A. Appendix

In this appendix, Section A.1 first describe the training details of our experiments for ImageNet classification, COCO detection/instance segmentation, and ADE20K semantic segmentation. Second, in Section A.2, we show further experimental analyses for ImageNet classification and COCO object detection. Finally, in Section A.3, we provide more qualitative analysis on the learned attention maps and failure cases.

A.1. Detailed Experimental Settings

ImageNet classification. Following the training recipe as in CoaT [65] and DeiT [14], we perform the same data augmentations such as MixUP [28], CutMix [70], random erasing [72], repeated augmentation [24], and label smoothing [48]. We train MPViTs for 300 epochs with the AdamW [38] optimizer, a batch size of 1024, weight decay of 0.05, five warm-up epochs, and an initial learning rate of 0.001, which is scaled by a cosine decay learning rate scheduler. We implement MPViTs based on CoaT official code [1] and `timm` library [59].

Object detection and Instance segmentation. For fair comparison, we follow the training recipe as in CoaT [65] and Swin Transformer [37] for RetinaNet [35] and Mask R-CNN [22]. Specifically, we train all models for 3$\times$ schedule (36 epochs) [22, 61] with multi-scale inputs (MS) [5, 45] which resizes the input such that the shorter side is between 480 and 800 while the longer side is at most 1333). We use the AdamW [38] optimizer, a weight decay of 0.05, a batch size of 16, and an initial learning rate of 0.0001 which is decayed by 10$\times$ at epochs 27 and 33. We set stochastic depth drop rates [27] to 0.1, 0.1, 0.2, and 0.4 for Tiny, XSmall, Small, and Base, respectively. We implement all models based on the `detectron2` library [61].

Semantic segmentation. Following the same training recipe as in Swin Transformer [37] and XCiT [17], we deploy UperNet [62] with the AdamW [38] optimizer, a weight decay of 0.01, an initial learning rate of 6×10^{-5} which is scaled using a linear learning rate decay, and linear warmup of 1,500 iterations. We train models for 160K iterations with a batch size of 16 and an input size of 512×512. We use the same data augmentations as [11, 37], utilizing random horizontal flipping, a random re-scaling ratio in the range [0,5, 2.0] and random photometric distortions. We set stochastic depth drop rates [27] to 0.2 and 0.4 for Small and Base, respectively. We implement all models based on the `mmseg` library [11].

