Receptive Fields As Experts in Convolutional Neural Architectures

Dongze Lian [1] **Weihao Yu** [1] **Xinchao Wang** [1]

Abstract

The size of spatial receptive fields, from the early 3×3 convolutions in VGGNet to the recent 7×7 convolutions in ConvNeXt, has always played a critical role in architecture design. In this paper, we propose a Mixture of Receptive Fields (MoRF) instead of using a single receptive field. MoRF contains the combinations of multiple receptive fields with different sizes, *e.g.*, convolutions with different kernel sizes, which can be regarded as experts. Such an approach serves two functions: one is to select the appropriate receptive field according to the input, and the other is to expand the network capacity. Furthermore, we also introduce two types of routing mechanisms, hard routing and soft routing to automatically select the appropriate receptive field experts. In the inference stage, the selected receptive field experts are merged via re-parameterization to maintain a similar inference speed compared to the single receptive field. To demonstrate the effectiveness of MoRF, we integrate the MoRF concept into multiple architectures, *e.g.*, ResNet and ConvNeXt. Extensive experiments show that our approach outperforms the baselines in image classification, object detection, and segmentation tasks without significantly increasing the inference time.

1. Introduction

Convolutional Neural Networks (CNNs) have established themselves as the cornerstone of various computer vision tasks, with a series of CNN-based architectures (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; He et al., 2016) being proposed in recent years. Transformers, due to their superior performance in natural language processing (NLP) (Vaswani et al., 2017) and weakly dependent on the inductive bias, have gradually been introduced to vision tasks as an alternative to CNNs. Despite this trend, different from the NLP, where the knowledge has extremely high information density, visual images are filled with amounts of redundant information. Thus, it remains arguable whether convolutions can be completely replaced by the transformer in vision architectures.

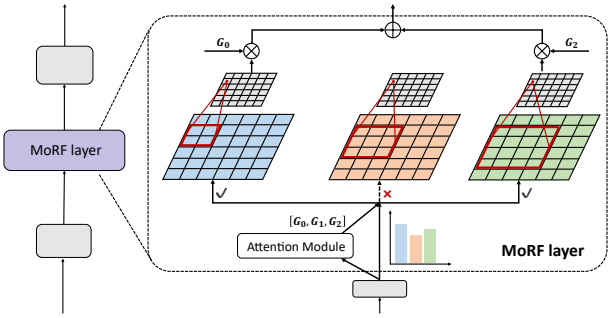

Figure 1. An example of a Mixture of Receptive fields (MoRF) layer involves selecting the first and third receptive fields via hard routing for interaction, and the final output is a weighted sum of the output of the two selected experts.

One critical issue in the design of vision backbones is the selection of the receptive field. VGGNet (Simonyan & Zisserman, 2015) popularizes the use of a 3×3 convolutional layer with a local receptive field, and subsequent architectures have followed this design by stacking more convolutional layers to increase the receptive field size. Dosovitskiy *et al.* (Dosovitskiy et al., 2021) introduce a paradigm shift with the Vision Transformer (ViT), which uses multi-head self-attention to obtain a global spatial receptive field. This design is later modified by Liu *et al.* (Liu et al., 2021b) with the Swin Transformer, which restricts the range of self-attention to a local window to obtain local receptive fields. Other works (Xiao et al., 2021; Ding et al., 2022; Liu et al., 2022a) have proposed alternative approaches, such as combining convolutions and self-attention, or employing large kernels instead of self-attention. Despite the differences in these approaches, all of them highlight the crucial role of the receptive field in visual understanding. However, designing manually an appropriate receptive field size is challenging. A large receptive field may result in optimization difficulty and bring more redundant computation costs. Consequently, a key question that arises is how to design a module that

[1]Department of Electrical and Computer Engineering, National University of Singapore, Singapore. Correspondence to: Xinchao Wang <xinchao@nus.edu.sg>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

covers the appropriate receptive field in vision architectures.

To tackle this issue, we propose a concept of a Mixture of Receptive Fields (MoRF), where the different receptive fields are treated as experts, and the model is able to select the appropriate receptive field expert for each input image. We illustrate an example of a MoRF layer in Figure 1. Given an input, the gating network chooses the appropriate receptive field, and the output results are weighted summed. Such an approach offers two benefits: first, it enables the selection of optimal receptive fields according to the input, and secondly, it facilitates the expansion of the model capacity while minimally affecting the speed of inference. Such an approach aligns with a philosophy similar to Mixture of Experts (MoEs) (Jacobs et al., 1991; Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2021; Du et al., 2022; Zoph et al., 2022). However, the core motivation of MoEs is the model scaling, whereas our focus is to select the appropriate receptive field according to the input.

Another issue is how the network selects the appropriate receptive field from the available experts in MoRF. We first employ an attention module to generate attention for different receptive field branches based on the input. For the selection of the appropriate receptive field, we introduce two solutions: hard routing and soft routing. Hard routing involves selecting the top-k receptive field experts and discarding the others, which is similar to the MoE approach. In contrast, soft routing uses weights generated by the attention module to perform a weighted sum across all receptive field experts, which yields better performance than hard routing but incurs additional computational costs. To circumvent this computational burden during inference, we implement a kernel fusion process through a re-parameterization strategy (Zagoruyko & Komodakis, 2017; Ding et al., 2021b) to merge all receptive field experts into a single convolutional kernel. This enables the model to maintain a similar inference speed compared to using a single receptive field.

To evaluate the effectiveness of our MoRF, we integrate the MoRF into multiple architectures, *e.g.*, ResNet, and ConvNeXt. Extensive experiments show that our approach outperforms the baselines in image classification, object detection, and segmentation tasks without significantly increasing the inference time. We summarize our contributions as the three folds.

- We introduce a novel concept of a Mixture of Receptive Fields (MoRF) in vision architectures, which allows for the dynamic selection of appropriate receptive field experts based on input images.
- We introduce hard routing and soft routing to automatically select the receptive field experts. In the inference stage, the selected receptive field experts are merged via re-parameterization to maintain a similar inference speed compared to the single receptive field.
- We systematically evaluate our MoRF approach on extensive experiments across multiple tasks, including image classification, object detection, and semantic segmentation, demonstrating its effectiveness compared to existing baseline models.

2. Related Work

2.1. Vision Architectures

Prior to the advent of Transformers, CNN-based architectures are widely adopted as the de-facto standard for computer vision, and amounts of CNNs and their variants (Simonyan & Zisserman, 2015; He et al., 2016; Huang et al., 2017b; Yu & Koltun, 2016; Dai et al., 2017; Yu et al., 2024b) are proposed for image classification, object detection, and semantic segmentation. Dosovitskiy *et al.* (Dosovitskiy et al., 2021) introduce a transformer from the NLP domain to vision and propose a ViT, where an image is divided into 16×16 patches and fed into a transformer as tokens for self-attention, achieving excellent performance. After that, a large number of transformer variants (Touvron et al., 2021b; Zhou et al., 2021; Touvron et al., 2021c; Chen et al., 2021; Chu et al., 2021; Graham et al., 2021; Wang et al., 2021; Liu et al., 2021b; Zhang et al., 2021; Liu et al., 2022b; Lian et al., 2022b; Ren et al., 2022; 2023) are introduced. For instance, Touvron *et al.* (Touvron et al., 2021b) propose DeiT with distillation and introduce an applicable recipe. Liu *et al.* (Liu et al., 2021b; 2022b) divide patches to perform self-attention within the local window, and design a hierarchical architecture for various vision tasks. In addition to CNNs and Transformers, some pure MLP-based vision architectures (Melas-Kyriazi, 2021; Touvron et al., 2021a; Liu et al., 2021a; Hou et al., 2021; Lian et al., 2022a; Chen et al., 2022; Yu et al., 2021; Ding et al., 2021a; Zheng et al., 2022) are also designed. Tolstikhin *et al.* (Tolstikhin et al., 2021) introduce MLP-Mixer, which consists of token-mixing MLPs and channel-mixing MLPs for the interaction of the spatial and channel information, respectively. Lian *et al.* (Lian et al., 2022a) propose to axially shift features along horizontal and vertical directions and introduce an AS-MLP architecture. Furthermore, some researchers rethink the necessity of self-attention in Transformers. MetaFormer (Yu et al., 2022; 2024a) replacing self-attention with a pooling layer and ConvNeXt replacing self-attention with a 7×7 convolution, both of which have shown results as good as or better than traditional Transformers. Additionally, some works (Ding et al., 2022; Liu et al., 2022a) also introduce large kernels for vision backbones. In this work, we incorporate the concept of MoE (Jacobs et al., 1991; Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2021; Du et al., 2022; Zoph et al., 2022) into vision architectures and propose a MoRF. It is able to automatically select the appro-

