PKCAM: Previous Knowledge Channel Attention Module

Eslam Mohamed BAKR
Deep Learning Researcher
Valeo R&D Cairo, EGYPT
Cairo University, Egypt
eslam.mohamed-abdelrahman@valeo.com

Ahmad El Sallab
AI Senior Expert
Valeo R&D Cairo, EGYPT
ahmad.el-sallab@valeo.com

Mohsen A. Rashwan
Professor
Electronics and Communications Department, Cairo University, Egypt
mrashwan@rdi-eg.com,

Abstract

Recently, attention mechanisms have been explored with ConvNets, both across the spatial and channel dimensions. However, from our knowledge, all the existing methods devote the attention modules to capture local interactions from a uni-scale. In this paper, we propose a Previous Knowledge Channel Attention Module(PKCAM), that captures channel-wise relations across different layers to model the global context. Our proposed module PKCAM is easily integrated into any feed-forward CNN architectures and trained in an end-to-end fashion with a negligible footprint due to its lightweight property. We validate our novel architecture through extensive experiments on image classification and object detection tasks with different backbones. Our experiments show consistent improvements in performances against their counterparts.

1 Introduction

Over the years, CNN architectures have evolved with many ideas to better deal with spatial image features. Moreover, their localized nature makes such features lack the global view of the image. Deeper architectures emerged that stack multiple convolution layers, known with different names; backbone, bottleneck, feature extractor, or encoder. The main feature of such architectures is their ability to cover spatial features at multiple scales. As we go deeper, the feature maps get smaller, while their content represents a wider region in the space, which gets us closer to better semantics of the image contents (22). With the emergence of AlexNet (18), many kinds of research investigate to further improve the performance of deep CNNs. (30) (13) (32) (34) (31) have sought to strengthen the CNNs by making it deeper and deeper as they have shown that increasing the depth of a network could significantly increase the quality of the learned representations. Many researchers are continuously investigating to further improve the performance of deep CNNs by incorporating attention mechanisms to exploit its ability to cover the relationship between the learned spatial features.

Attention modules, in general, are designed to suppress noise while keeping useful information by refining the learned features using attention scaling. By quoting from the human perception process (24) where the high-level information is used in guiding the bottom-up learning process by capturing more sophisticated features while disregarding irrelevant details. Human perception

Machine Learning for Autonomous Driving Workshop at the 35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

and visual attention (1) (7) (24) (8) is enhanced by top-down stimuli and non-relevant neurons will be suppressed in feedback loops. Referencing to human visual system, various different attention mechanisms (2) (38) (40) (35) (36) have been explored and integrated into deep CNNs. Attention mechanisms were introduced in the context of CNNs to capture the relations between features, either across the spatial dimension as in (3) (14) ,or across channel-wise dimension as in (38) (2) (15) (20) (37) or across both dimensions as in (26) (39) (9) (36) (21) (29) (4). Although these attention methods have achieved higher accuracy than their counterpart baselines which do not invoke any attention mechanisms in their architectures, they often bring higher model complexity and exploit only the current feature map while refining it, that's why we call it local attention mechanisms.

Exploiting previous knowledge has been applied to image classification (16) (17), image segmentation (28), tracking (23), and human pose estimation (25) where they obtain enhanced performance. DenseNets (16) encourage feature reuse by connecting each layer to every other layer in a feed-forward fashion. U-Net (28) consists of two paths, which are contracting path to capture context and a symmetric expanding path that enables precise localization, where feature reuse is introduced through using skip connection between two paths. Driven by the significance of employing feature reuse while learning different tasks (33) (16) (17) (5) (23) (25), a question arises: How can one incorporate previous knowledge aggregation while learning channel attention more efficiently?

To answer this question, we introduce PKCAM, a novel feature recalibration module based on channel attention, which improves the quality of the representations produced by a network using the global information to selectively emphasize informative features and suppress less useful ones. In contrast to the aforementioned attention mechanisms, our global context aware attention block obtains additional inputs from all preceding attention blocks, that have the same depth, and passes on its refined feature-maps to all subsequent blocks, creating global awareness from exploiting previous knowledge aggregation from earlier layers that can capture fine-grained information which is useful for precise localization while attending to features from earlier layers that can encode abstract semantic information, which is robust to target appearance changes.