[1] https://github.com/mlpc-ucsd/CoaT

Model	Param.(M)	GFLOPs	Top-1	Reference
DeiT-T [50]	5.7	1.3	72.2	ICML21
TnT-Ti [21]	6.1	1.4	73.9	NeurIPS21
ViL-Ti-RPB [71]	6.7	1.3	76.7	ICCV21
XCiT-T12/16 [17]	7.0	1.2	77.1	NeurIPS21
ViTAE-6M [66]	6.5	2.0	77.9	NeurIPS21
CoaT-Lite T [65]	5.7	1.6	76.6	ICCV21
MPViT-T	5.8	1.6	**78.2 (+1.6)**	
ResNet-18 [23]	11.7	1.8	69.8	CVPR16
PVT-T [58]	13.2	1.9	75.1	ICCV21
XCiT-T24/16 [17]	12.0	2.3	79.4	NeurIPS21
CoaT Mi [65]	10.0	6.8	80.8	ICCV21
CoaT-Lite Mi [65]	11.0	2.0	78.9	ICCV21
MPViT-XS	10.5	2.9	**80.9 (+2.0)**	
ResNet-50 [23]	25.6	4.1	76.1	CVPR16
PVT-S [58]	24.5	3.8	79.8	ICCV21
DeiT-S/16 [50]	22.1	4.6	79.9	ICML21
Swin-T [37]	29.0	4.5	81.3	ICCV21
Twins-SVT-S [10]	24.0	2.8	81.3	NeurIPS21
TnT-S [21]	23.8	5.2	81.5	NeurIPS21
CvT-13 [60]	20.0	4.5	81.6	ICCV21
XCiT-S12/16 [17]	26.0	4.8	82.0	NeurIPS21
ViTAE-S [66]	23.6	5.6	82.0	NeurIPS21
GG-T [68]	28.0	4.5	82.0	NeurIPS21
CoaT S [65]	22.0	12.6	82.1	ICCV21
Focal-T [67]	29.1	4.9	82.2	NeurIPS21
CrossViT-15 [6]	28.2	6.1	82.3	ICCV21
ViL-S-RPB [71]	24.6	4.9	82.4	ICCV21
CvT-21 [60]	32.0	7.1	82.5	ICCV21
CrossViT-18 [6]	43.3	9.5	82.8	ICCV21
HRFormer-B [69]	50.3	13.7	82.8	NeurIPS21
CoaT-Lite S [65]	20.0	4.0	81.9	ICCV21
MPViT-S	22.8	4.7	**83.0 (+1.1)**	
ResNeXt-101 [64]	83.5	15.6	79.6	CVPR17
PVT-L [58]	61.4	9.8	81.7	ICCV21
DeiT-B/16 [50]	86.6	17.6	81.8	ICML21
XCiT-M24/16 [17]	84.0	16.2	82.7	NeurIPS21
Twins-SVT-B [10]	56.0	8.3	83.1	NeurIPS21
Swin-S [37]	49.6	8.7	83.1	ICCV21
Twins-SVT-L [10]	99.2	14.8	83.3	NeurIPS21
Swin-B [37]	88.0	15.4	83.3	ICCV21
XCiT-S12/8 [17]	26.0	18.9	83.4	NeurIPS21
Focal-S [67]	51.1	9.1	83.5	NeurIPS21
XCiT-M24/8 [17]	84.0	63.9	83.7	NeurIPS21
Focal-B [67]	89.8	16.0	83.8	NeurIPS21
XCiT-S24/8 [17]	48.0	36.0	83.9	NeurIPS21
MPViT-B	74.8	16.4	**84.3**	

Table 8. **Full comparison on ImageNet-1K classification.** These models are trained with 224×224 resolution. For fair comparison, we do not include models that are distilled [50] or use 384×384 resolution. Note that CoaT-Lite [65] models are our single-path baselines.

A.2. More Experimental Analysis

ImageNet classification. We provide a full summary of comparisons on ImageNet-1K classification in Table 8 by adding more recent Vision Transformers including ViL [71], TnT [21], ViTAE [66], HRFormer [69], and Twins [10]. We can observe that MPViTs consistently achieve state-the-art performance compared to SOTA models with similar model capacity. Notably, the smaller MPViT variants often outperform their larger baseline counterparts even when the baselines use significantly more