priate receptive field based on the input while expanding the model capacity without increasing computational cost and inference time.

2.2. Receptive Fields in Neural Networks

The earliest known research on receptive fields can be traced back to LeNet (LeCun et al., 1998), which utilizes 5×5 convolutional kernels for the purpose of recognizing handwritten digits. Krizhevsky *et al.* (Krizhevsky et al., 2012) introduce 11×11 convolutional kernels to cover larger receptive fields. However, larger receptive fields result in higher computational complexity. Therefore, VGGNet (Simonyan & Zisserman, 2015) opts to use 3×3 convolutional kernels and achieve larger receptive fields by deepening the network. Afterward, dilated convolution (Yu & Koltun, 2016) and deformable convolution (Dai et al., 2017; Gao et al., 2019) are introduced to change the shape of the receptive field. With the advent of transformer models (Dosovitskiy et al., 2021; Touvron et al., 2021b; Liu et al., 2021b; Wang et al., 2021) in computer vision, self-attention has been employed to obtain global spatial receptive fields. However, recent studies (Xiao et al., 2021; Zhou et al., 2021; Yuan et al., 2021a) have shown that not all layers and images require global receptive fields. For instance, Xiao *et al.* (Xiao et al., 2021) demonstrate that using convolutions in the early layer is more effective and helps transformers see better. Swin Transformer (Liu et al., 2021b; 2022b), on the other hand, employs local receptive fields via a local window and performs self-attention within the window, achieving superior results. Additionally, some works (Liu et al., 2022c; Ding et al., 2022; Liu et al., 2022a) use large kernels as a replacement for self-attention in vision backbones. However, most of these methods are designed by manually adjusting the receptive field size. Motivated by the mixture of experts (MoE) (Jacobs et al., 1991; Shazeer et al., 2017; Lepikhin et al., 2020; Fedus et al., 2021; Du et al., 2022; Zoph et al., 2022), we view receptive fields as experts, where the specific expert will be activated based on the input image.

2.3. Dynamic Neural Networks

Dynamic neural networks are introduced to adaptively recalibrate either their topology structure (Huang et al., 2017a; Wang et al., 2018; Mullapudi et al., 2018; Fedus et al., 2021; Zoph et al., 2022) or model weights (Yang et al., 2019; Chen et al., 2020; Li et al., 2021; 2022) in response to varying inputs, thereby improving the accuracy or reducing the computational cost. Some works (Huang et al., 2017a; Li et al., 2019a) advocate for the implementation of early exiting for the reduction of parameters depending on how difficult the sample is. Concurrently, the Mixture of Experts (MoE) framework (Fedus et al., 2021; Lepikhin et al., 2020; Du et al., 2022; Hazimeh et al., 2021) has been leveraged to scale the model capacity while preserving the inference speed on par with non-MoE architectures. Some dynamic weight generation networks (Yang et al., 2019; Chen et al., 2020) are capable of learning different weights based on the input, which are then merged together during the inference process. Different from these approaches, our investigation delves into the impact of diverse receptive fields on model performance. Additionally, the inference speed of the entire network can be maintained via model re-parameterization (Zagoruyko & Komodakis, 2017; Ding et al., 2021b).

3. Approach

3.1. Mixture of Receptive Fields

MoRF represents a comprehensive concept for the integration of receptive fields to automatically identify and select receptive fields that are optimal for given inputs. Convolution is recognized as a pivotal and core operation for capturing spatial receptive fields in modern neural networks and is extensively employed. Therefore, we choose Convolution as the key operation within our MoRF paradigm.

MoRF with Convolution. Given an input $X \in \mathbb{R}^{H\times W\times C}$ where H, W, and C are the height, width, and the number of input channels, respectively, and a receptive field region $\mathcal{R}$ with a kernel size of $K \times K$, the output $Y_{i,j,:}$ can be computed using the convolution operation as follows

$$Y_{i,j,:} = \sum_{c=0}^{C} \sum_{(a,b)\in\mathcal{R}} X_{i+a,j+b,c} W_{a,b,c,:}, \tag{1}$$

where $W \in \mathbb{R}^{K\times K\times C\times C'}$ is the learnable weight and C' is the number of output channels. i, j are the index of the spatial position. The receptive field can be adjusted by modifying the size of the K, which is usually determined through manual design, such as using 3×3 in VGGNet (Simonyan & Zisserman, 2015) and ResNet (He et al., 2016) or 7×7 in ConvNeXt (Liu et al., 2022c).

In the MoRF framework, our objective is to incorporate multiple Convolution experts with different kernel sizes into a unified architecture, effectively replacing a single convolutional kernel. As depicted in Figure 1, we employ n Convolutions with distinct kernel sizes, where the respective receptive fields are denoted as $\{\mathcal{R}_1, ..., \mathcal{R}_n\}$. Given an input X, a routing network is first utilized to generate attention logits, following which the appropriate experts are selected to perform convolutional operations on the input. The results are then aggregated through a weighted sum based on routing probabilities. Formally, the output Y is computed using the following equation

$$Y_{i,j,:} = \sum_{r=1}^{n} A(X)_r \sum_{c=0}^{C} \sum_{(a,b)\in\mathcal{R}_r} X_{i+a,j+b,c} W^r_{a,b,c,:}, \tag{2}$$

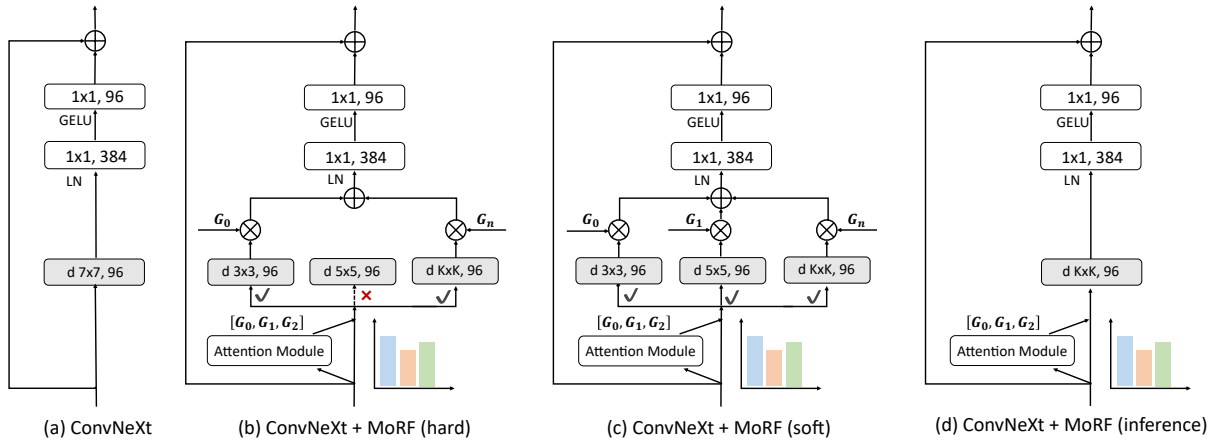

Figure 2. (a) shows the original ConvNeXt block (Liu et al., 2022c). (b) shows our integration of MoRF into ConvNeXt with the hard routing, where we illustrate the mixture of three receptive fields as an example. (c) shows the integration of MoRF into ConvNeXt with the soft routing. (d) shows the inference architecture via re-parameterization, where the weights of different convolutions are merged into one kernel.

where $\mathcal{R}_r$ represents the receptive field region of the r-th receptive field expert, and $A(X)$ is the routing network to generate the attention logits.

