27th European Conference on Artificial Intelligence (ECAI), Santiago de Compostela, Spain, 2024.

HyenaPixel: Global Image Context with Convolutions

Julian Spravil[a,*], **Sebastian Houben**[b,a] **and Sven Behnke**[c,d,a]

[a]Fraunhofer IAIS, Germany
[b]University of Applied Sciences Bonn-Rhein-Sieg, Germany
[c]University of Bonn, Computer Science Institute VI, Center for Robotics, Germany
[d]Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

Abstract. In computer vision, a larger effective receptive field (ERF) is associated with better performance. While attention natively supports global context, its quadratic complexity limits its applicability to tasks that benefit from high-resolution input. In this work, we extend Hyena, a convolution-based attention replacement, from causal sequences to bidirectional data and two-dimensional image space. We scale Hyena's convolution kernels beyond the feature map size, up to 191×191, to maximize ERF while maintaining subquadratic complexity in the number of pixels. We integrate our two-dimensional Hyena, HyenaPixel, and bidirectional Hyena into the MetaFormer framework. For image categorization, HyenaPixel and bidirectional Hyena achieve a competitive ImageNet-1k top-1 accuracy of 84.9% and 85.2%, respectively, with no additional training data, while outperforming other convolutional and large-kernel networks. Combining HyenaPixel with attention further improves accuracy. We attribute the success of bidirectional Hyena to learning the data-dependent geometric arrangement of pixels without a fixed neighborhood definition. Experimental results on downstream tasks suggest that HyenaPixel with large filters and a fixed neighborhood leads to better localization performance.

1 Introduction

The 35-year history of Convolutional Neural Networks' (ConvNets) [24] successful track record [25, 2, 5, 23, 43, 18, 45] has recently been challenged by Vision Transformers (ViTs) [13]. The ViT plays a significant role in the recent improvements in computer vision [47, 52, 61] due to its simple architecture: The input image is split into equal-sized patches further processed by a regular transformer encoder with bidirectional attention [48]. This design scales well in terms of data and parameters and achieves remarkable performance in a self-supervised setting. Under the pressure of competition, ConvNets are currently reassessed. For example, new evidence suggests that ConvNets follow similar scaling laws [42, 52, 49]. On the other hand, ConvNets serve as a source of inspiration for ViT enhancements. For instance, the use of a hierarchical network layout [30]. Hybrid models have emerged that use convolution in earlier layers [56] or as a replacement of or addition to the Feed Forward Network (FFN) in each transformer block [47, 61]. Other improvements focus on applying attention to local windows [30] or sparse grids [47]. Other work explores computationally cheaper attention alternatives based on the Fourier transform [26], simple pooling [55], local convolutions [56], or state space models [62], among others.

* Corresponding Author. Email: julian.spravil @ iais.fraunhofer.de

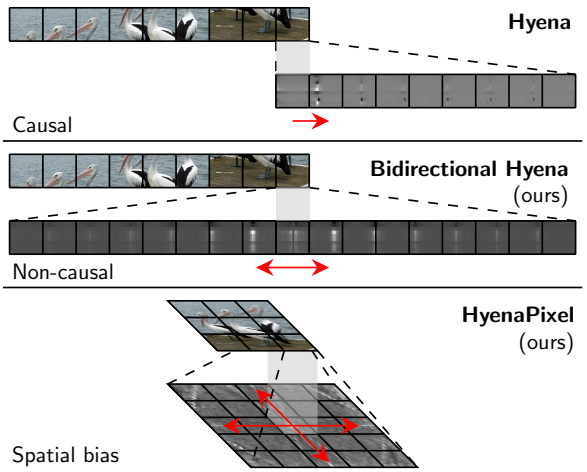


Figure 1. Our extensions of Hyena [36] (top). In bidirectional Hyena (center), a large non-causal filter is applied. HyenaPixel (bottom) uses a large convolutional kernel to process 2D feature maps. We show the evaluation of the rightmost token position and the resulting kernel overlap.

Token mixers with sub-quadratic complexity are highly sought after, as image resolution is one of the most important performance factors for image classification [47], vision language modeling [34], and other downstream tasks. Attention requires specialized strategies, such as subdividing images followed by separate processing [28], potentially limiting image context. Alternatively, for local operations, a deep network is essential [43, 18, 45]. A promising new path is the integration of large convolutional filters for sequence modeling [36, 14] and also for vision with medium [35, 31, 15] to large kernels sizes — up to 61× 61 [11, 29].

In this work, we explore the Hyena operator [36] as an attention replacement in vision applications. The Hyena operator uses long convolutions with gating and was originally proposed for causal language modeling. This token mixer qualifies for this exploration due to its sub-quadratic complexity and its use of convolution, native to computer vision. Hyena has an intuition similar to attention: It provides global context by computing a weighted sum, data-driven for attention and learned for Hyena, over all input tokens for each output token. We ask two research questions: i) Is an approximation of attention with fixed learned attention patterns, like the Hyena operator, a sufficient replacement for fine-granular, fully data-driven attention in vision applications? ii) Does adding a fixed pixel neighborhood or spatial bias impact performance?

Fig. 1 illustrates our approach. The main contributions of our work can be summarized as follows:

- We extend the causal convolution-based attention replacement Hyena [36] by considering bidirectional, non-causal information flow – bidirectional Hyena (H_b) – and by accommodating the 2D nature of images with spatial bias – HyenaPixel (H_{px}) – to build large effective receptive fields (ERFs) and capture the global image context. H_{px} inherits the properties of regular convolution and can thus be applied to arbitrary input sizes.
- We evaluate H_b and H_{px} in the MetaFormer framework [55] for image classification, and the latter also for object detection and semantic segmentation, outperforming other large-kernel networks and improving localization performance.
- We analyze the learned features of H_{px}, elaborate on the importance of global context, bidirectional modeling, and spatial bias with convolution, and compare our approach with different token mixer configurations. Finally, we gain valuable insights into the required global context for each network stage.

2 Related Work

Improvements to the ViT [13] focus on the architecture [30, 55], training strategy [52], and attention mechanism or generally token mixing [30, 12, 47, 55, 56, 61, 17]. Following a four-stage architecture with convolution-based downsampling layers, the hierarchical structure provides a consistent improvement [30]. There are different variants of attention for visual data: The Swin Transformer [30] applies attention to shifted rectangular windows, while MaxViT [47] uses window attention and sparse grid attention for global interactions. CSWin [12] uses parallel row and column attention with integrated position enhancement. BiFormer [61] implements data-driven key-value filtering to reduce computational overhead for irrelevant tokens. Similarly, DAT [53] selects important tokens based on fixed reference points and predicted offsets. GC-ViT [17] predicts semantically rich global query tokens used in cross-attention layers to aggregate global context from local features. Some methods use convolutional layers within each transformer block to enhance local positional information [12, 53, 61] while others replace the FFN with a convolutional component [47]. Current research is focused on self-supervised learning of visual features. [52].

