# Combining Different V1 Brain Model Variants to Improve Robustness to Image Corruptions in CNNs

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

While some convolutional neural networks (CNNs) have surpassed human visual abilities in object classification, they often struggle to recognize objects in images corrupted with different types of common noise patterns, highlighting a major limitation of this family of models. Recently, it has been shown that simulating a primary visual cortex (V1) at the front of CNNs leads to small improvements in robustness to these image perturbations. In this study, we start with the observation that different variants of the V1 model show gains for specific corruption types. We then build a new model using an ensembling technique, which combines multiple individual models with different V1 front-end variants. The model ensemble leverages the strengths of each individual model, leading to significant improvements in robustness across all corruption categories and outperforming the base model by 38% on average. Finally, we show that using distillation, it is possible to partially compress the knowledge in the ensemble model into a single model with a V1 front-end. While the ensembling and distillation techniques used here are hardly biologically-plausible, the results presented here demonstrate that by combining the specific strengths of different neuronal circuits in V1 it is possible to improve the robustness of CNNs for a wide range of perturbations.

## 1 Introduction

Over the past decade, convolutional neural networks (CNNs) not only dominated several computer vision applications but also surpassed human visual abilities in specific domains such as object classification [1, 2, 3, 4]. However, unlike humans, CNNs show a striking lack of robustness: CNNs are vulnerable to extremely small perturbations explicitly optimized to fool them, usually know as adversarial attacks [5, 6, 7]; and, perhaps more relevant for real-world applications, they struggle to recognize objects in images corrupted with different types of common noise patterns [8, 9, 10]. Interestingly, these two perturbation types expose different aspects of robustness: models designed to better withstand one usually fail to generalize to the other [11, 12].

Recently, Dapello, Marques et al. observed that models that were more robust to white box attacks also had early stages that better predicted neuronal responses in the macaque primary visual cortex (V1) [12]. Inspired by this, the authors developed a novel class of hybrid CNNs, containing a model of primate V1 as the front-end, followed by a trainable standard CNN architecture back-end. This new model family, the VOneNet, was substantially more robust to adversarial attacks than the corresponding base-models and rivaled considerably more computationally expensive methods such as adversarial training. Surprisingly, VOneNet models also showed small gains in robustness to common corruptions, with different variants of the V1 front-end leading to specific trade-offs in accuracy when considering all the individual corruption types. Here, we extend this last finding to

Submitted to 3rd Workshop on Shared Visual Representations in Human and Machine Intelligence (SVRHM 2021) of the Neural Information Processing Systems (NeurIPS) conference.

make the following novel contributions. First, we adapt the VOneNet model to the Tiny ImageNet dataset [13] and reproduce results from Dapello, Marques et al., particularly the existence of specific trade-offs in dealing with common corruptions for several variants. Then, we build a new model using an ensembling technique which combines VOneNet models with different V1 front-end variants, eliminating trade-offs and showing a remarkable improvement in robustness to common corruptions (38% overall). Finally, we show that distillation training is able to partially compress the knowledge in the ensemble into a single VOneNet model, resulting in a compact architecture that improves over the baseline on all the corruption categories (13% overall).

### 1.1 Related Work

**Common corruptions** Several recent works have studied the robustness of CNNs against common corruptions [10, 14, 15, 11, 16, 17, 18, 19, 20, 21]. The current state-of-the-art for common image corruptions (DeepAugment+AugMix) [15] involves using an image-to-image network to add perturbations to the input image combined with a technique that mixes randomly generated augmentations. Other data augmentation techniques have also been shown to improve robustness. [11] showed that augmentation with Gaussian noise or adversarial noise can significantly improve model robustness. [16] apply Gaussian noise to small image patches to improve robustness. Gaussian data augmentation, however, can impact clean image performance [11] and can cause models to be vulnerable to low frequency corruptions [17]. [18] assemble common CNN techniques, including knowledge distillation, into a single CNN to achieve improved performance on clean and corrupted images. Other techniques to increase robustness involve using: anti-aliasing module to restore the shift-equivariance [19], stylized images to increase shape bias [20] and stability training [21].

**Biologically-inspired methods for improving robustness** [12] showed that simulating V1 in front of CNNs can substantially improve white box adversarial robustness with smaller gains in the case of common corruptions. In a similar work, [22] replaced the first convolutional layer of a standard CNN by Gabor filters to improve robustness to noise. Other biologically-inspired works to improve robustness include: regularizing CNN models' representations to approximate mouse V1 [18] and training to predict neural activity in V1 while performing image classification [23].

**Ensemble and distillation** Ensembling is a well-known machine learning technique to combine smaller individual models into a larger model leading to superior performance compared to the individual models in a diverse range of supervised learning problems [24, 25, 26, 27, 28, 29], including generalization to out-of-domain (OOD) datasets [30, 31, 32, 33, 34]. Knowledge Distillation is a popular technique [35, 36, 37, 38, 39, 40, 41] to transfer the superior performance of the ensemble (teacher) into a single smaller model (student). This technique has been shown to improve the generalization ability of the student models [38, 40, 42, 18].

## 2 Results

### 2.1 V1 model variants show performance trade-offs on different corruptions

VOneNets are CNNs with a biologically-constrained fixed-weight front-end that simulates V1 (the VOneBlock) followed by a conventional neural network architecture [12]. The VOneBlock consists of a fixed-weight Gabor filter bank (GFB) [43], simple and complex cell [44] nonlinearities, and neuronal stochasticity (Fig. 1A) [45]. Here, we trained a VOneNet model for object classification on the Tiny ImageNet dataset [13] using the ResNet18 [4] as the back-end architecture, which we call the standard VOneResNet18 model. In addition, we created seven model variants by removing or modifying one of the VOneBlock components (Fig. 1B, Table B.1, Section B.1). All the variants' back-ends, including the standard model, were optimized from scratch on Tiny ImageNet following an identical training procedure (Section B.4). We evaluated model robustness using the Tiny ImageNet-C dataset [10] which consists of 15 different corruption types, each at five levels of severity, divided into four categories: digital, weather, blur and noise (Section A.2).

While all of the model variants were found to perform worse than ResNet18 on clean Tiny ImageNet images, all of them were considerably more robust on at least one corruption category (Fig. 1C, Table C.1). Still, no single variant outperformed ResNet18 in all four categories of image corruptions. For example, the standard VOneResNet18 and Low SF variant were more robust than ResNet18 to blur and noise corruptions but they performed worse on digital and weather corruptions. Similarly, Mid

SF and Only Simple variants were more robust than ResNet18 to all corruptions except for weather corruptions. Furthermore, some model variants that outperformed other variants in all four categories of corruptions performed worse on clean images. These results show that while some variants of the VOneBlock lead to large gains in robustness for specific corruption types, this comes with losses for others. This type of trade-off is present for all the variants analyzed.