Discussions. As highlighted in Sec. 1, the MoRF technique serves two purposes: i) dynamically selecting the optimal receptive field for the input; ii) expanding the capacity of the model. This distinction fundamentally differentiates our methodology from other existing approaches. Several methods, such as Mixture of Experts (MoE) (Fedus et al., 2021; Lepikhin et al., 2020; Du et al., 2022), employ multiple experts to construct Feed Forward Networks (FFNs), with their primary motivation being the scaling of networks. However, our approach additionally focuses on the dynamic selection of receptive fields, different from the primary intent of these methods. Although Neural Architecture Search (NAS) (Pham et al., 2018; Liu et al., 2018; Guo et al., 2020; Lian et al., 2020; Wan et al., 2020) dynamically searches for suitable operations (including convolutions with different kernel sizes) to obtain receptive fields, these methods often require an additional search phase and cannot automatically adjust the receptive field based on the input after the search phase is completed. In contrast, our MoRF can dynamically select the optimal receptive field for the input during inference. Other approaches, *e.g.*, dynamic convolution (Yang et al., 2019; Chen et al., 2020; Ma et al., 2020), generate weights of the convolution based on the input, but they typically consider convolutional kernels with identical receptive fields. Our approach employs a mixture of multiple receptive fields, thereby extending beyond the scope of existing dynamic convolution strategies. Some similar methods, *e.g.*, SKNet (Li et al., 2019b) and RF-Next (Gao et al., 2022), also explore receptive fields. However, compared to SKNet, we employ more receptive fields in vision architectures via re-parameterization trick instead of only two kernels. Compared to RF-Next, our MoRF is input-dependent and can choose the appropriate receptive field for each image. More importantly, we borrow the idea of mixture of experts to expand the receptive field size, and our network achieves superior performance, which shows the effectiveness of our method.

3.2. Model Instantiation

We primarily instantiate ConvNeXt as our model architecture for the visualization and detailed introduction of our method. MoRF is also integrated into other networks, *e.g.*, ResNet (He et al., 2016), by replacing the single receptive field convolutions with a mixture of multiple receptive field convolutions. Further results are listed in Sec. 4.

A Revisit of ConvNeXt. A complete ConvNeXt architecture includes: i) a patchy stem layer with a 4×4 non-overlapping convolution; ii) a four-stage feature extraction process with progressively increasing channels; iii) a downsampling component for connecting different stages; iv) a classification head for image classification. The core of each stage is the ConvNeXt block, composed of a 7×7 depthwise convolution (Howard et al., 2017; Chollet, 2017), a layer normalization (Ba et al., 2016), a GeLU activation function (Hendrycks & Gimpel, 2016), and two 1×1 convolutions, as shown in Figure 2(a).

ConvNeXt + MoRF. We integrate MoRF into the standard ConvNeXt architecture, incorporating three primary modifications as follows: i) The convolution replacement of a single kernel (7×7) with multiple kernel sizes, resulting in several branches with different kernel sizes; ii) Given the presence of multiple branches, we utilize an attention

module to generate attention for each branch; iii) The implementation of a routing operation is used to select the appropriate receptive field branch, where two routing mechanisms (hard routing and soft routing) are introduced.

Attention Module. The primary function of the attention module $A(X)$ is to generate attention for different receptive field branches based on the input X, thereby facilitating the selection of the appropriate receptive field. Given an input X, the module initially employs the Global Average Pooling (GAP) to transform X into a feature vector. This is followed by two Fully Connected (FC) layers, interconnected via a GELU activation function.

Hard Routing. Following existing MoE methods (Fedus et al., 2021; Lepikhin et al., 2020; Du et al., 2022), we can employ the hard routing strategy to select the receptive field branch. Assuming the output of the last FC layer in the attention module as $x \in \mathbb{R}^{N\times d}$ where N denotes the batch size and d denotes the dimension, and the routing network R and n expert networks. The output is computed as the weighted sum of multiple expert networks, with the weights determined by the gating network. Consequently, the output Y is

$$Y = \sum_{r=1}^{n} R(x)_r [W_r * X + B_r], \tag{3}$$

where W_r represents its corresponding kernel weight, and B is the bias term. The operator $*$ signifies the convolutional operation. The routing network R is usually chosen to be a noisy gating network as follows

$$R(x) = \text{Softmax}(\text{topk}(H(x), k)), \tag{4}$$

where the number of activated experts is determined by k. $H(x)$ is obtained using a trainable gating weight matrix $W_g \in \mathbb{R}^{d\times n}$ and a noise weight matrix $W_\sigma \in \mathbb{R}^{d\times n}$ as follows

$$H(x)_r = (x \cdot W_g)_r + \sigma \cdot \text{Softplus}((x \cdot W_\sigma)_r), \tag{5}$$

where $\sigma \sim \mathcal{N}(0, 1)$ represents the Gaussian noise. At the inference phase, we remove this noise term. Hard routing exclusively selects the top-k receptive field branches for convolution and weighted sum, concurrently setting the remaining branches to zero, as depicted in Figure 2(b).

Soft Routing. A hard routing strategy that leverages the top-k experts may result in overlooking the potential impacts of non-top-k experts. Therefore, we also propose a soft routing strategy by incorporating a comprehensive set of receptive fields, enabling a more flexible and adaptive feature extraction mechanism. This is accomplished by performing a weighted summation of the outputs from all the different receptive field convolutions, as illustrated in Figure 2(c). Specifically, we can reformulate Eq (3) into the following form

$$Y = \sum_{r=1}^{n} \alpha_r [W_r * X + B_r], \tag{6}$$

where α_r denotes the weighting coefficient for the r-th convolution expert.

While this soft routing technique provides a more representative feature map, it is computationally intensive as it requires processing across all convolutional experts. To circumvent this computational burden during inference, we implement a kernel fusion process through a re-parameterization strategy, as shown in Figure 2(d). This consolidation effectively merges all individual kernels into a single convolutional kernel. The re-parameterized convolutional operation thus maintains similar inference throughput compared to conventional single-kernel convolutions, while leveraging the adaptive benefits of the soft routing. The re-parameterization process is as follows

$$\begin{aligned} Y &= \sum_{r=1}^{n} \alpha_r [W_r * X + B_r] \\ &= \sum_{r=1}^{n} \alpha_r W_r * X + \sum_{r=1}^{n} \alpha_r B_r \\ &= W * X + B, \end{aligned} \tag{7}$$

where W and B are the merged convolutional weights and biases, respectively. With this re-parameterization, the model preserves computational efficiency without sacrificing the richness of the feature maps from different receptive field experts, enabling real-time inference with enhanced representational capabilities.

