MMCAP: LEARNING TO BROAD-SIGHT NEURAL NETWORKS BY CLASS ATTENTION POOLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recently, the global average pooling is believed to be losing the local information that saturates the performance of neural networks. In this lossy pooling operation, we propose a new interpretation, termed over-concentration, to explain the real reason why it degrades network performance. We argue that the problem of global average pooling is disregarding the local patterns by relying solely on the overly concentrated activation. Global average pooling enforces the network to learn objects regardless of their location, so features tend to be activated only in specific regions. To support this claim, we provide a novel analysis of the problems that over-concentration brings about in the network with extensive experiments. We analyze the over-concentration through problems arising from feature variance and dead neurons that are not activated. Based on our analysis, we introduce a multi-token and multi-scale class attention pooling layer to alleviate the over-concentration problem. The proposed attention pooling method captures rich, localized patterns with an efficient network design using multiple scales and tokens. Our method is highly applicable to downstream task and network architectures such as CNN, ViT, and MLP-Mixer. In our experiment, the proposed method improves MLP-Mixer, ViT, and CNN architectures with little additional resources, and a network employing our pooling method works well compared to even state-of-the-art networks. We will opensource the proposed pooling method.

1 INTRODUCTION

This paper presents a new analysis of the global average pooling (GAP) method that significantly impacts the learning process. Since gradients are backpropagated, generating discriminant features in last pooling layer is a critical stage in deep neural networks. For this reason, the lossy GAP, the last layer of a network, prevents entire networks from learning particular local information. Despite its importance in the learning process, the insufficient study has been conducted to establish why this information loss occurs and how it affects the learned network. Therefore, in this paper, we present a new analysis that the performance degradation stems from the inability to learn local patterns properly due to the **over-concentration** problem.

Since GAP over-concentrates the activated neurons into the most discriminant small region (*i.e.*, center of an object), the localized patterns disappear in the pooled feature (Christlein et al., 2019). We call this drawback of GAP as 'over-concentration' and point to it as the cause of the degradation of network performance. We empirically show that the over-concentrated feature map from GAP decreases the spatial and channel variance, which deteriorates the feature representation power of neural networks. The feature representation power grows when the variance of the feature map is high; however, GAP reduces the feature variance by over-concentrating only the specific region, which negatively affects the performance of the network. We analyze this over-concentration problem of GAP through extensive experiments from various perspectives in Sec.4.

Motivated by the analysis of the over-concentration problem, we introduce a multi-scale and multi-token class attention pooling (MMCAP) method. The proposed MMCAP use efficient attention architecture to learn the local spatial patterns, which does not concentrate on a specific region in the pooling layer. In order to maximize the capability of learning local patterns, we offer 1) **multi-token** to learn various patterns and 2) **multi-scale** to have broad-sight information on the pooled feature map. We also propose token-distillation to boost the performance of the MMCAP and architecture

suitable for dense prediction to increase its applicability. Existing class attention pooling cannot directly use tokens for dense prediction tasks such as pyramid networks, whereas we propose a new architecture in which encoded features of class tokens can be directly used in a dense prediction.

In summary, we give an analysis of the over-concentration problem owing to global average pooling, which provides the justification for creating the proposed MMCAP. We show that the proposed MMCAP improves the baseline backbone networks by 1.7% ResNet50, 1.6% PiT-S and 0.3% ResMLP-S24 with minimal overhead. We also conduct various experiments on downstream tasks and ablation studies to validate the proposed method. We condense the contribution of this paper as follows:

- We present a new analysis that the global average pooling degrades the performance of the network due to the over-concentration problem. We observe that global average pooling over-concentrate on the subtle region that disturbs a network to learn diverse local patterns.
- We, in accordance with the analysis, propose the MMCAP method that efficiently pools the local spatial patterns, which uses the multi-token and multi-scale strategy.
- We show the proposed MMCAP performs well compared to existing pooling layers and state-of-the-art networks. We also demonstrate that a network with our MMCAP achieves competitive performance on fine-tuning and downstream tasks.

2 Related Work

Traditional Pooling Layer. The fully connect (FC) layer is employed as the final aggregating method in early CNN architecture(LeCun et al., 1998), AlexNet(Krizhevsky et al., 2012) and VGGNet(Simonyan & Zisserman, 2014). FC layer is straightforward and effective in encoding fine-grained local features, but it requires a large number of parameters and is weak in translation invariance property. To improve the translation invariance property, the global average pooling (GAP) layer is introduced in modern CNN architectures (Szegedy et al., 2015; He et al., 2016). Although GAP has been the de facto standard pooling method, there have been numerous attempts to replace it with cutting-edge methods such as orderless (Gong et al., 2014), bilinear (Lin et al., 2015; Gao et al., 2016; Cui et al., 2017), and DFT pooling(Ryu et al., 2018; Xu & Nakayama, 2019). None of these methods, however, is able to exceed GAP in terms of throughput and model performance because GAP uses superfast average down operation layer while ensuring the translational invariance property. Therefore, GAP is still being used as last pooling layer in current state-of-the-art architectures(Brock et al., 2021; Tan & Le, 2021; Tu et al., 2022).

Class Attention Pooling. In the visual recognition task, the class token was first introduced by vision transformer (ViT) architecture Dosovitskiy et al. (2020). Touvron et al. (2021c) found that the class attention pooling can be improved when inserting it into the last multi-head self-attention (MHSA) layer in a network. As a result of this funding, class attention pooling has now been exploited in numerous vision transformer architectures(Yuan et al., 2021; Yu et al., 2022; Touvron et al., 2021c). In addition, there have been a few attempts to apply class attention pooling in CNN architectures(Touvron et al., 2021b; Ridnik et al., 2021b). However, these methods are difficult to be used generally for reasons such as tailoring the backbone architecture to a specific condition.

