Coordinate Attention for Efficient Mobile Network Design

Qibin Hou[1] Daquan Zhou[1] Jiashi Feng[1]
[1]National University of Singapore
{andrewhoux,zhoudaquan21}@gmail.com

Abstract

Recent studies on mobile network design have demonstrated the remarkable effectiveness of channel attention (e.g., the Squeeze-and-Excitation attention) for lifting model performance, but they generally neglect the positional information, which is important for generating spatially selective attention maps. In this paper, we propose a novel attention mechanism for mobile networks by embedding positional information into channel attention, which we call "coordinate attention". Unlike channel attention that transforms a feature tensor to a single feature vector via 2D global pooling, the coordinate attention factorizes channel attention into two 1D feature encoding processes that aggregate features along the two spatial directions, respectively. In this way, long-range dependencies can be captured along one spatial direction and meanwhile precise positional information can be preserved along the other spatial direction. The resulting feature maps are then encoded separately into a pair of direction-aware and position-sensitive attention maps that can be complementarily applied to the input feature map to augment the representations of the objects of interest. Our coordinate attention is simple and can be flexibly plugged into classic mobile networks, such as MobileNetV2, MobileNeXt, and EfficientNet with nearly no computational overhead. Extensive experiments demonstrate that our coordinate attention is not only beneficial to ImageNet classification but more interestingly, behaves better in down-stream tasks, such as object detection and semantic segmentation. Code is available at `https://github.com/Andrew-Qibin/CoordAttention`.

1. Introduction

Attention mechanisms, used to tell a model "what" and "where" to attend, have been extensively studied [47, 29] and widely deployed for boosting the performance of modern deep neural networks [18, 44, 3, 25, 10, 14]. However, their application for mobile networks (with limited model size) significantly lags behind that for large networks [36, 13, 46]. This is mainly because the computational overhead brought by most attention mechanisms is not affordable for mobile networks.

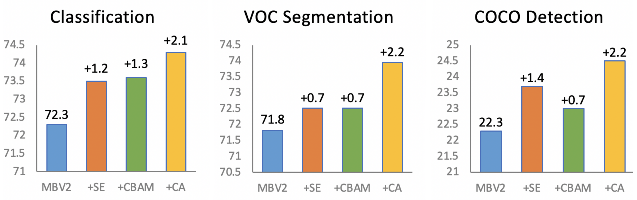

Figure 1. Performance of different attention methods on three classic vision tasks. The y-axis labels from left to right are top-1 accuracy, mean IoU, and AP, respectively. Clearly, our approach not only achieves the best result in ImageNet classification [33] against the SE block [18] and CBAM [44] but performs even better in down-stream tasks, like semantic segmentation [9] and COCO object detection [21]. Results are based on MobileNetV2 [34].

Considering the restricted computation capacity of mobile networks, to date, the most popular attention mechanism for mobile networks is still the Squeeze-and-Excitation (SE) attention [18]. It computes channel attention with the help of 2D global pooling and provides notable performance gains at considerably low computational cost. However, the SE attention only considers encoding inter-channel information but neglects the importance of positional information, which is critical to capturing object structures in vision tasks [42]. Later works, such as BAM [30] and CBAM [44], attempt to exploit positional information by reducing the channel dimension of the input tensor and then computing spatial attention using convolutions as shown in Figure 2(b). However, convolutions can only capture local relations but fail in modeling long-range dependencies that are essential for vision tasks [48, 14].

In this paper, beyond the first works, we propose a novel and efficient attention mechanism by embedding positional information into channel attention to enable mobile networks to attend over large regions while avoiding incurring significant computation overhead. To alleviate the positional information loss caused by the 2D global pooling, we factorize channel attention into two parallel 1D feature encoding processes to effectively integrate spatial coordinate information into the generated attention maps. Specifically, our method exploits two 1D global pooling operations to respectively aggregate the input features along the vertical and horizontal directions into two separate direction-aware feature maps. These two feature maps with embedded direction-specific information are then separately encoded into two attention maps, each of which captures long-range dependencies of the input feature map along one spatial direction. The positional information can thus be preserved in the generated attention maps. Both attention maps are then applied to the input feature map via multiplication to emphasize the representations of interest. We name the proposed attention method as *coordinate attention* as its operation distinguishes spatial direction (*i.e.,* coordinate) and generates coordinate-aware attention maps.

Our coordinate attention offers the following advantages. First of all, it captures not only cross-channel but also direction-aware and position-sensitive information, which helps models to more accurately locate and recognize the objects of interest. Secondly, our method is flexible and light-weight, and can be easily plugged into classic building blocks of mobile networks, such as the inverted residual block proposed in MobileNetV2 [34] and the sandglass block proposed in MobileNeXt [49], to augment the features by emphasizing informative representations. Thirdly, as a pretrained model, our coordinate attention can bring significant performance gains to down-stream tasks with mobile networks, especially for those with dense predictions (*e.g.,* semantic segmentation), which we will show in our experiment section.

