

RESEARCH ARTICLE

EMCA: Efficient Multiscale Channel Attention Module

ESLAM MOHAMED BAKR^{1,2}, AHMAD EL-SALLAB¹, AND MOHSEN RASHWAN²¹Research and Development, VALEO, Giza 12577, Egypt²Faculty of Engineering, Cairo University, Giza 12613, Egypt

Corresponding author: Eslam Mohamed Bakr (eslam.mohamed-abdelrahman@valeo.com)

ABSTRACT Attention mechanisms have been explored with CNNs across the spatial and channel dimensions. However, all the existing methods devote the attention modules to capture local interactions from a uni-scale. This paper tackles the following question: can one consolidate multi-scale aggregation while learning channel attention more efficiently? To this end, we avail channel-wise attention over multiple feature scales, which empirically shows its aptitude to replace the limited local and uni-scale attention modules. EMCA is lightweight and can efficiently model the global context further; it is easily integrated into any feed-forward CNN architectures and trained in an end-to-end fashion. We validate our novel architecture through comprehensive experiments on image classification, object detection, and instance segmentation with different backbones. Our experiments show consistent gains in performances against their counterparts, where our proposed module, named EMCA, outperforms other channel attention techniques in accuracy and latency trade-off. More specifically, compared to SENet, we boost the accuracy by 0.8 %, 0.6 %, and 1 % on ImageNet benchmark for ResNet-18, 34, and 50, respectively. For detection and segmentation tasks, MS-COCO are for benchmarking, Our EMCA module boost the accuracy by 0.5 % and 0.3 %, respectively. We also conduct experiments that probe the robustness of the learned representations. Our code will be published once the paper is accepted.

INDEX TERMS Channel attention module, deep learning, machine learning, computer vision, object classification, CNN backbones, CNN encoders, CNNs, convolutions, image processing.

I. INTRODUCTION

Over the years, CNN architectures have developed many ideas to better deal with spatial image features. Moreover, their limited receptive field makes such features lack the global view of the image. As a result, deeper architectures emerged that stack multiple convolution layers, known as backbone or encoder. The main advantage of such architectures is their ability to cover spatial features at multiple scales. As we go deeper in the network, the feature maps get smaller, while their content represents a broader region in the space, which puts us closer to better semantics of the image contents. With the emergence of AlexNet [1], various research has been conducted to improve deep CNNs' performance further. References [2], [3], [4], [5], [6] have sought to strengthen the CNNs by making them 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 consolidating attention mechanisms.

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 [7] 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 and visual attention [8], [9], [7], [10] are enhanced by top-down stimuli, and non-relevant neurons are suppressed in feedback loops. Inspired by the human visual system, various attention mechanisms [11], [12], [13], [14], [15] 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 [16] and [17] or across channel-wise dimension as


The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy .

TABLE 1. Comparison of various channel attention modules (CA-modules). Where MS indicates whether multi-scale (MS) cross-channel interactions are used or not, Dim. determines the attention dimension, where C indicates channel attention and S indicates spatial attention, and finally, the light indicates whether the CA-module is lighter than SENet or not in terms of the number of parameters and FLOPS.

	SE	ECA	SRM	GSop	GC	GE	CBAM	BAM	DAN	GALA	RAN	EMCA
MS	×	×	×	×	×	×	×	×	×	×	×	✓
Dim.	C	C	C	C	C	S	C + S	C + S	C + S	C + S	C + S	C
Light	-	✓	✓	×	×	×	×	×	×	×	×	✓

in [11], [12], [18], [19], and [20], or both dimensions as in [21], [22], [15], [23], [24], [25], and [26]. 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 is why we call it uni-scale or local attention mechanisms.

Employing multi-scale feature maps has been applied to image classification [27], [28], [29], image segmentation [30], tracking [31], and human pose estimation [32], where they obtain enhanced performance. Driven by the significance of employing multi-scale while learning different tasks [27], [28], [29], [30], [31], [32], a question arises: How can one incorporate multi-scale aggregation while learning channel attention more efficiently?

To answer this question, we introduce EMCA, 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 emphasize informative features and suppress less useful ones selectively. In contrast to the attention mechanisms mentioned earlier, our multi-scale attention block obtains additional inputs from all preceding attention blocks. It passes its refined feature maps to all subsequent blocks, creating global awareness by exploiting multi-scale aggregation. We use the previous larger scales from earlier layers that can capture fine-grained information, which is helpful for precise localization while attending to features from the last layers that can encode abstract semantic information, which is robust to target appearance changes.

Our contributions are summarized as follows:

- We propose a simple and effective attention module, EMCA, which can be integrated easily with any CNN backbone due to the lightweight computation of our novel architecture.
- We verify the effectiveness and robustness of EMCA throughout extensive experiments with various baseline architectures on multiple tasks and datasets.
- Through detailed analysis along with ablation studies, we examine the internal behavior and validity of our method.

The rest of the paper is organized as follows. First, we discuss the related work, followed by the details of the proposed model. Then we present detailed ablation studies to settle

on the best architectural design, and finally, illustrate the experimental setup for the various experiments we conducted for every contribution.

II. RELATED WORK

A. MULTI-SCALE

The Gaussian Scale-Space Paradigm [33] has explored the multi-scale contribution while representing an image and mapped its local behavior as a function of scales and resolution. This technique was applied to interpolation, extrapolation, image enhancement, and deblurring. Reference [34] combines multiple local cues into a globalization framework based on spectral clustering, which was applied to the contour detector problem by transforming its output into a hierarchical region tree. Reference [30] tackles semantic segmentation problem by adapting DeepLab [35] by joining multi-scale input images and the attention model for handling different input resolutions.

References [36], [37], [38], [39] refer to multi-scale as they resize the input image to different resolutions and fuse them on the input or output level. References [40], [41], [35], [42], [28] learn finer-scale prediction from lower layers, where these techniques use multi-scale features instead of multi-scale input resolution. For instance, [43] aims to tackle the fact that the high-frequency information and details in the low-resolution image are hard to be reconstructed by proposing a multi-scale generative adversarial network. Where [43] utilizes a pyramid module inside the generator to extract the features containing high-frequency information and to capture the multi-level features. GasHis-Transformer [44] introduces a multi-scale visual transformer model to tackle the Gastric Histopathological Image Detection (GHID), to enable the automatic global detection of gastric cancer images by integrating the describing capability of the global and the local information of vision-transformers and CNN's.

ABFPN [45] proposes an enhanced multi-scale feature fusion method to improve the detection performance of small objects by offering the atrous spatial pyramid pooling-balanced-feature pyramid network, termed ABFPN. ABFPN utilizes the atrous convolution operators with different dilation rates to fully use the context information and the skip connections to achieve sufficient feature fusions.

MU-Net [46] achieves accurate and low-cost remote sensing image registration by proposing a multi-scale framework with unsupervised learning. MU-Net stacks several deep neural network models on multiple scales to generate a coarse-to-fine registration pipeline to directly learns the end-to-end mapping from the image pairs to their transformation parameters.

CrossViT [47] builds above the tremendous success of the ViT [48] and explores how to learn multi-scale feature representations in transformer models for image classification by proposing a dual-branch transformer to combine image patches (i.e., tokens in a transformer) of different sizes to produce more powerful image features.

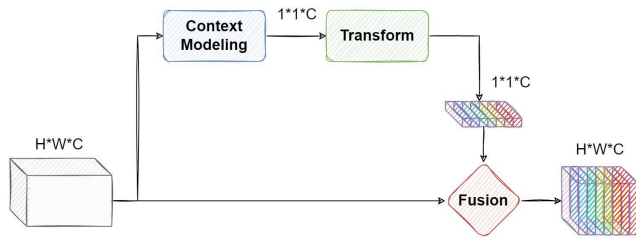


FIGURE 1. Abstract overview of the channel attention modules.

B. CHANNEL ATTENTION

Figure 1 depicts an abstract overview of general channel attention paradigm, where an arbitrary convolution layer is first feed to the context modeling module to squeeze the spatial dimensions ($H \times W$) followed by transform block which aims to learn correlation between channels C , then finally weight each channel by its importance factor. An arbitrary channel attention module could be formulated as three sub-blocks, i.e., context module, transformation and fusion. Where context modeling module aims to squeeze the spatial dimensions and keep the channel dimension only. Then, transform block learns the channel importance and the cross-correlation between different channels. Finally, the fusion block responsible to re-weight each channel based on its importance.

SENet [18] proposed SE block, squeeze, and excitation block, which comprises a lightweight gating mechanism that focuses on enhancing the representational power of the network by modeling channel-wise relationships using two fully connected layers. ECA-Net [20] empirically shows avoiding dimensionality reduction in [18] by using a simple 1-D convolution layer is essential for learning channel attention and appropriate cross-channel interaction. SRM [19] proposes a Style-based Recalibration Module, which adaptively recalibrates intermediate feature maps by exploiting their styles. Reference [13] explore two variations of self-attention, pairwise and patchwise, that produce more powerful refined features. The basic non-local block (NLB) [12] aims to strengthen the query position's features via aggregating information from other positions. GC-Net [11] introduces an abstract global context modeling framework that could be summarized into two blocks: context modeling and transform block, besides proposing a simplified local network as the context modeling and using SENet [18] as the transform block. GSoP [49] obtains a covariance matrix by exploiting holistic image information using global second-order pooling, which is used for tensor scaling along channel dimensions. ResNeSt [50] presents a modularized architecture that applies channel attention to different network branches to capture cross-feature interactions, where the feature is divided into several groups, then a series of transformations are applied to each group. CoordAttention [51] propose an attention mechanism limited for mobile networks only, where positional information is added, named coordinate attention.

EPSANet [52] proposes an efficient pyramid squeeze attention block on a convolutional neural network.

