WAVEMIX-LITE: A RESOURCE-EFFICIENT NEURAL NETWORK FOR IMAGE ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Gains in the ability to generalize on image analysis tasks for neural networks have come at the cost of increased number of parameters and layers, dataset sizes, training and test computations, and GPU RAM. We introduce a new architecture – WaveMix-Lite – that can generalize on par with contemporary transformers and convolutional neural networks (CNNs) while needing fewer resources. WaveMix-Lite uses 2D-discrete wavelet transform to efficiently mix spatial information from pixels. WaveMix-Lite seems to be a versatile and scalable architectural framework that can be used for multiple vision tasks, such as image classification and semantic segmentation, without requiring significant architectural changes, unlike transformers and CNNs. It is able to meet or exceed several accuracy benchmarks while training on a single GPU. For instance, it achieves state-of-the-art accuracy on five EMNIST datasets, outperforms CNNs and transformers in ImageNet-1K and Places-365, and achieves an mIoU of 77% on Cityscapes validation set, while using less than one-fifth the number parameters and half the GPU RAM of comparable CNNs or transformers. Our experiments show that while the convolutional elements of neural architectures exploit the shift-invariance property of images, new types of layers (e.g., wavelet transform) can exploit additional properties of images, such as scale-invariance and finite spatial extents of objects.

1 INTRODUCTION

Ever since it has been demonstrated that convolutional neural networks (CNNs) generalize better than the alternatives for image classification (Lecun et al., 1998; Krizhevsky et al., 2012), further improvements have relied on simple but effective architectural changes that allow training of deeper CNNs (Simonyan & Zisserman, 2014; Szegedy et al., 2014; He et al., 2015b; Howard et al., 2017; Hu et al., 2017; Huang et al., 2016). On the other hand, adapting transformers that were developed for natural language processing (NLP) for vision tasks was a departure from using convolutional processing, which showed that further improvements in generalization in image classification are possible by using architectures that scale with larger labeled datasets and computational resources (Zhao et al., 2020; Dosovitskiy et al., 2021). However, forsaking inductive priors suitable for images, such as 2D convolutional weight sharing, and adopting global self-attention with quadratic complexity meant that the training requirements increased to a prohibitive extent for most applications (Khan et al., 2022). Linear approximations to quadratic attention reduce the computational requirements only to some extent, and they do not resolve the problem of not having an appropriate inductive bias for images (Jeevan & Sethi, 2022).

More recently, there have been attempts to combine the inductive prior of convolutional design with the scalability of transformers in hybrid architectures for image analysis to reduce the training requirements of neural networks (pre-training dataset size, parameters, floating point operations (FLOPs)) (Graham et al., 2021; Hassani et al., 2021; Dosovitskiy et al., 2021; Wu et al., 2021; Jeevan & Sethi, 2022; Jeevan & sethi, 2022). The expensive training requirements of pure transformers also led to research into alternative token-mixing architectures that can replace self-attention (Tolstikhin et al., 2021; Lee-Thorp et al., 2021; Guibas et al., 2021; Trockman & Kolter, 2022). These works indicate that it is worth exploring priors other than shift-invariance (via convolutions) for images to reduce the training requirements with little to no sacrifice of generalization. However, most of these architectures cannot be easily adapted for multiple image sizes and other vision tasks, such as segmentation and object detection (Trockman & Kolter, 2022).

An important prior in images that has been much less explored in deep neural networks is the multi-resolution self-similarity (across spatial scales) and sparseness of edges (finite spatial extents of objects), with some notable exceptions (Mei et al., 2020). Two-dimensional discrete wavelet transforms (2D-DWT) capture this prior well and have been widely used in the pre-deep neural network era for various imaging applications, especially for compression and denoising (Lewis & Knowles, 1992; Ruikar & Doye, 2010).

In this work, we explore the use of a 2D-DWT as a spatial token mixer directly into a largely convolutional architecture, which we call *WaveMix-Lite*. WaveMix-Lite has the following benefits:

1. **Incorporate multi-resolution inductive bias without increasing training costs:** A predefined 2D-DWT allows us to mix features spatially in a multi-resolution manner without introducing extra parameters. Additionally, the 2D-DWT also scales the image dimension by positive integral powers of $\frac{1}{2} \times \frac{1}{2}$, which reduces the GPU RAM and FLOPs required per training image per forward or backward pass for the subsequent trainable layers. The resultant efficiency in terms of parameters, FLOPs, and GPU RAM allowed us to conduct experiments using a *single* GPU on Google Colab Pro+®.
2. **Impart versatility in architectural design for image recognition as well as segmentation:** Unlike other CNN and multi-head attention (MHA) models which need complete redesigning of the architecture to provide good performance in different vision tasks, the WaveMix-Lite architecture is versatile and can perform multiple vision tasks, such as image classification and semantic segmentation, without the need for special architectural modifications. We find this design simplicity and reusable block structure to be attractive in its own right.
3. **Obtain high accuracy with reduced training requirements:** For image classification and segmentation on multiple datasets WaveMix-Lite was able to match the test accuracy of some widely used convolutional and transformer architectures while using 5 to 10 times fewer parameters and 2 to 50 times lesser GPU RAM for a fixed batch size. Consequently, its training and testing throughputs were 1.5 to 6 times higher than the models compared. WaveMix-Lite also achieved state-of-the-art accuracy on 5 EMNIST datasets (Byclass, Bymerge, Letters, Digits and Balanced). Additionally, it gives compelling results on TinyImageNet, CIFAR 10 and 100, STL-10, Places-365, Caltech-256 and ImageNet-1K for classification; and Cityscapes for semantic segmentation.

After describing how WaveMix-Lite is related to previous works in Section 2, we describe its insights and details in Section 3. We compare WaveMix-Lite with other models on image classification and semantic segmentation and provide empirical evidence of its scalability and examine the importance of its components in Section 4. We conclude and list potential future directions in Section 5.

2 Related Works

Statistical properties of natural images, which have been studied for the last several decades, include shift-invariance (stationarity), scale-invariance (especially in 2D projections of a 3D world viewed from various distances), high spatial auto-correlation (monochromatic objects or regions), spatial sparseness of edges (finite spatial extent of objects), and certain chromatic contrasts (preponderance of certain colors) (Field, 1993; Ruderman, 1994; Lee, 1996; Párraga et al., 2002). Of these, only the shift-invariance has been widely exploited in neural architectures for image analysis in the form of convolutional filters (LeCun et al., 1998) and other architectural elements. There have been some exceptions that incorporate rotational-invariance for remote sensing images (Cheng et al., 2016), and multi-resolution analysis for segmentation of histopathology images (Kurian et al., 2022; van Rijthoven et al., 2021), but these methods have not been tested on general computer vision benchmarks.

Advances in CNN performance have mainly come from architectural changes with the goals of easing gradient flow to deeper layers (He et al., 2015a; Szegedy et al., 2014), or reducing parameters per layer by restricting convolutional kernel size (Simonyan & Zisserman, 2014) or their scope to only one dimension (Chollet, 2016). Attention mechanisms for space or channel (Chen et al., 2017) also seem to improve performance of CNNs, although it has not been explored why a stack of additional convolutional layers cannot model the same function as that of spatial or channel attention.

Vision transformers and hybrid architectures, inspired by their success on NLP tasks, have pushed the image classification accuracy beyond those of the largest CNNs, albeit at the cost of several times more data and network parameters (Dosovitskiy et al., 2021). Training such data hungry models with hundreds of millions of parameters requires access to large GPU clusters, which is impractical for resource-constrained applications. Reduction in the computational requirements of vision transformers have been made possible by architectural changes that provide image specific inductive biases creating hybrid models with elements including distillation (Touvron et al., 2020), convolutional embeddings (Jeevan & Sethi, 2022; Hassani et al., 2021), convolutional tokens (Wu et al., 2021), and encoding overlapping patches (Yuan et al., 2021). The quadratic complexity with respect to the sequence length (number of pixels) for vanilla transformers has also led to the search for other linear approximations of self-attention to efficiently mix tokens (Jeevan & Sethi, 2022).