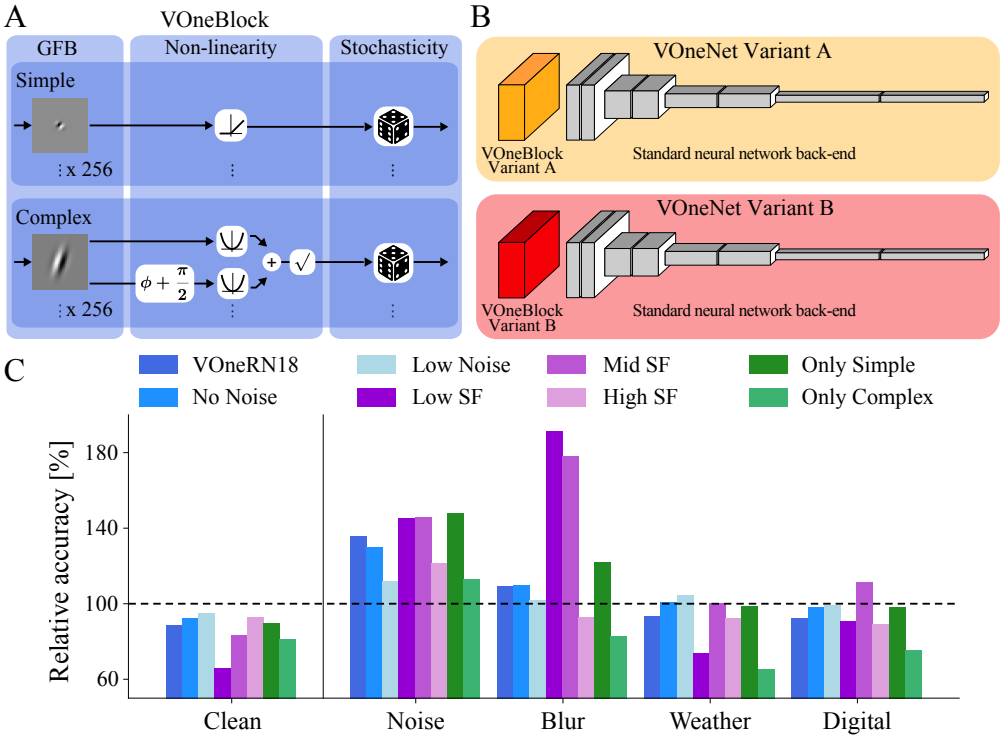


Figure 1: **Different V1 model variants show distinct robustness trade-offs.** **A** The VOneBlock is a model of V1 with a GFB, a non-linear stage and stochasticity generator. **B** Each VOneNet variant contains a different VOneBlock, built by removing or modifying one of its components. Here, we used eight different variants: standard VOneBlock, no neural stochasticity (No Noise), sub-Poisson stochasticity (Low Noise), only low SF filters (Low SF), only intermediate SF filters (Mid SF), only high SF filters (High SF), only simple-cells (Only Simple), and only complex-cells (Only Complex). **C** Relative accuracy (normalized by the base model, ResNet18) of the eight variants of VOneResNet18 for clean images and corruption categories (see Table C.1 for absolute accuracies).

### 2.2 Ensemble of different VOneNet variants eliminates robustness trade-offs

We used a common ensembling technique of uniformly averaging the outputs (logits) of the VOneNet variants described in the previous section to create the Variants Ensemble. The Variants Ensemble not only outperformed all the variants but also performed on par with ResNet18 on clean images (Fig. 2, Table C.2). Remarkably, the Variants Ensemble was found to be substantially more robust than ResNet18 in all corruption categories (and in 13 out of 15 individual corruption types, Fig. C.1), showing that ensembling is able to combine the diverse strengths of the individual variants. As a result, we developed a model that considerably outperforms ResNet18 in all corruption categories (between 17% and 60% with 35% overall) without compromising on clean accuracy.

While diversity in the members of an ensemble has been found to be important to its generalization ability [30, 33, 46, 47], ensembles of networks with the same architecture that only differ in their random initialization have also shown improved robustness [32, 31]. To test whether the diversity in the variants was critical for the Variant Ensemble's robustness, we created two Seeds Ensemble models by combining eight different training seeds of the standard VOneResNet18 and of the ResNet18. The Variants Ensemble consistently outperformed the other ensemble models on all the corruption categories (Fig. 2, Table C.2).

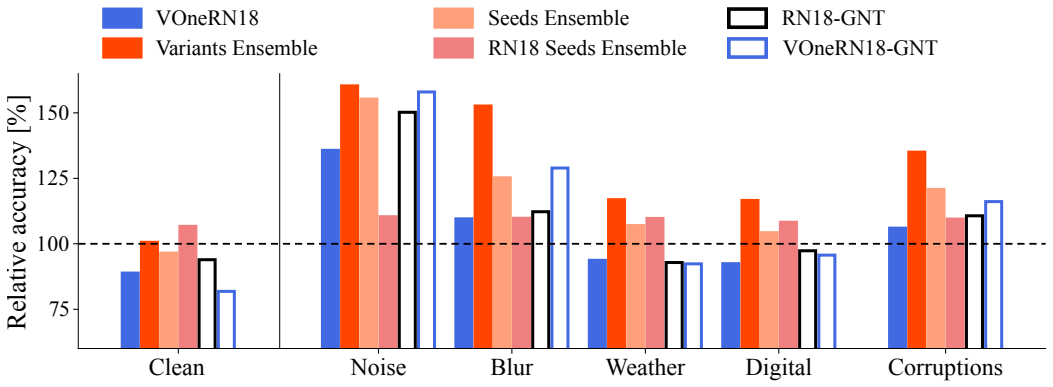


Figure 2: **Combining different VOneNet variants with model ensembling improves robustness to all corruption categories.** Relative accuracy (normalized by ResNet18) on clean images, all corruptions categories, and overall corruptions for the standard VOneResNet18, the Variants Ensemble, the Seeds Ensemble, the ResNet18 Seeds Ensemble, and the ResNet18 and VOneResNet18 trained with Gaussian Noise augmentation (see Table C.2 for absolute accuracies).

We also compared Variants Ensemble to a popular defense method that uses Gaussian Noise Training (GNT) as data augmentation [11]. We trained both ResNet18 (ResNet18-GNT) and standard VOneResNet18 (VOneResNet18-GNT) with GNT, observing an increased robustness for noise and blur categories (Fig. 2). However, models trained with GNT were significantly less robust than Variants Ensemble in all categories of corruptions and performed worse on clean images.

