RaMLP: Vision MLP via Region-aware Mixing

Shenqi Lai, Xi Du, Jia Guo and Kaipeng Zhang

1InsightFace.ai
2Kiwi Tech
3Shanghai AI Laboratory
laishenqi@qq.com, leo.du@kiwiar.com, guojia@gmail.com, kp_zhang@foxmail.com

Abstract

Recently, MLP-based architectures achieved impressive results in image classification against CNNs and ViTs. However, there is an obvious limitation in that their parameters are related to image sizes, allowing them to process only fixed image sizes. Therefore, they cannot directly adapt dense prediction tasks (e.g., object detection and semantic segmentation) where images are of various sizes. Recent methods tried to address it but brought two new problems, long-range dependencies or important visual cues are ignored. This paper presents a new MLP-based architecture, Region-aware MLP (RaMLP), to satisfy various vision tasks and address the above three problems. In particular, we propose a well-designed module, Region-aware Mixing (RaM). RaM captures important local information and further aggregates these important visual clues. Based on RaM, RaMLP achieves a global receptive field even in one block. It is worth noting that, unlike most existing MLP-based architectures that adopt the same spatial weights to all samples, RaM is region-aware and adaptively determines weights to extract region-level features better. Impressively, our RaMLP outperforms state-of-the-art ViTs, CNNs, and MLPs on both ImageNet-1K image classification and downstream dense prediction tasks, including MS-COCO object detection, MS-COCO instance segmentation, and ADE20K semantic segmentation. In particular, RaMLP outperforms MLPs by a large margin (around 1.5% Apb or 1.0% mIoU) on dense prediction tasks. The training code could be found at https://github.com/xiaolai-sqlai/RaMLP.

1 Introduction

In the past decade, Convolutional Neural Networks (CNNs) [Krizhevsky et al., 2012] have shown great success in various computer vision tasks. In recent years, transformers trained by large-scale data [Devlin et al., 2019] dominate most natural language processing tasks. Motivated by that, many research works proposed Vision Transformers (ViTs) [Dosovitskiy et al., 2021; Touvron et al., 2021b; Liu et al., 2021; Yang et al., 2021; Wu et al., 2021a; Wu et al., 2021b; Yuan et al., 2021; Wang et al., 2021], the transformer-based architectures specific for vision, and surpassed CNNs when using large-scale training data.

More recently, MLP-Mixer [Tolstikhin et al., 2021] was proposed to prove the potential of MLPs. Its parameters are almost all learned from fully-connected layers. It achieved comparable results in image classification against CNN-based or ViT-based architectures. Such promising results drive some exploration of MLP-based architectures.

Followed by MLP-Mixer, many advanced MLP-based architectures [Touvron et al., 2021a; Guo et al., 2021; Hou et al., 2021; Wang et al., 2022b] were proposed last year, and they achieved more impressive results in image classification even surpassing CNN-based or ViT-based architectures. However, they cannot be transferred to dense prediction tasks (e.g., object detection and semantic segmentation) since their parameters are image sizes related and can not cope with images from various image sizes. Specifically, the global recep-
tive field is crucial in computer vision tasks, and they obtain it through matrix transposition and token-mixing projection such that the long-range dependencies are covered. However, this operation to mix tokens and the learned parameters correlate to the fixed input size, limiting the usability of dense prediction tasks.

To overcome the above limitation of fixed input sizes, more advanced MLP-based architectures [Lian et al., 2022; Wang et al., 2022a; Chen et al., 2022; Guo et al., 2022] were proposed to adapt arbitrary resolution last year. But they brought some new problems. Spatial shift [Lian et al., 2022; Wang et al., 2022a] operation is proposed to aggregate spatial information to make it feasible for arbitrary resolutions. But it only covers the local receptive field, which contradicts dense prediction tasks. CycleMLP [Chen et al., 2022] is friendly to dense prediction, but sample points in a cyclical style may lose important visual cues and lead to bad results, especially in dense prediction tasks that require dealing with small objects. Hire-MLP [Guo et al., 2022] captures global context by circularly shifting all tokens along spatial directions, but may damage the positional prior information.

