Deep Connected Attention Learning for Image Classification

Anonymous CVPR submission

Paper ID ****

Abstract

Convolutional Neural Networks (CNNs) have been largely boosted by introducing self-attention mechanism. By biasing informative features in each convolutional block, attention mechanism is able to boost representation power of CNNs with fewer parameters. Although various attention block designs are proposed to enhance deep CNNs every year, small efforts were made to enhance attention blocks in the literature. In this paper, we propose a novel module to boost attention blocks, which is generic, not limited to a particular attention design. Our new module connects adjacent attention blocks to boost attention learning ability. This design could better explore the capacity of attention blocks by training attention blocks jointly instead of independently in a deep CNN. The enhanced network is called Deep Connected Attention Network (DCA). DCA can be integrated with various attention blocks regardless of the choice of base networks. Experiments on ImageNet show that our approach outperforms state-of-the-art models using a similar number of parameters and FLOPs. All code and models will be made publicly available.

1. Introduction

Self-Attention mechanism has shown its efficiency to boost deep convolutional neural networks. For each convolutional layer, attention mechanism tells the neural network where and what to pay attention to by biasing solid features. Stacking multiple attention guided convolutional layers, an attention mechanism based deep neural network is built. Recent efforts reveal that attention mechanism enhanced neural network can outperform a deeper one with fewer layers and parameters [38, 15].

The existing self-attention blocks are usually integrated on each convolutional block; thus each attention block learns weights independently and is only related to its current convolutional block. Directly using unrelated attention blocks in a deep CNN models could not efficiently capture the intrinsic properties about what to pay attention to and the focus in an attention block may vary dramatically from others.

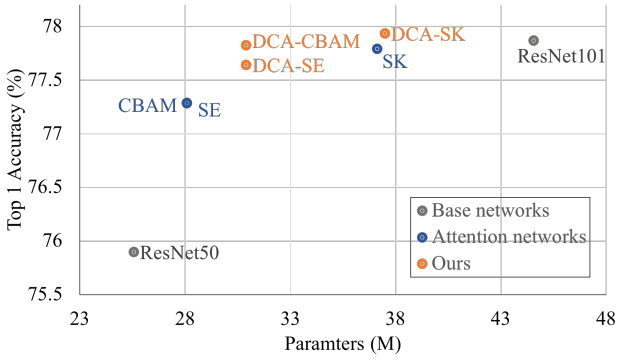

Figure 1. **Top1 classification accuracy and Parameters** on ImageNet 2012. We select several attention-based methods on ResNet50 and compare with our DCA boosted models. The vanilla attention models are marked in blue and our DCA boosted models are marked in orange. CBAM-ResNet50 and SE-ResNet50 achieve similar results, 77.2840% and 77.2877% respectively. The results show that the DCA enhanced models outperform the corresponding vanilla attention models with marginal parameter increases. More detailed results can be found in Table 3.4.

[Xu: To be replaced.] That is, simply integrating attention block to intermediate feature maps is not sufficient enough for attention learning. [Song: This is a strong argument, but there is little support to it. It could be moved to the last paragraph in this section.]

To tackle this issue, we propose to connect adjacent attention blocks by using residual connections. The goal is to gather precedent attention maps and transmit them to the next attention block, thus making all attention blocks conjoint and further extract a weakly-supervised attention. Since our DCA is an enhancement of vanilla attention blocks, as a result, it can be easily integrated into various attention-based networks, such as SENet [15], CBAM [40] and SKNet[19], regardless of the choice of base network.

In literature, various dimensions of attention have been proposed, including channel dependencies [15, 14], spatial correlations[17, 39, 43], and kernel sizes [19]. It is worth noticing that not all attention blocks only focus on one at-

tention dimension. Recent works reveal that inferring attention maps along multiple attention dimensions can achieve better results [25, 40, 38]. For multiple attention dimensions, we embrace the hypothesis used in [7, 13], that is it is preferable to sufficiently decouple correlations in different dimensions instead of mapping them jointly, and apply it to attention blocks. Following this idea, we decouple comprised attention dimensions and connect attention blocks in each individual attention dimension to keep them uncorrelated. This strategy has two advantages: 1) decoupling attention dimensions makes each one focuses more on a specific dimension, which eases the training process compared to mapping all attention dimensions jointly. 2) It assures attention blocks to be lightweight with minimal computation overhead added in attention blocks. By connecting blocks along each attention dimension and decoupling dimensions, we build a Deep Connected Attention module (DCA module). DCA aims to achieve better attention extraction of intermediate features with only a slight increase in model parameters, which are used to match tensor sizes between different stages in a network.

DCA module is conceptually simple and empirically powerful. We apply DCA to multiple state-of-the-art attention blocks and a range of base networks and evaluate its performance on image classification tasks. Without bells and whistles, our DCA enhanced networks outperform all of the original counterparts. Figure 1 shows a comparison between our DCA enhanced models and the corresponding original networks. For ImageNet 2012 classification [27], the DCA enhanced SE-ResNet50 achieves a 1.7428% improvement compared to ResNet50 and a 0.35% improvement compared to SE-ResNet50 with negligible parameters and FLOPs increase.

2. Related Work

Self-attention mechanisms. Attention mechanisms have been prevalent across a large range of tasks, from machine translation [1] in natural language processing to object detection [5] in computer vision. To the best of our knowledge, applying self-attention to explore global dependencies for machine translation was first proposed in [37]. Meanwhile, it draws increasing interests in the filed of computer vision. To investigate the channel interdependencies, SENet [15], GENet [14] and SGENet [18] leverage self-attention for context modeling. NLNet [39] and GCNet [5] introduce self-attention to capture long-range dependencies in non-local operations. BAM [25] and CBAM [40] consider both channel-wise and spatial attentions jointly. Beyond channel and space, SKNet [19] applies self-attention to kernel size selection.

