CMT: Convolutional Neural Networks Meet Vision Transformers (Supplementary Material)

Jianyuan Guo[1,2], Kai Han[2], Han Wu[1], Yehui Tang[2], Xinghao Chen[2], Yunhe Wang[2]*, Chang Xu[1]*
[1] School of Computer Science, Faculty of Engineering, University of Sydney. [2] Huawei Noah's Ark Lab.
{jianyuan.guo, kai.han, yunhe.wang}@huawei.com; c.xu@sydney.edu.au

In this supplementary material, we first compare the inference speed of our proposed CMT with other networks. Then we list the training strategy for transfer learning in details. We also show more experiment results under different settings for object detection and instance segmentation on COCO benchmark. Finally we compare CMT with EfficientNetV2 and provide training results of CMT with the same depth/width/resolution as others.

A. Inference Speed

We evaluate the inference speed (throughput, images processed per second) of our proposed CMT-S and CMT-B on ImageNet as shown in Table 2. Noting that while EfficientNet is searched via NAS method, our CMT has stronger potential and better speed-accuracy trade-off. The proposed method also outperforms other transformer-based networks, demonstrating that combining convolution and transformer to capture both local and global information can produce impressive results.

Table 1. **Datasets used for vision tasks.**

Dataset	Train size	Test size	# Classes
ImageNet [4]	1,281,167	50,000	1000
CIFAR10 [11]	50,000	10,000	10
CIFAR100 [11]	50,000	10,000	100
Stanford Cars [10]	8,144	8,041	196
Flowers [13]	2,040	6,149	102
Oxford-IIIT Pets [14]	3,680	3,669	37

B. Transfer Learning

B.1. Image Classification

Datasets. In addition to ImageNet, we also evaluate the proposed CMT on five commonly used transfer learning datasets, including CIFAR10 [11], CIFAR100 [11], Stanford Cars [10], Flowers [13], and Oxford-IIIT Pets [14]. The details of these datasets are listed in Table 1.

Training details. We describe our training strategy on the transfer learning datasets here. We build upon PyTorch [15], and adopt the same data augmentation strategy as that of ImageNet. We change the number of output units in the last classification layer to the number of classes in the target dataset and initialize the new classification layer randomly. The proposed CMT-S are fine-tuned with the image resolution of 224×224 on all datasets. For **CIFAR10** and **CIFAR100**, the model is fine-tuned for 150 epochs with 6e-5 initial learning rate. For **Flowers** and **Pets**, the model is fine-tuned for 300 epochs with 9e-5 and 6e-5 initial learning rate, respectively. For **Cars**, the model is fine-tuned for 500 epochs with 9e-5 initial learning rate. Specifically, our proposed CMT achieves better result with smaller computational cost and less training epochs compared to EfficientNet [18].

Results. In addition to the CMT-S shown in our main paper, we present the transfer learning result of CMT-B in Table 3. We can find that CMT-B outperforms other previous models with less computational cost.

B.2. Object Detection

In addition to the "1x" setting presented in main paper, we also evaluate our CMT-S under the "3×" schedule. We follow the common multi-scale training strategy [6, 21], *i.e.*, randomly resizing the input image so that its shorter side is between 640 and 800. The corresponding results are shown in Table 4.

B.3. Instance Segmentation

Similar to object detection, we show the result of CMT-S based Mask R-CNN under "3×" schedule in Table 5. The proposed CMT architecture can surpass other transformer-based counterparts by a large margin with less FLOPs.

*Corresponding author. Pytorch [15] implementation is available: https://github.com/ggjy/CMT.pytorch

Table 2. **Comparison of the inference speed between different models.** Throughput is measured on a single V100 GPU, following [2,12].

