PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Increasing neural network robustness improves match to macaque V1 eigenspectrum, spatial frequency preference and predictivity

Nathan C. L. Kong[1,3]*, **Eshed Margalit**[2,3], **Justin L. Gardner**[1,3], **Anthony M. Norcia**[1,3]

1 Department of Psychology, Stanford University, Stanford, California, United States of America, **2** Neurosciences Graduate Program, Stanford University, Stanford, California, United States of America, **3** Wu Tsai Neurosciences Institute, Stanford University, Stanford, California, United States of America

* nclkong@stanford.edu

Citation: Kong NCL, Margalit E, Gardner JL, Norcia AM (2022) Increasing neural network robustness improves match to macaque V1 eigenspectrum, spatial frequency preference and predictivity. PLoS Comput Biol 18(1): e1009739. https://doi.org/10.1371/journal.pcbi.1009739

Editor: Peter E. Latham, UCL, UNITED KINGDOM

Received: July 20, 2021

Accepted: January 4, 2022

Published: January 7, 2022

Data Availability Statement: There are no primary data in the paper; all code are available at https://github.com/nathankong/robustness_primary_visual_cortex.

Funding: N.C.L.K. is supported by the Stanford University Ric Weiland Graduate Fellowship. E.M. is supported by a National Science Foundation Graduate Research Fellowship. J.L.G. acknowledges the generous support of Research to Prevent Blindness and Lions Club International Foundation (https://www.rpbusa.org/rpb/low-vision/). A.M.N. is supported by the Stanford

Abstract

Task-optimized convolutional neural networks (CNNs) show striking similarities to the ventral visual stream. However, human-imperceptible image perturbations can cause a CNN to make incorrect predictions. Here we provide insight into this brittleness by investigating the representations of models that are either robust or not robust to image perturbations. Theory suggests that the robustness of a system to these perturbations could be related to the power law exponent of the eigenspectrum of its set of neural responses, where power law exponents closer to and larger than one would indicate a system that is less susceptible to input perturbations. We show that neural responses in mouse and macaque primary visual cortex (V1) obey the predictions of this theory, where their eigenspectra have power law exponents of at least one. We also find that the eigenspectra of model representations decay slowly relative to those observed in neurophysiology and that robust models have eigenspectra that decay slightly faster and have higher power law exponents than those of non-robust models. The slow decay of the eigenspectra suggests that substantial variance in the model responses is related to the encoding of fine stimulus features. We therefore investigated the spatial frequency tuning of artificial neurons and found that a large proportion of them preferred high spatial frequencies and that robust models had preferred spatial frequency distributions more aligned with the measured spatial frequency distribution of macaque V1 cells. Furthermore, robust models were quantitatively better models of V1 than non-robust models. Our results are consistent with other findings that there is a misalignment between human and machine perception. They also suggest that it may be useful to penalize slow-decaying eigenspectra or to bias models to extract features of lower spatial frequencies during task-optimization in order to improve robustness and V1 neural response predictivity.

Institute for Human Centered Artificial Intelligence. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Author summary

Convolutional neural networks (CNNs) are the most quantitatively accurate models of multiple visual areas. In contrast to humans, however, their image classification behaviour can be modified drastically by human-imperceptible image perturbations. To provide insight as to why CNNs are so brittle, we investigated the image features extracted by models that are robust or not robust to these image perturbations. We found that non-robust CNNs had a preference for high spatial frequency image features, unlike primary visual cortex (V1) cells. Models that were more robust to image perturbations had a preference for image features more aligned with those extracted by V1 and also improved predictions of neural responses in V1. This suggests that the dependence on high-frequency image features for image classification may be related to the image perturbations affecting models but not humans. Our work is consistent with other findings that CNNs may be relying on image features not aligned with those used by humans for image classification and suggests possible optimization targets to improve the robustness of and the V1 correspondence of CNNs.

Introduction

Our visual system has the seemingly effortless ability to extract relevant features from the environment to support behaviour. Computational models known as convolutional neural networks (CNNs) incorporate principles of the neurobiology of the visual system and have allowed us to mimic some capabilities of our visual system [1]. These models have had immense success in artificial intelligence and can be trained to perform at or above human capabilities on many tasks in the visual domain such as object categorization and semantic segmentation [2–6]. These capabilities have led to many comparisons between the internal representations of CNNs and those of the human and non-human primate ventral visual stream, showing that task-optimized CNNs are also quantitatively accurate models of visual processing [7–14].

Although these models show remarkable similarities to the primate ventral visual stream, they diverge significantly from humans in their classification behaviour on images that have been modified by human-imperceptible, non-random image perturbations. In particular, these *adversarial perturbations* can cause the model to completely mis-classify the image even though it could correctly classify the unperturbed image, resulting in poor *adversarial robustness* [15]. This is clearly an issue in safety-critical applications (e.g., self-driving cars), so the machine learning community has been developing techniques to train these models to be more robust to adversarial perturbations [16–21]. Robust optimization techniques have been shown to be able to defend against very strong adversarial attacks, although there still exist perturbations that can fool models trained with these techniques.

The striking misalignment between machine and human image classification on adversarially perturbed images suggests that humans are using image features for the task that are different from those used by models explicitly optimized to perform the task [22–25]. This leads to the following question: what are some properties of the internal representations that differ between primate and machine vision and that result in such brittleness?

Motivated to understand the dimensionality of the population code, Stringer et al. [26] developed a theory connecting the eigenspectrum of a system's neural responses to the system's vulnerability to small stimulus perturbations. In mouse primary visual cortex (V1), Stringer et al. [26] showed that the eigenspectrum of the neural responses to natural scenes

decays according to a power law with exponent $\alpha \approx 1$. Their theory predicted this power-law-like behaviour and states that if $\alpha < 1$ for neural responses to natural scenes, then the neural code is "pathological" in the sense that small perturbations in the stimulus could result in unbounded changes in the neural responses. Too much variation in the responses with respect to the stimulus would allow minute stimulus changes to drastically affect the neural responses. As the existence of adversarial examples in CNNs is, by definition, vulnerability of CNNs to small input perturbations, we investigated the eigenspectra of the representations of models that are either robust or not robust to these perturbations.