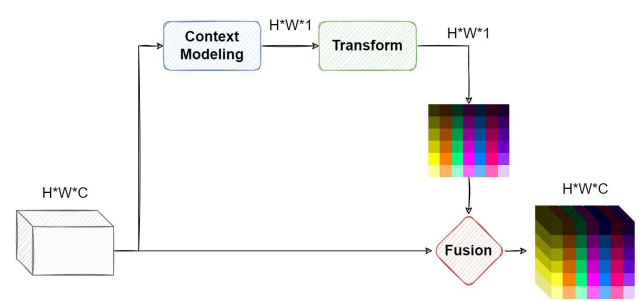


FIGURE 2. Abstract overview of the spatial attention modules.

EPSANet [52] aims to design an efficient and effective channel attention module; thus, two modules are introduced by EPSANet [52]. The first is the pyramid squeeze attention module, termed PSA, and the second is squeeze and concat, termed SPC. First, SPC generates multi-scale feature maps, which are then processed by an excitation module called PSA. PSA is a repeated SE-modules [18] which predicts scales indicating the importance of each channel, then fuse the anticipated scales and multiplies them with the original feature map. FcaNet [53] proposes a frequency channel attention networks that formulates the channel attention block as compression process using frequency analysis. Driven by an interesting finding that the feature decomposition in the frequency domain is a general formulation for the conventional global average pooling. Therefore, FcaNet [53] introduced a multi-spectral channel attention module. LAN [54] proposes a lightweight attention-based network that employs the channel attention modules to tackles the smartphones' limitations in both, size and cost, which negatively impact on the quality of the implemented sensors, through learning the input mosaic and an unsupervised pre-training strategy. FL-CSE-ROIE [55] proposes a full-level context squeeze-and-excitation ROI extractor alongside FPN to capture multi-scale features to boost the instance segmentation performance. To ease the background interference, FL-CSE-ROIE adds multi-context surroundings of different scopes to ROIs generated from FPN, by utilizing SENet [18]. ESE-FN [56] proposes nonlinear multi-modal fusion approach by utilizing nonlinear attention mechanism that is extended from Squeeze-and-Excitation Networks; SENet [18], to tackle the elderly activity recognition.

C. SPATIAL ATTENTION

Figure 2 depicts a general overview of the spatial attention modules. Driven by the formulation of the channel attention modules, an arbitrary spatial attention module on high-level mimics the channel attention module, which could be formulated as three sub-blocks, i.e., context module, transformation, and fusion. The context module squeezes the channel dimension while keeping the spatial dimension; then, the transformation module learns the correlation between the spatial locations.

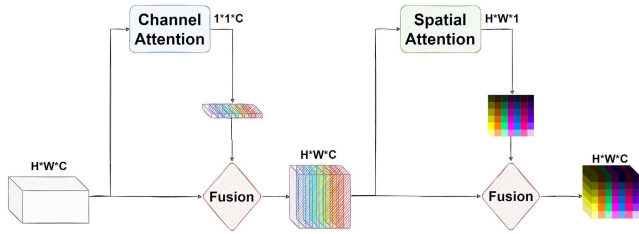


FIGURE 3. Abstract overview of the heterogeneous architecture that employ both spatial and channel attention modules.

GENet [17] consists of two operators that also follow the context modeling framework [11], gather and excite operators. GENet [17] uses stridden depth-wise convolution, which acts as the gather operator. The gather operator applies spatial filters to independent input channels, and a simple excite operator consists of sigmoid function and multiplication. Spatial Transformer Networks [57] tackle the lack of CNN ability to be spatially invariant to the input by integrating a learnable module, the Spatial Transformer, which can be inserted into CNNs, giving neural networks the ability to actively spatially transform feature maps, conditional on the feature map itself. DETR [16] stacks a spatial transformer after the CNN backbone to learn the interaction between each spatial position and its effect on different vision tasks, object detection, and instance segmentation.

D. SPATIAL AND CHANNEL ATTENTION

Figure 3 depicts a general overview of the heterogeneous architecture that employ both spatial and channel attention modules. The majority of the exiting architectures attend along the channel dimension first followed by a spatial attention module.

BAM [21], CBAM [22], DANet [23], Residual attention network [15], SCA-CNN [26], scSE [25] and GALA [24] show that taking the spatial axis into consideration besides channel axis boost the attention module accuracy. Given an intermediate feature map, they sequentially infer attention maps along two separate dimensions, channel and spatial, then the attention maps are multiplied to the input feature map for adaptive feature refinement. RAN [15] proposes a Residual Attention Network, which is built by stacking complex Attention Modules to generate attention-aware features that are changed adaptively in each layer. The Attention Module used in [15] is complex because of using bottom-up and top-down feedforward structure and due to inserting trunk-and-mask attention mechanism based on hourglass modules [32] between the intermediate stages. CANet [58] tackles the RGB-D semantic segmentation task by proposing a co-attention network to construct a proper interaction between RGB and depth features. CANet mainly proposes a co-attention fusion module that utilizes the position and channel co-attention to adaptively fuse RGB and depth features in spatial and channel dimensions. Several fusion co-attention modules are employed to obtain a more representative feature that is crucial for the semantic segmentation task.

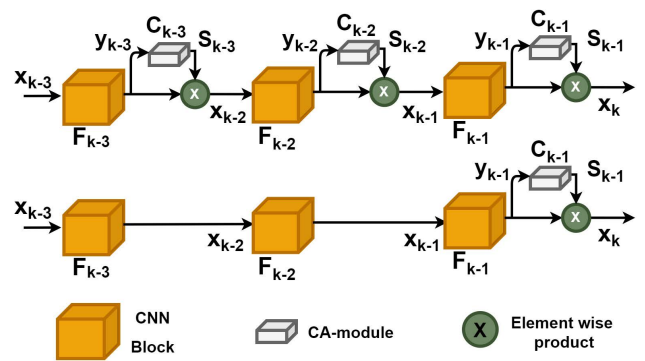


FIGURE 4. Abstract overview demonstrates two possibilities of integrating a channel attention module (CA-module) into an arbitrary CNN backbone. The upper part shows the dense integration mechanism that is followed by the existing channel attention modules. The lower part shows our proposed integration mechanism.

Table 1 summarizes the existing attention modules regarding whether multi-scale cross-channel interactions are incorporated or not, lightweight model, and attention type.

III. METHODOLOGY

In this section, we first revisit the integration mechanism that is used in the channel attention modules. Then, we show the drawbacks of the used integration mechanism mathematically and empirically, which we name dense integration. Accordingly, we are motivated to incorporate the multi-scale information while designing an efficient channel attention module by proposing our EMCA module. In addition, we dissect our EMCA module by detailing its main blocks.

A. REVISITING INTEGRATING CHANNEL ATTENTION MODULES

Let the function $F(x)$ represents a CNN block that consists of successive CNN layers interspersed with non-linear activation. Given an arbitrary input $x \in \mathbb{R}^{H_i \times W_i \times C_i}$, an output $y \in \mathbb{R}^{H_o \times W_o \times C_o}$ is generated using the mapping function F , where H_i , W_i , and C_i are height, width, and channel dimensions for the input x , while H_o , W_o , and C_o are height, width, and channel dimensions for the output y . The upper part of Figure 4 shows the integration mechanism followed by the existing channel attention modules (CA-module), discussed in Section II.

The CA-module C is attached at the tail of each CNN block F to generate meaningful scales S , representing the importance of each channel. Then the CNN output y is refined by multiplying it by the learned scales S , producing a refined input x for the next CNN block. We call this integration mechanism dense integration, as the CA-module is plugged into the network after each CNN block, which can be represented by Equation 1 as follows:

From the above equation, CA-module’s output relies on the outputs from all the previous layers, increasing the gradient path’s length. Increasing the gradient path complicates the backpropagation process. Moreover, during the backpropagation process for an arbitrary refined CNN output x_k , the gradients of all preceding CNN blocks

$F_{(k-1)}, F_{(k-2)}, \dots, F_{(1)}$ are taken into consideration as shown in Equation 2, which represents the updating steps for the CNN weights as follows:

$$\begin{aligned} x_k &= C_{k-1} (y_{k-1}) \otimes y_{k-1} \\ &= C_{k-1} \left(F_{k-1} (x_{k-1}) \right) \\ &\quad \otimes \left[C_{k-2} (F_{k-2} (x_{k-2})) \right. \\ &\quad \left. \otimes \left[\dots \otimes [C_0 (F_0 (x_0)) \otimes x_0] \right] \right] \end{aligned} \quad (1)$$

$$\begin{aligned} W'_1 &= U_1 (W_1, \{g_{C_0}, g_{F_0}\}) \\ W'_2 &= U_2 (W_2, \{g_{C_0}, g_{F_0}, g_{C_1}, g_{F_1}\}) \\ &\quad \vdots \\ W'_k &= U_k (W_k, \{g_{C_0}, g_{F_0}, g_{C_1}, g_{F_1}, \\ &\quad \dots, g_{C_{k-1}}, g_{F_{k-1}}\}) \end{aligned} \quad (2)$$

B. EFFICIENT MULTI-SCALE CHANNEL ATTENTION (EMCA) MODULE

1) AVOIDING DENSE INTEGRATION

Unlike the existing CA-modules that emphasize the internal design neglecting to study the best integration method, we propose a more efficient integration mechanism that avoids the dense integration technique, discussed in Section III-A. As demonstrated above, the dense integration, i.e., Equation 1, will cause a large duplicated amount of gradients used while updating the CA-modules weights, i.e., Equation 2.

