RepViT: Revisiting Mobile CNN From ViT Perspective

Ao Wang[1,2] Hui Chen[1,2*] Zijia Lin[1] Jungong Han[3] Guiguang Ding[1,2*]
[1]Tsinghua University [2]BNRist [3]The University of Sheffield
wa22@mails.tsinghua.edu.cn huichen@mail.tsinghua.edu.cn linzijia07@tsinghua.org.cn
jungonghan77@gmail.com dinggg@tsinghua.edu.cn

Appendix

Table of contents:

A. Architecture and Implementation Details

A.1. Architecture Details

Table 1 provides the architecture details of different RepViT variants.

Table 1. Architecture details of RepViT variants.

Stage	Resolution	Config	RepViT				
			M0.9	M1.0	M1.1	M1.5	M2.3
stem	$\frac{H}{2} \times \frac{W}{2}$	channels	24	28	32	32	40
	$\frac{H}{4} \times \frac{W}{4}$	channels	48	56	64	64	80
1	$\frac{H}{4} \times \frac{W}{4}$	channels	48	56	64	64	80
		blocks	2	2	2	4	6
2	$\frac{H}{8} \times \frac{W}{8}$	channels	96	112	128	128	160
		blocks	2	2	2	4	6
3	$\frac{H}{16} \times \frac{W}{16}$	channels	192	224	256	256	320
		blocks	14	14	12	24	34
4	$\frac{H}{32} \times \frac{W}{32}$	channels	384	448	512	512	640
		blocks	2	2	2	4	2

A.2. Implementation Details

We follow the same training recipe as [13, 14, 25] on ImageNet-1K [3]. We adopt the AdamW optimizer [20] and the cosine learning rate scheduler. The initial learning rate is set to 4×10^{-3}, and the minimum learning rate is set to 1×10^{-5}. The total batch size is set to 2048, and the weight decay is set to 2.5×10^{-2}. For data augmentation, we utilize Mixup [31], auto-augmentation [2], and random erasing [32]. Table 2 provides the detailed training hyper-parameters.

Table 2. Training hyper-parameters for ImageNet-1K.

Hyperparameters	Config
optimizer	AdamW
learning rate	0.004
batch size	2048
LR schedule	cosine
warmup epochs	5
training epochs	300 / 450
weight decay	0.025
augmentation	RandAug(9, 0.5)
color jitter	0.4
gradient clip	0.02
random erase	0.25
label smooth	0.1
mixup	0.8
cutmix	1.0

For object detection and instance segmentation tasks, following [13, 14, 23, 24], we adopt AdamW optimizer with an initial learning rate of 2×10^{-4} and train the model for 12 epochs with a standard resolution (1333×800). The backbones are initialized with pretrained ImageNet-1k weights.

For semantic segmentation, following [13, 14, 24], we train the models on ADE20K [33] for 40K iterations with a batch size of 32. We adopt AdamW optimizer, and employ a poly learning rate schedule with the power of 0.9. The initial learning rate is set to 2×10^{-4}. We employ the standard resolution (512×512) for training and report the single scale testing results on the validation set. The backbones are initialized with pretrained weights on ImageNet-1K.

B. Robustness Evaluation

We present robustness evaluation results for RepViT models in Table 3. Following [21, 25], we directly test ImageNet-1K trained models on a series of robustness benchmark

*Corresponding author.

Table 3. Results on robustness benchmark datasets.

