Supplementary Material

A Supplementary Material

A.1 Tiny ImageNet

We used the Tiny ImageNet dataset for model training and evaluating model clean accuracy [34]. Tiny ImageNet training dataset contains 100.000 colored images of 200 classes (500 for each class) at a resolution of 64×64 pixels. In addition, there are 50 additional images per class for validation. Tiny ImageNet is publicly available at `https://www.kaggle.com/c/tiny-imagenet`.

A.2 Common Image Corruptions

Tiny ImageNet-C is a dataset that contains the validation images from Tiny ImageNet under multiple forms of corruption [37]. Tiny ImageNet-C contains 15 different types of corruption at 5 levels of severity. These corruptions are Gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur. zoom blur, snow. frost, fog, brightness. contrast, elastic transform, pixelation, and JPEG compression. See Figure A.1 for examples.

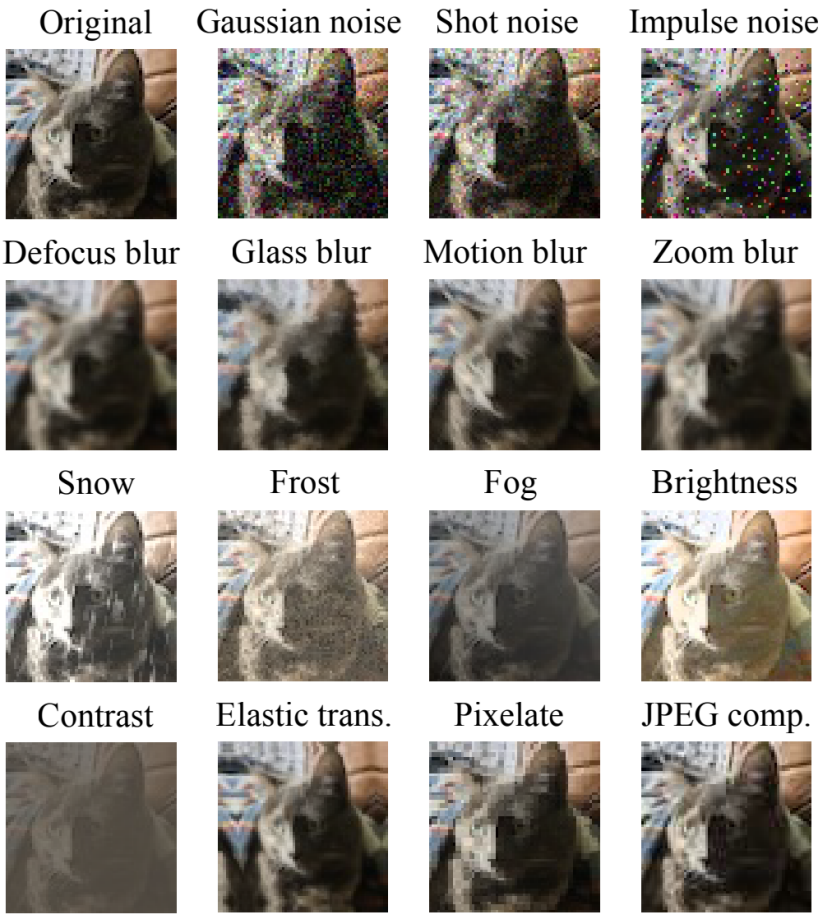

Figure A.1: **Common image corruptions at Tiny ImageNet resolution.** All 15 types of common image corruptions with a severity of 3.

A.3 Models

The following VOneNet and ResNet18 descriptions in the present paper are derived from Baidya et al. 2021 [36].

A.3.1 VOneNet

VOneNet [18] is a convolutional neural network with a front-end (VOneBlock) simulating the primary visual cortex (V1). In the present study, we used a modified ResNet18 architecture as the architecture back-end. The VOneBlock contains a fixed weight Gabor filter bank (GFB) [38] used to mimic the receptive fields found in V1. Each Gabor filter is constructed using parameters generated from empirically observed distributions in preferred orientation, spatial frequency, and size of receptive fields [39], [40], [41]. The Gabor filters are generated to simulate 256 simple and 256 complex cell receptive fields. Several changes were made to the VOneNet for compatibility with Tiny ImageNet as the original model was built for ImageNet. Importantly, the field-of-view of the model corresponding to the 64×64px inputs was adjusted to 2deg, giving the model a resolution of 32 pixels per degree (ppd). Further details on these changes can be found in [36]. Code for the VOneNet is publicly available at https://github.com/dicarlolab/ vonenet under GNU General Public License v3.0.