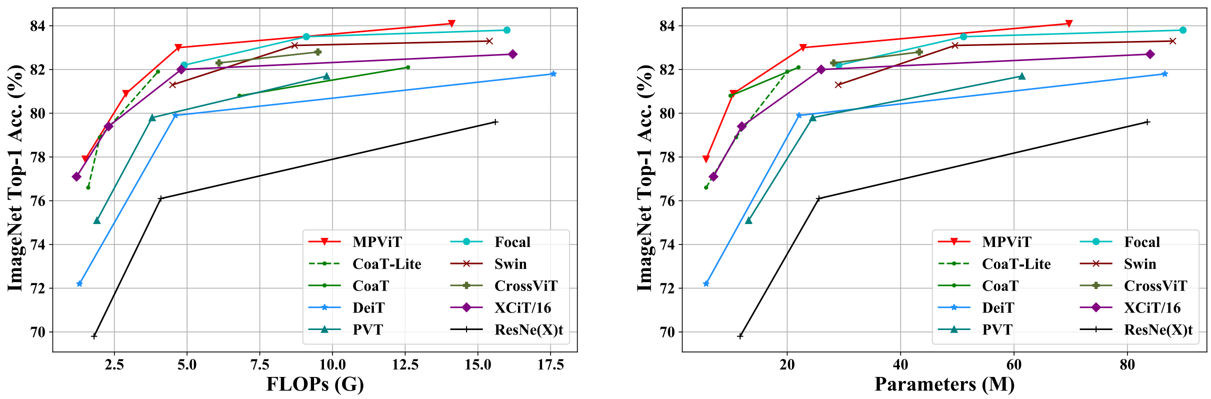

Figure 5. **Performance comparisons with respect to FLOPs and model parameters on ImageNet-1K classification.** These models are trained with 224×224 single-crop. For fair comparison, we do not include models that are distilled [50] or use 384×384 resolution.

parameters, as shown in Table 8 and Fig. 5 (right). Furthermore, Fig. 5 demonstrates that MPViT is a more *efficient* and *effective* Vision Transformer architecture in terms of computation and model parameters.

Deformable-DETR. Additionally, we compare our MPViT-Small with baselines, CoaT-Lite Small [65] and CoaT Small [65], on the Deformable DETR (DD) [74]. For fair comparison, we train MPViT for 50 epochs with the same training recipe[2] as in CoaT [65]. We use the AdamW [38] optimizer with a batch size of 16, a weight decay of 10^{-4}, and an initial learning rate of 2×10^{-4}, which is decayed by a factor of 10 at 40 epoch. Tab. 9 shows results comparing with CoaT-Lite Small and CoaT Small. MPViT-Small improves over both CoaT-Lite Small and CoaT Small. Notably, MPViT achieves a larger gain in small object AP (1.5% AP_S) as compared to others (*i.e.*, AP_M or AP_L).

COCO with $1\times$ **schedule.**. In addition to the $3\times$ schedule + multi-scale (MS) setting, we also evaluate MPViT on RetinaNet [35] and Mask R-CNN [22] with $1\times$ schedule (12 epochs) [61] using single-scale inputs. Tab. 10 shows result comparisons with state-of-the-art methods. In the results of $3\times$ schedule + multi-scale (MS), we can also observe that MPViTs consistently outperform on both RetinaNet and Mask R-CNN. We note that MPViTs surpass the most recent improved PVTv2 [57] models.

A.3. More Qualitative Results

Visualization of Attention Maps. As shown in Eq.(4), the factorized self-attention in [65] first extracts channel-wise attention softmax(K) by applying a softmax over spatial

[2] https://github.com/mlpc-ucsd/CoaT/tree/main/tasks/Deformable-DETR

Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-50 [23]	44.5	63.7	48.7	26.8	47.6	59.6
CoaT-Lite small [65]	47.0	66.5	51.2	28.8	50.3	63.3
CoaT Small [65]	48.4	68.5	52.4	30.2	51.8	63.8
MPViT-Small	**49.0**	**68.7**	**53.7**	**31.7**	**52.4**	**64.5**

Table 9. **COCO Object Detection results on Deformable DETR** [74]. These all models are trained using the same codebase.

dimensions (x, y). Then, softmax$(K)^T V$ is computed as below:

$$(\text{softmax}(K)^T V)(c_i, c_j) = \sum_{(x,y)} \text{softmax}(K)(x, y, c_i) V(x, y, c_j), \quad (5)$$

where x and y are position of tokens. c_i and c_j indicate channel indices of K and V, respectively. It can be interpreted as multiplying V by the channel-wise spatial attention in a pixel-wise manner followed by the sum over spatial dimension. In other words, softmax$(K)^T V$ represents the weighted sum of V where the weight of each position (x, y) is the channel-wise spatial attention. Therefore, to obtain the importance of each position, we employ the mean of softmax(K) over the channel dimension, resulting in spatial attention. Then, the spatial attention is overlaid to the original input image for better visualization, as shown in Fig. 6. In detail, we resize the spatial attention to the size of the original image, normalize the value to [0,1], and then multiply the attention map by the image.