The contributions of this paper are summarized as follows:

- We propose a simple and effective attention module, PKCAM, which can be integrated easily with any CNNs and applied across all it's blocks due to the lightweight computation of our novel architecture.
- We verify the effectiveness and robustness of PKCAM throughout extensive experiments with various baseline architectures on KITTI dataset.
- Through detailed analysis along with ablation studies, we examine the internal behavior and validity of our method.

2 Related work

Table 1: Comparison of channel attention module by projecting each channel mechanism according to the abstract global context modeling skeleton that was introduced by GCNet [1] in Figure 4(a).

Methods	Context Modeling	Transform	Fusion
SE (15)	$Y_c = GAP(X)$(1)	$Z = \sigma(F.C_c(Relu(F.C_{\frac{c}{r}}(Y_c))))$(2)	$F = Z \odot X$(3)
ECA (37)	$Y_c = GAP(X)$(4)	$Z = \sigma(C1D_k(Y_c))$	$F = Z \odot X$ (5)
SRM (20)	$Y_{c*d} = \text{concat}[GAP(X), SP(X)]$(6)	$Z = \sigma(\sum_d (Y_{c*d} \odot W_{c*d}))$ (7)	$F = Z \odot X$ (8)
GC (2)	$Y_c = X_{C*HW} \otimes S.M.(C1D_k(X))$(9)	$Z = F.C_c(Relu(F.C_{\frac{c}{r}}(Y_c)))$(10)	$F = Z + X$ (11)

The basic channel attention module (15) (2) (38) (20) (37) (10) aims at strengthening the output features of one convolution block, $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$, where W, H and C are width, height and channel dimension. Table 1 projects each channel mechanism according to the abstract global context modeling skeleton that was introduced by GCNet (2) in Figure 4(a). The abstract global context modeling framework (2) consists of three main blocks which are context modeling, transform and fusion blocks.

SE-Net (15) and ECA-Net (37) squeeze the spatial dimension of the output features of one convolution block $\mathbf{X}$ via aggregating it through channel-wise global average pooling (GAP), where

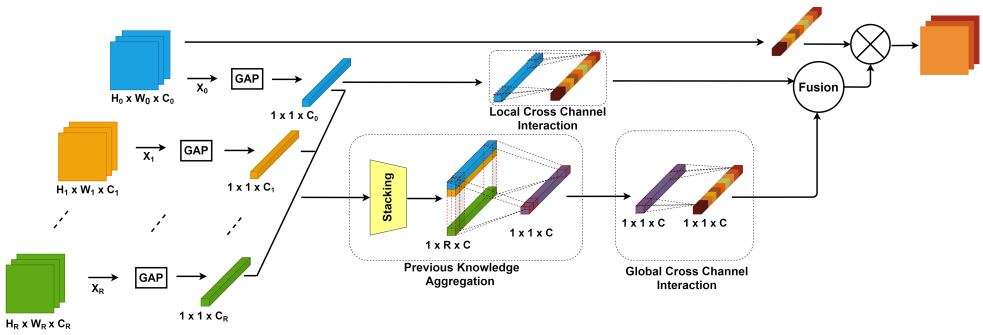

Figure 1: Diagram of our Previous Knowledge Channel Attention Module (PKCAM). Given a R aggregated features, PKCAM generates global aware channel weights by performing a fast 1D convolution of size R, accompanied by another 1D convolution, which represents global cross channel interaction, then fused with the standard local cross channel interaction.

$Y_c = \frac{1}{WH}\sum_{i=1,j=1}^{W,H} X_{ij}$, then SE-Net use non-linear transformation using two fully connected layers with a dimensionality reduction propriety to control the model complexity expansion, the transformation block is called excitation. ECA-Net (37) analyzing effects of dimensionality reduction at the SE-Net transformation block that shows that while dimensionality reduction can reduce model complexity, it destroys the direct correspondence between the channel and its weight. Therefore ECA-Net employs a more efficient transformation function as shown in Table 1, where a one-dimensional convolution layer (C1D) is used which only involves k parameters that guarantees both efficiency and effectiveness. SE-Net (15) and ECA-Net (37) opt to employ a simple gating mechanism with a sigmoid activation function σ that produces a channel weights Z. The third block of the abstract global context modeling framework (2) is fusing the channel weights Z with the original feature map X to produce a recalibrated feature map F. Both SE-Net (15) and ECA-Net (37) are using element-wise multiplication operation to recalibrate the original feature map X according to the learned channel weights Z.