Driven by these ideas, this paper explores how to design a vision MLP backbone not only to tackle arbitrary image sizes and scales but also to capture rich visual cues for various visual tasks. We propose Region-aware MLP (RaMLP), a vision MLP backbone for visual recognition and dense prediction. RaMLP mainly consists of a well-designed module, Region-aware Mixing (RaM), to capture local and global information in a region-aware manner. And it can adapt to arbitrary input sizes. First, inspired by recent researches [Diao et al., 2022] that find a simple spatial pooling can achieve competitive results against the attention module in transformers, we use a learnable pooling, to better capture local visual cues that are essential to the results, especially in dense prediction tasks. Second, we propose Dilated Fully-Connection (DFC) to aggregate these local visual cues to obtain global context. Third, we add a Region-aware layer to further adjust the spatial features, which can capture visual cues more robustly.

Our RaMLP achieves the best accuracy in ImageNet-1K image classification (see Figure 1) compared to state-of-the-art ViT-based, MLP-based, and CNN-based models with fewer parameters and FLOPs. Moreover, compared with state-of-the-art MLP-based models, Wave-MLP, our improvements (0.3% accuracy on the tiny scale, 0.4% on the small scale, and 0.5% on the base scale) are significant. Also, compared with the well-known ViT-based model, Swin Transformer, our improvements are 0.6%-1.6% accuracy using less computation.

Moreover, our RaMLP can be easily transferred to downstream dense prediction tasks and achieve great results. According to experimental results, our RaMLP outperforms state-of-the-art ViT-based, MLP-based, and CNN-based backbones on dense prediction tasks, including MSCOCO object detection, MS-COCO instance segmentation, and ADE20K semantic segmentation. In particular, our RaMLP outperforms previous state-of-the-art MLP-based backbones by a large margin (around 1.5% AAp and 1.0% mIoU improvements). It demonstrates the proposed RaM is effective for MLPs in dense prediction tasks.

The experimental results demonstrate not only the effectiveness of our model but the great potential of MLPs in both image classification and dense prediction. We believe this paper will raise more attention to MLPs for vision.

Our contributions can be summarized below:

- We introduce a vision MLP architecture named Region-aware MLP, which employs a well-designed module, Region-aware Mixing to capture visual dependence in a coarse-to-fine manner. It can cope with various image sizes and be transferred to dense prediction tasks easily.
- Our Region-aware Mixing can adaptively determine aggregation weights according to spatial features, which can capture spatial visual cues more robustly and lead to more robust spatial feature extraction.
- Extensive experiments demonstrate that RaMLP outperforms the state-of-the-art CNNs, ViTs, and MLPs in various vision tasks, including image classification, object detection, instance segmentation, and semantic segmentation.

2 Related Work

CNN-based Architectures. After AlexNet [Krizhevsky et al., 2012] won the 2012 ImageNet competition with an extremely great advantage, more and more CNN architectures were proposed. VGGNet [Simonyan and Zisserman, 2015] is a simple variant of Alexnet, by repeatedly stacking more convolutional layers. ResNet [He et al., 2016a; He et al., 2016b] explores the influence of depth. It even trains a 1001-layer network by an identity mapping branch. Inception models [Szegedy et al., 2015; Ioffe and Szegedy, 2015; Szegedy et al., 2016; Szegedy et al., 2017] design a series of multi-branch architectures, to indicate the importance of multi-scale information. These works provide efficient structure, and their variants are widely used in the succeeding works. Recently, researchers introduced Transformers to visual recognition and proposed Vision Transformers, which superseded CNNs on many visual tasks. ConvNeXt [Liu et al., 2022] discovers several key components to the performance and competes favorably with ViTs in terms of accuracy. However, ConvNeXt still inherits the weakness of CNN. The receptive field is far less than ViTs and MLPs, and sharing the same weights on spatial dimension also leads to a negative impact on extracting visual elements. Our RaMLP solves the above two problems at the same time.