Feature Aggregation. Fusing multi-scale features has been extensively studied in dense prediction tasks such as object detection (Lin et al., 2017) and semantic segmentation (Long et al., 2015). Also, in the image classification task, there have been studies using various levels of features (Sun et al., 2018; Huang et al., 2017; Anonymous, 2022). Aggregating multi-token has been considered in recent network architectures(Xu et al., 2022; Ridnik et al., 2021b). Xu et al. (2022) uses multi-tokens to better capture class-specific attention in a semantic map for the weakly supervised semantic segmentation task. While previous studies employed multi-tokens for the class-specific task, we discover that using multi-tokens is valid even in a single task. Therefore, we design the generalized architecture with multi-tokens that are effective for any backbones and tasks.

3 Background: Class Attention Pooling

We compare GAP with class attention pooling (CAP) to address the over-concentration problem. Since the CAP applies the class tokens to local pixels separately, locality remains in the pooled

features, thus avoiding the over-centralization problem. In this part, we introduce the CAP method before comparing it to GAP for the over-concentration problem in Sec.4.

CAP(Touvron et al., 2021c) aggregates image features $x_{img} \in \mathbb{R}^{n \times c}$ into class token $x_{class} \in \mathbb{R}^{1 \times c}$ by using multi-head class attention (MHCA) as:

$$Q = W_q x_{class} + b_q \in \mathbb{R}^{1 \times c}, \quad K = W_k z + b_k \in \mathbb{R}^{(1+n) \times c}, \quad V = W_v z + b_v \in \mathbb{R}^{(1+n) \times c} \\ \text{where} \quad z = [x_{class}, x_{img}] \in \mathbb{R}^{(1+n) \times c} \tag{1}$$

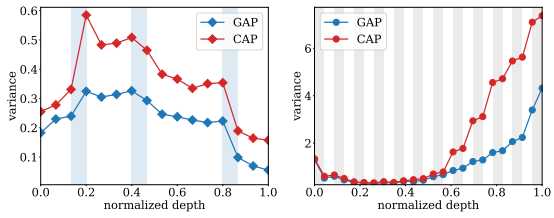
where $[W_q, W_k, W_v, W_o]$ denote embedding matrix, $[b_q, b_k, b_v, b_o]$ are the corresponding bias, and images features and class tokens are represented as x_{class} and x_{img}. The pooled feature $\mathcal{F}$ of CAP is, then, generated by multiplying class attention map C with value V as:

$$\mathcal{F} = W_o C V + b_o \in \mathbb{R}^{1 \times c} \quad \text{where} \quad C = Softmax(QK^T / \sqrt{d/h}) \in \mathbb{R}^{1 \times (1+n)}. \tag{2}$$

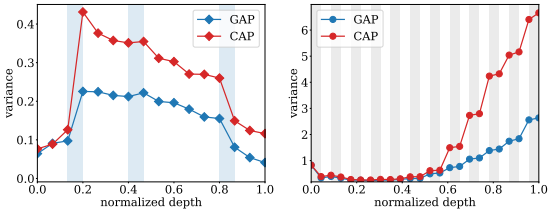
Complexity of CAP, as in Eq. 1 and 2, is much lower compared to the vanilla self-attention method. Since only class tokens are used as the query, which is much smaller than image features, the complexity of CAP is $O(N)$, which is proportional to the number of spatial pixels of image features N. In contrast, the complexity of vanilla self-attention, which uses image features for both querying and keying, is $O(N^2)$. Therefore, we are able to exploit CAP that maximizes local information using only manageable computational complexity.

4 Analysis

In this section, we analyze the GAP in terms of over-concentration by comparing it with the CAP. First, we tackle the over-concentration problem by the feature variance in Sec.4.1. To explore the problem in more detail, we investigate the dead neurons that express the degree of activation in a network depending on the over-concentration problem in Sec. 4.2. Consequently, we study the non-salient object recognition task (Naseer et al., 2021) in Sec. 4.3. For the empirical study, we compare both pooling layers to ResNet-50 and DeiT-S/16 networks to show consistent results.



(a) Channel variance



(b) Spatial variance

Figure 1: Experimental study on the channel and spatial variance(Park & Kim, 2022). We use ResNet (left plots) and DeiT-S/16 (right plots) as backbone networks. In all cases, CAP achieves higher variance compared to GAP.

4.1 Feature Variance

Feature over-centralization entails that semantically similar features span an entire spatial area, resulting in the extinction of minor but discriminant local features in a network. The over-centralization is primarily caused by GAP, which focuses only on the small regional information due to its spatially average down operation. This issue can also be found in network visualization studies where they found them the class activation map (CAM) resides in small regions. (Zhou et al., 2016; Kim et al., 2017; Li et al., 2018) To overcome this problem, they diversify the input (Wei et al., 2017; Jiang et al., 2022) and network architecture (Wang et al., 2020) to expand the activated regions in a feature map. These studies focus mostly on segmentation tasks in which CAM results are employed as weak labels by expanding the activation map to cover the entire object. We assume that the reason for this problem that activated neurons are concentrated in a narrow region is due to the GAP's average down operation. GAP recognizes objects regardless of their location, so only the specific location is activated, and the rest of the patterns are discarded(Qiu, 2018).

We verify our analysis of the over-centralization by feature variance. Each channel of the feature map represents the pattern of an object. If the channel variance is high, neurons are activated at

various spatial positions. Both channel and spatial variance indicate the diversity of information possessed by the feature map, and the more monotonic features, the lower the variance. Therefore, over-concentrated features are spatially monotonic, and thus the variance is inevitably low. We compare the feature variances of networks with GAP and CAP as shown in Fig.1. Unsurprisingly, the CAP, which pools each local pixel independently, achieve much higher variance on both CNN and ViT architecture. This finding of feature variances supports that the GAP leads to the over-centralization problem of entire layers of a network.

4.2 Dead Neuron

Over-centralization problems exacerbate the dead neuron(Lu et al., 2019) problem in deep neural networks. Due to over-centralization, local spatial patterns outside the central position are not activated even though they have information that can identify objects. A spatial position that is ignored in GAP at the end of a network loses the opportunity to be learned in all layers during backpropagation. Therefore, as shown in Fig. 2, a network trained with GAP generates much more dead neurons than CAP. These differences greatly affect the network performance, and we believe that a network with CAP learns richer local patterns and thus alleviates the dead neuron problem.