SRM (20) adaptively squeezes the spatial dimension for the input feature map X based on the style of an image via a channel-independent style pooling operator by adopting the channel-wise statistics—average and standard deviation—of each feature map as style features (i.e. d = 2). Accordingly SRM context modeling block producing style features $\mathbf{Y} \in \mathbb{R}^{C\times d}$ The style features Y_{c*d} are converted into channel-wise style weights Z by the transformation block that consists of a style integration operator as shown in Table 1. The style weights Z are supposed to model the importance of the styles associated with individual channels to emphasize or suppress them accordingly. In line with SE-Net (15) and ECA-Net (37), SRM (20) adopts the same fusion mechanism by simply employ element-wise multiplication operation to recalibrate the original feature map X according to the learned channel-style weights Z

GCNet (2) adopt the same transformation block from SE-Net (15) to produce channel weights Z while proposing a new context and fusion blocks. Given an input feature map X GCNet (2) squeeze the channel dimension then a query-independent attention map is explicitly used for all query positions to learn the spatial relation between pixels, producing Y.

3 Methodology

In this section, we first demonstrate an abstracted overview of our PKCAM. Then, we demonstrate the motivations to adopt the feature reuse concept via exploiting the previous knowledge to create a global aware attention block (i.e., PKCAM). In addition, we develop a method to efficiently fuse both local and global cross-channel interaction modules, and finally show how to integrate it for an arbitrary CNN architecture.

3.1 Previous Knowledge Channel Attention Module

3.1.1 Abstracted Overview of our PKCAM

By scrutinizing the aforementioned channel attention techniques, previous knowledge aggregation was not explored from the channel attention module perspective. Therefore we studied the previous knowledge cross-channel interaction by proposing PKCAM.

Fig.1 demonstrates our PKCAM, which exploits both local and global feature maps, through fusing two parallel attention paths while recalibrating the current feature map, where the first one consists of the local cross-channel interaction (LCCI) module that models channel-wise relationships of the aggregated feature map. We call this module local as it is operating on the current feature map only. The second path, which we call the global path, consists of two stacked modules: previous knowledge aggregation (PKA) and global cross channel interaction (GCCI). The PKA module covers the channel interactions across different preceding aggregated feature maps, while the GCCI module utilizes the refined features produce by the PKA module to model channel-wise relationships in a computationally efficient manner. Both LCCI and GCCI modules could be any one of the on-the-shelf channel attention techniques which are studied in Section 2.

3.1.2 Previous Knowledge Aggregation Block

In contradiction to the aforementioned channel attention techniques which relies on the current output of an arbitrary CNN block, our proposed PKCAM exploits both the current CNN block output , $x_0 \in \mathbb{R}^{H_0 \times W_0 \times C_0}$, and a range of earlier CNN blocks output , $X_p = [x_1, x_2, ..., x_R]$, where R is the coverage region that delimits how many previous CNN blocks output will be consolidated along side the current CNN block, $x_1 \in \mathbb{R}^{H_1 \times W_1 \times C_1}$, $x_2 \in \mathbb{R}^{H_2 \times W_2 \times C_2}$, and $x_R \in \mathbb{R}^{H_R \times W_R \times C_R}$.

In general, the earlier features X_p have different channel dimensions, as the conventional is as we go deeper the depth is increased. Therefore, the first operation in our Previous Knowledge Aggregation(PKA) block is aligning the channel dimension among different CNN blocks. As $C_0 \geq C_1 \geq C_2 \geq C_R$, aligning operation can be done be learnable upsampling techniques or a simple repeating operation to align with the channel dimension of the current CNN block C_0, producing channel aligned feature maps, $x'_0 \in \mathbb{R}^{H_0 \times W_0 \times C_0}$, $x'_1 \in \mathbb{R}^{H_1 \times W_1 \times C_0}$, $x'_2 \in \mathbb{R}^{H_2 \times W_2 \times C_0}$, and $x'_R \in \mathbb{R}^{H_R \times W_R \times C_0}$.

Analogous to aligning the channel dimensions, the spatial dimensions; H and W, is aligned through squeeze operation by adapting the general global average pooling equation as follows, $\widetilde{X} = \frac{1}{RWH} \sum_{k=1}^{R} \sum_{i=1}^{W} \sum_{j=1}^{H} X_{kij}$, where $\widetilde{X} \in \mathbb{R}^{R \times 1 \times 1 \times C_0}$ and represents the squeezed feature maps from the channel aligned aggregated feature maps x'_i, where i = 0, 1, …, R, producing $\widetilde{X} = [\widetilde{x}_0, \widetilde{x}_1, ..., \widetilde{x}_R]$, $\widetilde{x}_0 \in \mathbb{R}^{1 \times 1 \times C_0}$, $\widetilde{x}_1 \in \mathbb{R}^{1 \times 1 \times C_0}$ and $\widetilde{x}_R \in \mathbb{R}^{1 \times 1 \times C_0}$.