The eigenspectrum can also provide insight into the image features extracted by a system. Lower principal components are associated with neural response variance related to coarser stimulus features and higher principal components are associated with variance related to finer stimulus features (see Extended Data Fig 6 in Stringer et al. [26]). Thus, if one system's eigenspectrum decays slower (i.e., has a smaller power law exponent) than that of another system, it means that a larger amount of neural response variance is dedicated to the encoding of fine stimulus features in the first system than that of the second system. We therefore hypothesized that model responses with eigenspectra of small power law exponents have many artificial neurons tuned to image features of high spatial frequencies.

To test the hypothesis that many artificial neurons are tuned to high spatial frequencies, we investigated the preferred spatial frequency tuning distributions of these models and compared these distributions to that of cells in the foveal area of macaque V1. This resulted in three main contributions. Firstly, we found that models with higher adversarial robustness have internal representations whose eigenspectra decay slightly faster than those of their non-robust counterparts and is consistent with the theory of Stringer et al. [26]. Secondly, by performing in-silico electrophysiology experiments, we found that non-robust models had a large proportion of neurons tuned to high spatial frequencies. Moreover, the similarity between a model's preferred spatial frequency distribution and that of cells in the foveal area of macaque V1 was higher for robust models than that of non-robust models. Robust models, however, still had many artificial neurons preferring high spatial frequencies (though less than that of non-robust models). Finally, we found that robust models are better models of V1 than non-robust models in terms of their neural response predictivity.

Altogether, although CNNs are some of the best models of the ventral visual stream in terms of neural response predictions, there are still many differences between human and machine perception that need to be improved upon to gain a deeper understanding of our visual system. The results suggest that one way in which our visual system is robust to minute image perturbations is by ignoring (i.e., not encoding) high spatial frequency information in the inputs. They also suggest that explicitly reducing the dimensionality of internal representations (e.g., by penalizing the eigenspectrum so that it decays faster) and reducing the preferred spatial frequency of artificial neurons during task-optimization may improve a model's adversarial robustness and that this may also lead to better models of V1.

Results

We performed in-silico electrophysiology experiments and linearly mapped model neurons to macaque V1 neurons. A schematic of the analyses performed in this work is provided in Fig 1. Model layer activations were recorded in response to a set of natural images and a set of Gabor patches in order to obtain the model layer's eigenspectrum and preferred spatial frequency distribution (Fig 1A and 1B). Using a previously collected set of macaque V1 neural responses [13], we linearly mapped model neurons to V1 neurons (Fig 1C).

