SpectFormer: Frequency and Attention is what you need in a Vision Transformer-Supplementary

Badri N. Patro
Microsoft
badripatro@microsoft.com

Vinay P. Namboodiri
University of Bath
vpn22@bath.ac.uk

Vijay S. Agneeswaran
Microsoft
vagneeswaran@microsoft.com

1. Introduction

This document contains information about different versions of the vanilla transformer architecture, presented in Table-4. It also includes details on the training configuration for transfer learning, task learning, and fine-tuning tasks. Table-5 displays the dataset information used for transformer learning. In the results section, we present the fine-tuned results for models trained on 224 x 224 and fine-tuned on 384 x 384, as depicted in Table 7. Additionally, we report the object detection performance of the GFL [10] model and Cascade Mask R-CNN [1] on the MS COCO val2017 dataset, which demonstrates an improvement in performance, as shown in Table- 6. We have compared the results of different training processes like Deit and Deit-III as shown in the table- 2. We have included scaling results like a large and huge model as shown in the table- 3. We provide comparison results of ImageNet Top-1 Accuracy (%) vs GFLOPs and Parameters(M) of various models in Vanilla and Hierarchical architecture as shown in figure-2 and figure-3.

Table 1. This table shows the SpectFormer performance based on different model size. The first part shows results on vanilla architecture for Fourier Gating Network based model. The second part shows results for Hierarchical architecture indicated by 'H'. These results are for $\alpha = 4$.

Model	Params (M)	FLOPs (G)	Top-1 (%)	Top-5 (%)
SpectFormer-T	9.15	1.8	76.89	93.38
SpectFormer-XS	20.02	4.0	80.21	94.76
SpectFormer-S	32.56	6.6	81.70	95.64
SpectFormer-B	57.15	11.5	**82.12**	**95.75**
SpectFormer-H-FN-S	21.17	3.9	84.02	96.77
SpectFormer-H-FN-B	31.99	6.3	85.04	97.37
SpectFormer-H-WGN-S	22.59	3.9	83.7	96.56
SpectFormer-H-WGN-B	33.42	6.3	84.57	96.97
SpectFormer-H-S	22.22	3.9	84.25	96.93
SpectFormer-H-B	33.05	6.3	85.05	97.30
SpectFormer-H-L	54.67	12.7	**85.7**	**97.52**

Table 2. This table shows the SpectFormer performance based on the Deit-iii [20] and the Deit [19] training recipe for image size 224^2.

Model	Top-1(%)	Top-5(%)
Deit-s	79.8	95.0
SpectFormer-XS(Deit)	80.21	94.76
SpectFormer-S(Deit)	81.70	95.64
ViT-S (Deit-iii)	80.7	95.12
SpectFormer-XS(Deit-iii)	81.32	95.39
SpectFormer-S(Deit-iii)	82.10	95.96

Table 3. This table shows the SpectFormer performance for Large and Huge Model for image size 224^2.

Model	Params	FLOPs	Top-1(%)	Top-5(%)
SpectFormer-H-L	54.67M	12.7G	**85.7**	**97.52**
SpectFormer-H-H	156.43M	36.57G	**86.16**	**97.92**

2. Experiment

2.1. Dataset and Training setup for Image classification task

We describe the training process of the image recognition task using the ImageNet1K benchmark dataset, which includes 1.28 million training images and 50K validation images belonging to 1,000 categories. The vision backbones are trained from scratch using data augmentation techniques like RandAug, CutOut, and Token Labeling objectives with MixToken. The performance of the trained backbones is evaluated using both top-1 and top-5 accuracies on the validation set. The optimization process involves using the AdamW optimizer with a momentum of 0.9, 10 epochs of linear warm-up, and 310 epochs of cosine decay learning rate scheduler. The batch size is set to 128 and is distributed on 8 A100 GPUs. The learning rate and weight decay are fixed at 0.00001 and 0.05, respectively.

Table 4. In this table, we present detailed configurations of various versions of SpectFormer for the vanilla transformer architecture. The table provides information on the number of heads, embedding dimensions, the number of layers in each variant, and the training resolution. For hierarchical SpectFormer-H models, we provide information in Table-1 of the main paper, which includes details for four stages. The FLOPs (floating-point operations) are calculated for both 224×224 and 384×384 input sizes. For the vanilla SpectFormer architecture, we use four spectral layers with $\alpha = 4$, while the remaining attention layers are equal to $(L - \alpha)$.