Model	# FLOPs	Throughput (image/s)	ImageNet Top-1 Acc.	Model	# FLOPs	Throughput (image/s)	ImageNet Top-1 Acc.
DeiT-S/16 [20]	4.6B	940.4	79.8%	DeiT-B/16 [20]	17.6B	292.3	81.8%
RegNetY-4G [16]	4.0B	1156.7	80.0%	RegNetY-16G [16]	16.0B	334.7	82.9%
PVT-M [21]	6.7B	528.1	81.2%	PVT-L [21]	9.8B	358.8	81.7%
CPVT-S-GAP [3]	4.6B	942.3	81.5%	CPVT-B [3]	17.6B	285.5	82.3%
Swin-S [12]	8.7B	436.9	83.0%	Swin-B [12]	15.4B	278.1	83.3%
Twins-SVT-B [2]	8.3B	469.0	83.2%	Twins-SVT-L [2]	14.8B	288.0	83.7%
EfficientNet-B4 [18]	4.2B	349.4	82.9%	EfficientNet-B6 [18]	19.0B	96.9	84.0%
CMT-S (ours)	4.0B	562.5	83.5%	CMT-B (ours)	9.3B	285.4	84.5%

Table 3. **More transfer learning results of CMT**. All results are fine-tuned with the ImageNet pretrained checkpoint. † means the transfer results are from [9].

Model	# Params	# FLOPs	CIFAR10	CIFAR100	Cars	Flowers	Pets
ResNet-152† [8]	58.1M	11.3B	97.9%	87.6%	92.0%	97.4%	94.5%
Inception-v4† [17]	41.1M	16.1B	97.9%	87.5%	93.3%	98.5%	93.7%
RegNetY-8GF [16]	39.2M	8.0B	-	-	94.0%	**99.0%**	-
CeiT-S [24]	**24.2M**	4.5B	99.0%	90.8%	93.2%	98.2%	94.6%
EfficientNet-B5$_{\uparrow 456}$ [18]	28.0M	9.9B	98.7%	91.1%	93.9%	98.5%	94.9%
CMT-S (ours)	25.1M	**4.0B**	**99.2%**	**91.7%**	**94.4%**	98.7%	**95.2%**
ViT-B/16$_{\uparrow 384}$ [5]	85.8M	17.6B	98.1%	87.1%	-	89.5%	93.8%
DeiT-B [20]	85.8M	17.6B	99.1%	90.8%	92.1%	98.4%	-
CeiT-S$_{\uparrow 384}$ [24]	24.2M	12.9B	99.1%	90.8%	94.1%	98.6%	94.9%
TNT-S$_{\uparrow 384}$ [7]	**23.8M**	17.3B	98.7%	90.1%	-	98.8%	94.7%
TNT-B$_{\uparrow 384}$ [7]	65.6M	36.6B	99.1%	91.1%	-	**99.0%**	95.0%
EfficientNet-B7$_{\uparrow 600}$ [18]	64.0M	37.2B	98.9%	91.7%	94.7%	98.8%	95.4%
CMT-B (ours)	49.1M	**9.3B**	**99.3%**	**91.9%**	**94.9%**	**99.0%**	**95.5%**

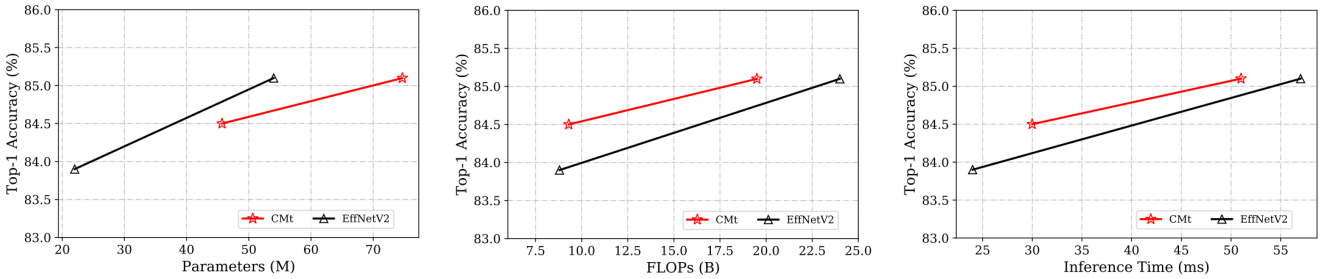

Figure 1. Comparison between CMT and EfficientNet-V2 [19].