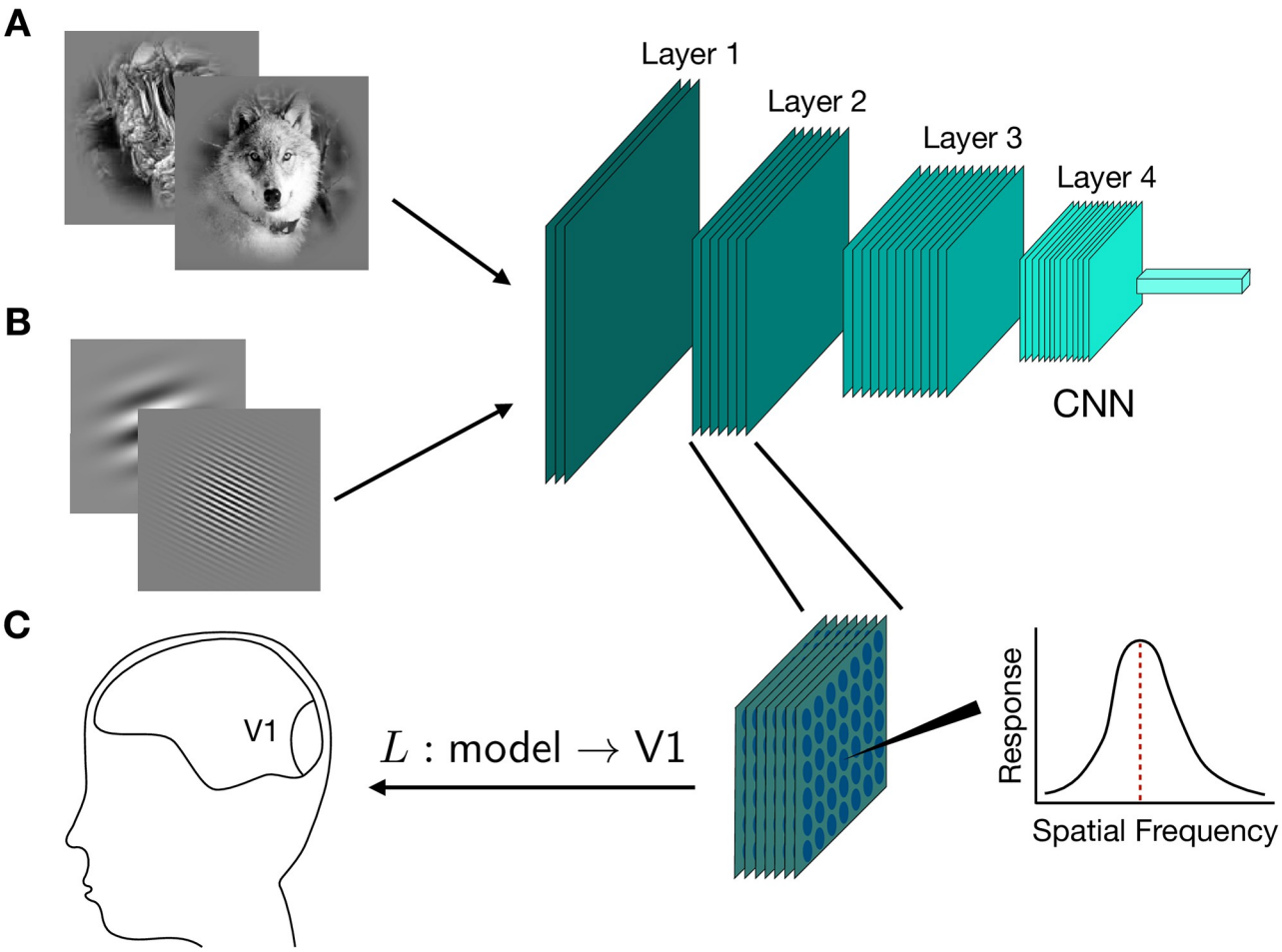

Fig 1. A schematic of the model analyses. A. Images from the stimulus set of Cadena et al. [13] were used to extract model responses for neural response prediction. In addition, a random set of natural images from the ImageNet database [27] was used to obtain a model's responses at each layer, from which the eigenspectrum and the power law exponent were computed. **B**. In-silico electrophysiology experiments were performed by presenting models with a set of Gabor patches that varied in spatial frequency, orientation and phase. A spatial frequency tuning curve was then computed using a single artificial neuron's responses and its preferred spatial frequency is the frequency at which the tuning curve reaches its maximum value. By performing this analysis for each convolutional filter, we computed the distribution of preferred spatial frequencies for a model layer. **C**. A model layer's responses were linearly mapped (denoted as L) to macaque V1 neural responses using partial least squares regression and the linear map's performance was defined to be the noise-corrected Pearson's correlation between the model predictions and the observed neural responses.

https://doi.org/10.1371/journal.pcbi.1009739.g001

Eigenspectrum of macaque V1 neural responses also follows a power law with exponent at least one

It was observed by Stringer et al. [26] that the eigenspectrum of mouse V1 neural responses to natural scenes decays according to a power law with exponent close to one. If the power-law-like behaviour of the eigenspectrum is a strong biological constraint, then we would expect that it would generalize across species (e.g., macaques).

We therefore computed the eigenspectrum using cross-validated principal components analysis (cvPCA, [26]) of a previously collected set of macaque V1 neural responses to natural scenes [13]. As shown in Fig 2, we found that the eigenspectrum of macaque V1 neural responses follows a power law with exponent close to one, as in mouse V1 [26]. As shown in S5 Fig, the power-law-like behaviour of the eigenspectrum can also be observed in subsets of the macaque neural dataset, where only a fraction of the stimuli was used. We note, however,

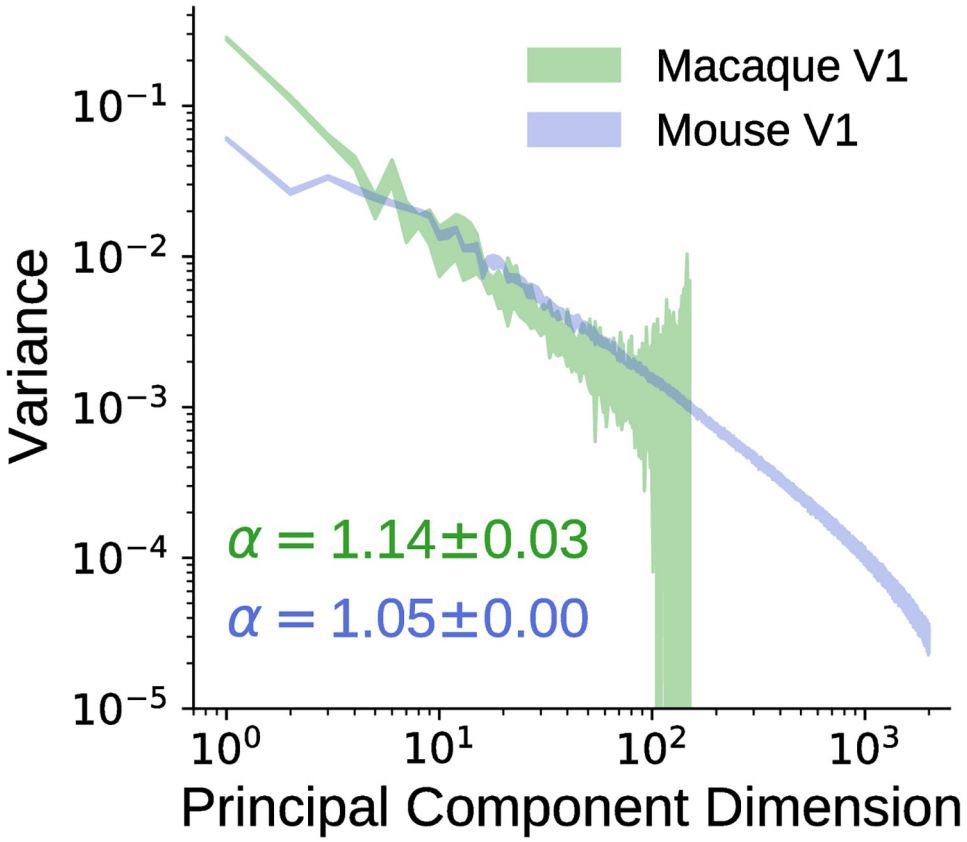

Fig 2. Eigenspectra of macaque and mouse V1 neural responses to natural scenes. Using a previously collected set of macaque V1 neural responses to natural scenes, we computed its eigenspectrum using cross-validated principal components analysis (cvPCA) and found that it obeyed a power law and had a power law exponent close to one, similar to the observation in mouse V1 [26]. Shaded regions denote the standard deviation of the variance explained by each principal component across the 20 runs of cvPCA. The inset indicates the mean and standard deviation of the power law exponent, α, of each eigenspectrum across the 20 runs of cvPCA. Note that the standard deviation is not symmetrical (about the mean) due to the log-scale of the vertical axis.

https://doi.org/10.1371/journal.pcbi.1009739.g002

that the convergence of the power law exponent to one may not be complete in this dataset due to the limited number of neurons (166 neurons in total, as compared to the approximately 10 000 neurons in the mouse V1 neural response dataset). Recordings with more neurons in macaque V1 are needed to further verify this power-law-like phenomenon. Nonetheless, our observations suggest that the power-law-like behaviour of the neural response eigenspectrum does indeed generalize across species.