Transformer-based Architectures. Due to the successful applications in natural language processing [Devlin et al., 2019; Brown et al., 2020], recent works, called Vision Transformers (ViTs) [Dosovitskiy et al., 2021; Touvron et al., 2021b], attempt to directly apply transformer to vision tasks such as image classification. They achieve comparable results with CNNs and even outperform using huge training data. However, directly applying self-attention to vision tasks leads to large computational costs, which is unacceptable for dense prediction tasks. Swin Transformer [Liu et al., 2021] introduces pyramid structure and non-overlapping window partitions to ViTs. Thus it has linear computational complexity with respect to input image size. Recently, researchers
also pointed out that ViTs and CNNs are complementary. CvT [Wu et al., 2021a] and LocalViT [Li et al., 2021] insert depthwise convolution into Multi-head Self-Attention or MLP module to enhance local context. CPVT [Chu et al., 2021b] also adds an extra depthwise convolution, to generate conditional position encoding dynamically by the local neighborhood of the input tokens. However, compared with CNNs and MLPs, the components in ViTs are not friendly to most of the existing hardware, which limits their application.

MLP-based Architectures. MLP-Mixer [Tolstikhin et al., 2021] and ResMLP [Touvron et al., 2021a] are proposed almost at the same time, which shows that multi-layer perceptrons can also attain good accuracy/complexity trade-offs on ImageNet. To reduce the computational complexity but still capture long-range dependencies, VIP [Hou et al., 2021] separately encodes the feature representations along the height and width dimensions and then aggregates the outputs in a mutually complementing manner. However, all these methods can only cope with fixed image size and are unfriendly to dense prediction tasks. Shift [Wang et al., 2022a] and AS-MLP [Lian et al., 2022] aggregate spatial information with spatial shift operation along spatial dimensions to make it flexible with various image sizes. CycleMLP [Chen et al., 2022] is friendly to dense prediction, but sampling points in a cyclical style may lose important visual cues and lead to bad results, especially in dense prediction tasks that require dealing with small objects. Hire-MLP [Guo et al., 2022] proposes the cross-region rearrangement to enable information communication between different regions by circularly shifting but may affect the positional prior. These models lack the ability to capture rich long-range dependencies, leading to unsatisfactory results for dense prediction in downstream tasks. Our RaMLP uses RaM to capture all visual dependencies in a coarse-to-fine manner and is seamlessly used for dense prediction.

3 Method
In this section, we first describe the overall architecture of RaMLP. Then we make a detailed introduction of the Region-aware Mixing (RaM) module, which is the key component of RaM block. Finally, we give brief configurations of the architecture variants.

3.1 Overall Architecture
An overview of the RaMLP architecture is presented in Figure 2, which illustrates the tiny version with $H \times W$ image. Followed by existing MLP-based architectures, we use a naive convolution layer for tokenizing input images and also three naive convolution layers for token merging across different RaM blocks.

The RaM blocks are the main components of our network, they are MLP-based architecture to enhance the representation of tokens before merging. We will introduce the details of RaMLP below.

3.2 Region-aware Mixing
The standard spatial FC used in MLP-Mixer [Tolstikhin et al., 2021] and ResMLP [Touvron et al., 2021a] computes all relations between tokens. The complexity is unacceptable, and FC weights are correlated to the number of tokens, which requires a fixed image scale and thus is infeasible for dense prediction. The spatial shift is a computation-free operation to overcome the above problems and thus is introduced by some recent MLP-based architectures [Lian et al., 2022; Wang et al., 2022a; Guo et al., 2022]. But it cannot model long-range visual dependencies well, which is critical in dense prediction. CycleMLP [Chen et al., 2022] is dense prediction friendly, but its cyclical sampling limits it to capture some visual cues, especially for small objects.