Model	Latency ↓ (ms)	Clean	IN-C (↓)	IN-A	IN-R	IN-SK
MobileOne-S1	0.9	75.9	80.4	2.7	36.7	22.6
FastViT-T8	0.9	76.7	68.7	7.9	37.2	25.8
MobileViG-Ti	1.0	75.7	70.9	5.8	38.9	26.1
SwiftFormer-XS	1.0	75.7	77.6	5.0	34.0	22.4
RepViT-M0.9	**0.9**	**78.7**	**64.8**	**10.4**	**42.7**	**30.4**
RepViT-M1.0	**1.0**	**80.0**	**61.6**	**13.7**	**43.9**	**30.7**
MobileOne-S2	1.1	77.4	73.6	4.8	40.0	26.4
MobileOne-S3	1.3	78.1	71.6	7.1	42.1	28.5
FastViT-T12	1.4	80.3	60.6	**15.8**	41.8	29.5
MobileViG-S	1.2	78.2	64.7	10.9	41.9	28.9
EfficientFormer-L1	1.4	79.2	64.0	11.1	41.8	29.2
SwiftFormer-S	1.2	78.5	66.3	9.1	38.7	27.8
RepViT-M1.1	**1.1**	**80.7**	**59.9**	15.5	**44.9**	**31.7**
PoolFormer-S12	1.5	77.2	67.7	6.9	37.7	25.2
MobileOne-S4	1.6	79.4	68.1	10.8	41.8	29.2
FastViT-S12	1.5	80.9	60.3	17.3	42.1	29.9
FastViT-SA12	1.8	81.9	58.0	21.8	43.8	30.8
MobileViG-M	1.6	80.6	59.1	17.1	45.7	32.5
SwiftFormer-L1	1.6	80.9	58.0	16.8	44.5	31.2
RepViT-M1.5	**1.5**	**82.3**	**55.1**	**22.7**	**47.2**	**34.0**
PoolFormer-S24	2.4	80.3	60.6	14.5	41.4	28.8
PoolFormer-S36	3.5	81.4	58.4	18.5	42.1	30.3
MobileViG-B	2.7	82.6	53.9	26.0	48.8	35.7
EfficientFormer-L3	2.7	82.4	54.6	23.9	46.0	33.6
EfficientFormer-L7	6.6	83.3	52.6	29.9	47.6	34.8
SwiftFormer-L3	2.9	83.0	52.9	25.9	47.2	34.3
RepViT-M2.3	**2.3**	**83.3**	**51.6**	**30.2**	**50.0**	**36.6**

datasets, including (1) ImageNet-C [6], which comprises algorithmically generated corruptions applied to the ImageNet test set; (2) ImageNet-A [8], which includes naturally occurring examples that are misclassified by ResNets [4]; (3) ImageNet-R [7], which consists of natural renditions of ImageNet object classes, incorporating diverse textures and local image statistics; (4) ImageNet-Sketch [27], which includes black and white sketches of all ImageNet classes, obtained through google image queries. Similar to [19, 21, 25], we report mean corruption error (mCE) for ImageNet-C. A smaller mCE indicates a higher level of robustness exhibited by the model under corruptions. For other benchmark datasets, top-1 accuracies are presented.

As shown in Table 3, RepViT demonstrates strong robustness to corruptions and promising domain generalization capabilities, outperforming previous state-of-the-art models. For example, compared with EfficientFormer-L3, RepViT-M2.3 obtains 3.0 improvement in terms of mCE on ImageNet-C with a smaller latency. Besides, it significantly outperforms SwiftFormer-L3 by 4.3%, 2.8% and 2.3% top-1 accuracies on ImageNet-A, ImageNet-R and ImageNet-Sketch, respectively. These results well demonstrate the strong robustness behaviors of RepViT models.

Table 4. Aerial semantic segmentation results on iSAID. Backbone latencies are measured with image crops of 512×512 on iPhone 12 by Core ML Tools.

Model	Latency ↓	mIoU
EfficientFormerV2-S0	6.3ms	58.56
RepViT-M1.0	**4.2ms**	**64.49**
EfficientFormerV2-S2	12.0ms	63.79
RepViT-M1.5	**6.4ms**	**65.57**
EfficientFormerV2-L	18.2ms	66.05
RepViT-M2.3	**9.9ms**	**67.42**

C. Generalizability to Out-of-Domain Tasks

In order to demonstrate the generalizability of our RepViT to out-of-domain downstream tasks, we conduct experiments on aerial semantic segmentation using the large-scale iSAID [28] benchmark dataset. The iSAID dataset includes a background class and 15 foreground classes over the aerial objects. Its training, validation and test sets separately consist of 1411, 458, and 937 images. Following [26], we integrate RepViT into the UperNet [29] framework and initialize the backbones with pretrained weights on ImageNet-1K. Following [26], We adopt AdamW optimizer, and the initial learning rate and weight decay are set to 6×10^{-5} and 1×10^{-2}, respectively. All models are trained for 80k iterations with a batch size of 8. We introduce the state-of-the-art EfficientFormerV2 [13] as the baseline. Besides, we conduct all evaluations on a single scale for fair comparisons, following [26].