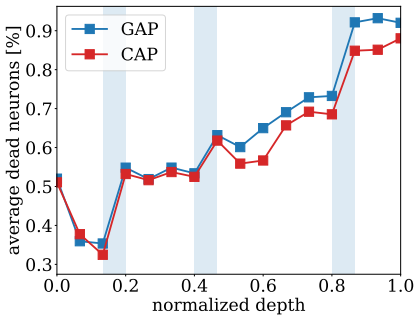


Figure 2: Experimental study on the dead neurons. We measure the ratio of non-activated neurons in each layer. In this result, CAP activates more neurons compared to GAP.

4.3 Non-salient recognition

We point out that GAP is vulnerable to the disappearance of salient regions due to over-concentration problems. Since GAP recognizes objects by concentrating on a narrow local region, if the region is occluded, the problem of not recognizing objects occurs well. As shown in Fig.3, as the disappeared salient region becomes larger, the performance of the GAP decreases rapidly. This is because the over-concentrated region of GAP is too narrow, so GAP does not activate features on the salient but outside the over-concentration region. However, the CAP that independently pools all local pixels has a broad concentration region, so it recognizes the occluded object better than the GAP.

5 Method

On the basis of our analysis, we propose multi-token and multi-scale class attention pooling (MMCAP) that learns enhanced local patterns with a minimal computational budget. Our design principle is to 1) learn richer localities while 2) use fewer resources compared to the vanilla CAP method. As shown in Fig. 4 vanilla CAP performs attention operation on a single-scale feature map with a sin-

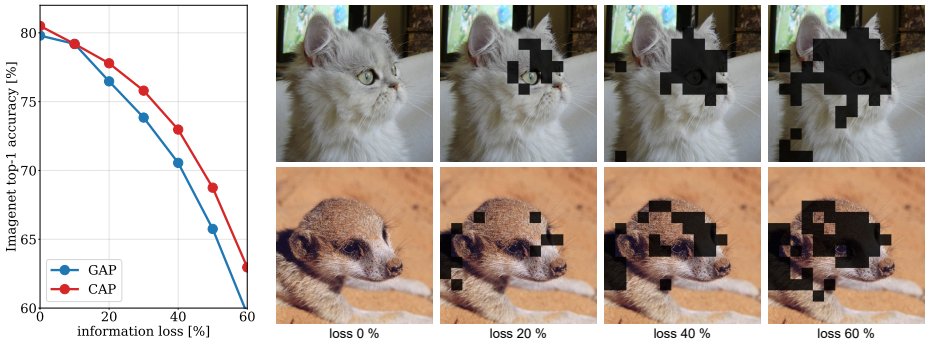
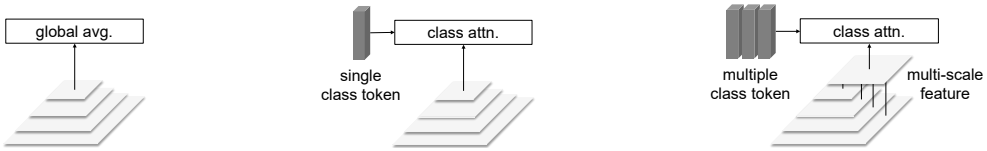


Figure 3: Experimental study on non-salient object recognition. We measure image classification accuracy after artificially removing the object's salient area using a self-supervised ViT model DINO (Caron et al., 2021; Naseer et al., 2021).



(a) GAP (b) CAP (c) MMCAP

Figure 4: Schematic illustration of GAP, CAP, and MMCAP methods. The proposed MMCAP exploit the multi-tokens and multi-scale features in a network.

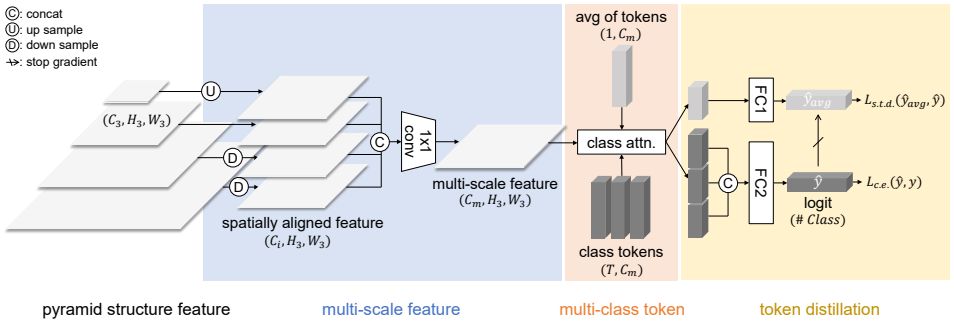


Figure 5: Workflow of the proposed MMCAP. We combine multi-scale features (blue region), apply multi-tokens (red region), and then train a network by token-distillation (yellow region).

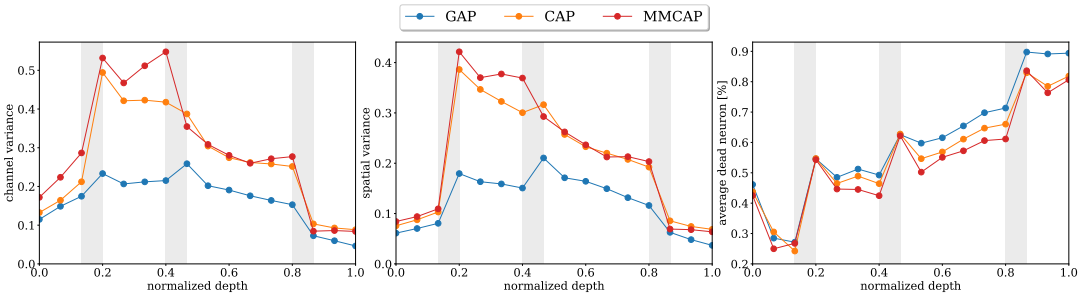
gle token; it is limited in learning local patterns and also has the disadvantage of rapidly increasing parameters in a specific architecture. Single-level feature learning has a drawback in that it does not fully leverage the expressive feature representation of deep neural networks; we address this issue with efficient multi-token and multi-scale approaches. Also, while the parameters of vanilla CAP significantly increase according to the dimension of its input channel (*e.g.*, 2048 of ResNet), our MMCAP reduce it greatly reduce due to the efficient architecture design as shown in Fig.5. In the appendix, we explore the applicability of the proposed MMCAP to the fine-tuning and dense prediction of the downstream tasks. Therefore, the proposed MMCAP extends the vanilla CAP method with better performance, faster, and fewer resources. For a better understanding, each component of the MMCAP is detailed, along with our design choices according to the analysis in Sec.4.