To overcome the above problems, we propose Region-aware Mixing (RaM) to capture visual dependence in a coarse-to-fine region-aware manner and adapt arbitrary input
sizes. As illustrated in Figure 2, a RaM block consists of a Learnable Pooling (LP), a Dilated Fully-Connection (DFC), and five Channel Fully-Connection (CFC) layers. Three layer-norm layers and two residual connections are applied for each block. The last layer-norm layer, the last two CFC layers, and the last residual connection constitute a channel MLP module. Besides, same as Twins [Chu et al., 2021a], we also introduce Conditional Positional Encoding (CPE), to handle the local positional information better. With these modules, RaM blocks are computed as:

\[ t^l = x^{l-1} + \text{CPE}(x^{l-1}), \]  
\[ u^l = \text{LN}(u^l), \]  
\[ v^l = \text{CFC}(u^l), \]  
\[ w^l = \text{LP}(w^l), \]  
\[ y^l = \text{DFC}(\text{LN}(w^l)), \]  
\[ z^l = \text{CFC}(u^l), \]  
\[ s^l = t^l + \text{CFC}(y^l \cdot z^l), \]  
\[ x^l = s^l + \text{CFC}(\text{CFC}(\text{LN}(s^l)))) \]

where LN(·) refers to layer normalization, and \( l, u^l, v^l, w^l, y^l, z^l, s^l, x^l \) are the outputs of these operations. The CPE is implemented as a simple depthwise convolution, which is widely used in previous works [Chu et al., 2021a; Dong et al., 2022] for its compatibility with the arbitrary size of the input.

Learnable Pooling. Inspired by recent researches [Diao et al., 2022] that find a simple spatial pooling can achieve competitive results against the attention module in transformers, we also use a variant of pooling to better capture local visual cues that are essential to the results. For every spatial positional feature in spatial pooling, we assign learnable weight to better aggregate local visual cues. Actually, we could find it is very similar to a depthwise convolution. Thus, our Learnable Pooling is implemented as a simple depthwise convolution.

Dilated Fully-Connection. Dilated Fully-Connection (DFC) is a novel module to model long-range visual dependencies. As shown in Figure 3, we first reshape the input in dilated manner, to obtain dilated feature. It could be partitioned into several sparse global regions, and every region samples the point all over the input feature map. Specifically, given the input feature \( F \in \mathbb{R}^{C \times H \times W} \), we partition it into \( \frac{W}{S} \times \frac{H}{S} \) non-overlapping regions with a fixed region size of \( S \times S \) in a dilated manner, to produce dilated feature \( F' \). Then, we perform spatial FC for every region, to obtain \( \frac{W}{S} \times \frac{H}{S} \) global enhance feature \( E' \) with a fixed region size of \( \frac{S}{S} \times \frac{S}{S} \). At last, we use a inverted dilated reshape to restore the position of features, to obtain the final output.

Region-aware Layer. Features after LP and DFC could capture all visual cues, but too much spatial information may introduce noises and easily lead to over-fitting. To solve the above issues, we add a Region-aware layer to further adjust the spatial importance of the features, which can capture visual cues more robustly. Specifically, we perform channel FC for input features and then do Hadamard Product between it and the output of DFC \( O \). Thus, we adaptively determine aggregation over the whole spatial dimension and produce more robust regional features.

3.3 Architecture Variants

We build three models, called RaMLP-T (Tiny), RaMLP-S (Small), and RaMLP-B, to have the model size and computation complexity similar to Hire-MLP and Wave-MLP. We show the detailed configurations of all variants in Table 1.

### 4 Experiments

In this section, we first examine RaMLP by conducting experiments on ImageNet-1K [Deng et al., 2009] image classification, and then for dense prediction tasks, including MS-COCO [Lin et al., 2014] object detection, MS-COCO instance segmentation, and ADE20K [Zhou et al., 2019] semantic segmentation.