ConvNets and large kernels. ConvNets first proposed in the 1980s [24] are responsible for may advancements in computer vision [25, 2, 5, 23, 43, 18, 45]. The success of the transformers has led to new advances in ConvNet research [48, 13]. Typically, the attention layer of the transformer architecture is replaced with a combination of convolutional layers [55, 56, 49]. InternImage [49] applies deformable convolutions to realize long-range data-driven dependencies and scale the model to one billion parameters. ConvNeXt [31] builds on a deep stack of small convolutional blocks, which is also suitable for unsupervised training as a masked autoencoder [52].

Recent research focuses on ConvNets with large kernels. Common across these networks is their regularization through parameterizing the convolution weights to guarantee smoothness [38, 39, 14, 36] or by applying sparsity of some form [9, 35, 15, 29]. Romero et al. [38] proposed parametric filters with dynamic size and discovered that the filter size increases with depth. Parameterized kernels require assumptions about how the input is processed [38, 39, 14, 36]. The global convolution network [35] applies separable convolutions (21×1 and 1×21) to improve classification while maintaining localization for semantic segmentation. SegNeXt [15] also utilizes parallel separable convolutions with sizes between 7 and 21.

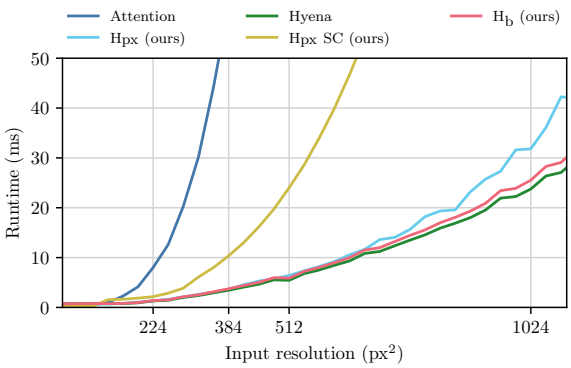


Figure 2. Runtime scaling of token mixers with global token interactions. Input images are patched with a patch size of 4. H_{px} SC uses separable convolutions (SC) instead of an implicit filter. The experiment was conducted on an Nvidia A100 GPU.

RepLKNet [11] uses full convolutions with sizes up to 31×31, while the large kernels are fused by re-parameterization of multiple smaller kernels. SLaK [29] proposes two parallel kernels spanning 61×5 with dynamic sparsity. However, dynamic sparsity, which theoretically reduces the multiply-accumulate operations (MACs), requires an efficient hardware implementation, still being sought.

Substitutes for attention. While attention is a powerful and flexible mechanism, its complexity is quadratic in the number of tokens [48]. Linear attention [21] uses a kernel formulation to express token similarity. However, finding expressive kernel functions is challenging [16]. MLP-Mixer [46] uses multiple linear layer stacks applied alternating on the channel and token dimensions. The idea of basic token mixing is further extended to a mean-pooling approach [55] and simple convolutional layers [56]. FNet [26] applies the Fourier transform along the token and channel dimensions. Hyena [36] uses long and short convolutions for causal token mixing. Convolution appears to be a promising solution for vision-related [56] but also sequence-modeling [36] tasks as many other alternatives struggle to achieve high performance.

The simultaneous work by Zimerman and Wolf [63], like ours, aims to raise the dimensional extent of Hyena. The authors evaluate their approach on small datasets in different transformer frameworks. Their approach improved the performance over their baselines but also benefited from additional subsequent attention layers. The causality of Hyena is addressed by rotating the input after each layer. We propose a non-causal Hyena layer that does not require input transformations like rotation. While the authors showed improved classification performance for small datasets by adding spatial bias, we observe the opposite for larger corpora. In this case, we show that sequential bidirectional data modeling is superior.

3 Method

Motivation. Vision Transformers (ViTs) are one point ahead of Convolutional Neural Networks (ConvNets): A single attention layer has a global context. Current ConvNets scale kernels to at most 61×61 [11, 29] and thus only give the center pixel full context. However, kernels larger than the feature map proved beneficial [11]. A recent approach designed for language modeling promises global context based on gated global convolution, namely the Hyena operator [36]. Motivated by Hyena's promising properties for sequence modeling, we apply it to the 2D-pixel space with drastically larger kernels than previously considered.