Previous knowledge cross-channel attention Given the aggregated feature $\widetilde{X}$, previous knowledge cross-channel attention can be learned by $Y = f(\widetilde{X})$, where $Y \in \mathbb{R}^{1 \times 1 \times C_0}$, $f(\widetilde{X}) = W' \widetilde{X}$, and W' could take one of the following forms,

$$W^{'} = \begin{cases} W_1^{'} = \begin{bmatrix} W_{1,1}^{'} & \cdots & W_{1,RC_0} \\ \vdots & \ddots & \vdots \\ W_{RC_0,1}^{'} & \cdots & W_{RC_0,RC_0}^{'} \end{bmatrix} \\ W_2^{'} = \begin{bmatrix} W_{1,1}^{'} & 0 & \cdots & 0 \\ 0 & W_{2,2}^{'} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W_{RC_0,RC_0}^{'} \end{bmatrix} \end{cases} \tag{12}$$

where W'_1 is a $RC_0 \times RC_0$ parameter matrix which learns previous knowledge interaction in conjunction with cross-channel interaction. In contrast W'_2 is a $1 \times RC_0$ parameter matrix which learns previous knowledge interaction and channel interaction neglecting the cross channel relations. The key difference between W'_1 and W'_2 is that W'_1 considers previous knowledge cross-channel interaction while W'_2 does not, leading W'_1 to be more complex than W'_2. Interpreting Eq. 12 to

neural networks W_1' and W_2' can be regarded as a fully connected layer and depth-wise separable convolution layer respectively. However, obviously from Eq. 12, W_1' and W_2' have a tremendous number of parameters, driving to high model complexity, especially for large channel numbers as mainly $C_0 >> R$.

Therefore, we divide learning the previous knowledge cross-channel interaction into two sub-modules as shown in Fig. 1, learning previous knowledge interaction, and exploiting the cross-channel interaction. Consequently, in contrast to Eq. 12, $f(\widetilde{X})$ is splitted into two cascaded functions, $f_1(\widetilde{X})$ and $f_2(\widetilde{X})$, where $f_1(\widetilde{X})$ is responsible to learn the previous knowledge channel interaction and $f_2(\widetilde{X})$ is responsible to learn the cross-channel interaction.

Previous knowledge channel interaction Previous knowledge channel attention can be learned by Eq. 13, where for each channel the global information is aggregated using simple summation operation, where no learnable parameters are invoked.

$$Y = f_1(\widetilde{X}) = \sum_{L=1}^{C_0} \sum_{K=1}^{R} \widetilde{X}_{LK} \tag{13}$$

A possible compromise between Eq. 12 and Eq. 13 is Eq. 14, where a tiny number of parameters are used whereas $W^{'} \in \mathbb{R}^{1\times1\times R}$ compared to the tremendous number of parameters that are invoked in Eq. 12 while learning the previous knowledge channel interaction. From the perspective of the convolution neural network, Eq. 14 could be readily interpreted to a 1-D convolution layer with kernel $k = \widetilde{W}$.

$$Y = f_1(\widetilde{X}) = \sum_{L=1}^{C_0} W^{'} \widetilde{X}_L \tag{14}$$

Global cross-channel interaction Global cross-channel interaction could be learned by adopting one of the local channel attention modules (15) (2) (38) (20) (37) (10) producing $Z_1 = f(Y)$ where $Z_1 \in \mathbb{R}^{1\times1\times C_0}$. (37) (10) (15) (2) (38) (20) refer to the term global as they are taking into consideration the whole spatial dimension from the fed features using GAP - Global Average Pooling. In contrast we refer to the term global as previous knowledge aggregation.

