

# Fine-Grained Classification via Mixture of Deep Convolutional Neural Networks

ZongYuan Ge<sup>†‡</sup>, Alex Bewley<sup>‡</sup>, Christopher McCool<sup>‡</sup>, Peter Corke<sup>†‡</sup>, Ben Upcroft<sup>†‡</sup>, Conrad Sanderson<sup>\*◇</sup>

<sup>†</sup> Australian Centre for Robotic Vision, Brisbane, Australia

<sup>‡</sup> Queensland University of Technology (QUT), Brisbane, Australia

<sup>\*</sup> University of Queensland, Brisbane, Australia

<sup>◇</sup> Data61 / NICTA, Australia

z.ge@qut.edu.au    c.mccool@qut.edu.au

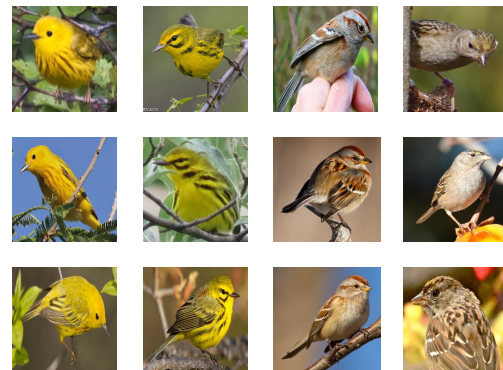
## Abstract

We present a novel deep convolutional neural network (DCNN) system for fine-grained image classification, called a mixture of DCNNs (MixDCNN). The fine-grained image classification problem is characterised by large intra-class variations and small inter-class variations. To overcome these problems our proposed MixDCNN system partitions images into  $K$  subsets of similar images and learns an expert DCNN for each subset. The output from each of the  $K$  DCNNs is combined to form a single classification decision. In contrast to previous techniques, we provide a formulation to perform **joint** end-to-end training of the  $K$  DCNNs simultaneously. Extensive experiments, on three datasets using two network structures (AlexNet and GoogLeNet), show that the proposed MixDCNN system consistently outperforms other methods. It provides a relative improvement of 12.7% and achieves state-of-the-art results on two datasets.

## 1. Introduction

Fine-grained image classification consists of discriminating between classes in a sub-category of objects, for instance the particular species of bird or dog [2, 5, 8, 9, 23]. This is a very challenging problem due to large intra-class variations (due to pose and appearance changes), as well as small inter-class variation (due to only subtle differences in the overall appearance between classes). See Fig. 1 for examples.

To cope with the above problems, many fine-grained classification methods have performed parts detection [2, 5, 20, 24] in order to decrease the intra-class variation. Recently, an alternative approach was introduced by Ge et al. [13] where the images were first partitioned into  $K$  non-overlapping sets and  $K$  expert systems were learned. By grouping similar images, the input space is being partitioned so that an expert network can better learn the subtle differences between similar samples. Expert selection was performed by training a dedicated gating network which as-



**Figure 1.** Example images from the Birdsnap dataset [3] which exhibits large intra-class variations and low inter-class variations. Each column represents a unique class.

signs samples to the most appropriate expert network. This approach has two downsides. Firstly, a separate gating network (subset selector) needs to be trained. Secondly, the expert networks are trained only to extract features, leaving the final classification to be performed by a linear support vector machine (SVM).

We propose a novel system based on a mixture of deep convolutional neural networks (DCNNs) that provides state-of-the-art performance along with several important properties. Similar to Ge et al. [13], we partition the data into  $K$  non-overlapping sets to learn  $K$  expert DCNNs. However, unlike [13], the classification decision from the each expert is weighted proportional to the confidence of its decision. This allows us to define a single network (MixDCNN), comprised of  $K$  sub-networks (expert DCNNs), that can be trained to perform classification. This is in contrast to [13], where each expert is used just for feature extraction. Our system has similarities to the gated network approach proposed by Jacobs et al. [16], which utilises a separately trained network to select the most appropriate expert network.