Model	#Layers	#heads	# Dim	Params (M)	Resolution	FLOPs (G)
SpectFormer-Ti	12	4	256	9	224	1.8
SpectFormer-XS	12	6	384	20	224	4.0
SpectFormer-S	19	6	384	32	224	6.6
SpectFormer-B	19	8	512	57	224	11.5
SpectFormer-XS	12	6	384	21	384	13.1
SpectFormer-S	19	6	384	33	384	22.0
SpectFormer-B	19	8	512	57	384	37.3

Table 5. This table presents information about datasets used for transfer learning. It includes the size of the training and test sets, as well as the number of categories included in each dataset such as CIFAR-10 [9], CIFAR-100 [9], Flowers-102 [15], Stanford Cars [8].

Dataset	CIFAR-10	CIFAR-100	Flowers-102	Stanford Cars
Train Size	50,000	50,000	8,144	2,040
Test Size	10,000	10,000	8,041	6,149
#Categories	10	100	196	102

Table 6. The performances of various vision backbones on COCO val2017 dataset for the downstream task of object detection. Four kinds of object detectors, *i.e.*, GFL [10], and Cascade Mask R-CNN [1] in mmdetection [2], are adopted for evaluation. We report the bounding box Average Precision (AP^b) in different IoU thresholds.

Backbone	Method	AP^b	AP^b_{50}	AP^b_{75}
ResNet50 [7]	GFL [10]	44.5	63.0	48.3
Swin-T [13]		47.6	66.8	51.7
PVTv2-B2 [21]		50.2	69.4	54.7
SpectFormer-H-S-FN		**50.3**	**70.0**	**55.2**
ResNet50 [7]	Cascade Mask [1] R-CNN	46.3	64.3	50.5
Swin-T [13]		50.5	69.3	54.9
PVTv2-B2 [21]		51.1	69.8	55.3
SpectFormer-H-S-FN		**51.5**	**70.2**	**56.3**

2.2. Training setup for Transfer Learning

To test the effectiveness of our architecture and learned representation, we evaluated vanilla SpectFormer on commonly used transfer learning benchmark datasets, including CIFAR-10 [9], CIFAR100 [9], Oxford-IIIT-Flower [15] and Standford Cars [8]. Our approach followed the methodology of previous studies [4, 16–19], where we initialized the model with ImageNet pre-trained weights and fine-tuned it on the new datasets. In table-7 of the main paper, we have presented a comparison of the transfer learning performance of our basic and best models with state-of-the-art CNNs and vision transformers. The transfer learning setup employs a batch size of 64, a learning rate (lr) of 0.0001, a weight-decay of 1e-4, a clip-grad of 1, and warmup epochs of 5. We have utilized a pre-trained model trained on the Imagenet-1K dataset, which we have fine-tuned on the transfer learning dataset specified in table-5 for 1000 epochs.

2.3. Task Learning: Object Detection

Training setup: In this section, we examine the pre-trained SpectFormer-H-small behavior on COCO dataset for two downstream tasks that localize objects ranging from bounding-box level to pixel level, *i.e.*, object detection and instance segmentation. Two mainstream detectors, *i.e.*, RetinaNet [11] and Mask R-CNN [6] as shown in table-8 of the main paper, and two state-of-the-art detectors *i.e.*, GFL [10], and Cascade Mask R-CNN [1] in mmdetec-