Accordingly, we avoid dense integration by proposing a more light and efficient integration mechanism. The architecture of our proposed integration mechanism is shown in the lower part of Figure 4. In addition, to avoid using duplicated gradients while updating CA-module weights, we integrate it into the last CNN block only instead of integrating it into each CNN block. The equations of the feed-forward pass and the weight updating of our mechanism are shown in Equations 3 and 4, respectively.

$$\begin{aligned} x_k &= C_{k-1} (y_{k-1}) \otimes y_{k-1} \\ &= C_{k-1} \left(F_{k-1} (x_{k-1}) \right) \\ &\quad \otimes F_{k-2} \left(F_{k-1} \left(\dots F_0 (x_0) \right) \right) \end{aligned} \quad (3)$$

$$\begin{aligned} W'_1 &= U_1 (W_1, \{g_{F_0}\}) \\ W'_2 &= U_2 (W_2, \{g_{F_0}, g_{F_1}\}) \\ &\quad \vdots \\ W'_k &= U_k (W_k, \{g_{F_0}, g_{F_1}, \\ &\quad \dots, g_{F_{k-1}}, g_{C_{k-1}}\}) \end{aligned} \quad (4)$$

From the above equations, we can see that the refined output x_k relies only on the outputs from the associated CA-module C_{k-1} instead of relying on the whole previous CA-module, as shown in Equation 1.

TABLE 2. Comparison of various integration mechanisms, i.e., All, First, and Last, for integrating CA-module into Deep CNN backbone. Where All, First, and Last determine whether the CA-module will be integrated into the whole CNN blocks or the first or the last CNN block only, respectively. ResNet-18, 34, and 50 are used on the ImageNet dataset.

		FPS	#P (M)	Top-1	FPS	#P (M)	Top-1	FPS	#P (M)	Top-1
		SE			ECA			SRM		
All		187	11.231	70.59	192	11.148	70.75	154	11.152	70.96
First	R-18	204	11.189	70.91	212	11.148	70.63	165	11.150	71.31
Last		204	11.189	70.92	212	11.148	70.81	165	11.150	71.04
All		101	20.938	73.87	107	20.788	74.13	82	20.795	73.98
First	R-34	122	20.829	73.84	122	20.788	74.20	96	20.790	74.51
Last		122	20.829	73.64	122	20.788	73.75	96	20.790	73.63
All		90	26.772	76.80	87	24.373	77.12	71	24.402	77.13
First	R-50	97	25.037	76.56	98	24.373	77.02	81	24.380	76.98
Last		97	25.037	75.71	98	24.373	76.37	81	24.380	76.73

As shown in Table 2, avoiding dense integration pays off in terms of speed across different network sizes, ranging from ResNet-18 to ResNet-50. However the accuracy decreased for the extensive backbones, i.e., ResNet-34 and ResNet-50. This degradation in the accuracy is justified by dropping the feature reuse advantages by avoiding the dense integration, especially for the large networks which contain more blocks at each stage, e.g., ResNet-50 contains 3, 4, 23, 3 CNN blocks at each stage, respectively. Thus, we have to preserve the advantages of the feature reuse characteristics and capture the long-range dependencies, but at the same time prevent the excessive amount of duplicate gradient information.

2) MULTI-SCALE INCORPORATION

Driven by the above analysis, the network must have some mechanism to effectively process and consolidate features across different scales from the preceding CNN blocks. By scrutinizing the channel attention techniques mentioned earlier, as presented in Table 1, multi-scale aggregation was not explored from the channel attention module perspective. In contradiction to the channel attention techniques as mentioned above, which relies on an arbitrary CNN block's output, our proposed EMCA module, as shown in Figure 5, exploits both the current CNN block output, $x_0 \in \mathbb{R}^{H_0 \times W_0 \times C_0}$, and a range of preceding multi-scale feature maps, $X_m = [x_1, x_2, \dots, x_R]$. Where $[x_1, x_2, \dots, x_R]$ refers to the concatenation of the feature-maps, R is the coverage region that delimits how many preceding multi-scale CNN blocks output will be consolidated alongside 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 $x_R \in \mathbb{R}^{H_R \times W_R \times C_R}$.

To consolidate the preceding multi-scale features, Multi-scale Aggregation Block (MAB) is proposed. Moreover, to control how many preceding multi-scale features will be consolidated, a Coverage Region (R) is introduced. The convention for the CNN backbone is that the spatial dimensions are shrunk as we go deep, and the depth is increased. Therefore, two alignments operations are essential, i.e., spatial dimension alignment and channel dimension alignment.

a: COVERAGE REGION (R)

To control the information flow between CNN blocks, we introduce the coverage region R . The lower part in Figure 5 illustrates the layout of the multi-scale connections

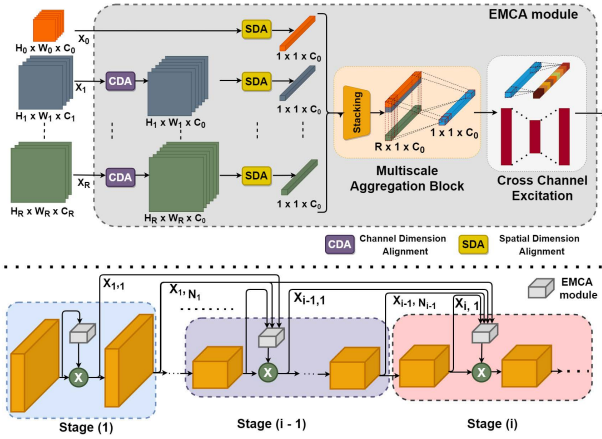


FIGURE 5. Upper part shows the diagram of our Efficient Multi-scale Channel Attention (EMCA) module. Given aggregated features, EMCA generates multi-scale aware channel weights by fast 1D convolution, i.e., Multi-scale Aggregation Block (MAB). The lower part shows our proposed integration method that consolidates multi-scale information based on the Coverage Region (R), i.e., Equation 5.

for an arbitrary CNN backbone, where it consists of stages and blocks; we follow the definition of the stage in [18] which refers to a group of convolutions with an identical spatial dimension. Where connection $X_{i,k}$ indicates the CNN output of block k in the i^{th} stage. For example, the connection $X_{i-2,1}$ indicates the CNN output of the first block in stage $i-2$, and the connection $X_{i-2,N_{i-2}}$ indicates the CNN output of the last block in stage $i-2$, where N_{i-2} represents the number of the blocks in stage $i-2$. Consequently, the i^{th} channel attention block C_i receives R_i feature-maps from the preceding CNN blocks, thus in general:

$$R_i = 1 + \sum_{j=1}^S N'_{i-j}, \quad (5)$$

where the one indicates the associated connection $X_{i,1}$, S indicates the number of the stages that should be considered, and N'_{i-j} indicates the number of the blocks' output in the stage $i-j$ utilized into the channel attention block C_i . Consequently, in case we consolidate the whole preceding multi-scales connection, where $N'_{i-j} = N_{i-j}$ and $S = i-1$, the R_i will hit the upper bound; $R_i^{max} = N_{i-1} + N_{i-2} + \dots + N_1 + 1$.

b: CHANNEL DIMENSION ALIGNMENT (CDA)

In general, the earlier multi-scale features X_m have different channel dimensions, as the convention is as we go deeper in the network, the depth is increased. Therefore, the first operation in our EMCA module 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 by learnable upsampling techniques or a simple repeating operation to align with the channel dimension of the current CNN block C_0 . Consequently, channel-aligned feature maps are produced, $x'_i \in \mathbb{R}^{H_i \times W_i \times C_0}$.

c: SPATIAL DIMENSION ALIGNMENT (SDA)

Analogous to aligning the channel dimensions, the spatial dimensions; H and W , are aligned through squeeze

operation by adopting the general global average pooling equation as follows, $\tilde{x}_i = \frac{1}{WH} \sum_{j=1}^W \sum_{k=1}^H x'_i(j, k)$, where $\tilde{x}_i \in \mathbb{R}^{1 \times 1 \times C_0}$ and represents the squeezed feature maps from the channel aligned aggregated feature maps x'_i , where $i = 0, 1, \dots, R-1$.

d: MULTI-SCALE AGGREGATION BLOCK (MAB)

Algorithm 1 EMCA Module Algorithm

```

Input:  $x_0$ : Current feature map.
           $X_m$ : List of preceding feature maps.
Output:  $Z$ : Learned channel scales
1:  $R \leftarrow \text{Length}(Y) + 1$ 
2: for  $r$  in range( $R$ ) do
3:    $x'[r] \leftarrow \text{CDA}(X_m[r])$ 
4:    $\tilde{x}[r] \leftarrow \text{SDA}(x'[r])$ 
5: end for
6:  $S \leftarrow \text{Stack}(x_0, \tilde{x})$ 
7:  $X_a \leftarrow \text{ReLU}(1D\text{Conv}(S, \text{Kernel} = R))$ 
8:  $Z \leftarrow \sigma(W_{m_2}(X_a))$ 
9: return  $Z$ 

```

Since our EMCA module aims at appropriately fusing local and global cues, various possibilities are discussed to settle down on the best fusion mechanism. By global cues, we mean the aggregated multi-scale features. A general form of our proposed MAB can be seen as $X_a = \nu(W_m(\tilde{X}))$, where $X_a \in \mathbb{R}^{1 \times 1 \times C_0}$ is the aggregated multi-scale features, ν represents a non-linear activation function, W_m is the learnable weights associated with our MAB, and $\tilde{X} = [x_0, \tilde{x}_1, \dots, \tilde{x}_{R-1}]$. Where $[x_0, \tilde{x}_1, \dots, \tilde{x}_{R-1}]$ refers to the concatenation operation of the preceding aligned multi-scale feature maps \tilde{x}_i with the current CNN block output x_0 . To fully capture the multi-scale interactions in conjunction with cross-channel interactions, W_m can be interpreted as a fully connected layer where $W_m \in \mathbb{R}^{RC_0 \times RC_0}$. In contrast, to learn the multi-scale interactions and channel interactions with neglecting the cross-channel relations, W_m can be interpreted as a depth-wise separable convolution layer, where $W_m \in \mathbb{R}^{1 \times RC_0}$. Consequently, both approaches involve a tremendous number of parameters. Thus a possible compromise can be achieved if we split W_m into two sub-functions. The first function, $W_{m_1} \in \mathbb{R}^{RC_0 \times RC_0}$, will capture the multi-scale interactions, which can be readily interpreted as a 1-D convolution layer with kernel size equals R . The second function W_{m_2} , fully captures channel-wise dependencies adopting one of the on-the-shelf local channel attention techniques that are discussed in Section II. Consequently, the final form of our EMCA module is $Z = \sigma(W_{m_2}(\nu(W_{m_1}(\tilde{X}))))$, where Z is the learned scales that represent the importance of each channel from the input feature map x_0 , and σ is the sigmoid activation function.