5.1 Multi-token class attention pooling

Output features ideally have as many local patterns as the number of multi-tokens (Xu et al., 2022; Ridnik et al., 2021b) compared to vanilla CAP. As described in Sec.4, class tokens effectively capture local spatial patterns; hence, increasing the number of class tokens used in CAP ensures a rich representation of the feature localization. To investigate this further, we conduct experiments to determine how the learned model changed as the number of class tokens grew. As shown in Fig.6, the number of class tokens, the degree of channel variance, and the dead neurons are close to a linear relationship. Also, in Fig.7, the performance of the image classification task improves as the number of tokens increases. These results support the use of multiple tokens to more efficiently learn local information. In spite of its efficiency, the multi-token approach has the downside of increasing the output feature's size by a factor of t. Since the output feature of MMCAP is connected to the FC layer, the number of parameters increases substantially when the dimension rises. We overcome this issue by decreasing the channel dimension c in the following multi-scale feature method.

5.2 Multi-scale feature

In deep neural networks, multi-scale features are beneficent for learning local patterns of varying sizes. Pyramid aggregation architectures(Lin et al., 2017) have been widely used to learn multi-scale features, but they are difficult to employ with attention pooling due to their complex connectivity.



(a) Channel variance (b) Spatial variance (c) Dead Neuron

Figure 6: Experimental study on the feature variance and dead neurons for the proposed MMCAP. The networks used in this comparison are trained for 100 epochs on the ImageNet-1k dataset. (a) and (b): MMCAP increases channel and spatial variation. (c): MMCAP reduces dead neurons.

Therefore, we adopt light-weight multi-level feature aggregation method from (Anonymous, 2022) for our pooling method. In the case of ResNet, the four-stage output features of 56×56, 28×28, 14×14, and 7×7 resolution are interpolated to have the same resolution, and then the four feature maps are concatenated. To produce the final multi-scale features, the channel dimension of the concatenated features is decreased by the MLP layer. This straightforward procedure (Anonymous, 2022) is suitable for our pooling approach since it uses a limited resource budget for the multi-scale feature aggregation and also controls the resolution and channel dimension of features that are fed to MMCAP. The optimal number of class tokens is dependent on the feature resolution, so we empirically find the balanced resolution and channel dimension. As shown in Fig.7, when the feature resolution is 7×7, the performance reaches to plateau at four tokens, while features of 14×14 further improve the performance at more tokens. This result indicates that the higher the resolution of the feature map, the more local patterns it has, so more tokens are needed.

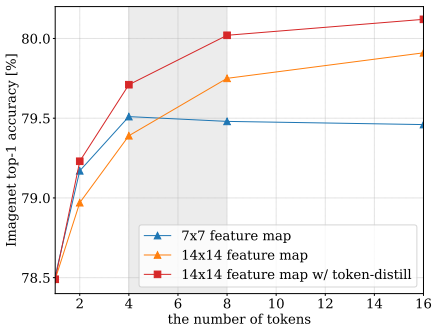


Figure 7: Experimental study on the feature resolution for MMCAP. We confirm that the higher feature resolution gives better performance with sufficient tokens.

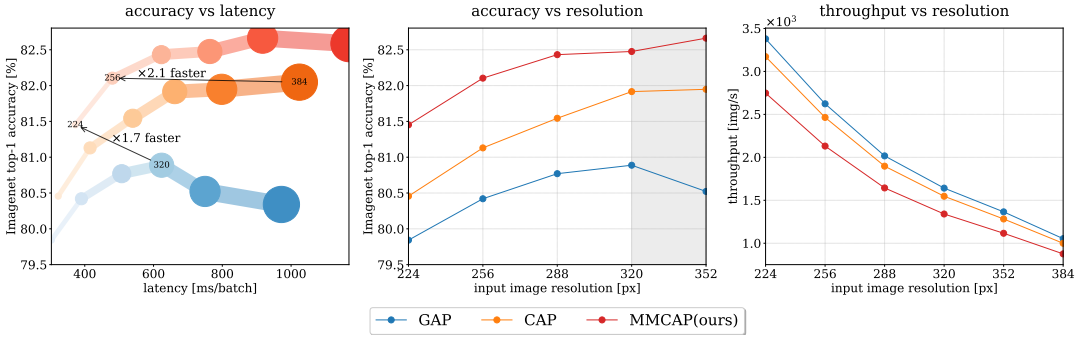
5.3 TOKEN DISTILLATION

We introduce the self token distillation that reduces the Kullback-Leibler divergence between the prediction of the multiple class tokens and the average of them. We pass the average class token through the MMCAP together with the original multi-tokens and then connect each FC layer to each output token of MMCAP to obtain the probability of each output token. The probability of the average-token is then distilled with the predicted probability of the original multi-tokens. In this way, the average-token learns common knowledge from the multi-tokens and the performance of the network improves when the average-token is removed during inference. Also, even if only one average-token is used, it shows quite good performance compared to a network trained with a single token without our token distillation.