Channel and space decoupling. A traditional convolution layer learns a transformation function, which calculate correlations of features from all *channels* in the *kernel-size space*. To ease the convolution process, Inception networks [34, 35, 33] split a convolution layer to two sub-layers. In particular, Xception [7] explicitly indicates that cross-channel correlations and spatial correlations can be learnt sequentially, rather than in one stage. MobileNet[13, 28, 12] is an example following this direction. To build an effective model for mobile vision applications, MobileNet factorizes a standard convolution into two light-weight convolutions: depth-wise convolution and point-wise convolution. Depth-wise convolution computes space correlation for each channel, while point-wise convolution computes channel correlation for each pixel. Inspired by MobileNet, lots of efforts have been made to develop light-weight CNN architectures by decomposing channel and space correlations [24, 42]. In this paper, we find that decoupling channel and spatial correlations is also suitable for attention mechanisms, such as [25, 40].

Residual connections. The idea of Residual connection comes from [29]. By introducing a shorcut, neural networks are decomposed into biased and centered subnets to accelerate gradient descent. ResNet [10, 11] adds an identity mapping to connect the input and output of each convolutional block, which drastically alleviates its degradation problem [10] and makes convolutional neural networks deeper. Instead of connecting adjacent convolutional blocks, DenseNet [16] connects each block to every other block in a feed-forward fashion. FisheNet [31] connects layers with the same solutions in pursuit of propagating gradient from deep layers to shallow layers. DLA [41] shows that residual connection is a common approach of layer aggregation by iteratively and hierarchically aggregating layers in a network in order to reuse feature maps generated by each layer.

Residual connection has been well studied for base network architectures. Nevertheless, the study of residual connection in attention mechanisms is still new. RANet [38] utilizes residual units in attention block. In [38, 25], residual learning is used in attention modules to facilitate the gradient flow. In contrast to leveraging residual connection *in* attention blocks, we explore residual connections *between* attention blocks.

Connected Attention Recently, there is a growing interest in building connections in attention blocks. In [9], a new network structure named RA-CNN is proposed for fine-grained image recognition. RA-CNN recurrently generates attention region based on current prediction to learn the most discriminative region. By doing so, it obtains an attention region from coarse to fine. In GANet [3], the top attention maps generated by customized background attention block are up-sampled and sent to bottom background attention blocks to guide attention learning.

Different from the recurrent and feed-backward methods, our DCA module enhances attention blocks in a feed-

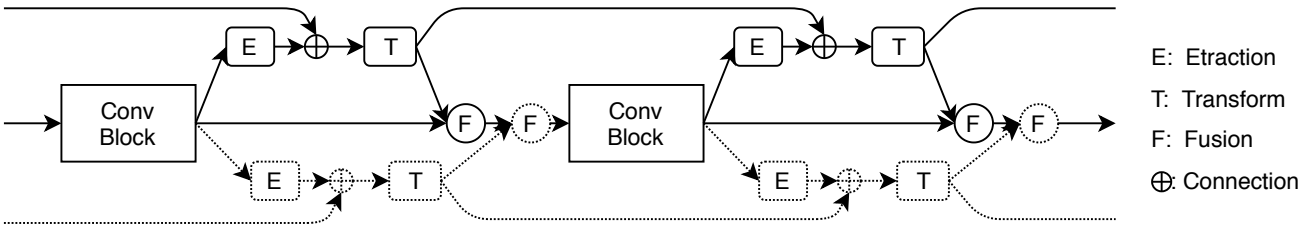

Figure 2. **Architecture of our Connected Attention.** For each attention dimension, we connect the output of a transform module in a previous attention block to the output of an extraction module in the next attention block. In the context of multiple attention dimensions, we connect attentions along each dimension respectively. Dashed elements indicate another attention dimension if exists. Note that we give an example of two attention dimensions case, one can easily deduce situations with more dimensions.

forward fashion which is more computation-friendly and easy to implement.

3. Deep Connected Attention Method

Deep Connected Attention (DCA) augmentation is conceptually simple. We connect adjacent attention blocks to ensure they are trained jointly and prevent attention information from varying dramatically in each step. An attractive feature of connected attention augmentation is that it is not confined to some particular attention method. We showcase that DCA can boost various state-of-the-art attention blocks[15, 40, 19] and the extra computation overhead is negligible.

3.1. Generalizing Self-Attention Blocks

The idea of self-attention is to boost a base network by emphasizing important features and suppressing trivial ones. Usually, we achieve this by adding lightweight attention blocks. Different attention blocks are tailored for different purposes. As a result, their implementations diverse. For instance, SE block composes of two fully connected layers while GC block includes several convolutional layers. Intuitively, it is not easy to provide a common connection schema which is generic enough to cover most attention blocks. To tackle this problem, we study state-of-the-art attention blocks and summarize their processing and components in common.

Inspired by recent works [44, 5, 8] that formulate attention modules and their components (limited to SENet and Non-Local Net), we study various attention modules and develop a generalized attention framework, in which an attention block consists of three components: context extraction, context transformation , and fusion. Extraction severs as a simply feature extractor, transformation transforms the extracted features to a new non-linear attention space, while fusion combines attention and original features together. These components are generic and not confined to particular attention block. Figure 3 exemplifies four well-known attention blocks and their modeling by using the three components. Some attention blocks may include a subset of the components for performance consideration, e.g., GENet-θ $(E8)$ does not have the context extraction. Missing certain component would not affect the function of our DCA module since our DCA module connects outputs of selected components instead of using their internal structures.