Algorithm 1 combines the blocks mentioned above and demonstrates a pseudo-code for our EMCA module.

IV. EXPERIMENTS

This section performs controlled ablation experiments to settle on the best internal design for our proposed module and assess its sub-modules. Then we evaluate the performance of the proposed Multi-Scale Attention module on a series of benchmark datasets across different tasks include classification, detection, and segmentation. To assess our EMCA module on the classification task, Tiny-ImageNet [59], and ImageNet [60] are used. While for the detection task MS-COCO [61] and KITTI [62], are used. Also, we benchmark our proposed module on instance segmentation task using MS-COCO [61]. Finally, We conduct empirical experiments that probe the robustness of the representations learned by EMCA compared to other attention mechanisms.

A. DATASETS

In this section, we will cover the details of the used datasets while evaluating our module (EMCA). We have evaluated our module on well-known set of benchmarks the covers wide range of applications ranging from image classification ending with downstreams tasks like object detection and segmentation.

- **ImageNet.** ImageNet is an image database categorized according to the WordNet hierarchy (currently only the nouns), in which thousands of images describe each node of the hierarchy. The project has been instrumental in promoting computer vision and deep learning research. The data was made publicly available for free to researchers for non-commercial use. ImageNet offers many variants of the dataset. Each version is labeled by the year of publication. For instance, ImageNet 2012 was published in 2012, the most commonly used while reporting the accuracy. When ImageNet was firstly released, it aimed to enhance and reinforce the research progress on computer vision tasks. What makes the data very challenging is containing a vast number of categories, i.e., 1000 classes. The data is not perfectly balanced, where each class has a different number of images ranging from tens to hundreds.
- **Tiny-ImageNet.** Tiny ImageNet is a subset of the ImageNet dataset in the famous ImageNet Large Scale Visual Recognition Challenge (ILSVRC). When ImageNet was firstly released, it aimed to enhance and reinforce the research progress on computer vision tasks. However, it is not straightforward to download and store it due to its vast size. Even the more complicated part is to train using the entire dataset. Therefore many researchers create their own mini-version of the data to be able to train on this newly created sub-set of the data. The disadvantage of this technique is that other researchers have to reproduce the same subset of the data to be able to benchmark it. This motivates the computer vision community to offer a well-defined subset of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) The dataset dubbed Tiny-ImageNet contains

100,000 images of 200 classes (500 for each class) downsized to 64×64 colored images. Each class has 500 training images, 50 validation images, and 50 test images.

- **KITTI.** KITTI develops real-world challenges in the computer vision domain by proposing many benchmarks covering a wide range of real applications, such as stereo-based detection and tracking, optical flow estimation, visual odometry prediction, 3D object detection, segmentation, and 3D tracking. For this purpose, they equipped the car with two colorful high-resolution cameras mounted on the roof of the vehicle to provide us with a stereo vision. In addition, they provide an accurate 3D representation of the scene using a lidar sensor. The dataset is captured by driving around a mid-size city in rural areas and on highways. Therefore the scenes captured can be considered crowded scenes where each scene contains up to 15 cars and 30 pedestrians. They also provide an evaluation metric for each benchmark. KITTI-RGB [62] 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 in the image classification section.
- **MS-COCO.** COCO has been focused on advancing computer vision and deep learning research progress in general and explicitly advancing the state-of-the-art in object detection and recognition tasks. The data is built by collecting images from regular daily activities and is categorized into 91 object types. The total number of labeled items is 2.5 million for almost 328k images. COCO provides a wide range of annotations for different tasks, i.e., object detection and recognition and semantic and instance segmentation.

B. IMPLEMENTATION DETAILS

For the classification task, two datasets are used, i.e., Tiny-ImageNet dataset [59] and ImageNet dataset [60], to evaluate our proposed module and show its effectiveness, where the same data augmentation and hyper-parameter settings in [18] are adopted. For the Tiny-ImageNet dataset [59], input images are randomly cropped to 64×64 with random horizontal flipping. For the ImageNet dataset [60], we adopt the same training setup as [3], [18], where a 224×224 crop is randomly sampled from an image or its horizontal flip, with the per-pixel RGB mean value subtracted. All models are trained for 100 epochs from scratch, using the weight initialization strategy described in [63], 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 10^{-4} , the momentum of 0.9, and mini-batch size of 32 are used for Tiny-ImageNet [59], and 256 for ImageNet [60]. Our module is implemented in Python using the PyTorch framework using four PCs with Intel Xeon(R) 4108 1.8GHz CPU, 64G RAM, Nvidia Titan-XP.

For detection and instance segmentation tasks, we evaluate our EMCA module on MS-COCO [61] using Faster R-CNN [64], Mask R-CNN [65], and RetinaNet [66], where ResNet-50 along with FPN [67] are used as a backbone. We implement all detectors based on the MMDetection toolkit [68] and employ the default settings as described in [20]. Specifically, We resize the shorter side of input images to 800. Then The learning rate is initialized to 0.01 and decreased by 10 after 8 and 11 epochs, respectively. All models are optimized using SGD with a weight decay of $1e-4$, momentum of 0.9, and mini-batch size of 8. Finally, we train all detectors within 12 epochs on train2017 of COCO and report the results on val2017.

Furthermore, we evaluate our proposed EMCA module on object detection task using KITTI dataset [62] on a modified version of the YOLO-V3 detector, explicitly tailored to integrate the different channel attention modules to it, including ours, EMCA. The backbone is replaced by the modified version of the ResNet backbone, where the different channel attention modules are integrated into it. We followed the original setup proposed by YOLO-V2 and YOLO-V3. The input images are resized into colored square images with the following shape, $448*448$. The stochastic gradient descent with a starting learning rate of 0.1, polynomial rate decay with a power of 4, weight decay of 0.0005, and momentum of 0.9 is used. The standard data augmentation techniques are followed, including random crops, rotations, hue, saturation, and exposure shifts. Also, no hard negative mining is used. Multi-scale training is followed, where the network first is trained on a smaller resolution; $224*224$, then trained on the final resolution; $448*448$, where these tricks are proposed by the original YOLO detectors that showed a significant impact.

C. EFFECT OF COVERAGE REGION (R)

To assess the proposed coverage region (R) and provide a clear picture of its role, we experiment with four coverage region variants based on three ResNet family variants, i.e., R-18, R-34, and R50, on the ImageNet dataset. Specifically, we vary the S and N'_{i-j} parameters in Equation 5 to demonstrate the coverage region effect in terms of inference speed for the model by inferring one image at a time (FPS), network parameters (#P in millions), and Top-1 accuracy (in %). The first row in Table 3 refers to the original channel attention module [18], [19], [20]. In the second row, S and N'_{i-j} parameters are set to zeros, which refers to the simplest form of our EMCA module that avoids the dense integration as discussed in Section III-B1, where there is no multi-scale information is propagated from the preceding CNN blocks. The third row demonstrates the results of setting S and N'_{i-j} parameters to ones, which means the current channel module C_i will aggregate the multi-scale information from the last block at the preceding stage only. While the fourth row shows the results of setting S and N'_{i-j} parameters to one and N_{i-j} , respectively, which means the current channel module, C_i , will aggregate the multi-scale information from the whole blocks at the preceding stage only. Finally, the last row shows

TABLE 3. Effect of Coverage Region (R). Comparison between different values of S and N'_{i-j} parameters in Equation 5. Setting both parameters to zero refers to the original channel attention module without incorporating multi-scale information.

S,	N'_{i-j}		SE			ECA			SRM		
			FPS	#P(M)	Top-1	FPS	#P(M)	Top-1	FPS	#P (M)	Top-1
N/A,	N/A		187	11.23	70.59	192	11.14	70.75	154	11.152	70.96
0,	0		204	11.18	70.91	212	11.14	70.63	165	11.150	71.31
1,	1	R-18	156	11.18	71.02	174	11.14	70.83	123	11.150	71.20
1,	N_{i-j}		160	11.19	71.00	170	11.14	71.04	113	11.150	71.02
i-1,	1		153	11.19	71.02	169	11.14	70.59	113	11.150	71.00
N/A,	N/A		101	20.93	73.87	107	20.78	74.13	82	20.795	73.98
0,	0		122	20.82	73.84	122	20.78	74.20	96	20.790	74.51
1,	1	R-34	109	20.82	74.33	109	20.78	74.39	82	20.790	74.39
1,	N_{i-j}		107	20.82	74.40	107	20.78	74.46	81	20.790	74.38
i-1,	1		103	20.82	74.02	108	20.78	74.14	80	20.790	74.57
N/A,	N/A		90	26.77	76.80	87	24.37	77.12	71	24.402	77.13
0,	0		97	25.03	76.56	98	24.37	77.02	81	24.380	76.98
1,	1	R-50	88	25.03	77.10	94	24.37	76.98	70	24.380	77.00
1,	N_{i-j}		90	25.03	77.33	92	24.37	77.13	70	24.380	77.20
i-1,	1		89	25.03	76.85	91	24.37	76.82	71	24.380	77.05

the results of setting S and N'_{i-j} parameters to $i - 1$ and 1, respectively, which means the current channel module C_i will aggregate the multi-scale information from the last block only from the whole preceding stages. To keep the module compactness and efficiency, we did not study the extreme case, where the current channel module C_i will aggregate the multi-scale information from the whole blocks from the whole preceding stages; $S = i - 1$ and $N'_{i-j} = N_{i-j}$.