#### 4.1 ImageNet-1K Classification

**Settings.** We train our models on the ImageNet-1K [Deng et al., 2009] dataset from scratch, which contains 1.2M training images and 50K validation images evenly spreading 1,000 categories. We report the top-1 accuracy on the validation set following the standard practice in this community. For fair comparisons, our training strategy is mostly adopted from CycleMLP, including RandAugment, Mixup, Cutmix, random erasing, and stochastic depth. AdamW and cosine learning rate schedules with the initial value of \( 1 \times 10^{-3} \) are
<table>
<thead>
<tr>
<th>Output stride</th>
<th>Layer</th>
<th>RaMLP-T</th>
<th>RaMLP-S</th>
<th>RaMLP-B</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Patch Merging</td>
<td>$P_1 = 4$</td>
<td>$P_1 = 4$</td>
<td>$P_1 = 4$</td>
</tr>
<tr>
<td></td>
<td>RaM block</td>
<td>$K_1 = 5$</td>
<td>$K_1 = 5$</td>
<td>$K_1 = 5$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$S_1 = 8$</td>
<td>$S_1 = 8$</td>
<td>$S_1 = 8$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$M_1 = 4$</td>
<td>$M_1 = 4$</td>
<td>$M_1 = 3$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$E_1 = 3$</td>
<td>$E_1 = 3$</td>
<td>$E_1 = 3$</td>
</tr>
<tr>
<td>8</td>
<td>Patch Merging</td>
<td>$P_2 = 2$</td>
<td>$P_2 = 2$</td>
<td>$P_2 = 2$</td>
</tr>
<tr>
<td></td>
<td>RaM block</td>
<td>$K_2 = 5$</td>
<td>$K_2 = 5$</td>
<td>$K_2 = 5$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$S_2 = 4$</td>
<td>$S_2 = 4$</td>
<td>$S_2 = 4$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$M_2 = 4$</td>
<td>$M_2 = 4$</td>
<td>$M_2 = 3$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$E_2 = 3$</td>
<td>$E_2 = 3$</td>
<td>$E_2 = 3$</td>
</tr>
<tr>
<td>16</td>
<td>Patch Merging</td>
<td>$P_3 = 2$</td>
<td>$P_3 = 2$</td>
<td>$P_3 = 2$</td>
</tr>
<tr>
<td></td>
<td>RaM block</td>
<td>$K_3 = 5$</td>
<td>$K_3 = 5$</td>
<td>$K_3 = 5$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$S_3 = 2$</td>
<td>$S_3 = 2$</td>
<td>$S_3 = 5$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$M_3 = 3$</td>
<td>$M_3 = 3$</td>
<td>$M_3 = 2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$E_3 = 2$</td>
<td>$E_3 = 2$</td>
<td>$E_3 = 2$</td>
</tr>
<tr>
<td>32</td>
<td>Patch Merging</td>
<td>$P_4 = 2$</td>
<td>$P_4 = 2$</td>
<td>$P_4 = 2$</td>
</tr>
<tr>
<td></td>
<td>RaM block</td>
<td>$K_4 = 5$</td>
<td>$K_4 = 5$</td>
<td>$K_4 = 5$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$S_4 = 1$</td>
<td>$S_4 = 1$</td>
<td>$S_4 = 1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$M_4 = 3$</td>
<td>$M_4 = 3$</td>
<td>$M_4 = 2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$E_4 = 2$</td>
<td>$E_4 = 2$</td>
<td>$E_4 = 2$</td>
</tr>
<tr>
<td>-</td>
<td>Params</td>
<td>25M</td>
<td>38M</td>
<td>58M</td>
</tr>
<tr>
<td>-</td>
<td>FLOPs</td>
<td>4.2G</td>
<td>7.8G</td>
<td>12.0G</td>
</tr>
</tbody>
</table>

Table 1: Overall architecture of RaMLP with three different levels of complexities. As shown in Section 3.3, $P_i$ denotes the spatial reduction factor, $C_i$ denotes the channel number, $K_i$ denotes the kernel size for the LP, $S_i$ denotes the region size for the DFC, $E_i$ denotes the expansion ratio for channel FC in RaM, $M_i$ denotes the expansion ratio for channel FC in channel MLP. FLOPs are evaluated on 224 × 224 resolution.

Comparison with MLP-based Models. We compare our RaMLP with MLP-based models proposed in recent two years, and we show the results in Table 2. First, we get a breakthrough in image classification. Hire-MLP-B, the previous state-of-the-art model, achieves 83.8% accuracy with 13.4G FLOPs and 96M parameters. In comparison, our RaMLP achieves the same accuracy using much less computation (7.8G FLOPs) and much fewer parameters (38M). Second, our RaMLP achieves the best results in three different computation scales (0-4G FLOPs, 4-8G FLOPs, and 8-16G FLOPs), demonstrating that our RaMLP performs well on different computation resources. Besides, Wave-MLP is the most related model designed for dense prediction tasks. Our RaMLP surpasses it by 0.2% accuracy with only 76% FLOPs (seeing the results of Wave-MLP-B, and our RaMLP-S). It demonstrates the effectiveness of the proposed modules to capture visual dependencies in a coarse-to-fine manner with region-aware modeling.

