This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Growing a Brain: Fine-Tuning by Increasing Model Capacity

Yu-Xiong Wang Deva Ramanan Martial Hebert
Robotics Institute, Carnegie Mellon University
{yuxiongw,dramanan,hebert}@cs.cmu.edu

Abstract

CNNs have made an undeniable impact on computer vision through the ability to learn high-capacity models with large annotated training sets. One of their remarkable properties is the ability to transfer knowledge from a large source dataset to a (typically smaller) target dataset. This is usually accomplished through fine-tuning a fixed-size network on new target data. Indeed, virtually every contemporary visual recognition system makes use of fine-tuning to transfer knowledge from ImageNet. In this work, we analyze what components and parameters change during fine-tuning, and discover that increasing model capacity allows for more natural model adaptation through fine-tuning. By making an analogy to developmental learning, we demonstrate that "growing" a CNN with additional units, either by widening existing layers or deepening the overall network, significantly outperforms classic fine-tuning approaches. But in order to properly grow a network, we show that newly-added units must be appropriately normalized to allow for a pace of learning that is consistent with existing units. We empirically validate our approach on several benchmark datasets, producing state-of-the-art results.

1. Motivation

Deep convolutional neural networks (CNNs) have revolutionized visual understanding, through the ability to learn "big models" (with hundreds of millions of parameters) with "big data" (very large number of images). Importantly, such data must be annotated with human-provided labels. Producing such massively annotated training data for new categories or tasks of interest is typically unrealistic. Fortunately, when trained on a large enough, diverse "base" set of data (*e.g.*, ImageNet), CNN features appear to transfer across a broad range of tasks [32, 4, 56]. However, an open question is how to best adapt a pre-trained CNN for novel categories/tasks.

Fine-tuning: Fine-tuning is by far the dominant strategy for transfer learning with neural networks [28, 4, 32, 53, 10, 12]. This approach was pioneered in [14] by transferring knowledge from a generative to a discriminative model, and has since been generalized with great success [10, 57]. The basic pipeline involves replacing the last "classifier" layer of a pre-trained network with a new randomly initialized layer for the target task of interest. The modified network is then fine-tuned with additional passes of appropriately tuned gradient descent on the target training set. Virtually *every* contemporary visual recognition system uses this pipeline. Even though its use is widespread, fine-tuning is still relatively poorly understood. For example, what fraction of the pre-trained weights actually change and how?

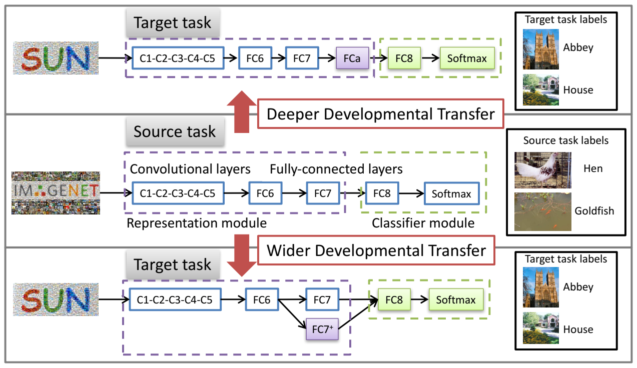


Figure 1: Transfer and developmental learning of pre-trained CNNs by increasing model capacity for the recognition of novel categories from few examples. The network (*e.g.*, AlexNet) is pre-trained on the source task (*e.g.*, ImageNet classification) with abundant data (middle row). Different from the dominant paradigm of fine-tuning a fixed-capacity model, we *grow this network* when adapting it to a novel target task (*e.g.*, SUN-397 scene classification) in two ways: (1) going deeper by adding more layers (top) and (2) going wider by adding more channels per layer (bottom).

Developmental networks: To address this issue, we explore "developmental" neural networks that grow in model capacity as new tasks as encountered. We demonstrate that growing a network, by adding additional units, facilitates knowledge transfer to new tasks. We explore two approaches to adding units as shown in Figure 1: going deeper (more layers) and wider (more channels per layer). Through visualizations, we demonstrate that these additional units help guide the adaptation of pre-existing units. Deeper units allow for new compositions of pre-existing units, while wider units allow for the discovery of complementary cues that address the target task. Due to their progressive nature, developmental networks still remain accurate on their source task, implying that they can learn without forgetting. Finally, we demonstrate that developmental networks particularly facilitate continual transfer across multiple tasks.

Developmental learning: Our approach is loosely inspired by developmental learning in cognitive science. Humans, and in particular children, have the remarkable ability to continually transfer previously-acquired knowledge to novel scenarios. Much of the literature from both neuroscience [26] and psychology [16] suggests that such sequential knowledge acquisition is intimately tied with a child's growth and development.

Contributions: Our contributions are three-fold. (1) We first demonstrate that the dominant paradigm of fine-tuning a fixed-capacity model is sub-optimal. (2) We explore several avenues for increasing model capacity, both in terms of going deeper (more layers) and wider (more channels per layer), and consistently find that increasing capacity helps, with a slight preference for widening. (3) We show that additional units must be normalized and scaled appropriately such that the "pace of learning" is balanced with existing units in the model. Finally, we use our analysis to build a relatively simple pipeline that "grows" a pre-trained model during fine-tuning, producing state-of-the-art results across a large number of standard and heavily benchmarked datasets (for scene classification, fine-grained recognition, and action recognition).

2. Related Work

While there is a large body of work on transfer learning, much of it assumes a *fixed* capacity model [32, 3, 6, 58, 15]. Notable exceptions include [28], who introduce an adaptation layer to facilitate transfer. Our work provides a systematic exploration of various methods for increasing capacity, including both the addition of new layers and widening of existing ones. Past work has explored strategies for preserving accuracy on the source task [22, 8], while our primary focus is on improving accuracy on the target task. Most relevant to us are the *progressive* networks of [34], originally proposed for reinforcement learning. Interestingly, [34, 38] focus on widening a target network to be twice as large as the source one, but fine-tune only the new units. In contrast, we add a small fraction of new units (both by widening and deepening) but fine-tune *the entire network*, demonstrating that adaptation of old units is crucial for high performance.