Extraction is designed for gathering feature information in a feature map. For a given feature map $\mathbf{X} \in \mathbb{R}^{C\times W\times H}$ produced by a convolutional block, we extract features from $\mathbf{X}$ by an extractor g: $\mathbf{G} = g\left(\mathbf{X}, w_g\right)$, where w_g is parameter for the extraction operation and $\mathbf{G}$ is the output. Extractor g is a flexible building operation and the parameter w_g could not be exist if g is a parameter-free operation, such as the global average pooling in an SE block. [Song: The two parts in this sentence are unrelated. I don't understand why they are put into one sentence. Also, what do "flexible" and the next "flexibility" mean? Need a little more description.][Xu: I want to say that g is not fixed to a certain operation, it can be global pooling or anything else. "Flexible" and "flexibility" means the same thing, the operations of extraction.] The flexibility of g makes $\mathbf{G}$ be arbitrary shape depending on the extraction operation. For instance, SENet and GCNet gather feature map $\mathbf{X}$ as a vector $\mathbf{G} \in \mathbb{R}^{C}$ while the spatial attention module in CBAM gathers feature map to a tensor $\mathbf{G} \in \mathbb{R}^{W\times H}$. [Song: Does "flexible" mean feature maps can be in various dimensions?][Xu: It means the extraction can be any operations, as a result the size of output could be different.]

transformation process the gathered features from Extraction. Formally, we define t as a feature transform, and the output of an attention block can be expressed as $\mathbf{T} = t\left(\mathbf{G}, w_t\right)$. Here w_t denotes parameters used in the transform operation and $\mathbf{T}$ is the output of the extraction module. [Song: Can you provide some examples of transformation operations used in existing attention networks?]

Fusion integrates the attention map with the output of the original convolutional block. An attention guided output

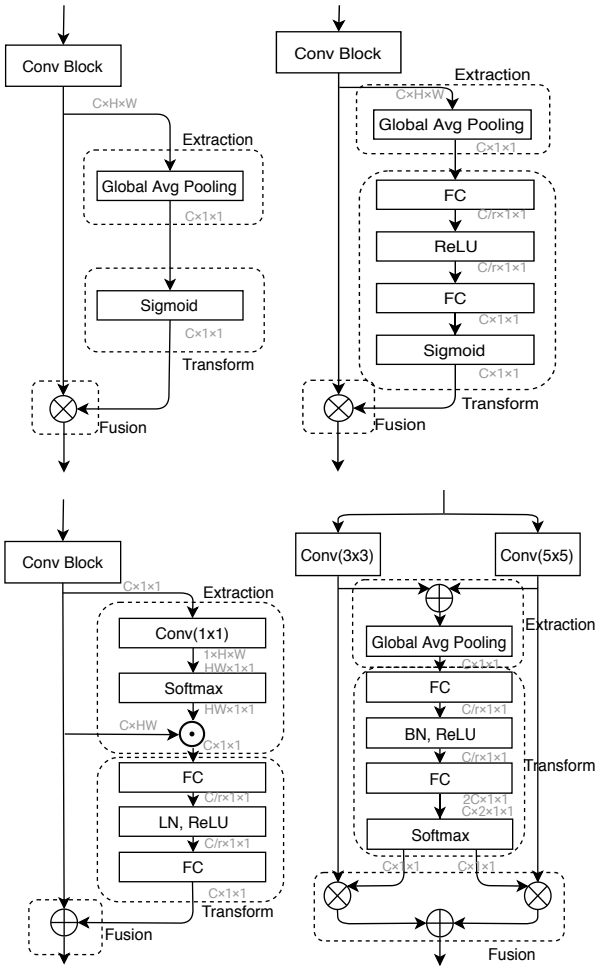

Figure 3. Architectures of attention blocks. We model an attention block by three components: Feature extraction, Transformation and Fusion. **Top-left:** GE-θ^- [14] block. **Top-right:** SE [15] block. **Bottom-left:** GC [5] block. **Bottom-right:** SK [19] block. "$\oplus$" denotes element-wise summation, "$\otimes$" represents element-wise multiplication, and "$\odot$" performs matrix multiplication.

$\mathbf{X}' \in \mathbb{R}^{C\times W\times H}$ can be presented as

$$\mathbf{X}'_i = \mathbf{T}_i \circledast \mathbf{X}_i \tag{1}$$

where i is the index in a feature map and $\circledast$ denotes a fusion function. Basically, $\circledast$ performs element-wise multiplication when $\mathbf{T}$ is re-scaled to the range $(0, 1)$ [19, 15, 40], and summation otherwise[5]. [WARNING: The definition of $\odot$ in sentence overlaps with the $\odot$ in figures. To make them different, I use $\circledast$ to replace.]

3.2. Attention Connection

[WARNING: Unclear definition of previous attention block outputs. Using $\tilde{X}$ or X' to present previous outputs?]

Next, we present a generalized attention connection schema by using the preceding attention components. Regardless of the implementation details, an attention block depicted in Fig. 3 can be modeled as:

$$\mathbf{X}' = \mathbf{X} \odot t\left(g\left(\mathbf{X}, w_g\right), w_t\right) \tag{2}$$

Generally speaking, the transformation component t transforms the extraction result $\mathbf{X}$ to a non-linear space for attention learning. As explained in the previous section, the attention maps generated by the transformation component in an attention block is crucial for attention learning.[Song: What is the connection between the first sentence with the following ones?] To construct a connected attention, we feed the previous attention map to the current transformation component, which merges previous transformation output and the current extraction output together.[Song: The parts before "which" and after are the same. Can we say "which merges previous and current attention maps?] [Xu: They are different. One is the output of extraction, another one is the output of transformation.] This schema leads current attention block to be weakly-supervised by previous attention block. [Song: Need to explain why designed this way? What are the benefits?] [Xu: If this explanation good?] The resulting attention block can be described as:

$$\mathbf{X}' = \mathbf{X} \odot t\left(f\left(\mathbf{G}, \alpha\tilde{\mathbf{T}}\right), w_t\right), \tag{3}$$

where $f(\cdot)$ denotes the connection function, α is a learnable eight, and $\tilde{\mathbf{T}}$ is the attention map generated by the previous attention block. In some cases (e.g., SE block and GE block), $\tilde{\mathbf{T}}$ is scaled to a range of $(0, 1)$. For those kinds of attention blocks, we multiply $\tilde{\mathbf{T}}$ by $\tilde{\mathbf{E}}$ to match the scale, where $\tilde{\mathbf{E}}$ is the output of the Extraction component in the previous Attention. We also note that if α is set to 0, the attention connections are not used and the attention block is reduced to the vanilla network. That is the vanilla network is a special case of our DCA enhanced attention network.

Next, we present three schemas that instantiate the connection function $f(\cdot)$.