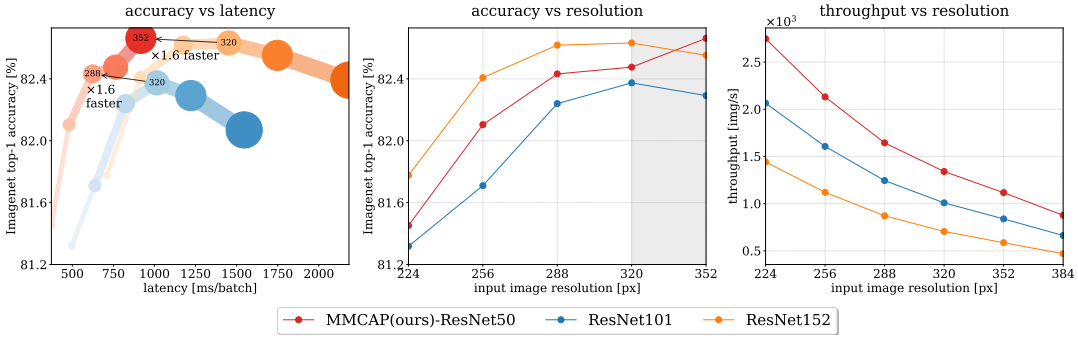
We borrow the mathematical definition for the proposed token-distillation from (Touvron et al., 2021a) as:

$$\mathcal{L}_{global} = (1-\lambda)\mathcal{L}_{CE}(\sigma(Z_{multi}), y) + \lambda T^2 \mathrm{KL}(\sigma(Z_{avg}/T), \sigma(Z_{multi}/T)), \tag{3}$$

where Z_{multi} denotes the logits of multiple class tokens, Z_{avg} average of them, T the distillation temperature, λ the balancing constant ratio between the Kullback-Leibler divergence (KL) and the cross-entropy loss($\mathcal{L}_{CE}$) , y the ground truth labels and σ the softmax function. For simplicity, we fix



(a) Comparison on the performance of GAP, CAP, and MMCAP with regard to the latency and input image resolution.



(b) Comparison on the performance of ResNet50+MMCAP, ResNet101, and ResNet152 with regard to the latency and input image resolution.

Figure 8: Experimental study of extensive comparison of the proposed MMCAP with other pooling methods and networks. In the left figures, larger points signify that high input resolution is employed. (a) We confirm that, compared to GAP and CAP, MMCAP achieves much higher performance while using fewer resources. (b) MMCAP-ResNet50 shows better performance compared to vanilla ResNet with deeper layers, and in particular, as the resolution increases, MMCAP performs better.

distillation hyper-parameters as $\lambda = 0.5$ and $T = 1.0$. It is confirmed that MMCAP, to which token-distillation is applied along with multi-token and multi-scale, shows significantly better performance in various aspects compared to GAP and CAP, as shown in Fig.8.

6 Experiment

In this section, we perform extensive experiments to validate the proposed method on four visual recognition tasks: image classification, object detection, semantic segmentation, and instance segmentation. In the image classification task, we investigate the effectiveness of the proposed MMCAP in four sub-experiments. In the appendix, the network is also transferred to a dense prediction architecture, demonstrating the applicability of the downstream tasks.

6.1 Image Classification

In this section, we evaluate the proposed MMCAP using two sub-experiments such as 1) ablation study, 2) SOTA comparison. To train our model, we use ImageNet-1k dataset (Russakovsky et al., 2015) with 300 epochs. For a fair comparison, we do not use any external dataset. The details of training hyperparameters can be found in the appendix.

#Token	M-scale	Res.	T-distill	#Epoch	Top-1 (Acc. %)	Throughput (img/s)	Params (M)	FLOPs (G)
0	-	7	-	100	77.05	3414.7	25.6	4.1
1	-	7	-	100	78.51 (+1.5)	3163.9	59.1	4.5
4	-	7	-	100	79.12 (+0.6)	3094.5	65.3	4.6
4	✓	7	-	100	79.51 (+0.4)	2944.3	73.3	5.0
4	✓	**14**	-	100	79.36 (- 0.2)	2735.8	40.0	5.3
8	✓	14	-	100	79.75 (+0.4)	2706.2	44.1	5.4
8	✓	14	✓	100	80.02 (+0.3)	2693.4	45.1	5.4
8	✓	14	✓	**300**	81.43 (+1.4)	2693.4	45.1	5.4

Table 1: Ablation study on the proposed MMCAP. We use ResNet-50 as a backbone network. M-scale and T-distill denote the multi-scale and token-distillation, and the column of Res. indicates the input feature resolution for the MMCAP.

Model	Method	Top-1 acc.(%)	Throughput (img/s)	Param (M)	FLOPs (G)
ResNet50	GAP	79.8	3400.7	25.6	4.1
	CAP	80.6(+0.8)	3176.0	59.1	4.5
	MMCAP-light	80.8(+1.0)	2821.7	36.9	5.3
	MMCAP	81.5(+1.7)	2757.6	45.1	5.4
PiT-S	GAP	79.8	2709.6	23.3	2.4
	CAP	81.3 (+1.5)	2667.1	25.9	2.4
	MMCAP	81.4 (+1.6)	2543.0	28.6	2.6
MLP-Mixer	GAP	79.4	2186.9	30.0	6.0
	CAP	78.5(-0.9)	1996.6	31.2	6.0
	MCAP	79.7(+0.3)	1989.7	32.7	6.0

Table 2: Experimental study on the proposed method for CNN, ViT, and MLP-Mixer architecture. In all three architectures, the proposed MMCAP performs well with the manageable resource overhead. MMCAP-light is a result of using only the average distillation token, so it has the advantage of reducing the parameters in FC layers.

6.1.1 Ablation study

Table 1, 2, and 3 demonstrates that the proposed MMCAP improves performance in various settings. Each element (*e.g.*, multi-token, multi-scale, and token-distillation) of the proposed method performs better than its respective baseline in Table 1. We further evaluate GAP, CAP, and MMCAP on CNN (*i.e.*, ResNet50), ViT (*i.e.*, PiT-S), MLP-Mixer architectures in Table 2. The proposed MMCAP outperforms GAP and CAP in all three architectures by a significant margin. Specifically, the suggested MMCAP exhibits little delay across all architectures, and the increase in parameters and FLOPs of the PiT-S and MLP-Mixer is likewise insignificant. We also compare them on scale-up architectures in Table 3. The proposed MMCAP performs consistently better than CAP, particularly in bigger architectures with large input sizes. This demonstrates that MMCAP has excellent scalability; specifics are explained in Sec. A.3.