4. Experiments

4.1. ImageNet Classification

Settings. To evaluate the effectiveness of our proposed MoRF, we select a wide range of model sizes to conduct experiments with ResNet-18 (He et al., 2016), ResNet-50 (He et al., 2016), and ConvNeXt (Liu et al., 2022c) on the ImageNet-1K dataset (Deng et al., 2009). We follow the experimental settings of ConvNeXt (Liu et al., 2022c), and train our model for 300 epochs with the first 20 epochs used for warm-up. The initial learning rate is set to 0.001 with cosine decay and we use a batch size of 1024. We employ the AdamW (Loshchilov & Hutter, 2019) to optimizer our networks. The more specific details and the hyperparameters can be found in the appendix.

Results. Table 1 presents the results of image classification on ImageNet-1K. We compare our method with existing transformers-based and CNN-based architectures across a variety of model scales and the results outperform the

Model	Input Resolution	Top-1 (%)	Params.	Throughput (imgs / s)
DY-ResNet-18 (Chen et al., 2020)	224×224	73.8	45M	-
ResNet-18 (He et al., 2016)	224×224	70.4	12M	1643.2
+MoRF (hard)	224×224	74.2	20M	1488.9
+MoRF (soft)	224×224	74.6	20M	1402.6
DY-ResNet-50 (Chen et al., 2020)	224×224	79.0	101M	-
SKNet-50 (Li et al., 2019b)	224×224	79.2	28M	1004.4
ResNet-50 (He et al., 2016)	224×224	78.4	26M	1250.3
+MoRF (hard)	224×224	79.2	37M	1028.6
+MoRF (soft)	224×224	79.5	37M	992.9
RegNetY-8GF (Radosavovic et al., 2020)	224×224	81.7	39M	591.6
DeiT-S/16 (Touvron et al., 2021b)	224×224	79.8	22M	940.4
Swin-T (Liu et al., 2021b)	224×224	81.3	29M	755.2
Focal-T (Yang et al., 2022)	224×224	82.2	29M	-
ViP-Small/7 (Hou et al., 2021)	224×224	81.5	25M	719.0
AS-MLP-T (Lian et al., 2022a)	224×224	81.3	28M	1047.7
ConvNeXt-T (Liu et al., 2022c)	224×224	82.1	29M	774.7
+MoRF (hard)	224×224	82.3	31M	732.8
+MoRF (soft)	224×224	82.6	31M	718.5
Swin-S (Liu et al., 2021b)	224×224	83.0	50M	436.9
Focal-S (Yang et al., 2022)	224×224	83.5	51M	-
T2T-ViT$_t$-24 (Yuan et al., 2021b)	224×224	82.6	65M	-
ViP-Medium/7 (Hou et al., 2021)	224×224	82.7	55M	418.0
AS-MLP-S (Lian et al., 2022a)	224×224	83.1	50M	619.5
ConvNeXt-S (Liu et al., 2022c)	224×224	83.1	50M	447.1
+MoRF (hard)	224×224	83.5	52M	418.5
+MoRF (soft)	224×224	83.7	52M	406.7
DeiT-B (Touvron et al., 2021b)	224×224	81.8	86M	292.3
Swin-B (Liu et al., 2021b)	224×224	83.3	88M	278.1
Focal-B (Yang et al., 2022)	224×224	83.8	90M	-
ViP-Large/7 (Hou et al., 2021)	224×224	83.2	88M	298.0
AS-MLP-B (Lian et al., 2022a)	224×224	83.3	88M	455.2
ConvNeXt-B (Liu et al., 2022c)	224×224	83.8	89M	292.1
+MoRF (hard)	224×224	84.0	91M	263.8
+MoRF (soft)	224×224	84.2	91M	254.1
Swin-B (Liu et al., 2021b)	384×384	84.5	88M	85.1
ConvNeXt-B (Liu et al., 2022c)	384×384	85.1	89M	95.7
+MoRF (hard)	384×384	85.2	91M	69.6
+MoRF (soft)	384×384	85.3	91M	60.3

Table 1. The experimental results of different architectures on ImageNet-1K. '+MoRF (hard/soft)' means that we integrate the MoRF method in the above architectures with hard routing and soft routing. For that, we show the Params. and throughput in the inference stage. Throughput is measured with a batch size of 64 on a single V100 GPU (32GB).

baselines. Specifically, for ResNet-18 and ResNet-50, our method with soft routing surpasses the baseline by 7.4% (74.6% *vs.* 70.4%) and 1.4% (79.5% *vs.* 78.4%) with similar parameters and throughput[1], respectively. Such results also exceed dynamic convolution (Chen et al., 2020) based ResNet. From ConvNeXt-T to ConvNeXt-B, integrating our MoRF also outperforms the baselines at 224×224 and 384×384 resolutions, further demonstrating the effectiveness of our method. In addition, employing soft routing achieves marginally superior performance compared to hard routing. This improvement is attributed to the enhanced spatial feature interaction facilitated by the use of a more diverse range of receptive fields. Compared to the baselines, the introduction of MoRF incurs some additional parameters, primarily from the attention module. However, this increase in the number of parameters is marginal relative to those of the whole architecture. Additionally, thanks to the re-parameterization of our method at the inference stage, the inference throughput remains comparable to the standard ConvNeXt.

[1]We measure the parameters and throughput in the inference stage after re-parameterization.

4.2. The Impacts of Model Configurations

In order to investigate the impacts of different configurations on performance improvement, we conduct a series of experiments to analyze the following structures and hyper-parameters configurations. Specifically, We systematically evaluate the impacts of the receptive field experts and the routing functions. Unless specified, all experiments are implemented in ImageNet-1K with running 100 epochs.

The Receptive Field Experts. In the case of ConvNeXt, a kernel size of 7×7 is used to obtain the local receptive field, which does not require a large receptive field. In MoRF, we first evaluate the impact of the number of receptive fields. Table 2a presents the results of different mixtures of receptive field experts. We show the performance of a mixture of receptive field experts with $\{3, 5, 7\}$, $\{3, 5, 7, 9, 11\}$ and $\{3, 5, 7, 9, 11, 13, 15\}$, respectively. When the receptive field experts are $\{3, 5, 7, 9, 11\}$, the model achieves better results. It is possible because a small number of experts limit the model's capacity, while too many experts make it challenging for the model to optimize. Furthermore, we consider the impact of larger receptive field experts in Table 2b. In our MoRF setting, we do not observe better results with larger kernels. When the kernel size increases, the number of parameters will also increase and throughput will decrease, although the re-parameterization trick is employed.

Attention Functions. In our study, we also validate the impact of various attention functions within the soft routing mechanism. The results are shown in Table 2c. Specifically, we evaluate the effectiveness of sigmoid, softmax, and unnormalized attention functions. Our findings indicate that the unnormalized attention mechanism notably enhances accuracy, establishing its superiority in this context.

4.3. Frequency Analysis

One of the primary objectives of MoRF is to enable the model to learn the adaptive receptive field based on an input image. To investigate the preferences of different receptive field experts for each sample in every block, we visualize the frequency with which these experts are selected in various layers in hard routing, as shown in Figure 3. We observe that from the first block to the final block, the selection of receptive field experts for each sample in each block is relatively random and does not reflect a particularly pronounced statistical preference. This phenomenon is interesting because it contradicts our intuition. Previous hybrid networks

Model	Top-1 (%)	Params.	Throughput. (imgs/s)
ConvNeXt	80.19	29M	775
{3, 5, 7}	80.26/80.41	30M	762
{3, 5, 7, 9, 11}	80.43/80.62	31M	742
{3, 5, 7, 9, 11, 13, 15}	80.39/80.59	31M	726

(a) The impacts of the number of receptive field experts. We choose {3, 5, 7}, {3, 5, 7, 9, 11}, {3, 5, 7, 9, 11, 13, 15} as the receptive field experts, where the number shows the kernel size in convolutions.