Transfer learning is related to both multi-task learning [32, 4, 28, 10, 11, 45, 24, 2] and learning novel categories from few examples [49, 19, 21, 35, 51, 22, 5, 13, 50, 47, 31]. Past techniques have applied such approaches to transfer learning by learning networks that predict models rather than classes [51, 31]. This is typically done without dynamically growing the number of parameters across new tasks (as we do).

In a broad sense, our approach is related to developmental learning [26, 16, 36] and lifelong learning [41, 25, 39, 29]. Different from the non-parametric shallow models (*e.g.*, nearest neighbors) that increase capacity when memorizing new data [40, 42], our developmental network cumulatively grows its capacity *from novel tasks*.

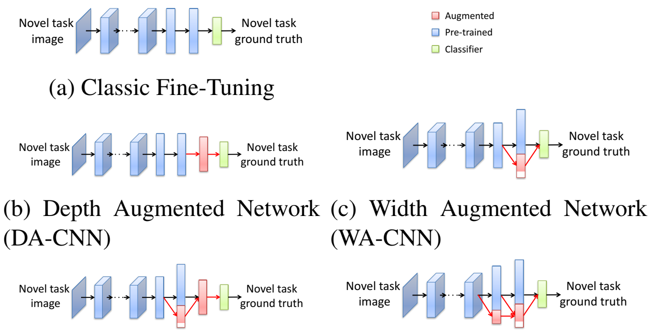
3. Approach Overview

Let us consider a CNN architecture pre-trained on a source domain with abundant data, *e.g.*, the vanilla AlexNet pre-trained on ImageNet (ILSVRC) with $1{,}000$ categories [20, 33]. We note in Figure 1 that the CNN is composed of a feature representation module $\mathcal{F}$ (*e.g.*, the five convolutional layers and two fully connected layers for AlexNet) and a classifier module $\mathcal{C}$ (*e.g.*, the final fully-connected layer with $1{,}000$ units and the $1{,}000$-way softmax for ImageNet classification). Transferring this CNN to a novel task with limited training data (*e.g.*, scene classification of 397 categories from SUN-397 [52]) is typically done through fine-tuning [3, 1, 15].

In classic fine-tuning, the target CNN is instantiated and initialized as follows: (1) the representation module $\mathcal{F}_T$ is copied from $\mathcal{F}_S$ of the source CNN with the parameters $\Theta_T^{\mathcal{F}} = \Theta_S^{\mathcal{F}}$; and (2) a new classifier model $\mathcal{C}_T$ (*e.g.*, a new final fully-connected layer with 397 units and the 397-way softmax for SUN-397 classification) is introduced with the parameters $\Theta_T^{\mathcal{C}}$ randomly initialized. All (or a portion of) the parameters $\Theta_T^{\mathcal{F}}$ and $\Theta_T^{\mathcal{C}}$ are fine-tuned by continuing the backpropagation, with a smaller learning rate for $\Theta_T^{\mathcal{F}}$. Because $\mathcal{F}_T$ and $\mathcal{F}_S$ have identical network structure, the representational capacity is fixed during transfer.

Our underlying thesis is that fine-tuning will be facilitated by *increasing representational capacity* during transfer learning. We do so by adding S new units $\{u_s\}_{s=1}^{S}$ into $\mathcal{F}_T$. As we will show later in our experiments, this significantly improves the ability to transfer knowledge to target tasks, particularly when fewer target examples are provided [43]. We call our architecture a *developmental network*, in which the new representation module $\mathcal{F}_T^* = \mathcal{F}_T \cup \{u_s\}_{s=1}^{S}$, and the classifier module remains $\mathcal{C}_T$.

Conceptually, new units can be added to an existing network in a variety of ways. A recent analysis, however, suggests that early network layers tend to encode generic features, while later layers tend to endode task-specific features [56]. Inspired from this observation, we choose to explore new units at later layers. Specifically, we either construct a completely new top layer, leading to a *depth augmented network* (DA-CNN) as shown in Figure 2b, or widen an existing top layer, leading to a *width augmented network*



(a) Classic Fine-Tuning

(b) Depth Augmented Network (DA-CNN)

(c) Width Augmented Network (WA-CNN)

(d) Jointly Depth and Width Augmented Network (DWA-CNN)

(e) Recursively Width Augmented Network (WWA-CNN)

Figure 2: Illustration of classic fine-tuning (a) and variations of our developmental networks with augmented model capacity (b–e).

(WA-CNN) as shown in Figure 2c. We will explain these two types of network configurations in Section 4. Their combinations—a jointly depth and width augmented network (DWA-CNN) as shown in Figure 2d and a recursively width augmented network (WWA-CNN) as shown in Figure 2e—will also be discussed in Section 5.

4. Developmental Networks

For the target task, let us assume that the representation module $\mathcal{F}_T$ with fixed capacity consists of K layers $L_k, k = 1, \ldots, K$ with hidden activations $\boldsymbol{h}^k \in \mathcal{R}^{n_k}$, where n_k is the number of units at layer k. Let $\boldsymbol{W}^k$ be the weights between layer k and layer $k-1$. That is, $\boldsymbol{h}^k = f\left(\boldsymbol{W}^k \boldsymbol{h}^{k-1}\right)$, where $f(\cdot)$ is a non-linear function, such as ReLU. For notational simplicity, $\boldsymbol{h}^k$ already includes a constant 1 as the last element and $\boldsymbol{W}^k$ includes the bias terms.