### 2.3 Training with distillation improves VOneNet robustness against corruptions

While the Variants Ensemble is consistently more robust and has better clean accuracy than any of the individual variants, it is also computationally more expensive. Knowledge Distillation can be used to compress the knowledge in an ensemble (teacher) into a single model (student) [35, 37, 38, 39, 40, 41]. Using this technique, we separately trained a ResNet18, a standard VOneResNet18 and a No Noise variant by distilling the Variants Ensemble into these three models (Section B.4). Interestingly, distillation has little effect on the performance of the standard ResNet18 and VOneResNet18 (Fig. 3, Table C.3). While the first result using ResNet18 suggests that the VOneBlock in the student architecture is critical for the success of this approach, the latter result with the VOneResNet18 implies that stochasticity undermines the ability of the student to distill the knowledge in the teacher. Surprisingly, we observed consistent and considerable improvement in the performance of the No Noise variant on clean images and all corruption categories (Fig. 3, Table C.3). In fact, the distilled version of the No Noise variant performed nearly as well as ResNet18 on clean images (98% relative accuracy) and considerably outperformed ResNet18 in all categories of corruptions (between 9% and 21% with 13% overall). Still, it failed to come close to the performance of the much larger Variants Ensemble, showing that diversity at the level of the VOneBlock is required for the magnitude of the gains observed before. As a result, we developed a VOneNet model that can partially compress the knowledge in the Variants Ensemble, outperforming ResNet18 in all categories of corruptions while maintaining clean accuracy and using a similar architecture.

## 3 Discussion

In this work, we make progress towards closing the gap between CNNs and human vision in terms of robustness to common image corruptions. First, we demonstrate that different variants of a V1 front-end, the VOneBlock, lead to models that are more robust to specific types of corruptions while being more vulnerable to others and with degraded accuracy on clean images. Remarkably, by creating an ensemble of different variants we eliminate these performance trade-offs, developing a model that is considerably more robust in all categories of corruptions than the corresponding base model without compromising on clean accuracy. We also show that using knowledge distillation to compress the knowledge in the ensemble into an individual VOneNet generates a model of comparable size to the base model that is more robust in all categories of corruptions while maintaining clean accuracy.

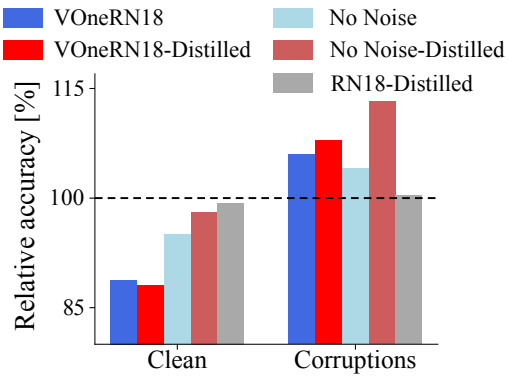


Figure 3: **VOneResNet18 without stochasticity can partially compress the knowledge in the Variants Ensemble with distillation.** Relative accuracy (normalized by ResNet18) on clean images and overall corruptions for VOneResNet18 trained with and without knowledge distillation, the VOneResNet18 No Noise variant trained with and without knowledge distillation, and ResNet18 trained with knowledge distillation. In all distilled models the Variants Ensemble was used as the teacher (see Table C.3 for absolute accuracies).

While there have been extensive studies characterizing the different neuronal populations in primate V1 [48, 49, 50, 51], there is a significant gap in our understanding of the roles played by the various neuronal types in dealing with different image statistics. By simulating a V1-like front-end whose components are mappable to the brain, Dapello, Marques, et al. took the first steps to investigate the role of various neuronal types in dealing with adversarial attacks and common image corruptions [12]. Here, we build on their work by generating variants of additional cell types (e.g. cells with peak spatial frequencies > 5.6cpd) and by investigating their roles in robustness to image perturbations on a separate dataset. Consistent with their findings, our results demonstrate that different V1 cell types are vulnerable to different perturbations while conferring benefits to others. For example, neurons with high peak spatial frequencies are important for clean image performance but cause models to be more susceptible to blur and noise corruptions. Interestingly, simple cells were found to be beneficial in all cases while complex cells increased vulnerability to all corruptions (although complex cells had been found to be beneficial for adversarial robustness [12]).

Ensembling techniques constructed using diverse individual architectures have better generalization ability [30, 33, 46, 47]. To our knowledge, this is the first study to leverage the different robustness properties of specific V1 neuronal populations to create diverse members of an ensemble. The gains achieved by the ensemble are substantial (35% over all corruptions with same clean accuracy). For example, GNT, a popular defense method, leads to only 11% better accuracy to corruptions while also decreasing the clean performance by 6% in our experiments. Furthermore, since training-based defense methods can be applied to individual VOneNet variants, the performance gains of the ensemble could potentially be stacked with other methods to achieve even better overall robustness. Finally, while our ensemble is created simply by averaging the outputs of its individual members, other more elaborate approaches can be used to learn the optimal weights to further improve performance.

Knowledge distillation has been shown to improve robustness to image corruptions [38, 40, 42, 18]. Here, we demonstrate, for the first time, that a V1-inspired CNN can lead to higher robustness gains through distillation. Furthermore, our results show that neuronal stochasticity in the student hinders robustness improvements through distillation. While we are aware of studies [42, 52] that add noise to the teacher, labels, or inputs to help improve distillation, this is the first study to report effects of neuronal stochasticity in the student.

Developing models that are more robust to image perturbations and can better generalize to OOD stimuli is a major goal in computer vision. The work presented here takes an important step in this direction. By combining machine learning techniques and a biologically-constrained front-end, we built two models, the Variants Ensemble and the No Noise distilled, that improve robustness to all categories of common image corruptions without compromising on clean accuracy. These models, however, remain far from robust generalization. Still, the direction taken here can be further expanded. For example, future work may explore how interactions between different V1 neuronal properties affect a network's performance to develop even more robust VOneNet variants. Furthermore, while the ensembling approach taken here to combine different V1 neuronal populations is not biologically-plausible, novel architectures incorporating cortical circuit motifs or performing cortical computations such as divisive normalization or gain-control could better approximate visual processing in the primate brain while generating even stronger improvements in robustness.

# References


[1] Alex Krizhevsky, Ilya Sutskever, and Hinton Geoffrey E. "ImageNet Classification with Deep Convolutional Neural Networks". In: *NIPS*. ISSN: 10495258. 2012, pp. 1097–1105. ISBN: 978-1-62748-003-1. DOI: 10.1109/5.726791.

[2] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 07-12-June (2015). ISBN: 9781467369640, pp. 1–9. ISSN: 10636919. DOI: 10.1109/CVPR.2015.7298594.

[3] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *ICLR*. ISSN: 15352900. 2015, pp. 1–14. ISBN: 1097-0142 (Electronic)\n0008-543X (Linking). DOI: 10.2146/ajhp170251. URL: http://arxiv.org/abs/1409.1556.

[4] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CVPR*. Dec. 2016, pp. 1–12. URL: http://arxiv.org/abs/1512.03385.

[5] Christian Szegedy et al. "Intriguing properties of neural networks". en. In: *arXiv:1312.6199 [cs]* (Feb. 2014). arXiv: 1312.6199. URL: http://arxiv.org/abs/1312.6199.

[6] Nicholas Carlini et al. "On Evaluating Adversarial Robustness". In: (2019), pp. 1–24. URL: http://arxiv.org/abs/1902.06705.