Model	Top-1 (%)	Params.	Throughput. (imgs/s)
ConvNeXt	80.19	29M	775
{3, 5, 7, 9, 11}	80.43/80.62	31M	742
{3, 7, 11, 15, 19}	80.38/80.58	31M	720
{3, 9, 15, 21, 27}	80.41/80.62	32M	701

(b) The impacts of large receptive fields. We choose {3, 5, 7, 9, 11}, {3, 7, 11, 15, 19}, and {3, 9, 15, 21, 27} as the receptive field experts, where the number shows the kernel size in convolutions.

Attention function	Top-1 (%)	Params.	Throughput. (imgs/s)
ConvNeXt	80.19	29M	775
sigmoid	80.37	30M	759
softmax	80.41	30M	742
unnormalized	80.62	31M	726

(c) The impacts of the attention function in the soft routing. The channel-wise attention establishes a better accuracy.

Table 2. The impacts of different model configurations. We show the accuracy with the hard routing and soft routing for the evaluation of receptive field experts and split them with a slash (/).

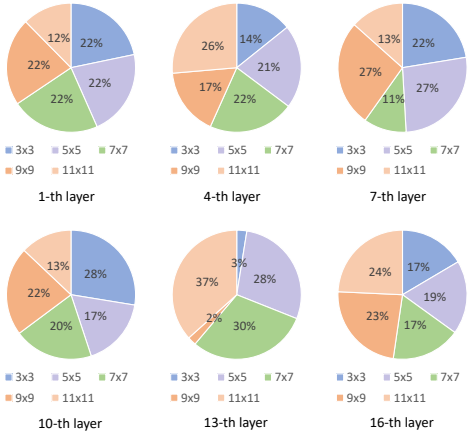

Figure 3. From top to bottom, from left to right, we visualize the frequency of selected experts for each block inserted in MoRF. We forward all the samples in the ImageNet-1K validation set to the pre-trained ConvNeXt + MoRF model. Different colors in the pie chart represent different receptive field experts, such as convolution with a kernel size of 3×3. The number on top of the chart represents the frequency of choosing this expert.

that combine convolutions and transformers typically use convolutional layers in the initial layers and self-attention in the final layers. However, our results show that each sample has its own preference, and some samples still choose larger receptive fields in the initial layers and local receptive fields in the later layers. Moreover, due to the presence of downsampling in various layers, the feature map size in the last layer is 7×7. Therefore, when we observe the last figure (bottom-right), we can see that most samples choose a kernel larger than or equal to 7×7 (64%), which is equivalent to covering the global receptive field. In contrast, in the first block (top-left), the samples tend to choose the local receptive field. As a result, previous works using convolutions in the initial layers and self-attention in the final layers satisfy most samples (though not all) and achieve excellent performance.

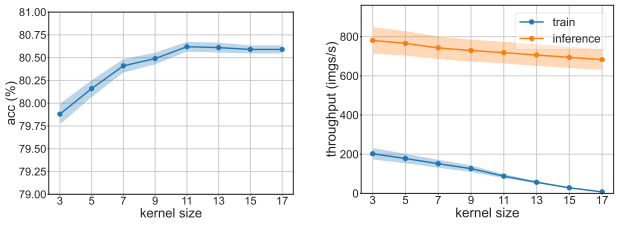

Figure 4. The curves of accuracy, training and inference throughputs with the cumulative increase in kernel sizes. Left: accuracy; Right: throughput.

4.4. Training and Inference Cost

In light of our implementation involving a mixture of receptive fields for network training, an increased training cost is consequently introduced compared to the single receptive field. To systematically assess this, we conduct a quantitative evaluation of the impact of kernel size on several key performance: accuracy, training throughput, and inference throughput, as depicted in Figure 4. We show curves of accuracy, training and inference throughputs with the cumulative increase in kernel sizes, where we train the model for 100 epochs, as done in Sec. 4.2. We find that the introduction of a greater number and larger size of kernels leads to a more gradual smoothing of the accuracy curve. It is particularly noteworthy that, while training throughput encounters a rapid decline with the enlargement of kernel size, the decline in inference throughput tends to be more gradual.

4.5. Object Detection

In addition to validating the efficacy of our MoRF model in image classification, we further explore its effectiveness in object detection. To this end, we adopt the Mask R-CNN framework and mmdetection (Chen et al., 2019), following the settings of ConvNeXt (Liu et al., 2022c). The experiments are conducted in the COCO datasets (Lin et al., 2014), which contains 118K training data and 5K validation data.

Backbone	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	Params.
Mask R-CNN (3×)							
ResNet50 (He et al., 2016)	41.0	61.7	44.9	37.1	58.4	40.1	44M
ResNet101 (He et al., 2016)	42.8	63.2	47.1	38.5	60.1	41.3	63M
PVT-Small (Wang et al., 2021)	43.0	65.3	46.9	39.9	62.5	42.8	44M
Swin-T (Liu et al., 2021b)	46.0	68.2	50.2	41.6	65.1	44.8	48M
ConvNeXt-T (Liu et al., 2022c)	46.2	67.9	50.8	41.7	65.0	44.9	48M
+MoRF (hard)	46.5	68.3	50.8	41.9	65.2	45.0	50M
+MoRF (soft)	46.8	68.7	51.0	42.2	65.5	45.2	50M
Cascade Mask R-CNN (3×)							
Swin-T (Liu et al., 2021b)	50.5	69.3	54.9	43.7	66.6	47.1	86M
ConvNeXt-T (Liu et al., 2022c)	50.4	69.1	54.8	43.7	66.5	47.3	86M
+MoRF (hard)	50.8	69.6	55.1	44.0	67.0	47.4	88M
+MoRF (soft)	51.1	69.8	55.4	44.2	67.3	47.5	88M
Swin-S (Liu et al., 2021b)	51.9	70.7	56.3	45.0	68.2	48.8	107M
ConvNeXt-S (Liu et al., 2022c)	51.9	70.8	56.5	45.0	68.4	49.1	108M
+MoRF (hard)	52.1	71.0	56.6	45.1	68.5	49.1	110M
+MoRF (soft)	52.2	71.1	56.7	45.2	68.5	49.2	110M
Swin-B (Liu et al., 2021b)	51.9	70.5	56.4	45.0	68.1	48.9	145M
ConvNeXt-B (Liu et al., 2022c)	52.7	71.3	57.2	45.6	68.9	49.5	146M
+MoRF (hard)	52.8	71.5	57.2	45.6	69.0	49.6	148M
+MoRF (soft)	52.9	71.7	57.3	45.7	69.1	49.6	148M

Table 3. The object detection and instance segmentation results with Mask R-CNN and Cascade Mask R-CNN (3x schedule) on the COCO val2017 dataset, respectively. For ConvNeXt + MoRF (hard/soft), Params. means the inference parameters after re-parameterization.

To train the ConvNeXt + MoRF (hard/soft) backbones, we use the model pre-trained on ImageNet-1K and use a typical 3x schedule (36 epochs). We employ a multi-scale training strategy (Carion et al., 2020; Sun et al., 2021) that resizes the shorter side to 800 and the longer side to at most 1333. Our model is trained using the AdamW optimizer with a batch size of 16 (2 images per GPU × 8 GPUs) and a learning rate of 0.0001.