Direct Connection. A simple design of $f(\cdot)$ is adding the two terms directly. The connection function can then be presented as:

$$f\left(\mathbf{G}_i, \alpha\tilde{\mathbf{T}}_i\right) = \mathbf{G}_i + \alpha\tilde{\mathbf{T}}_i, \tag{4}$$

where i is the index of a feature. Equation (4) can be considered as an enhancement of $\mathbf{G}$.

Weighted Connection. Direct connection can be augmented by using weighted summation. To avoid introducing extra parameters, we can calculate weights from $\mathbf{G}$ and $\alpha\tilde{\mathbf{T}}$. The connection function is represented as

$$f\left(\mathbf{G}_i, \alpha\tilde{\mathbf{T}}_i\right) = \frac{\mathbf{G}_i}{\mathbf{G}_i + \alpha\tilde{\mathbf{T}}_i}\mathbf{G}_i + \frac{\alpha\tilde{\mathbf{T}}_i}{\mathbf{G}_i + \alpha\tilde{\mathbf{T}}_i}\alpha\tilde{\mathbf{T}}_i. \tag{5}$$

Softmax Connection. Softmax provides another way to calculate weights, which makes the connection function become

$$f\left(\mathbf{G}_i, \alpha\tilde{\mathbf{T}}_i\right) = \frac{e^{-\mathbf{G}_i}}{e^{-\mathbf{G}_i} + e^{-\alpha\tilde{\mathbf{T}}_i}}\mathbf{G}_i + \frac{e^{-\alpha\tilde{\mathbf{T}}_i}}{e^{-\mathbf{G}_i} + e^{-\alpha\tilde{\mathbf{T}}_i}}\alpha\tilde{\mathbf{T}}_i. \quad (6)$$

Compared to the weighted connection, the softmax connection is more robust and less sensitive to trivial features. However, it takes previous attention $\tilde{\mathbf{T}}$ into consideration even if α is 0. [Song: If α = 0, the three connections are the same. How does this statement come from?] [WARNING: No. Even if $\alpha = 0$, softmax is different from previous two. $\frac{e^{-\mathbf{G}_i}}{e^{-\mathbf{G}_i}+e^{-\alpha\tilde{\mathbf{T}}_i}}$ is not equal to $\frac{e^{-\mathbf{G}_i}}{e^{-\mathbf{G}_i}}$ and is not equal to 1] Experimental results from ablation studies (presented in Table 2) show that the difference between these connection schemas is not significant, indicating that the performance improvement [Song: What performance gain?][Xu: If i can change it to performance improvement] comes more from connections between attention blocks than the specific form of the connection function.

Thus, we use a direct connection in the following design. Similar to [10, 16], we keep previous attention maps clean in connections without embedding to new space.[Song: How to keep the maps clean? What does "embedding to new space" mean?] [Xu: That means we do not use some new layers (like conv, fc) to convert previous attention,this is what we called clean. If we use, that is called embedding.]

In this paper, we use summation operation for our connection, and not not consider concatenation operation because it would change the shape of $\mathbf{E}$ and affect the original structure of attention blocks. A combination of concatenation and fully-connected layer (for channel dimension) or convolutional layer (for spatial dimension) is feasible for adopting concatenation and keeping original shape of E simultaneously. In this case, it is in a similar manner as weighted connection and softmax connection, and the only difference is calculation of weight. As a result, we do not take this method into consideration. [Song: What is the connection between this paragraph (concatenation) with the three connection functions presented in this section?] [Xu: Emmm, looks like no connections. I just want to specify how we implement it. So do we need to delete this?]

3.3. Size Matching

Feature maps produced at different stages in an CNN model may have different size. Thus, the size of the corresponding attention maps may vary as well. such a mismatch leads our DCA module impossible to be simply applied between two stages. In order to tackle this issue, we match attention maps along the channel and spatial dimensions.

For channel, we match size using a fully-connected layer to convert C'channels to C channels, where C' and C are the channel number of previous and current feature maps, respectively. [Song: Which map has C' dimension and which one has C dimension? Need to understand why converting C' to C.] [WARNING: Referring the WARNING message under section 3.2. Unclear definition of previous outputs. To be revised.] [Xu: Updated. If I can change the sentence like this?] Omitting biases for clarity, parameters introduced for channel size matching is $C' \times C$. To further reduce parameter burdens in attention connections, a simple yet effective strategy is to replace the direct fully-connected layer by two lightweight fully-connected layers; the output sizes are C'/r and C' respectively, where r is reduction ratio. This significantly reduce the number of parameters without decreasing performance. The detail influence of channel size matching strategy can be found in Table 4. In all our experiments, we use one fully-connected layer to match channel size for simplicity, unless otherwise noted.

To match the spatial resolution, a simple yet effective strategy is to adopt an average-pooling layer. We set stride and receptive field size to resolution reduction scale. Max-pooling also works well in our method, but it only computes most important weights instead of the whole attention maps. Besides pooling operation, an alternative solution is convolutional operation. However, we argue that it is not suitable for our propose since it introduce many parameters and can not generalize well. Detail ablations on spatial resolution size matching can be found in Table 5.

3.4. Multi-dimensional attention connection

We note that some attention blocks focus on more than one attention dimension. For instance, BAM[25] and CBAM[40] infer attention map along both channel and spatial dimensions. Inspired by Xception [7] and MobileNet [13, 28], we design attention connection for one attention dimension at a time. To build a multi-dimensional attention block, we connect attention maps along each dimension and assure connections in different dimensions are independent from each other. This decoupling of attention connections has two advantages: 1) it reduces the number of parameters and computation overhead; [Song: By coordinating the attention connections of two dimensions, parameter volume and computation time may decrease comparing to having 2 sets of similar parameters and doing similar computation twice.][Xu: Sorry I didn't understand this. Decouple differnt dimension would reduce parameters and computations. For instance, a depthwise convolution layer (which didn't map channel and spatial together) would have fewer parameters and FLOPs than a norm convolutional layer (which mapped channel and spatial together).] 2) each dimension can focus on its intrinsic property. As shown in Fig. 2, the attention connections on the top are decoupled from the attention connections at the bottom. [Song: In Figure 2, there are two "F" before a Conv Block with each taking output from a dimension. Is there any sequential relation between them, or are they independent? Would one "F" taking outputs from both dimensions work?][Xu: I also considered this questions. I don't know how to plot it. As I think it should follows the designs of the vanilla attention module. For example, CBAM considers channel first and then spatial.] [WARNING: If we should make it more clear?]