As shown in Table 4, RepViT shows superior performance with low latencies. For example, RepViT-M1.0 presents a significant performance improvement of 5.93 mIoU over EfficientFormerV2-S0. RepViT-M1.5 outperforms EfficientFormerV2-S2 by a considerable margin of 1.78 mIoU with a nearly 50% latency. These results well demonstrate the strong generalizability of our RepViT.

D. Visualization Results

D.1. Visualization on Downstream Tasks

Figure 1 and Figure 2 presents the visualization results when integrating RepViT into the Mask-RCNN framework [5] for object detection/instance segmentation and into the Semantic FPN framework [11] for semantic segmentation, respectively. It can be observed that our RepViT based models can accurately detect and segment the instance in various images. It can also provide high-quality semantic segmentation masks.

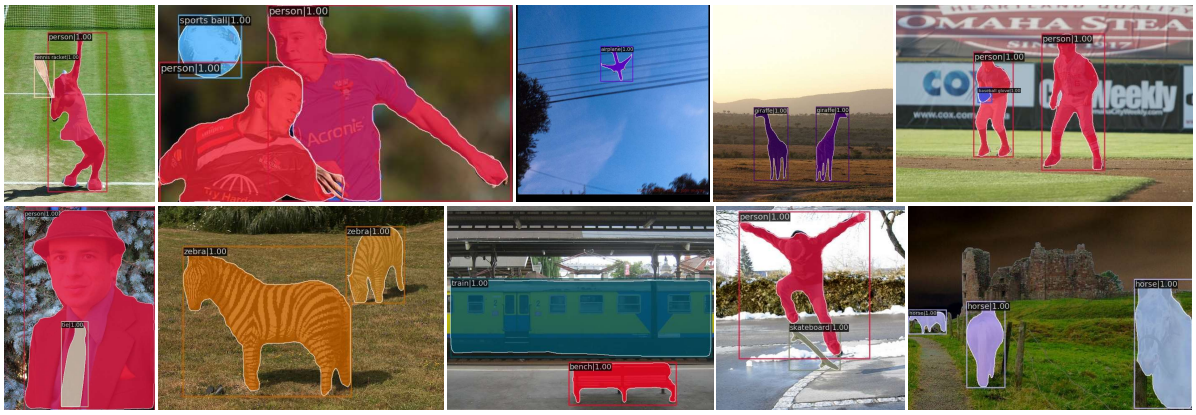

Figure 1. Visualization results for object detection and instance segmentation on COCO [15].