The proposed MixDCNN system allows us to jointly train the network, which has two advantages: **(i)** it obviates the need for a separate gating network, and **(ii)** samples can be re-assigned to the most appropriate expert network

during the training process. Empirical evaluations show that this approach outperforms related approaches such as subset feature learning [13], a gated DCNN approach similar to [16], and an ensemble of classifiers.

The paper is continued as follows. In Section 2 we briefly review recent advances in fine-grained classification and overview approaches to learn multiple expert classifiers, particularly within the field of neural networks. In Section 3 we present our proposed MixDCNN approach in detail. Section 4 is devoted to a comparative evaluation against several recent methods on the task of fine-grained classification. Conclusions and possible future avenues of research are given in Section 5.

## 2. Prior Work

Prior work for fine-grained image classification has concentrated on performing parts detection [2, 5, 20, 24] in order to decrease the intra-class variation. The part-based one-vs-one feature system [2] is an example of this, where parts-based features are progressively selected to improve classification. An alternative is the deformable parts-model which obtains a combined feature from a set of pre-defined parts [24]. Chai et al. [6] proposed a symbiotic model where part localisation is helped by segmentation and, conversely, the segmentation is helped by parts detection. Zhang et al. [24] extract pose-normalised features based on weak semantic annotations to learn cross-component correspondences of various parts.

Recent work has shown the effectiveness of DCNNs for fine-grained image classification, but again, predominantly to perform parts detection. Region proposal methods combined with a DCNN were shown to more accurately localise object parts [23]. Lin et al. [19] showed that a DCNN can be trained to perform both parts localisation and visibility prediction, achieving state-of-the-art results on the CUB dataset [22]. Although the above parts-based approaches are fully automatic at test time, they require a large number of images to be manually annotated in order to train the model.

To remove the need for time-consuming manual annotations, recent work has explored ways to perform fine-grained classification without using part annotations. Zhang et al. [23] and Ge et al. [12] showed that, even without part annotations, DCNNs can provide impressive performance for fine-grained classification tasks. Of particular interest is the approach of Ge et al. [10] which showed that the data can be partitioned into  $K$  non-overlapping sets and an expert feature extraction algorithm, utilising DCNNs, can be trained for each of the  $K$  sets.

Learning algorithms which construct a set of  $K$  classifiers and make decisions by taking a weighted or average of their predictions are often referred to as ensemble methods. A simple ensemble approach called *bagging* has been used

to improve the overall performance of a system [4]. Bagging manipulates the training examples to generate multiple hypotheses. In this case, a set of  $K$  classifiers is learned using a randomly selected subset of the training data. We use this bagging approach on a set of DCNNs for a baseline method and refer to it as an Ensemble approach (Section 4).

Ensemble approaches, or learning  $K$  expert classifiers, has been explored by several researchers within the context of neural networks. In 1991 Jacobs et al. [16] described a gated network structure to learn  $K$  expert neural networks and applied it to multi-speaker vowel recognition. The underlying idea is to only allocate a small region of the input space to a particular expert system. This was achieved by having  $K$  expert systems (neural networks) which were allocated samples selected by a separate gating network. In [16], the gating network determines the probability that a sample is associated to one of the  $K$  expert systems.