Token mixers that replace the self-attention in transformers with fixed token mixing mechanisms, such as the Fourier transform (FNet), achieves comparable generalization with lower computational requirements (Lee-Thorp et al., 2021). Other token-mixing architectures have also been proposed that use standard neural components, such as convolutional layers and multi-layer perceptrons (MLPs) for mixing visual tokens. MLP-mixer (Tolstikhin et al., 2021) uses two MLP layers (cascade of 1×1 convolutions) applied first to image patch sequence and then to the channel dimension to mix tokens. ConvMixer (Trockman & Kolter, 2022) uses standard convolutions along image dimensions and depth-wise convolutions across channels to mix token information. These token mixing models perform well with lower computational costs compared to transformers without compromising generalization.

Wavelets for images: Extensive prior research has uncovered and exploited various multi-resolution analysis properties of wavelet transforms on image processing applications, including denoising (Ruikar & Doye, 2010), super-resolution (Guo et al., 2017), recognition (Mahmood et al., 2018), and compression (Lewis & Knowles, 1992). Features extracted using wavelet transforms have also been used extensively with machine learning models (Mowlaei et al., 2002), such as support vector machines and neural networks (Ranaware & Deshpande, 2016), especially for image classification (Nayak et al., 2016). Representative instances of integration with neural architectures include the following. ScatNet architecture cascades wavelet transform layers with nonlinear modulus and average pooling to extract translation invariant features that are robust to deformations and preserves high-frequency information for image classification (Bruna & Mallat, 2013). WaveCNets replaces max-pooling, strided-convolution, and average-pooling of CNNs with 2D-DWT for noise-robust image classification (Li et al., 2020a). Multi-level wavelet CNN (MWCNN) has been used for image restoration as well with U-Net architectures for better trade-off between receptive field size and computational efficiency (Liu et al., 2018). Wavelet transform has also been combined with a fully convolutional neural network for image super resolution (Kumar et al., 2017).

We propose using the two-dimensional discrete wavelet transform (2D-DWT) for token mixing. Among the different types of mother wavelets available, we used the Haar wavelet (a special case of the Daubechies wavelet (Daubechies, 1990) , also known as Db1), which is frequently used due to its simplicity and faster computation. Haar wavelet is both orthogonal and symmetric in nature, and has been used to extract basic structural information from images (Porwik & Lisowska, 2004). For even-sized images, it reduces the dimensions exactly by a factor of 2, which simplifies the designing of subsequent layers.

3 WaveMix-Lite Architecture

Image pixels have several interesting co-dependencies. The localized and stationary nature of certain image features (e.g., edges) have been exploited using linear space-invariant filters (convolutional kernels) of limited size. Scale-invariance of natural images has been exploited to some extent by pooling (LeCun et al., 1998). However, we think that scale-invariance can be better modeled by wavelet decomposition due to its natural multi-resolution analysis properties. Additionally, the finer scale of a multi-level wavelet decomposition also incorporates the idea of linear space-invariant feature extraction using convolutional filters of small support; albeit using predefined weights. The basic idea, therefore, behind our proposed architecture is to alternate between learnable spatially repeated feature extraction using convolutional layers (includes, 3×3 conv, MLP, as well as upconv layers), and fixed token mixing using 2D-DWT for a few layer blocks. Injecting fixed (unlearnable)

spatial token-mixing that also reduces the image dimensions by a factor of $\frac{1}{2} \times \frac{1}{2}$, similar to a pooling layer (which we do not use), reduces the number of computations in some of the subsequent learnable layers and increases the effective receptive field to capture distant spatial relationships more efficiently with the number of layers. This combination requires far fewer layers and parameters than using only convolutional layers with pooling. On the other hand, while transformers and other token mixers have very large effective receptive fields right from the first few layers, they do not utilize inductive priors that are suitable for images. This is where the wavelet transform plays its role in both increasing the effective receptive field at an exponential rate per layer (unlike the linear rate of convolutional layers), while still retaining the essence of convolutional design to keep the architecture flexible. Additionally, compared to pooling, wavelet is a lossless transform.

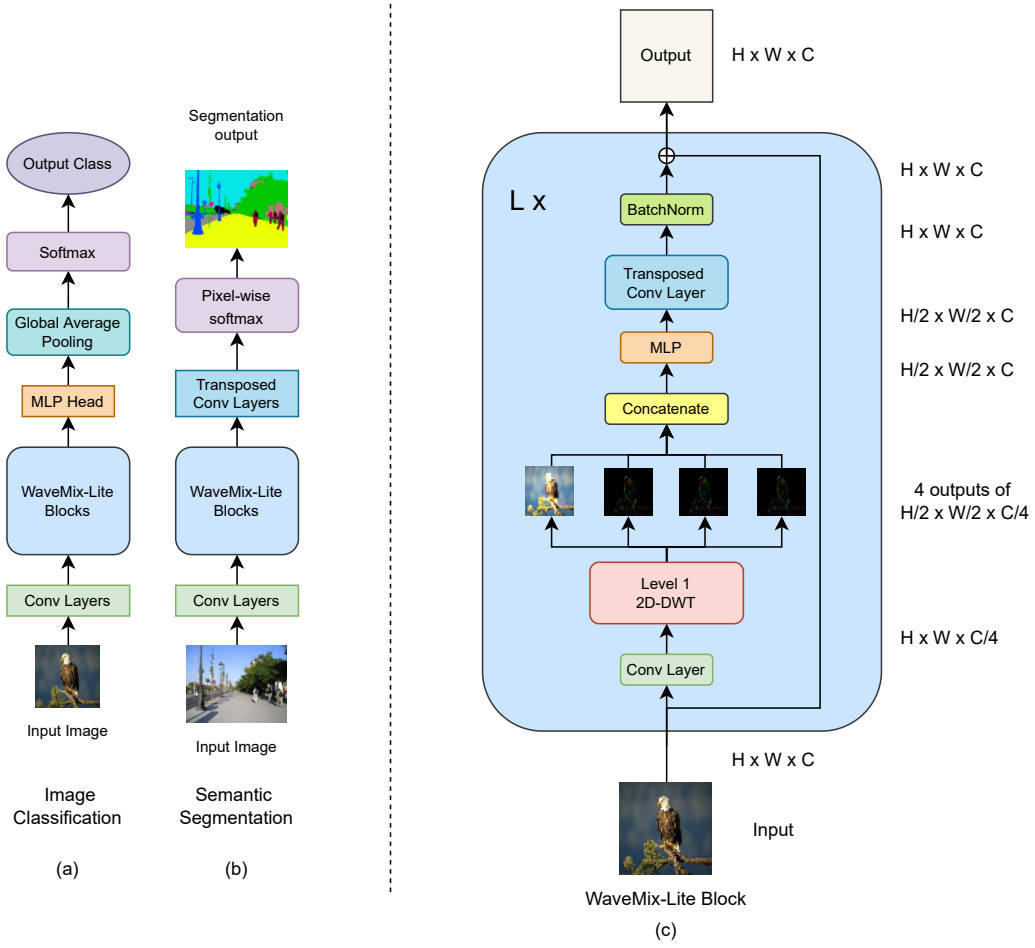

Figure 1: WaveMix-Lite architecture: Overall architecture for (a) classification and (b) semantic segmentation, along with (c) details of the WaveMix-Lite block

3.1 Overall Architecture

As shown in Figure 1, the input image is first passed through a convolutional layer that creates feature maps of the image. The use of trainable convolutions *before* the wavelet transform is a key aspect of our architecture, as it allows the extraction of only those feature maps that are suitable for the chosen wavelet family. This is followed by a series of WaveMix-Lite blocks[1]. A task specific output layer is then attached to the end. For image classification, we add an MLP head, a global average pooling layer, and a softmax layer for generating the class probabilities. For semantic segmentation, we use

[1]Our code is available at XXXX

deconvolution layers to expand the output from WaveMix-Lite block back to the input resolution. A pixel-wise softmax layer is then added to generate the class probabilities for the required number of semantic classes. For both the tasks, the core architecture remains the same and we only replace the classification head with the segmentation head. WaveMix-Lite processes image as a 2D graph and not as sequence of pixels/patches. That is, at no point in the model do we unroll the image into a sequence of pixels/patches as done in transformer models. This key feature allows transfer learning from a source to a target dataset even when the two have different image sizes or tasks.