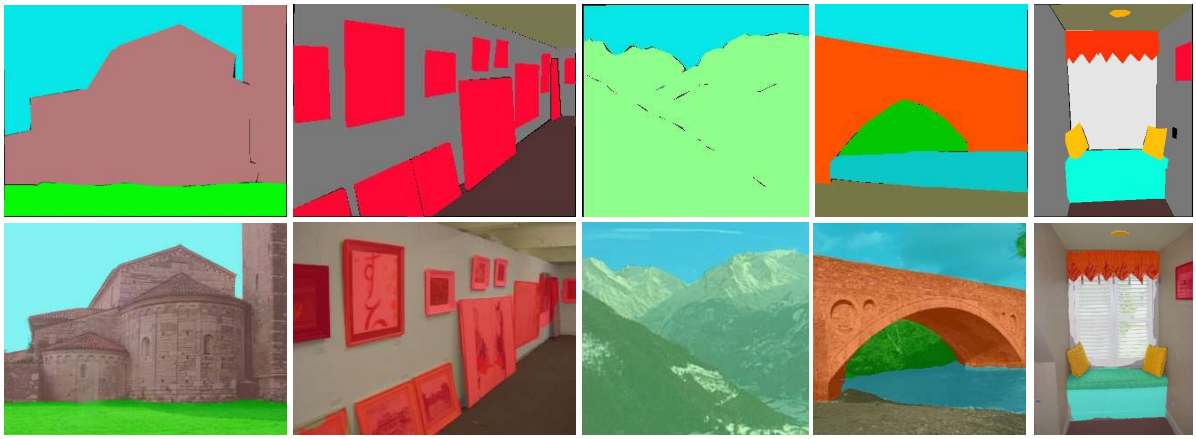

Figure 2. Visualization results for semantic segmentation on ADE20K [33]. The top row presents the ground truth masks and the bottom row shows the mask predictions.

D.2. Visualization of the Training Process

To better show the benefit of each architectural design incorporated in our RepViT, we conduct visualization for the top-1 accuracy during the training process. As shown in Figure 3, the introduced architectural designs well improve the optimization of the model, leading to the higher top-1 accuracy during the training process.

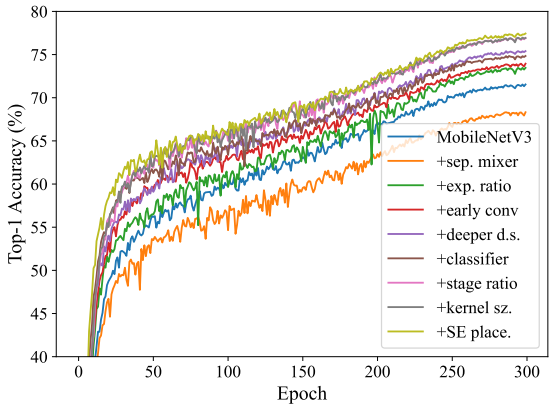

Figure 3. The top-1 accuracy during the training process. Note that these results are obtained without the distillation.

E. More RepViT-SAM Details and Results

E.1. Implementation Details

Following [30], we distill the image encoder, *i.e.*, RepViT-M2.3, directly from the ViT-H in the original SAM [12], leveraging a simple MSE loss. Like [30], we set the stride of 2 in the last downsampling depthwise convolution to 1 for making the output resolution compatible with that of the image encoder in the original SAM [12]. To speedup the training, we save the image embeddings from the ViT-H beforehand without running the forward process of ViT-H during the distillation, which is the same as [30]. Regarding to zero-shot edge detection on BSDS500 [1, 22], we