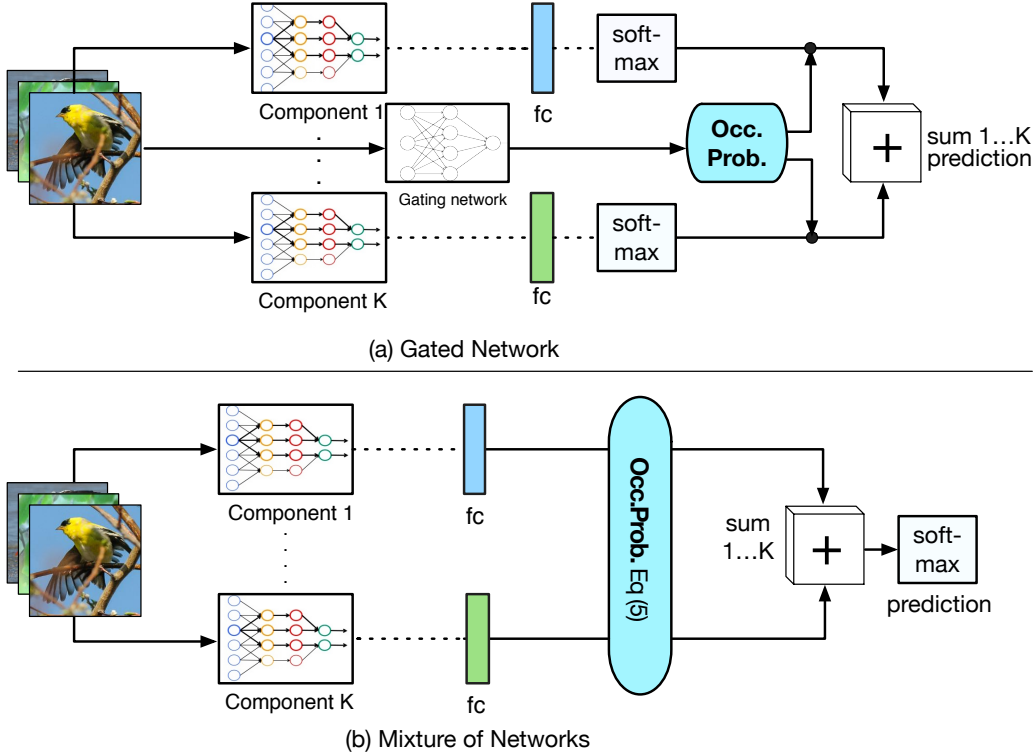
More recently, Ge et al. [13] outlined a subset feature learning (Subset FL) approach using  $K$  expert DCNNs. The data is partitioned into  $K$  non-overlapping sets and for each set an expert DCNN is learned to extract set-specific features. A gating network is then used to extract only the most relevant features from these  $K$  DCNNs. Classification is then performed by training an SVM on these features, yielding impressive performance for fine-grained bird and plant classification [11]. An issue with this work is the reliance of an independent gating network  $\mathcal{G}$  and the fact that feature extraction and classification are treated as independent steps.

## 3. Proposed Approach

We propose a novel mixture of DCNNs (MixDCNN) to improve fine-grained image classification by partitioning the data into  $K$  non-overlapping sets and learning an expert classifier for each set. This approach has similarities to the gated neural network proposed by Jacobs et al. [16], which has never been applied to DCNNs nor to the fine-grained classification problem. As such, we also outline a gated DCNN (GatedDCNN). An overview of these two approaches is given in Figure 2.

The main idea behind the MixDCNN and GatedDCNN approaches is to learn  $K$  expert networks,  $[\mathcal{S}_1, \dots, \mathcal{S}_K]$ , which make decisions about a subset of the data. This simplifies the space that is being modelled by each component. Key to both approaches is being able to assign a sample to the appropriate network.

A GatedDCNN assigns samples by learning a separate gating neural network which produces the probability,  $\alpha_k$ , that the sample belongs to the  $k$ -th network. Learning this gating neural network requires ground truth labels about which sample should be assigned to a particular network, which for our work is an open question. In contrast, a MixDCNN assigns samples based on the confidence of the



**Figure 2.** GatedDCNN structure (top) and MixDCNN structure (bottom). The term *Occ. Prob.* refers to occupation probability (responsibility)  $\alpha$ . In GatedDCNN, the gating network uses the image, the same input as each component (subset networks), to estimate  $\alpha$ . In contrast, MixDCNN estimates  $\alpha$  without the need for an external network.

prediction from each network, which leads us to consider  $\alpha_k$  to be the occupation probability of the sample for the  $k$ -th network.