[7] Wieland Brendel et al. "Accurate, reliable and fast robustness evaluation". In: 1 (2019), pp. 1–11. URL: http://arxiv.org/abs/1907.01003.

[8] Samuel Dodge and Lina Karam. "A Study and Comparison of Human and Deep Learning Recognition Performance Under Visual Distortions". en. In: *arXiv:1705.02498 [cs]* (May 2017). arXiv: 1705.02498. URL: http://arxiv.org/abs/1705.02498.

[9] Robert Geirhos et al. "Generalisation in humans and deep neural networks". In: *NeurIPS*. 2018, pp. 1–13. URL: http://arxiv.org/abs/1808.08750.

[10] Dan Hendrycks and Thomas Dietterich. "Benchmarking neural network robustness to common corruptions and perturbations". In: *7th International Conference on Learning Representations, ICLR 2019* (2019), pp. 1–16.

[11] Evgenia Rusak et al. "A simple way to make neural networks robust against diverse image corruptions". In: (2020), pp. 1–34. URL: http://arxiv.org/abs/2001.06057.

[12] Joel Dapello et al. "Simulating a primary visual cortex at the front of CNNs improves robustness to image perturbations". In: *NeurIPS* (2020), pp. 1–30. ISSN: 26928205. DOI: 10.1101/2020.06.16.154542.

[13] Ya Le and Xuan Yang. "Tiny ImageNet Visual Recognition Challenge". en. In: (), p. 6.

[14] Dan Hendrycks et al. "AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty". In: *ICLR* (2020), pp. 1–15.

[15] Dan Hendrycks et al. "The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization". In: (2020). URL: http://arxiv.org/abs/2006.16241.

[16] Raphael Gontijo Lopes et al. "Improving Robustness Without Sacrificing Accuracy with Patch Gaussian Augmentation". en. In: *arXiv:1906.02611 [cs, stat]* (June 2019). arXiv: 1906.02611. URL: http://arxiv.org/abs/1906.02611.

[17] Dong Yin et al. "A Fourier Perspective on Model Robustness in Computer Vision". en. In: *arXiv:1906.08988 [cs, stat]* (Sept. 2020). arXiv: 1906.08988. URL: http://arxiv.org/abs/1906.08988.

[18] Jungkyu Lee et al. "Compounding the Performance Improvements of Assembled Techniques in a Convolutional Neural Network". en. In: *arXiv:2001.06268 [cs]* (Mar. 2020). arXiv: 2001.06268. URL: http://arxiv.org/abs/2001.06268.

[19] Cristina Vasconcelos et al. "An Effective Anti-Aliasing Approach for Residual Networks". en. In: *arXiv:2011.10675 [cs]* (Nov. 2020). arXiv: 2011.10675. URL: http://arxiv.org/abs/2011.10675.

[20] Robert Geirhos et al. "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness". In: *ICLR*. 2019, pp. 1–22. URL: http://arxiv.org/abs/1811.12231.

[21] Jan Laermann, Wojciech Samek, and Nils Strodthoff. "Achieving Generalizable Robustness of Deep Neural Networks by Stability Training". en. In: *arXiv:1906.00735 [cs, stat]* 11824 (2019). arXiv: 1906.00735, pp. 360–373. DOI: 10.1007/978-3-030-33676-9_25. URL: http://arxiv.org/abs/1906.00735.

[22] Gaurav Malhotra, Benjamin D. Evans, and Jeffrey S. Bowers. "Hiding a plane with a pixel: examining shape-bias in CNNs and the benefit of building in biological constraints". en. In: *Vision Research* 174 (Sept. 2020), pp. 57–68. ISSN: 00426989. DOI: 10.1016/j.visres.2020.04.013. URL: https://linkinghub.elsevier.com/retrieve/pii/S0042698920300742.

[23] Shahd Safarani et al. "Towards robust vision by multi-task learning on monkey visual cortex". en. In: *arXiv:2107.14344 [cs]* (July 2021). arXiv: 2107.14344. URL: http://arxiv.org/abs/2107.14344.

[24] Thomas G. Dietterich. "Ensemble Methods in Machine Learning". In: *Multiple Classifier Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 1–15. ISBN: 978-3-540-45014-6.

[25] L.K. Hansen and P. Salamon. "Neural network ensembles". en. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.10 (Oct. 1990), pp. 993–1001. ISSN: 01628828. DOI: 10.1109/34.58871. URL: http://ieeexplore.ieee.org/document/58871/.

[26] Anders Krogh and Jesper Vedelsby. "Neural Network Ensembles, Cross Validation, and Active Learning". en. In: (), p. 8.

[27] D. Opitz and R. Maclin. "Popular Ensemble Methods: An Empirical Study". en. In: *Journal of Artificial Intelligence Research* 11 (Aug. 1999), pp. 169–198. ISSN: 1076-9757. DOI: 10.1613/jair.614. URL: https://jair.org/index.php/jair/article/view/10239.

[28] Lior Rokach. "Ensemble-based classifiers". en. In: *Artificial Intelligence Review* 33.1-2 (Feb. 2010), pp. 1–39. ISSN: 0269-2821, 1573-7462. DOI: 10.1007/s10462-009-9124-7. URL: http://link.springer.com/10.1007/s10462-009-9124-7.

[29] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. "Deep Ensembles: A Loss Landscape Perspective". en. In: *arXiv:1912.02757 [cs, stat]* (June 2020). arXiv: 1912.02757. URL: http://arxiv.org/abs/1912.02757.

[30] Asa Cooper Stickland and Iain Murray. "Diverse Ensembles Improve Calibration". en. In: *arXiv:2007.04206 [cs, stat]* (July 2020). arXiv: 2007.04206. URL: http://arxiv.org/abs/2007.04206.

[31] Yaniv Ovadia et al. "Can You Trust Your Model's Uncertainty? Evaluating Predictive Uncertainty Under Dataset Shift". en. In: *arXiv:1906.02530 [cs, stat]* (Dec. 2019). arXiv: 1906.02530. URL: http://arxiv.org/abs/1906.02530.

[32] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles". en. In: *arXiv:1612.01474 [cs, stat]* (Nov. 2017). arXiv: 1612.01474. URL: http://arxiv.org/abs/1612.01474.

[33] Tianyu Pang et al. "Improving Adversarial Robustness via Promoting Ensemble Diversity". en. In: (), p. 10.

[34] Yeming Wen and Dustin Tran. "BATCHENSEMBLE: AN ALTERNATIVE APPROACH TO EFFICIENT ENSEMBLE AND LIFELONG LEARNING". en. In: (2020), p. 19.

[35] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. "Distilling the Knowledge in a Neural Network". en. In: *arXiv:1503.02531 [cs, stat]* (Mar. 2015). arXiv: 1503.02531. URL: http://arxiv.org/abs/1503.02531.

[36] Christian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. "Model Compression". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (2006), pp. 535–541.