The experimental results are reported in Table 3. We measure the object detection and instance segmentation with AP^b for the box and AP^m for the mask. Our ConvNeXt + MoRF achieves better performance compared to the baselines (ConvNeXt).

Specifically, Mask R-CNN + ConvNeXt-T achieves 46.2 AP^b and 41.7 AP^m with the 48M parameters. Our Mask R-CNN + ConvNeXt + MoRF (soft) has 46.8 AP^b and 42.2 AP^m with similar parameters, which is better. Additionally, our model with Cascade Mask R-CNN also achieves better performance compared to the baselines, which shows the effectiveness of our MoRF in object detection and instance segmentation. Similar to image classification, the increased number of parameters in our MoRF models is primarily due to features selecting larger receptive fields in some blocks to cover long-range dependencies in object detection, resulting in a larger kernel size of the convolution for computation. However, such an increase in parameters is marginal.

Method	Backbone	mIoU (ms)	Params.
DANet (Fu et al., 2019a)	ResNet-101	45.2	69M
DeepLabv3+ (Chen et al., 2018)	ResNet-101	44.1	63M
ACNet (Fu et al., 2019b)	ResNet-101	45.9	-
DNL (Yin et al., 2020)	ResNet-101	46.0	69M
OCRNet (Yuan et al., 2020)	ResNet-101	45.3	56M
UperNet (Xiao et al., 2018)	ResNet-101	44.9	86M
OCRNet (Yuan et al., 2020)	HRNet-w48	45.7	71M
DeepLabv3+ (Chen et al., 2018)	ResNeSt-101	46.9	66M
DeepLabv3+ (Chen et al., 2018)	ResNeSt-200	48.4	88M
UperNet (Xiao et al., 2018)	Swin-T (Liu et al., 2021b)	45.8	60M
	ConvNeXt-T (Liu et al., 2022c)	46.7	60M
	+MoRF (hard)	47.2	63M
	+MoRF (soft)	47.7	63M
	Swin-S (Liu et al., 2021b)	49.5	81M
	ConvNeXt-S (Liu et al., 2022c)	49.6	82M
	+MoRF (hard)	49.9	85M
	+MoRF (soft)	50.1	85M
	Swin-B (Liu et al., 2021b)	49.7	121M
	ConvNeXt-B (Liu et al., 2022c)	49.9	122M
	+MoRF (hard)	50.3	125M
	+MoRF (soft)	50.6	125M

Table 4. The semantic segmentation results on the ADE20K validation set. For ConvNeXt + MoRF (hard/soft), Params. means the inference parameters after re-parameterization.

4.6. Semantic Segmentation

Following the settings of ConvNeXt (Liu et al., 2022c), we conduct experiments of semantic segmentation on the ADE20K dataset (Zhou et al., 2017), which consists of 20,210 training images and 2,000 validation images. UperNet (Xiao et al., 2018) and mmsegmentation (Contributors, 2020) frameworks are employed with our ConvNeXt + MoRF (hard/soft) backbones. We train our models with a batch size of 16 (2 images per GPU × 8 GPUs) and a learning rate of 6×10^{-5} for 160K iterations using the AdamW optimizer. We use data augmentation techniques such as horizontal flipping, random re-scaling within a ratio range of [0.5, 2.0], and random photometric distortion, as employed by ConvNeXt (Liu et al., 2022c). The input resolution is set to 512×512, and all the models are fine-tuned with pre-trained models on ImageNet-1K.

We list the experimental results in Table 4, where the performance is measured by multi-scale mIoU. With similar parameters, UperNet + ConvNeXt-T (soft) achieves a better result than UperNet + ConvNeXt-T (47.7 *vs.* 46.7) in terms of ms mIoU, which shows the effectiveness of our MoRF in the semantic segmentation task.

5. Conclusion

In this paper, we explore the impact of dynamic receptive fields on vision architectures. Specifically, we introduce a novel framework, termed Mixture of Receptive Fields (MoRF), which diverges from the conventional single receptive field approach. MoRF encompasses a combination of

various receptive fields, such as convolutions with diverse kernel sizes, where each convolution is regarded as a specialized expert. Such an approach allows for both the dynamic selection of suitable receptive fields for the given input and the expansion of the network capacity. Additionally, we incorporate two distinct routing mechanisms—hard routing and soft routing—to autonomously determine the optimal receptive field experts. During the inference phase, these selected experts are merged through re-parameterization, thereby preserving inference speeds comparable to those of architectures with a single receptive field. To validate the effectiveness of MoRF, we integrate this concept into various types and scales of architectures, including ResNet and ConvNeXt. Our comprehensive experiments demonstrate that MoRF-enhanced architectures achieve superior performance in tasks such as image classification, object detection, and segmentation, without markedly increasing model parameters and inference times.

Limitations. Due to resource constraints, the implementation of a vast array of receptive field experts, as used in models like the Mixture of Experts (MoE), for instance, 128 experts, remains beyond our current experimental capacity. Consequently, our investigations are constrained to a selection of seven receptive field experts. The integration of a larger mixture of receptive field experts will result in an increase in training resources. A potential solution is the optimization using CUDA kernels, which will be considered as future work. Furthermore, during inference, re-parameterization reduces the computational cost, but the training phase remains resource-intensive. Future work could apply this concept to transformer architectures, such as by designing varying window sizes for attention interactions to dynamically select receptive fields.

Acknowledgements

This project is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Award Number: MOE-T2EP20122-0006), and the Singapore National Research Foundation ("CogniVision – Energy-autonomous always-on cognitive and attentive cameras for distributed real-time vision with milliwatt power consumption" grant NRF-CRP20-2017-0003) – www.green-ic.org/CogniVision.

Impact Statement

This work presents a significant contribution to the field of computer vision by introducing a flexible and efficient method for leveraging multiple receptive fields in CNNs. This advancement has the potential to drive further research and development in neural network architectures, leading to more adaptive and high-performing models in various applications. While advancements in the field of Machine Learning often carry potential risks and societal consequences, we believe that our proposed method, MoRF, does not present specific concerns that need to be explicitly highlighted.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, pp. 213–229. Springer, 2020.

Chen, C.-F. R., Fan, Q., and Panda, R. CrossViT: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 357–366, October 2021.

Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., et al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 801–818, 2018.

Chen, S., Xie, E., Ge, C., Liang, D., and Luo, P. Cyclemlp: A mlp-like architecture for dense prediction. *International Conference on Learning Representations (ICLR)*, 2022.

Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., and Liu, Z. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11030–11039, 2020.

Chollet, F. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1251–1258, 2017.

Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., and Shen, C. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

Contributors, M. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 764–773, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, 2009.

Ding, X., Xia, C., Zhang, X., Chu, X., Han, J., and Ding, G. Repmlp: Re-parameterizing convolutions into fully-connected layers for image recognition. *arXiv preprint arXiv:2105.01883*, 2021a.

Ding, X., Zhang, X., Ma, N., Han, J., Ding, G., and Sun, J. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13733–13742, 2021b.

Ding, X., Zhang, X., Han, J., and Ding, G. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11963–11975, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

Du, N., Huang, Y., Dai, A. M., Tong, S., Lepikhin, D., Xu, Y., Krikun, M., Zhou, Y., Yu, A. W., Firat, O., et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning*, pp. 5547–5569. PMLR, 2022.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res*, 23:1–40, 2021.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., and Lu, H. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3146–3154, 2019a.

