VanillaNet: the Power of Minimalism in Deep Learning (Supplementary Material)

Hanting Chen[1], **Yunhe Wang**[1], **Jianyuan Guo**[1], **Dacheng Tao**[2]
[1] Huawei Noah's Ark Lab. [2] School of Computer Science, University of Sydney.
`{chenhanting,yunhe.wang,jianyuan.guo}@huawei.com, dacheng.tao@sydney.edu.au`

A Network Architectures

The detailed architecture for VanillaNet with 7-13 layers can be found in Table 1, where each convolutional layer is followed with an activation function. For the VanillaNet-13-1.5×, the number of channels are multiplied with 1.5. For the VanillaNet-13-1.5×†, we further use adaptive pooling for stage 2,3 and 4 with feature shape 40×40, 20×20 and 10×10, respectively.

	Input	VanillaNet-5	VanillaNet-6	VanillaNet-7/8/9/10/11/12/13
stem	224×224	4×4, 512, stride 4		
stage1	56×56	[1×1, 1024]×1 MaxPool 2×2	[1×1, 1024]×1 MaxPool 2×2	[1×1, 1024]×2 MaxPool 2×2
stage2	28×28	[1×1, 2048]×1 MaxPool 2×2	[1×1, 2048]×1 MaxPool 2×2	[1×1, 2048]×1 MaxPool 2×2
stage3	14×14	[1×1, 4096]×1 MaxPool 2×2	[1×1, 4096]×1 MaxPool 2×2	[1×1, 4096]×1/2/3/4/5/6/7 MaxPool 2×2
stage4	7×7	-	[1×1, 4096]×1	[1×1, 4096]×1
classifier	7× 7	AvgPool 7×7 1×1, 1000		

Table 1: Detailed architecture specifications.

B Training Details

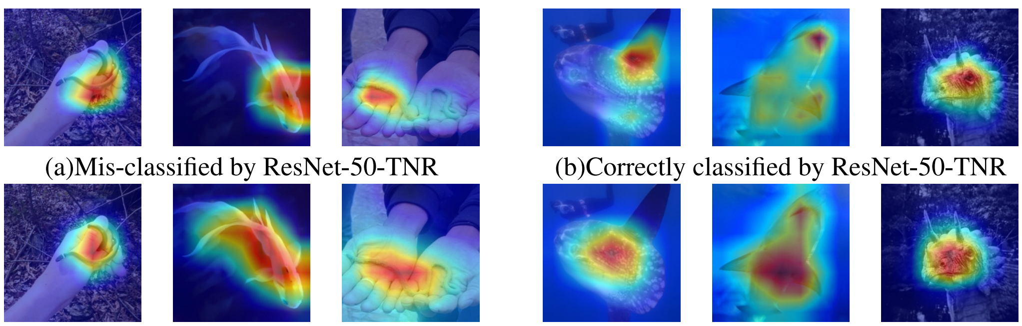
For classification on ImageNet, we train the VanillaNets for 300 epochs utilizing the cosine learning rate decay [5]. The λ is linearly decayed from 1 to 0 on epoch 0 and 100, respectively. The training details can be fould in Table 2. For the VanillaNet-11, since the training difficulty is relative large, we use the pre-trained weight from the VanillaNet-10 as its initialization. The same technique is adopt for VanillaNet-12/13.

For detection and segmentation on COCO, we train all the networks using 12 epochs, multi-scale training augmentation and a linear learning rate decay for fair comparison. Following ConvNextV2 [9] which utilize self-supervised training, we use the ImageNet pre-trained weight using knowledge distillation with $n = 4$ for a higher receptive field. We train the VanillaNet-13 using the Adamw optimizer with a batch size of 32, an initial learning rate of 8e-5 for RetinaNet and 1.3e-4 for Mask RCNN, an 0.05 weight decay and an 0.6 layer wise decay.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Training Config	VanillaNet-{5/6/7/8/9/10/11/12/13}
weight init	trunc. normal (0.2)
optimizer	LAMB [10]
loss function	BCE loss
base learning rate	3.5e-3 {5,8-13} /4.8e-3 {6-7}
weight decay	0.35/0.35/0.35/0.3/0.3/0.25/0.3/0.3/0.3
optimizer momentum	β_1, β_2=0.9, 0.999
batch size	1024
training epochs	300
learning rate schedule	cosine decay
warmup epochs	5
warmup schedule	linear
dropout	0.05
layer-wise lr decay [3, 1]	0 {5,8-12} /0.8 {6-7,13}
randaugment [4]	(7, 0.5)
mixup [12]	0.1/0.15/0.4/0.4/0.4/0.4/0.8/0.8/0.8
cutmix [11]	1.0
color jitter	0.4
label smoothing [7]	0.1
exp. mov. avg. (EMA) [6]	0.999996 {5-10} /0.99992 {11-13}
test crop ratio	0.875 {5-11} /0.95 {12-13}

Table 2: ImageNet-1K training settings.



(a)Mis-classified by ResNet-50-TNR

(b)Correctly classified by ResNet-50-TNR

(c)Mis-classified by VanillaNet-9

(d)Correctly classified by VanillaNet-9

Figure 1: Visualization of attention maps of the classified samples by ResNet-50 and VanillaNet-9. We show the attention maps of their mis-classified samples and correctly classified samples for comparison.

C Visualization of Attention

To have a better understanding of the proposed VanillaNet, we further visualize the features using GradCam++ [2], which utilizes a weighted combination of the positive partial derivatives of the feature maps generated by the last convolutional layer with respect to the specific class to generate a good visual explanation.

Figure 1 shows the visualization results for VanillaNet-9 and ResNets-50-TNR [8] with similar performance. The red color denotes that there are high activation in this region while the blue color denotes the weak activation for the predicted class. We can find that these two networks have different attention maps for different samples. It can be easily found that for ResNet-50, the area of active region is smaller. For the VanillaNet with only 9 depth, the active region is much larger than that of deep networks. We suggest that VanillaNet may be strong in extract all relative activations in the input images and thoroughly extract their information by using large number of parameters and FLOPs. In contrast, VanillaNet may be weak on analyzing part of the useful region since the non-linearity is relatively low.

References

[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[2] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018.

[3] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

[4] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[5] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[6] Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

[7] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[8] Ross Wightman, Hugo Touvron, and Hervé Jégou. Resnet strikes back: An improved training procedure in timm. *arXiv preprint arXiv:2110.00476*, 2021.

[9] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. *arXiv preprint arXiv:2301.00808*, 2023.

[10] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

[11] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

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