Eigenspectra of robust models decay slightly faster than those of non-robust models, but slower relative to those observed in neurophysiology

The theory of Stringer et al. [26] predicts that if the eigenspectrum of a system's neural responses to natural scenes decays with power law exponent less than one (i.e., $\alpha < 1$), then the system will be affected by small perturbations to the inputs, suggesting that this phenomenon is related to the existence of adversarial examples, where human-imperceptible perturbations to an image can cause a model to mis-classify the image even though the unperturbed image could be classified correctly [15, 26]. This implies that the internal representations of non-robust CNNs are greatly affected by "small" image perturbations. Motivated by this

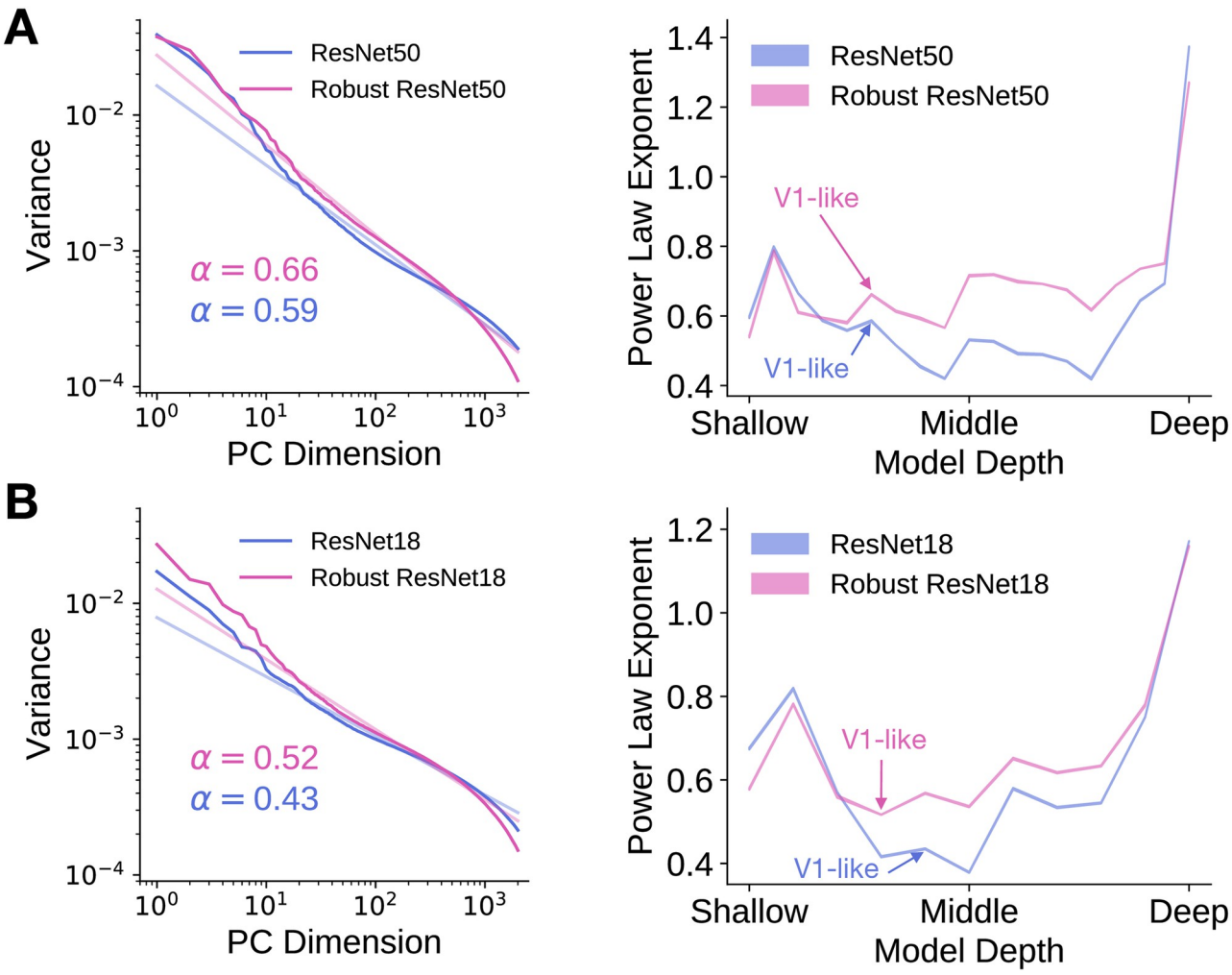

Fig 3. The internal representations of robust models is slightly lower dimensional than those of non-robust models. The eigenspectrum of the robust model decays slightly faster than that of the non-robust model. The eigenspectrum of the artificial neural responses in each layer of a model to a random subset of 2816 ImageNet validation set images was computed using principal components analysis and the power law exponent was computed by obtaining the slope of the line of best fit to the variances within the principal component range of 10 and 999. Changing the range of variances used to compute the power law exponents to those between 1 and 1000 does not alter the relationship between the power law exponents of the robust models and those of the non-robust models (S6 Fig). **A**. Left: We plot the eigenspectrum for the most "V1-like" layer (as determined by neural predictivity) of a robust and a non-robust ResNet-50. Light lines are the lines of best fit to the eigenspectrum (in dark colour) in log-log space between principal component dimensions 10 and 999. The inset indicates the estimated power law exponent of the eigenspectrum. Right: Power law exponents of model layers for ResNet-50. Shaded regions (too small to be easily visible) indicate standard deviation across random ImageNet validation set subsets. **B**. As in **A**, but for ResNet-18.

https://doi.org/10.1371/journal.pcbi.1009739.g003

theory, we asked whether or not the power law exponents of the eigenspectra of the representations of robust models were higher and closer to one than those of non-robust models.