Model	GAP				MMCAP			
	Top-1 Acc.(%)	Throughput (img/s)	Param (M)	FLOPs (G)	Top-1 Acc.(%)	Throughput (img/s)*	Param (M)	FLOPs (G)
ResNetD-T	81.1	1470.1	25.6	8.8	82.9(+1.8)	1233.4	45.2	11.3
ResNetD-S	82.7	866.2	50.8	17.0	83.5(+0.8)	734.8	78.9	20.8
ResNetD-B	83.4	486.2	72.6	27.8	83.9(+0.5)	431.4	103.4	32.2
ConvNext	82.4	1030.4	28.6	9.1	83.0(+0.6)	988.2	33.1	9.5

Table 3: Experimental study of the comprehensive comparison on GAP and MMCAP. Configuration of ResNetD-T, S, B are shown Table 6.

6.1.2 SOTA COMPARISON

We compare the proposed method with the current SOTA networks as shwo in Table **??**. We apply MMCAP to the ResNet-D(He et al., 2019) and ConvNextliu2022convnet backbones in comparison with the SOTA networks such as ResMLP(Touvron et al., 2022), DeiT(Touvron et al., 2021a), SwinTransformer(Liu et al., 2021), CaiT(Touvron et al., 2021c), ResNet(He et al., 2016), EfficientNet(Tan & Le, 2019), ResNest(Zhang et al., 2022), ConViTd2021convit, and TResNet(Ridnik et al., 2021a). Although we only replace the pooling layer with the proposed MMCAP, we achieve considerably better performance than the SOTA methods. In addition, its validity is verified in important architectures such as MLP-Mixer, ViT, CNN, and hybrid.

Architecture	Network	Train (px)	Test (px)	Top-1 Acc.(%)	Throughput (img/s)	Params (M)	FLOPs (G)
MLP-Mixer	ResMLP-S12	224	224	76.6	4255.5	15.4	3.0
	ResMLP-S24	224	224	79.4	2144.0	30.0	6.0
	ResMLP-S36	224	224	79.8	1414.7	44.7	8.9
	MMCAP-ResMLP-S24	224	224	79.7	1989.7	32.7	6.0
Transformer	DeiT-S	224	224	79.8	2664.1	22.0	4.2
	Swin-T	224	224	81.3	1710.7	28.3	4.4
	CaiT-XXS-36	224	224	79.1	1024.1	17.3	3.8
	MMCAP-PiT-S	224	224	81.4	2543.0	28.6	2.6
CNN (Hybrid)	ResNet50	224	224	79.8	3400.7	25.6	4.1
	EfficientNet-B3	300	300	81.6	1951.6	12.0	1.8
	CrossViT-15	224	240	81.5	1594.5	27.5	5.2
	MMCAP-ResNetD-T	224	224	81.8	2557.1	45.2	5.6
	MMCAP-ConvNext-T	224	224	82.3	2006.8	33.1	4.6
	ResNet152	224	224	81.8	1446.1	60.2	11.5
	ConViT-S	224	240	81.3	1288.6	27.8	5.4
	TResNet-M	224	224	80.8	1223.8	31.4	5.7
	EfficientNet-B4	380	380	82.9	932.9	19.0	4.4
	CrossViT-18	224	224	82.5	921.3	43.3	8.2
	MMCAP-ResNetD-T	224	320	82.9	1233.4	45.2	11.3
	MMCAP-ResNetD-TΥ	224	320	83.7	1222.6	53.4	11.3
	MMCAP-ConvNext-S	224	224	83.3	1223.2	62.3	9.2
	ResNest101	224	224	83.0	678.2	48.3	10.2
	ConViT-B	224	224	82.4	627.1	86.5	16.8
	EfficientNet-B5	456	456	83.6	444.3	30.4	10.3
	TResNet-M	448	448	83.2	301.3	31.4	22.9
	MMCAP-ResNetD-S	224	320	83.5	771.5	78.9	20.8
	MMCAP-ResNetD-SΥ	224	320	84.3	742.5	89.2	20.8
	MMCAP-ConvNext-S	224	320	83.8	602.4	62.3	18.7

Table 4: Experimental study on the comparison between the proposed method with SOTA networks. We only include networks that are trained without extra data. Υ indicates that the knowledge distillation is utilized. We do not fine-tune the model on high resolutions. Throughput is measured on RTX 3090 GPU device using TIMM library (Wightman, 2019). Results of ResNet (He et al., 2016) and RegNet (Radosavovic et al., 2020) are imported from (Wightman et al., 2021; Touvron et al., 2021a).

7 CONCLUSION

In this paper, we present a new analysis of the global average pooling that leads to the problem of over-concentration. We analyze the consequences of the over-concentration problem on network learning via feature variance, dead neurons, and salient object recognition. This analysis serves as the basis for our MMCAP method, which focuses on the acquisition of abundant local patterns. With our MMCAP layer, we also offer token-distillation for the efficient usage of the multi-tokens and multi-scales. Moreover, the suggested MMCAP is readily applicable to downstream tasks, such as object detection and segmentation. Despite the fact that we replace the pooling layer with MMCAP, it exhibits significantly better performance than SOTA networks, and we anticipate that these results will serve as a groundwork for future research.

REFERENCES

Anonymous. Learning less-correlated features in network aggregation. OpenReview Preprint, anonymous preprint under review, 2022.

Andy Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning*, pp. 1059–1071. PMLR, 2021.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Vincent Christlein, Lukas Spranger, Mathias Seuret, Anguelos Nicolaou, Pavel Král, and Andreas Maier. Deep generalized max pooling. In *2019 International conference on document analysis and recognition (ICDAR)*, pp. 1090–1096. IEEE, 2019.

Yin Cui, Feng Zhou, Jiang Wang, Xiao Liu, Yuanqing Lin, and Serge Belongie. Kernel pooling for convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2930, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 317–326, 2016.

Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *European conference on computer vision*, pp. 392–407. Springer, 2014.

Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12259–12269, 2021.

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*, pp. 770–778, 2016.

Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 558–567, 2019.

Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11936–11945, 2021.

Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens Van Der Maaten, and Kilian Q Weinberger. Multi-scale dense convolutional networks for efficient prediction. *arXiv preprint arXiv:1703.09844*, 2:2, 2017.

Peng-Tao Jiang, Yuqi Yang, Qibin Hou, and Yunchao Wei. L2g: A simple local-to-global knowledge transfer framework for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16886–16896, 2022.

Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Two-phase learning for weakly supervised object localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 3534–3543, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9215–9223, 2018.

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

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 1449–1457, 2015.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11976–11986, 2022.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*, 2019.

Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.

Namuk Park and Songkuk Kim. How do vision transformers work? *arXiv preprint arXiv:2202.06709*, 2022.

Suo Qiu. Global weighted average pooling bridges pixel-level localization and image-level classification. *arXiv preprint arXiv:1809.08264*, 2018.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020.

Tal Ridnik, Hussam Lawen, Asaf Noy, Emanuel Ben Baruch, Gilad Sharir, and Itamar Friedman. Tresnet: High performance gpu-dedicated architecture. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1400–1409, 2021a.

Tal Ridnik, Gilad Sharir, Avi Ben-Cohen, Emanuel Ben-Baruch, and Asaf Noy. Ml-decoder: Scalable and versatile classification head. *arXiv preprint arXiv:2111.12933*, 2021b.

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

Jongbin Ryu, Ming-Hsuan Yang, and Jongwoo Lim. Dft-based transformation invariant pooling layer for visual classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 84–99, 2018.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Shuyang Sun, Jiangmiao Pang, Jianping Shi, Shuai Yi, and Wanli Ouyang. Fishnet: A versatile backbone for image, region, and pixel level prediction. *Advances in neural information processing systems*, 31, 2018.

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*, pp. 1–9, 2015.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pp. 10096–10106. PMLR, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021a.

Hugo Touvron, Matthieu Cord, Alaaeldin El-Nouby, Piotr Bojanowski, Armand Joulin, Gabriel Synnaeve, and Hervé Jégou. Augmenting convolutional networks with attention-based aggregation. *arXiv preprint arXiv:2112.13692*, 2021b.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 32–42, 2021c.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. *arXiv preprint arXiv:2204.01697*, 2022.

Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12275–12284, 2020.

Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1568–1576, 2017.

Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.

Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, and Dan Xu. Multi-class token transformer for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4310–4319, 2022.

Yuhao Xu and Hideki Nakayama. Dct based information-preserving pooling for deep neural networks. In *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 894–898. IEEE, 2019.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *arXiv preprint arXiv:2106.13112*, 2021.

Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2736–2746, 2022.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

A APPENDIX

A.1 REPRODUCABILITY: IMPLEMENTATION DETAILS

Table 5 provides our training hyper-parameters used to train multiple backbone architectures on different datasets. Except for ResNet, we adhere to the original train design used to train backbone networks for the ILSVRC-2012 pretraining task. Hyperparameters of ResNet training reference an A2 configuration in (Wightman et al., 2021). In the A2 configuration, we decrease the batch size to accommodate our computational capabilities (2 RTX 3090 24GB GPU) and adjust the learning rate correspondingly. We exploit the standard fine-tuning receipt for downstream tasks, which comes from the (Touvron et al., 2021a).

configuration reference work	ResNet Wightman et al.	ConvNext Liu et al.	DeiT(PiT) Touvron et al.	ResMLP Touvron et al.
train res.	224	224	224	224
test res.	224	224	224	224
test crop ratio	0.95	0.95	0.95	0.95
epoch	300	300	300	350
batch size	512	512	512	512
criterion	BCE	CE	CE	CE
optimizer	LAMB	AdamW	AdamW	LAMB
lr	3.5e-3	4e-3	7e-3	3.5e-3
lr decay	cosine	cosine	cosine	cosine
weight decay	0.02	0.05	0.05	0.2
warmup epochs	5	20	5	5
h.flip	✓	✓	✓	✓
rand augmentation	7/0.5	9/0.5	9/0.5	9/0.5
cutmix alpha	1.0	1.0	1.0	1.0
mixup alpha	0.1	0.8	0.8	0.8
erasing prob.	0.0	0.25	0.25	0.25
ema	-	✓	-	-

Table 5: Summary of ILSVRC-2012 training hyper-parameters.

Stage No.	Tiny		Small		Base	
	# Layers	# Channels	# Layers	# Channels	# Layers	# Channels
1	3	64	4	64	4	84
2	4	128	5	128	5	168
3	6	256	13	320	21	336
4	3	512	3	640	3	672

Table 6: Model configurations used for scaling up baseline ResNet50D(Tiny) model. This series of models are used to compare proposed method with SOTA network in Table 4

A.2 DOWNSTREAM TASK WITH MMCAP

In this subsection, we propose architecture tweaks for dense prediction tasks, including object detection and segmentation. In the previous work(Touvron et al., 2021c;b; Ridnik et al., 2021b), CAP could not be used directly for dense prediction; therefore, CAP is removed, and only the backbone should be used for dense prediction. The inability of their architecture to fine-tune the CAP prevents it from taking advantage of attention pooling; hence its performance is even worse than that of a network that has been trained with GAP. Therefore, we tweak a network architecture to have the connectivity between MMCAP and output layers in a dense prediction backbone network.

In the proposed dense prediction network, we divide the output features of MMCAP by the number of stages and add them to each stage, as shown in Fig.9. We broadcast the output features of MMCAP to the features of each stage. In this way, after the multi-scale feature passes the MMCAP, the divided tokens are fed back to each scale, resulting in diversifying the connectivity of the network. Thus, each stage of a network will be able to enjoy the benefits of MMCAP for the dense prediction tasks.

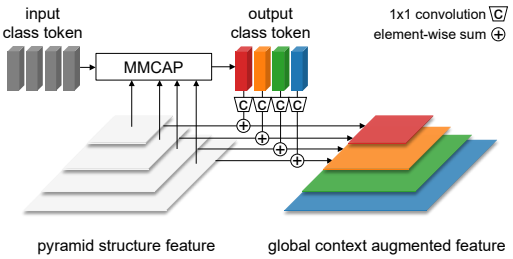


Figure 9: Workflow of the proposed dense prediction network. We distribute the output tokens to each backbone layer.