To demonstrate the advantages of the proposed approach over previous attention methods for mobile networks, we conduct extensive experiments in both ImageNet classification [33] and popular down-stream tasks, including object detection and semantic segmentation. With a comparable amount of learnable parameters and computation, our network achieves 0.8% performance gain in top-1 classification accuracy on ImageNet. In object detection and semantic segmentation, we also observe significant improvements compared to models with other attention mechanisms as shown in Figure 1. We hope our simple and efficient design could facilitate the development of attention mechanisms for mobile networks in the future.

2. Related Work

In this section, we give a brief literature review of this paper, including prior works on efficient network architecture design and attention or non-local models.

2.1. Mobile Network Architectures

Recent state-of-the-art mobile networks are mostly based on the depthwise separable convolutions [16] and the inverted residual block [34]. HBONet [20] introduces down-sampling operations inside each inverted residual block for modeling the representative spatial information. ShuffleNetV2 [27] uses a channel split module and a channel shuffle module before and after the inverted residual block. Later, MobileNetV3 [15] combines with neural architecture search algorithms [50] to search for optimal activation functions and the expansion ratio of inverted residual blocks at different depths. Moreover, MixNet [39], EfficientNet [38] and ProxylessNAS [2] also adopt different searching strategies to search for either the optimal kernel sizes of the depthwise separable convolutions or scalars to control the network weight in terms of expansion ratio, input resolution, network depth and width. More recently, Zhou *et al.* [49] rethought the way of exploiting depthwise separable convolutions and proposed MobileNeXt that adopts a classic bottleneck structure for mobile networks.

2.2. Attention Mechanisms

Attention mechanisms [41, 40] have been proven helpful in a variety of computer vision tasks, such as image classification [18, 17, 44, 1] and image segmentation [14, 19, 10]. One of the successful examples is SENet [18], which simply squeezes each 2D feature map to efficiently build interdependencies among channels. CBAM [44] further advances this idea by introducing spatial information encoding via convolutions with large-size kernels. Later works, like GENet [17], GALA [22], AA [1], and TA [28], extend this idea by adopting different spatial attention mechanisms or designing advanced attention blocks.

Non-local/self-attention networks are recently very popular due to their capability of building spatial or channel-wise attention. Typical examples include NLNet [43], GCNet [3], A^2Net [7], SCNet [25], GSoP-Net [11], or CCNet [19], all of which exploit non-local mechanisms to capture different types of spatial information. However, because of the large amount of computation inside the self-attention modules, they are often adopted in large models [13, 46] but not suitable for mobile networks.

Different from these approaches that leverage expensive and heavy non-local or self-attention blocks, our approach considers a more efficient way of capturing positional information and channel-wise relationships to augment the feature representations for mobile networks. By factorizing the 2D global pooling operations into two one-dimensional encoding processes, our approach performs much better than other attention methods with the lightweight property (*e.g.,* SENet [18], CBAM [44], and TA [28]).

3. Coordinate Attention

A *coordinate attention* block can be viewed as a computational unit that aims to enhance the expressive power of the learned features for mobile networks. It can take any intermediate feature tensor $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_C] \in$

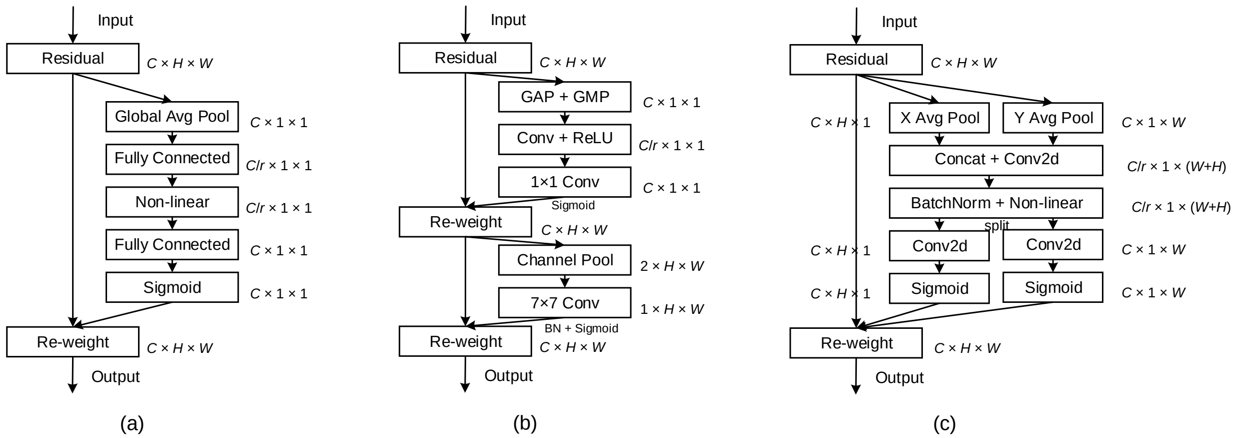

Figure 2. Schematic comparison of the proposed coordinate attention block (c) to the classic SE channel attention block [18] (a) and CBAM [44] (b). Here, "GAP" and "GMP" refer to the global average pooling and global max pooling, respectively. 'X Avg Pool' and 'Y Avg Pool' refer to 1D horizontal global pooling and 1D vertical global pooling, respectively.