Before we describe these two approaches in more detail we define some notation. The output of a DCNN, trained for classification, is an  $N$ -dimensional vector  $z$  of class predictions, where  $N$  indicates the number of classes. These predictions then are normalised by a softmax [18, 21] to give the probability that the sample belongs to the  $n$ -th class:

$$c_n = \frac{\exp\{z_n\}}{\sum_{j=1}^N \exp\{z_j\}} \quad (1)$$

In the approaches described below, we are most interested in the vector of predictions  $z$  prior to applying the softmax.

### 3.1. GatedDCNN

Inspired by [15, 16], we define a GatedDCNN that consists of  $K$  components (DCNNs) and an additional gating network. The overall structure of this network is shown in Fig. 2a. In this arrangement, the  $k$ -th DCNN  $\mathcal{S}_k$  is given greater responsibility for learning to discriminate subtle differences of the  $k$ -th subset of images, while the gating network  $\mathcal{G}$  is responsible for associating the image  $I$  with the most appropriate component. The gating network  $\mathcal{G}$  is a

fine-tuned DCNN that is learned using the cross-entropy loss to produce a  $K$ -dimensional vector of probabilities  $\alpha$ . The  $k$ -th value denotes the probability that the input image  $I$  is associated with the  $k$ -th component. We refer to this as an occupation probability.

A fundamental difficulty with training the GatedDCNN is how to provide the  $T$  training labels  $\mathbf{y}$ . This label vector is a  $K$ -dimensional label vector which indicates which of the  $K$  subsets the sample belongs to. To deal with this issue we consider two ways of estimating these labels. The first approach is to initialise the labels  $\mathbf{y}$  using the partitioning of the training images into  $K$  subsets. The gated network  $\mathcal{G}$  is then trained using these labels and the  $K$  DCNNs (components) are then trained independently so that  $\mathcal{S}_k$  is trained exclusively with data from the  $k$ -th subset. The second approach is to use the above gated network (and  $K$  components) as an initialisation and to iteratively retrain by:

1. Fixing  $\mathcal{G}$ , and then updating  $[\mathcal{S}_1, \dots, \mathcal{S}_K]$  using the assignments from  $\mathcal{G}$ .
2. Fixing the  $K$  components  $[\mathcal{S}_1, \dots, \mathcal{S}_K]$  and using these to estimate new labels  $\mathbf{y}$ . The network  $\mathcal{G}$  is then updated using these new labels.

The labels  $\mathbf{y}$  estimated in step 2 are obtained by taking

the network which is most confident about its decision. Formally,  $y_t$  for the  $t$ -th training sample is given by:

$$y_t = \arg \max_{k=1 \dots K} C_{k,t} \quad (2)$$

where  $C_{k,t}$  is the best classification result for  $S_k$  using the  $t$ -th sample:

$$C_{k,t} = \max_{n=1 \dots N} z_{k,n,t} \quad (3)$$

Classification with the GatedDCNN is performed using a weighted summation of the classification results from the  $K$  components:

$$c_n = \sum_{k=1}^K c_{k,n} \alpha_k \quad (4)$$

where  $c_{k,n}$  is the probability of the sample belonging to the  $n$ -th class for the  $k$ -th component, and  $\alpha_k$  is the probability that the sample is assigned to the  $k$ -th component  $S_k$ .

An issue with the GatedDCNN system is that a separate gating network has to be trained to assign a sample to a particular component  $S_k$ . This provides the further complication of having to estimate the labels  $\mathbf{y}$  in order to train the gating network  $\mathcal{G}$ . In this paper the first GatedDCNN training approaches provides marginally better performance. In the experiment section, we will report results based on the first approach.

### 3.2. Mixture of DCNNs (MixDCNN)

We propose a mixture of DCNNs approach where the occupation probabilities  $\alpha$  are based on the classification confidence from each component. An advantage of this structure is that we can jointly train the  $K$  DCNNs (components) without having to estimate a separate label vector  $\mathbf{y}$  or train a separate gating network  $\mathcal{G}$ .