To validate the effectiveness of our attention map qualitatively, we compare attention maps of MPViT and CoaT-Lite [65] in Fig. 6. We compare the attention maps of each method generated from the 4th stage in the same way. For a fair comparison, we pick the best qualitative attention map of each method since both CoaT-Lite and MPViT have eight heads for each layer. Furthermore, we visualize attention

Backbone	Params. (M)	GFLOPs	Mask R-CNN 1×						RetinaNet 1×					
			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^b_S	AP^b_M	AP^b_L
PVTv2-B0 [57]	23 (13)	195 (177)	38.2	60.5	40.7	36.2	57.8	38.6	37.2	57.2	39.5	23.1	40.4	49.7
MPViT-T	28 (17)	216 (196)	**42.2**	64.2	45.8	**39.0**	61.4	41.8	**41.8**	62.7	44.6	27.2	45.1	54.2
PVT-T [58]	33 (23)	240 (221)	39.8	62.2	43.0	37.4	59.3	39.9	39.4	59.8	42.0	25.5	42.0	52.1
PVTv2-B1 [57]	33 (23)	243 (225)	41.8	54.3	45.9	38.8	61.2	41.6	41.2	61.9	43.9	25.4	44.5	54.3
MPViT-XS	30 (20)	231 (211)	**44.2**	66.7	48.4	**40.4**	63.4	43.4	**43.8**	65.0	47.1	28.1	47.6	56.5
ResNet-50 [23]	44 (38)	260 (239)	38.0	58.6	41.4	34.4	55.1	36.7	36.3	55.3	38.6	19.3	40.4	48.8
PVT-S [58]	44 (34)	305 (226)	43.0	65.3	46.9	39.9	62.5	42.8	42.2	62.7	45.0	26.2	45.2	57.2
PVTv2-B2 [57]	45 (35)	309 (290)	45.3	67.1	49.6	41.2	64.2	44.4	44.6	65.6	47.6	27.4	48.8	58.6
Swin-T [37]	48 (39)	267 (245)	43.7	66.6	47.7	39.8	63.3	42.7	42.0	63.0	44.7	26.6	45.8	55.7
Focal-T [67]	49 (39)	291 (265)	44.8	67.7	49.2	41.0	64.7	44.2	43.7	65.2	46.7	28.6	47.4	56.9
MPViT-S	43 (32)	268 (248)	**46.4**	68.6	51.2	**42.4**	65.6	45.7	**45.7**	57.3	48.8	28.7	49.7	59.2
ResNeXt101-64x4d [64]	102 (96)	493 (473)	42.8	63.8	47.3	38.4	60.6	41.3	41.0	60.9	44.0	23.9	45.2	54.0
PVT-M [58]	64 (54)	392 (283)	42.0	64.4	45.6	39.0	61.6	42.1	41.9	63.1	44.3	25.0	44.9	57.6
PVT-L [58]	81 (71)	494 (345)	42.9	65.0	46.6	39.5	61.9	42.5	42.6	63.7	45.4	25.8	46.0	58.4
PVTv2-B5 [57]	101 (91)	557 (538)	47.4	68.6	51.9	42.5	65.7	46.0	46.2	67.1	49.5	28.5	50.0	62.5
Swin-S [37]	69 (60)	359 (335)	46.5	68.7	51.3	42.1	65.8	45.2	45.0	66.2	48.3	27.9	48.8	59.5
Swin-B [37]	107 (98)	496 (477)	46.9	69.2	51.6	42.3	66.0	45.5	45.0	66.4	48.3	28.4	49.1	60.6
Focal-S [67]	71 (62)	401 (367)	47.4	69.8	51.9	42.8	66.6	46.1	45.6	67.0	48.7	29.5	49.5	60.3
Focal-B [67]	110 (101)	533 (514)	47.8	70.2	52.5	43.2	67.3	46.5	46.3	68.0	49.8	31.7	50.4	60.8
MPViT-B	95 (85)	503 (482)	**48.2**	70.0	52.9	**43.5**	67.1	46.8	**47.0**	68.4	50.8	29.4	51.3	61.5