3.2 WaveMix-Lite block

In a WaveMix-Lite block (Figure 1(c)), the input is first passed through a convolutional layer which decreases the embedding dimension by a factor of four, so that the concatenated output after 2D-DWT has the same dimension as input.[2] We only use one level 2D-DWT in WaveMix-Lite to reduce the parameters and computations. The 2D-DWT produces four output channels (one approximation and three details (Daubechies, 1990)) for each input channel. The four outputs are concatenated together (depth or channel-wise) and this output has the same number of channels as the input to the WaveMix-Lite block (embedding dimension). The output resolution (height $\times$ width) after 2D-DWT will be half that of the input; i.e., if the input is 64×64, the output will be 32×32.

The concatenated output from 2D-DWT is passed to an MLP layer (two 1×1 convolutional layers separated by a GELU non-linearity) having a multiplication factor more than one where channel mixing is performed by the MLP. Since wavelet transform reduces the image resolution by half, the GPU consumption and computations needed by the MLP significantly reduces in each layer. The image size reconciliation is performed using transposed convolutions (up-convolutions) which resizes the image back to the original input resolution. The kernel size and stride of deconvolutional layers were chosen such that the output has the same size as the input to WaveMix-Lite block. We chose deconvolutional layer rather than an inverse 2D-DWT because the former is much faster and consumes less GPU than the latter. The outputs from the deconvolutional layers are then passed through batch normalization. A residual connection (He et al., 2015a) is provided within each WaveMix-Lite block so that the model can be made deeper with a larger number of blocks, if necessary.

4 Experiments and Results

4.1 Datasets

To demonstrate the applicability of WaveMix-Lite for image classification, we used multiple types of publicly available (under MIT Licenses) datasets based on the number of images and image size. Small datasets of smaller image sizes included CIFAR-10, CIFAR-100 (Krizhevsky, 2009), EMNIST (Cohen et al., 2017), Fashion MNIST (Xiao et al., 2017), and SVHN (Netzer et al., 2011). Small datasets of larger image sizes included STL-10 (Coates et al., 2011), Caltech-256 (Griffin et al., 2007) and Tiny ImageNet (Le & Yang, 2015). We also used larger datasets with larger images sizes such as, ImageNet-1K (Deng et al., 2009), Places-365 (Zhou et al., 2017) and iNaturalist2021-10k (iNAT mini) (Horn et al., 2021). We used Cityscapes (Cordts et al., 2016) dataset for semantic segmentation experiments and evaluated performance in the Cityscapes validation dataset.

4.2 Models compared

WaveMix-Lite was compared with various other CNNs, transformers, and token-mixing models. These include ResNets (He et al., 2015a), MobileNet (Sandler et al., 2018), UNets and DeepLabV2 as CNNs; ViT (vision transformer) (Dosovitskiy et al., 2021), hybrid ViN (vision Nystromformer) (Jeevan & Sethi, 2022), CPV (convolutional performer for vision) (Jeevan & sethi, 2022), CCT (compact convolutional transformer) (Hassani et al., 2021), CvT (convolutional vision transformer) (Wu et al., 2021), and SegFormer (Xie et al., 2021) as transformers; and FNet (Lee-Thorp et al., 2021), ConvMixer (Trockman & Kolter, 2022) and MLP-Mixer (Tolstikhin et al., 2021) as token-mixers. Results of the other models that were directly taken from their original papers are cited in results tables.

[2]Base code: `https://pytorch-wavelets.readthedocs.io/en/latest/readme.html`

Table 1: Image classification on ImageNet-1K dataset shows improved accuracy as well as throughput due to decreased parameter count and GPU RAM consumption by WaveMix-Lite (arrows show desired directions)

Architecture	Top-1 Accu. (%) ↑	# Param. ↓	GPU RAM for batch size 64 ↓	Max batch size in 16 GB GPU ↑	Throughput (im/s) Train ↑	Test ↑
ResNet-18	55.12	11.7 M	2.7 GB	384	450	439
ResNet-34	57.02	21.8 M	3.1 GB	330	414	410
ResNet-50	61.76	25.6 M	6.2 GB	164	638	617
ResNet-101	64.60	44.5 M	9.6 GB	106	487	725
ResNet-152	65.86	60.2 M	12.7 GB	80	344	758
MobileNetv3-small	51.57	2.5 M	1.4 GB	751	255	229
MobileNetv3-large	58.89	5.5 M	3.5 GB	289	492	481
ViT-B-16	39.53	86.6 M	10.0 GB	102	140	420
ViT-B-32	30.11	88.2 M	2.2 GB	474	1,595	1,613
CPV-256/5x4	63.02	6.7 M	12.9 GB	79	364	728
ConvMixer-512/12	60.24	4.2 M	10.8 GB	94	292	735
ConvMixer-512/16	62.24	5.4 M	14.1 GB	72	220	725
ConvMixer-1024/12	64.13	14.6 M	23.6 GB	43	251	667
WaveMix-Lite-128/8	54.12	3.9 M	4.5 GB	228	1,242	1,724
WaveMix-Lite-144/8	55.42	4.6 M	4.9 GB	208	943	1,714
WaveMix-Lite-256/24	**67.71**	32.4 M	19.2 GB	53	222	649

For model notation we use the format *Model_Name-Embedding Dimension/Layers×Heads* for transformers and exclude the *heads* for the other architectures. For example, CCT with embedding dimension of 128 having 4 layers and 4 heads is labelled as CCT-$128/4 \times 4$.

4.3 Implementation details

We trained models using AdamW optimizer ($\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$) with a weight decay coefficient of 0.01 during initial epochs and then used SGD (stochastic gradient descent) with learning rate of 0.001 and momentum $= 0.9$ during the final 20 epochs (Keskar & Socher, 2017). We used automatic mixed precision in PyTorch during training to optimize speed and memory consumption. Almost all experiments were done with a 16 GB Tesla V100-SXM2 GPU available in Google Colab Pro+®. *No image augmentations were used while training the models*. Maximum number of epochs in all experiments was set to 150. GPU usage for a batch size of 64 was reported for image classification along with top-1% accuracy from best of three runs with random initialization based on prevailing protocols (Hassani et al., 2021). We report the semantic segmentation performance using mean intersection over union (mIoU) metric. Cross-entropy loss was used for image classification and pixel-wise focal loss was used for semantic segmentation. Few segmentation experiments were run on A100-SXM4 GPU in Google Colab Pro+®. Due to resource constraints, for segmentation experiments in 16 GB V100 GPU, the original 1024×2048 image was resized to 256×512 and for 40 GB A100 GPU, it was resized to 512×1024. We also adjusted the stride of the initial convolutional layers in all WaveMix-Lite models that handled high-resolution images to ensure that smaller side of input was always 64 before it reached WaveMix-Lite blocks. The classification head of ConvMixer was replaced with a segmentation head similar to WaveMix-Lite for segmentation. *No pre-training was performed on any of the WaveMix-Lite models*. For training ImageNet-1K, we used the ResNets, MobileNets and ViT models available in pytorch library (Paszke et al., 2019) without using pre-trained weights.