$\mathbb{R}^{C\times H\times W}$ as input and outputs a transformed tensor with augmented representations $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_C]$ of the same size to $\mathbf{X}$. To provide a clear description of the proposed coordinate attention, we first revisit the SE attention, which is widely used in mobile networks.

3.1. Revisit Squeeze-and-Excitation Attention

As demonstrated in [18], the standard convolution itself is difficult to model the channel relationships. Explicitly building channel inter-dependencies can increase the model sensitivity to the informative channels that contribute more to the final classification decision. Moreover, using global average pooling can also assist the model in capturing global information, which is a lack for convolutions.

Structurally, the SE block can be decomposed into two steps: squeeze and excitation, which are designed for global information embedding and adaptive recalibration of channel relationships, respectively. Given the input $\mathbf{X}$, the squeeze step for the c-th channel can be formulated as follows:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i, j), \tag{1}$$

where z_c is the output associated with the c-th channel. The input $\mathbf{X}$ is directly from a convolutional layer with a fixed kernel size and hence can be viewed as a collection of local descriptors. The squeeze operation makes collecting global information possible.

The second step, excitation, aims to fully capture channel-wise dependencies, which can be formulated as

$$\hat{\mathbf{X}} = \mathbf{X} \cdot \sigma(\hat{\mathbf{z}}), \tag{2}$$

where $\cdot$ refers to channel-wise multiplication, σ is the sigmoid function, and $\hat{\mathbf{z}}$ is the result generated by a transformation function, which is formulated as follows:

$$\hat{\mathbf{z}} = T_2(\text{ReLU}(T_1(\mathbf{z}))). \tag{3}$$

Here, T_1 and T_2 are two linear transformations that can be learned to capture the importance of each channel.

The SE block has been widely used in recent mobile networks [18, 4, 38] and proven to be a key component for achieving state-of-the-art performance. However, it only considers reweighing the importance of each channel by modeling channel relationships but neglects positional information, which as we will prove experimentally in Section 4 to be important for generating spatially selective attention maps. In the following, we introduce a novel attention block, which takes into account both inter-channel relationships and positional information.

3.2. Coordinate Attention Blocks

Our coordinate attention encodes both channel relationships and long-range dependencies with precise positional information in two steps: coordinate information embedding and coordinate attention generation. The diagram of the proposed coordinate attention block can be found in the right part of Figure 2. In the following, we will describe it in detail.

3.2.1 Coordinate Information Embedding

The global pooling is often used in channel attention to encode spatial information globally, but it squeezes global spatial information into a channel descriptor and hence is difficult to preserve positional information, which is essential for capturing spatial structures in vision tasks. To encourage attention blocks to capture long-range interactions spatially with precise positional information, we factorize the global pooling as formulated in Eqn. (1) into a pair of

1D feature encoding operations. Specifically, given the input $\mathbf{X}$, we use two spatial extents of pooling kernels $(H, 1)$ or $(1, W)$ to encode each channel along the horizontal coordinate and the vertical coordinate, respectively. Thus, the output of the c-th channel at height h can be formulated as

$$z_c^h(h) = \frac{1}{W} \sum_{0 \le i < W} x_c(h, i). \tag{4}$$

Similarly, the output of the c-th channel at width w can be written as

$$z_c^w(w) = \frac{1}{H} \sum_{0 \le j < H} x_c(j, w). \tag{5}$$

The above two transformations aggregate features along the two spatial directions respectively, yielding a pair of direction-aware feature maps. This is rather different from the squeeze operation (Eqn. (1)) in channel attention methods that produce a single feature vector. These two transformations also allow our attention block to capture long-range dependencies along one spatial direction and preserve precise positional information along the other spatial direction, which helps the networks more accurately locate the objects of interest.

3.2.2 Coordinate Attention Generation

As described above, Eqn. (4) and Eqn. (5) enable a global receptive field and encode precise positional information. To take advantage of the resulting expressive representations, we present the second transformation, termed coordinate attention generation. Our design refers to the following three criteria. First of all, the new transformation should be as simple and cheap as possible regarding the applications in mobile environments. Second, it can make full use of the captured positional information so that the regions of interest can be accurately highlighted. Last but not the least, it should also be able to effectively capture inter-channel relationships, which has been demonstrated essential in existing studies [18, 44].