C. Compare with EffNetV2.

EffNetV2-S / EffNetV2-M obtains 83.9 (22M,8.8B,24ms) / 85.1 (54M,24B,57ms) on ImageNet-1K. And our CMT-B / CMT-L achieves 84.5[1] (45.7M,9.3B,30ms) / 85.1 (74.7M,19.5B,51ms) on ImageNet-1K, respectively. As shown in Figure 1, CMT achieves better FLOPs/speed and accuracy trade-off than EffNetv2 [19].

D. Same depth/width/resolution as others.

The width/depth/resolution heavily affect the performance of models. To show the effectiveness of our pro-

[1]Inference time is tested on V100 FP16 with batch 16, following EffNetV2. CMT-L is trained with *dp=0.4* and input size of 288^2. CMT-L in main paper is trained with 300 epochs, here we report the newly trained CMT-L with 350 epochs (the same as EffNetV2).

Table 4. **Object detection results on COCO val2017.** All models use RetinaNet as basic framework. "# P" means parameters, "# F" means FLOPs. FLOPs are calculated on 1280×800 input. "1x" indicates 12 epochs, "3x" indicates 36 epochs, and "MS" indicates multi-scale training. † means the results are from [2].

Backbone	# F	# P	RetinaNet 1x						RetinaNet 3x + MS					
			mAP	AP_{50}	AP_{75}	mAP	AP_{50}	AP_{75}						
ConT-M [23]	217B	27.0M	37.9	58.1	40.2	23.0	40.6	50.4	-	-	-	-	-	-
ResNet-101 [8]	315B	56.7M	38.5	57.6	41.0	21.7	42.8	50.4	40.9	60.1	44.0	23.7	45.0	53.8
RelationNet++ [1]	266B	39.0M	39.4	58.2	42.5	-	-	-	-	-	-	-	-	-
ResNeXt-101-32x4d [22]	319B	56.4M	39.9	59.6	42.7	22.3	44.2	52.5	41.4	61.0	44.3	23.9	45.5	53.7
PVT-S [21]	226B	34.2M	40.4	61.3	43.0	25.0	42.9	55.7	42.2	62.7	45.0	26.2	45.2	57.2
Swin-T† [12]	245B	38.5M	41.5	62.1	44.2	25.1	44.9	55.5	43.9	64.8	47.1	28.4	47.2	57.8
Twins-SVT-S [2]	209B	34.3M	42.3	63.4	45.2	26.0	45.5	56.5	45.6	67.1	48.6	29.8	49.3	60.0
Twins-PCPVT-S [2]	226B	34.4M	43.0	64.1	46.0	**27.5**	46.3	57.3	45.2	66.5	48.6	30.0	48.8	58.9
CMT-S (ours)	231B	34.6M	**44.3**	**65.5**	**47.5**	27.1	**48.3**	**59.1**	**46.9**	**67.1**	**50.5**	**30.4**	**49.8**	**61.0**

Table 5. **Instance segmentation results on COCO val2017**. All models use Mask R-CNN as basic framework. "# P" means parameters, "# F" means FLOPs. FLOPs are calculated on 1280×800 input. "1x" indicates 12 epochs, "3x" indicates 36 epochs, and "MS" indicates multi-scale training. † means the results are from [2].