[37] Zeyuan Allen-Zhu and Yuanzhi Li. "Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning". en. In: *arXiv:2012.09816 [cs, math, stat]* (July 2021). arXiv: 2012.09816. URL: http://arxiv.org/abs/2012.09816.

[38] Xu Lan, Xiatian Zhu, and Shaogang Gong. "Knowledge Distillation by On-the-Fly Native Ensemble". en. In: (), p. 11.

[39] Guobin Chen et al. "Learning Efficient Object Detection Models with Knowledge Distillation". en. In: (), p. 10.

[40] Antonio Greco et al. "Effective training of convolutional neural networks for age estimation based on knowledge distillation". en. In: *Neural Computing and Applications* (Apr. 2021). ISSN: 0941-0643, 1433-3058. DOI: 10.1007/s00521-021-05981-0. URL: https://link.springer.com/10.1007/s00521-021-05981-0.

[41] Jia Cui et al. "Knowledge distillation across ensembles of multilingual models for low-resource languages". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 4825–4829. DOI: 10.1109/ICASSP.2017.7953073.

[42] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. "Noise as a Resource for Learning in Knowledge Distillation". en. In: *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Waikoloa, HI, USA: IEEE, Jan. 2021, pp. 3128–3137. ISBN: 978-1-66540-477-8. DOI: 10.1109/WACV48630.2021.00317. URL: https://ieeexplore.ieee.org/document/9423222/.

[43] J. P. Jones and L. A. Palmer. "The two-dimensional spatial structure of simple receptive fields in cat striate cortex". In: *Journal of Neurophysiology* 58.6 (1987), pp. 1187–1211. ISSN: 00223077. DOI: 10.1152/jn.1987.58.6.1187.

[44] E.H. Adelson and J.R. Bergen. "Spatiotemporal energy models for the perception of motion". In: *Journal of the Optical Society of America A, Optics and image science* 2.2 (1985). ISBN: 0740-3232 (Print), pp. 284–299. ISSN: 1084-7529. DOI: 10.1364/JOSAA.2.000284. URL: http://www.opticsinfobase.org/abstract.cfm?URI=josaa-2-2-284.

[45] W. R. Softky and C. Koch. "The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs". In: *Journal of Neuroscience* 13.1 (1993), pp. 334–350. ISSN: 02706474. DOI: 10.1523/jneurosci.13-01-00334.1993.

[46] Florian Wenzel et al. "Hyperparameter Ensembles for Robustness and Uncertainty Quantification". en. In: *arXiv:2006.13570 [cs, stat]* (Jan. 2021). arXiv: 2006.13570. URL: http://arxiv.org/abs/2006.13570.

[47] Sheheryar Zaidi et al. "Neural Ensemble Search for Uncertainty Estimation and Dataset Shift". en. In: *arXiv:2006.08573 [cs, stat]* (June 2021). arXiv: 2006.08573. URL: http://arxiv.org/abs/2006.08573.

[48] Russell L. De Valois, Duane G. Albrecht, and Lisa G. Thorell. "Spatial Frequency Selectivity of Cells in Macaque Visual Cortex". In: *Vision Research* 22 (1982), pp. 545–559.

[49] Russell L. De Valois, E. W. Yund, and Norva Hepler. "The orientation and direction selectivity of cells in macaque visual cortex". In: *Vision Research* 22 (1982), pp. 531–544.

[50] Dario L Ringach, Robert M Shapley, and Michael J Hawken. "Orientation selectivity in macaque V1: diversity and laminar dependence." In: *The Journal of Neuroscience* 22.13 (2002). ISBN: 1529-2401 (Electronic)\n0270-6474 (Linking), pp. 5639–5651. ISSN: 1529-2401. DOI: 20026567. URL: http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?dbfrom=pubmed&id=12097515&retmode=ref&cmd=prlinks%5Cnpapers3://publication/uuid/EA0B2785-11B2-41CE-AF09-71647A4E026D.

[51] Dario L Ringach. "Spatial Structure and Symmetry of Simple-Cell Receptive Fields in Macaque Primary Visual Cortex". In: *Journal of Neurophysiology* 88 (2002), pp. 455–463.

[52] Yang Liu, Sheng Shen, and Mirella Lapata. "Noisy Self-Knowledge Distillation for Text Summarization". en. In: *arXiv:2009.07032 [cs]* (July 2021). arXiv: 2009.07032. URL: http://arxiv.org/abs/2009.07032.

[53] Nicole C Rust et al. "Spatiotemporal Elements of Macaque V1 Receptive Fields". In: *Neuron* 46 (2005), pp. 945–956. DOI: 10.1016/j.neuron.2005.05.021.

# Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The three main claims made at the end of the introduction and summarized in the abstract are properly supported by the results described in the corresponding section of the paper.
   (b) Did you describe the limitations of your work? [Yes] See Discussion where we describe that our models do not fully solve the problem of robustness and present future work directions that extend some preliminary results we report in the paper.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] We do not anticipate any potential negative societal impacts of our work. Instead, our work focuses on improving current computer vision models in making them better approximate human visual abilities.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We will release the model weights and instructions on how to reproduce the results.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section B.4
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Evaluation results are deterministic, except for stochastic models, and therefore do not change when running multiple times. For the ensemble and distillation models, the novel models presented here and on which our claims are based, we have only trained one seed due to computational constraints and therefore cannot report error bars. However, the gains of these models are substantial (two digits relative improvements) and consistent (over several corruption categories).
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section B.4
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] see Sections A.1, A.2, B.1, B.2
   (b) Did you mention the license of the assets? [Yes] see Sections A.1, A.2, B.1, B.2
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We will include the repository containing the adaptation of the VOneNet to Tiny ImageNet and the code for ensembling and distillation.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

# Supplementary Material

## A Datasets

### A.1 Tiny ImageNet

We used the Tiny ImageNet dataset for model training and evaluating model clean accuracy [13]. Tiny ImageNet contains 100.000 images of 200 classes (500 for each class) downsized to 64×64 colored images. Each class has 500 training images, 50 validation images and 50 test images. Tiny ImageNet is publicly available at `https://www.kaggle.com/c/tiny-imagenet`.

### A.2 Common Corruptions (Tiny ImageNet-C)

For evaluating model robustness to common corruptions we used the Tiny ImageNet-C dataset [10]. The Tiny ImageNet-C dataset consists of 15 different corruption types, each at 5 levels of severity for a total of 75 different perturbations, applied to validation images of Tiny ImageNet A.1. The individual corruption types are: Gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness, contrast, elastic transform, pixelate and JPEG compression (Fig. A.1). The individual corruption types are grouped into 4 categories: noise, blur, weather, and digital effects. The Tiny ImageNet-C is publicly available at `https://github.com/hendrycks/robustness` under Creative Commons Attribution 4.0 International.