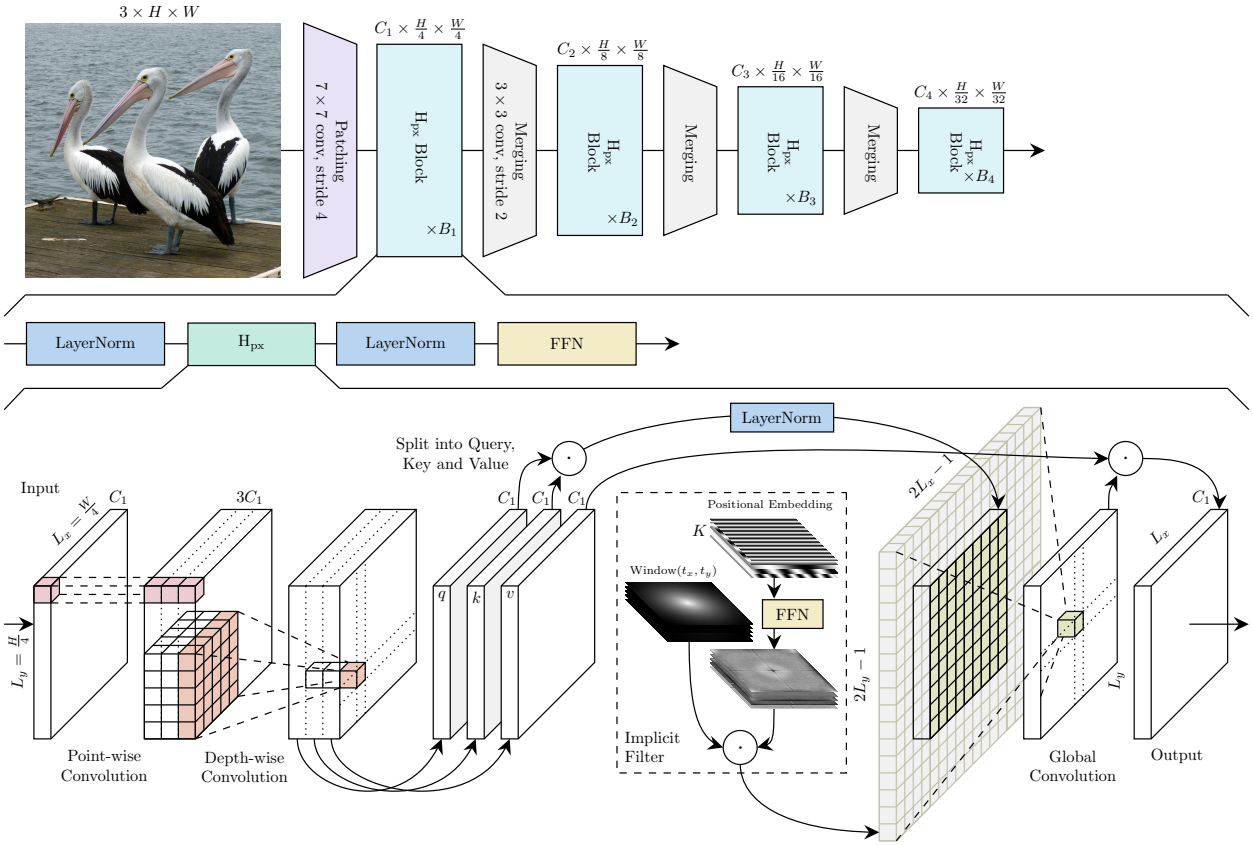


Figure 3. The HyenaPixel ($\mathrm{H_{px}}$) operator embedded in the MetaFormer framework. The first row shows the MetaFormer framework [56] with an input of size $H \times W$. The input is divided into overlapping patches and processed by a sequence of $\mathrm{H_{px}}$ blocks with intermediate merging layers to reduce the spatial resolution. The second row shows the $\mathrm{H_{px}}$ block with Layer Norm [1] and a Feed Forward Network (FFN). The last row shows the $\mathrm{H_{px}}$ operator. The input feature map has two spatial dimensions $L_y = H/4$ and $L_x = W/4$ and the channel dimension C_1. First, the dimension is increased to $3C_1$ by a point-wise and a depth-wise 5×5 convolution. The resulting feature map is split into three equal-sized chunks: query q, key k, and value v. The result of the element-wise multiplication $\odot$ of q and k is normalized and convolved with a global implicit filter. The final output is the element-wise multiplication with v.

Hyena. The Hyena operator by Poli et al. [36] first projects the input sequence x of length L into different spaces $p_0(x), \ldots, p_{O+1}(x)$. The number of projections is determined by the order parameter O. The projection $p_i(\cdot)$ is obtained by a linear mapping of x, followed by a convolution. Output aggregation is done recursively by element-wise multiplication of the previous result with the next projection:

$$y_{i+1} = g(y_i) \cdot p_{i+2}(x). \tag{1}$$

The initial value is set to $y_0 = p_0(x) \cdot p_1(x)$. $g(y_i)$ denotes a circular global convolution on y_i accelerated by the convolution theorem, whose filter weights are implicitly modeled by applying an FFN with sinusoidal activations to a position embedding. The positional embedding is a truncated complex exponential basis

$$\rho_k(t) = \exp\left(i 2\pi k t / L\right) \tag{2}$$

for $k \in \{0, \ldots, K-1\}$ with embedding dimension K and $t \in \{0, \ldots, L-1\}$. The real and imaginary parts of $\rho_k(t)$ are concatenated along the embedding dimension. To reduce the influence of distant tokens, the exponential decay function

$$\mathrm{Window}(t) = \exp\left(-\alpha t\right) + b, \tag{3}$$

is used, with scaling factor α and bias b. Causality, where the next token in a sequence can only attend to previous tokens, is achieved by zero padding input and filter to length $2L-1$ and keeping the L leftmost output positions of the circular convolution. We simplify Hyena by setting $O = 2$ and rewrite the recursive function as

$$y = g(q \cdot k) \cdot v, \tag{4}$$

with query $q = p_0(x)$, key $k = p_1(x)$ and value $v = p_2(x)$.

Bi-directional Hyena ($\mathrm{H_b}$). Causality is unnatural for offline signal processing. We extend the Hyena operator to bidirectional sequences, namely $\mathrm{H_b}$, by increasing the implicit filter size. The position embedding $\rho_k(t_b)$ is expanded with $t_b \in \{0, \ldots, 2L-1\}$ and by replacing L with $2L$. The absolute exponential decay $\mathrm{Window}(|t'_b|)$ is indexed with $t'_b \in \{-L+1, \ldots, L-1\}$. A centered evaluation region ensures complete sequence coverage by the filter at each token position, i.e., selecting the output indices $\lfloor \frac{L}{2} \rfloor, \lfloor \frac{L}{2} \rfloor + 1, \ldots, \lfloor \frac{L}{2} \rfloor + L$ of $g(\cdot)$. Note that the complexity is the same as for the causal Hyena operator, i.e. $\mathcal{O}(L \log_2 L)$.

HyenaPixel ($\mathrm{H_{px}}$). Image processing could benefit from taking 2D-neighborhood information into account. To add spatial bias to $\mathrm{H_b}$, we replace the implicit filter with a 2D pendant for a $L_y \times L_x$ feature map size. We use the 2D positional embedding proposed by Wang and Liu [50], which is defined as follows

$$\rho_k(t_x, t_y) = \begin{cases} \exp\left(i \frac{t_x}{10000^{2k/K}}\right) & \text{if } 0 \le k < \frac{K}{2} \\ \exp\left(i \frac{t_y}{10000^{2(k-K/2)/K}}\right) & \text{if } \frac{K}{2} \le k < K \end{cases}, \tag{5}$$

with $t_x \in \{0, \dots, 2L_x - 1\}$ and $t_y \in \{0, \dots, 2L_y - 1\}$. $\rho_k(t_x, t_y)$ uses the first half of the K dimensions to encode the horizontal and the second half to encode the vertical direction. The embedding is transformed into the filter by flattening the spatial dimensions and applying Hyena's FFN. The exponential decay is defined by

$$\text{Window}(t'_x, t'_y) = \exp\left(-\alpha\sqrt{t'^2_x + t'^2_y}\right) + b, \tag{6}$$

with filter positions $t'_x \in \{-L_x + 1, \dots, L_x - 1\}$ and $t'_y \in \{-L_y + 1, \dots, L_y - 1\}$. The centered evaluation region of H_b is applied to the vertical and horizontal axes. We name this extension HyenaPixel (H_px). The asymptotic complexity is $\mathcal{O}(L_x L_y \log_2 (L_x L_y))$. In practice, the performance of H_px is comparable to that of H_b for resolutions below $512\,\text{px}^2$, see Fig. 2.

Hierarchical transformer. We embed H_b and H_px in the MetaFormer framework [55, 56], a transformer encoder [48, 13] with a hierarchical structure. We chose this framework because MetaFormer has already been evaluated with different token mixer types, and hierarchical models consistently outperform their isomorphic counterparts [31]. The architecture is shown in Fig. 3.