Here and in the subsequent two sections, we focus on two model architectures—ResNet-50 and ResNet-18 [6]. Both of these task-optimized architectures have been shown to achieve good neural predictivity [11], good task performance [6] and are architectures that have been previously trained with and without robustness penalties. The robust ResNet-50 model was adversarially trained to be robust to perturbations, $\boldsymbol{\delta}$, of ℓ_2-norm at most three (i.e., $\|\boldsymbol{\delta}\|_2 \leq 3$) and the robust ResNet-18 model was adversarially trained to be robust to perturbations of ℓ_∞-norm at most 0.5/255 (i.e., $\|\boldsymbol{\delta}\|_\infty \leq 0.5/255$) [18, 28]. For a particular layer in each model,

we computed the eigenspectrum of the artificial neural responses to random sets of approximately 3000 natural images from the ImageNet validation set images [27]. As shown on the left of Fig 3A and 3B, we found that the eigenspectrum of a particular layer of the robust model has a larger power law exponent than that of its non-robust counterpart. In fact, across model layers, we found that the power law exponents of the robust models were higher than those of their non-robust counterparts, as shown on the right of Fig 3A and 3B. This relationship between the power law exponents of robust and non-robust models was consistent also for different image resolutions (S7 Fig). These findings are consistent with the theory of Stringer et al. [26], as the models with higher power law exponents are slightly more robust to small image perturbations. We note, however, that the power law exponents of robust models are much less than one and still mismatch with that of V1, consistent with the fact that there still exist image perturbations that can fool these robust models.

Preferred spatial frequency distributions of robust models are more similar to V1 than those of non-robust models

As described above, we observed that the eigenspectra of the internal representations of two robust models decayed slightly faster than those of their non-robust counterparts. Since the eigenspectra for the robust models decay slightly faster than those of the non-robust model, the representations learned by the robust models are lower-dimensional than those learned by the non-robust models. One possible reason for the lower-dimensionality is that "fine stimulus features" may not be contributing to the artificial neural response variance for the large principal component dimensions of robust models as much as they do for the non-robust models.

We therefore hypothesized that non-robust models extract image features that are of higher spatial frequencies than those extracted by robust models. This would mean that its "V1-like" layer (defined as the model layer that has the highest V1 neural response predictivity) consists of a large proportion of artificial neurons tuned to mid to high spatial frequencies. We tested this hypothesis by performing in-silico electrophysiology experiments to estimate the spatial frequency tuning of artificial neurons in each model. We first generated fixed-size Gabor patches of ten orientations, ten spatial frequencies and ten phases, of which a few examples are shown in Fig 4A. The Gabor patches were then presented to the models and artificial neural responses were obtained from the most "V1-like" model layer. Using these responses, we computed spatial frequency tuning curves, of which a few examples are shown in Fig 4B. The preferred spatial frequency of an artificial neuron was then defined to be the spatial frequency at which the tuning curve achieves its maximum value.

To gain more intuition into the image features that are extracted by artificial neurons that prefer various spatial frequencies, we generated stimuli that maximally excite channels of the most "V1-like" model layer. We show spatial frequency tuning curves of example artificial neurons and their associated optimal stimuli in Fig 4B. As can be seen, the optimal stimulus for artificial neurons that respond maximally to high spatial frequencies contain high frequency image features. On the opposite end of the spectrum, the optimal stimulus for artificial neurons that respond maximally to low spatial frequencies contain relatively low spatial frequency content.

We constructed the preferred spatial frequency distribution of a model by aggregating the preferred spatial frequencies across the artificial neurons and found that robust models had distributions more similar to that of cells in the foveal area of macaque V1 than those of non-robust models. From the distributions shown in Fig 4C, we observed that both robust and non-robust models have many artificial neurons that are tuned to the highest spatial frequency bin (we assumed the field of view of these models was 6.4 degrees of visual angle, as in the