4.4 Image classification

Table 1 shows the performance of WaveMix-Lite compared to the other architectures on image classification using supervised learning on ImageNet-1K on a single GPU within a reasonable time. Similarly, for the transformer models, in order to train on a single GPU, fewer layers gave better results, for which we used smaller patch sizes. We see that WaveMix-Lite models outperform ResNets, transformers, hybrid xformers, and ConvMixer while requiring lesser GPU RAM and number of parameters. WaveMix-Lite does not need a large number of parameters to give performance

comparable to large ResNets and they require very less GPU RAM compared to transformer models for achieving similar results.

Even though convolution has been widely regarded as a GPU-efficient operation, the need for deeper architectures have necessitated the use of networks having over tens to hundreds of layers for achieving high generalization. Even though a single convolutional operation is comparatively cheaper than a 2D-DWT, we can achieve generalization comparable to deep convolutional networks with far fewer layers of the wavelet transforms. This ability of the wavelet transform to provide competitive performance without needing large number of layers helps in improving the efficiency of the network by consuming much lesser GPU RAM than deep convolutional models like ResNets. We also observe that deeper WaveMix-Lite models perform better and this suggests that even further scale-up of WaveMix-Lite in multi-GPU setting could be possible. Even shallow WaveMix-Lite models are performing comparable to ResNets and MobileNets, and they are much 2 times faster in training 4 times faster in inference. Even larger WaveMix-Lite models provide faster inference and better performance compared to other models. ConvMixers are parameter efficient and provide good accuracy, but they consume much higher GPU RAM compared to WaveMix-Lite.

Table 2: Results for Image classification on various datasets show improved accuracy compared to ResNets when trained on a single 16 GB V100 GPU.

Models	STL-10 96×96	SVHN 32×32	Caltech-256 256×256	Places-365 256×256	iNAT-2021 256×256
ResNet-18	70.41	97.40	52.97	48.74	26.35
ResNet-34	68.07	97.47	50.92	49.02	31.02
ResNet-50	66.04	97.32	49.97	49.80	33.14
WaveMix-Lite	**70.88**	**98.42**	**54.62**	**56.45**	**33.23**

WaveMix-Lite outperforms ResNets in all the datasets tested as shown in Table 2. We also achieved State-of-the-art accuracy of 56.45% on Places-365 Standard dataset in the category of models which does not use pre-training on larger datasets, out-performing previous best (Wang et al., 2020). The results of WaveMix-Lite on several smaller resolution image datasets are provided in the Appendix. The lower performance of standard models due to lack of image augmentations and learning rate schedulers is also discussed in Appendix.

We can see from Table 3 that WaveMix-Lite models outperform ResNets on all datasets (28×28) tested. It also establishes a new state-of-the-art by outperforming the previous best results (Kabir et al., 2020; Pad et al., 2020) by 0.01, 0.08, 0.01, 0.31 and 0.01 percentage points, respectively for Balanced, Letters, Digits, Byclass and Bymerge subsets within EMNIST (Cohen et al., 2017).

4.5 SEMANTIC SEGMENTATION

WaveMix-Lite can be directly used for semantic segmentation by replacing the classifier head with deconvolution layers and a linear layer to generate the segmentation maps. On the other hand, architectural changes – such as encoder-decoder and UNet structures (Ronneberger et al., 2015) – are required for base CNNs and transformers, including SegFormer (Xie et al., 2021). The WaveMix-Lite performs on par with the other models on half-resolution Cityscapes validation set. We can

Table 3: WaveMix-Lite outperforms ResNets (Gavrikov & Keuper, 2022) for image classification on various EMNIST, MNIST, and Fashion MNIST datasets (28×28) and achieves SOTA on 5 datasets

Models	Byclass	Bymerge	Letters	Digits	Balanced	MNIST	Fashion MNIST
ResNet-18	87.98	91.09	94.76	99.67	89.00	99.64	93.97
ResNet-34	88.10	91.13	95.04	99.68	89.17	99.60	93.91
ResNet-50	88.18	91.29	94.64	99.62	89.76	99.56	93.81
WaveMix-Lite	**88.43**	**91.80**	**95.96**	**99.80**	**91.06**	**99.75**	**94.32**

Table 4: Results for semantic segmentation on Cityscapes validation dataset show improved mIoU by WaveMix-Lite without compromising throughput

Architecture	mIoU ↑	# Param. ↓	GPU RAM for batch size 64 ↓	Max batch size for 16 GB ↑	Throughput (im/s) Train ↑	Test ↑	Notes
UNet Siam et al. (2018)	57.90	28.9 M	-	-	1	-	ResNet-18 Encoder
UNet Siam et al. (2018)	61.00	9.0 M	-	-	-	-	MobileNetV2 Encoder
DeepLabV2-CRF Chen et al. (2016)	71.40	20.5 M	-	-	5	-	ResNet-101, Augmentations
SegFormer (MiT-B0) Xie et al. (2021)	71.90	3.4 M	-	-	47	-	Augmentations, ImageNet Pretrained
ConvMixer-512/16	53.40	7.8 M	42 GB	24	10	11	256×512, 16 GB
SegFormer (MiT-B0)	62.56	7.7 M	232 GB	4	16	16	512×1024, 40 GB
WaveMix-Lite 128/8	63.33	2.9 M	19 GB	55	18	18	256×512, 16 GB
WaveMix-Lite 256/12	67.46	16.9 M	38 GB	25	11	12	256×512, 16 GB
WaveMix-Lite 256/16	**71.75**	44.1 M	49 GB	21	9	11	256×512, 16 GB
WaveMix-Lite 256/16	**76.79**	22.2 M	189 GB	6	14	16	512×1024, 40 GB

Table 5: WaveMix-Lite needs very few parameters to achieve the benchmark results on the tasks mentioned below compared to other architectures

Task	Model	Parameters	Expansion
99% accuracy on MNIST	WaveMix-Lite-8/10	3,566	Upsampling
90% accuracy on Fashion MNIST	WaveMix-Lite-8/5	7,156	Deconvolution
80% accuracy on CIFAR-10	WaveMix-Lite-32/7	37,058	Upsampling
90% accuracy on CIFAR-10	WaveMix-Lite-64/6	520,106	Deconvolution

see from Table 4 that WaveMix-Lite performs better than deep architectures like DeepLabV2 (Chen et al., 2016) and SegFormer model which uses an encoder pre-trained on ImageNet-1K dataset. The low mIoU obtained by replacing the classification head of ConvMixer (Trockman & Kolter, 2022) with segmentation head similar to WaveMix-Lite shows that token-mixing architectures which work well for classification cannot translate that performance in segmentation tasks without significant architectural modifications. This shows the versatility of our model which can provide high performance efficiently in multiple tasks. See Appendix for detailed results.

4.6 Parameter efficiency

Since WaveMix-Lite heavily uses unlearnable token-mixers, it uses much fewer parameters compared to the commonly used architectures. Table 5 shows that WaveMix-Lite can achieve all the tasks mentioned with far fewer parameters compared to previous models (Jha et al., 2021; Wu, 2018). We can further reduce the parameter count of WaveMix-Lite by replacing the deconvolution layers with Upsampling layers using unlearnable interpolation techniques (e.g., IDWT, bilinear or bicubic).

4.7 Ablation studies

We performed multiple ablations on WaveMix-Lite using the CIFAR-10 dataset to understand the effect of each type of layer on performance.