4.1. Depth Augmented Networks

A straightforward way to increase representational capacity is to construct a new top layer L_a of size S using $\{u_s\}_{s=1}^S$ on top of L_K, leading to the depth augmented representation module $\mathcal{F}_T^*$ as shown in Figure 2b. We view L_a as an adaptation layer that allows for novel compositions of pre-existing units, thus avoiding dramatic modifications to the pre-trained layers for their adaptation to the new task. The new activations $\boldsymbol{h}^a = f\left(\boldsymbol{W}^a \boldsymbol{h}^K\right)$ in layer L_a become the representation that is fed into the classifier module $\mathcal{C}_T$, where $\boldsymbol{W}^a$ denotes the weights between layers L_a and L_K.

4.2. Width Augmented Networks

An alternative way is to expand the network by adding $\{u_s\}_{s=1}^S$ to some existing layers while keeping the depth of the network fixed as shown in Figure 2c. Without loss of generality, we add all the units to the top layer L_K. Now the new top representation layer L_K^* consists of two blocks: the original L_K and the added L_K^+ with units $\{u_s\}_{s=1}^S$, leading to the width augmented representation module $\mathcal{F}_T^*$. The connection weights between L_K and the underneath layer L_{K-1} remains, *i.e.*, $\boldsymbol{h}^K = f\left(\boldsymbol{W}^K \boldsymbol{h}^{K-1}\right)$. We introduce additional lateral connection weights $\boldsymbol{W}^{K^+}$ between L_K^+ and L_{K-1}, which are randomly initialized, *i.e.*, $\boldsymbol{h}^{K^+} = f\left(\boldsymbol{W}^{K^+} \boldsymbol{h}^{K-1}\right)$. Finally, the concatenated activations $\left[\boldsymbol{h}^K, \boldsymbol{h}^{K^+}\right]$ of size $n_K + S$ from layer L_K^* are fed into the classifier module.

4.3. Learning at the Same Pace

Ideally, our hope is that the new and old units cooperate with each other to boost the target performance. For width augmented networks, however, the units start to learn at a different pace during fine-tuning: while the original units at layer L_k are already well learned on the source domain and only need a small modification for adaptation, the new set of units at layer L_k^+ are just set up through random initialization. They thus have disparate learning behaviors, in the sense that their activations generally have different scales. Naïvely concatenating these activations would restrict the corresponding units, leading to degraded performance and even causing collapsed networks, since the larger activations dominate the smaller ones [23]. Although the weights might adjust accordingly as fine-tuning processes, they require very careful initialization and tuning of parameters, which is dataset dependent and thus not robust. This is partially the reason that the previous work showed that network expansion was inferior to standard fine-tuning [22].

To reconcile the learning pace of the new and pre-existing units, we introduce an additional normalization and adaptive scaling scheme in width augmented networks, which is inspired by the recent work on combining multi-scale pre-trained CNN features from different layers [23]. More precisely, after weight initialization of $\mathcal{F}_T^*$, we first apply an $\mathcal{L}_2$-norm normalization to the activations $\boldsymbol{h}^k$ and $\boldsymbol{h}^{k^+}$, respectively:

$$\widehat{\boldsymbol{h}}^k = \boldsymbol{h}^k \Big/ \left\|\boldsymbol{h}^k\right\|_2, \quad \widehat{\boldsymbol{h}}^{k^+} = \boldsymbol{h}^{k^+} \Big/ \left\|\boldsymbol{h}^{k^+}\right\|_2. \tag{1}$$

By normalizing these activations, their scales become homogeneous. Simply normalizing the norms to 1 slows down the learning and makes it hard to train the network, since the features become very small. Consistent with [23], we normalize them to a larger value (*e.g.*, 10 or 20), which encourages the network to learn well. We then introduce a scaling parameter γ for each channel to scale the normalized value:

$$y_i^k = \gamma_i \widehat{h}_i^k, \quad y_j^{k^+} = \gamma_j \widehat{h}_j^{k^+}. \tag{2}$$

We found that for depth augmented networks, while this additional stage of normalization and scaling is not crucial, it is still beneficial. In addition, this stage only introduces negligible extra parameters, whose number is equal to the total number of channels. During fine-tuning, the scaling factor γ is fine-tuned by backpropagation as in [23].

5. Experimental Evaluation

In this section, we explore the use of our developmental networks for transferring a pre-trained CNN to a number of supervised learning tasks with insufficient data, including scene classification, fine-grained recognition, and action recognition. We begin with extensive evaluation of our approach on scene classification of the SUN-397 dataset, focusing on the variations of our networks and different design choices. We also show that the network remains accurate on the source task. We then provide an in-depth analysis of fine-tuning procedures to qualitatively understand why fine-tuning with augmented network capacity outperforms classic fine-tuning. We further evaluate our approach on other novel categories and compare with state-of-the-art approaches. Finally, we investigate whether progressive augmenting outperforms fine-tuning a fixed large network and investigate how to cumulatively add new capacity into the network when it is gradually adapted to multiple tasks.

Implementation details: Following the standard practice, for computational efficiency and easy fine-tuning we use the Caffe [17] implementation of AlexNet [20], pre-trained on ILSVRC 2012 [33], as our reference network in most of our experiments. We found that our observations also held for other network architectures. We also provide a set of experiment using VGG16 [37]. For the target tasks, we randomly initialize the classifier layers and our augmented layers. During fine-tuning, after resizing the image to be 256×256, we generate the standard augmented data including random crops and their flips as implemented in Caffe [17]. During testing, we only use the central crop, unless otherwise specified. For a fair comparison, fine-tuning is performed using stochastic gradient descent (SGD) with the "step" learning rate policy, which drops the learning rate in steps by a factor of 10. The new layers are fine-tuned at a learning rate 10 times larger than that of the pre-trained layers (if they are fine-tuned). We use standard momentum 0.9 and weight decay 0.0005 without further tuning.

5.1. Evaluation and Analysis on SUN-397

We start our evaluation on scene classification of the SUN-397 dataset, a medium-scale dataset with around $108K$ images and 397 classes [52]. In contrast to other fairly small-scale target datasets, SUN-397 provides sufficient number of categories and examples while demonstrating apparent dissimilarity with the source ImageNet dataset. This greatly benefits our insight into fine-tuning procedures and leads to clean comparisons under controlled settings.