Backbone	# F	# P	Mask R-CNN 1x						Mask R-CNN 3x + MS					
			AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet-101 [8]	336B	63.2M	40.0	60.5	44.0	36.1	57.5	38.6	42.8	63.2	47.1	38.5	60.1	41.3
PVT-S [21]	245B	44.1M	40.4	62.9	43.8	37.8	60.1	40.3	43.0	65.3	46.9	39.9	62.5	42.8
ConT-M [23]	237B	34.2M	40.5	-	-	38.1	-	-	-	-	-	-	-	-
ResNeXt-101-32x4d [22]	340B	62.8M	41.9	62.5	45.9	37.5	59.4	40.2	44.0	64.4	48.0	39.2	61.4	41.9
Swin-T† [12]	264B	47.8M	42.2	64.6	46.2	39.1	61.6	42.0	46.0	68.2	50.2	41.6	65.1	44.8
Twins-SVT-S [2]	228B	44.0M	42.7	65.6	46.7	39.6	62.5	42.6	46.8	69.2	51.2	42.6	66.3	45.8
Twins-PCPVT-S [2]	245B	44.3M	42.9	65.8	47.1	40.0	62.7	42.9	46.8	69.3	51.8	42.6	66.3	46.0
CMT-S (ours)	249B	44.5M	**44.6**	**66.8**	**48.9**	**40.7**	**63.9**	**43.4**	**48.3**	**70.4**	**52.3**	**43.7**	**67.7**	**47.1**

posed modules, we also provide training results of CMT with the same depth/width/resolution as others. We construct two models, namely CMT-PVT-S and CMT-Swin-T, with the same depth/width/resolution as PVT and Swin for a fairer comparison. As shown in Table 6, CMT based models surpass others by a large margin.

References

[1] Cheng Chi, Fangyun Wei, and Han Hu. Relationnet++: Bridging visual representations for object detection via transformer decoder. *arXiv preprint arXiv:2010.15831*, 2020. 3

[2] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting spatial attention design in vision transformers. *arXiv preprint arXiv:2104.13840*, 2021. 2, 3

[3] Xiangxiang Chu, Bo Zhang, Zhi Tian, Xiaolin Wei, and Huaxia Xia. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 2

[4] 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*, 2009. 1

[5] 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. *arXiv preprint arXiv:2010.11929*, 2020. 2

[6] Jianyuan Guo, Kai Han, Han Wu, Chao Zhang, Xinghao Chen, Chunjing Xu, Chang Xu, and Yunhe Wang. Positive-unlabeled data purification in the wild for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1

[7] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021. 2

[8] 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*, 2016. 2, 3

[9] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of*

Model	Resolution	# Params	# FLOPs	Top-1	Top-5
PVT-Small [21]	224^2	25M	3.8B	79.8	-
PVT-V2-B2 [21]	224^2	25M	4.0B	82.0	-
CMT-PVT-S	224^2	22M	4.0B	82.9	96.2
Swin-T [12]	224^2	28M	4.5B	81.3	95.5
CMT-Swin-T	224^2	31M	4.4B	83.1	96.3

Table 6. CMT-PVT-S follows the setting of PVT-Small and PVT-V2-B2: depth=[3,4,6,3], dim=[64,128,320,512], num_heads=[1,2,5,8], mlp_ratios=[8,8,4,4]. CMT-Swin-T follows the setting of Swin-T: depth=[2,2,6,2], dim=[96,192,384,768], num_heads=[3,6,12,24].

the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 2

[10] 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*, 2013. 1

[11] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[12] 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. *arXiv preprint arXiv:2103.14030*, 2021. 2, 3, 4

[13] 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*, 2008. 1

[14] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, 2012. 1

[15] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019. 1

[16] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 2

[17] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. 2

[18] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019. 1, 2

[19] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, 2021. 2

[20] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 2

[21] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 1, 2, 3, 4

[22] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017. 3

[23] Haotian Yan, Zhe Li, Weijian Li, Changhu Wang, Ming Wu, and Chuang Zhang. Contnet: Why not use convolution and transformer at the same time? *arXiv preprint arXiv:2104.13497*, 2021. 3

[24] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. *arXiv preprint arXiv:2103.11816*, 2021. 2