When we removed the 2D-DWT layers from WaveMix-Lite, the GPU RAM requirement of the model increased by 61.8 % and accuracy fell by 5%. This is due to the MLP receiving the full resolution instead of the half-resolution feature map from 2D-DWT.

Replacing the 2D-DWT with the real part of a 2D-discrete Fourier transform showed 12% decrease in accuracy along with 73% increase in GPU consumption as the Fourier transform also does not downscale an image. Additionally, the Fourier transform has global and spatially smoothly varying

kernels, which do not model objects in images in a sparse manner. Objects have finite and abruptly-ending spatial extents, which are better modeled by wavelet functions that have the local kernels (finite support set) with sharp transitions.

GPU RAM consumption increased by 7.8% and accuracy decreased by 5% when we replaced the 2D-DWT with 2D-MaxPooling, indicating that the loss of information by the latter hurts generalization.

Additional ablation studies on the number of layers, embedding dimension, MLP dimension, and multi-level 2D-DWT are included in the Appendix.

5 Conclusions, Future Directions, and Impact

We proposed a novel and versatile neural architecture – WaveMix-Lite – that can generalize at par with both self-attention networks (transformers), CNNs, and their hybrids for image classification and segmentation, while needing fewer parameters, GPU RAM, and FLOPs. In addition to convolutions, WaveMix-Lite uses a 2D wavelet transform for token mixing in images, which exploits additional image priors, such as scale-invariance and finite spatial extents of objects. It is also better tailored for computer vision applications than transformers as it handles the data in a 2D format without unrolling it as a sequence. It is easy to adapt WaveMix-Lite for different image sizes and tasks, such as image classification and semantic segmentation, without changing its core architecture, as shown by our experiments on multiple datasets and tasks.

Limitations: Although our own lack of access to a large GPU cluster inspired the objective of designing a neural architecture that can generalize well with limited resources, scalability of WaveMix-Lite to larger datasets (in terms of image size and the number of images) needs to be tested with larger compute resources. Additionally, WaveMix-Lite should also be tested on other vision tasks, such as object detection and image enhancement (e.g., super-resolution, deblurring, denoising, and inpainting) using task-specific architectural variations.

Impact: The high accuracy of image classification by transformers and CNNs comes with high costs in terms of training data, computations, GPU RAM, hardware costs, form factors, and energy consumption Li et al. (2020b). Our research shows that architectural innovations can still reduce these computational requirements by exploiting priors of natural images, such as scale-invariance and finite spatial extents (in addition to shift-invariance that is already exploited by convolutional design elements), without sacrificing accuracy. Such architectures would also be more environment-friendly due to lower energy consumption and accessible for modification to more researchers who may not have access to large computational resources. We hope that our research has shed light into the less traversed area of resource-efficient models that exploit more priors of natural images.

This work can be extended in several directions, some of which are mentioned in its limitations above. Exploitation of additional properties of images and videos can also be explored. Instead of using a fixed function (e.g., Haar), we can also make the learning of the wavelet function itself as a part of the training process.

References

Joan Bruna and Stephane Mallat. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1872–1886, 2013. doi: 10.1109/TPAMI.2012.230.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2016. URL `https://arxiv.org/abs/1606.00915`.

Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Gong Cheng, Peicheng Zhou, and Junwei Han. Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415, 2016.

François Chollet. Xception: Deep learning with depthwise separable convolutions, 2016. URL https://arxiv.org/abs/1610.02357.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 215–223, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/coates11a.html.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 2921–2926, 2017. doi: 10.1109/IJCNN.2017.7966217.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding, 2016. URL https://arxiv.org/abs/1604.01685.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Transactions on Information Theory*, 36(5):961–1005, 1990. doi: 10.1109/18.57199.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

DJ Field. Scale-invariance and self-similar 'wavelet'transforms: an analysis of natural scenes and mammalian visual systems. *Wavelets, fractals, and Fourier transforms*, pp. 151–193, 1993.

Paul Gavrikov and Janis Keuper. Cnn filter db: An empirical investigation of trained convolutional filters, 2022. URL https://arxiv.org/abs/2203.15331.

Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference, 2021.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset, 2007.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers, 2021. URL https://arxiv.org/abs/2111.13587.

Tiantong Guo, Hojjat Seyed Mousavi, Tiep Huu Vu, and Vishal Monga. Deep wavelet prediction for image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1100–1109, 2017. doi: 10.1109/CVPRW.2017.148.

Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015b. URL https://arxiv.org/abs/1512.03385.

Grant Van Horn, Elijah Cole, Sara Beery, Kimberly Wilber, Serge Belongie, and Oisin Mac Aodha. Benchmarking representation learning for natural world image collections, 2021.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. URL https://arxiv.org/abs/1704.04861.

Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-excitation networks, 2017. URL https://arxiv.org/abs/1709.01507.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2016. URL https://arxiv.org/abs/1608.06993.

Pranav Jeevan and Amit Sethi. Resource-efficient hybrid x-formers for vision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2982–2990, January 2022.

Pranav Jeevan and Amit sethi. Convolutional xformers for vision, 2022. URL https://arxiv.org/abs/2201.10271.

Debesh Jha, Anis Yazidi, Michael A. Riegler, Dag Johansen, Håvard D. Johansen, and Pål Halvorsen. Lightlayers: Parameter efficient dense and convolutional layers for image classification, 2021. URL https://arxiv.org/abs/2101.02268.

H. M. D. Kabir, Moloud Abdar, Seyed Mohammad Jafar Jalali, Abbas Khosravi, Amir F. Atiya, Saeid Nahavandi, and Dipti Srinivasan. Spinalnet: Deep neural network with gradual input. *ArXiv*, abs/2007.03347, 2020.

Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from adam to sgd, 2017.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys*, jan 2022. doi: 10.1145/3505244. URL https://doi.org/10.1145%2F3505244.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

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

Neeraj Kumar, Ruchika Verma, and Amit Sethi. Convolutional neural networks for wavelet domain super resolution. *Pattern Recognition Letters*, 90:65–71, 2017.

Nikhil Cherian Kurian, Amit Lohan, Gregory Verghese, Nimish Dharamshi, Swati Meena, Mengyuan Li, Fangfang Liu, Cheryl Gillet, Swapnil Rane, Anita Grigoriadis, and Amit Sethi. Deep multi-scale u-net architecture and noise-robust training strategies for histopathological image segmentation, 2022. URL https://arxiv.org/abs/2205.01777.

Ya Le and X. Yang. Tiny imagenet visual recognition challenge, 2015.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pp. 2278–2324, 1998. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.7665.

Tai Sing Lee. Image representation using 2d gabor wavelets. *IEEE Transactions on pattern analysis and machine intelligence*, 18(10):959–971, 1996.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms, 2021.

A.S. Lewis and G. Knowles. Image compression using the 2-d wavelet transform. *IEEE Transactions on Image Processing*, 1(2):244–250, 1992. doi: 10.1109/83.136601.

Qiufu Li, Linlin Shen, Sheng Guo, and Zhihui Lai. Wavelet integrated cnns for noise-robust image classification, 2020a.

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In *International Conference on Machine Learning*, pp. 5958–5968. PMLR, 2020b.

Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration, 2018.

Maria Mahmood, Ahmad Jalal, and Hawke A. Evans. Facial expression recognition in image sequences using 1d transform and gabor wavelet transform. In *2018 International Conference on Applied and Engineering Mathematics (ICAEM)*, pp. 1–6, 2018. doi: 10.1109/ICAEM.2018.8536280.

Yiqun Mei, Yuchen Fan, Yulun Zhang, Jiahui Yu, Yuqian Zhou, Ding Liu, Yun Fu, Thomas S. Huang, and Humphrey Shi. Pyramid attention networks for image restoration, 2020. URL https://arxiv.org/abs/2004.13824.