Fu, J., Liu, J., Wang, Y., Li, Y., Bao, Y., Tang, J., and Lu, H. Adaptive context network for scene parsing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 6748–6757, 2019b.

Gao, H., Zhu, X., Lin, S., and Dai, J. Deformable kernels: Adapting effective receptive fields for object deformation. *arXiv preprint arXiv:1910.02940*, 2019.

Gao, S., Li, Z.-Y., Han, Q., Cheng, M.-M., and Wang, L. Rf-next: Efficient receptive field search for convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2984–3002, 2022.

Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jegou, H., and Douze, M. Levit: A vision transformer in convnet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12259–12269, 2021.

Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., and Sun, J. Single path one-shot neural architecture search with uniform sampling. In *European Conference on Computer Vision*, pp. 544–560. Springer, 2020.

Hazimeh, H., Zhao, Z., Chowdhery, A., Sathiamoorthy, M., Chen, Y., Mazumder, R., Hong, L., and Chi, E. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34:29335–29347, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Hou, Q., Jiang, Z., Yuan, L., Cheng, M.-M., Yan, S., and Feng, J. Vision permutator: A permutable mlp-like architecture for visual recognition. *arXiv preprint arXiv:2106.12368*, 2021.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. Deep networks with stochastic depth. In *European Conference on Computer Vision (ECCV)*, pp. 646–661. Springer, 2016.

Huang, G., Chen, D., Li, T., Wu, F., Van Der Maaten, L., and Weinberger, K. Q. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017a.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017b.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. Adaptive mixtures of local experts. *Neural computation*, 3(1): 79–87, 1991.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

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

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., and Chen, Z. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

Li, C., Zhou, A., and Yao, A. Omni-dimensional dynamic convolution. *arXiv preprint arXiv:2209.07947*, 2022.

Li, D., Hu, J., Wang, C., Li, X., She, Q., Zhu, L., Zhang, T., and Chen, Q. Involution: Inverting the inherence of convolution for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12321–12330, 2021.

Li, H., Zhang, H., Qi, X., Yang, R., and Huang, G. Improved techniques for training adaptive deep networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1891–1900, 2019a.

Li, X., Wang, W., Hu, X., and Yang, J. Selective kernel networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 510–519, 2019b.

Lian, D., Zheng, Y., Xu, Y., Lu, Y., Lin, L., Zhao, P., Huang, J., and Gao, S. Towards fast adaptation of neural architectures with meta learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Lian, D., Yu, Z., Sun, X., and Gao, S. As-mlp: An axial shifted mlp architecture for vision. In *International Conference on Learning Representations (ICLR)*, 2022a.

Lian, D., Zhou, D., Feng, J., and Wang, X. Scaling & shifting your features: A new baseline for efficient model tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pp. 740–755. Springer, 2014.

Liu, H., Simonyan, K., and Yang, Y. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Liu, H., Dai, Z., So, D. R., and Le, Q. V. Pay attention to mlps. *arXiv preprint arXiv:2105.08050*, 2021a.

Liu, S., Chen, T., Chen, X., Chen, X., Xiao, Q., Wu, B., Pechenizkiy, M., Mocanu, D., and Wang, Z. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022a.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021b.

Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12009–12019, 2022b.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022c.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Ma, N., Zhang, X., Huang, J., and Sun, J. Weightnet: Revisiting the design space of weight networks. In *European Conference on Computer Vision*, pp. 776–792. Springer, 2020.

Melas-Kyriazi, L. Do you even need attention? a stack of feed-forward layers does surprisingly well on imagenet. *arXiv preprint arXiv:2105.02723*, 2021.

Mullapudi, R. T., Mark, W. R., Shazeer, N., and Fatahalian, K. Hydranets: Specialized dynamic architectures for efficient inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8080–8089, 2018.

Pham, H., Guan, M., Zoph, B., Le, Q., and Dean, J. Efficient neural architecture search via parameters sharing. In *International conference on machine learning*, pp. 4095–4104. PMLR, 2018.

Polyak, B. T. and Juditsky, A. B. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. Designing network design spaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Ren, S., Zhou, D., He, S., Feng, J., and Wang, X. Shunted self-attention via multi-scale token aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Ren, S., Yang, X., Liu, S., and Wang, X. Sg-former: Self-guided transformer with evolving token reallocation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al. Sparse R-CNN: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14454–14463, 2021.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016.

Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Keysers, D., Uszkoreit, J., Lucic, M., et al. MLP-Mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

Touvron, H., Bojanowski, P., Caron, M., Cord, M., El-Nouby, A., Grave, E., Joulin, A., Synnaeve, G., Verbeek, J., and Jégou, H. ResMLP: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021a.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning (ICML)*, 2021b.

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., and Jégou, H. Going deeper with image transformers. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021c.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Wan, A., Dai, X., Zhang, P., He, Z., Tian, Y., Xie, S., Wu, B., Yu, M., Xu, T., Chen, K., et al. Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12965–12974, 2020.

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

Wang, X., Yu, F., Dou, Z.-Y., Darrell, T., and Gonzalez, J. E. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 409–424, 2018.

Xiao, T., Liu, Y., Zhou, B., Jiang, Y., and Sun, J. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 418–434, 2018.

Xiao, T., Singh, M., Mintun, E., Darrell, T., Dollár, P., and Girshick, R. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34:30392–30400, 2021.

Yang, B., Bender, G., Le, Q. V., and Ngiam, J. Condconv: Conditionally parameterized convolutions for efficient inference. *Advances in neural information processing systems*, 32, 2019.

Yang, J., Li, C., Dai, X., and Gao, J. Focal modulation networks. *Advances in Neural Information Processing Systems*, 35:4203–4217, 2022.

Yin, M., Yao, Z., Cao, Y., Li, X., Zhang, Z., Lin, S., and Hu, H. Disentangled non-local neural networks. In *European Conference on Computer Vision (ECCV)*, pp. 191–207. Springer, 2020.

Yu, F. and Koltun, V. Multi-scale context aggregation by dilated convolutions. In *International Conference on Learning Representations (ICLR)*, 2016.

Yu, T., Li, X., Cai, Y., Sun, M., and Li, P. S2-mlpv2: Improved spatial-shift mlp architecture for vision. *arXiv preprint arXiv:2108.01072*, 2021.

Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., and Yan, S. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10819–10829, 2022.

Yu, W., Si, C., Zhou, P., Luo, M., Zhou, Y., Feng, J., Yan, S., and Wang, X. Metaformer baselines for vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024a.

Yu, W., Zhou, P., Yan, S., and Wang, X. Inceptionnext: When inception meets convnext. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024b.

Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., and Wu, W. Incorporating convolution designs into visual transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 579–588, 2021a.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F. E., Feng, J., and Yan, S. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021b.

Yuan, Y., Chen, X., and Wang, J. Object-contextual representations for semantic segmentation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pp. 173–190. Springer, 2020.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Zagoruyko, S. and Komodakis, N. Diracnets: Training very deep neural networks without skip-connections. *arXiv preprint arXiv:1706.00388*, 2017.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhang, Z., Zhang, H., Zhao, L., Chen, T., and Pfister, T. Aggregating nested transformers. *arXiv preprint arXiv:2105.12723*, 2021.

Zheng, H., He, P., Chen, W., and Zhou, M. Mixing and shifting: Exploiting global and local dependencies in vision mlps. *arXiv preprint arXiv:2202.06510*, 2022.

Zhong, Z., Zheng, L., Kang, G., Li, S., and Yang, Y. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13001–13008, 2020.

Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., and Torralba, A. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 633–641, 2017.

Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., Hou, Q., and Feng, J. DeepViT: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.