3.1.3 Combining global and local cross-channel interaction

PKCAM contains two cross-channel attention modules as shown in Figure 1, one learns the cross-channel interaction from the global feature map, while the other one learns the cross-channel interaction from the local feature map. For both modules ECA-Net (37) is adopted empirically based on Section 4.1.2. Finally, the current convolution output x_0 is recalibrated by the learned scales S, following Eq. 15, where $F \in \mathbb{R}^{1\times1\times C_0}$ and $S \in \mathbb{R}^{1\times1\times C_0}$ is obtained by fusing the global learned scales Z_1 with the local learned scales Z_2, following Eq. 16.

$$F = \widetilde{x}_{\widetilde{W}} \odot \tag{15}$$

$$S = \phi(Z) : Z = Concat(Z_1, Z_2) \tag{16}$$

where $\phi(Z)$ represents the mapping function from the global and local scales to the final scales S. The simplest mapping is summing both local and global scales as shown in Eq. 17, where no learnable parameters are included. We called this, a shallow fusion mechanism, as a fixed mapping is followed regardless of the feature map's structure and nature.

$$\phi(Z) = \sum_{L=1}^{C_0} \sum_{K=1}^{2} Z_{LK} \tag{17}$$

Other possible mapping function is considering the full cross-channel interaction between local and global scales, which could be regarded as a fully connected layer with tremendous number of parameters. But we waive this costly full cross-channel interaction to abide by our objective which

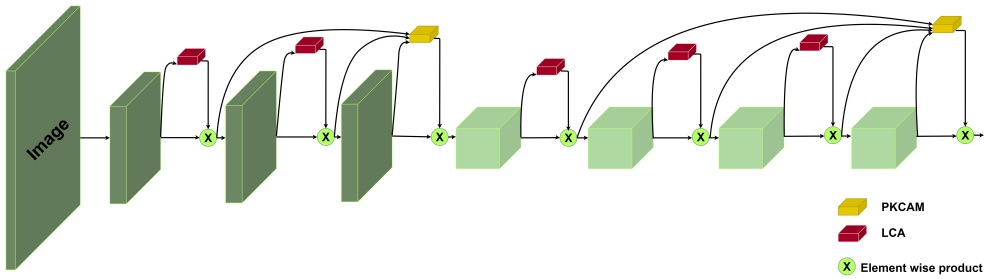

Figure 2: Illustration for the general idea of integrating PKCAM Module into an arbitrary Deep CNNs alongside local channel attention (LCA) module.

Table 2: Comparison of the various ways to integrate PKCAM into CNNs using the ImageNet dataset. Top-1 accuracy is reported.

Integration type	All blocks	Last block
Resnet-18	71.1	**71.15**
Resnet-34	74.25	**74.43**
Resnet-50	77.50	**77.56**

aims to learn channel attention more efficiently in a lightweight manner besides avoiding the shallow summation fusion mechanism Eq. 17, therefore Eq. 18 can be considered as a compromised mapping function, where we only concern about the channel interaction between each local channel and its counterpart in the global scales, which could be achieved easily using a 1-D convolution layer with kernel size equals two.

$$\phi(Z) = \begin{bmatrix} W_Z^1 & W_Z^2 \end{bmatrix} \otimes \begin{bmatrix} Z_1 & \cdots & Z_1^{C_0} \\ Z_2 & \cdots & Z_2^{C_0} \end{bmatrix} \tag{18}$$

3.1.4 Integrating PKCAM Module into Deep CNNs

Figure 2 illustrate a general way for integrating PKCAM into an arbitrary CNN architecture, where PKCAM was integrated into the last CNN block for each stage alongside arbitrary local channel attention (LCA) module for the rest of the blocks. Due to the lightweight topology of our PKCAM, it could be integrated to each block, where LCA is totally replaced. Figure 3 demonstrate a pseudo code for our PKCAM to show how easily it could be integrated to any CNN architecture.

4 Experiments

In this section, we perform controlled ablation experiments to settle on the best design for our proposed module and assess its sub-modules. Then we evaluate the performance of the proposed Previous Knowledge Attention module, on a series of benchmark datasets across different tasks including Tiny-ImageNet (19), and ImageNet (6) for the classification task, and KITTI (11) for detection. Finally, We conduct empirical experiments that probe the robustness of the representations learned by PKCAM, compared to convolutional baselines and other attention mechanisms.