Table 10. **COCO detection and instance segmentation** with RetinaNet [35] and Mask R-CNN [22]. Models are trained for 1× schedule [61] with single-scale training inputs. All backbones are pretrained on ImageNet-1K. We omit models pretrained on larger-datasets (*e.g.*, ImageNet-21K). The GFLOPs are measured at resolution 800 × 1280. Mask R-CNN's parameters/FLOPs are followed by RetinaNet in parentheses.

maps extracted from all three paths of MPViT to observe the individual effects of each path.

As mentioned in Section 5, the three paths of MPViT can capture objects of varying sizes due to the multi-scale embedding of MPViT as the similar effect of multiple receptive fields. In other words, path-1 concentrates on small objects or textures while path-3 focuses on large objects or high-level semantic concepts. We support this intuition by observing more examples shown in Fig. 6. Attention maps of path-1 (3rd column) capture small objects such as small ducks (4th row), an orange (5th row), a small ball (6th row), and an antelope (8th row). In addition, since path-1 also captures textures due to a smaller receptive field, a relatively low level of attention is present in the background. In contrast, we can observe different behavior for path-3, which can be seen in the rightmost column. Path-3 accentuates large objects while suppressing the background and smaller objects. For example, the ducks (4th row), orange (5th row), and ball (8th row) are masked out in the rightmost column since path-3 concentrates on larger objects. The attention maps of path-2 (4th column) showcase the changing behavior between paths-1 and 3 since the scale of path-2 is in-between the scales of paths-1 and 3, and accordingly, the attention maps also begin to transition from smaller to larger objects. In other words, although the attention map of path-2 attends similar regions as path-1, it is also more likely to emphasize larger objects, as path-3 does. For example, in the last row, path-2 attends to similar regions as path-1 while emphasizing the large giraffes more than path-1. Therefore, although the three paths independently deal with different scales, they act in a complementary manner, which is beneficial for dense prediction tasks.

Since Coat-Lite has a single-path architecture, the singular path needs to deal with objects of varying sizes. Therefore, attention maps from CoaT-Lite (2nd column) simultaneously attend to large and small objects, as shown in the 4th row. However, it is difficult to capture all objects with a single path, as CoaT-Lite misses the orange (5th row) and ball (7th row). In addition, Coat-Lite cannot capture object boundaries as precisely as path-3 of MPViT since path-3 need not attend to small objects or textures. As a result, MPViT shows superior results compared to Coat-Lite on classification, detection, and segmentation tasks.

Failure case. In order to verify the effects of attention from a different perspective, we further analyze failure cases on the ImageNet *validation* images. We show attention maps of each path corresponding to the input image along with the ground truth and the predicted labels of MPViT in Fig. 7. For example, in the first row, the ground truth of the input image is a forklift, while the predicted label is a trailer truck. Although the attention map from path-1 places light emphasis on the forklift, the attention maps from all paths commonly accentuate the trailer truck rather than the forklift, which leads to classifying the image as a trailer truck and not a forklift. Other classification results in Fig. 7 fail in similar circumstances, except for the last row. In the last row, MPViTs attention maps correctly capture the beer bottle. However, the attention maps also attend to the face near the bottle. Therefore, the bottle is misunderstood as a microphone since the image of "drinking a bottle of beer" and