Zoph, B., Bello, I., Kumar, S., Du, N., Huang, Y., Dean, J., Shazeer, N., and Fedus, W. Designing effective sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

A. Training Recipes and Dataset Details

We list the detailed training recipes for our two models: ConvNeXt-T and ResNet-50 with MoRF in Table 5. To be specific, we use a batch size of 4096 and a learning rate of $4e^{-3}$ for the above models. We use the Adam optimizer and train the model for 300 epochs. We also employ data augmentation techniques such as mixup (Zhang et al., 2017) and cutmix (Yun et al., 2019) to increase the size of our training dataset. We also use label smoothing (Szegedy et al., 2016) with a smooth ratio of 0.1 and DropPath (Huang et al., 2016) strategy. After training our models using the specified recipe, we achieved higher accuracy on the ImageNet-1K dataset compared to the baselines, which demonstrates the effectiveness of our training recipe for optimizing the performance of machine learning models.

We conduct experiments on the ImageNet-1K dataset, which contains about 1.28 million training samples and 50,000 validation samples. These samples are categorized into 1,000 different classes, providing a diverse range of categories such as different breeds of dogs, various types of vehicles, and numerous everyday objects.

	ConvNeXt-T	ResNet-50
training config	ImageNet-1K 224^2	ImageNet-1K 224^2
weight init	trunc. normal (0.2)	kaiming_normal
optimizer	AdamW	AdamW
base learning rate	4e-3	4e-3
weight decay	0.05	0.05
optimizer momentum	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999
batch size	4096	4096
training epochs	300	300
learning rate schedule	cosine decay	cosine decay
warmup epochs	20	20
warmup schedule	linear	linear
layer-wise lr decay	None	None
randaugment (Cubuk et al., 2020)	(9, 0.5)	(9, 0.5)
mixup (Zhang et al., 2017)	0.8	0.8
cutmix (Yun et al., 2019)	1.0	1.0
random erasing (Zhong et al., 2020)	0.25	0.25
label smoothing (Szegedy et al., 2016)	0.1	0.1
stochastic depth (Huang et al., 2016)	0.1	0.1
layer scale (Touvron et al., 2021c)	1e-6	1e-6
head init scale (Touvron et al., 2021c)	1.0	1.0
gradient clip	None	None
exp. mov. avg. (EMA) (Polyak & Juditsky, 1992)	0.9999	None

Table 5. The training recipes for ConvNeXt-T and ResNet-50 with MoRF, respectively.

B. Architecture Details

In our work, we perform the instantiation of the ConvNeXt model, integrating it with the MoRF framework. To facilitate a comprehensive understanding of the models, we present the architecture details in Table 5, assuming an input image size of 224×224. The second column of Table 5 illustrates the output size of the image output at each stage within the models. We employ the notation "Concat $n \times n$" to denote the concatenation of $n \times n$ neighboring features in a patch following Swin Transformer (Liu et al., 2021b). This technique has been shown to be effective in capturing local information in images. Additionally, we incorporate receptive field experts through the deployment of five distinct convolutional layers, each characterized by varying kernel dimensions, specifically denoted as d(3, 5, 7, 9, 11). These techniques enable the model to capture both local and global features of an image.

C. Visualizations

In our study, we employ the Grad-CAM algorithm (Selvaraju et al., 2017), to methodically analyze and compare the attention maps of various models. This algorithm computes the gradients of the predicted class score relative to the feature maps of the last convolutional layer. We effectively visualize these attention maps for two key model configurations: the standard ConvNeXt and the enhanced ConvNeXt integrated with MoRF, as depicted in Figure 6.

All models are trained on the ImageNet-1K dataset, and the specific configurations of these experiments are systematically presented in Table 5. A critical analysis of these results show that our proposed ConvNeXt + MoRF model demonstrates superior performance in generating high-quality attention maps. This enhanced performance of the ConvNeXt + MoRF

	downsp. rate (output size)	ConvNeXt-T + MoRF
stage 1	4× (56×56)	concat 4×4, 96-d, LN , [d(3, 5, 7, 9, 11), 96 1×1, 384 1×1, 96] × 3
stage 2	8× (28×28)	concat 2×2, 192-d , LN [d(3, 5, 7, 9, 11), 192 1×1, 768 1×1, 192] × 3
stage 3	16× (14×14)	concat 2×2, 384-d , LN [d(3, 5, 7, 9, 11), 384 1×1, 1536 1×1, 384] × 9
stage 4	32× (7×7)	concat 2×2, 768-d , LN [d(3, 5, 7, 9, 11), 768 1×1, 3072 1×1, 768] × 3
Global Average Pooling (GAP), Head		

Figure 5. Architecture details of ConvNeXt-T with MoRF.

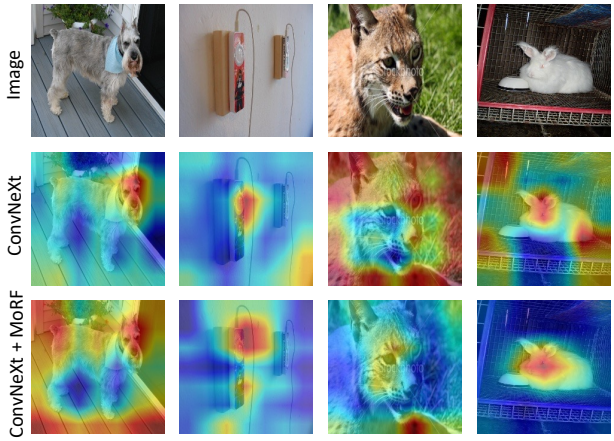

Figure 6. Visualization of attention maps. From top to bottom, each row shows the RGB image, ConvNeXt, and our ConvNeXt with MoRF.

model is primarily attributed to its innovative use of a mixture of receptive fields. This feature enables the model to adaptively and selectively focus on the most helpful receptive fields for different objects, thereby optimizing the attention mechanism for improved image classification accuracy.

D. More Ablation Experiments

In this study, we present an experimental study of the effectiveness of integrating a mixture of receptive fields (MoRF) within Convolutional Neural Networks (CNNs) to augment their performance in image classification. To this end, we insert the MoRF into ConvNeXt in different layers, and evaluate its performance on the ImageNet-1K dataset. ConvNeXt consists of four stages with [3, 3, 9, 3] blocks, and we insert MoRF from stage-1 to stage-4 to investigate its impact on the model's accuracy. We train the model on ImageNet-1K for 100 epochs by default and the results are presented in Table 6.

stage-1	stage-2	stage-3	stage-4	Accuracy (%)
				80.19
✓				80.34/80.42
	✓			80.32/80.40
		✓		80.29/80.40
			✓	80.24/80.35
✓	✓	✓	✓	80.43/80.62

Table 6. The impacts of inserting MoRF into different layers. We show the accuracy with the hard routing and soft routing for the evaluation of receptive field experts and split them with a slash (/).

From this table, we observe a significant improvement in accuracy when MoRF is inserted in different layers of the model. Notably, the most substantial improvement is recorded when MoRF is incorporated into all stages, achieving accuracy levels of 80.43% and 80.62%, thereby underscoring the potential of MoRF to significantly elevate CNN performance in image classification tasks. Furthermore, our findings reveal a more pronounced benefit when MoRF is added to the lower-level layers as compared to its inclusion solely in the higher-level layers. This differential impact may be attributed to the inherent nature of high-level layers, which typically already possess a global receptive field for features. In contrast, the integration of MoRF in lower-level layers appears to be more advantageous, likely due to their enhanced capacity for local feature refinement and extraction.