4.1 Ablation studies and analysis

We have conducted three ablations studies to settle on the best architecture design and analyze the effectiveness of each component in our PKCAM. The first one investigating the different approaches for previous knowledge channel interaction that are discussed in Section 3.1.2. Then, we assess the choice of basic attention modules that are used in the local and global cross channel interaction modules as shown in Fig. 1. Finally, we demonstrate the effectiveness of our proposed previous knowledge cross-channel interaction module compared to the naive local cross-channel interaction.

```
def PKCAM_Module():
    """ Calculate the channel's importance using our proposed PKCAM module.
        - x: current feature map, of shape [N, C_x, H, W]
        - y: List of feature maps from latter layers, of length S.
             Each feature map of shape [N, C_y, H_y, W_y]
        - PKA  --> Previous Knwledge Aggregation.
        - GCCI --> Global Cross Channel Interaction.
        - LCCI --> Local Croos Channel Interaction.
    """
    K_size = 3
    N, C_x, H, W = x.size()
    S = len(y)
    PKA    = Conv1D(in_c=1, out_c=1, kernel=S,
                    stride=S, padding=0, bias=False)
    LCCI   = Conv1D(in_c=1, out_c=1, kernel=K_size,
                    padding=(K_size - 1) // 2, bias=False)
    GCCI   = Conv1D(in_c=1, out_c=1, kernel=K_size,
                    padding=(K_size - 1) // 2, bias=False)
    Fusion = Conv1D(in_c=1, out_c=1, kernel=2, stride=2,
                    padding=0, bias=False)

    x = spatial_squeeze(x)              # [N, C_x, H, W]-->[N, C_x]
    for i in range(S):
        N, C_y, H_y, W_y = y.size()
        y[i] = spatial_squeeze(y[i])    # [N, C_y, H_y, W_y]-->[N, C_y]
        if C_y not equal C_x:
            y[i] = upsample(y[i])       # [N, C_y]-->[N, C_x]
        y = Reshape(y)                  # [S, N, C_x]-->[N, 1, S*C_x]
        y = PKA(y)                      # [N, 1, S*C_x]-->[N, 1, C_x]
        x = LCCI(x)                     # [N, 1, C_x]-->[N, 1, C_x]
        y = GCCI(y)                     # [N, 1, C_x]-->[N, 1, C_x]
        y = stack(x, y)                 # [N, 1, C_x]-->[2, N, 1, C_x]
        y = Fusion(y)                   # [2, N, 1, C_x]-->[N, 1, C_x]

        return x*y.expand_as(x)
```

Figure 3: Pseudo code for our PKCAM.

4.1.1 Global and local cross-channel interaction

We have investigated empirically the different global channel interaction techniques that were described in detail in Section3.1.2, besides examined the various global-local fusion mechanisms that combine the global and the local cross-channel learned representations which were cover in Section3.1.3. To cover the whole possible combinations of each sub-module, nine experiments are conducted for each adopted backbone, which are ResNet-18 (13), ResNet-34 (13), ResNet-50 (13). All experiments in Table 3 are conducted using the Tiny-ImageNet dataset (19), where the same data augmentation and hyper-parameter settings in (15) are adopted. Input images are randomly cropped to 64×64 with random horizontal flipping. Stochastic gradient descent (SGD) with weight decay of $1e{-}4$, the momentum of 0.9, and mini-batch size of 32 is used. Models are trained for 100 epochs from scratch, using the weight initialization strategy described in (12) and the initial learning rate is set to 0.1 and decreased by a factor of 10 every 30 epochs. As shown in Table3 exploiting the previous knowledge channel interaction through 1-D Conv. layer by following Eq.14, and fuse the global and local scales by following Eq.18, is the best compromise to solve the paradox of performance and complexity trade-off, where it shares almost the same model complexity (i.e., network parameters and FLOPs) with the original ResNet while at the same time it achieves the best accuracy for ResNet-50 and comparable accuracy for ResNet-18 and ResNet-34 compared to fusing the global and local scales using fully connected layer which invokes a tremendous number of parameters. Based on the aforementioned results in Table 3, our novel approach follows the compromised combination while exploiting the previous knowledge cross channel interaction by following Eq.14 to capture the previous knowledge channel interaction and Eq.18 to fuse the global and local learned representations.

Table 3: Comparison of different previous knowledge Aggregation(PKA) techniques on the horizontal dimension, where 1-D Conv., Sum and FC stands for one dimensional convolution layer Eq.14, summation Eq.13, and fully connected layer Eq.12 respectively. While the vertical dimension compares various global-local fusion mechanisms, where 1-D Conv., Sum and FC stands for one dimensional convolution layer Eq.18, summation Eq.17, and fully connected layer respectively.