There are a few key differences that set MetaFormer apart from the more commonly known Swin Transformer: The convolutions in the image patching and in-between patch merging layers have overlap, the network depth is increased while the network width is decreased, and the StarReLU [56] activation function is used.

Model sizes. We explore the following model sizes:

- S4: $C = (64, 128, 320, 512)$, $B = (1, 1, 1, 1)$;
- S12: $C = (64, 128, 320, 512)$, $B = (2, 2, 6, 2)$;
- S18: $C = (64, 128, 320, 512)$, $B = (3, 3, 9, 3)$; and
- B36: $C = (128, 256, 512, 768)$, $B = (3, 12, 18, 3)$.

Here, C is the channel dimension and B is the number of blocks per stage. We use the syntax of Yu et al. [56] and classify the channel dimensionality with the letter S (small) followed by the total number of blocks $\|B\|_1$. The full model is depicted in Fig. 3.

Token mixer layout. The main layout has H_b or H_px in each stage of the network, namely H_bFormer and H_pxFormer. The hyperparameters of H_b at stage i are set to sequence length $L = \left[56^2, 28^2, 14^2, 7^2\right]$ and hidden filter projection dimension of $4K_i$ for an input resolution of $224\,\text{px}^2$ and position embedding dimensions of $K = [16, 16, 24, 32]$. H_px parameters are similar, with the difference that the feature map size is set to $L_x = L_y = [56, 28, 14, 7]$. The global context provided by attention proved beneficial in later stages [56, 12]. Inspired by this observation, we also formulate the CH_pxFormer, with local convolutions in the first two and H_px in the last two stages. The local convolution follows the inverse separable convolution proposed in MobileNetV2 [40] that is also employed in the ConvFormer [55] with a kernel size of 7. Accordingly, we propose H_bAFormer and H_pxAFormer to evaluate whether attention has any additional value beyond the capabilities of H_b and H_px.

4 Evaluation

4.1 Image Classification

Training on ImageNet-1k. We train on ImageNet-1k (IN-1k) [10] consisting of 1.3M and 50K images in the training and validation set, respectively. The images are categorized into 1000 classes. We follow the training strategy of Yu et al. [56] and optimize with AdamW [32], a batch size of 4096, a learning rate of $4e^{-3}$, and a weight decay of 0.05 for 310 epochs. The learning rate is scheduled with a linear warm-up of 20 epochs followed by a cosine decay of 280 epochs and 10 cool-down epochs with a final learning rate of $1e^{-5}$. Regularization is added by stochastic depth [19] (0.6 for B36 scale, otherwise 0.2), label smoothing [44] with 0.1, and ResScale [41] in the last two stages. We do not apply token labeling [20]. We apply the following data augmentations: Mixup [58], Cutmix [57], RandAugment [8], and Random Erasing [59]. Our implementation is based on the *timm* framework [51].

Fine-tuning on higher resolution. ConvNets naturally scale to different resolutions and can show improved accuracy for higher input resolutions [45]. This also applies to H_pxFormer. On the other hand, H_bFormer is fitted to the specific input shape and would require an interpolation of the learned one-dimensional filters. Note that a similar procedure is required for ViTs where the positional embedding needs to be resampled [13].

We fine-tune H_pxFormer-S18 on IN-1k with the resolutions $384\,\text{px}^2$ and $512\,\text{px}^2$. Interpolating the positional embedding of the implicit filter of H_pxFormer-S18 to the sizes $[191, 95, 47, 23]$ showed no significant improvement for $384\,\text{px}^2$. For simplicity, we decided to keep the original filter sizes and apply zero padding. In accordance with the MetaFormer training scheme [56], we fine-tune for 30 epochs with AdamW, a learning rate of $5e^{-5}$, a batch size of 1024, exponential moving average [37] and head dropout of 0.4. Learning rate scheduling, Mixup, and Cutmix are disabled.

Results on ImageNet-1k. Tab. 1 reports the results on IN-1k for $224\,\text{px}^2$ images. For validation, a center-cropped region of the input image is selected with a crop size between 0.8 and 1.0, maximizing accuracy. Reference methods are selected based on a comparable training strategy and computational requirement.

We have three models that qualify as ConvNets: H_bFormer, H_pxFormer, and CH_pxFormer. Our best model, H_bFormer, outperforms other strong ConvNets, namely ConvNeXt [31], SLaK [29], and ConvFormer [56], and achieves on par performance to InternImage [49] on small scale (InternImage-T, 5.0G MACs, 83.5% accuracy) and surpasses it by 0.3% on a larger scale (InternImage-B, 18.0G MACs, 84.9% accuracy) with an accuracy of 85.2%. H_bFormer shows competitive performance compared to attention-based and hybrid models. On a small scale, the BiFormer-S [61] surpasses the H_bFormer-S18 by 0.3%, while it loses its advantage with increasing scale. H_bFormer-B36 is on par with the MaxViT-L [47] (43.9G MACs, 85.2% accuracy) while requiring 52% and 46% fewer parameters and MACs, respectively. However, the CAFormer-B36 [56] is 0.3% accuracy points ahead.

A fixed neighborhood definition slightly reduces the categorization performance. Compared to H_bFormer-B36, we observe a decrease of 0.3% accuracy points for H_pxFormer-B36.

Combining H_px or H_b with attention following CAFormer [56] leads to mixed results. H_b is incompatible with attention, leading to a 0.4% advantage of CAFormer-S18 over H_bAFormer-S18. We assume that the local positional information learned by the earlier H_b layers is not representative enough. On the other hand, H_pxAFormer-S18 is 0.1% better than GC-ViT-T and achieves equivalent performance to MaxViT-T and CAFormer-S18. These results suggest that global context in earlier layers does not affect categorization performance. This is consistent with our observation that H_px learns mostly local features in earlier stages (see Section 5).

We find that H_pxFormer-S18 and ConvFormer-S18 differ in about 50% of the wrongly classified images. With a simple training-free ensemble of these two models by mean pooling the predictions, namely H_px / Conv, the accuracy improves to 84.0%. By adding H_bFormer-S18 and CAFormer-S18 to the ensemble, i.e. H_px / Conv / H_b / CA, the accuracy further increases to 84.7%.

Tab. 2 reports results for higher-resolution inputs. Fine-tuning

Table 1. IN-1k validation set results with input resolutions of $224\,\text{px}^2$. We compare different attention (A), convolution (C), and hybrid (H) approaches. The approaches are categorized into the following groups based on the computational requirements: up to 8G MACs, 8-12G MACs, 12-18G MACs, and more than 18G MACs. MACs are calculated using `fvcore` [6]. The entries in each group are sorted in ascending order by the primary key "Top-1" accuracy and in descending order by the secondary key "MACs". Note that the reported parameter count and MACs of SLaK [29] marked with a "*" require specialized hardware supporting sparse convolution. Our models are highlighted in gray.

