that the co-occurrence features recognize objects by putting emphasis on semantic object parts, such as eyes, beaks, and heads, even though part-level annotation or object bounding boxes are not given during training. Furthermore, these parts are consistently detected across im-

(a) (b) (c)

Figure 5. Visualization of five co-occurrence features, each in a row, that are the most *influential* for bird species `parakeet auklet`, `painted bunting`, `bronzed cowbird`, `least flycatcher`, and `ovenbird`, respectively. Each co-occurrence feature is visualized through three images with their heatmap pairs shown in columns (a), (b), and (c), respectively.

ages of the same class. For instance in the first row of Figure 5, the beaks of the three parakeet auklets are highlighted in the first heatmap, while their eyes are highlighted in the second one. In the second, third, and fifth rows, the heads and the bodies of painted buntings, bronzed cowbirds, and ovenbirds are accordingly detected in the two heatmaps. For least flycatchers in the fourth row, their heads and wings are detected. The generated co-occurrence features are rotationally and translationally invariant, and are robust to object deformation. As shown in Figure 5, the co-occurrence features can detect object parts robustly against variations of poses and viewpoints.

5. Conclusions

We have presented a new network layer, called the co-occurrence layer, that enables convolutional neural networks to learn part-based features and better solve increasingly complex object recognition tasks. It performs mutual correlation filtering between feature maps, and learns the co-occurrence features between the numerous object parts detected by the neurons. Unlike most part-based methods, it requires neither pre-defined object parts nor part-level annotation during training. Besides, the proposed co-occurrence layer almost introduces no extra parameters, and is end-to-end trainable. The generated co-occurrence features are rotationally and translationally invariant, and are robust to object deformation. We evaluated this new layer on the Caltech-UCSD bird dataset for fine-grained recognition. The experimental results show that it can reach the state-of-the-art performance based on a model with much fewer parameters. In the future, we plan to generalize this work and apply it the vision applications where CNN with part-based information are appreciated, such as generic object recognition and weakly supervised object detection.

Acknowledgement. This work was supported in part by grants MOST 105-2221-E-001-030-MY2, MOST 105-2218-E-002-032, MOST 105-2218-E-001-006, MOST 105-2221-E-002-090, and MOEA 106-EC-17-A-24-1177.

References

[1] T. Berg and P. Belhumeur. POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013. 1, 3, 7

[2] S. Branson, G. Horn, S. Belongie, and P. Perona. Bird species categorization using pose normalized deep convolutional nets. In *BMVC*, 2014. 3, 6, 7

[3] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013. 3, 7

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[5] J. Deng, W. Dong, R. Socher, L.-J. Li., K. Li, and F.-F. Li. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010. 1, 2

[7] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 2005. 1, 2

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 2

[9] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003. 2

[10] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015. 3

[11] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on computers*, 1973. 2

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 3

[13] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *CVPR*, 2015. 1, 2

[14] C. Göering, E. Rodner, A. Freytag, and J. Denzler. Nonparametric part transfer for fine-grained recognition. In *CVPR*, 2014. 1, 3, 7

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2015. 4

[16] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 6

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 5

[18] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015. 3, 7

[19] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *FGVC*, 2011. 1

[20] J. Krause, H. Jin, J. Yang, and L. Fei-Fei. Fine-grained recognition without part annotations. In *CVPR*, 2015. 7

[21] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. 3

[22] D. Lin, X. Shen, C. Lu, and J. Jia. Deep LAC: Deep localization, alignment and classification for fine-grained recognition. In *CVPR*, 2015. 3

[23] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear CNN models for fine-grained visual recognition. In *ICCV*, 2015. 1, 3, 7

[24] L. Liu, C. Shen, and A. van den Hengel. The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. In *CVPR*, 2015. 7

[25] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1

[26] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010. 6

[27] M. Simon and E. Rodner. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In *ICCV*, 2015. 7

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2, 4, 5

[29] M. Srinivas, Y.-Y. Lin, and H.-Y. M. Liao. Learning deep and sparse feature representation for fine-grained recognition. In *ICME*, 2017. 1

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 3, 5

[31] Y. Tian, P. Luo, X. Wang, and X. Tang. Deep learning strong parts for pedestrian detection. In *ICCV*, 2015. 2

[32] A. Vedaldi and K. Lenc. MatConvNet – convolutional neural networks for matlab. In *ACMMM*, 2015. 6

[33] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, 2011. 1, 2, 5

[34] L. Wan, D. Eigen, and R. Fergus. End-to-end integration of a convolution network, deformable parts model and non-maximum suppression. In *CVPR*, 2015. 2

[35] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang. Multiple granularity descriptors for fine-grained categorization. In *ICCV*, 2015. 2, 3, 7

[36] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *CVPR*, 2000. 2

[37] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR*, 2015. 3, 7

[38] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011. 2

[39] M. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 2

[40] H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A. Elgammal, and D. Metaxas. SPDA-CNN: Unifying semantic part detection and abstraction for fine-grained recognition. In *CVPR*, 2016. 1, 2, 3

[41] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based R-CNNs for fine-grained category detection. In *ECCV*, 2014. 1, 3, 7

[42] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, 2013. 2, 7

[43] N. Zhang, E. Shelhamer, Y. Gao, and T. Darrell. Fine-grained pose prediction, normalization, and recognition. In *ICLR*, 2016. 3

[44] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene CNNs. In *ICLR*, 2015. 2, 4, 5

[45] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *CVPR*, 2012. 2