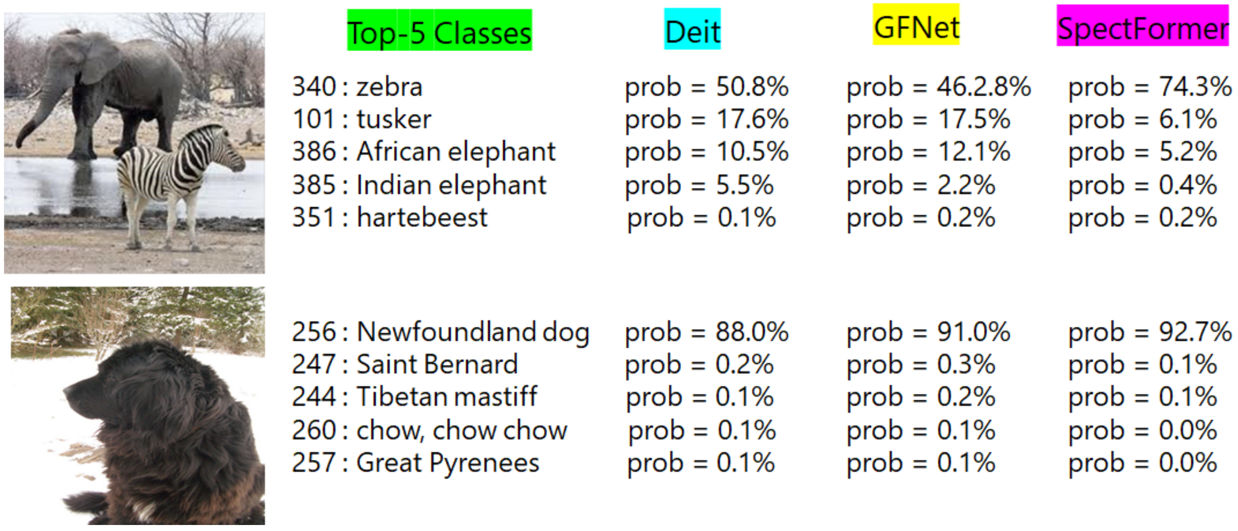
Table 7. We conducted a comparison of various transformer-style architectures for image classification on ImageNet. This includes **vision transformers [19], MLP-like models [12, 18], spectral transformers [16] and our SpectFormer models**, which have similar numbers of parameters and FLOPs. The top-1 accuracy on ImageNet's validation set, as well as the number of parameters and FLOPs, are reported. All models were trained using 224 $\times$ 224 images. We used the notation "↑384" to indicate models fine-tuned on 384 $\times$ 384 images for 30 epochs.

Model	Params (M)	FLOPs (G)	Resolution	Top-1 (%)	Top-5 (%)
gMLP-Ti [12]	6	1.4	224	72.0	-
DeiT-Ti [19]	5	1.2	224	72.2	91.1
GFNet-Ti [16]	7	1.3	224	74.6	92.2
SpectFormer-T	9	1.8	224	76.8	93.3
ResMLP-12 [18]	15	3.0	224	76.6	-
GFNet-XS [16]	16	2.9	224	78.6	94.2
SpectFormer-XS	20	4.0	224	80.2	94.7
DeiT-S [19]	22	4.6	224	79.8	95.0
gMLP-S [12]	20	4.5	224	79.4	-
GFNet-S [16]	25	4.5	224	80.0	94.9
SpectFormer-S	32	6.6	224	81.7	95.6
ResMLP-36 [18]	45	8.9	224	79.7	-
GFNet-B [16]	43	7.9	224	80.7	95.1
gMLP-B [12]	73	15.8	224	81.6	-
DeiT-B [19]	86	17.5	224	81.8	95.6
SpectFormer-B	57	11.5	224	**82.1**	**95.7**
GFNet-XS↑384 [16]	18	8.4	384	80.6	95.4
GFNet-S↑384 [16]	28	13.2	384	81.7	95.8
GFNet-B↑384 [16]	47	23.3	384	82.1	95.8
SpectFormer-XS↑384	21	13.1	384	82.1	95.7
SpectFormer-S↑384	33	22.0	384	83.0	96.3
SpectFormer-B↑384	57	37.3	384	82.9	96.1

Table 8. Detailed architecture specifications for three variants of our SpectFormer with different model sizes, *i.e.*, SpectFormer-S (small size), SpectFormer-B (base size), and SpectFormer-L (large size). E_i, G_i, H_i, and C_i represent the expansion ratio of the feed-forward layer, the spectral gating number, the head number, and the channel dimension in each stage i, respectively.

	Size	SpectFormer-H-S	SpectFormer-H-B
Stage 1	$\frac{H}{4}\times\frac{W}{4}$	$\begin{bmatrix}E_1=8\\G_1=1\\C_1=64\end{bmatrix}\times 2$, $\begin{bmatrix}E_1=8\\H_1=2\\C_1=64\end{bmatrix}\times 1$	$\begin{bmatrix}E_1=8\\G_1=1\\C_1=64\end{bmatrix}\times 2$, $\begin{bmatrix}E_1=8\\H_1=2\\C_1=64\end{bmatrix}\times 1$
Stage 2	$\frac{H}{8}\times\frac{W}{8}$	$\begin{bmatrix}E_2=8\\G_2=1\\C_2=128\end{bmatrix}\times 2$, $\begin{bmatrix}E_2=8\\H_2=4\\C_2=128\end{bmatrix}\times 2$	$\begin{bmatrix}E_2=8\\G_2=1\\C_2=128\end{bmatrix}\times 2$, $\begin{bmatrix}E_2=8\\H_2=4\\C_2=128\end{bmatrix}\times 2$
Stage 3	$\frac{H}{16}\times\frac{W}{16}$	$\begin{bmatrix}E_3=4\\H_3=10\\C_3=320\end{bmatrix}\times 6$	$\begin{bmatrix}E_3=4\\H_3=10\\C_3=320\end{bmatrix}\times 12$
Stage 4	$\frac{H}{32}\times\frac{W}{32}$	$\begin{bmatrix}E_4=4\\H_4=14\\C_4=448\end{bmatrix}\times 3$	$\begin{bmatrix}E_4=4\\H_4=16\\C_4=512\end{bmatrix}\times 3$