Specifically, given the aggregated feature maps produced by Eqn. 4 and Eqn. 5, we first concatenate them and then send them to a shared 1×1 convolutional transformation function F_1, yielding

$$\mathbf{f} = \delta(F_1([\mathbf{z}^h, \mathbf{z}^w])), \tag{6}$$

where $[\cdot, \cdot]$ denotes the concatenation operation along the spatial dimension, δ is a non-linear activation function and $\mathbf{f} \in \mathbb{R}^{C/r \times (H+W)}$ is the intermediate feature map that encodes spatial information in both the horizontal direction and the vertical direction. Here, r is the reduction ratio for controlling the block size as in the SE block. We then split $\mathbf{f}$ along the spatial dimension into two separate tensors $\mathbf{f}^h \in \mathbb{R}^{C/r \times H}$ and $\mathbf{f}^w \in \mathbb{R}^{C/r \times W}$. Another two 1×1 convolutional transformations F_h and F_w are utilized to separately transform $\mathbf{f}^h$ and $\mathbf{f}^w$ to tensors with the same channel number to the input $\mathbf{X}$, yielding

$$\mathbf{g}^h = \sigma(F_h(\mathbf{f}^h)), \tag{7}$$

$$\mathbf{g}^w = \sigma(F_w(\mathbf{f}^w)). \tag{8}$$

Recall that σ is the sigmoid function. To reduce the overhead model complexity, we often reduce the channel number of $\mathbf{f}$ with an appropriate reduction ratio r (*e.g.,* 32). We will discuss the impact of different reduction ratios on the performance in our experiment section. The outputs $\mathbf{g}^h$ and $\mathbf{g}^w$ are then expanded and used as attention weights, respectively. Finally, the output of our coordinate attention block $\mathbf{Y}$ can be written as

$$y_c(i, j) = x_c(i, j) \times g_c^h(i) \times g_c^w(j). \tag{9}$$

Discussion. Unlike channel attention that only focuses on reweighing the importance of different channels, our coordinate attention block also considers encoding the spatial information. As described above, the attention along both the horizontal and vertical directions is simultaneously applied to the input tensor. Each element in the two attention maps reflects whether the object of interest exists in the corresponding row and column. This encoding process allows our coordinate attention to more accurately locate the exact position of the object of interest and hence helps the whole model to recognize better. We will demonstrate this exhaustively in our experiment section.

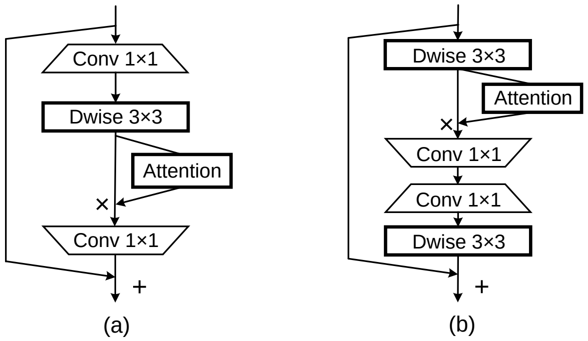

Figure 3. Network implementation for different network architectures. (a) Inverted residual block proposed in MobileNetV2 [34]; (b) Sandglass bottleneck block proposed in MobileNeXt [49].

3.3. Implementation

As the goal of this paper is to investigate a better way to augment the convolutional features for mobile networks, here we take two classic light-weight architectures with different types of residual blocks (*i.e.,* MobileNetV2 [34] and MobileNeXt [49]) as examples to demonstrate the advantages of the proposed coordinate attention block over other

Table 1. Result comparisons under different experiment settings of the proposed coordinate attention. Here, r is the reduction ratio and the baseline result is based on the MobileNetV2 model. As can be seen, the model with either the horizontal (X) attention or the vertical (Y) attention added achieves the same performance as the one with SE attention. However, when taking both horizontal and vertical attentions into account (coordinate attention), our approach yields the best result. The latency is tested on a Google Pixel 4 device.

Settings	Param.	M-Adds	r	Latency	Top-1 (%)
Baseline	3.5M	300M	-	14-16ms	72.3
+ SE	3.89M	300M	24	16-18ms	$73.5_{+1.2}$
+ X Attention	3.89M	300M	24	16-18ms	$73.5_{+1.2}$
+ Y Attention	3.89M	300M	24	16-18ms	$73.5_{+1.2}$
+ Coord. Attention	3.95M	310M	32	17-19ms	$\mathbf{74.3}_{+2.0}$

famous light-weight attention blocks. Figure 3 shows how we plug attention blocks into the inverted residual block in MobileNetV2 and the sandglass block in MobileNeXt.

4. Experiments

In this section, we first describe our experiment settings and then conduct a series of ablation experiments to demonstrate the contribution of each component in the proposed coordinate attention to the performance. Next, we compare our approach with some attention based methods. Finally, we report the results of the proposed approach compared to other attention based methods on object detection and semantic segmentation.

4.1. Experiment Setup

We use the PyTorch toolbox [31] to implement all our experiments. During training, we use the standard SGD optimizer with decay and momentum of 0.9 to train all the models. The weight decay is set to 4×10^{-5} always. The cosine learning schedule with an initial learning rate of 0.05 is adopted. We use four NVIDIA GPUs for training and the batch size is set to 256. Without extra declaration, we take MobileNetV2 as our baseline and train all the models for 200 epochs. For data augmentation, we use the same methods as in MobileNetV2. We report results on the ImageNet dataset [33] in classification.

4.2. Ablation Studies

Importance of coordinate attention. To demonstrate the performance of the proposed coordinate attention, we perform a series of ablation experiments, the corresponding results of which are all listed in Table 1. We remove either the horizontal attention or the vertical attention from the coordinate attention to see the importance of encoding coordinate information. As shown in Table 1, the model with attention along either direction has comparable performance to the one with the SE attention. However, when both the horizontal attention and the vertical attention are incorporated, we obtain the best result as highlighted in Table 1. These experiments reflect that with comparable learnable parameters and computational cost, coordinate information embedding is more helpful for image classification.