Model	Type	#Param.	MACs	Top-1
Swin-T [30]	A	28M	4.5G	81.4
DAT-T [53]	A	29M	4.6G	82.0
CSWin-T [12]	A	23M	4.3G	82.7
CSWin-S [12]	A	35M	6.9G	83.6
ConvNeXt-T [31]	C	29M	4.5G	82.1
SLaK-T [29]	C	*30M	*5.0G	82.5
CH$_{\text{px}}$Former-S18	C	28M	4.3G	83.0
ConvFormer-S18 [56]	C	27M	3.9G	83.0
H$_{\text{px}}$Former-S18	C	29M	4.9G	83.2
InternImage-T [49]	C	30M	5.0G	83.5
H$_{\text{b}}$Former-S18	C	28M	4.4G	83.5
H$_{\text{b}}$AFormer-S18	H	27M	4.4G	83.2
GC-ViT-T [17]	H	28M	4.7G	83.5
MaxViT-T [47]	H	31M	5.6G	83.6
H$_{\text{px}}$AFormer-S18	H	28M	4.7G	83.6
CAFormer-S18 [56]	H	26M	4.1G	83.6
BiFormer-S [61]	H	26M	4.5G	83.8
Swin-S [30]	A	50M	8.7G	83.3
DAT-S [53]	A	50M	9.0G	83.7
ConvNeXt-S [31]	C	50M	8.7G	83.2
SLaK-S [29]	C	*55M	*9.8G	83.8
H$_{\text{px}}$ / Conv	C	56M	8.8G	84.0
ConvFormer-S36 [56]	C	40M	7.6G	84.0
InternImage-S [49]	C	50M	8.0G	84.2
BiFormer-B [61]	H	59M	9.8G	84.3
GC-ViT-S [17]	H	51M	8.5G	84.3
MaxViT-S [47]	H	69M	11.7G	84.5
CAFormer-S36 [56]	H	39M	8.0G	84.5
Swin-B [30]	A	88M	15.4G	83.6
DAT-B [53]	A	88M	15.8G	84.0
CSWin-B [12]	A	78M	15.0G	84.2
RepLKNet-31B [11]	C	79M	15.3G	83.5
ConvNeXt-B [31]	C	89M	15.4G	83.9
SLaK-B [29]	C	*95M	*17.1G	84.0
ConvFormer-M36 [56]	C	57M	12.8G	84.5
H$_{\text{px}}$ / Conv / H$_{\text{b}}$ / CA	H	111M	17.3G	84.7
GC-ViT-B [17]	H	90M	14.8G	85.0
CAFormer-M36 [56]	H	56M	13.2G	85.1
ConvNeXt-L [31]	C	198M	34.4G	84.3
ConvFormer-B36 [56]	C	100M	22.6G	84.8
InternImage-B [49]	C	97M	18.0G	84.9
H$_{\text{px}}$Former-B36	C	111M	25.3G	84.9
H$_{\text{b}}$Former-B36	C	102M	23.8G	85.2
MaxViT-B [47]	H	120M	23.4G	85.0
MaxViT-L [47]	H	212M	43.9G	85.2
CAFormer-B36 [56]	H	99M	23.2G	85.5
GC-ViT-L [17]	H	201M	32.6G	85.7

on a resolution of $384\,\text{px}^2$ puts H$_{\text{px}}$Former-S18 with an accuracy of 84.7% ahead of ConvFormer-S18. Increasing the resolution to $512\,\text{px}^2$ further boosts the accuracy to 84.8%. MaxViT-T performs significantly better but requires more MACs.

Our results support the assumptions on MetaFormer [56] as a strong baseline model and the expressiveness of Hyena. Interestingly, we observe that features produced by different token mixers can be incompatible. Moreover, we close the gap between ConvNets and Transformers with a radical new approach: A ConvNet for vision without a predefined neighborhood – H$_{\text{b}}$Former.

Table 2. IN-1k validation set results with input resolutions of $384\,\text{px}^2$ and $512\,\text{px}^2$.

Model	Type	#Param.	MACs	Top-1
$384\,\text{px}^2$				
ConvFormer-S18 [56]	C	27M	11.6G	84.4
H$_{\text{px}}$Former-S18	C	29M	12.9G	84.7
CAFormer-S18 [56]	H	13.4G	85.0	
MaxViT-T [47]	H	31M	17.7G	85.2
$512\,\text{px}^2$				
H$_{\text{px}}$Former-S18	C	29M	22.3G	84.8
MaxViT-T [47]	H	31M	33.7G	85.7

Table 3. Effect of different ablations on the IN-1k top-1 accuracy.

		Top-1
H$_{\text{px}}$Former-S12 (Baseline)	$224\,\text{px}^2$	80.3
Kernel Size for Global Convolution		
$[55^2, 27^2, 13^2, 7^2]$	$224\,\text{px}^2$	80.3
$[27^2, 13^2, 7^2, 3^2]$	$224\,\text{px}^2$	80.1
$[9^2, 9^2, 9^2, 9^2]$	$224\,\text{px}^2$	80.3
Token Mixer		
Hyena	$224\,\text{px}^2$	79.9
H$_{\text{b}}$	$224\,\text{px}^2$	81.0
H$_{\text{px}}$ with Separable Conv.	$224\,\text{px}^2$	79.9
LayerNorm (LN)		
H$_{\text{px}}$Former-S18	$224\,\text{px}^2$	83.2
H$_{\text{px}}$Former-S18 without LN	$224\,\text{px}^2$	83.0
Network Depth		
H$_{\text{px}}$Former-S4	$224\,\text{px}^2$	73.4
ConvFormer-S4	$224\,\text{px}^2$	73.0
H$_{\text{px}}$Former-S4	$512\,\text{px}^2$	76.6
ConvFormer-S4	$512\,\text{px}^2$	75.5

4.2 Ablation Study

We test different aspects of H$_{\text{px}}$Former-S12. The training is conducted on IN-1k, and mainly follows the procedure described in Section 4.1, but, if not otherwise stated, we reduce the number of epochs from 310 to 160 and adjust the cosine decay accordingly. Tab. 3 reports the results of the ablation study.

Kernel size. The global convolution is the main component of H$_{\text{px}}$Former and is almost twice as large as the feature map, such that each output position can "see" all input positions, similar to attention. However, halving the kernel size does not affect performance, while using only a quarter of the original kernel size causes a slight drop in accuracy of 0.2%. Interestingly, using a constant kernel size of 9 causes no accuracy drop. The hierarchical structure of the network counteracts the loss of global context in each layer. However, once the later layers lose global context, accuracy is reduced. Due to the different means of exchanging global information, H$_{\text{px}}$Former has an inherent robustness to hyperparameter changes.