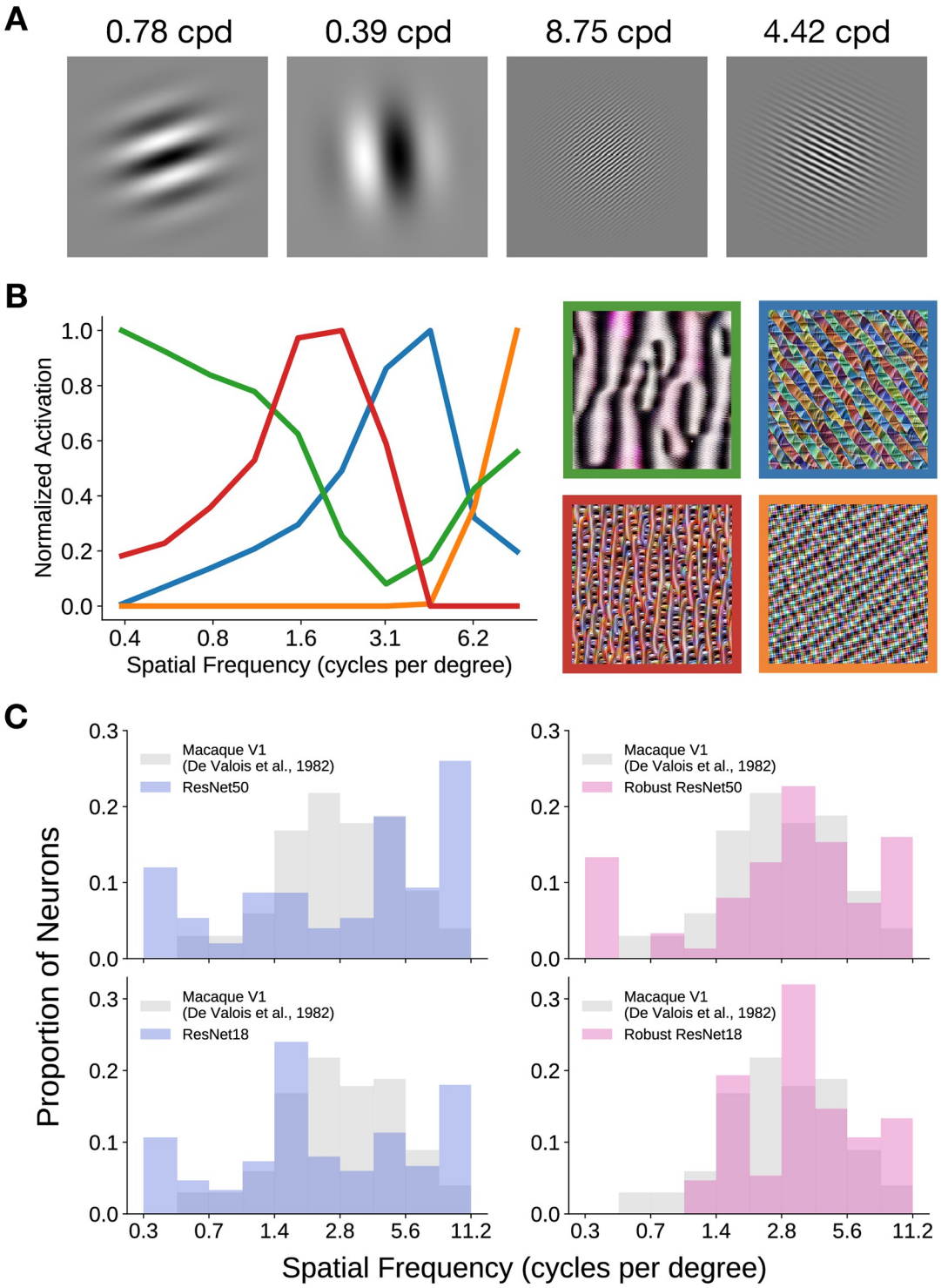

Fig 4. Assessing the spatial frequency tuning of models. We performed in-silico electrophysiology experiments to extract spatial frequency tuning curves of representative artificial neurons from the most "V1-like" layer of a model, as determined by neural predictivity. The field of view of these models was assumed to be 6.4˚ per image. **A**. A few example Gabor patch stimuli of different orientations, phases and frequencies, that were used in the in-silico electrophysiology experiments. cpd: cycles per degree. **B**. A few example spatial frequency tuning curves are plotted, each corresponding to an artificial neuron (i.e., convolutional filter). The images are the stimuli that optimally excite the corresponding channel in the output of the most "V1-like" ResNet-50 model layer. **C**. Preferred spatial frequency distributions for robust and non-robust models from one in-silico experiment. We can see that the distributions of the robust models are more similar to that of macaque V1 cells than the distributions of non-robust models. Top: Robust and non-robust ResNet-50. Bottom: Robust and non-robust ResNet-18.

https://doi.org/10.1371/journal.pcbi.1009739.g004

work of Cadena et al. [13]; 56 cycles per image ≈ 56 cycles / 6.4 degrees = 8.75 cycles per degree). Comparing these distributions with that of cells in the foveal area of macaque V1 (cf. Fig 6 in De Valois et al. [29]), we note that there are a relatively small number of cells in macaque V1 that are tuned to high spatial frequencies (coloured in gray in Fig 4C), suggesting that this is one way (of many) in which current task-optimized CNN models of the ventral visual stream deviate from the neurophysiology.

Although both robust and non-robust models have preferred spatial frequency distributions that are quite unlike that of macaque V1 foveal cells, the distributions of robust models were more similar to that of macaque V1 than the distributions of non-robust models (Fig 4C). To quantify this similarity, we used a metric based on the maximum absolute difference between the two cumulative distributions (see Methods), where smaller scores indicate that the two distributions are dissimilar and larger scores indicate that the two distributions are similar. For the ResNet-50 architecture, the non-robust model had a score of 0.763 ± 0.033, whereas the robust model had a score of 0.817 ± 0.032. For the ResNet-18 architecture, the non-robust model had a score of 0.771 ± 0.039, whereas the robust model had a score of 0.790 ± 0.036. The error in all cases denotes the standard deviation across 1000 in-silico electrophysiology experiments.

Robust models better predict macaque V1 neural responses than non-robust models

We observed that the power law exponent and the preferred spatial frequency distribution of robust models are closer to those of macaque V1, suggesting that robust models better predict V1 neural responses than non-robust models, which we found to be the case. For each model, we assumed its field of view was 6.4 degrees of visual angle, as in prior work [13]. For each model layer, we performed a partial least squares regression to find a linear mapping between the model features and the macaque V1 neural responses, consistent with the procedure described in prior work [8, 11, 30, 31]. The goodness-of-fit of the linear mapping was defined as the correlation between the predicted and the observed neural responses, noise-corrected by the Spearman-Brown corrected cross-trial correlation (i.e., internal consistency) of each neuron. As expected, the feature space provided by the model layers between the shallow and middle portions of the models best corresponded to macaque V1 neural responses, consistent with prior work [13, 14]. Furthermore, as shown in Fig 5, we found that robust models (adversarially trained with a particular perturbation type and size) provided feature spaces that better correspond to macaque V1 neural responses than those of non-robust models ($p < 0.01$ for both architectures). Improved correspondence to macaque V1 neural responses held not only for one particular perturbation size used for adversarial training, but also for a wide range of perturbation sizes (S3 Fig).