We follow the experimental setup in [1, 15], which uses a nonstandard train/test split since it is computationally expensive to run all of our experiments on the 10 standard subsets proposed by [52]. Specifically, we randomly split the dataset into train, validation, and test parts using 50%, 10%, and 40% of the data, respectively. The distribution of categories is uniform across all the three sets. We report 397-way multi-class classification accuracy averaged over all categories, which is the standard metric for this dataset. We report the results using a single run due to computational constraints. Consistent with the results reported in [1, 15], the standard deviations of accuracy on SUN-397 classification are negligible, and thus having a single run should not affect the conclusions that we draw. For a fair comparison, fine-tuning is performed for around 60 epochs using SGD with an initial learning rate of 0.001, which is reduced by a factor of 10 around every 25 epochs. All the other parameters are the same for all approaches.

Network	Type	Method	Acc (%)			
			New	FC_7–New	FC_6–New	All
AlexNet	Baselines	Finetuning-CNN	53.63	54.75	54.29	55.93
		[1, 15]	48.4	—	51.6	52.2
	Single (Ours)	DA-CNN	54.24	56.48	57.42	58.54
		WA-CNN	**56.81**	56.99	57.84	58.95
	Combined (Ours)	DWA-CNN	56.07	56.41	56.97	57.75
		WWA-CNN	56.65	**57.10**	**58.16**	**59.05**
VGG16	Baselines	Finetuning-CNN	60.77	59.09	50.54	62.80
	Single (Ours)	DA-CNN	61.21	62.85	63.07	65.55
		WA-CNN	**63.61**	**64.00**	**64.15**	**66.54**

Table 1: Performance comparisons of classification accuracy (%) between the variations of our developmental networks *with augmented model capacity* and classic fine-tuning *with fixed model capacity* on scene classification of the SUN-397 dataset. The variations include: (1) for AlexNet, depth augmented network (DA-CNN), width augmented network (WA-CNN), jointly depth and width augmented network (DWA-CNN), and recursively width augmented network (WWA-CNN); and (2) for VGG16, DA-CNN and WA-CNN. Both our networks and the baselines are evaluated in four scenarios of gradually increasing the degree of fine-tuning, including fine-tuning only new layers, from FC_7 to new layers, from FC_6 to new layers, and the entire network. Ours significantly outperform the vanilla fine-tuned CNN in all these scenarios.

Learning with augmented network capacity: We first evaluate our developmental networks obtained by introducing a *single* new layer to deepen or expand the pre-trained AlexNet. For the depth augmented network (DA-CNN), we add a new fully connected layer FCa of size S^D on top of FC_7 whose size is $4{,}096$, where $S^D \in \{1{,}024, 2{,}048, 4{,}096, 6{,}144\}$. For the width augmented network (WA-CNN), we add a set of S^W new units as FC_7^+ to FC_7, where $S^W \in \{1{,}024, 2{,}048\}$. After their structures are adapted to the target task, the networks then continue learning in four scenarios of gradually increasing the degree of fine-tuning: (1) "New": we only fine-tune the new layers, including the classifier layers and the augmented layers, while freezing the other pre-trained layers (*i.e.*, the off-the-shelf use case of CNNs); (2) "FC_7–New": we fine-tune from the FC_7 layer; (3) "FC_6–New": we fine-tune from the

Method	Configuration	New	FC_7–New	FC_6-New	All
DA-CNN	FC_a–1,024	53.36	56.31	57.22	57.98
	FC_a–2,048	53.82	56.47	57.14	58.07
	FC_a–4,096	54.02	56.46	57.41	58.32
	FC_a–6,144	54.24	56.48	57.42	58.54
WA-CNN	FC_7^+–1,024	56.46	56.71	57.55	58.90
	FC_7^+–2,048	56.81	56.99	57.84	58.95
DWA-CNN	FC_7^+–1,024–FC_a–1,024	55.44	55.77	56.71	57.49
	FC_7^+–2,048-FC_a–2,048	56.07	56.41	56.97	57.75
WWA-CNN	FC_6^+–512–FC_7^+–1,024	56.13	57.10	57.65	58.80
	FC_6^+–1,024–FC_7^+–2,048	56.49	57.10	57.98	59.05
	FC_6^+–2,048–FC_7^+–4,096	56.65	57.03	58.16	58.98

Table 2: Diagnostic analysis of classification accuracy (%) for the variations of our developmental networks with different number of new units on SUN-397.

Method	Scaling	New	FC_7–New	FC_6–New	All
DA-CNN	*w/o*	**53.82**	**56.47**	56.25	57.21
FC_a–2,048	*w/*	53.51	56.15	**57.14**	**58.07**
WA-CNN	*w/o* (rand)	53.78	54.66	49.72	51.34
FC_7^+–2,048)	*w/o* (copy+rand)	53.62	54.35	53.70	55.31
	w/	**56.81**	**56.99**	**57.84**	**58.95**

Table 3: Performance comparisons of classification accuracy (%) for our depth (DA-CNN) or width (WA-CNN) augmented network *with and without* introducing normalization and scaling on SUN-397. The number of new units in FC_a for DA-CNN or in FC_7^+ for WA-CNN is generally 2,048. Our normalization and scaling strategy reconciles the learning pace of new and old units, and thus greatly benefits both types of networks, *in particular WA-CNN*.

FC_6 layer; (4) "All": we fine-tune the entire network.

Table 1 summarizes the performance comparison with classic fine-tuning. The performance gap between our implementation of the fine-tuning baseline and that in [1, 15] is mainly due to different number of iterations: we used twice of the number of epochs in [1, 15] (30 epochs), leading to improved accuracy. Note that these numbers cannot be directly compared against other publicly reported results due to different data split. With relatively sufficient data, fine-tuning through the full network yields the best performance for all the approaches. Both our DA-CNN and WA-CNN significantly outperform the vanilla fine-tuned CNN *in all the different fine-tuning scenarios*. This verifies the effectiveness of increasing model capacity when adapting it to a novel task. While they have achieved comparable performance, WA-CNN slightly outperforms DA-CNN.