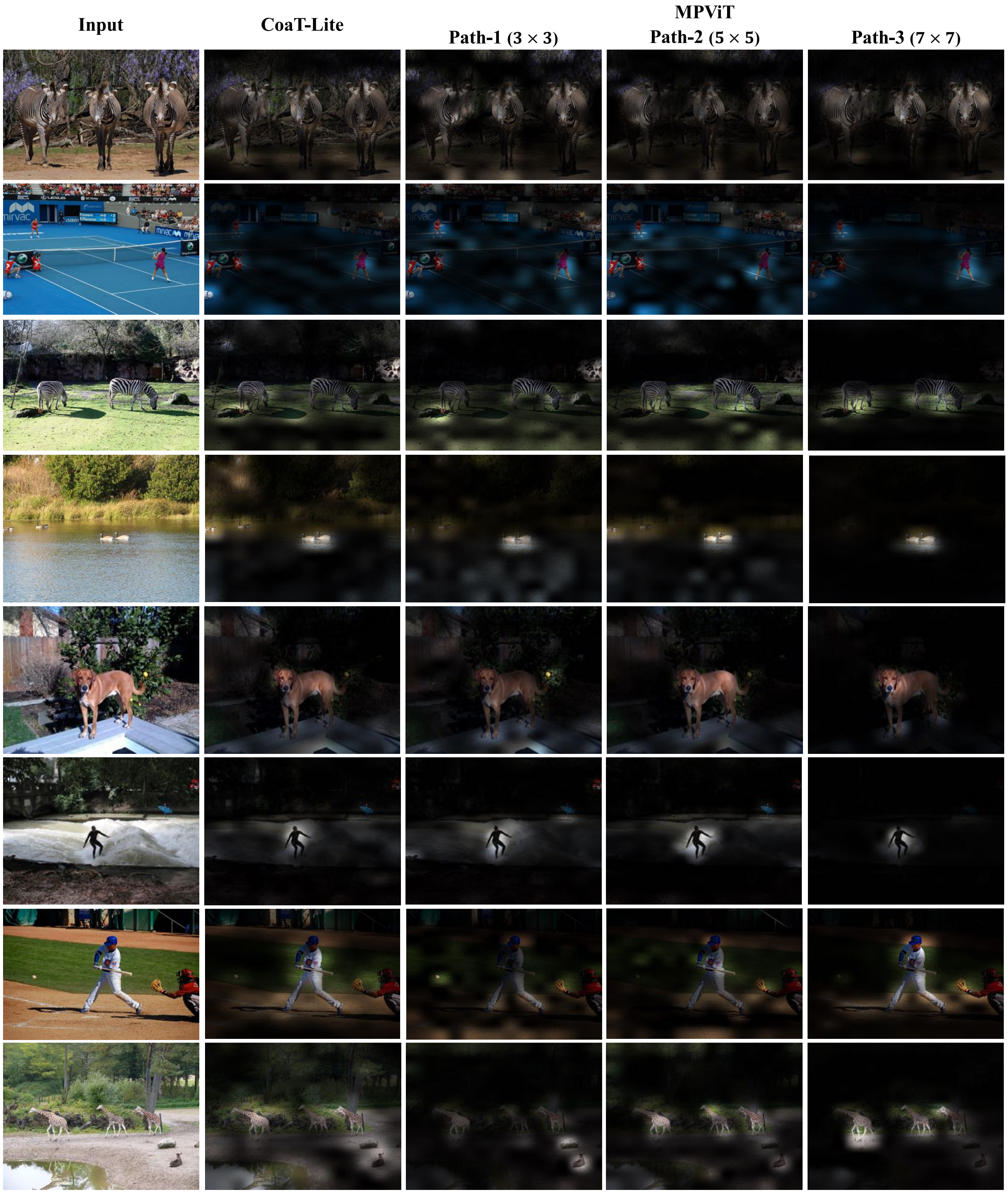

Figure 6. **Additional Attention Maps** generated by CoaT-Lite [65] and our MPViT. MPViT has a triple-path structure with patches of various sizes (*e.g.*, $3 \times 3, 5 \times 5, 7 \times 7$), leading to fine and coarse features.