For MixDCNN, the occupation probability for the  $k$ -th component is:

$$\alpha_k = \frac{\exp\{C_k\}}{\sum_{c=1}^K \exp\{C_c\}} \quad (5)$$

where  $C_k$  is given by Eq. (3). This occupation probability gives higher weight to components that are confident about their prediction. The overall structure of this network is shown in Fig. 2b.

Classification is performed by multiplying the output of the final layer from each component by the occupation probability and then summing over the  $K$  components:

$$z_n = \sum_{k=1}^K z_{k,n} \alpha_k \quad (6)$$

This mixes the network outputs together and the probability for each class is then produced by applying the softmax function in Eq. (1). As a consequence our MixDCNN is optimised using the cross-entropy loss<sup>1</sup>.

<sup>1</sup>Optimised in a mini-batch Stochastic Gradient Descent framework.

### 3.3. Differences Between MixDCNN and Ensembles

The aim of the MixDCNN approach is that each component takes greater responsibility for a portion of the data allowing each component to concentrate on samples (or classes) that are more difficult to differentiate. This will allow the MixDCNN to learn subtle differences for similar classes. This is in contrast to an Ensemble approach which randomly excludes a portion of the training data for each DCNN. Therefore, the key difference between the proposed MixDCNN approach and an ensemble of DCNNs (Ensemble) is the use of the occupation probability. For training, this means the MixDCNN approach does not randomly select the data. Instead, each sample is weighted proportional to its relevance to each DCNN  $S_{1, \dots, K}$ . For testing, the MixDCNN approach is able to adaptively calculate the occupation probability for each sample, whereas an Ensemble approach will use pre-defined weights or, more commonly, equal weights.

## 4. Experiments

### 4.1. Datasets

We present results on three fine-grained image classification datasets using two network structures. The three datasets are the Caltech-UCSD-2011 (CUB200-2011) [22], Birdsnap [3], and PlantCLEF 2015 [14]. Example images are shown in Figures 1 and 3.

CUB200-2011 is a fine-grained bird classification task with 11,788 images from 200 bird species in North America. This dataset has become a *de facto* standard for the bird classification task. Each species has approximately 30 images for training and 30 for testing. Birdsnap is a much larger bird dataset consisting of 49,829 images from 500 bird species with 47,386 images used for training and 2,443 images used for testing. PlantCLEF 2015 is a large plant classification dataset that has seven content types. To demonstrate the capabilities of the proposed MixDCNN ap-



Figure 3. Examples from CUB-200-2011 and PlantCLEF Flower.



proach for the task of fine-grained classification, we analyse its effectiveness on one content type, Flower. This portion of the dataset consists of 28,705 images from 967 species. We split this data into training and test sets. The training set consists of 25,025 images from 967 species, while the test set has 3,200 images from 801 species.

Both CUB200-2011 and Birdsnap have bounding box annotations around the object of interest. We use this information to extract just the object of interest from the image. PlantCLEF 2015 does not come with bounding box information making it a more challenging dataset.

Prior work [13, 23] has shown the importance of transfer learning for the fine-grained image classification problem. Results have shown that training a DCNN from scratch for either the fine-grained CUB200-2011 or Birdsnap dataset leads to overfitting on the training samples. As such, for all the of our experiments we use pre-trained networks from ImageNet [7] to provide a good initialisation for each DCNN and then perform transfer learning. We consider this to be our baseline and refer to it as **DCNN-tl**. All of our networks are trained using Caffe [17] and partitioning was performed using the Bob toolkit [1].