A.3.2 ResNet18

We used a modified Resnet18 architecture as both a base model and as a back-end for VOneNet. The Resnet18 architecture was modified such that the stride of the first convolutional layer was changed from two to one and the first maxpool layer was kept at a stride of two. This results in a combined stride of two in the first block, which is the same as the VOneBlock. Baidya et al. [36] found that this modification leads to a significant improvement in accuracy, with the modified ResNet18 achieving an accuracy of 58.93% when trained and evaluated on Tiny ImageNet in comparison to 50.45% prior to the modification.

A.4 Adding Divisive Normalization to VOneNet

The implementation of divisive normalization in this study was based on its generalized form which features in Burg et al. [32]. Neuronal responses of a neuron l to stimulus x ($y_l(x)$) are normalized by the a factor that depends on the responses of each neuron k in a pool of K neurons ($y_k(x)$ for $k \in K$), according to the following equation:

$$z_l(x) = \frac{y_l^{n_l}(x)}{\sigma_l^{n_l} + \Sigma_{k \in K} P_{kl} \cdot y_k^{n_k}(x)} \tag{1}$$

where z_l represents the normalized response of neuron l, σ_l is a semi-saturation constant, n_l represents a learned parameter relative to neuron l to exponentiate $y_l(x)$ and σ_l, and P_{kl} represents the normalization weights of neuron k onto neuron l.

We adapted this formulation to create the DNBlock to serve as a divisive normalization module directly following VOneBlock (see Figure **??**). The output of the VOneBlock consists of 512 32×32 response channels per image. For an image x and a Gabor filter l, the response channel is $Z_l(x)$. To divisively normalize the response channel $Z_l(x)$ with respect to each response channel $Z_k(x)$ in a pool of K response channels, we used the following version of the previous equation 1:

$$\bar{Z}_l(x) = \frac{Z_l(x)}{\beta + \sum_{k \in K} \alpha_{kl} \cdot Z_k(x)} \tag{2}$$

where $\bar{Z}_l(x)$ represents the divisively normalized response channel $Z_l(x)$, β is a bias term and α_{kl} represents the normalization weights of response channel $Z_k(x)$ onto response channel $Z_l(x)$. The exponential term n has not been included as, in training, caused significant instability. In [32], the normalization weights P_kl were learned without constraint and covered 0.5 degrees of the visual field, which leads to only local interactions being captured. In our modified implementation, the normalization weights α_{kl} were constrained to a two-dimensional Gaussian kernel, greatly reducing the number of parameters to be learned, and covered about two degrees of the visual field, thus capturing more distal interactions. The equation for building the Gaussian kernel is the following:

$$\alpha_{kl} = G(k, l|\theta, v, w, \rho, \sigma, A) = \frac{A}{2\pi\rho\sigma}\exp\left(-\frac{1}{2}\left[\frac{(x_{rot}-v)^2}{\rho^2} + \frac{(y_{rot}-w)^2}{\sigma^2}\right]\right) \tag{3}$$

where

$$x_{rot} = xcos(\theta) + ysin(\theta)$$
$$y_{rot} = -xsin(\theta) + ycos(\theta)$$

Each parameter of this kernel is optimized during training. Each kernel is unique to each combination of channel being normalized and channel normalizing it, resulting in $512^2 = 262,144$ total Gaussian kernels.

A.5 Training

Extensive details on image preprocessing, loss functions, and optimization, can be found in Baidya et al. [36]. In summary, image preprocessing consisted of randomly scaling each image by a factor of 1-1.2 and randomly rotating each image between 30 and -30 degrees. Images were shifted between -5% and 5% of the images total height and width the vertical and horizontal directions respectively. Images were normalized by subtraction and division by [0.5, 0.5, 0.5]. The model was optimized using Stochastic Gradient Descent with a momentum of 0.9, a weight decay of 0.0005, and an initial learning rate of 0.1. The learning rate was reduced by a factor of 10 whenever 5 training epochs passed simultaneously with no improvement in accuracy. Models were trained with an image batch size of 128 over 60 epochs. Weight decay was disabled for all parameters in the DNBlock.

A.6 Brain-Score Benchmarks

The Brain-Score platform [35, 7] offers several benchmarks to test the biological accuracy. These benchmarks may be found at `https://www.brain-score.org/`. To compare the models with and without divisive normalization, we used benchmarks that measure the alignment of model to V1 at the level of single-neuron response properties. These benchmarks measure the similarity between the distributions of single neuron response properties in the model and V1. Detailed description of 22 of these benchmarks can be found in Marques, et al. 2021 [15]. These are grouped in seven categories: orientation tuning, spatial frequency tuning, receptive field size, surround modulation, texture modulation, response selectivity, and response magnitude. Here, we also included eight new benchmarks from contrast and luminance response property categories described in a study under review (will be updated to include citation).