As shown in Table 3, our four proposed variants achieve better performance than the original CA-modules in terms of memory usage, inference speed, and accuracy. Based on the results above, S and N'_{i-j} in Equation 5 is set to one and N_{i-j} , respectively, in the rest of our experiments.

D. IMAGE CLASSIFICATION

We evaluate the performance of the proposed EMCA module on classification benchmark datasets include Tiny-ImageNet [59], which is mentioned in the supplementary materials, and ImageNet [60]. ImageNet LSVRC 2012 dataset [60] contains 10^3 classes with 1.2 million training images, 50×10^3 validation images, and 10^5 test images. The evaluation is measured on the non-blacklist images of the ImageNet LSVRC 2012 validation set.

All the classification experiments follow the same training procedure that is discussed in Section IV-B. However, not all attention methods followed the same training and testing procedure where: 1) SRM [19] is trained for 90 epochs only. 2) FCANet [53] is trained for 100 epochs with cosine learning rate decay [72] and label-smoothing regularization [5] with the coefficient value as 0.1 during training. 3) SANet [69] starts from the initial learning rate of 0.1 with a linear warm-up [73] of 5 epochs and follows a different method to initialize the parameters. 4) EPSANet [52] uses label-smoothing regularization [5], where the coefficient value is set to 0.1 during training and is trained for 120 epochs instead of 100 epochs. 5) ECA [20] follows the same experimental setup; however, we notice difficulties reproducing and verifying their results.¹

¹Referring to issues number 21, 52, 62, 24, 46, and 58 from the official ECA-Net implementation.

TABLE 4. Comparison between our three proposed versions of EMCA module that adopt SE, ECA, and SRM as the cross channel excitation block producing, EMCA-SE, EMCA-ECA, and EMCA-SRM, respectively. The comparison is built on the ImageNet dataset and covers the following aspects; network parameters (#.P in millions), floating-point operations per second in Giga (GFLOPs), Top-1 and Top-5 accuracy (in %), inference speed for the model by inferring one image at a time (FPS), inference speed for the model by inferring batch of images at a time (FPS[†]) and inference speed for the whole validation process by inferring batch of images at a time (FPS^{††}). The * symbol indicates we retrain the model for fair comparison. Top-1 relative improvement results are reported between parentheses w.r.t the corresponding CA-module improvement over the Vanilla Resnet.

Methods	#.P (M)	GFLOPs	Top-1(σ) (RI)	Top-5	FPS	FPS [†]	FPS ^{††}
ResNet [3]	11.148	1.694	70.40	89.45	270	23552	859
+SENet [18]	11.231	1.695	70.59	89.78	187	21760	839
+EMCA-SE	11.190	1.695	71.00\pm0.09(315)	90.00	160	17313	813
+ECANet [20]	11.148	1.695	70.78	89.92	192	22287	848
+ECANet* [20]	11.148	1.695	70.75	89.74	192	22287	848
+EMCA-ECA	11.148	1.695	71.04\pm0.07(182)	89.99	170	19023	833
+SRM* [19]	11.152	1.695	70.96	89.81	154	18794	823
+EMCA-SRM	11.150	1.694	71.32\pm0.05(164)	90.00	113	15190	803
ResNet [3]	20.788	3.419	73.31	91.40	168	19712	840
+SENet [18]	20.938	3.421	73.87	91.65	101	14279	805
+EMCA-SE	20.829	3.421	74.41\pm0.06(196)	91.90	107	14372	812
+ECANet [20]	20.788	3.420	74.21	91.83	107	14067	825
+ECANet* [20]	20.788	3.420	74.13	91.68	107	14067	825
+EMCA-ECA	20.788	3.421	74.46\pm0.05(140)	91.70	107	14080	822
+SRM* [19]	20.795	3.419	73.98	91.68	82	12655	803
+EMCA-SRM	20.790	3.419	74.38\pm0.06(159)	91.87	81	12579	795
ResNet [3]	24.373	3.829	75.89	92.85	124	10032	668
+SENet [18]	26.772	3.837	76.80	93.39	90	8156	597
+EMCA-SE	25.037	3.835	77.33\pm0.06(158)	93.52	90	8099	589
+ECANet [20]	24.373	3.834	77.48	93.68	87	8517	591
+ECANet* [20]	24.373	3.834	77.12	93.68	87	8517	591
+EMCA-ECA	24.373	3.834	77.64\pm0.09(142)	93.49	92	8615	600
+SRM* [19]	24.402	3.829	77.13	93.51	71	6745	536
+EMCA-SRM	24.380	3.829	77.70\pm0.10(146)	93.54	70	6698	532

The retrained models are marked with the * symbol and report the results of other compared methods from their original papers. Our evaluation metrics incorporate both efficiency and effectiveness. The efficiency is measured by the network parameters (#.P) in millions, the inference frame rate per second (FPS), and the floating-point operations per second (FLOPs) in Giga. The effectiveness is measured by the Top-1 and Top-5 accuracies.

Firstly, we adopt three well-known channel attention modules, i.e., SENet, ECANet, SRMNet, and incorporate them as the cross channel excitation block, as shown in Figure 5. Thus, three versions of our EMCA module are produced, i.e., EMCA-SE, EMCA-ECA, and EMCA-SRM, where SENet, ECANet, and SRMNet are regarded as the cross channel excitation block respectively. Then, in Table 4, we compare the three proposed variants of our EMCA module against their original CA-modules. The results are given in Table 4 show that our EMCA module has fewer model complexity (i.e., network parameters, GFLOPs, and inference speed) than the respective original CA-modules while achieving better accuracy across different ResNet sizes and based on various CA-modules. Top-1 relative improvement (RI) results in percentage are reported between parentheses in red w.r.t the corresponding CA-module improvement over Vanilla Resnet.

Then, we compare our EMCA module with several state-of-the-art attention methods using the ResNet family. As shown in Table 5, our proposed EMCA module reduces

TABLE 5. Comparison of different attention methods on ImageNet in terms of network parameters (#.P in millions), floating-point operations per second in Giga (GFLOPs), Top-1 and Top-5 accuracy (in %), inference speed for the model by inferring one image at a time (FPS), inference speed for the model by inferring batch of images at a time (FPS[†]) and inference speed for the whole validation process by inferring batch of images at a time (FPS^{††}). The * symbol indicates we retrain the model as it is trained initially with different training settings.

Methods	#.P (M)	GFLOPs	Top-1	Top-5	FPS	FPS [†]	FPS ^{††}
ResNet [3]	11.148	1.694	70.40	89.45	270	23552	859
SENet [18]	11.231	1.695	70.59	89.78	187	21760	839
ECANet* [20]	11.148	1.695	70.75	89.74	192	22287	839
SRM* [19]	11.152	1.694	70.96	89.81	154	18794	823
FCANet* [53]	11.231	1.694	70.98	90.00	119	17680	808
BAM [21]	11.712	1.821	75.98	92.82	91	7159	527
CBAM [22]	11.234	1.695	70.73	89.91	104	8734	789
EMCA-ECA	11.148	1.695	71.04	89.99	170	19023	833
EMCA-SRM	11.150	1.694	71.32	90.00	113	15190	803
EMCA-SE	11.190	1.695	71.00	90.00	160	17313	813
ResNet [3]	20.788	3.419	73.31	91.4	168	19712	840
SENet [18]	20.938	3.421	73.87	91.65	101	14279	805
ECANet* [20]	20.788	3.420	74.13	91.68	107	14067	825
SRM* [19]	20.795	3.419	73.98	91.68	82	12655	803
FCANet* [53]	20.938	3.419	74.18	91.75	87	13094	812
CBAM [22]	20.943	3.420	74.01	91.76	59	12001	760
EMCA-ECA	20.788	3.421	74.46	91.70	107	14080	822
EMCA-SRM	20.790	3.419	74.38	91.87	81	12579	795
EMCA-SE	20.829	3.421	74.41	91.90	107	14372	812
ResNet [3]	24.373	3.829	75.89	92.85	124	10032	668
SENet [18]	26.772	3.837	76.80	93.39	90	8156	597
ECANet* [20]	24.373	3.834	77.12	93.68	87	8517	591
SRM* [19]	24.402	3.829	77.13	93.51	71	6745	536
FCANet* [53]	26.772	3.831	77.27	93.70	74	7984	549
EPANet* [52]	21.517	3.373	77.31	93.72	28	802	388
SA ² Net* [69]	24.373	3.832	77.25	93.66	68	6670	406
A ² Net [70]	33.006	6.502	77.00	93.50	N/A	N/A	N/A
ABN [71]	43.594	7.183	76.90	N/A	N/A	N/A	N/A
BAM [21]	25.92	3.946	75.98	92.82	91	7159	527
CBAM [22]	26.775	3.837	77.34	93.69	55	2460	208
EMCA-ECA	24.373	3.834	77.64	93.49	92	8615	600
EMCA-SRM	24.380	3.829	77.70	93.54	71	6698	532
EMCA-SE	25.037	3.835	77.33	93.52	90	8099	589

computation cost and memory usage of these networks and benefits inference speed and accuracy. To measure the inference speed, three methods are reported in Table 4 and Table 5 as follows: 1) Inferring one image at a time to the model (FPS), to this end, we use only one worker at the data loader and set the model's batch size to one. 2) To cope with the reported results at [20], we measure the inference speed while inferring a batch of images at a time, where we set the model's batch size to 256. We refer to this as FPS[†] 3) Inference speed for the whole validation process is calculated (FPS^{††}) by inferring a batch of images at a time, where we set the model's batch size to 256 and use eight workers while loading the data. We believe the first method (FPS) is the most accurate to show the actual effect of adding an attention module to the vanilla ResNet. However, we reported the other measures to cope with other results reported in previous work.