Increasing network capacity through combination or recursion: Given the promise of DA-CNN and WA-CNN, we further augment the network by making it both deeper and wider or two-layer wider. For the jointly depth and width augmented network (DWA-CNN) (Figure 2d), we add FC_a of size S^{DW} on top of FC_7 while expanding FC_7 using FC_7^+ of size S^{DW}, where $S^{DW} \in\{1{,}024, 2{,}048\}$. For the recursively width augmented network (WWA-CNN) (Figure 2e), we both expand FC_7 using FC_7^+ of size S_7^{WW} and FC_6 using FC_6^+ of size S_6^{WW}, where $S_7^{WW} \in \{1{,}024, 2{,}048, 4{,}096\}$ and S_6^{WW} is half of S_7^{WW}.

We compare DWA-CNN and WWA-CNN with DA-CNN and WA-CNN in Table 1. The two-layer WWA-CNN generally achieves the best performance, indicating the importance of augmenting model capacity at different and complementary levels. The jointly DWA-CNN lags a little bit behind the purely WA-CNN. This implies different learning behaviors when we make the network deeper or wider. Their combination thus becomes a non-trivial task.

Diagnostic analysis: While we summarize the best performance in Table 1, a diagnostic experiment in Table 2 on the number of augmented units S^D, S^W, S^{DW}, and S^{WW} shows that *all of these variations of network architectures significantly outperform classic fine-tuning*, indicating the robustness of our approach. We found that this observation was also consistent with other datasets, which we evaluated in the later section. Overall, the performance increases with the augmented model capacity (represented by the size of augmented layers), although the performance gain diminishes with the increasing number of new units.

Importance of reconciling the learning pace of new and old units: The previous work showed that network expansion did not introduce additional benefits [22]. We argue that its unsatisfactory performance is because of the failure of taking into account the different learning pace of new and old units. After exploration of different strategies, such as initialization, we found that the performance of a width augmented network significantly improves by a simple normalization and scaling scheme when concatenating the pre-trained and expanded layers. This issue is investigated for both types of model augmentation in Table 3. The number of new units is generally 2,048; in the case of copying weights of the pre-trained FC_7 and then adding random noises as initialization for FC_7^+, we use 4,096 new units.

For WA-CNN, if we naïvely add new units without considering scaling, Table 3 shows that the performance is either only marginally better or even worse than classic fine-tuning (when fine-tuning more aggressively) in Table 1. This is consistent with the observation made in [22]. However, once the learning pace of the new and old units is re-balanced by scaling, WA-CNN exceeds the baseline by a large margin. For DA-CNN, directly adding new units without scaling already greatly outperforms the baseline, which is consistent with the observation in [28], although scaling provides additional performance gain. This suggests slightly different learning behaviors for depth and width augmented networks. When a set of new units are added to form a purely new layer, they have relatively more freedom to learn from scratch, making the additional scaling *beneficial yet inessential*. When the units are added to expand a pre-trained layer, however, the constraints from the synergy require them to learn to collaborate with the pre-existing units, which is explicitly achieved by the additional scaling.

Evaluation with the VGG16 architecture: Table 1 also summarizes the performance of DA-CNN and WA-CNN using VGG16 [37] and shows the generality of our approach. Due to GPU memory and time constraints, we reduce the batch size and perform fine-tuning for around 30 epochs using SGD. All the other parameters are the same as before. Also, following the standard practice in Fast R-CNN [9], we fine-tune from the layer $Conv2\_1$ in the "All" scenario.

Learning without forgetting: Conceptually, due to their developmental nature, our networks should remain accurate on their source task. Table 4 validates such ability of *learning without forgetting* by showing their classification performance on the source ImageNet dataset.

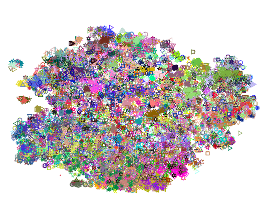
Type	Method	Acc (%)
Oracle	*ImageNet-AlexNet*	*56.9*
References	LwF [22]	55.9
	Joint [22]	56.4
Ours	DA-CNN	55.3
	WA-CNN	51.5

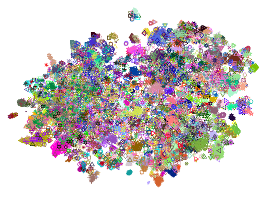
Table 4: Demonstration of the ability of *learning without forgetting* on the source (ImageNet) ILSVRC 2012 validation set. For our DA-CNN and WA-CNN that are fine-tuned on SUN-397, we re-fine-tune on the source ILSVRC 2012 training set, *i.e.*, re-training a new 1,000-way classifier layer and fine-tuning the augmented layers. We show the results of the oracle (*i.e.*, the original AlexNet) and the approaches that are *specifically designed* to preserve the performance on the source task during transfer [22] as references. While our approach focuses on improving the performance on the target task, it remains accurate on the source task. In addition, the existing approaches [22] can be naturally incorporated into our approach to further improve the performance on both source and target tasks.

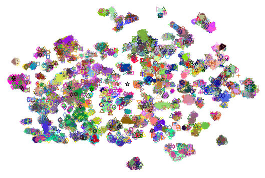
5.2. Understanding of Fine-Tuning Procedures

We now analyze the fine-tuning procedures from various perspectives to gain insight into how fine-tuning modifies the pre-trained network and why it helps by increasing model capacity. We evaluate on the SUN-397 validation set. For a clear analysis and comparison, we focus on DA-CNN and WA-CNN, both with 2,048 new units.