Table 2. Comparisons of different attention methods under different weight multipliers when taking MobileNetV2 as the baseline.

Settings	Param. (M)	M-Adds (M)	Top-1 Acc (%)
MobileNetV2-1.0	3.5	300	72.3
+ SE	3.89	300	$73.5_{+1.2}$
+ CBAM	3.89	300	$73.6_{+1.3}$
+ CA	3.95	310	$\mathbf{74.3}_{+2.0}$
MobileNetV2-0.75	2.5	200	69.9
+ SE	2.86	210	$71.5_{+1.6}$
+ CBAM	2.86	210	$71.5_{+1.6}$
+ CA	2.89	210	$\mathbf{72.1}_{+2.2}$
MobileNetV2-0.5	2.0	100	65.4
+ SE	2.1	100	$66.4_{+1.0}$
+ CBAM	2.1	100	$66.4_{+1.0}$
+ CA	2.1	100	$\mathbf{67.0}_{+1.6}$

Different weight multipliers. Here, we take two classic mobile networks (including MobileNetV2 [34] with inverted residual blocks and MobileNeXt [49] with sandglass bottleneck block) as baselines to see the performance of the proposed approach compared to the SE attention [18] and CBAM [44] under different weight multipliers. In this experiment, we adopt three typical weight multipliers, including $\{1.0, 0.75, 0.5\}$. As shown in Table 2, when taking the MobileNetV2 network as baseline, models with CBAM have similar results to those with the SE attention. However, models with the proposed coordinate attention yield the best results under each setting. Similar phenomenon can also be observed when the MobileNeXt network is used as listed in Table 3. This indicates that no matter which of the sandglass bottleneck block or the inverted residual block is considered and no matter which weight multiplier is selected, our coordinate attention performs the best because of the advanced way to encode positional and inter-channel information simultaneously.

The impact of reduction ratio r. To investigate the impact of different reduction ratios of attention blocks on the model performance, we attempt to decrease the size of the reduction ratio and see the performance change. As shown in Table 4, when we reduce r to half of the original size, the model size increases but better performance can be yielded. This demonstrates that adding more parameters by reducing the reduction ratio matters for improving the model performance. More importantly, our coordinate attention still per-

Table 3. Comparisons of different attention methods under different weight multipliers when taking MobileNeXt [49] as the baseline.

Settings	Param. (M)	M-Adds (M)	Top-1 Acc (%)
MobileNeXt	3.5	300	74.0
+ SE	3.89	300	$74.7_{+0.7}$
+ CA	4.09	330	$\mathbf{75.2}_{+1.2}$
MobileNeXt-0.75	2.5	210	72.0
+ SE	2.9	210	$72.6_{+0.6}$
+ CA	3.0	220	$\mathbf{73.2}_{+1.2}$
MobileNeXt-0.5	2.1	110	67.7
+ SE	2.4	110	$68.7_{+1.0}$
+ CA	2.4	110	$\mathbf{69.4}_{+1.7}$

Table 4. Comparisons of models equipped with different attention blocks under different reduction ratios r. The baseline result is based on the MobileNetV2 model. Obviously, when the reduction ratio decreases, our approach still yields the best results.

Settings	Param.	M-Adds	r	Top-1 Acc (%)
Baseline	3.5M	300M	-	72.3
+ SE	3.89M	300M	24	$73.5_{+1.2}$
+ CBAM	3.89M	300M	24	$73.6_{+1.3}$
+ CA (Ours)	3.95M	310M	32	$\mathbf{74.3}_{+2.0}$
+ SE	4.28M	300M	12	$74.1_{+1.8}$
+ CBAM	4.28M	300M	12	$74.1_{+1.8}$
+ CA (Ours)	4.37M	310M	16	$\mathbf{74.7}_{+2.4}$