E. OBJECT DETECTION

1) MS-COCO

We evaluate our EMCA module on object detection task using Faster R-CNN [64], Mask R-CNN [65], and RetinaNet [66] on MS COCO dataset [61]. We implement our EMCA module using the MM-Detection framework [68]. All CNN models are pre-trained on ImageNet, and their results are reported in Table 6 from their original papers except for ECANet [20], where it is pre-trained using the weights produced by us after retraining on ImageNet, as mentioned in Section IV-D. As shown in Table 6, integration of our EMCA module based

TABLE 6. Object detection results of different attention methods on COCO val2017.

Methods	Detectors	#.P (M)	GFLOPs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-50 [3]		41.53	207.07	36.4	58.2	39.2	21.8	40.0	46.2
+SE [18]		44.02	207.18	37.7	60.1	40.9	22.9	41.9	48.2
+EMCA+SE	Faster	42.56	207.18	38.1	60.6	50.2	23.6	42.2	48.4
+ECA [20]	R-CNN	41.53	207.18	38.0	60.6	40.9	23.4	42.1	48.0
+EMCA+ECA		41.53	207.18	38.2	60.9	50.0	23.7	42.2	48.2
ResNet-50 [3]		44.18	275.58	37.2	58.9	40.3	22.2	40.7	48.0
+1 NL [12]		46.50	288.70	38.0	59.8	41.0	N/A	N/A	N/A
+GC [11]		46.90	279.60	39.4	61.6	42.4	N/A	N/A	N/A
+SE [18]	Mask	46.67	275.69	38.7	60.9	42.1	23.4	42.7	50.0
+EMCA+SE	R-CNN	45.13	275.69	39.0	61.4	42.3	23.7	42.9	50.1
+ECA [20]		44.18	275.69	39.0	61.3	42.1	24.2	42.8	49.9
+EMCA+ECA		44.18	275.69	39.1	61.5	42.1	24.4	42.9	49.9
ResNet-50 [3]		37.74	239.32	35.6	55.5	38.2	20.0	39.6	46.8
+SE [18]		40.23	239.43	37.1	57.2	39.9	21.2	40.7	49.3
+EMCA+SE	RetinaNet	38.88	239.43	37.2	57.4	39.9	21.2	40.7	49.3
+ECA [20]		37.74	239.43	37.3	57.7	39.6	21.9	41.3	48.9
+EMCA+ECA		37.74	239.43	37.3	57.8	39.6	21.9	41.3	48.9

TABLE 7. Comparisons with state-of-the-art attention modules on KITTI-RGB [62] in terms of mAP.

	ResNet	SE	ECA	CBAM	BAM	SRM	EMCA
Resnet-18	57.87	59.32	58.55	57.90	59.61	59.20	59.66
Resnet-50	64.19	65.08	64.34	64.18	65.10	64.82	65.21

on either SE or ECA modules can improve the performance of downstream tasks like object detection.

2) KITTI

Furthermore, we evaluate our proposed EMCA module on object detection task using KITTI dataset [62]. We adapt YOLOV3 [74] detector by replacing its original DarkNet backbone with the different channel attention networks that are built based on the ResNet backbone family. KITTI-RGB [62] 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 in the image classification section. As shown in Table 7, EMCA considerably improves the accuracy more than other attention modules compared to the baseline [3]. EMCA-SE variant is used in these experiments as it achieves the best performance on the ImageNet dataset.

F. INSTANCE SEGMENTATION

To prove the effectiveness of our EMCA module, we assess it using another downstream task, i.e., instance segmentation using Mask R-CNN [65] on the MS COCO dataset [61]. We implement our EMCA module using the MM-Detection framework [68]. The ResNet-50 variant is used as a backbone. All models are optimized using SGD with a weight decay of $1e-4$, momentum of 0.9, and mini-batch size of 8 and trained for 12 epochs. Where, the learning rate is initialized to 0.01 and decreased by 10 after 8 and 11 epochs, respectively. The train2017 and val2017 splits are used for the training and the evaluation, respectively.

As compared to Table 8, our EMCA module achieves better performance than the original ResNet, SE, and ECA modules.

TABLE 8. Instance segmentation results of different methods using Mask R-CNN on COCO val2017.

Methods	#.P (M)	GFLOPs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-50 [3]	44.18	275.58	34.1	55.5	36.2	16.1	36.7	50.0
+SE [18]	46.67	275.69	35.4	57.4	37.8	17.1	38.6	51.8
+EMCA+SE	45.13	275.69	35.7	58.1	38.0	17.8	39.0	51.9
+ECA [20]	44.18	275.69	35.6	58.1	37.7	17.6	39.0	51.8
+EMCA+ECA	44.18	275.69	35.7	58.4	37.7	17.9	39.1	51.9

TABLE 9. Analyzing the robustness of CA-modules on ImageNet.

	0°				90°				180°				270°			
	ResNet-18								ResNet-50							
ResNet	70.40	41.25	41.95	41.14	75.89	46.71	47.09	46.64	77.13	48.12	49.01	48.19	77.20	48.08	49.11	48.21
+ EMCA-ECA	71.04	43.14	44.20	43.26	77.13	48.12	49.01	48.19	77.20	48.08	49.11	48.21	77.20	48.08	49.11	48.21
+ EMCA-SRM	71.02	43.18	44.29	43.15	77.20	48.08	49.11	48.21	77.20	48.08	49.11	48.21	77.20	48.08	49.11	48.21
+ EMCA-SE	71.00	43.21	44.26	43.22	77.33	48.15	49.20	48.19	77.33	48.15	49.20	48.19	77.33	48.15	49.20	48.19

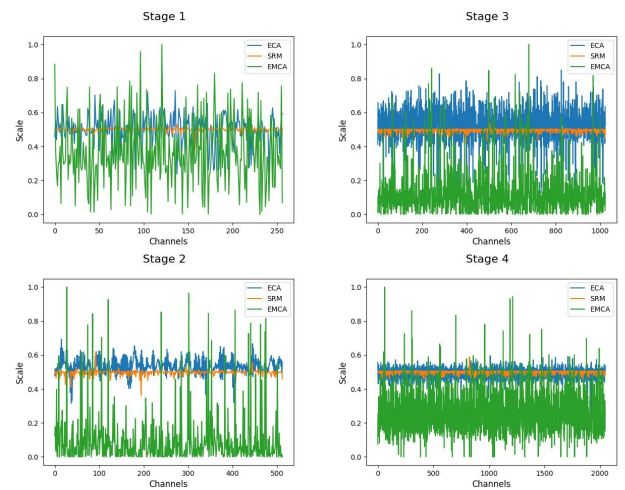


FIGURE 6. Comparison for the learned channel scales by our novel EMCA module against local channel attention modules. Better view with zooming in.

G. ROBUSTNESS

We have conducted experiments to probe the robustness of the representations learned by our proposed module EMCA, compared to other channel attention mechanisms on the ImageNet dataset, by rotating the testing images deliberately in one of three ways: clockwise 90° , clockwise 180° , clockwise 270° . These rotations were not scrutinized at the training. As shown in Table 9, our EMCA module is less vulnerable than other attention modules.

H. DISSECTING THE PRODUCED LEARNABLE SCALES

To further analyze the effect of our EMCA module on learning channel attention, we visualize the scales learned by our novel EMCA modules and compare it against local channel attention modules; ECA and SRM. As discussed in Section III-A, avoiding dense integration is a key factor in boosting the attention mechanisms performance. Thus we have integrated our EMCA module into the first CNN block only for each stage. Consequently, for fair comparison, we have compared the learnable scales produced by our