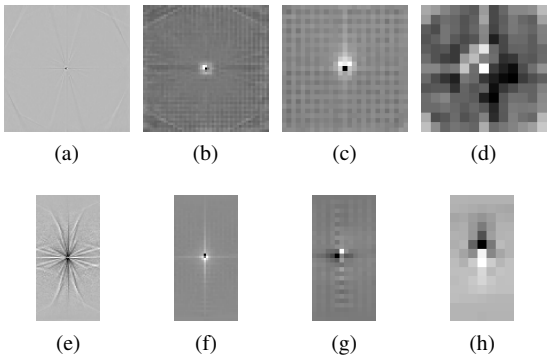


Figure 4. Hand-picked normalized mean kernel weights from each stage of the 2D global convolution layers in H_{px}Former-S18 (a) - (d) and the reshaped 1D global convolution layers in H_bFormer-S18 (e) - (h). Note that for H_b we wrap the kernel for a specific location for better visualization. The kernel would wrap differently at other evaluation positions due to the nature of the 1D convolution and the flattened input image patches (see Fig. 1).

Other token mixers. We already compared the runtime of different basic token mixers (see Fig. 2). The capabilities of token mixers can also vary drastically even within the same architecture [56]. Bidirectional (H_b) instead of causal sequence modeling with Hyena boosts the top-1 accuracy from 79.9% to 81.0%. Adding a fixed neighborhood definition (H_{px}) decreases the accuracy by 0.7%. One reason for this decrease could be that H_{px} has a strong bias towards horizontal and vertical directions due to the pronounced image border with almost $3\times$ more padded zeros than H_b. This is also reflected in the learned weights (see Fig. 4). To further test our hypothesis, we examine spatially separable convolutions restricted to the principal axes and observe a further drop in accuracy of 0.4%. Interestingly, this restriction has a similar effect as adding causality. We suspect that positional embeddings, with no preference for a particular direction, might improve H_{px}. However, this remains for future work.

Normalization for stability. Applying layer normalization [1] after the multiplication of query q and value k (see Fig. 3) improves the accuracy by 0.2% with the 310 epoch schedule. Normalizing also makes training larger network variants more stable.

Network depth and context size. While ConvNets typically require many layers for global image context, H_{px} ideally only needs one layer. We investigate this by comparing two shallow networks: H_{px}Former-S4 and ConvFormer-S4 with one block per stage. The accuracy on IN-1k with $224\,\text{px}^2$ input differs by 0.4% in favor of H_{px}Former-S4. The advantage increases to 1.1% by increasing the resolution to $512\,\text{px}^2$. This supports our hypothesis. However, building a large ERF with a hierarchical network is also effective due to the multiplicative effect on the receptive field [33].

4.3 Downstream Tasks

Object detection and instance segmentation on MS COCO. Following common practice [30, 31], we evaluate the localization properties of H_{px}Former-S18 with the Cascade Mask R-CNN [3] on MSCOCO [27]. The H_{px} feature maps are extracted at each stage and passed through an additional stage-specific layer normalization. With the MMDetection framework [4], we train the model with AdamW, a batch size of 16, a learning rate of $2e^{-5}$, and a stochastic depth of 0.4 for a $3\times$ schedule (36 epochs), halving the learning rate after 27 and 33 epochs. Moreover, we apply multi-scale training, i.e., resizing the shorter side between 480 and 800 pixels and limiting the longer side to 1333 pixels. Tab. 4 reports the results. The reference models

Table 4. Object detection and instance segmentation results on the MS COCO validation set with Cascade Mask R-CNN. Input resolution is 800×1333px (except MaxViT with 896px^2).

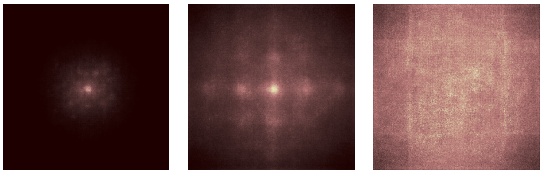
Model	#Param.	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Swin-T [30]	86M	50.4	69.2	54.7	43.7	66.6	47.3
ConvNeXt-T [31]	86M	50.4	69.1	54.8	43.7	66.5	47.3
SLaK-T [29]	-	51.3	70.0	55.7	44.3	67.2	48.1
ConvFormer-S18 [56]	-	51.5	70.7	55.8	44.6	67.8	48.2
GC-ViT-T [17]	85M	51.6	70.4	56.1	44.6	67.8	48.3
MaxViT-T [47]	69M	52.1	71.9	56.8	44.6	69.1	48.4
CAFormer-S18 [56]	-	52.3	71.3	56.9	45.2	68.6	48.8
CSWin-T [12]	80M	52.5	71.5	57.1	45.3	68.8	48.9
H_{px}Former-S18	84M	52.6	71.3	57.3	45.6	68.7	49.5

Table 5. Semantic segmentation on ADE20k validation set using UperNet [54] with an input resolution of 512px^2. MACs are calculated based on an input resolution of 512×2048px.

Model	#Param.	MACs	mIoU	mIoU_{MS}
Swin-T [30]	60M	945G	44.5	45.8
ConvNeXt-T [31]	60M	939G	46.0	46.7
GC-ViT-T [17]	58M	947G	47.0	-
SLaK-T [29]	65M	936G	47.6	-
InternImage-T [49]	59M	944G	47.9	48.1
ConvFormer-S18 [56]	54M	925G	-	48.6
H_{px}Former-S18	56M	928G	48.1	48.7
CAFormer-S18 [56]	54M	1024G	-	48.9
CSWin-T [12]	60M	959G	49.3	50.7
BiFormer-S [61]	-	-	49.8	50.8

also investigate the downstream performance of a given backbone using the same framework and share a similar computational complexity. H_{px}Former-S18 achieves the best performance in object detection with a precision of $52.6\,AP^b$, outperforming CSWin-T [12] by $0.1\,AP^b$, CAFormer-S18 [56] by $0.3\,AP^b$ and ConvNeXt-T [31] by $2.2\,AP^b$. A similar situation is observed for instance segmentation with a precision of $45.6\,AP^m$. CSwin-T, CAFormer-S18, and ConvNeXt-T are trailing by $0.3\,AP^m$, $0.4\,AP^m$, and $1.9\,AP^m$, respectively. For both tasks, the superior performance can be attributed to the better localization capabilities with higher AP_{75}, while AP_{50} is comparable or slightly lower than for the competition. One reason for better localization could be that the image borders are present for each H_{px} layer at every pixel position and serve as reference guides (see Fig. 4). Furthermore, large filters enable the model to recognize object shapes, which is more similar to human vision [11].