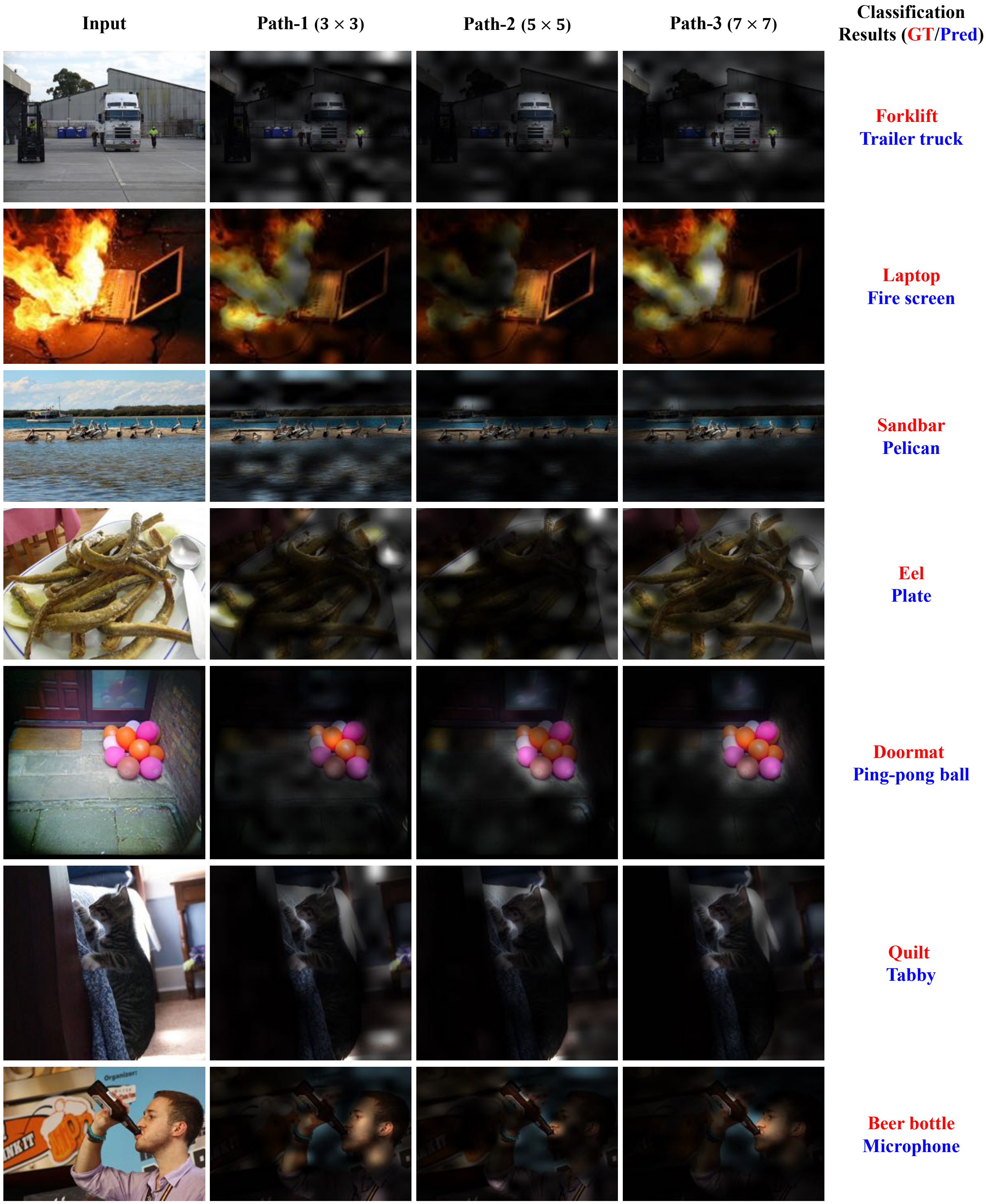

Figure 7. **Attention Maps of failure cases on ImageNet *validation* images.** The input image and corresponding attention maps from each path are illustrated. In the rightmost column, we show the ground truth labels and predicted labels colored with red and blue, respectively.