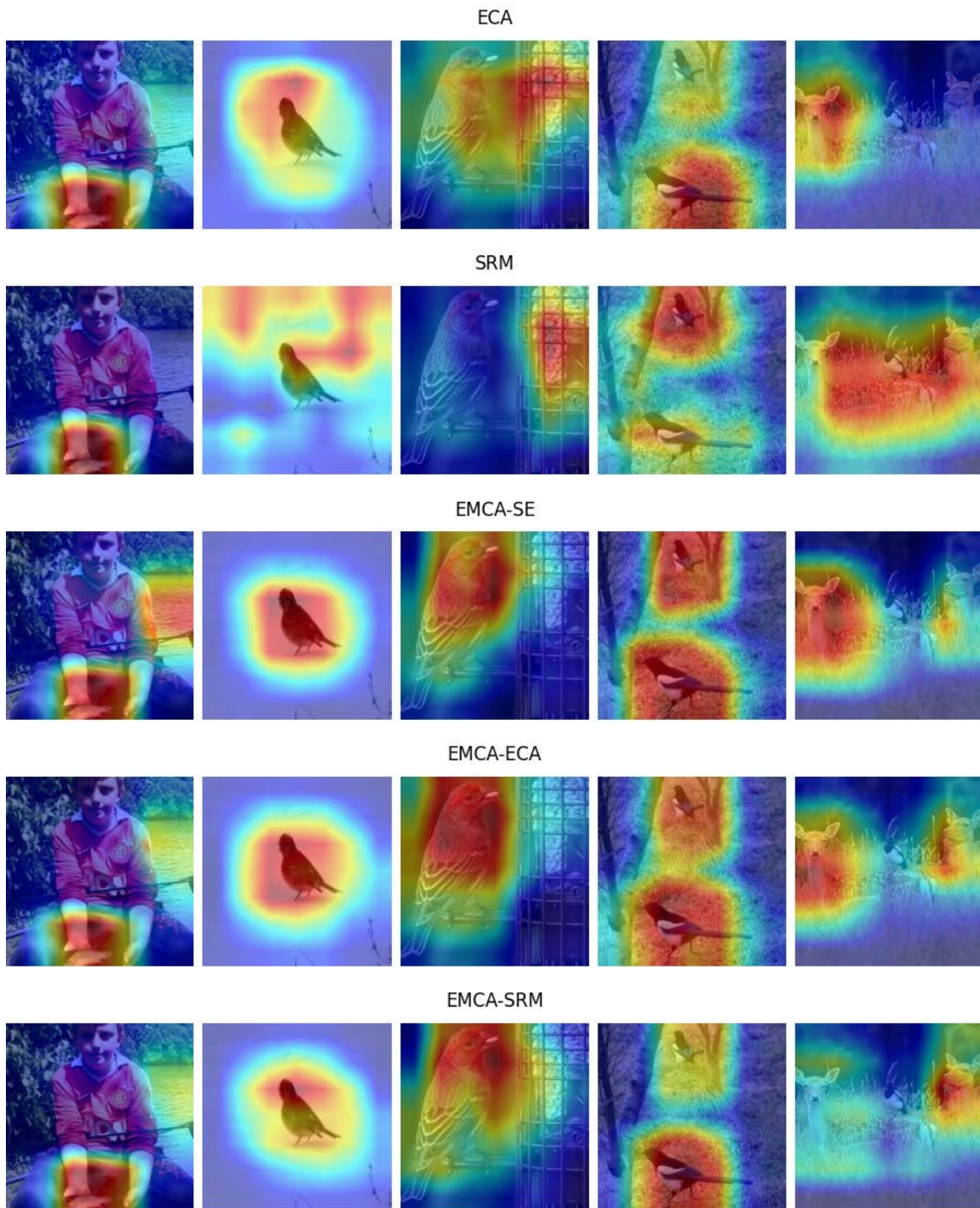


FIGURE 7. Sample visualization on ImageNet dataset [60] generated by GradCAM [75].

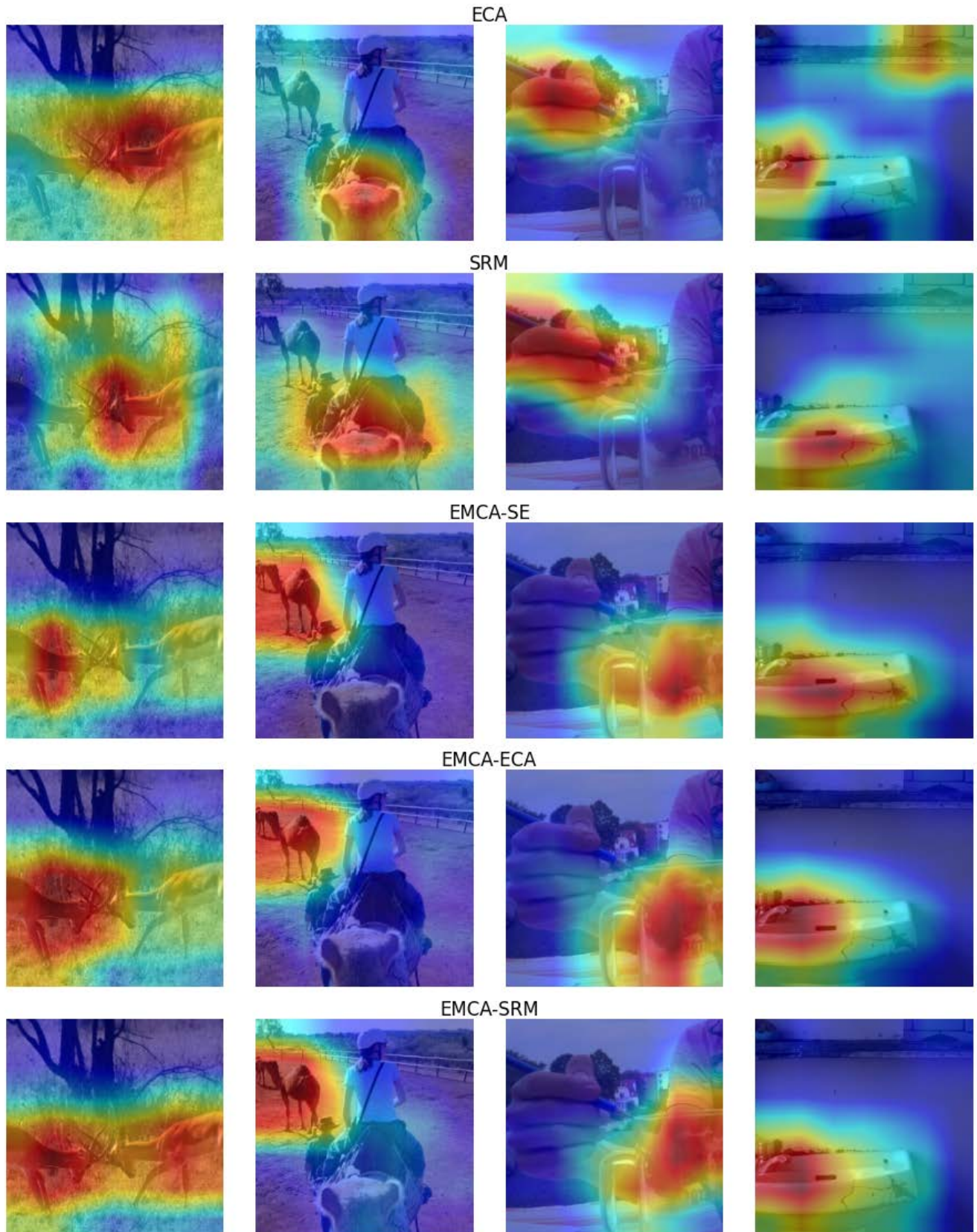


FIGURE 8. Sample visualization on ImageNet dataset [60] generated by GradCAM [75].

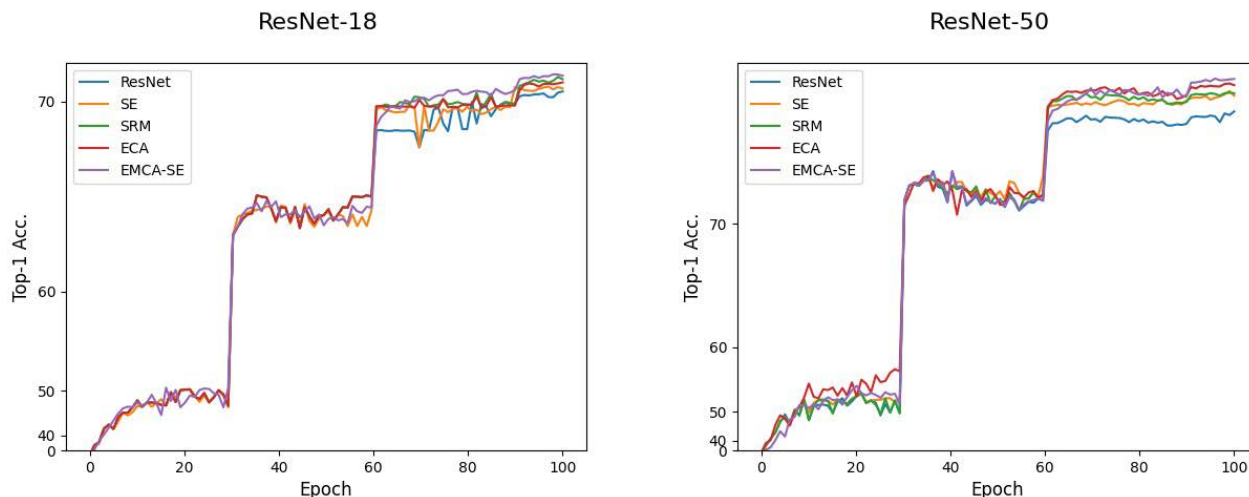


FIGURE 9. Training ResNet, local channel attention modules (LCA) baseline architectures and their EMCA counterparts on ImageNet validation set. EMCA exhibits improved optimization characteristics and produces consistent gains in performance which are sustained throughout the training process.

EMCA module with only the scales produced from the first attention module at each stage. For this experiment, we adopt ResNet-50 as backbone and use ImageNet validation set.

In contrast to ECA setup, where a random sample consists of four classes only from ImageNet dataset, i.e., hammerhead shark, ambulance, medicine chest and butternut squash, are involved while producing the scales, we have used a more generic and fair way to analyze the learned scales by averaging them over the whole validation dataset instead of using only four selected classes. Figure 6 visualizes the channel learned scales for each first block from each stage for each attention module; ECA in blue, SRM in orange, and ECA in orange.

Driven by the results in Figure 6, we make some observations about the role of our EMCA module:

- EMCA scales have larger variance than ECA and SRM learned scales, which indicates a better discriminative ability which necessarily reflect the quality of the learned scales.
- ECA and SRM learned scales have a mean around 0.5 which indicates almost the majority of the channels have the same importance. In contrast, the EMCA scales mean is around much lower value than 0.5, i.e., 0.1 and 0.3 for stage 2,3 and stage 1,4 respectively.
- The majority of EMCA scales are below 0.5, which is very beneficial in compressing the network, by pruning the channels that have importance factor less than a certain threshold, e.g. 0.2. In addition, 1.5 % of the produced scales by EMCA are zeros, while ECA and SRM produce non-zero scales. Thus by pruning the channels that have zero importance; scale, we achieve a further gain in the performance in terms of GFLOPs, FPS, and number of parameters while achieving the same accuracy.
- Finally, our learnable scales show empirically that they are more representative as they boost the accuracy in a

consistent manner over different architectures and different tasks.