Experimental result. We validate the applicability of the proposed MMCAP to the downstream task by evaluating its performance in three dense prediction experiments. Unlike existing CAP methods, a network employing MMCAP may directly use out features of the pooling layer for the dense prediction. Therefore, we confirm that MMCAP outperforms GAP and CAP in terms of object detection, semantic segmentation, and instance segmentation in Table 7, 8, 9, and 10. In all experiments, our MMCAP uses a modest amount of more resources while obtaining much better performance.

Downstream task	GAP cls. acc:77.0(%)	CAP cls. acc:78.5(%)	
		w/o tweak	w/ tweak
Detection mAP(%)	38.1	37.7(-0.3)	**39.2(+1.1)**
Segmentation mIOU(%)	37.1	36.6(-0.5)	**37.7(+0.6)**

Table 7: Experimental study on the dense prediction task. The performance is greatly increased when our architectural tweak for dense prediction is used.

	Method	Top-1 Acc.(%)	boxAP (%)	AP@50 (%)	AP@75 (%)	Params (M)	FLOPs (G)	Troughput (img/s)
Faster R-CNN	GAP	79.8	39.4	60.9	42.8	41.8	216.7	28.1
	CAP	80.5	40.0	61.9	43.1	79.5	225.6	27.0
	MMCAP	81.4	40.7	62.5	44.4	58.1	242.4	26.7
Cascade Faster R-CNN	GAP	79.8	42.9	61.4	46.9	69.4	244.3	24.8
	CAP	80.5	43.6	63.2	47.1	107.2	253.2	22.5
	MMCAP	81.4	44.1	63.5	47.8	85.7	270.1	22.3

Table 8: Experimental study of the object detection task on MSCOCO-2017 dataset. We use Mask R-CNN as the backbone network.

	Method	Top-1 Acc.(%)	mIOU (%)	mACC (%)	Params (M)	FLOPs (G)	Troughput (img/s)
ResNet50	GAP	79.8	37.3	47.9	28.5	177.9	38.6
	CAP	80.5	36.4(-0.9)	45.8	66.3	186.6	35.8
	MMCAP	81.4	39.2(+1.9)	49.7	44.8	203.0	34.6

Table 9: Experimental study of the semantic segmentation task on ADE20K dataset. We use FPN architecture as the backbone network.

A.3 Scalability on input resolution

The proposed MMCAP works well with scale-up methods for high-resolution input images. Since scalability is one of the most significant aspects of the recent visual recognition task, it is the subject

	Method	Top-1 Acc.(%)	boxAP (%)	MaskAP (%)	Params (M)	FLOPs (G)	Troughput (img/s)
ResNet50	GAP	79.8	40.2	36.8	44.4	269.8	21.8
	CAP	80.5	41.0(+0.8)	37.7(+0.9)	82.2	278.6	19.8
	MMCAP	81.4	41.5(+1.3)	38.1(+1.3)	60.7	295.5	18.7

Table 10: Experimental study on the instance segmentation task using MSCOCO-2017 dataset.

of our experimental studies. As shown in the 'accuracy vs. resolution' plots of Fig.8, the proposed method delivers more performance gains as the input image's resolution increases. We assume that the reason for these results is that as the input resolution grows, there is more local information, but the current GAP is unable to learn it well.

A.4 TRANSFER LEARNING

Transfer learning is used to examine the generalization ability by fine-tuning pre-trained networks to other small datasets. We use four datasets for this transfer learning task such as CIFAR10, CIFAR100, Flowers102, and Stanford-Car. ResNet50 (He et al., 2016) and PiT-S (Heo et al., 2021) are used as the pretrained backbone networks. In table 11, the proposed MMCAP improves the accuracy of image classification in all transfer learning datasets. ResNet50 with MMCAP outperforms the baseline by about 0.5% and 1.5% in CIFAR10 and CIFAR100 datasets. We observe a similar performance improvement in PiT network for all datasets. This finding verifies that substituting GAP with MMCAP improves generalization ability.

Model	Method	Throughput (img/s)	ImageNet (%)	CIFAR10 (%)	CIFAR100 (%)	Cars (%)
ResNet50	GAP	3400.7	79.8	98.2	88.7	87.8
	CAP	3008.4	80.6	98.6	89.6	91.3
	MMCAP	2752.1	81.5	98.7(+0.5)	90.3(+1.5)	91.5(+3.7)
PiT-S	GAP	2709.6	79.8	98.8	90.1	90.4
	CAP	2667.1	81.3	99.0(+0.2)	91.0(+0.9)	90.2
	MMCAP	2543.0	81.4	99.0(+0.2)	91.3(+1.2)	90.5(+0.1)

Table 11: Experimental study on the fine-tuning task. We transfer our networks pre-trained on ImageNet-1k to small datasets.

A.5 CNN DISTILLATION FROM VIT

Most previous studies (Touvron et al., 2021a;c; Graham et al., 2021) distill ViT from CNN model using the distillation token. Initially, network learning utilizing tokens was developed in ViT; therefore the distillation direction (CNN $\rightarrow$ ViT) has been a prevalent strategy. However, the proposed MMCAP uses class tokens on the last pooling layer, so we apply the distillation method in which ViT teaches CNN, as shown in Table.12.

Model	ImageNet Top-1 acc. (%)				Throughput (img/s)		FLOPs (G)	
	224	320	224Υ	320Υ	224	320	224	320
ResNetD-T	81.8	82.9	82.5	83.7	2557.1	1233.4	5.6	11.3
ResNetD-S	82.5	83.5	83.4	84.3	1587.0	771.5	10.3	20.8
ResNetD-B	82.9	83.9	83.7	84.4	935.3	454.2	15.8	32.2

Table 12: Experimental study on knowledge distillation and scale-up architectures with the proposed MMCAP. Υ denotes a network trained by knowledge distillation from VOLO-D1(Yuan et al., 2021).