4. Performance Evaluation

We have conducted a series of experiments on the ImageNet 2012 classification dataset [27], which contains 1.28 million training images and 50k validation images. We train a number of selected models on the training set and measure the single-crop (224×224 pixels) top-1 and top-5 accuracy on the validation set. Our implementations are based on PyTorch [26]. For training ResNet and variants, we use the setup in [10]. We train models for 100 epochs on 8 TESLA V100 GPUs with 32 images per GPU (The batch size is 256). All models are trained using synchronous

	re-implement				DCANet			
	Top-1 acc.	Top-5 acc.	GFLOPs	Params	Top-1 acc.	Top-5 acc.	GFLOPs	Params
ResNet50[10]	75.8974	92.7224	4.122	25.56M	-	-	-	-
SE-ResNet50[15]	77.2877	93.6478	4.130	28.09M	**77.6402**	93.7400	4.133	30.90M
SK-ResNet50 [19]	77.7885	93.758	5.979	37.12M	**77.9357**	93.8955	5.982	37.48M
GE-ResNet50[14]	77.1146	93.7101	4.143	26.06M	77.3531	93.7942	4.455	28.84M
GC-ResNet50[5]	74.8944	92.2812	4.130	28.11M	**75.4245**	92.4725	4.131	28.63M
CBAM-ResNet50[40]	77.2840	93.6005	4.139	28.09M	**77.8258**	93.7167	4.142	30.90M
Mnas1_0 [36]	71.7195	90.3201	0.330	4.38M	-	-	-	-
SE-Mnas1_0 [15]	69.6907	89.1203	0.331	4.42M	**71.7574**	90.3978	0.331	4.48M
GE-Mnas1_0 [14]	70.8394	89.7341	0.332	4.44M	71.5427	90.5752	0.348	4.53M
CBAM-Mnas1_0 [40]	69.1287	88.9210	0.332	4.42M	71.0024	89.7832	0.333	4.56M
MoibleNetV2[28]	71.032	90.067	0.320	3.50M	-	-	-	-
SE-MoibleNetV2[15]	72.0484	90.5812	0.321	3.56M	**72.0623**	90.5991	0.321	3.60M
SK-MoibleNetV2 [19]	74.0456	91.8487	0.354	5.28M	**74.4503**	91.8527	0.355	5.91M
GE-MoibleNetV2 [14]	70.8367	89.7341	0.332	3.55M	**71.1043**	89.9781	0.331	3.63M
CBAM-MoibleNetV2 [40]	71.9069	90.5114	0.324	3.57M	**71.9428**	90.4377	0.324	3.65M

Table 1. Single crop classification accuracy (%) on ImageNet validation set. We re-train all models using PyTorch framework and report results in "re-implement" column. The corresponding DCANets are presented in "DCANet" column. The best performances are marked as **bold**. "-" means no experiments since our DCA module is designed for enhancing attention blocks, which are not existent in base networks.

SGD with Nesterov momentum [32] of 0.9 and a weight decay of 0.0001. The learning rate is set to 0.1 initially and lowered by a factor of 10 every 30 epochs. For lightweight models like MnasNet and MobileNetV2, we adopt cosine annealing learning rate scheduler [23] and train the models for 150 epochs with 64 images per GPU.

4.1. Classification on ImageNet

We apply our DCA module to a number of state-of-the-art attention blocks, including SE-block[15], SK-block[19], GE-block[14], GC-block[5], and CBAM-block[40]. We use ResNet50[10] as the base network for illustration. As lightweight CNN models attract increasing attention due to their efficiency on mobile devices, we also experiment on lightweight models to evaluate the performance of DCANet. Specifically, we select a representative lightweight model, i.e., MobileNetV2 [28], as a base model. Additionally, we select MnasNet1_0 [36] as an example method on neural architecture search. We add DCA connections to the original attention networks and measure the performance improvement for image classification.

Table 3.4 shows the experimental results. In the table, we can see that adding the DCA module improves the classification performance compared to all vanilla attention models. Note that we compare with corresponding attention based networks, which is better than the base networks. Among the tested networks, DCA-CBAM-ResNet50 improves the top-1 accuracy by 0.51% compared with CBAM-ResNet50, and DCA-SE-ResNet50 improves the top-1 accuracy by 0.36% compared with SE-ResNet50. The computation overhead is comparable. The improvement demonstrates the efficiency of the DCA module. Besides, we find that Global Context (GC) [5] module significantly decreases the performance of ResNet50 and cost more time to converge normally. The reason is that we trained GC-ResNet50 from scratch as common while the training strategy in original paper of GCNet is to fine tune GC-ResNet50 based on a pre-trained ResNet50. Hence, we do not experiment GC module on Mnas1_0 and MobileNetV2.

Somewhat surprisingly, we notice that directly applying attention blocks to MnasNet always decreases the performance, no matter what the attention block is. One possible reason is that MnasNet1_0 is a model searched from pre-defined network searching space while attention block is not included in the searching space of MnasNet. However, when integrating our DCA module, MnasNet with attention blocks is able to achieve comparable performance with original Mnas1_0. We argue that this is because our DCA module adaptively regulate the influence of attention blocks. Recall formulation 3, α is a learnable weights. Thus, it is able to dynamically balance the impacts of previous attention block and current one, even suppress later attention models.

4.2. Ablation Study

In this subsection, we report ablation experiments on ImageNet dataset to thoroughly investigate the efficiency of our DCA module.