Semantic segmentation on ADE20k. We evaluate the downstream performance on semantic segmentation with UperNet [54] on the ADE20k benchmark [60], following related work [30]. We base our implementation on MMSegmentation [7] and train with AdamW for 160k steps with a batch size of 16, a learning rate of $1e^{-4}$, and a stochastic depth of 0.4. Tab. 5 reports the results. H_{px}Former-S18 beats Swin-T by $4.2\,\text{mIoU}$, ConvNeXt-T by $2.1\,\text{mIoU}$, SLaK-T by $0.5\,\text{mIoU}$ and InternImage-T by $0.2\,\text{mIoU}$ in the single scale setting. CSWin-T and BiFormer-S perform significantly better with an improvement of $1.2\,\text{mIoU}$ and $1.7\,\text{mIoU}$, respectively. Semantic segmentation is more difficult for H_{px}Former-S18 than instance segmentation. We assume that while global context is relevant, the model has no mechanism to filter the features in a data-driven way similar to attention [48] or more sophisticated approaches [53, 61]. Furthermore, we expect that semantic segmentation will benefit from local texture-focused operations.



(a) ConvFormer-S18 (b) SLaK-T (c) H_{px}Former-S18

Figure 5. Effective Receptive Field (ERF) of different models sampled over 50 images of size 1024px^2 from the IN-1k validation set.

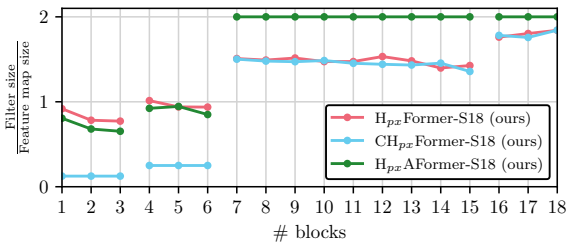


Figure 6. Learned filter sizes in H_{px}Former-S18 relative to feature map sizes at different network depths. For attention, we set the relative feature map coverage to 2, and for convolution, we use the kernel size relative to the feature map size. Note that the feature map coverage can be greater than one because the kernel size of H_{px} is almost twice the feature map size.

5 Analysis

Effective receptive field. The Effective Receptive Field (ERF) measures the influence of each input pixel on the center-most output value by tracking the gradients in a backward pass [33]. A large ERF is often associated with a better performance in vision tasks [11, 29]. We follow related work [11, 29] and compare the ERFs [22]. Fig. 5 shows the ERFs of three models. ConvFormer and SLaK have a strong local bias caused by local convolutions. SLaK features off-center areas with high gradients caused by the separable sparse convolution. H_{px}Former has a large ERF with no obvious center location, but some vertical and horizontal artifacts. This finding shows that H_{px} captures the global image context.

We hypothesize that H_{px} could benefit from an additional residual connection with a small convolution. This modification could be useful for localization and categorization tasks and has been successfully applied to attention-based networks [47, 12, 61]. We leave this study for future research.

Truncate kernels in trained models. Due to the learnable exponential decay parameter α, we can estimate the required kernel size at different depths. By setting all values of Window(t_x, t_y) to zero that are smaller than 0.05, we can measure the diameter of the non-zero values. Fig. 6 shows the mean relative feature map coverage of the token mixers in each block. H_{px} learns similar kernel sizes at the same stage regardless of other token mixers involved in earlier or later stages. The coverage in each stage stays almost constant, while former layers of a stage have slightly larger kernels. Overall, the optimal feature map coverage increases with depth, consistent with the observation of Romero et al. [38]. To further investigate the importance of filter size, we truncate the filters of a pre-trained H_{px}Former-S18 and visualize the IN-1k classification results in Fig. 7. The truncation of the third stage has the biggest impact, with an accuracy drop of 2.9%. Surprisingly, the first, second, and fourth stages are more local and can benefit from truncation, which slightly improves performance. These insights might help to construct better model layouts.

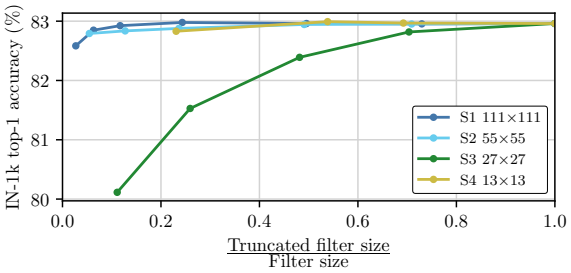


Figure 7. Impact of truncated filters in H_{px}Former-S18 on the top-1 IN-1k accuracy. For each stage (S), we modify the large kernels within the current stage by setting all values to zero that are larger than the relative filter size.

6 Conclusion

In this work, we studied whether the Hyena operator is a sufficient replacement for attention in computer vision applications. We extended Hyena to non-causal, bidirectional sequence modeling and added spatial bias with a fixed pixel neighborhood. We found the Hyena formulation useful for training extremely large convolutional kernels of up to 191×191. Analyzing trained models with these token mixers showed that bidirectional modeling is sufficient to achieve competitive categorization accuracy, while a fixed pixel neighborhood hurts the final performance. However, spatial bias with large kernels improves performance for downstream tasks that depend on exact localization. Our analysis showed that the ERF for our two-dimensional Hyena lacks the local bias present in other approaches.

In conclusion, our results suggest large, non-causal, bidirectional, spatially unbiased convolution as a promising avenue for future research. Exploring different positional embeddings for the implicit filter of H_{px} and incorporating residual connections with small convolutions could enhance texture-based categorization. Finally, future work could optimize model layouts by considering layer-wise global context requirements for potential applications in vision language tasks and video understanding, which require efficient processing.

Acknowledgements

This research has been funded by the Federal Ministry of Education and Research of Germany under grant no. 01IS22094C WEST-AI.

References

[1] L. J. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[2] S. Behnke. *Hierarchical Neural Networks for Image Interpretation*, volume 2766 of *Lecture Notes in Computer Science*. Springer, 2003.

[3] Z. Cai and N. Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018.

[4] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, et al. MMDetection: Open MMLab detection toolbox and benchmark. *CoRR*, abs/1906.07155, 2019.

[5] D. C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, pages 3642–3649, 2012.

[6] Contributors. fvcore Library. https://github.com/facebookresearch/fvcore, 2019.

[7] Contributors. MMSegmentation: OpenMMLab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

[8] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR*, pages 3008–3017, 2020.

[9] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017.

[10] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[11] X. Ding, X. Zhang, J. Han, and G. Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *CVPR*, pages 11963–11975, 2022.

[12] X. Dong, J. Bao, D. Chen, W. Zhang, N. Yu, L. Yuan, D. Chen, and B. Guo. CSWin Transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, pages 12114–12124, 2022.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[14] D. Y. Fu, E. L. Epstein, E. Nguyen, A. W. Thomas, M. Zhang, T. Dao, A. Rudra, and C. Ré. Simple hardware-efficient long convolutions for sequence modeling. In *ICML*, pages 10373–10391, 2023.