Feature visualization: To roughly understand the topology of the feature spaces, we visualize the features using the standard t-SNE algorithm [46]. As shown in Figure 3, we embed the 4,096-dim FC_7 features of the pre-trained and fine-tuned networks, the 6,144-dim wider $FC_7 + FC_7^+$ features, and the 2,048-dim deeper FC_a features into a 2-dim space, respectively, and plot them as points colored


(a) Pre-Trained Network


(b) Classic Fine-Tuning


(c) Depth Augmented Network (DA-CNN)

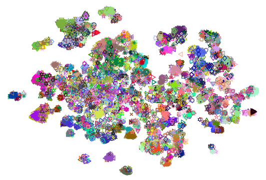

(d) Width Augmented Network (WA-CNN)

Figure 3: t-SNE visualizations of the top feature layers on the SUN-397 validation set. DA-CNN and WA-CNN show significantly better semantic separations.

depending on their semantic category. While classic fine-tuning somehow improves the semantic separation of the pre-trained network, both of our networks demonstrate significantly clearer semantic clustering structures, which is compatible with their improved classification performance.

Maximally activating images: To further analyze how fine-tuning changes the feature spaces, we retrieve the top-5 images that maximally activate some unit as in [10]. We first focus on the *common units* in FC_7 of the pre-trained, fine-tuned, and width augmented networks. In addition to using the SUN-397 images, we also include the maximally activating images from the ILSVRC 2012 validation set for the pre-trained network as references. Figure 4 shows an interesting transition: while the pre-trained network learns certain concentrated concept specific to the source task (left), such concept spreads over as a mixture of concepts for the novel target task (middle left). Fine-tuning tries to re-centralize one of the concepts suitable to the target task, but with limited capability (middle right). Our width augmented network facilitates such re-centralization, leading to more discriminative patterns (right). Similarly, we illustrate the maximally activating images for units in FC_a of the depth augmented network in Figure 5, which shows quite different behaviors. Compared with the object-level concepts in the width augmented network, the depth augmented network appears to have the ability to model a large set of compositions of the pre-trained features and thus generates more scene-level, better clustered concepts.

5.3. Generalization to Other Tasks and Datasets

We now evaluate whether our developmental networks facilitate the recognition of other novel categories. We compare with publicly available baselines and report multi-class

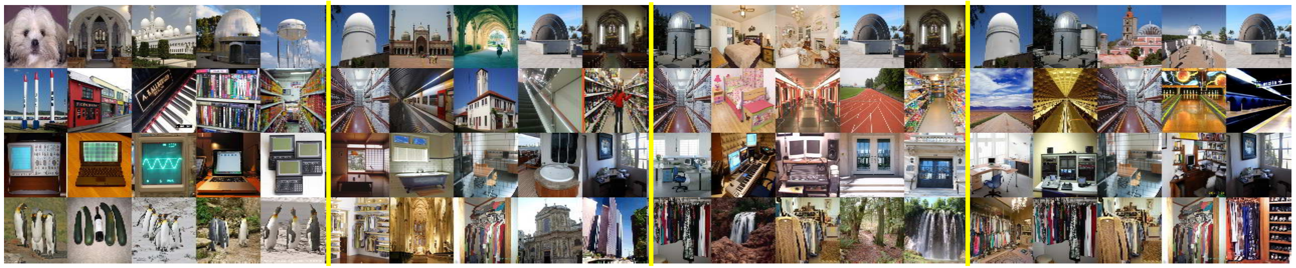


Figure 4: Top 5 maximally activating images for four FC_7 units. From left to right: ILSVRC 2012 validation images for the pre-trained network, and SUN-397 validation images for the pre-trained, fine-tuned, and width augmented (WA-CNN) networks. Each row of images corresponds to a common unit from these networks, indicating that our WA-CNN facilitates the specialization of the pre-existing units towards the novel target task. For example, the bottom row shows a transition from a penguin-like vertically repeated pattern in the pre-trained ImageNet network to several mixed concepts in the fine-tuned network, and finally to a wardrobe-like vertically repeated pattern in our SUN-397 WA-CNN.

Type	MIT-67		102 Flowers		CUB200-2011		Stanford-40	
	Approach	Acc(%)	Approach	Acc(%)	Approach	Acc(%)	Approach	Acc(%)
ImageNet CNNs	Finetuning-CNN	61.2	Finetuning-CNN	75.3	Finetuning-CNN	62.9	Finetuning-CNN	57.7
	Caffe [53]	59.5	CNN-SVM [32]	74.7	CNN-SVM [32]	53.3	Deep Standard [4]	58.9
	—	—	CNNaug-SVM [32]	86.8	CNNaug-SVM [32]	61.8	—	—
Task Customized CNNs	Caffe-DAG [53]	64.6	LSVM [30]	87.1	LSVM [30]	61.4	Deep Optimized [4]	66.4
	—	—	MsML+ [30]	89.5	DeCaf+DPD [7]	65.0	—	—
	Places-CNN [59]	**68.2**	MPP [55]	91.3	MsML+ [30]	66.6	—	—
	—	—	Deep Optimized [4]	91.3	MsML+* [30]	67.9	—	—
Data Augmented CNNs	Combined-AlexNet [18]	58.8	Combined-AlexNet [18]	83.3	—	—	Combined-AlexNet [18]	56.4
Multi-Task CNNs	Joint [22]	63.9	—	—	Joint [22]	56.6	—	—
	LwF [22]	64.5	—	—	LwF [22]	57.7	—	—
Ours	WA-CNN	66.3	WA-CNN	**92.8**	WA-CNN	**69.0**	WA-CNN	**67.5**

Table 5: Performance comparisons of classification accuracy (%) between our developmental networks (WA-CNN) and the previous work for scene classification, fine-grained recognition, and action recognition. We roughly divide the baselines into four types: (1) ImageNet CNNs, which post-process the off-the-shelf CNN or fine-tune it in a standard manner; (2) task customized CNNs, which modify a standard CNN for a particular target task (*e.g.*, for MIT-67, Places-CNN trains a customized CNN on the Places dataset with 400 scene categories [59]); (3) data augmented CNNs, which concatenate features from the ImageNet AlexNet and an additional CNN trained on 100 million Flickr images in a weakly supervised manner [18]; (4) multi-task CNNs, which (approximately) train a CNN jointly from both the source and target tasks. Ours show consistently superior performance and generality for a wide spectrum of tasks.