Connection Schema. As described in Section 3.2, we introduce three schemas for connecting previous attention map to current attention blocks. To fully investigate the effect of each connection schema, we apply these versions of DCA module to SE-ResNet50 and report results in Table 2. Across all three connection schemas we observe that they all outperform vanilla SE-ResNet50, suggesting that the performance improvement essentially comes from the connections between attention blocks rather than connection fashion. Meanwhile, we notice that direct connection performs slightly better than softmax and weighted connections and the last two variants perform approximately. It demonstrates that DCA module

Model	Top-1.	Top-5.	GFLOPs	Params
SE-ResNet50	77.2877	93.6478	4.130	28.09M
Direct	**77.6402**	93.7400	4.133	30.90M
Softmax	77.4853	93.6783	4.133	30.90M
Weighted	77.4693	93.6703	4.133	30.90M

Table 2. Performance of different connection schemas. The baseline is SE-ResNet50.

Model	Top-1.	Top-5.	GFLOPs	Params
ResNet50	75.8974	92.7224	4.122	25.56M
SE	77.2877	93.6478	4.130	28.09M
DCA-SE	**77.6402**	93.74	4.133	30.90M
ResNet101	77.8659	93.7998	7.849	44.55M
SE	78.3353	94.1606	7.863	49.33M
DCA-SE	**78.4399**	94.2228	7.866	52.17M
ResNet152*	78.4	94.2	11.581	60.19M
SE	78.5179	94.0729	11.600	66.82M
DCA-SE	**78.5517**	94.1566	11.603	69.70M

Table 3. Comparison among ResNet, SE-ResNet and DCA-SE-ResNet with different depths. * indicates the results presented in AANet[2].

should add no restrictions to the weights of previous attention and current feature extractor and they don't compete with each other.

Network Depth Our work connects attention blocks in a deep CNN model. As the model goes deeper, more attention blocks are incorporated and naturally more connections between attention blocks would be added. More attention blocks have been demonstrated helpful for better performance, but if more attention block connections also improve performance is not explored. In order to tackle this issue, we investigate the effect of network depth on DCANet. Since SENet is a simple yet effective self-attention module, we select SENet as example of self-attention module and ResNet as base network. We first compare SE-ResNet50 against DCA boosted counterpart and then increase the depth from 50 to 101 and 152, respectively. Table 3 shows the result of deep network.

Across all three comparisons of different depth we observe that our DCA-SE-ResNet always outperforms SE-ResNet and base ResNet. However, the gain of performance improvement becomes smaller and smaller as the depth increases. Similar phenomena are also shown when applying SE module to ResNet. The key insight behind these phenomena is that applying attention modules to deep networks would obtain less gain of performance improvement than applying to shallow networks since the deep base networks is able to achieve satisfying results.

Size matching. Next, we disentangle the influence of size matching on DCA module performance. We note that the shape of attention map between stages may different from each other, leading them impossible to be merged directly. When the channel number increases and spatial resolution decreases, we consider these solutions for size matching: we adopt fully-connected layers for channel number matching and pooling operation for spatial resolution matching.

For channel number matching, we evaluate the performance of several channel number matching methods descried in Section 3.3. Specifically, we test several reduction ratio values in our implements. The studies are based on SE-ResNet50 due to its pure concerns on channel dependencies. As Section 3.3 describes, we use "1 FC" to present directly matching and "2 FC" to present two lightweight fully-connected layers where r is reduction rate.

Model	Top-1.	Top-5.	GFLOPs	Params
Se-ResNet50	77.2877	93.6478	4.130	28.09M
1 FC	**77.6402**	93.74	4.133	30.90M
2 FC (r=16)	77.5541	93.7720	4.131	**28.65M**
2 FC (r=8)	77.497	93.708	4.132	29.87M
2 FC (r=4)	77.419	93.736	4.134	32.31M

Table 4. Performance of different channel matching methods based on SE-ResNet50.

Model	Top-1.	Top-5.	GFLOPs	Params
CBAM	77.2840	93.6005	4.139	28.09M
Max Pooling	77.4299	93.7679	4.139	28.09M
Avg Pooling	**77.5789**	**93.7978**	4.139	28.09M

Table 5. Performance of different spatial matching methods based on CBAM-ResNet50. We connected spatial attention module in CBAM module and keep channel attention unchanged.

Table 4 comprehensively summarizes performances of various channel matching strategies. Directly applying one fully-connected layer is able to achieve the best performance while setting reduction to 16 in 2 fully-connected layers is able to reduce parameters with minimal accuracy loss.

For spatial resolution, we simply adopt average pooling to reduce resolution. We also compare to max pooling operation and present results in Table 5. We notice that max pooling performed slightly inferior compared to average pooling, indicating that all attention features should be passed to next stage.

Multiple Attention dimensions Thus far we have empirically evaluated DCANet on single attention dimension. It should be noted that not all self-attention modules only consider one attention dimension, some attention modules like CBAM and BAM take both channel dependencies and spatial correlations into consideration. Rather than only boosting one dimension attention module, our DCANet is capable to boost a attention module which integrates multiple attention dimensions. For illustration, we use CBAM-ResNet50 as a baseline since CBAM module integrates channel-wise attention and spatial attention. BAM is not our prime choice since CBAM can be consider as an improved version of BAM. We first integrate DCA module at each attention dimension respectively, then we integrate at both dimensions.

Table 6 shows the results of DCA enhancement on multiple dimensions. Consistent with our expectations, applying DCA enhancement will constantly improve the accuracy along any dimension. From Table 6, we have observed two interesting phenomenons. First, we found that the improvements are slightly different, applying DCA module to channel-wise attention achieve 0.2% improvement gain than applying to spatial attention. Besides, the parameters increase of channel-wise attention is a bit large than spatial attention. Whether the improvement gap comes from the extra parameters or channel intrinsic properties is not revealed. Second, Compared to applying DCA module to single at-

Model	Top-1 err.	Top-5 err.	GFLOPs	Params
CBAM	77.2840	93.6005	4.139	28.09M
DCA-C	77.7902	93.7079	4.142	30.90M
DCA-S	77.5789	93.7978	4.139	28.09M
DCA-All	**77.8258**	93.7167	4.142	30.90M

Table 6. Performances of applying DCA moduel to different attention dimensions in CBAM-ResNet50. Here "CBAM" indicates base CBAM-ResNet50. We use DCA-C/DCA-S to present applying DCA module on channel/spatial attention and DCA-All indicates we apply DCA module on both attention dimensions for CBAM-ResNet50.

tention dimension, applying DCA module to all attention dimensions would certainly achieve better performance. We note that improvement of enhancing both is greater than enhancing any single dimension but smaller than the summation of improvements along these two attention dimensions.