Fusion \ MSCI		1-D Conv.	Sum	FC	1-D Conv.	Sum	FC	1-D Conv.	Sum	FC
		ResNet-18			ResNet-34			ResNet-50		
1-D Conv.	Acc.	55.70	55.28	54.63	**56.94**	56.26	56.52	**57.89**	56.18	56.41
	#.P (M)	**10.749**	10.749	11.413	**20.389**	20.389	23.049	**22.824**	22.824	65.387
	GFLOPs	**2.075**	2.075	2.076	**4.329**	4.329	4.331	**4.878**	4.878	4.878
Sum	Acc.	54.96	55.10	54.24	56.00	56.02	55.40	57.14	56.55	57.16?
	#.P (M)	10.749	10.749	11.413	20.389	20.389	23.049	22.824	22.824	65.387
	GFLOPs	2.075	2.075	2.076	4.329	4.329	4.331	4.878	4.878	4.878
FC	Acc.	**56.01**	55.35	54.41	56.20	56.38	56.64	57.53	56.69	56.95
	#.P (M)	11.413	11.413	12.077	22.124	22.124	24.784	50.574	50.574	93.137
	GFLOPs	2.076	2.076	2.076	4.330	4.330	4.333	4.906	4.906	4.947

Table 4: Comparison of the various basic attention modules in our proposed module PKCAM using the Tiny-ImageNet dataset.

Basic Attention Module	SE	SRM	ECA
Resnet-18	54.07	55.10	**55.70**
Resnet-34	56.61	56.52	**56.94**
Resnet-50	57.42	57.12	**57.89**

4.1.2 Basic attention module

We next assess the choice of basic attention modules that are used in the local and global cross channel interaction modules as shown in Fig.1. Three channel attention mechanisms are evaluated on the Tiny-Imagenet dataset (19), including SE-Net (15), SRM (20), and ECA-Net (37). As shown in Table 4 ECA-Net achieves the best accuracy across different ResNet backbones. However, building up our PKCAM using other channel attention mechanism boost the accuracy compared to their original results. For example, SRM using ResNet-18 as a backbone achieves 53.39% while our PKCAM module builds upon SRM achieves 55.1%.

4.1.3 Local Vs. global cross channel interaction

We conduct experiments to validate the effectiveness of our proposed global cross-channel interaction module by comparing it with the naive local cross-channel interaction. Experiments at Table 5 are conducted using the Tiny-Imagenet dataset (19) and ECA-Net (37) as basic channel module as discussed at Section 4.1.2. Table 5 shows that the global cross-channel interaction module alone achieves better accuracy than the local one, and combining both of them achieves the best accuracy. Results that are shown in Table 5 suggest that the recalibration scales learning process will benefit from global information.

Table 5: Showing effectiveness of previous knowledge cross-channel interaction module using the Tiny-ImageNet dataset.

Local Vs. Global	Local	Global	Both
Resnet-18	53.76	54.82	**55.70**
Resnet-34	55.66	56.18	**56.94**
Resnet-50	56.59	56.89	**57.89**

Table 6: Comparisons with state-of-the-art attention modules on ImageNet in terms of the number of parameters (#P.) in millions, GFLOPs, top-1, and top-5 accuracy. Top-1 relative improvement results is reported between parentheses w.r.t SENet improvement over Vanilla Resnet.

Methods	#.P.(M)	GFLOPs	Top-1	Top-5	#.P.(M)	GFLOPs	Top-1	Top-5	#.P.(M)	GFLOPs	Top-1	Top-5
	ResNet-18				ResNet-34				ResNet-50			
ResNet (13)	11.14	1.699	70.42	89.45	20.78	3.427	73.31	91.40	24.37	3.86	75.2	92.52
SENet (15)	11.23	1.700	70.59	89.78	20.93	3.428	73.87	91.65	26.77	3.87	76.71	93.38
CBAM (39)	11.23	1.700	70.73(182%)	89.91	20.94	3.428	74.01(125%)	91.76	26.77	3.87	77.34(141%)	93.69
ECA (37)	11.14	1.699	70.78(211%)	89.92	20.78	3.427	74.21(160%)	91.83	24.37	3.86	77.48(151%)	93.68
PKCAM	**11.14**	**1.699**	**70.98(329%)**	**90.12**	**20.78**	**3.427**	**74.43(200%)**	**91.87**	**24.37**	**3.86**	**77.56(156%)**	**93.70**

Table 7: Comparisons with state-of-the-art attention modules on KITTI-RGB in terms of mAP using YOLOV3 on Resnet-18 and 34 backbones.