As a future work driven by the earlier observations and analysis, more experiments are needed to validate EMCA contribution regarding pruning the network more efficiently.

I. VISUALIZING ATTENTION MAPS

In order to validate the effectiveness of EMCA module more intuitively, we sample nine images from ImageNet dataset [60] validation split. We use Grad-CAM [75] to visualize their heatmaps at the last attention module based on ResNet50 backbone. As shown in Figure 7 and Figure 8, our proposed EMCA module allows the classification model to focus on more relevant regions with more object details, which means the EMCA module can effectively improve the classification accuracy. Also, it is obvious that our EMCA module can handle more than one object in the scene at the same time thanks to the multi-scale aggregation module. For instance, as shown in Figure 7, the last column depicts an image with two deer, where ECA and SRM, first two rows, are focusing only on one of the deer. In contrast, the three variants of our proposed module EMCA are successfully paying more attention to the two deer. Therefore, the proposed EMCA module is validated to indeed enhance the representation power of networks.

J. TOP-1 VALIDATION ACCURACY CURVES

To provide some insight into the influence of our EMCA module on the optimization process of these models during the training phase, we tracked the validation Top-1 accuracy during the training. As shown in Figure 9, we compare the training curves, where each epoch's validation accuracy is reported. In this analysis, our EMCA module is compared against the naive ResNet [3] and the local channel attention modules, i.e., SE [18], SRM [19], and ECA [20]. Driven by this analysis, we observe that our EMCA module yields a

steady improvement throughout the optimization procedure. Moreover, this trend is relatively consistent across a range of network architectures considered as baselines.

V. CONCLUSION

In this paper, we concentrate on determining an effective channel attention module with low model complexity. To this end, we propose Efficient Multi-scale Channel Attention Module (EMCA). As discussed in Section III-B2, our EMCA module preserves the advantages of feature reuse characteristics thanks to the proposed coverage region (R), but at the same time prevents an excessive amount of duplicate gradient information by truncating the gradient flow, Section III-A. Because of the lightweight computation of the EMCA module, it can be integrated into all modern CNN architectures across all layers and trained end-to-end. While most previous works utilized uni-scale features, EMCA is designed to employ multi-scale information while re-calibrating feature maps. Our experiments demonstrate that simply inserting EMCA into standard CNN architectures boosts the performance across different tasks. Furthermore, we verify the robustness of the representations learned by EMCA and its generalization ability via zero-shot experiments to rotated images.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25. Stateline, NV, USA, Dec. 2012, pp. 1097–1105.
- [2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
- [4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, 2017, pp. 1–7.
- [5] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jan. 2016, pp. 2818–2826.
- [6] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," 2015, *arXiv:1507.06228*.
- [7] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," 2014, *arXiv:1406.6247*.
- [8] D. M. Beck and S. Kastner, "Top-down and bottom-up mechanisms in biasing competition in the human brain," *Vis. Res.*, vol. 49, no. 10, pp. 1154–1165, 2009.
- [9] R. Desimone, "Visual attention mediated by biased competition in extrastriate visual cortex," *Phil. Trans. Roy. Soc. London. Ser. B, Biol. Sci.*, vol. 353, no. 1373, pp. 1245–1255, 1998.
- [10] R. Desimone and J. Duncan, "Neural mechanisms of selective visual attention," *Ann. Rev. Neurosci.*, vol. 18, no. 1, pp. 193–222, 1995.
- [11] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, Oct. 2019, p. 1971.
- [12] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.
- [13] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 10076–10085.
- [14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [15] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3156–3164.
- [16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer*, 2020, pp. 213–229.
- [17] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," 2018, *arXiv:1810.12348*.
- [18] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
- [19] H. Lee, H.-E. Kim, and H. Nam, "SRM: A style-based recalibration module for convolutional neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2019, pp. 1854–1862.
- [20] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11531–11539, doi: [10.1109/CVPR42600.2020.01155](https://doi.org/10.1109/CVPR42600.2020.01155).
- [21] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "BAM: Bottleneck attention module," 2018, *arXiv:1807.06514*.
- [22] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–19.
- [23] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3146–3154.
- [24] D. Linsley, D. Shiebler, S. Eberhardt, and T. Serre, "Learning what and where to attend," 2018, *arXiv:1805.08819*.
- [25] A. G. Roy, N. Navab, and C. Wachinger, "Recalibrating fully convolutional networks with spatial and channel 'squeeze and excitation' blocks," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 540–549, Feb. 2019.
- [26] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5659–5667.
- [27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
- [28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.
- [29] F. Iandola, M. Moskewicz, S. Karayev, R. Girshick, T. Darrell, and K. Keutzer, "DenseNet: Implementing efficient ConvNet descriptor pyramids," 2014, *arXiv:1404.1869*.
- [30] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3640–3649.
- [31] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3074–3082.
- [32] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer*, 2016, pp. 483–499.
- [33] L. Florack, B. T. H. Romeny, M. Viergever, and J. Koenderink, "The Gaussian scale-space paradigm and the multiscale local jet," *Int. J. Comput. Vis.*, vol. 18, no. 1, pp. 61–75, Apr. 1996.
- [34] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2010.
- [35] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*.
- [36] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.
- [37] G. Lin, C. Shen, A. van den Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3194–3203.
- [38] P. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene parsing," in *Proc. Int. Conf. Mach. Learn. PMLR*, 2014, pp. 82–90.
- [39] J. Dai, K. He, and J. Sun, "Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 1635–1643, 2015.

- [40] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jan. 2015, pp. 447–456.
- [41] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3376–3385.
- [42] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.
- [43] J. Daihong, Z. Sai, D. Lei, and D. Yueming, "Multi-scale generative adversarial network for image super-resolution," *Soft Comput.*, vol. 26, no. 8, pp. 3631–3641, 2022.
- [44] H. Chen, "Gashis-transformer: A multi-scale visual transformer approach for gastric histopathological image detection," *Pattern Recognit.*, vol. 130, Oct. 2022, Art. no. 108827.
- [45] N. Zeng, P. Wu, Z. Wang, H. Li, W. Liu, and X. Liu, "A small-sized object detection oriented multi-scale feature fusion approach with application to defect detection," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.
- [46] Y. Ye, T. Tang, B. Zhu, C. Yang, B. Li, and S. Hao, "A multiscale framework with unsupervised learning for remote sensing image registration," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–15, 2022.
- [47] C.-F. R. Chen, Q. Fan, and R. Panda, "CrossViT: Cross-attention multi-scale vision transformer for image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 357–366.
- [48] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
- [49] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 3024–3033.
- [50] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola, "ResNeSt: Split-attention networks," 2020, *arXiv:2004.08955*.
- [51] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 13713–13722.
- [52] H. Zhang, K. Zu, J. Lu, Y. Zou, and D. Meng, "EPSANet: An efficient pyramid squeeze attention block on convolutional neural network," 2021, *arXiv:2105.14447*.
- [53] Z. Qin, P. Zhang, F. Wu, and X. Li, "FcaNet: Frequency channel attention networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 783–792.
- [54] D. W. Raimundo, A. Ignatov, and R. Timofte, "LAN: Lightweight attention-based network for RAW-to-RGB smartphone image processing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun. 2022, pp. 808–816.
- [55] T. Zhang and X. Zhang, "A full-level context squeeze-and-excitation ROI extractor for SAR ship instance segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.
- [56] X. Shu, J. Yang, R. Yan, and Y. Song, "Expansion-squeeze-excitation fusion network for elderly activity recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5281–5292, Aug. 2022.
- [57] M. Jaderberg, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.
- [58] H. Zhou, L. Qi, H. Huang, X. Yang, Z. Wan, and X. Wen, "CANet: Co-attention network for RGB-D semantic segmentation," *Pattern Recognit.*, vol. 124, Apr. 2022, Art. no. 108468.
- [59] Y. Le and X. Yang, "Tiny ImageNet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.
- [60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [61] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.
- [62] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
- [63] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.
- [64] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," 2015, *arXiv:1506.01497*.
- [65] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.
- [66] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.
- [67] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2117–2125.
- [68] K. Chen et al., "MMDetection: Open MMLab detection toolbox and benchmark," 2019, *arXiv:1906.07155*.
- [69] Q.-L. Zhang and Y.-B. Yang, "SA-Net: Shuffle attention for deep convolutional neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 2235–2239.
- [70] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, "A²-Nets: Double attention networks," 2018, *arXiv:1810.11579*.
- [71] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 10705–10714.
- [72] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," 2016, *arXiv:1608.03983*.
- [73] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: Training ImageNet in 1 hour," 2017, *arXiv:1706.02677*.
- [74] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
- [75] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 618–626.



ESLAM MOHAMED BAKR is currently pursuing the master's degree with Cairo University. He is a Senior Machine Learning Engineer at VALEO. He also works as a TA at Zewail University. Recently, he joined KAUST University, as a Visiting Student. He has published around ten papers in machine learning and computer vision tracks and filed one patent.



AHMAD EL-SALLAB received the Ph.D. degree in machine learning from Cairo University. He is an AI Senior Expert at VALEO. He has published more than 30 papers in machine learning and computer vision fields and filed five patents.



MOHSEN RASHWAN received the Ph.D. degree from Queen's University, Canada. He is a Cofounder, the Vice Chairperson, and the Managing Director of the Corporate Software House; and a RDI Professor at Cairo University. He is a member of the Egyptian Export Council for Information Technology (following the ministry of Foreign exports) on behalf of the industry of e-learning and e-content, since 2005. He was a Cofounder of Information Dynamix Company (IDX), Egypt,

for Advanced Internet and Telephony Applications, in 2000. He was also a Cofounder of Al-Rowad Software and Training Company, in 1988.

• • •