4.3. Object detection on MS COCO

We further evaluate the use of DCA module in object detection to showcase its generality on other vision tasks. We present mean average precision of bounding box detection on the challenging COCO 2017 dataset [22]. The input images are resized to make the shorter side to be 800 pixels [20]. We follow the default settings in [6] and train all models with a total of 16 images per mini-batch (2 images per GPU). We employ two state-of-the-art detectors, RetinaNet[21] and Cascade R-CNN [4], as our detector, with SE-ResNet50 variants and GC-ResNet50 variants as corresponding backbones respectively. All backbones are pre-trained on ImageNet as presented in Table 3.4. The detection models are trained for 24 epochs using synchronized SGD with a weight decay of 0.0001 and momentum of 0.9. The learning rate is set to 0.02 for Cascade R-CNN and 0.01 for RetinaNet as previous work. We decay the learning rate by a factor of 10 at 18th and 22th epochs.

The results are reported in Table 7. We have observed that our method almost achieves the best performance at all IoU threshold values and all object scales. Note that we only replaced backbone models to our methods, which means that all the performance improvements are from our DCA module. The promising results for object detection indicates that our proposed method rather than only improves the performance of classification, but also is in favor of other vision tasks.

4.4. Visualization

In order to verify the basic idea of deep connected attention module that it is able to adjust attention progressive for a better feature extraction, we train CBAM-SENet50 and DCA enhanced counterpart for 100 epochs on ImageNet. For DCA enhanced CBAM-ResNet50, we only connect attention modules along spatial dimension in CBAM module, which is more convenient for visualization. We visualize intermediate outputs leveraging Grad-CAM [30].

Fig. 4 shows the comparison between vanilla CBAM-ResNet50 and DCA enhanced variant. As listed in 5 columns, the pictures from left to right are original images, heatmaps of stage1, 2, 3 and 4 respectively. Top line is generated by CBAM-ResNet50 while bottom line is generated by our DCA module enhanced CBAM-ResNet50 along spatial dimension. **We notice that our DCA module consistently focus on several key parts in an image and the most discriminative parts merely vary little among all intermediate features**. In stark contrast to our DCA enhanced model, feature map response from the vanilla CBAM-ResNet50 changes drastically as shown in top line. Interestingly, stage 3 feature map response in top line is trivial while it is unambiguously overlapped with key parts of the dog in bottom line, showing that DCA module is able to boost learning power of attention module and feature extraction ability of CNN models.

4.5. Analysis

To better understand how previous attention contributes, we analyze the contribution ratio in each attention connection. We calculate the contribution ratio by comparing the weight of the previous attention to the sum of the weights of the two. By doing so, the contribution rate of previous attention will be fixed in the range of $[0, 1]$. We take ResNet50 as a base network and comprehensively analyze DCA module application on SE, SK, GC and CBAM blocks. We present the contribution ratio in each connection in Figure 5.

At a first glance, we have seen that the contribution ratios are different from each other, showing that each attention module has its own intrinsic properties and the contribution ratios are not follow one certain paradigm. However, we found something interesting. First, we notice that from $7th$ to $13th$ connections, the contribution ratios of all attention modules are always stable compared to other connections. Somewhat interestingly, the $7th$ to $13th$ connections are connections in stage 3 of a ResNet. The key insight behind this observation is that our DCA module would largely boost the feature extraction ability of later layers. Such an observation can also be confirmed in Figure 4. in stark contrast to CBAM-ResNet50 and ResNet50, our DCA-CBAM-ResNet50 closely pay attention to the dog and almost not activation on unrelated regions in stage 3. Second, the contribution ratios in $14th$ connection of all attention modules are close to 20% and then enlarged in the last connection. Holistically speaking, the contribution ratio in all connections are always greater than 0 (although occasionally close to 0 and GC module always less than 20%), which means that the previous attention always makes contribution to the attention learning in current attention block.

5. Discussion

Intuitively, our DCA module simply deploys shortcut connections in ResNet to attention blocks. However, they are essentially different in nature. First, we connect attention blocks for jointly training, whereas shortcuts in ResNet is designed for mitigating "degradation" problem; Besides, we connect output of previous attention block to output of extraction module in current block, while ResNet directly connects previous convolutional block output to current output. Our DCA module is a simply preliminary approach for connected attention. More connection fashion can be explored, such as connecting each attention block to every other attention block in a dense fashion like DenseNet [16], and connecting attention blocks in a tree structure like DLA [41]. We hope our simple and effective approach will serve as a baseline and help ease future research in attention block boosting.

detector	backbone	$AP_{50:95}$	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet	ResNet50	36.2%	55.9%	38.5%	19.4%	39.8%	48.3%
RetinaNet	SE-ResNet50	37.4%	57.8%	39.8%	20.6%	40.8%	50.3%
RetinaNet	DCA-SE-ResNet50	**37.7%**	**58.2%**	**40.1%**	**20.8%**	**40.9%**	**50.4%**
Cascade R-CNN	ResNet50	40.6%	58.9%	44.2%	22.4%	43.7%	**54.7%**
Cascade R-CNN	GC-ResNet50	41.1%	59.7%	44.6%	**23.6%**	44.1%	54.3%
Cascade R-CNN	DCA-GC-ResNet50	**41.4%**	**60.2%**	**44.7%**	22.8%	**45.0%**	54.2%

Table 7. Detection performances with different backbone architectures. We employed two state-of-the-art detectors: RetinaNet [21] and Cascade R-CNN [4] in our detection experiments. To illustrate the generalization of our DCA module, we apply DCA module to Global Context (GC) block and Squeeze-Excitation (SE) block. We report mean Average Precision at three different IoU values and three different object scales on COCO 2017 validation set.