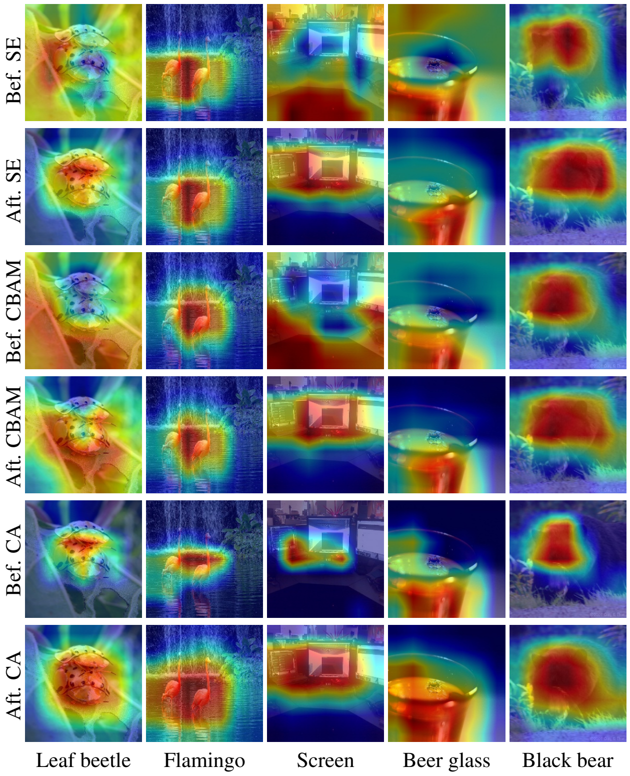

Figure 4. Visualization of feature maps produced by models with different attention methods in the last building block. We use Grad-CAM [35] as our visualization tool. Both feature maps before and after each attention block are visualized. It is obvious that our coordinate attention (CA) can more precisely locate the objects of interest than other attention methods.

forms better than the SE attention and CBAM in this experiment, reflecting the robustness of the proposed coordinate attention to the reduction ratio.

4.3. Comparison with Other Methods

Attention for Mobile Networks. We compare our coordinate attention with other light-weight attention methods for mobile networks, including the widely adopted SE attention [18] and CBAM [44] in Table 2. As can be seen, adding the SE attention has already raised the classification performance by more than 1%. For CBAM, it seems that its spatial attention module shown in Figure 2(b) does not contribute in mobile networks compared to the SE attention. However, when the proposed coordinate attention is considered, we achieve the best results. We also visualize the feature maps produced by models with different attention methods in Figure 4. Obviously, our coordinate attention can help better in locating the objects of interest than the SE attention and CBAM.

We argue that the advantages of the proposed positional information encoding manner over CBAM are two-fold. First, the spatial attention module in CBAM squeezes the channel dimension to 1, leading to information loss. However, our coordinate attention uses an appropriate reduction ratio to reduce the channel dimension in the bottleneck, avoiding too much information loss. Second, CBAM utilizes a convolutional layer with kernel size 7 × 7 to encode local spatial information while our coordinate attention encodes global information by using two complementary 1D global pooling operations. This enables our coordinate attention to capture long-range dependencies among spatial locations that are essential for vision tasks.

Stronger Baseline. To further demonstrate the advantages of the proposed coordinate attention over the SE attention in more powerful mobile networks, we take EfficientNet-b0 [38] as our baseline here. EfficientNet is based on architecture search algorithms. and contains SE attention. To investigate the performance of the proposed coordinate attention on EfficientNet, we simply replace the SE attention with our proposed coordinate attention. For other settings, we follow the original paper. The results have been listed in Table 5. Compared to the original EfficientNet-b0 with SE attention included and other methods that have comparable

Table 5. Experimental results when taking the powerful EfficientNet-b0 [38] as baseline. We also compare with other methods that have comparable parameters and computations to EfficientNet-b0.

Settings	Param.	M-Adds	Top-1 Acc (%)
PNAS [23]	5.1M	588M	72.7
DARTS [24]	4.7M	574M	73.3
ProxylessNAS-M [2]	4.1M	330M	74.4
AmoebaNet-A [32]	5.1M	555M	74.5
FBNet-C [45]	5.5M	375M	74.9
MobileNeXt [49]	6.1M	590M	76.1
MNasNet-A3 [37]	5.2M	403M	76.7
EfficientNet-b0 (w/ SE) [38]	5.3M	390M	76.3
EfficientNet-b0 (w/ CA)	5.4M	400M	**76.9**

parameters and computations to EfficientNet-b0, our network with coordinate attention achieves the best result. This demonstrates that the proposed coordinate attention can still performance well in powerful mobile networks.

4.4. Applications

In this subsection, we conduct experiments on both the object detection task and the semantic segmentation task to explore the transferable capability of the proposed coordinate attention against other attention methods.

4.4.1 Object Detection

Implementation Details. Our code is based on PyTorch and SSDLite [34, 26]. Following [34], we connect the first and second layers of SSDLite to the last pointwise convolutions with output stride of 16 and 32, respectively and add the rest SSDLite layers on top of the last convolutional layer. When training on COCO, we set the batch size to 256 and use the synchronized batch normalization. The cosine learning schedule is used with an initial learning rate of 0.01. We train the models for totally 1,600,000 iterations. When training on Pascal VOC, the batch size is set to 24 and all the models are trained for 240,000 iterations. The weight decay is set to 0.9. The initial learning rate is 0.001, which is then divided by 10 at 160,000 and again at 200,000 iterations. For other settings, readers can refer to [34, 26].

Results on COCO. In this experiment, we follow most previous work and report results in terms of AP, AP_{50}, AP_{75}, AP_S, AP_M, and AP_L, respectively. In Table 6, we show the results produced by different network settings on the COCO 2017 validation set. It is obvious that adding coordinate attention into MobileNetV2 substantially improve the detection results (24.5 v.s. 22.3) with only 0.5M parameters overhead and nearly the same computational cost. Compared to other light-weight attention methods, such as the SE attention and CBAM, our version of SSDLite320 achieves the best results in all metrics with nearly the same number of parameters and computations.