Comparison with SOTA Models. We compare our RaMLP with state-of-the-art models, including CNN-based models, ViT-based models, and MLP-based models, and we show the results in Table 3. First, it is encouraging to us that our RaMLP surpasses SOTAs in all four computation scales (4-6GFLOPs, 6-10GFLOPs, and 10GFLOPs). It demonstrates the great potential of MLP-based models, and it is promising to do more research on MLPs. Second, our RaMLP achieves more improvements in accuracy on a tiny scale, which means our model is suitable for low computation capability scenarios. Moreover, our method gets better results in accuracy/FLOPs trade-off compared with the state-of-the-art transformer-based models across different computation scales. It demonstrates that well-designed MLP modules may be more suitable than self-attention modules in computer vision tasks. In this experiment, we demonstrated the superiority of MLP-based models against CNN-based models and ViT-based models in image classification, the most common computer vision task. And thus, we think our research made a solid contribution to MLP research and can attract many followers to explore MLPs and bring more exciting results.

4.2 Ablation Study

In this section, we utilize RaMLP-T to verify the effectiveness of the proposed components by conducting extensive ablation studies.

Study on Region Size. We evaluate the effectiveness of adjusting the region size in Table 4 and find that increasing the region size could improve the performance. Small region size decreases the density of the sampling point and increases the loss of spatial information. Too small a region count even leads to non-convergence.

Study on Effectiveness of Different Components. As shown in Table 5, we set a RaMLP without RaM as a base-
line, then, we add LP, DFC, and Ra layer to verify the effectiveness. All these components have obvious effects.

### 4.3 Object Detection and Instance Segmentation

#### Settings

We conduct object detection experiments with RetinaNet [Lin et al., 2017], and instance segmentation experiments with Mask R-CNN [He et al., 2017] on COCO [Lin et al., 2014] dataset by following the experimental settings of CycleMLP [Chen et al., 2022].

#### Results on Object Detection

Object detection is a typical dense prediction task. We separate the models into three scales according to the number of parameters and show the results in Table 6. First, our RaMLP achieves the best results across three scales using nearly the least parameters, which demonstrates the effectiveness and efficiency of our RaMLP in dense prediction tasks. And interestingly, our improvement in the first scale is the hugest, which demonstrates that our model design is feasible for low computation resource scenarios. It is worth noting that our RaMLP outperforms the ResNet-50 and ResNet-101, the two most widely used CNN-based backbones, by a large margin (around 7% APb) with fewer parameters. Moreover, our model outperforms previous state-of-the-art MLPs by a large margin. Compared with Hire-MLP, we get +1.9%, +1.2%, and +1.5% accuracy, respectively, using a similar number of parameters. It demonstrates the effectiveness of our proposed RaP and RaDFC for dense prediction.

#### Results on Instance Segmentation

Instance segmentation is a more challenging dense prediction task against object detection. Following the evaluation in object detection, we separate the models into three scales and show the results in Table 7. First, compared with object detection, our improvements to the state-of-the-art is more obvious (more than 1% APb), which demonstrates the effectiveness of our RaMLP in dense prediction tasks. And the dense prediction task is more challenging, more improvements RaMLP can achieve. Second, other phenomena in object detection also occur, which demonstrates the generality of RaMLP to various dense predictions.

#### Settings

Following the PVT [Wang et al., 2021], we evaluate the potential of RaMLP on the challenging semantic seg-
Table 5: The impacts of the components.