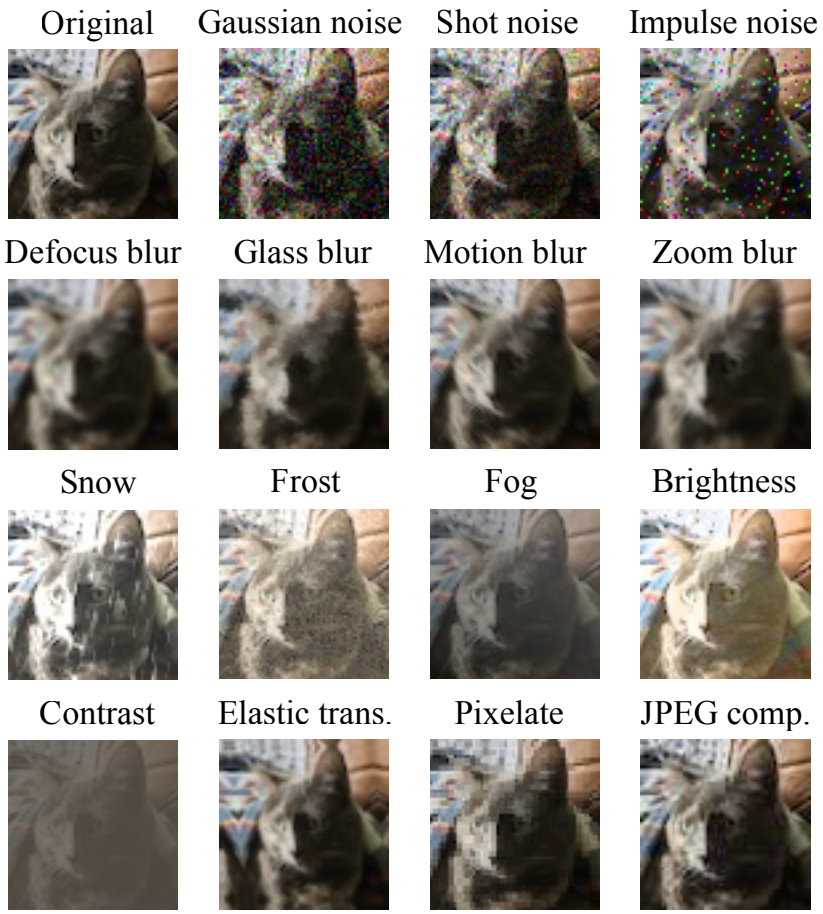


Figure A.1: **Common image corruptions at Tiny ImageNet resolution.** All 15 types of common image corruptions evaluated at severity = 3 for an image at the resolution of Tiny ImageNet (64px).

# B Models

## B.1 VOneNets

**VOneNet Model Family** VOneNets [12] are CNNs with a biologically-constrained fixed-weight front-end that simulates V1, the VOneBlock - a linear-nonlinear-Poisson (LNP) model of V1 [53], consisting of a fixed-weight Gabor filter bank (GFB) [43], simple and complex cell [44] nonlinearities, and neuronal stochasticity [45]. For the standard model, the GFB parameters are generated by randomly sampling from empirically observed distributions of preferred orientation, peak spatial frequency (SF), and size/shape of receptive fields [48, 49, 51], the channels are divided equally between simple- and complex-cells (256 each), and a Poisson-like stochasticity generator is used. Code for the VOneNet model family is publicly available at `https://github.com/dicarlolab/vonenet` under GNU General Public License v3.0.

**Adapting VOneNets to Tiny ImageNet** To facilitate model development and evaluation, we adapted the VOneNet architecture to the Tiny ImageNet image size and chose the ResNet18 architecture [4] for the back-end. Specifically, VOneResNet18 is built by replacing the first block (one stack of convolution, normalization, non-linearity and pooling layers) of ResNet18 [4] by the VOneBlock and a trainable bottleneck layer. Due to the difference in input size (from 224px for ImageNet to 64px in Tiny ImageNet), we made several modifications to the model architecture. First, we set the stride of the convolution layer (GFB) at two instead of four such that the output of the VOneBlock does not have a very small spatial map. We also adjusted the input field of view to 2deg for Tiny ImageNet instead of 8deg for ImageNet to account for the fact that images in the first represent a narrower portion of the visual space. This change resulted in an input resolution - number of pixels per degree (ppd) - of 32 ppd for Tiny ImageNet which is similar to that of ImageNet (28 ppd).

**VOneResNet18 Variants** We created seven VOneResNet18 model variants by removing or modifying one of the VOneBlock components (Table B.1). Three variants targeted the GFB: one with only low SF filters (Low SF; SF < 2cpd), one with only intermediate SF filters (Mid SF; 2cpd < SF < 5.6 cpd), and one with only high SF filters (High SF; SF > 5.6cpd). Two additional variants targeted the nonlinearities: one with only simple-cells (Only Simple) and one with only complex-cells (Only Complex). Finally, the last two variants targeted the stochasticity generator: one with a sub-Poisson like ($\sigma = \frac{\sqrt{\mu}}{2}$) stochasticity generator (Low Noise) and one without the stochasticity generator (No Noise).

| Variant Name | Spatial Frequency [cpd] | Cell Types [simple/complex ] | Stochasticity [type] |
|---|---|---|---|
| VOneResNet18 | 0.5 - 11.2 | 256/256 | Poisson |
| Low SF | 0.5 - 2.0 | 256/256 | Poisson |
| Mid SF | 2.0 - 5.6 | 256/256 | Poisson |
| High SF | 5.6 - 11.2 | 256/256 | Poisson |
| Only Simple | 0.5 - 11.2 | 512/0 | Poisson |
| Only Complex | 0.5 - 11.2 | 0/512 | Poisson |
| Low Noise | 0.5 - 11.2 | 256/256 | Sub-Poisson |
| No Noise | 0.5 - 11.2 | 256/256 | None |

Table B.1: **Parameters of VOneBlock Variants**.

## B.2 ResNet18

We used a variant of the Torchvision implementation of ResNet18 [4] as the base model and as the model back-end for the VOneResNet18. The original ResNet18 model, contains a combined stride of four in the first block (two in the convolution layer and two in the maxpool layer) which is the block replaced by VOneBlock in VOneResNet18. In order to maintain the size of the model comparable to the VOneResNet18, we adapted the ResNet18 architecture so that it has a stride of one in the first convolutional layer and kept the stride of two in the maxpool layer, resulting in a combined stride of two in the first block which is the same as the VOneBlock. We found that this variant of ResNet18 (58.93% accuracy) performed considerably better than the standard Torchvision implementation of

ResNet18 (50.45% accuracy) on clean Tiny ImageNet images after training from scratch following an identical training procedure (Section B.4).

### B.3 Ensembles

**Variants Ensemble** We created this ensemble by uniformly averaging the outputs (logits) of the eight VOneNet variants shown in Table B.1. The variants were trained individually prior to ensembling.