A. Mowlaei, K. Faez, and A.T. Haghighat. Feature extraction with wavelet transform for recognition of isolated handwritten farsi/arabic characters and numerals. In *2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628)*, volume 2, pp. 923–926 vol.2, 2002. doi: 10.1109/ICDSP.2002.1028240.

Deepak Ranjan Nayak, Ratnakar Dash, and Banshidhar Majhi. Brain mr image classification using two-dimensional discrete wavelet transform and adaboost with random forests. *Neurocomputing*, 177:188–197, 2016.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning, 2011.

Pedram Pad, Simon Narduzzi, Clement Kundig, Engin Turetken, Siavash A. Bigdeli, and L. Andrea Dunbar. Efficient neural vision systems based on convolutional image acquisition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

CA Párraga, T Troscianko, and DJ Tolhurst. Spatiochromatic properties of natural images and human vision. *Current biology*, 12(6):483–487, 2002.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Piotr Porwik and Agnieszka Lisowska. The haar-wavelet transform in digital image processing: Its status and achievements. *Machine graphics & vision*, 13:79–98, 2004.

Preeti N. Ranaware and Rohini A. Deshpande. Detection of arrhythmia based on discrete wavelet transform using artificial neural network and support vector machine. In *2016 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1767–1770, 2016. doi: 10.1109/ICCSP.2016.7754470.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Daniel L Ruderman. The statistics of natural images. *Network: computation in neural systems*, 5 (4):517, 1994.

Sachin Ruikar and D D Doye. Image denoising using wavelet transform. In *2010 International Conference on Mechanical and Electrical Technology*, pp. 509–515, 2010. doi: 10.1109/ICMET.2010.5598411.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2018. URL https://arxiv.org/abs/1801.04381.

Mennatullah Siam, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, and Martin Jagersand. Rtseg: Real-time semantic segmentation comparative study, 2018. URL https://arxiv.org/abs/1803.02758.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. URL https://arxiv.org/abs/1409.1556.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. URL https://arxiv.org/abs/1409.4842.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers: distillation through attention, 2020. URL https://arxiv.org/abs/2012.12877.

Asher Trockman and J Zico Kolter. Patches are all you need?, 2022. URL https://openreview.net/forum?id=TVHS5Y4dNvM.

Mart van Rijthoven, Maschenka Balkenhol, Karina Siliņa, Jeroen van der Laak, and Francesco Ciompi. Hooknet: Multi-resolution convolutional neural networks for semantic segmentation in histopathology whole-slide images. *Medical Image Analysis*, 68:101890, 2021. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2020.101890. URL https://www.sciencedirect.com/science/article/pii/S1361841520302541.

Qilong Wang, Jiangtao Xie, Wangmeng Zuo, Lei Zhang, and Peihua Li. Deep CNNs meet global covariance pooling: Better representation and generalization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020. doi: 10.1109/tpami.2020.2974833. URL https://doi.org/10.1109%2Ftpami.2020.2974833.

Chai Wah Wu. Prodsumnet: reducing model parameters in deep neural networks via product-of-sums matrix decompositions, 2018. URL https://arxiv.org/abs/1809.02209.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers, 2021.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017. URL http://arxiv.org/abs/1708.07747.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers, 2021. URL https://arxiv.org/abs/2105.15203.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet, 2021. URL https://arxiv.org/abs/2101.11986.

Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition, 2020. URL https://arxiv.org/abs/2004.13621.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

A FEEDFORWARD DIMENSION AND MLP MULTIPLICATION FACTOR

The feedforward dimension (ff) is the dimension of the embeddings of ouput from the MLP layer before it is passed to the deconvolution layer. The deconvolution layer then changes the embedding dimension back to the value set in the model name. Unless otherwise mentioned, the value of feedforward dimension is set by default as the embedding dimension specified in the model name. Using a value higher than embedding dimension as ff dimension increases the number of parameters of the model and GPU consumption. Feedforward dimension is different from the MLP mulitplication factor (mul) which describes the increase in embedding dimension within the MLP layer (the first 1×1 convolution increases it by a factor and the second 1×1 convolution decreases it after is passes through the GELU activation). For example, MLP multiplication factor of 2 in a WaveMix-Lite-128 will use the first 1×1 convolutional layer inside MLP to increase the embedding dimension from 128 to 256. After the GELU activation, the second 1×1 convolutional layer inside MLP will decreases the embedding dimension back to 128. If we specify the ff dimension to be different from one provided in model name, then the second 1×1 convolutional layer inside MLP will change it to the ff dimension as specified. Unless otherwise mentioned, an MLP multiplication factor of 2 was used in all the models.

B IMAGE CLASSIFICATION

B.1 RESULTS ON SMALLER RESOLUTION IMAGE DATASETS

Table 6: Results for image classification on small datasets (32x32, 64x64) show improved accuracy as well as decreased parameter count and GPU RAM consumption by WaveMix-Lite

Model	#Param. ↓	GPU RAM for batch size 64 ↓	Accuracy (%) ↑ CIFAR-10	CIFAR-100	TinyImNet
ResNet-18 (Hassani et al., 2021)	11.20 M	1.2 GB	90.27	63.41	48.11
ResNet-34 (Hassani et al., 2021)	21.30 M	1.4 GB	90.51	64.52	45.60
ResNet-50 (Hassani et al., 2021)	25.20 M	3.3 GB	90.60	61.68	48.77
MobileNetV2 (Hassani et al., 2021)	8.72 M	-	91.02	67.44	-
ViT-128/4×4	0.53 M	13.8 GB	56.81	30.25	26.43
ViT-384/12x6 (Hassani et al., 2021)	85.60 M	-	76.42	46.61	-
ViT-Lite-256/6x4 (Hassani et al., 2021)	3.19 M	-	90.94	69.20	-
HybridViN-128/4×4	0.62 M	4.8 GB	75.26	51.44	34.05
CCT-128/4×4	0.91 M	15.8 GB	82.23	57.09	39.05
CvT-128/4×4	1.12 M	15.4 GB	79.93	48.29	40.69
MLP-Mixer-512/8	2.41 M	1.0 GB	72.22	44.23	26.83
WaveMix-Lite-16/7	0.04 M	0.1 GB	64.98	23.03	19.15
WaveMix-Lite-32/7	0.15 M	0.3 GB	84.67	46.89	34.34
WaveMix-Lite-64/7	0.60 M	0.6 GB	87.81	62.72	46.31
WaveMix-Lite-128/7	2.42 M	1.1 GB	91.08	68.40	52.03
WaveMix-Lite-144/7	3.01 M	1.2 GB	**92.97**	68.86	52.38
Wavemix-Lite-160/13	6.90 M	9.4 GB	-	-	**54.76**
WaveMix-Lite-256/7	9.62 M	2.3 GB	90.72	**70.20**	51.37

In Table 6 we see that on CIFAR and TinyImageNet datasets, WaveMix-Lite performs much better than the other models, giving accuracy higher than ResNets and MobileNets with 4 to 10 times fewer parameters and less GPU consumption. GPU consumption of WaveMix-Lite is sometimes 50 times lower for similar performance when compared to transformer models.