Figure 4. Attention maps (heatmaps) visualization using Grad-CAM [30]. Top line is generated by ResNet50, middle line is generated by CBAM-ResNet50 and bottom line is generated by DCA-CBAM-ResNet50 (along spatial dimension). From column2 to column5 are outputs of stage1, stage2, stage3 and stage4 respectively. Obviously, our DCA enhanced counterpart is much easier to focus on the object than vanilla attention model. We can see that our DCA enhanced CBAM-ResNet50 closely pays attention to the dog and the key parts (bell, ear, back and legs) are progressively enhanced. On the contrary, vanilla CBAM-ResNet50 varies attention a lot in different stages. In stage3, it even pay attention to the top left, where we don't care. This double confirms our basic idea of deep connected attention (Best viewed in color).

We observe that connecting attention blocks de facto is a departure from self-attention mechanism which learns from feature map itself. By connecting attention blocks, each attention block learns from two terms: previous attention map and current feature map. The previous attention map could be considered as a guideline for attention map learning from current feature map. Boosted by our DCA module, original attention module

6. Conclusion

Extensive researches has been conducted on self-attention mechanism to boost base CNN models, whereas efforts applied to boost attention blocks are few. In this paper, we present DCANet, which novelly boosts attention learning by connecting adjacent attention blocks. DCANet is able to boost a range of attention blocks like SENet, SKNet and CBAM with negligible parameters increase, regardless of the choice of base networks. Comprehensive experiments showcase the efficiency and performance of our approach.

One of the main contribution is that it is first time to introduce connections between attention blocks to boost attention blocks. We believe that our idea of simply connecting adjacent attention blocks can be potentially generalized to other connection schemas, like dense or tree structure fashion. Note that by adding a connection between two attention blocks, our DCA module virtually is a departure from self-attention which only considers current features. We also suggest a deep rethink on self-attention mechanism.

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 2

[2] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. *arXiv preprint arXiv:1904.09925*, 2019. 7

[3] Yuanqiang Cai, Dawei Du, Libo Zhang, Longyin Wen, Weiqiang Wang, Yanjun Wu, and Siwei Lyu. Guided attention network for object detection and counting on drones, 2019. 2

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the*

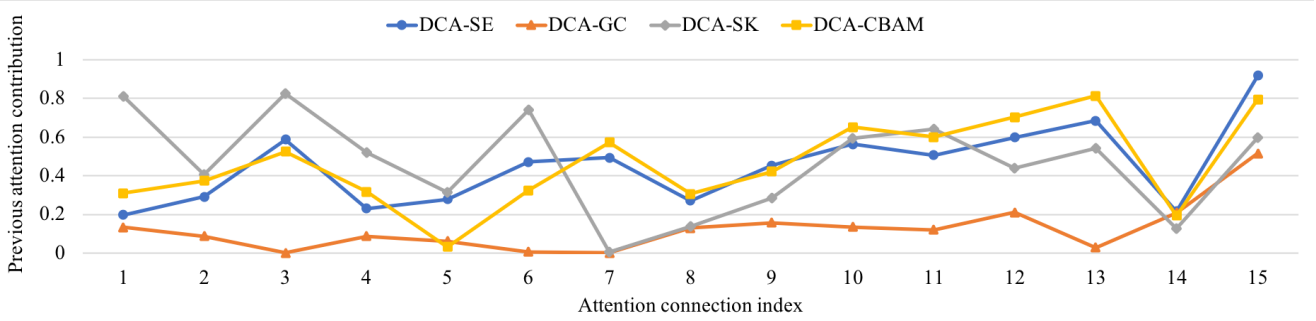

Figure 5. The contribution ratio of the previous attention in each connection.

IEEE conference on computer vision and pattern recognition, pages 6154–6162, 2018. 8, 9

[5] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *arXiv preprint arXiv:1904.11492*, 2019. 2, 3, 4, 6

[6] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 8

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 2, 5

[8] Ruan Dongsheng, Wen Jun, and Zheng Nenggan. Linear context transform block, 2019. 3

[9] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4438–4446, 2017. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 5, 6

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 2

[12] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. *arXiv preprint arXiv:1905.02244*, 2019. 2

[13] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2, 5

[14] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 9401–9411, 2018. 1, 2, 4, 6

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 1, 2, 3, 4, 6

[16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 2, 5, 8

[17] Xiang Li, Xiaolin Hu, and Jian Yang. Spatial group-wise enhance: Improving semantic feature learning in convolutional networks. *arXiv preprint arXiv:1905.09646*, 2019. 1

[18] Xiang Li, Xiaolin Hu, and Jian Yang. Spatial group-wise enhance: Improving semantic feature learning in convolutional networks, 2019. 2

[19] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–519, 2019. 1, 2, 3, 4, 6

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 8

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 8, 9

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 8

[23] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[24] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architec-

ture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. 2

[25] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In-So Kweon. Bam: Bottleneck attention module. In *British Machine Vision Conference (BMVC)*. British Machine Vision Association (BMVA), 2018. 2, 5

[26] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 2, 5

[28] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 2, 5, 6

[29] Nicol Schraudolph. Accelerated gradient descent by factor-centering decomposition. *Technical report/IDSIA*, 98, 1998. 2

[30] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017. 8, 9

[31] Shuyang Sun, Jiangmiao Pang, Jianping Shi, Shuai Yi, and Wanli Ouyang. Fishnet: A versatile backbone for image, region, and pixel level prediction. In *Advances in Neural Information Processing Systems*, pages 754–764, 2018. 2

[32] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013. 6

[33] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 2

[34] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 2

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 2

[36] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828, 2019. 6

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2

[38] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017. 1, 2

[39] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 1, 2

[40] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 1, 2, 3, 4, 5, 6

[41] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018. 2, 8

[42] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018. 2

[43] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. *arXiv preprint arXiv:1904.05873*, 2019. 1

[44] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks, 2019. 3