## 4.2. Comparative Evaluation

We compare the proposed MixDCNN approach against four other related methods: (1) the baseline DCNN-tl, (2) an ensemble of  $K$  DCNNs, (3) an implementation of Gated-DCNN, and (4) Subset FL [13]. Two network structures considered are the well known AlexNet [18] and the Large Scale Visual Recognition Challenge (ILSVRC) 2014 winner GoogLeNet [21]. AlexNet is a deep network consisting of 8 layers, while ILSVRC has 22 layers<sup>2</sup>. We follow the same procedure as Ge et. al [13] to cluster the data. For the AlexNet structure we use the output of the first fully connected layer as features for clustering. For GoogLeNet we use the output of the last layer, prior to classification, as features. In both cases linear discriminant analysis (LDA) is applied to reduce the dimensionality to  $D = 128$ . In our initial experiments, we varied  $D$  and results showed no impact of that.

The results in Table 1 show that the proposed MixDCNN approach provides consistent improvement regardless of network structure or dataset. MixDCNN provides the best performance for all of the network and dataset combinations, with the exception of the MixDCNN model using the GoogLeNet structure on CUB. It provides an average relative performance improvement of 12.7% over the baseline DCNN-tl approach, excluding CUB.

For the CUB dataset, using multiple expert networks provides limited performance improvement. This is true for all of the methods examined. We attribute this to the

<sup>2</sup>To prevent GoogLeNet from over-fitting we use a higher dropout rate equal to 0.5 for the final loss layer, as opposed to the original setting of 0.4.

fact that CUB200-2011 is a small dataset consisting of just 5,994 training images. This is an order of magnitude fewer samples than other datasets such as Birdsnap. Furthermore, applying transfer learning to GoogLeNet already provides exceptional performance and so minimises the improvement introduced by the MixDCNN framework, or any multi-expert approach.

The proposed MixDCNN method achieves state-of-the-art results on the challenging Birdsnap and PlantCLEF-Flower datasets. For Birdsnap the previous state-of-the-art performance was 48.8% [3]. Applying transfer learning to GoogLeNet already outperforms this prior art with an accuracy of 67.4%. MixDCNN provides a further relative performance improvement of 9.9%. For the PlantCLEF-Flower dataset the baseline performance of DCNN-tl (using GoogLeNet) is 48.7%. MixDCNN provides state-of-the-art performance with a relative performance improvement of 7.0%.

The MixDCNN approach consistently outperforms the Ensemble, GatedDCNN and Subset FL approaches. Interestingly, it provides a considerable improvement over the closely related GatedDCNN approach, with an average relative performance improvement of 9.1%. We attribute this to the ability of the MixDCNN approach to adaptively re-assign samples to the most appropriate expert network, in spite of the original partitioning.

In our experiments, component sizes greater than  $K = 6$  were not considered as we could not store these in memory on a single GPU<sup>3</sup>. This highlights one of the limitations with this technique as it currently requires all of the networks to be stored on a single GPU; future work should consider how to extend the architecture across multiple GPUs.

## 5. Conclusion

We have proposed a novel mixture of deep neural networks, termed MixDCNN, which achieves state-of-the-art performance for fine-grained classification. It provides an average relative performance improvement of 12.7% and has been shown to consistently outperform several related methods: subset feature learning, GatedDCNN, and an ensemble of classifiers.

The key advantage of our proposed approach is the use of an occupation probability that weights each sample proportional to its relevance to each DCNN  $\mathcal{S}_{1,\dots,K}$ . This approach obviates the need for a separate gating function and highlights the importance of being able to adaptively weight samples based on their relevance to a component (DCNN).

Future work will explore alternative methods for initialising the clustering and its impact upon performance. For instance, the impact of grouping images together in terms

<sup>3</sup>The GPU used in all our experiments was an Nvidia K40 Tesla with 12 Gb of memory.

**Table 1.** Comparison of the proposed MixDCNN approach against DCNN-tl, Ensemble, GatedDCNN and Subset FL on three datasets: CUB, BirdSnap and PlantCLEF-Flower. Two network structures are used: AlexNet and GoogLeNet.