B.2 AUGMENTATIONS AND LEARNING RATE TUNING

The previously reported results for the other architectures include the effect of various well-intentioned incremental training methods (tips and tricks), including RandAugment, mixup, CutMix, random erasing, gradient norm clipping, learning rate warmup and cooldown. These additional methods improve the results of the core architectures trained using simple methods by a few percentage points each. For example, Mixup, Cutmix, Random Erasing, RandAug, Random

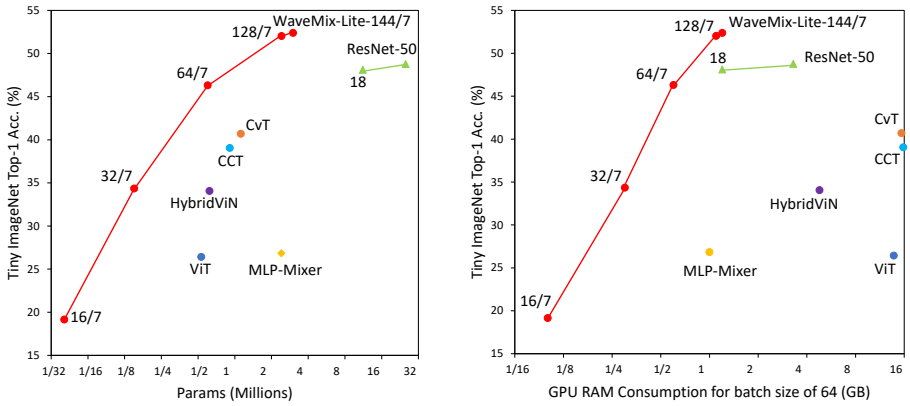

Figure 2: Accuracy scales more efficiently with parameters and GPU RAM for WaveMix-Lite compared to transformers and ResNets

Table 7: State-of-the-art (SOTA) models for the 5 EMNIST Datasets.

Dataset	Model	#Params	SOTA Accu. (%)
By_Class	WaveMix-Lite-128/7	2.4 M	**88.43**
Balanced	WaveMix-Lite-128/7	2.4 M	**91.06**
Letters	WaveMix-Lite-112/16	4.1 M	**95.96**
Digits	WaveMix-Lite-112/16	4.1 M	**99.80**
By_Merge	WaveMix-Lite-128/16 (ff =256)	10.1 M	**91.80**

Scaling and Gradient Norm Clipping improved accuracy of ConvMixer by 9.55 percentage points in image classification Trockman & Kolter (2022). However, experimenting with these additional training methods requires extensive hyperparameter tuning. On the other hand, by excluding these methods, we were able to compare the contribution of the base architectures in a uniform manner. The accuracy obtained in our experiments for the other architectures are thus slightly lower than the previously reported numbers, but the results are still within the expected range when such training methods are not used. We noticed an improvement of 1.16 percentage point (68.87) just by using RandAugment (Cubuk et al., 2019) in ImageNet-1K classification using WaveMix-Lite. WaveMix-Lite should be able to outperform the reported results of other models when we use all the augmentations and learning rate tuning mentioned above.

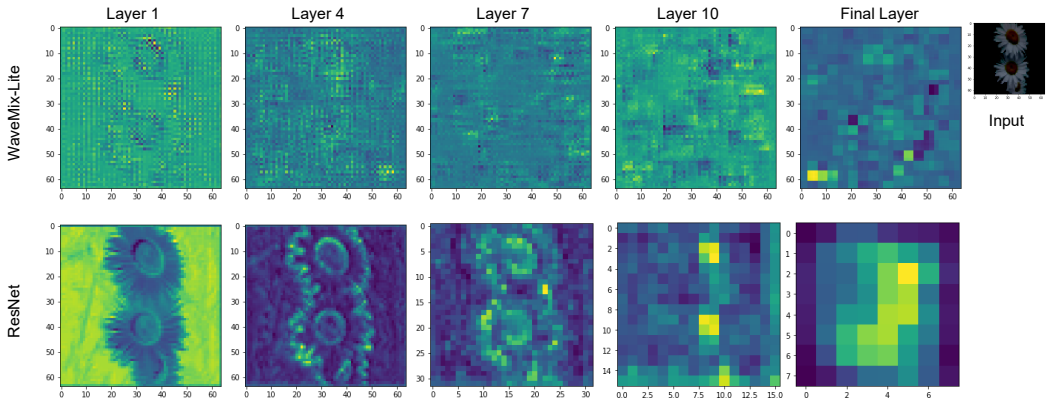

Figure 3: Visualisation of outputs after each layer shows that the image representation is different in WaveMix-Lite and ResNet.

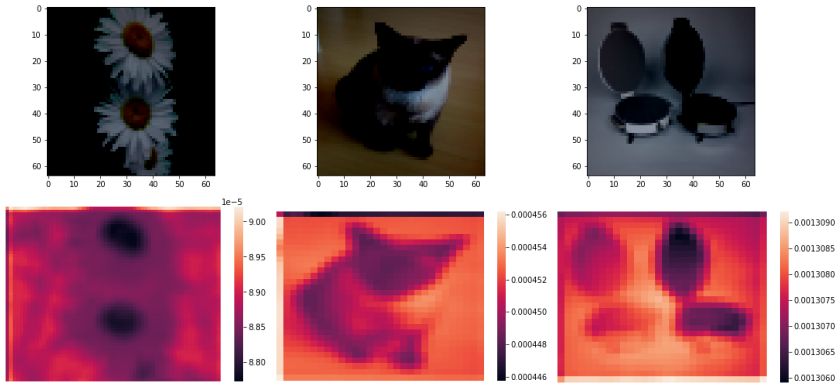

Figure 4: The results of occlusion analysis to find the significance of each pixel in the output decision shows that WaveMix-Lite identifies all the right pixels in an image for making the classification decision. The darker pixels in the image contribute more to the decision than lighter ones. The numbers show the probability of class output when the pixel is occluded.

Table 8: Variation of performance and resource consumption of WaveMix-Lite-144/7 on classification of CIFAR-10 dataset using different levels of 2D-DWT.

Level of 2D-DWT	Accu. (%)	# Params	GPU for batch size of 1024
1	**91.61**	3 M	19.6 GB
2	87.39	5.9 M	14.2 GB
3	78.07	15.2 M	13.7 GB
4	65.40	47.7 M	13.2 GB

B.3 Ablation Studies

Influence of levels of 2D-DWT. Table 8 shows the variation of performance and resource consumption as we use higher levels of 2D-DWT. Level-1 2D-DWT reduced image resolution by half, reducing the computational cost and GPU consumption compared to convolutional layers. Using further levels of 2D-DWT could further reduce the image resolution to one-fourth, one-eight and so on which can further reduce the computational costs. But the deconvolution layer used to resize the output back to input size will need a lot more parameters. This will consume more resources in terms of GPU and time. Each increment in the level of 2D-DWT results in doubling of the number of parameters, but provides only a very small reduction in GPU consumption, especially when we go to higher levels of 2D-DWT. In each of the higher level decompositions, when the approximation and detail coefficients are concatenated as a tensor, the noise intensity in the detail coefficients would be stronger than that of useful details (object edge, texture, or contour, etc.) which could be the cause of degradation in the performance.

Influence of number of layers. The performance of WaveMix-Lite models generally improve as the number of layer increases. The behaviour observed in smaller datasets show that the accuracy increases with increase in number of layers, peaks at a particular value and then do not show any increase for any further addition of layers.

Influence of the Embedding Dimension. Our experiments showed that increasing the embedding dimension of a model usually improved the model performance, but the resource-utilization also increased significantly. Doubling the embedding dimension of model from 128 to 256 results in an increase of parameter count by more than three times and doubles the GPU RAM consumption.

Table 9: Results for semantic segmentation using WaveMix-Lite models on Cityscapes validation dataset at different image resolutions in V100 and A100 GPUs.