tion [2] in this supplementary doc. We are employed for each downstream task, and we replace the CNN backbones in each detector with our SpectFormer-H-small for evaluation. Specifically, each vision backbone is first pre-trained



Top-5 Classes	Deit	GFNet	SpectFormer
340 : zebra	prob = 50.8%	prob = 46.2.8%	prob = 74.3%
101 : tusker	prob = 17.6%	prob = 17.5%	prob = 6.1%
386 : African elephant	prob = 10.5%	prob = 12.1%	prob = 5.2%
385 : Indian elephant	prob = 5.5%	prob = 2.2%	prob = 0.4%
351 : hartebeest	prob = 0.1%	prob = 0.2%	prob = 0.2%
256 : Newfoundland dog	prob = 88.0%	prob = 91.0%	prob = 92.7%
247 : Saint Bernard	prob = 0.2%	prob = 0.3%	prob = 0.1%
244 : Tibetan mastiff	prob = 0.1%	prob = 0.2%	prob = 0.1%
260 : chow, chow chow	prob = 0.1%	prob = 0.1%	prob = 0.0%
257 : Great Pyrenees	prob = 0.1%	prob = 0.1%	prob = 0.0%

Figure 1. The figure shows the top-5 class prediction probability scores for Deit [19], GFNet [16], and our SpectFormer model, indicating that SpectFormer predicts the 'Zebra' class (Top-1 Class) with greater confidence than GFNet and Deit.

over ImageNet1K, and the newly added layers are initialized with Xavier [5]. Next, we follow the standard setups in [13] to train all models on the COCO train2017 ($\sim$118K images). Here the batch size is set as 16, and AdamW [14] is utilized for optimization (weight decay: 0.05, initial learning rate: 0.0001, betas=(0.9, 0.999)). We used learning rate (lr) configuration with step lr policy, linear warmup at every 500 iterations with warmup ration 0.001. All models are finally evaluated on the COCO val2017 (5K images). For state-of-the-art models like GFL [10], and Cascade Mask R-CNN [1], we utilize 3 $\times$ schedule (*i.e.*, 36 epochs) with the multi-scale strategy for training, whereas for RetinaNet [11] and Mask R-CNN [6] we utilize 1 $\times$ schedule (*i.e.*, 12 epochs).

2.4. Training setup for Fine-tuning task

Our main experiments are conducted on ImageNet [3], a popular benchmark for large-scale image classification. To ensure a fair comparison with previous research [16, 18, 19], we adopt the same training details for our SpectFormer models. For the vanilla transformer architecture (SpectFormer), we use the hyper-parameters recommended by the GFNet implementation [16], and for the hierarchical architecture (SpectFormer-H), we use the hyper-parameters recommended by the WaveVit implementation [22]. During fine-tuning at higher resolutions, we use the hyper-parameters recommended by the GFNet implementation [16] and train our models for 30 epochs. All models are trained on a single machine equipped with 8 A100 GPUs. In our experiments, we compared the fine-tuning performance of our models with GFNet [16]. Our observations indicate that our SpectFormer model outperforms GFNet's base spectral network. Specifically, SpectFormer-S(384) achieves a performance of 83.0%, which is 1.2% higher than GFNet-S(384), as shown in Table 7. Similarly, SpectFormer-XS and SpectFormer-B perform better than GFNet-XS and GFNet-B, respectively.

3. Implementation: Spectformer

The objective of the spectral layer is to capture the different frequency components of the image to comprehend localized frequencies. This can be achieved using a spectral gating network, that comprises a Fast Fourier Transform (FFT) layer, followed by a weighted gating, followed by an inverse FFT layer. The spectral layer converts physical space into spectral space using FFT. We use a learnable weight parameter to determine the weight of each frequency component so as to capture the lines and edges of an image appropriately. The learnable weight parameter is specific to each layer of SpectFormer and is learned using back-propagation techniques. The spectral layer uses an inverse Fast Fourier Transform (IFFT) to bring the spectral space back to the physical space. Following the IFFT, the spectral layer has layer normalization and Multi-Layer Perceptron (MLP) block for channel mixing, while token mixing is done using the spectral gating technique.