**Seeds Ensemble** We created this ensemble by combining eight different training seeds of the standard VOneResNet18 model using the same approach as with the Variants Ensemble. The different training seeds of the standard VOneResNet18 model were created by using different seeds to instantiate the GFB parameters of the VOneBlock and then training the back-ends.

**ResNet18 Seeds Ensemble** We created this ensemble by combining eight different training seeds of the ResNet18 model using the same approach as with the Variants Ensemble and the VOneResNet18 Seeds Ensemble.

### B.4 Training

PyTorch version 1.9.0 was used. All models were trained on Google Colab which provided access to 1 GPU (Nvidia K80, T4, P4 or P100) per session. The details of the training are described as follows.

**Preprocessing** During training, preprocessing included scaling the images using a scaling factor randomly sampled between 1-1.2, rotating the images using a rotation angle randomly sampled between -30 to 30 degrees, shifting the images in the horizontal direction by a pixel distance randomly sampled between -5% to 5% of the image width, shifting the images in the vertical direction by a pixel distance randomly sampled between -5% to 5% of the image height, and flipping the images horizontally with a random probability of 0.5. Images were normalized by subtraction and division by [0.5, 0.5, 0.5]. For models trained with GNT [11], preprocessing involved an additional step of adding Gaussian noise (standard deviation of 0.6) to 50% of all images. During evaluation, preprocessing only involved image normalization, i.e. subtraction and division by [0.5, 0.5, 0.5].

**Loss Functions** For models trained without knowledge distillation, the loss function was given by a cross-entropy loss between image labels and model predictions (logits). For models trained with knowledge distillation, the loss function was given by a weighted average of two loss functions [35]. The first loss function with a weight 100 minimized the cross-entropy between the output probability distributions (soft targets) of the distilled and the emsemble model. The soft targets for both those models were computed using temperature 5. The second loss function with a weight 5 minimized the cross-entropy between image labels and the distilled model predictions (logits).

**Optimization** For optimization, we used Stochastic Gradient Descent with momentum 0.9, a weight decay 0.0005, and an initial learning rate 0.1. The learning rate was dynamically adjusted by dividing it by 10 whenever there is no improvement in validation loss for 5 consecutive epochs. All models were trained using a batch size of 128 images for 60 epochs.

# C Detailed Results

| | Noise | | | Blur | | | |
|---|---|---|---|---|---|---|---|
| Model | Gaussian [%] | Shot [%] | Impulse [%] | Defocus [%] | Glass [%] | Motion [%] | Zoom [%] |
| ResNet18 | 19.8 | 23.2 | 21.9 | 14.5 | 20.0 | 20.5 | 16.6 |
| VOneResNet18 | 28.7 | 32.7 | 26.5 | 16.9 | 19.8 | 22.3 | 18.8 |
| Low Noise | 27.1 | 31.5 | 25.6 | 16.9 | 20.3 | 22.4 | 18.6 |
| No Noise | 22.9 | 27.6 | 22.1 | 15.4 | 19.0 | 21.2 | 17.1 |
| Low SF | 31.2 | 32.8 | **29.9** | **33.4** | **32.5** | **33.8** | **34.3** |
| Mid SF | 31.1 | **35.1** | 28.1 | 29.2 | 28.7 | **33.8** | 33.7 |
| High SF | 24.9 | 29.0 | 24.8 | 13.9 | 18.0 | 18.8 | 15.5 |
| Only Simple | **31.9** | 34.9 | 29.1 | 19.3 | 21.2 | 24.5 | 21.4 |
| Only Complex | 23.7 | 27.0 | 22.3 | 12.8 | 15.7 | 16.3 | 14.1 |

| | Weather | | | | Digital | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Snow [%] | Frost [%] | Fog [%] | Bright. [%] | Contrast [%] | Elastic [%] | Pixelate [%] | JPEG [%] |
| ResNet18 | 24.7 | 26.0 | **22.8** | 28.6 | **10.5** | 39.1 | 33.6 | 25.6 |
| VOneResNet18 | 26.0 | 25.3 | 17.7 | 26.9 | 6.2 | 34.3 | 37.1 | 28.7 |
| Low Noise | 27.0 | 27.1 | 20.1 | 29.0 | 7.7 | 36.1 | 38.5 | 29.1 |
| No Noise | **27.7** | **28.0** | 22.4 | 28.8 | 9.2 | 36.1 | 37.3 | 27.4 |
| Low SF | 19.9 | 19.9 | 13.9 | 22.4 | 4.7 | 34.4 | 33.8 | 33.3 |
| Mid SF | 26.5 | 27.5 | 19.5 | **29.2** | 7.1 | **40.4** | **41.7** | **38.9** |
| High SF | 24.9 | 24.8 | 18.1 | 26.8 | 6.7 | 34.5 | 35.1 | 25.5 |
| Only Simple | 26.4 | 27.5 | 18.8 | 28.5 | 6.8 | 35.5 | 38.7 | 30.9 |
| Only Complex | 16.8 | 16.1 | 13.8 | 20.5 | 4.6 | 29.3 | 31.8 | 22.8 |

Table C.1: **Absolute accuracies of ResNet18, standard VOneResNet18 and the seven additional variants on the 15 types of common image corruptions (averaged over perturbation severities)**.

| | Noise | | | Blur | | | |
|---|---|---|---|---|---|---|---|
| Model | Gaussian [%] | Shot [%] | Impulse [%] | Defocus [%] | Glass [%] | Motion [%] | Zoom [%] |
| ResNet18 | 19.8 | 23.2 | 21.9 | 14.5 | 20.0 | 20.5 | 16.6 |
| VOneResNet18 | 28.7 | 32.7 | 26.5 | 16.9 | 19.8 | 22.3 | 18.8 |
| Variants Ensemble | **33.8** | **38.3** | **31.7** | **24.3** | **26.5** | **30.3** | **26.9** |
| Seeds Ensemble | 32.9 | 37.0 | 30.7 | 19.5 | 22.8 | 24.7 | 21.8 |
| ResNet18 Seeds Ensemble | 21.6 | 25.4 | 24.6 | 16.2 | 21.5 | 22.3 | 18.3 |

| | Weather | | | | Digital | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Snow [%] | Frost [%] | Fog [%] | Bright. [%] | Contrast [%] | Elastic [%] | Pixelate [%] | JPEG [%] |
| ResNet18 | 24.7 | 26.0 | 22.8 | 28.6 | 10.5 | 39.1 | 33.6 | 25.6 |
| VOneResNet18 | 26.0 | 25.3 | 17.7 | 26.9 | 6.2 | 34.3 | 37.1 | 28.7 |
| Variants Ensemble | **31.8** | **31.7** | **22.9** | **33.4** | 8.3 | **42.9** | **44.8** | **37.0** |
| Seeds Ensemble | 30.0 | 29.0 | 20.0 | 30.6 | 7.0 | 38.8 | 41.6 | 32.5 |
| ResNet18 Seeds Ensemble | 27.5 | 28.7 | 24.6 | 31.2 | **11.1** | 42.4 | 36.4 | 28.2 |