Architecture	mIoU ↑	# Param. ↓	Max batch size for 16 GB ↑	Throughput (im/s) Train ↑	Test ↑
Image resolution 256×512 on 16 GB V100 GPU					
WaveMix-Lite-128/8	63.33	2.9 M	55	18	18
WaveMix-Lite-128/20 (mul 3)	67.76	7.5 M	32	17	17
WaveMix-Lite-160/12	65.08	6.6 M	45	18	18
WaveMix-Lite-192/12	65.92	9.5 M	40	19	19
WaveMix-Lite-224/12	66.67	12.9 M	32	18	18
WaveMix-Lite-256/7 (ff 160)	65.30	7.2 M	45	18	18
WaveMix-Lite-256/7 (ff 160, mul 3)	64.34	7.9 M	40	19	19
WaveMix-Lite-256/7 (ff 192, mul 3)	65.27	8.9 M	30	17	17
WaveMix-Lite-256/12 (ff 160)	67.11	11.5 M	30	18	18
WaveMix-Lite-256/12 (ff 192)	65.46	13.3 M	25	18	17
WaveMix-Lite-256/12 (ff 224)	66.75	15.0 M	30	18	18
WaveMix-Lite-256/12	67.46	16.9 M	25	11	12
WaveMix-Lite-256/12 (ff 1024, mul 3)	62.39	63.2 M	22	14	17
WaveMix-Lite-256/16 (ff 272, mul 3)	67.46	25.5 M	20	17	17
WaveMix-Lite-256/16 (ff 512, mul 3)	**71.75**	44.2 M	18	17	18
WaveMix-Lite-256/16 (ff 512, mul 4)	67.17	47.3 M	18	17	18
WaveMix-Lite-256/16 (ff 1024, mul 3)	69.94	84 M	18	11	17
WaveMix-Lite-256/18 (ff 512, mul 3)	65.16	49.6 M	16	15	17
WaveMix-Lite-256/20 (mul 3)	67.65	30.1 M	16	18	18
WaveMix-Lite-256/20 (ff 512, mul 3)	67.80	55.0 M	16	14	17
WaveMix-Lite-272/16	67.48	25.0 M	22	17	17
WaveMix-Lite-288/16	67.90	28.0 M	20	17	17
WaveMix-Lite-304/16	67.76	31.2 M	18	17	18
WaveMix-Lite-320/16 (ff 512, mul 3)	67.94	56.5 M	15	13	16
WaveMix-Lite-512/12 (ff 1024, mul 3)	65.09	133.2 M	11	6	17
Image resolution 512×1024 on 40 GB A100 GPU					
WaveMix-Lite-128/8	67.55	2.9 M	32	16	16
WaveMix-Lite-256/7	70.43	10.2 M	18	16	16
WaveMix-Lite-256/10	72.54	14.2 M	17	16	16
WaveMix-Lite-256/12	73.01	16.9 M	16	16	16
WaveMix-Lite-256/14	73.86	19.5 M	15	15	16
WaveMix-Lite-256/16	**76.79**	22.2 M	13	14	16
WaveMix-Lite-256/17	74.26	23.5 M	12	14	16
WaveMix-Lite-256/18	74.67	24.9 M	12	14	16
WaveMix-Lite-256/20	74.42	27.5 M	11	12	16
WaveMix-Lite-240/16	74.45	19.5 M	15	14	16
WaveMix-Lite-272/16	73.21	25.1 M	12	12	16
WaveMix-Lite-288/16	73.06	28.1 M	11	12	16

C Semantic Segmentation

C.1 Detailed Results

The original input image resolution of 1024×2048 could not be used in our experiments due to resource constraints. Experiments which used strided convolutions to process the 1024×2048 input performed worse than using downsized 512×1024 images as input. Even with 512×1024 input, two strided convolutional layers will reduce it to 128×256 before it reaches the WaveMix-Lite layers. The 512×1024 input size was only used in 40 GB A100 GPU and 256×512 input size was used for16 GB V100 GPU. Table 9 shows the detailed results of our experiments in Cityscapes dataset. Few examples of qualitative results are provided in Figure 5.

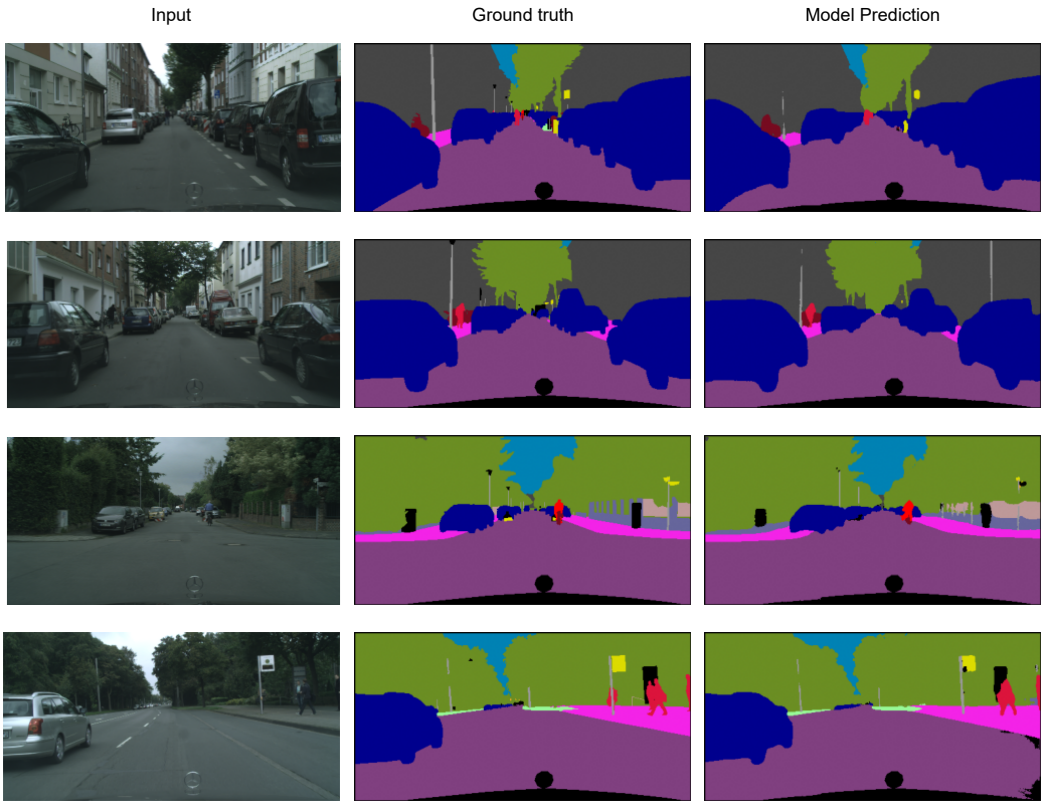

Figure 5: Qualitative results of semantic segmentation on Cityscapes dataset by WaveMix-Lite.

C.1.1 Ablation Studies

Influence of input image size. From the Table 9 we can see that for the same model size, larger input image resolution gave better results. The results for 512×1024 input was 6-8% better than the corresponding results obtained while using input size of 256×512.

Influence of number of layers. The number of layers that could be tested were limited due to the GPU constraints as well as the batch size requirements. We observed an increase in mIoU as the number of layers increases, then it peaks at around 16 layers for both the 512×1024 and 256×512 input and then gradually decreases for each additional layer.

Influence of embedding dimension. We varied the embedding dimension from 128 to 512 in the V100 GPU. The number of layers were adjusted to keep the model fit in the single GPU. For the A100 GPU, it was varied from 128 to 288. Variation of embedding dimension showed a behaviour similar to that shown by increasing the number of layers where mIoU first increases with increase in embedding dimension, then peaks at around 256, and then starts to decrease for both the input image sizes.

Influence of the MLP multiplication factor. The Table 9 shows that increasing the multiplication factor (mul) does not increase the parameter count significantly. It can be used to vary the parameter count slightly for a marginal increase the performance. Increasing the MLP multiplication factor beyond 3 showed slight deterioration in performance with input images of size 256×512.