Figure 5: Top 5 maximally activating images from the SUN-397 validation set for six FC_a units of the depth augmented network (DA-CNN). Each row of 5 images in the left and right columns corresponds to a unit, respectively, which is well aligned to a scene-level concept for the target task, *e.g.*, auditorium and veterinary room in the first row.

classification accuracy. While the different variations of our networks outperform these baselines, we mainly focus on the width augmented networks (WA-CNN).

Tasks and datasets: We evaluate on standard benchmark datasets for scene classification: MIT-67 [44], for fine-grained recognition: Caltech-UCSD Birds (CUB) 200-2011 [48] and Oxford 102 Flowers [27], and for action recognition: Stanford-40 actions [54]. These datasets are widely used for evaluating the CNN transferability [3], and we consider their diversity and coverage of novel categories. We follow the standard experimental setup (*e.g.*, the train/test splits) for these datasets.

Baselines: While comparing with classic fine-tuning is the fairest comparison, to show the superiority of our approach, we also compare against other baselines that are specifically designed for certain tasks. For a fair comparison, we focus on the approaches that use single scale AlexNet CNNs. Importantly, our approach can be also combined with other CNN variations (*e.g.*, VGG-CNN [37], multi-scale CNN [12, 53]) for further improvement.

Table 5 shows that our approach achieves state-of-the-art performance on these challenging benchmark datasets and significantly outperforms classic fine-tuning by a large margin. In contrast to task customized CNNs that are only

Dataset	CNN	WA-CNN-scratch	WA-CNN-grow (Ours)
ImageNet	56.9	57.6	**57.8**

Table 6: Performance comparisons of classification accuracy (%) on the source dataset between a standard AlexNet (CNN), a wide AlexNet trained from scratch (WA-CNN-scratch), and a wide network trained progressively by fine-tuning on the source task itself (WA-CNN-grow). Progressive learning appears to help even on the source task.

Dataset	CNN-FT	WA-CNN-ori	WA-CNN-grow (Ours)
MIT-67	61.2	62.3	**66.3**
CUB200-2011	62.9	63.2	**69.0**

Table 7: Performance comparisons of classification accuracy (%) on the target datasets between standard fine-tuning of a standard AlexNet (CNN-FT), standard fine-tuning of a wide AlexNet (WA-CNN-ori), and fine-tuning by progressive widening of a standard AlexNet (WA-CNN-grow). With the same model capacity, WA-CNN-grow significantly outperforms WA-CNN-ori. See Figure 6 for a discussion.

suitable for particular tasks and categories, the consistently superior performance of our approach suggests that it is generic for a wide spectrum of tasks.

5.4. A Single Universal Higher Capacity Model?

One interesting question is that our results could imply that standard models should have used higher capacity even for the source task (*e.g*., ImageNet). To examine this, we explore progressive widening of AlexNet (WA-CNN). Specifically, in the source domain, Table 6 shows that progressive widening of a network outperforms a fixed wide network trained from scratch. More importantly, in the target domain, Table 7 shows that our progressive widening *significantly outperforms* fine-tuning a fixed wide network.

Cooperative learning: Figure 6 and Figure 7 provide an in-depth analysis of the cooperative learning behavior between the pre-existing and new units and show that developmental learning appears to regularize networks in a manner that encourages diversity of units.

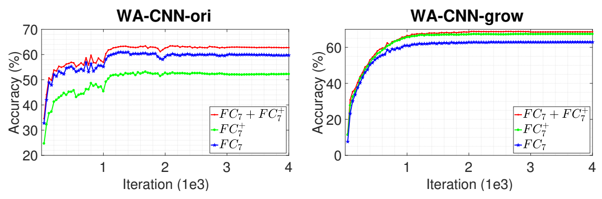


Figure 6: Learning curves of separate FC_7 and FC_7^+ and their combination for WA-CNN on the CUB200-2011 test set. Left and Right show different learning behaviors: the FC_7^+ curve is *below* the FC_7 curve for WA-CNN-ori, and *above* for WA-CNN-grow. Units in WA-CNN-ori appear to *overly-specialize* to the source, while the new units in WA-CNN-grow appear to be *diverse experts* better tuned for the novel target task. Interestingly, these experts allow for better adaptation of pre-existing and new units (Figure 7).

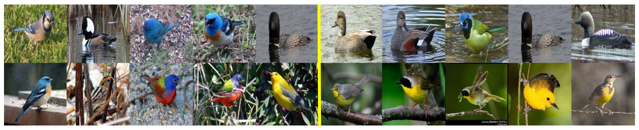


Figure 7: Top 5 maximally activating CUB200-2011 images for a representative FC_7 unit (1st row) and an FC_7^+ unit (2nd row). Each row of images corresponds to a common unit from two networks: WA-CNN-ori (left) and WA-CNN-grow (right). Compared to WA-CNN-ori, WA-CNN-grow facilitates the adaptation of pre-existing and new units towards the novel task by capturing discriminative patterns (top: birds in water; bottom: birds with yellow belly).

Continual transfer across multiple tasks: Our approach is in particular suitable for *continual, smooth* transfer across multiple tasks since we are able to cumulatively increase model capacity as demonstrated in Table 8.

Scenario	WA-CNN (Ours)		Baselines	
	ImageNet→MIT67	ImageNet→SUN→MIT67	Places [59]	ImageNet-VGG [22]
Acc(%)	66.3	**79.3**	68.2	74.0

Table 8: Through progressive growing via SUN-397, a widened AlexNet significantly improves the performance on MIT-67, and even outperforms fine-tuning a Places AlexNet that is directly trained on the Places dataset with 400 scene categories [59] and fine-tuning a fixed ImageNet VGG16 with higher capacity by a large margin.