A.6.1 Contrast

For these benchmarks, achromatic sinusoidal gratings of varying contrast are presented at the preferred orientation and spatial frequency of each model neuron. Then, for each neuron, its responses to different contrasts are fit using the hyperbolic function: $R = R_{max}\frac{c^n}{c^n + c_{50}^n}$

Distributions of four response properties were used to compare the fits of brain and model neurons; the number of standard contrast responses, the maximum response R_{max}, the semisaturation constant(c_{50}^n), and the exponent n of the function.

A.6.2 Luminance

For luminance benchmarks, uniform stimuli varying from 0.1 cd/m^2 to 100 cd/m^2 in seven steps on a logarithmic scale were presented. For each neuron, luminance tuning curves were calculated and the responses to dark (below 3 cd/m^2) and bright (above 3 cd/m^2) stimuli were fitted logarithmically.

Four response properties were used in these benchmarks: the number of surface luminance response neurons, the slope of the firing rate versus log(luminance) for dark and bright stimuli and the normalized difference between the two slopes.

A.7 Detailed Results

Contrast

Benchmark	VOneResnet18 Performance	VOneResnet18 DN Performance
Standard Neuron	**1.000**	0.996
Maximum Response	0.161	**0.689**
Semisaturation Constant	0.192	**0.345**
Exponent	0.677	**0.736**

Luminance

Benchmark	VOneResnet18 Performance	VOneResnet18 DN Performance
Surface Responsive	0.762	**0.852**
Dark Slope	**0.846**	0.756
Bright Slope	**0.385**	0.367
Delta Slope Norm	**0.746**	0.706

Orientation Tuning

Benchmark	VOneResnet18 Performance	VOneResnet18 DN Performance
Circular Variance	**0.766**	0.731
Or. Bandwidth	0.923	**0.956**
Orth. Pref. Ratio	**0.725**	0.675
OR. Selective	**1.000**	0.994
CV Bandwidth Ratio	0.770	**0.793**
Opr. CV Diff	**0.886**	0.883
Preferred Orientation	0.991	**0.994**

Spatial Frequency Tuning

Benchmark	VOneResnet18 Performance	VOneResnet18 DN Performance
Peak SF.	**0.981**	0.929
SF. Selectivity	0.981	**0.984**
SF. Bandwidth	0.937	**0.977**

Receptive Field Size

Benchmark	VOneResnet18 Performance	VOneResnet18 DN Performance
Grating Summation Field	**0.589**	0.426
Surround Diameter	0.370	**0.392**

Table A.1: **Single neuron response property V1 benchmarks (Contrast, Luminance, Orientation Tuning, Spatial Frequency Tuning, and Receptive Field Size categories)**.

Surround Modulation		
Benchmark	VOneResnet18 Performance	VOneResnet18 DN Performance
Surround Suppression Index	0.385	**0.954**

Texture Modulation		
Benchmark	VOneResnet18 Performance	VOneResnet18 DN Performance
Texture Modulation Index	**0.898**	0.836
Absolute Texture Modulation Index	**0.944**	0.880

Response Selectivity		
Benchmark	VOneResnet18 Performance	VOneResnet18 DN Performance
Texture Selectivity	0.795	**0.958**
Texture Sparseness	**0.927**	0.783
Texture Variance Ratio	**0.719**	0.674
Modulation Ratio	**0.723**	0.718

Response Magnitude		
Benchmark	VOneResnet18 Performance	VOneResnet18 DN Performance
Max DC	0.838	**0.976**
Max Texture	**0.914**	0.688
Max Noise	**0.923**	0.725

Table A.2: **Single neuron response property V1 benchmarks (Surround Modulation, Texture Modulation, Response Selectivity, and Response Magnitude categories)**.

		Noise			Blur			
Model	Clean [%]	Gaussian [%]	Shot [%]	Impulse [%]	Defocus [%]	Glass [%]	Motion [%]	Zoom [%]
ResNet18	**58.5**	20.0	23.1	22.4	13.9	18.9	19.6	15.9
VOneResNet18	55.3	24.3	28.7	24.4	**15.6**	**19.7**	**21.6**	**17.2**
VOneResNet18DN	57.8	**25.2**	**29.2**	**24.6**	15.0	**19.7**	21.4	16.6

	Weather				Digital			
Model	Snow [%]	Frost [%]	Fog [%]	Bright. [%]	Contrast [%]	Elastic [%]	Pixelate [%]	JPEG [%]
ResNet18	24.1	25.2	21.6	27.0	9.8	24.3	37.8	32.0
VOneResNet18	27.5	27.6	22.9	28.9	9.5	**28.5**	36.6	37.3
VOneResNet18DN	**29.0**	**30.1**	**29.2**	**31.6**	**18.3**	**28.5**	**38.5**	**38.2**

Table A.3: **Absolute accuracies (top-1) of Resnet18, standard VOneResnet18 and VOneResnet18DN (averaged over perturbation severities)**.