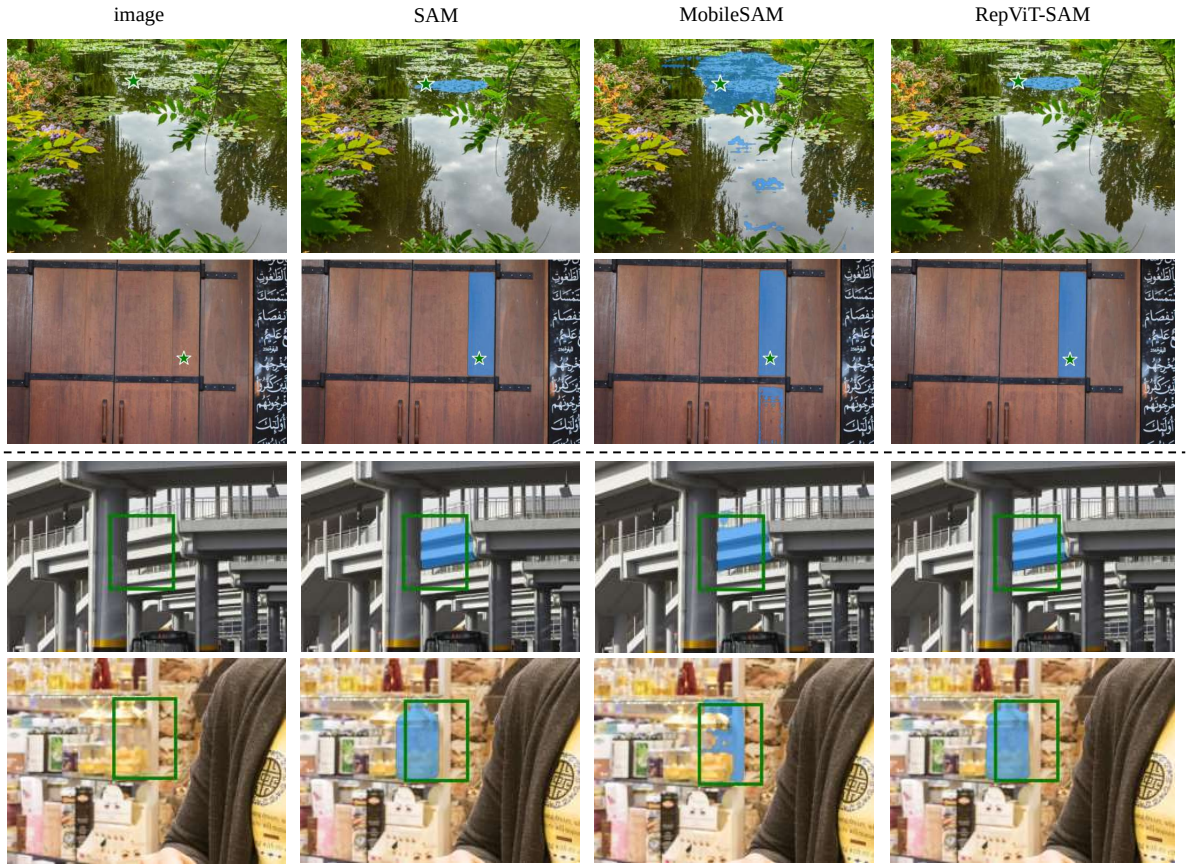

Figure 4. Mask predictions of SAM, MobileSAM, and RepViT-SAM with point prompts (top) and box prompts (bottom).

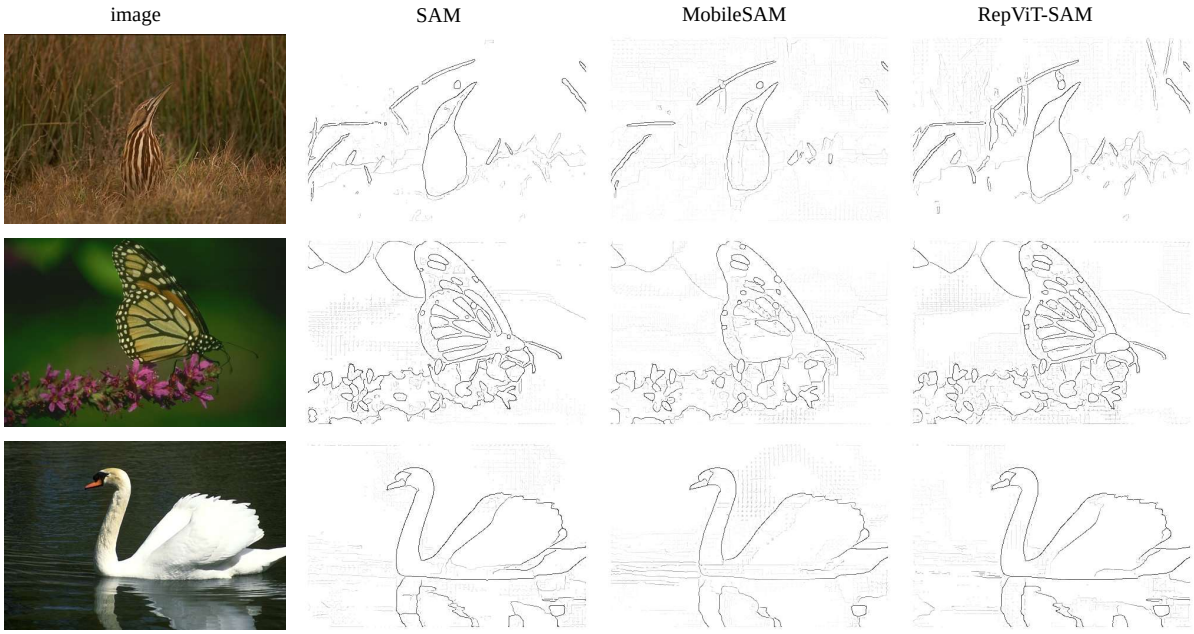

Figure 5. Visualization results of SAM, MobileSAM, and RepViT-SAM for zero-shot edge detection on BSDS500.