		DCNN-tl	Ensemble	GatedDCNN	Subset FL	MixDCNN
AlexNet	CUB	68.3%	71.2%	69.2%	72.0%	<b>73.4%</b>
	BirdSnap	55.7%	57.2%	57.4%	59.3%	<b>63.2%</b>
	PlantCLEF-Flower	29.1%	30.2%	30.2%	31.1%	<b>35.0%</b>
GoogLeNet	CUB	80.0%	80.9%	81.0%	<b>81.2%</b>	81.1%
	BirdSnap	67.4%	71.4%	70.1%	72.8%	<b>74.1%</b>
	PlantCLEF-Flower	48.7%	50.2%	49.7%	51.7%	<b>52.1%</b>

of their pose rather than similar visual appearance. Furthermore, we will examine the role of the occupation probability in two ways: (i) whether the responsibility for a sample is shared between components, and (ii) deeper analysis of how this occupation probability changes during the training process. Additionally, we intend on exploring different methods for computing the occupational probability via alternative aggregation techniques.

**Acknowledgements.** The Australian Centre for Robotic Vision is supported by the Australian Research Council via the Centre of Excellence program. NICTA is funded by the Australian Government through the Department of Communications, as well as the Australian Research Council through the ICT Centre of Excellence program.

## References

- [1] A. Anjos, L. E. Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel. Bob: a free signal processing and machine learning toolbox for researchers. In *ACM International Conference on Multimedia*, pages 1449–1452, 2012.
- [2] T. Berg and P. N. Belhumeur. POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013.
- [3] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *CVPR*, pages 2019–2026, 2014.
- [4] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
- [5] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013.
- [6] Y. Chai, V. Lempitsky, and A. Zisserman. Symbiotic segmentation and part localization for fine-grained categorization. In *ICCV*, 2013.
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [8] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011.
- [9] E. Gavves, B. Fernando, C. G. Snoek, A. W. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013.
- [10] Z. Ge, C. McCool, C. Sanderson, A. Bewley, Z. Chen, and P. Corke. Fine-grained bird species recognition via hierarchical subset learning. In *IEEE International Conference on Image Processing (ICIP)*, pages 561–565, 2015.
- [11] Z. Ge, C. McCool, C. Sanderson, and P. Corke. Content specific feature learning for fine-grained plant classification. In *Working Notes of the CLEF 2015 Conference*, 2015.
- [12] Z. Ge, C. McCool, C. Sanderson, and P. Corke. Modelling local deep convolutional neural network features to improve fine-grained image classification. In *IEEE Int. Conference on Image Processing (ICIP)*, pages 4112–4116, 2015.
- [13] Z. Ge, C. McCool, C. Sanderson, and P. Corke. Subset feature learning for fine-grained classification. In *DeepVision Workshop, Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 46–52, 2015.
- [14] H. Goëau, A. Joly, P. Bonnet, S. Selmi, J.-F. Molino, D. Barthélémy, and N. Boujema. Lifeclef plant identification task 2014. In *Working Notes for CLEF 2014 Conference*, pages 598–615. CEUR-WS, 2014.
- [15] J. B. Hampshire II and A. Waibel. The meta-pi network: Building distributed knowledge representations for robust multisource pattern recognition. *PAMI*, 14(7):751–769, 1992.
- [16] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.
- [17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, pages 675–678, 2014.
- [18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.
- [19] D. Lin, X. Shen, C. Lu, and J. Jia. Deep LAC: Deep localization, alignment and classification for fine-grained recognition. In *CVPR*, pages 1666–1674, 2015.
- [20] J. Liu, A. Kanazawa, D. Jacobs, and P. Belhumeur. Dog breed classification using part localization. In *ECCV*, 2012.
- [21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CVPR*, 2014.
- [22] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. In *Computation & Neural Systems Technical Report, California Institute of Technology*, number CNS-TR-2011-001, 2011.
- [23] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based R-CNNs for fine-grained category detection. In *ECCV*, pages 834–849, 2014.
- [24] N. Zhang, R. Farrell, F. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, 2013.