SpectFormer block has been illustrated in Figure -2 of the main paper, in the staged architecture. We introduce an alpha factor in the SpectFormer block, which controls the number of spectral layers and attention layers. If α=0, SpectFormer comprises all attention layers, similar to DeIT-s, while with an α value of 12, SpectFormer becomes similar to GFNet, with all spectral layers.

The above explanation is mainly for the vanilla Spect-

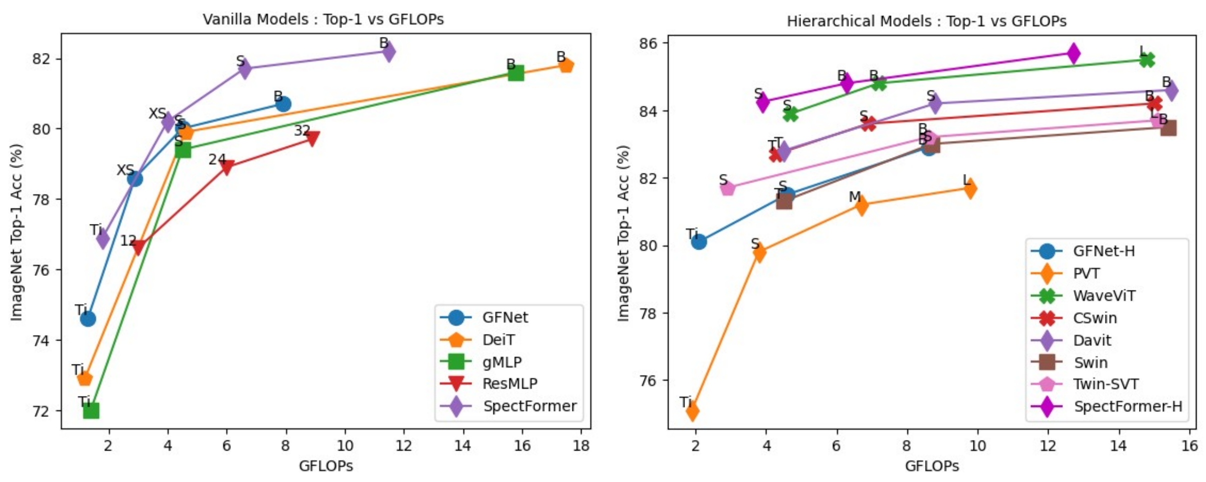


Figure 2. Comparison of ImageNet Top-1 Accuracy (%) vs GFLOPs of various models in Vanilla and Hierarchical architecture.

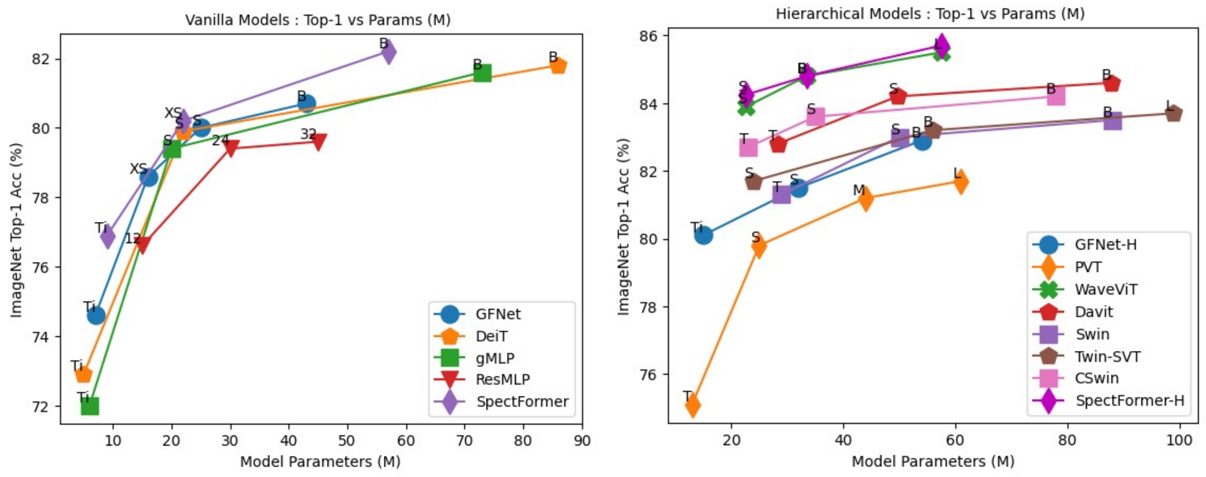


Figure 3. Comparison of ImageNet Top-1 Accuracy (%) vs Parameters (M) of various models in Vanilla and Hierarchical architecture.