Adversarial robustness is correlated to V1 predictivity and is not correlated to power law exponent

We observed that two instances of the CNN class of models (ResNet-18 and ResNet-50) that were adversarially trained better corresponded to macaque V1 neural responses than their non-robust counterparts. In addition, the robust models had larger power law exponents across model layers. We next asked whether these observations extended across a wider range of CNN architectures. We therefore performed a large-scale benchmarking of 40 models to ascertain whether or not there was a relationship between a model's robustness, defined by its accuracy on adversarially perturbed images from the ImageNet validation set [27], and its V1 neural response predictivity. Each model's maximum neural predictivity is presented in S1

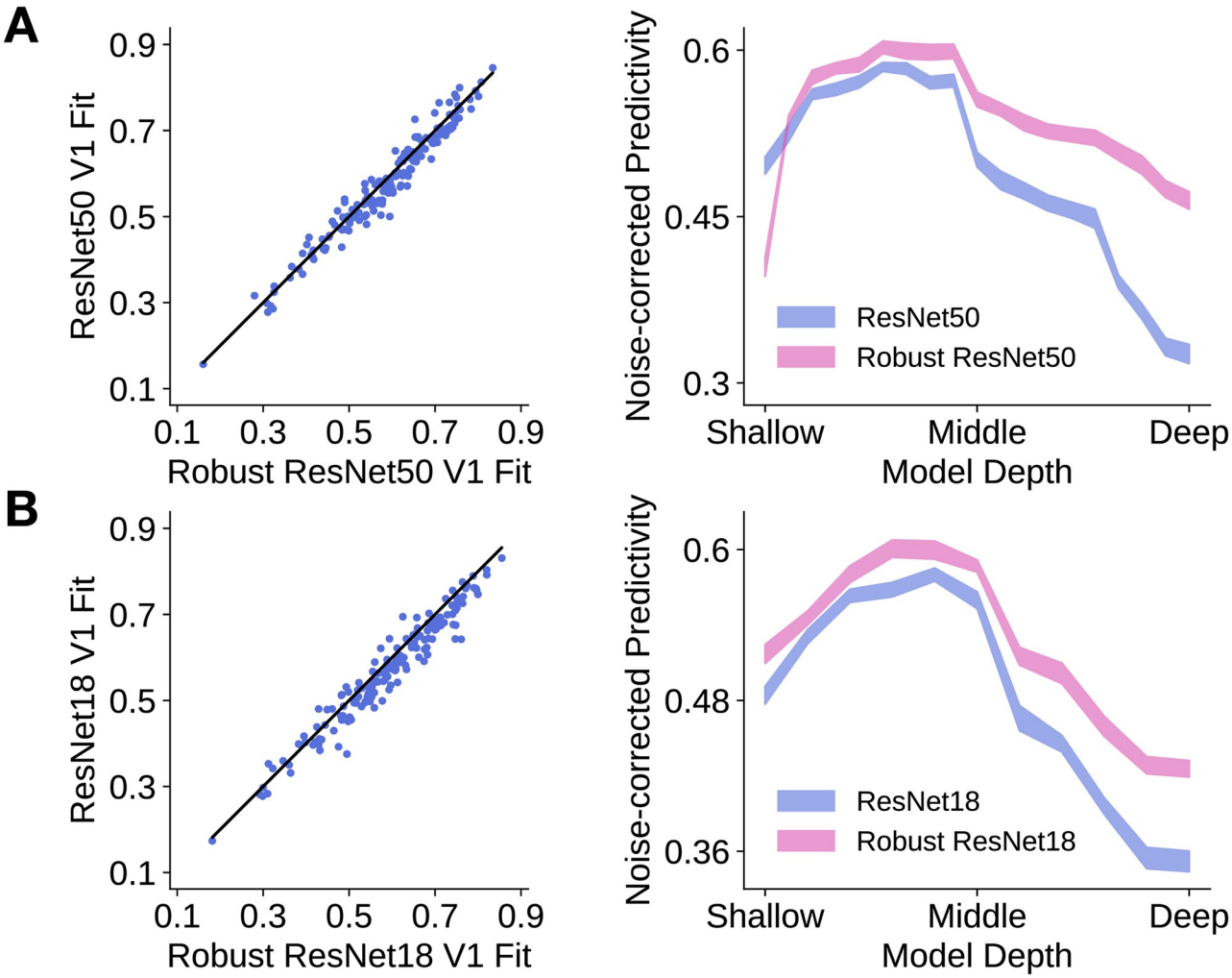

Fig 5. Robust models better predict macaque V1 neural responses than non-robust models. The noise-corrected predictivity for a neuron was defined to be the correlation between the predicted and observed responses, corrected by the neuron's reliability. **A**. Left: For most neurons (represented by each dot), robust ResNet-50 has higher neural predictivity than non-robust ResNet-50. The black line denotes the identity line. Right: Median neural predictivity across neurons of a robust and non-robust ResNet-50 across model layers. Shaded region indicates the standard deviation of the median neural predictivity across 20 train-test image splits. **B**. As in **A**, but for robust and non-robust ResNet-18. For both ResNet-50 and ResNet-18, the neural predictivities for the robust models are significantly better than those of the non-robust models ($p < 0.01$ for both architectures). Statistical significance was determined by bootstrap resampling of the neurons (with replacement) 10 000 times.

https://doi.org/10.1371/journal.pcbi.1009739.g005

Table and the predictivity across each model layer is shown in S4 Fig. We also compared each model's robustness with its power law exponent, which was obtained from the model layer that had the highest macaque V1 neural predictivity, determined by partial least squares regression.