[15] M. Guo, C. Lu, Q. Hou, Z. Liu, M. Cheng, and S. Hu. SegNeXt: Rethinking convolutional attention design for semantic segmentation. In *NeurIPS*, 2022.

[16] D. Han, X. Pan, Y. Han, S. Song, and G. Huang. Flatten Transformer: Vision transformer using focused linear attention. In *ICCV*, pages 5961–5971, 2023.

[17] A. Hatamizadeh, H. Yin, G. Heinrich, J. Kautz, and P. Molchanov. Global context vision transformers. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *ICML*, pages 12633–12646, 2023.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[19] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. In *ECCV*, pages 646–661, 2016.

[20] Z. Jiang, Q. Hou, L. Yuan, D. Zhou, Y. Shi, X. Jin, A. Wang, and J. Feng. All tokens matter: Token labeling for training better vision transformers. In M. Ranzato, A. Beygelzimer, Y. N. Dauphin, P. Liang, and J. W. Vaughan, editors, *NeurIPS*, pages 18590–18602, 2021.

[21] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *ICML*, pages 5156–5165, 2020.

[22] B. J. Kim, H. Choi, H. Jang, D. G. Lee, W. Jeong, and S. W. Kim. Dead pixel test using effective receptive field. *Pattern Recognition Letters*, 167:149–156, 2023.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, pages 1106–1114, 2012.

[24] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, pages 2278–2324, 1998.

[26] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontañón. FNet: Mixing tokens with Fourier transforms. In *NAACL-HLT*, pages 4296–4313, 2022.

[27] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014.

[28] Z. Lin, C. Liu, R. Zhang, P. Gao, L. Qiu, H. Xiao, H. Qiu, C. Lin, W. Shao, K. Chen, J. Han, S. Huang, Y. Zhang, X. He, H. Li, and Y. Qiao. SPHINX: the joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *CoRR*, abs/2311.07575, 2023.

[29] S. Liu, T. Chen, X. Chen, X. Chen, Q. Xiao, B. Wu, T. Kärkkäinen, et al. More ConvNets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. In *ICLR*, 2023.

[30] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 9992–10002, 2021.

[31] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A ConvNet for the 2020s. In *CVPR*, pages 11966–11976, 2022.

[32] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[33] W. Luo, Y. Li, R. Urtasun, and R. S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *NeurIPS*, pages 4898–4906, 2016.

[34] B. McKinzie, Z. Gan, J. Fauconnier, S. Dodge, B. Zhang, P. Dufter, D. Shah, X. Du, F. Peng, F. Weers, A. Belyi, H. Zhang, K. Singh, D. Kang, A. Jain, H. Hè, M. Schwarzer, T. Gunter, X. Kong, A. Zhang, J. Wang, C. Wang, N. Du, T. Lei, S. Wiseman, G. Yin, M. Lee, Z. Wang, R. Pang, P. Grasch, A. Toshev, and Y. Yang. MM1: methods, analysis & insights from multimodal LLM pre-training. *CoRR*, abs/2403.09611, 2024.

[35] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel matters - improve semantic segmentation by global convolutional network. In *CVPR*, pages 1743–1751, 2017.

[36] M. Poli, S. Massaroli, E. Nguyen, D. Y. Fu, T. Dao, S. A. Baccus, Y. Bengio, S. Ermon, and C. Ré. Hyena Hierarchy: Towards larger convolutional language models. In *ICML*, pages 28043–28078, 2023.

[37] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4): 838–855, 1992.

[38] D. W. Romero, R. Bruintjes, J. M. Tomczak, E. J. Bekkers, M. Hoogendoorn, and J. van Gemert. FlexConv: Continuous kernel convolutions with differentiable kernel sizes. In *ICLR*, 2022.

[39] D. W. Romero, A. Kuzina, E. J. Bekkers, J. M. Tomczak, and M. Hoogendoorn. CKConv: Continuous kernel convolution for sequential data. In *ICLR*, 2022.

[40] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.

[41] S. Shleifer, J. Weston, and M. Ott. NormFormer: Improved transformer pretraining with extra normalization. *CoRR*, abs/2110.09456, 2021.

[42] S. L. Smith, A. Brock, L. Berrada, and S. De. ConvNets match vision transformers at scale. *CoRR*, abs/2310.16764, 2023.

[43] C. Szegedy, W. Liu, Y. Jia, P.Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the Inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[45] M. Tan and Q. V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019.

[46] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy. MLP-Mixer: An all-MLP architecture for vision. In *NeurIPS*, pages 24261–24272, 2021.

[47] Z. Tu, H. Talebi, H. Zhang, F. Yang, P. Milanfar, A. C. Bovik, and Y. Li. MaxViT: Multi-axis vision transformer. In *ECCV*, pages 459–479, 2022.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[49] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, X. Wang, and Y. Qiao. InternImage: Exploring large-scale vision foundation models with deformable convolutions. In *CVPR*, pages 14408–14419, 2023.

[50] Z. Wang and J. Liu. Translating math formula images to LaTeX sequences using deep neural networks with sequence-level training. *Int. J. Document Anal. Recognit.*, 24(1):63–75, 2021.

[51] R. Wightman. PyTorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[52] S. Woo, S. Debnath, R. Hu, X. Chen, Z. Liu, I. S. Kweon, and S. Xie. ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders. In *CVPR*, pages 16133–16142, 2023.

[53] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang. Vision transformer with deformable attention. In *CVPR*, pages 4784–4793, 2022.

[54] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 432–448, 2018.

[55] W. Yu, M. Luo, P. Zhou, C. Si, Y. Zhou, X. Wang, J. Feng, and S. Yan. MetaFormer is actually what you need for vision. In *CVPR*, pages 10809–10819, 2022.

[56] W. Yu, C. Si, P. Zhou, M. Luo, Y. Zhou, J. Feng, S. Yan, and X. Wang. MetaFormer baselines for vision. *IEEE TPAMI*, 46(2):896–912, 2024.

[57] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe. CutMix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6022–6031, 2019.

[58] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[59] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. In *AAAI*, pages 13001–13008, 2020.

[60] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ADE20K dataset. *IJCV*, 127(3):302–321, 2019.

[61] L. Zhu, X. Wang, Z. Ke, W. Zhang, and R. W. H. Lau. BiFormer: Vision transformer with bi-level routing attention. In *CVPR*, pages 10323–10333, 2023.

[62] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *CoRR*, abs/2401.09417, 2024.

[63] I. Zimerman and L. Wolf. Multi-dimensional hyena for spatial inductive bias. In *AISTATS*, pages 973–981, 2024.