Former architecture. We have also come up with a staged architecture, which comprises four stages, with each stage having a varying number of SpecfFormer blocks. Stage 1 has 3 SpectFormer blocks, while Stage 2 has 4, Stage 3 has 6 and Stage 4 has 3 SpectFormer blocks, as shown in table 8. In the stage of SpectFormer-s, there are 2 spectral layers and 1 attention layer, while Stage 2 comprises 2 spectral and 2 attention layers, to capture the local information. Stages 3 and 4 comprise only attention layers, to capture the semantic information. The details of SpectFormer-s architecture were explained above, while SpectFormer-B and SpectFormer-L are depicted in the table. We came up with several variants of the spectral layer including using FNet, FNO, GFNet and AFNO. We also provide a details SpectFormer architecture of the vanilla transformer model, presented in table-4.

4. Results and Analysis

4.1. Comparison with Similar Architectures

We compared the vanilla architecture of SpectFormer to the hierarchical architecture of SpectFormer in two parts of the table-1. In the vanilla architecture, we developed tiny (SpectFormer-T), extra small (SpectFormer-XS), small (SpectFormer-S), and base (SpectFormer-B) models that are similar in layer count and hidden dimensions to GFNet [16], while the attention blocks are similar to Deit [19]. Similarly, in the hierarchical architecture, we developed small (SpectFormer-H-S), base (SpectFormer-H-B), and large (SpectFormer-H-L) models using the Fourier gating network. We also developed small and base models using the Fourier and wavelet gating networks, as shown in table-1. We observed that all the hierarchical models (SpectFormer-H-S, SpectFormer-H-B, and SpectFormer-H-L) performed better than the vanilla architecture and are state-of-the-art, as shown in main paper.

4.2. Model Fine-tuning for High Resolution input

Our main experiments are conducted on ImageNet [3], a popular benchmark for large-scale image classification. To ensure a fair comparison with previous research [16,18,19], we adopt the same training details for our SpectFormer models. For the vanilla transformer architecture (SpectFormer), we use the hyper-parameters recommended by the GFNet implementation [16]. For the hierarchical architecture (SpectFormer-H), we use the hyper-parameters recommended by the WaveVit implementation [22]. We use the hyper-parameters recommended by the GFNet implementation [16] and train our models for 30 epochs during fine-tuning at higher resolutions. All models are trained on a single machine equipped with 8 A100 GPUs. In our experiments, we compared the fine-tuning performance of our models with GFNet [16]. Our observations indicate that our SpectFormer model outperforms GFNet's base spectral network. Specifically, SpectFormer-S(384) achieves a performance of 83.0%, which is 1.2% higher than GFNet-S(384), as shown in Table 7. Similarly, SpectFormer-XS and SpectFormer-B perform better than GFNet-XS and GFNet-B, respectively. In the results section, we present the fine-tuned results for models trained on 224 x 224 and fine-tuned on 384 x 384, as depicted in Table-7.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 1, 2, 4

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 2, 3

[3] 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*, pages 248–255. Ieee, 2009. 4, 6

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 2

[5] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010. 4

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2, 4

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[8] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 2

[9] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 2

[10] Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Advances in Neural Information Processing Systems*, 33:21002–21012, 2020. 1, 2, 4

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 4

[12] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in Neural Information Processing Systems*, 34:9204–9215, 2021. 3

[13] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2, 4

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 4

[15] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008. 2

[16] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. *Advances in Neural Information Processing Systems*, 34:980–993, 2021. 2, 3, 4, 5, 6

[17] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 2

[18] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2, 3, 4, 6

[19] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1, 2, 3, 4, 5, 6

[20] Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. *arXiv preprint arXiv:2204.07118*, 2022. 1

[21] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 2

[22] Ting Yao, Yingwei Pan, Yehao Li, Chong-Wah Ngo, and Tao Mei. Wave-vit: Unifying wavelet and transformers for visual representation learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*, pages 328–345. Springer, 2022. 4, 6