	Vanilla	SE	ECA	CBAM	BAM	SRM	PKCAM
R-18	57.87	59.32	58.55	57.90	59.61	59.20	**59.66**
R-50	64.19	65.08	64.34	64.18	65.10	64.82	**65.21**

4.2 Image classification on ImageNet

In this section, we evaluate the performance of proposed PKCAM network on ImageNet (6). All the classification experiments follows the same training procedure, where the same data augmentation and hyper-parameter settings in (15) are adopted. Models are trained for 100 epochs from scratch, using the weight initialization strategy described in (12) and the initial learning rate is set to 0.1 and decreased by a factor of 10 every 30 epochs. Stochastic gradient descent (SGD) with weight decay of $1e{-}4$, the momentum of 0.9, and mini-batch size of 256 for ImageNet (6). The evaluation metrics incorporate both efficiencies (i.e., network parameters (#P.) in millions, and floating-point operations per second (FLOPs) in Gigas) and effectiveness (i.e., Top-1 accuracy).

ImageNet LSVRC 2012 dataset (6), which contains 10^3 classes with 1.2 million training images, 50×10^3 validation images, and 100×10^3 test images. The evaluation is measured on the non-blacklist images of the ImageNet LSVRC 2012 validation set. A 224×224 crop is randomly sampled from an image or its horizontal flip, with the per-pixel RGB mean value subtracted.

We compare our PKCAM module with several state-of-the-art attention methods using ResNet-18 and ResNet-34 backbones (13) on ImageNet. Efficiency and effectiveness are measured, and the results are reported in Table 6 from their original papers. We adopt the same training setup as (13) (15) for fair comparison. Results show that our proposed PKCAM achieves the best accuracy besides be the lightest model compared to other attention modules (15) (39). Top-1 relative improvement results is reported between parentheses w.r.t SENet improvement over Vanilla Resnet.

4.3 Object detection

KITTI-RGB (11) consists of 7,481 training images and 7,518 test images, comprising a total of 80,256 labeled objects of eight different classes. Each image has 3 RGB color channels and pixel dimensions 1242×375 which is resized to 224×224. We follow the same training setup as mentioned at Section 4.2. As shown in Table 7, PKCAM considerably improves the accuracy more than other attention modules compared to the baseline (13). YOLOV3 (27) detector is used.

5 Conclusion

In this paper, we concentrate on determining an effective channel attention module with low model complexity. To this end, we propose efficient channel attention (PKCAM). Because of the lightweight computation of the PKCAM block, it can be integrated into all modern CNN architectures across the whole layers and trained end-to-end. While most previous works utilized uni-scale features, PKCAM is designed to employ the ability of global information while recalibrating feature maps. Our experiments demonstrate that simply inserting PKCAM into standard CNN architectures boosts

the performance across different tasks. Furthermore, we verify the robustness of the representations learned by PKCAM and its generalization ability via zero-shot experiments to rotated images.

References

[1] Diane M Beck and Sabine Kastner. Top-down and bottom-up mechanisms in biasing competition in the human brain. *Vision research*, 49(10):1154–1165, 2009.

[2] 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/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020.

[4] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5659–5667, 2017.

[5] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3640–3649, 2016.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] Robert Desimone. Visual attention mediated by biased competition in extrastriate visual cortex. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 353(1373):1245–1255, 1998.

[8] Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995.

[9] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

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

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[13] 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.

[14] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. *arXiv preprint arXiv:1810.12348*, 2018.

[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.

[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.

[17] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[19] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7, 2015.

[20] HyunJae Lee, Hyo-Eun Kim, and Hyeonseob Nam. Srm: A style-based recalibration module for convolutional neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1854–1862, 2019.

[21] Drew Linsley, Dan Shiebler, Sven Eberhardt, and Thomas Serre. Learning what and where to attend. *arXiv preprint arXiv:1805.08819*, 2018.

[22] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 4905–4913, 2016.

[23] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 3074–3082, 2015.

[24] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. *arXiv preprint arXiv:1406.6247*, 2014.

[25] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *European conference on computer vision*, pages 483–499. Springer, 2016.

[26] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*, 2018.
[27] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
[29] Abhijit Guha Roy, Nassir Navab, and Christian Wachinger. Recalibrating fully convolutional networks with spatial and channel "squeeze and excitation" blocks. *IEEE transactions on medical imaging*, 38(2):540–549, 2018.
[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[31] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. *arXiv preprint arXiv:1507.06228*, 2015.
[32] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
[33] 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.
[34] 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.
[35] 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.
[36] 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.
[37] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
[38] 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.
[39] 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.
[40] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10076–10085, 2020.