follow [12] to prompt the model with a 16×16 regular grid of foreground points. NMS is applied to remove redundant masks. Then, we leverage a Sobel filter to the masks' unthreshold probability maps, along with standard lightweight postprocessing, including edge NMS, like [12]. For transferring to zero-shot instance segmentation on COCO [16], we leverage the state-of-the-art H-Deformable-DETR [9] with Swin-L [18] as the object detector and prompt the model with its output boxes, like [10, 12]. For the segmentation in the wild benchmark, we follow [10] to equip the Grounding-DINO [17] as box prompts to evaluate the model on the zero-shot track.

E.2. Visualization of RepViT-SAM

We present the predicted masks of SAM [12], MobileSAM [30] and RepViT-SAM with point and box prompts in Figure 4. It can be observed that our RepViT-SAM can make high-quality mask predictions similar to that of SAM in various scenarios where MobileSAM may fail to predict accurate masks. Additionally, we provide qualitative comparison results on zero-shot edge detection in Figure 5. RepViT-SAM convincingly produces reasonable edge maps, achieving comparable performance with SAM. In contrast, MobileSAM may struggle to accurately generate edge maps in regions containing intricate details.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010. 3

[2] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123, 2019. 1

[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. 1

[4] 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

[5] 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

[6] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 2

[7] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 2

[8] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021. 2

[9] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Detrs with hybrid matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19702–19712, 2023. 5

[10] Lei Ke, Mingqiao Ye, Martin Danelljan, Yifan Liu, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Segment anything in high quality. *arXiv preprint arXiv:2306.01567*, 2023. 5

[11] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6399–6408, 2019. 2

[12] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 3, 5

[13] Yanyu Li, Ju Hu, Yang Wen, Georgios Evangelidis, Kamyar Salahi, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Rethinking vision transformers for mobilenet size and speed. *arXiv preprint arXiv:2212.08059*, 2022. 1, 2

[14] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35: 12934–12949, 2022. 1

[15] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3

[16] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[17] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 5

[18] 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. 5

[19] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 2

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[21] Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Ranjie Duan, Shaokai Ye, Yuan He, and Hui Xue. Towards robust vision transformer. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 12042–12051, 2022. 1, 2

[22] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, pages 416–423. IEEE, 2001. 3

[23] Mustafa Munir, William Avery, and Radu Marculescu. Mobilevig: Graph-based sparse attention for mobile vision applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2210–2218, 2023. 1

[24] Abdelrahman Shaker, Muhammad Maaz, Hanoona Rasheed, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Swiftformer: Efficient additive attention for transformer-based real-time mobile vision applications. *arXiv preprint arXiv:2303.15446*, 2023. 1

[25] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. Fastvit: A fast hybrid vision transformer using structural reparameterization. *arXiv preprint arXiv:2303.14189*, 2023. 1, 2

[26] Di Wang, Jing Zhang, Bo Du, Gui-Song Xia, and Dacheng Tao. An empirical study of remote sensing pretraining. *IEEE Transactions on Geoscience and Remote Sensing*, 2022. 2

[27] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[28] Syed Waqas Zamir, Aditya Arora, Akshita Gupta, Salman Khan, Guolei Sun, Fahad Shahbaz Khan, Fan Zhu, Ling Shao, Gui-Song Xia, and Xiang Bai. isaid: A large-scale dataset for instance segmentation in aerial images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 28–37, 2019. 2

[29] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018. 2

[30] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023. 3, 5

[31] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1

[32] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13001–13008, 2020. 1

[33] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 1, 3