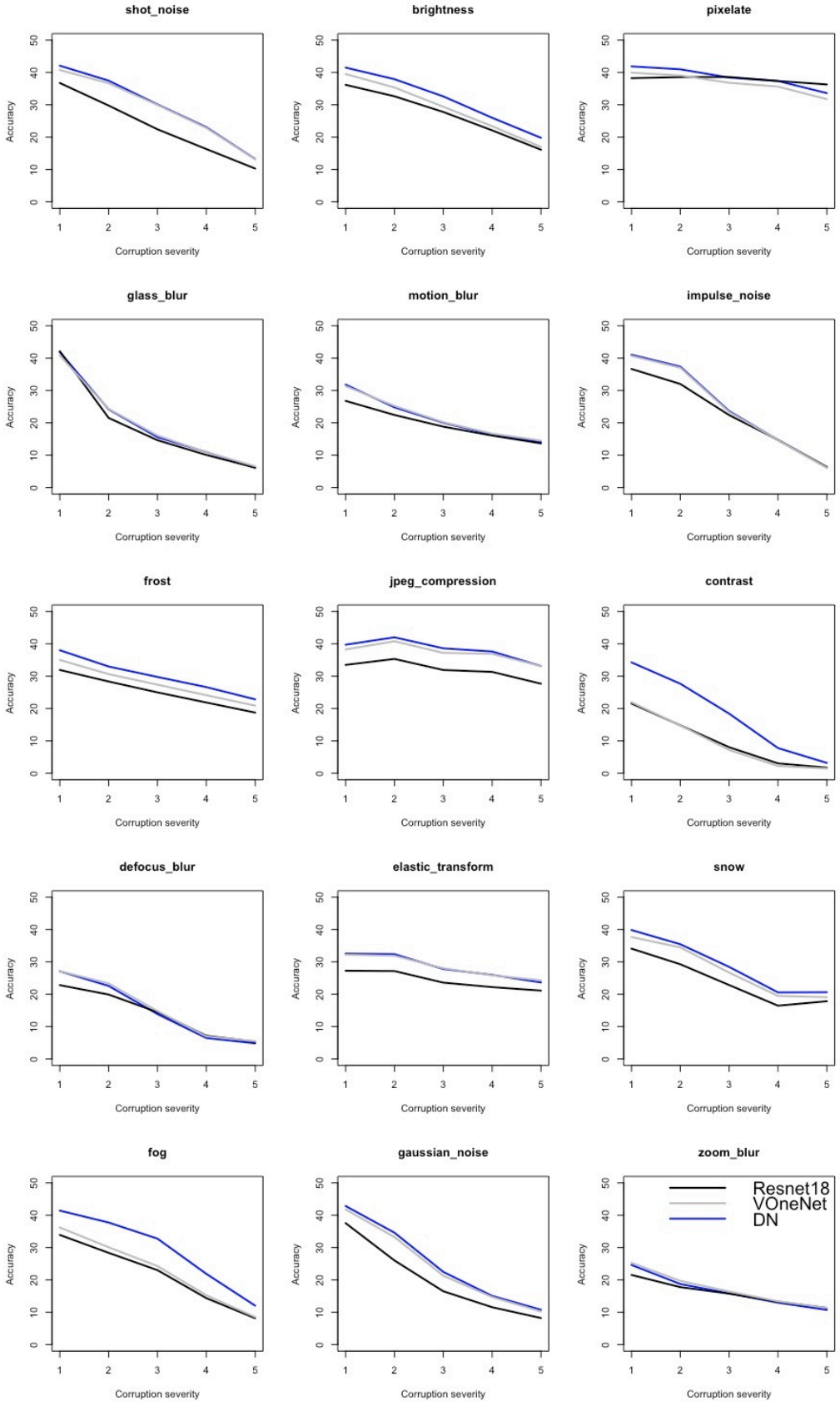

Figure A.2: **Absolute accuracies (top-1) on common corruptions for ResNet18, VOneResNet18, and VOneResNet18DN.** All 15 types of common corruptions at all perturbation severity levels.