Table C.2: **Absolute accuracies of ResNet18, standard VOneResNet18, and the three ensemble models: Variants Ensemble, Seeds Ensemble and ResNet18 Seeds Ensemble (averaged over perturbation severities)**.

| | Noise | | | Blur | | | |
|---|---|---|---|---|---|---|---|
| Model | Gaussian [%] | Shot [%] | Impulse [%] | Defocus [%] | Glass [%] | Motion [%] | Zoom [%] |
| ResNet18 | 19.8 | 23.2 | 21.9 | 14.5 | 20.0 | 20.5 | 16.6 |
| VOneResNet18 | 28.7 | 32.7 | 26.5 | 16.9 | 19.8 | 22.3 | 18.8 |
| VOneResNet18-Distilled | **29.1** | **32.9** | **27.0** | **17.4** | 20.4 | 22.8 | **19.3** |
| No Noise | 22.9 | 27.6 | 22.1 | 15.4 | 19.0 | 21.2 | 17.1 |
| No Noise-Distilled | 25.0 | 30.0 | 24.1 | 16.2 | **20.6** | **23.0** | 18.1 |
| ResNet18-Distilled | 20.2 | 23.8 | 22.5 | 15.5 | 19.2 | 20.8 | 17.5 |

| | Weather | | | | Digital | | | |
|---|---|---|---|---|---|---|---|---|
| Model | Snow [%] | Frost [%] | Fog [%] | Bright. [%] | Contrast [%] | Elastic [%] | Pixelate [%] | JPEG [%] |
| ResNet18 | 24.7 | 26.0 | 22.8 | 28.6 | **10.5** | 39.1 | 33.6 | 25.6 |
| VOneResNet18 | 26.0 | 25.3 | 17.7 | 26.9 | 6.2 | 34.3 | 37.1 | 28.7 |
| VOneResNet18-Distilled | 26.4 | 25.8 | 18.0 | 27.0 | 6.5 | 34.9 | 37.5 | 29.3 |
| No Noise | 27.7 | 28.0 | 22.4 | 28.8 | 9.2 | 36.1 | 37.3 | 27.4 |
| No Noise-Distilled | **30.1** | **30.3** | **24.2** | **32.0** | **10.5** | **39.4** | **40.6** | **30.2** |
| ResNet18-Distilled | 24.9 | 25.7 | 22.1 | 27.8 | 9.9 | 38.8 | 33.2 | 26.0 |

Table C.3: **Absolute accuracies of ResNet18, VOneResNet18, No Noise variant, and the three distillation models: VOneResNet18-Distilled, No Noise-Distilled and ResNet18-Distilled (averaged over perturbation severities)**.

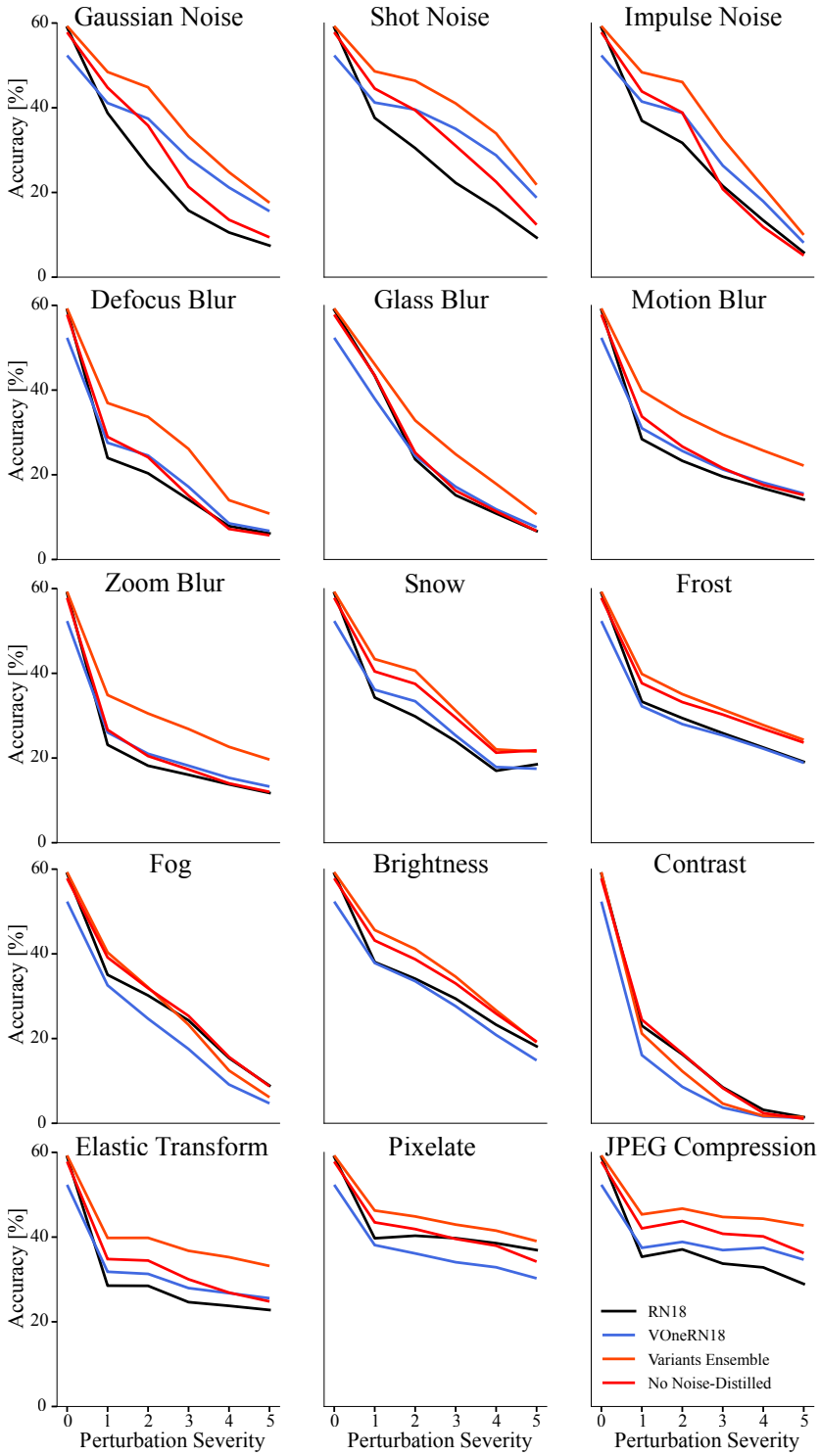


Figure C.1: **Variants Ensemble and distilled VOneResNet18 No Noise show consistent improvements in robustness** Absolute accuracies for ResNet18, VOneResNet18, Variants Ensemble, and distilled VOneResNet18 without stochasticity for the 15 corruption types at all perturbation severity levels. Accuracy represents top-1. Perturbation 0 corresponds to clean images.