"using a microphone" are semantically similar. From the above, we can observe that the attention maps and the predicted results are highly correlated.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021. 1

[2] Nicholas Baker, Hongjing Lu, Gennady Erlikhman, and Philip J Kellman. Deep convolutional networks do not classify based on global object shape. *PLoS computational biology*, 14(12):e1006613, 2018. 4

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 1

[4] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 5, 9

[6] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *ICCV*, 2021. 3, 5, 9

[7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. In *CVPR*, 2021. 1

[8] Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. In *ICCV*, 2021. 3

[9] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 4

[10] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *NeurIPS*, 2021. 9

[11] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 6, 9

[12] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, 2021. 1

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 5

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, 2019. 1, 9

[15] Piotr Dollár, Mannat Singh, and Ross Girshick. Fast and accurate model scaling. In *CVPR*, 2021. 8

[16] 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 *ICLR*, 2021. 1, 2, 3

[17] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. In *NeurIPS*, 2021. 1, 2, 3, 5, 6, 8, 9

[18] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, 2021. 1

[19] Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. Res2net: A new multi-scale backbone architecture. *TPAMI*, 2019. 1, 2

[20] 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. In *ICCV*, 2021. 3

[21] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. In *NeurIPS*, 2021. 9

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *CVPR*, 2017. 5, 6, 7, 9, 10, 11

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 3, 4, 5, 9, 10, 11

[24] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *CVPR*, 2020. 9

[25] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *CVPR*, 2019. 3, 4

[26] 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. *arXiv preprint arXiv:1704.04861*, 2017. 4

[27] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 5, 9

[28] Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018. 9

[29] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 3, 4

[30] Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? *arXiv preprint arXiv:2001.08248*, 2020. 4

[31] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *CVPR*, 2020. 4

[32] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation

efficient backbone network for real-time object detection. In *CVPRW*, 2019. 1, 2

[33] Youngwan Lee, Hieun Kim, Eunsoo Park, Xuenan Cui, and Hakil Kim. Wide-residual-inception networks for real-time object detection. In *IEEE Intelligent Vehicles Symposium (IV)*, 2017. 1

[34] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 1

[35] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 5, 6, 9, 10, 11

[36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5

[37] 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 *ICCV*, 2021. 1, 2, 3, 5, 6, 8, 9, 11

[38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5, 6, 9, 10

[39] David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 4

[40] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018. 8

[41] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021. 1

[42] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016. 1

[43] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. *Technical report, OpenAI*, 2018. 1

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2014. 2, 3

[45] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *CVPR*, 2021. 5, 6, 9

[46] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, 2017. 1, 2

[47] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 1

[48] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 9

[49] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 8

[50] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 1, 2, 4, 5, 9, 10

[51] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, 2021. 1, 5

[52] Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L Griffiths. Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*, 2021. 4

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[54] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021. 1

[55] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2020. 1

[56] Ning Wang, Wengang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *ICCV*, 2021. 1

[57] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv preprint arXiv:2106.13797*, 2021. 1, 2, 5, 10, 11

[58] 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. In *ICCV*, 2021. 1, 2, 3, 5, 6, 9, 11

[59] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 9

[60] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *ICCV*, 2021. 1, 2, 3, 5, 9

[61] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 5, 6, 9, 10, 11

[62] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 6, 9

[63] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 1

[64] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017. 5, 8, 9, 11

[65] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *ICCV*, 2021. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12

[66] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. In *NeurIPS*, 2021. 9

[67] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. In *NeurIPS*, 2021. 1, 2, 3, 5, 6, 8, 9, 11

[68] Qihang Yu, Yingda Xia, Yutong Bai, Yongyi Lu, Alan Yuille, and Wei Shen. Glance-and-gaze vision transformer. In *NeurIPS*, 2021. 9

[69] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. Hrformer: High-resolution transformer for dense prediction. In *NeurIPS*, 2021. 9

[70] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 9

[71] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *ICCV*, 2021. 2, 9

[72] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020. 9

[73] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 5, 6

[74] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 1, 10