Moreover, we also show results produced by previous state-of-the-art models based on SSDLite320 as listed in Table 6. Note that some methods (*e.g.,* MobileNetV3 [15] and MnasNet-A1 [37]) are based on neural architecture search methods but our model does not. Obviously, our detection model achieves the best results in terms of AP compared to other approaches with close parameters and computations.

Results on Pascal VOC. In Table 7, we show the detection results on Pascal VOC 2007 test set when different attention methods are adopted. We observe that the SE attention and CBAM cannot improve the baseline results. However, adding the proposed coordinate attention can largely raise the mean AP from 71.7 to 73.1. Both detection experiments on COCO and Pascal VOC datasets demonstrate that classification models with the proposed coordinate attention have better transferable capability compared to those with other attention methods.

4.4.2 Semantic Segmentation

We also conduct experiments on semantic segmentation. Following MobileNetV2 [34], we utilize the classic DeepLabV3 [6] as an example and compare the proposed approach with other models to demonstrate the transferable capability of the proposed coordinate attention in semantic segmentation. Specifically, we discard the last linear operator and connect the ASPP to the last convolutional operator. We replace the standard 3×3 convolutional operators with the depthwise separable convolutions in the ASPP to reduce the model size considering mobile applications. The output channels for each branch in ASPP are set to 256 and other components in the ASPP are kept unchanged (including the 1×1 convolution branch and the image-level feature encoding branch). We report results on two widely used semantic segmentation benchmarks, including Pascal VOC 2012 [9] and Cityscapes [8]. For experiment settings, we strictly follow the DeeplabV3 paper except for the weight decay that is set to 4e-5. When the output stride is set to 16, the dilation rates in the ASPP are $\{6, 12, 18\}$ while $\{12, 24, 36\}$ when the output stride is set to 8.

Results on Pascal VOC 2012. The Pascal VOC 2012 segmentation benchmark has totally 21 classes including one background class. As suggested by the original paper, we use the split with 1,464 images for training and the split with 1,449 images for validation. Also, as done in most previous work [6, 5], we augment the training set by adding extra images from [12], resulting in totally 10,582 images for training.

We show the segmentation results when taking different models as backbones in Table 8. We report results un-

Table 6. Object detection results on the COCO validation set. In all experiments here, we use the SSDLite320 detector. As can be seen, the backbone model with our coordinate attention achieves the best results in terms of all kinds of measuring metrics. Note that all the results are based on single-model test. Besides hand-designed mobile networks, we also show results produced by architecture search-based methods (*i.e.,* MobileNetV3 [15] and MnasNet-A1 [37]).

No.	Method	Backbone	Param. (M)	M-Adds (B)	AP	AP$_{50}$	AP$_{75}$	AP$_{S}$	AP$_{M}$	AP$_{L}$
1	SSDLite320	MobileNetV1 [16]	5.1	1.3	22.2	-	-	-	-	-
2	SSDLite320	MobileNetV2 [34]	4.3	0.8	22.3	37.4	22.7	2.8	21.2	42.8
3	SSDLite320	MobileNetV3 [15]	5.0	0.62	22.0	-	-	-	-	-
4	SSDLite320	MnasNet-A1 [37]	4.9	0.8	23.0	-	-	3.8	21.7	42.0
5	SSDLite320	MobileNeXt [49]	4.4	0.8	23.3	38.9	23.7	2.8	22.7	45.0
6	SSDLite320	MobileNetV2 + SE	4.7	0.8	23.7	40.0	24.3	2.2	25.4	44.7
7	SSDLite320	MobileNetV2 + CBAM	4.7	0.8	23.0	38.6	23.3	2.7	22.2	44.5
8	SSDLite320	MobileNetV2 + CA	4.8	0.8	**24.5**	40.7	25.4	2.3	26.2	45.9

Table 7. Object detection results on the Pascal VOC 2007 test set. We can observe that when the same SSDLite320 detector is adopted, MobileNetV2 network with our coordinate attention added achieves better results in terms of mAP.

Backbone	Param. (M)	M-Adds (B)	mAP (%)
MobileNetV2 [34]	4.3	0.8	71.7
MobileNetV2 + SE	4.7	0.8	71.7
MobileNetV2 + CBAM	4.7	0.8	71.7
MobileNetV2 + CA	4.8	0.8	**73.1**

Table 8. Semantic segmentation results on the Pascal VOC 2012 validation set. All the results are based on single-model test and no post-processing tools are used. We can see that the models equipped with all attention methods improve the segmentation results. However, when the proposed coordinate attention is used, we achieve the best result, which is much better than models with other attention methods. 'Stride' here denotes the output stride of the segmentation network.

Backbone	Param. (M)	Stride	mIoU (%)
MobileNetV2 [34]	4.5	16	70.84
MobileNetV2 + SE	4.9	16	71.69
MobileNetV2 + CBAM	4.9	16	71.28
MobileNetV2 + CA (ours)	5.0	16	**73.32**
MobileNetV2 [34]	4.5	8	71.82
MobileNetV2 + SE	4.9	8	72.52
MobileNetV2 + CBAM	4.9	8	71.67
MobileNetV2 + CA (ours)	5.0	8	**73.96**

der two different output strides, *i.e.,* 16 and 8. Note that all the results reported here are not based on COCO pretraining. According to Table 8, models equipped with our coordinate attention performs much better than the vanilla MobileNetV2 and other attention methods.