<table>
<thead>
<tr>
<th>Models</th>
<th>Arch.</th>
<th>Params</th>
<th>(A_P^{50})</th>
<th>(A_P^{75})</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet50</td>
<td>CNN</td>
<td>38</td>
<td>36</td>
<td>53.3</td>
</tr>
<tr>
<td>PVT-S</td>
<td>ViT</td>
<td>34</td>
<td>40.4</td>
<td>61.3</td>
</tr>
<tr>
<td>Swin-T</td>
<td>ViT</td>
<td>29</td>
<td>41.5</td>
<td>62.1</td>
</tr>
<tr>
<td>CycleMLP-B2</td>
<td>MLP</td>
<td>37</td>
<td>40.9</td>
<td>61.8</td>
</tr>
<tr>
<td>Hire-MLP-S</td>
<td>MLP</td>
<td>43</td>
<td>41.7</td>
<td>-</td>
</tr>
<tr>
<td>RaMLP-T</td>
<td>MLP</td>
<td>35</td>
<td>43.6</td>
<td>64.9</td>
</tr>
<tr>
<td>ResNet101</td>
<td>CNN</td>
<td>57</td>
<td>38.5</td>
<td>57.8</td>
</tr>
<tr>
<td>PVT-M</td>
<td>ViT</td>
<td>54</td>
<td>41.9</td>
<td>63.1</td>
</tr>
<tr>
<td>Swin-S</td>
<td>ViT</td>
<td>60</td>
<td>44.5</td>
<td>65.7</td>
</tr>
<tr>
<td>CycleMLP-B3</td>
<td>MLP</td>
<td>48</td>
<td>42.5</td>
<td>63.2</td>
</tr>
<tr>
<td>Hire-MLP-B</td>
<td>MLP</td>
<td>68</td>
<td>44.3</td>
<td>-</td>
</tr>
<tr>
<td>RaMLP-S</td>
<td>MLP</td>
<td>49</td>
<td>45.5</td>
<td>66.7</td>
</tr>
<tr>
<td>PVT-L</td>
<td>ViT</td>
<td>71</td>
<td>42.6</td>
<td>63.7</td>
</tr>
<tr>
<td>Swin-B</td>
<td>ViT</td>
<td>98</td>
<td>44.7</td>
<td>65.9</td>
</tr>
<tr>
<td>CycleMLP-B4</td>
<td>MLP</td>
<td>62</td>
<td>43.2</td>
<td>63.9</td>
</tr>
<tr>
<td>Hire-MLP-L</td>
<td>MLP</td>
<td>106</td>
<td>44.9</td>
<td>-</td>
</tr>
<tr>
<td>RaMLP-B</td>
<td>MLP</td>
<td>70</td>
<td>46.4</td>
<td>67.7</td>
</tr>
</tbody>
</table>

Table 6: Object detection on COCO val2017 with RetinaNet.

Results. Semantic segmentation is also one of the most common dense prediction tasks. We separate the models into three scales according to FLOPs and show the results in Table 8. First, impressively, our RaMLP outperforms previous SOTAs by a large margin (0.9%, 1.3%, and 1.5% improvements on three scales, respectively). It is interesting that Hire-MLP, the previous state-of-the-art MLP-based model, does not show significant superiority against transformer-based models, but our RaMLP does. Hire-MLP uses hierarchical rearrangement to capture spatial information but may lose important visual cues in semantic segmentation, while RaMLP can capture rich visual cues for various visual tasks. Second, our RaMLP achieves the best results using the least computation and parameters in the second and third scales.

5 Conclusion

We introduce a new MLP-based architecture named Region-aware MLP (RaMLP) with a well-designed module, Region-aware Mixing (RaM), to capture visual dependence in a coarse-to-fine region-aware manner. It can adaptively determine aggregation weights according to regions and inputs, to extract regional features more robustly. It also can cope with various image sizes and be transferred to dense prediction tasks easily. The results on image classification, object detection, instance segmentation, and semantic segmentation, show that our RaMLP outperforms the SOTAs.

Acknowledgments

Shenqi Lai and Xi Du contribute equally. Kaipeng Zhang is the corresponding author. This work is partially supported by the National Key R&D Program of China(No.2022ZD0160100) and in part by Shanghai Committee of Science and Technology (Grant No. 21DZ1100100).
References


[Szegedy et al., 2017] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4,