6. Conclusions

We have performed an in-depth study of the ubiquitous practice of fine-tuning CNNs. By analyzing what changes in a network and how, we conclude that increasing model capacity significantly helps existing units better adapt and specialize to the target task. We analyze both depth and width augmented networks, and conclude that they are useful for fine-tuning, with a slight but consistent benefit for widening. A practical issue is that newly added units should have a pace of learning that is comparable to the pre-existing units. We provide a normalization and scaling technique that ensures this. Finally, we present several state-of-the-art results on benchmark datasets that show the benefit of increasing model capacity. Our conclusions support a developmental view of CNN optimization, in which model capacity is progressively grown throughout a lifelong learning process when learning from continuously evolving data streams and tasks.

Acknowledgments: We thank Liangyan Gui for valuable and insightful discussions. This work was supported in part by ONR MURI N000141612007 and U.S. Army Research Laboratory (ARL) under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016. DR was supported by NSF Grant 1618903 and Google. We also thank NVIDIA for donating GPUs and AWS Cloud Credits for Research program.

References

[1] P. Agrawal, R. Girshick, and J. Malik. Analyzing the performance of multilayer neural networks for object recognition. In *ECCV*, 2014.

[2] R. K. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6:1817–1853, 2005.

[3] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Factors of transferability for a generic ConvNet representation. *TPAMI*, 38(9):1790–1802, 2016.

[4] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. In *CVPR Workshops*, 2015.

[5] L. Bertinetto, J. F. Henriques, J. Valmadre, P. Torr, and A. Vedaldi. Learning feed-forward one-shot learners. In *NIPS*, 2016.

[6] B. Chu, V. Madhavan, O. Beijbom, J. Hoffman, and T. Darrell. Best practices for fine-tuning visual classifiers to new domains. In *ECCV Workshops*, 2016.

[7] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[8] T. Furlanello, J. Zhao, A. M. Saxe, L. Itti, and B. S. Tjan. Active long term memory networks. arXiv preprint arXiv:1606.02355, 2016.

[9] R. Girshick. Fast R-CNN. In *ICCV*, 2015.

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

[11] S. Gupta, J. Hoffman, and J. Malik. Cross modal distillation for supervision transfer. In *CVPR*, 2016.

[12] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015.

[13] B. Hariharan and R. Girshick. Low-shot visual object recognition. arXiv preprint arXiv:1606.02819, 2016.

[14] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[15] M. Huh, P. Agrawal, and A. A. Efros. What makes ImageNet good for transfer learning? arXiv preprint arXiv:1608.08614, 2016.

[16] W. Huitt and J. Hummel. Piaget's theory of cognitive development. *Educational psychology interactive*, 3(2):1–5, 2003.

[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 MM*, 2014.

[18] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache. Learning visual features from large weakly supervised data. In *ECCV*, 2016.

[19] G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Workshops*, 2015.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[21] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[22] Z. Li and D. Hoiem. Learning without forgetting. In *ECCV*, 2016.

[23] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. In *ICLR workshop*, 2016.

[24] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016.

[25] T. M. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. D. Mishra, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. A. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. Never-ending learning. In *AAAI*, 2015.

[26] C. A. Nelson, M. L. Collins, and M. Luciana. *Handbook of developmental cognitive neuroscience*. MIT Press, 2001.

[27] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

[28] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

[29] M. Pickett, R. Al-Rfou, L. Shao, and C. Tar. A growing long-term episodic & semantic memory. In *NIPS Workshops*, 2016.

[30] Q. Qian, R. Jin, S. Zhu, and Y. Lin. Fine-grained visual categorization via multi-stage metric learning. In *CVPR*, 2015.

[31] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017.

[32] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *CVPR Workshops*, 2014.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[34] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

[35] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap. One-shot learning with memory-augmented neural networks. In *ICML*, 2016.

[36] O. Sigaud and A. Droniou. Towards deep developmental learning. *TCDS*, 8(2):90–114, 2016.

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

[38] A. V. Terekhov, G. Montone, and J. K. O'Regan. Knowledge transfer in deep block-modular neural networks. In *Conference on Biomimetic and Biohybrid Systems*, 2015.

[39] C. Tessler, S. Givony, T. Zahavy, D. J. Mankowitz, and S. Mannor. A deep hierarchical approach to lifelong learning in minecraft. *arXiv preprint arXiv:1604.07255*, 2016.

[40] S. Thrun. Is learning the n-th thing any easier than learning the first? In *NIPS*, 1996.

[41] S. Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer, 1998.

[42] S. Thrun and J. O'Sullivan. Clustering learning tasks and the selective cross-task transfer of knowledge. In *Learning to learn*, pages 235–257. Springer, 1998.

[43] T. Tommasi, F. Orabona, and B. Caputo. Learning categories from few examples with multi model knowledge transfer. *TPAMI*, 36(5):928–941, 2014.

[44] A. Torralba and A. Quattoni. Recognizing indoor scenes. In *CVPR*, 2009.

[45] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015.

[46] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 9:2579–2605, 2008.

[47] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *NIPS*, 2016.

[48] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

[49] Y.-X. Wang and M. Hebert. Model recommendation: Generating object detectors from few samples. In *CVPR*, 2015.

[50] Y.-X. Wang and M. Hebert. Learning from small sample sets by combining unsupervised meta-training with CNNs. In *NIPS*, 2016.

[51] Y.-X. Wang and M. Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, 2016.

[52] J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva. SUN database: Exploring a large collection of scene categories. *IJCV*, 119(1):3–22, 2016.

[53] S. Yang and D. Ramanan. Multi-scale recognition with DAG-CNNs. In *ICCV*, 2015.

[54] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *ICCV*, 2011.

[55] D. Yoo, S. Park, J.-Y. Lee, and S. Kweon. Multi-scale pyramid pooling for deep convolutional representation. In *CVPR Workshops*, 2015.

[56] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014.

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

[58] L. Zheng, Y. Zhao, S. Wang, J. Wang, and Q. Tian. Good practice in CNN feature transfer. arXiv preprint arXiv:1604.00133, 2016.

[59] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.