Results on Cityscapes. Cityscapes [8] is one of the most popular urban street scene segmentation datasets, containing totally 19 different categories. Following the official suggestion, we use the split with 2,975 images for training and 500 images for validation. Only the fine-annotated images are used for training. In training, we randomly crop the original images to 768 × 768. During testing, all images are kept the original size (1024 × 2048).

Table 9. Semantic segmentation results on the Cityscapes [8] validation set. We report results on single-model test and full image size (*i.e.,* 1024 × 2048) is used for testing. We do not use any post-processing tools.

Backbone	Param. (M)	Output Stride	mIoU (%)
MobileNetV2	4.5	8	71.4
MobileNetV2 + SE	4.9	8	72.2
MobileNetV2 + CBAM	4.9	8	71.4
MobileNetV2 + CA	5.0	8	**74.0**

In Table 9, we show the segmentation results produced by models with different attention methods on the Cityscapes dataset. Compared to the vanilla MobileNetV2 and other attention methods, our coordinate attention can improve the segmentation results by a large margin with comparable number of learnable parameters.

Discussion. We observe that our coordinate attention yields larger improvement on semantic segmentation than ImageNet classification and object detection. We argue that this is because our coordinate attention is able to capture long-range dependencies with precise postional information, which is more beneficial to vision tasks with dense predictions, such as semantic segmentation.

5. Conclusions

In this paper, we present a novel light-weight attention mechanism for mobile networks, named coordinate attention. Our coordinate attention inherits the advantage of channel attention methods (*e.g.,* the Squeeze-and-Excitation attention) that model inter-channel relationships and meanwhile captures long-range dependencies with precise positional information. Experiments in ImageNet classification, object detection and semantic segmentation demonstrate the effectiveness of our coordination attention.

Acknowledgement. This research was partially supported by AISG-100E-2019-035, MOE2017-T2-2-151, NUS_ECRA_FY17_P08 and CRP20-2017-0006.

References

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

[2] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. *arXiv preprint arXiv:1812.00332*, 2018.

[3] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[4] Liang-Chieh Chen, Maxwell Collins, Yukun Zhu, George Papandreou, Barret Zoph, Florian Schroff, Hartwig Adam, and Jon Shlens. Searching for efficient multi-scale architectures for dense image prediction. In *NeurIPS*, pages 8699–8710, 2018.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[7] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. A^ 2-nets: Double attention networks. In *Advances in neural information processing systems*, pages 352–361, 2018.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[9] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015.

[10] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019.

[11] Zilin Gao, Jiangtao Xie, Qilong Wang, and Peihua Li. Global second-order pooling convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2019.

[12] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[14] Qibin Hou, Li Zhang, Ming-Ming Cheng, and Jiashi Feng. Strip pooling: Rethinking spatial pooling for scene parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4003–4012, 2020.

[15] 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. In *ICCV*, pages 1314–1324, 2019.

[16] 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.

[17] 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.

[18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018.

[19] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. *arXiv preprint arXiv:1811.11721*, 2018.

[20] Duo Li, Aojun Zhou, and Anbang Yao. Hbonet: Harmonious bottleneck on two orthogonal dimensions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3316–3325, 2019.

[21] 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 *ECCV*, 2014.

[22] Drew Linsley, Dan Shiebler, Sven Eberhardt, and Thomas Serre. Learning what and where to attend. In *ICLR*, 2019.

[23] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018.

[24] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[25] Jiang-Jiang Liu, Qibin Hou, Ming-Ming Cheng, Changhu Wang, and Jiashi Feng. Improving convolutional networks with self-calibrated convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10096–10105, 2020.

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[27] 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.

[28] Diganta Misra, Trikay Nalamada, Ajay Uppili Arasanipalai, and Qibin Hou. Rotate to attend: Convolutional triplet attention module. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3139–3148, 2021.

[29] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[30] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*, 2018.

[31] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.

[32] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789, 2019.

[33] 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. *IJCV*, 115(3):211–252, 2015.

[34] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.

[35] 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 *ICCV*, pages 618–626, 2017.

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[37] 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.

[38] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019.

[39] Mingxing Tan and Quoc V Le. Mixconv: Mixed depthwise convolutional kernels. *CoRR, abs/1907.09595*, 2019.

[40] John K Tsotsos. *A computational perspective on visual attention*. MIT Press, 2011.

[41] John K Tsotsos et al. Analyzing vision at the complexity level. *Behavioral and brain sciences*, 13(3):423–469, 1990.

[42] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. *arXiv preprint arXiv:2003.07853*, 2020.

[43] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[44] 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.

[45] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019.

[46] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[47] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015.

[48] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[49] Daquan Zhou, Qibin Hou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Rethinking bottleneck structure for efficient mobile network design. In *ECCV*, 2020.

[50] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *CVPR*, pages 8697–8710, 2018.