Across the set of 40 models, we observed that a model's adversarial accuracy was strongly correlated to its V1 neural response predictivity ($R = 0.772$, $p < 0.001$; Fig 6A). This correlation was robust to the linear regression procedure used (S1 Fig left shows the relationship when ridge regression was used instead). This corroborates prior work of Dapello et al. [30], who had similar observations using a slightly different set of models and a different macaque V1 neural response dataset.

When comparing adversarial accuracy to power law exponent across models, we found a weak relationship between these two quantities ($R = 0.362$, $p = 0.022$; Fig 6B). This relationship, however, was not robust to the linear regression procedure used, as shown in S1 Fig

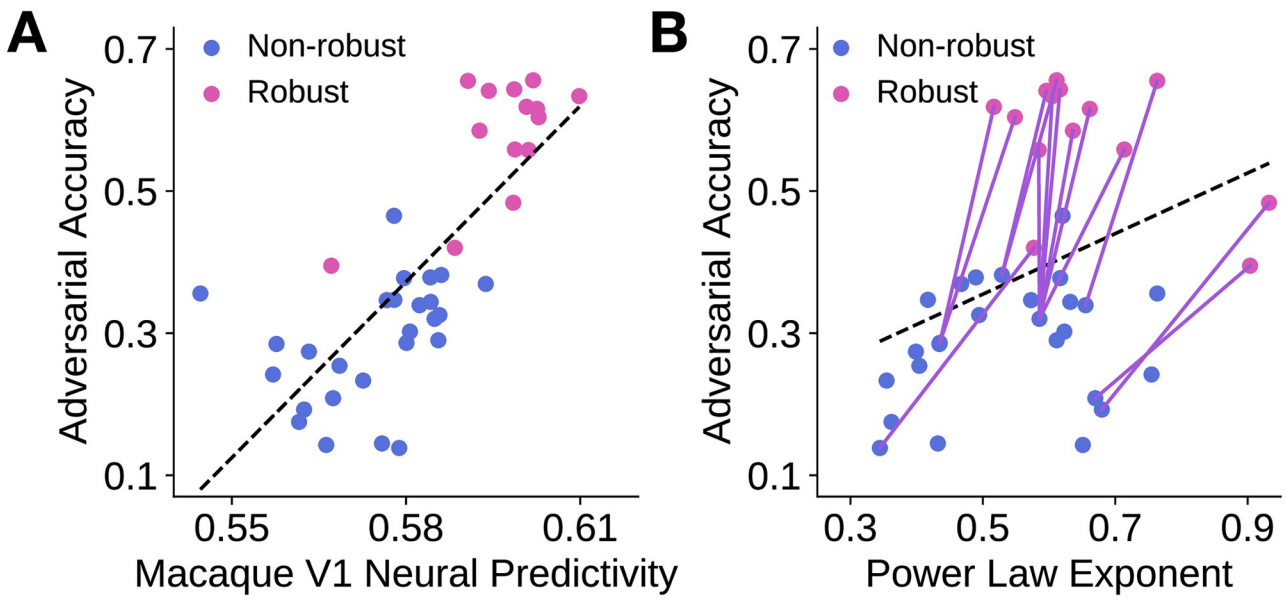

Fig 6. Adversarial accuracy is correlated with V1 neural predictivity and is weakly correlated with power law exponent. Each model is represented by a dot in each subfigure, with blue denoting non-robust models and pink denoting robust models. Dashed line indicates the line of best fit through the data points. **A**. A model's adversarial accuracy is plotted against its (maximum) V1 neural response predictivity. As mentioned previously, neural predictivity was defined to be the noise-corrected Pearson correlation between the predicted and observed neural responses. $R = 0.772$, $p < 0.001$. **B**. A model's adversarial accuracy is plotted against the power law exponent of its most "V1-like" layer, which was determined by neural predictivity. Each purple line connects two models of the *same* architecture, where one is trained *without* robustness penalties (blue) and the other is trained *with* robustness penalties (pink). $R = 0.362$, $p = 0.022$.

https://doi.org/10.1371/journal.pcbi.1009739.g006

(right). Although there was not a strong linear relationship *across* models, we found that when comparing a robust model with its non-robust counterpart (i.e., where both models have the same architecture, but one is "robustified" using robustness penalties), the robust model generally had a higher power law exponent. This is shown by the purple lines pointing to the upper right in S1 Fig (right). This result is consistent with the predictions of the theory proposed by Stringer et al. [26].

Higher model robustness is associated with higher alignment with V1 of their preferred spatial frequency tuning distributions

Previously, we described two robust models whose preferred spatial frequency tuning distributions of their most "V1-like" model layers were more like that of macaque V1 than the distributions of non-robust models. In particular, robust models had more artificial neurons that preferred "middle" spatial frequencies (i.e., approximately 3 cpd). We next investigated whether or not this pattern extended across a larger set of architectures. We found that the adversarial accuracy of a model was weakly correlated to its spatial frequency score ($R = 0.544$, $p < 0.001$; Fig 7A), indicating that the image features extracted by more robust models have spatial frequencies that might be more aligned with the image features extracted by V1. This result was robust to the linear regression procedure used to select the "V1-like" model layer, as shown in S2 Fig (left).

Finally, as shown in Fig 7B, we found that the more similar a model's preferred frequency distribution is to that of macaque V1, the higher the model's macaque V1 neural response predictivity ($R = 0.663$, $p < 0.001$). This indicates that our metric for the similarity of a preferred spatial frequency distribution to that of macaque V1 can serve as a reasonable proxy for how good a

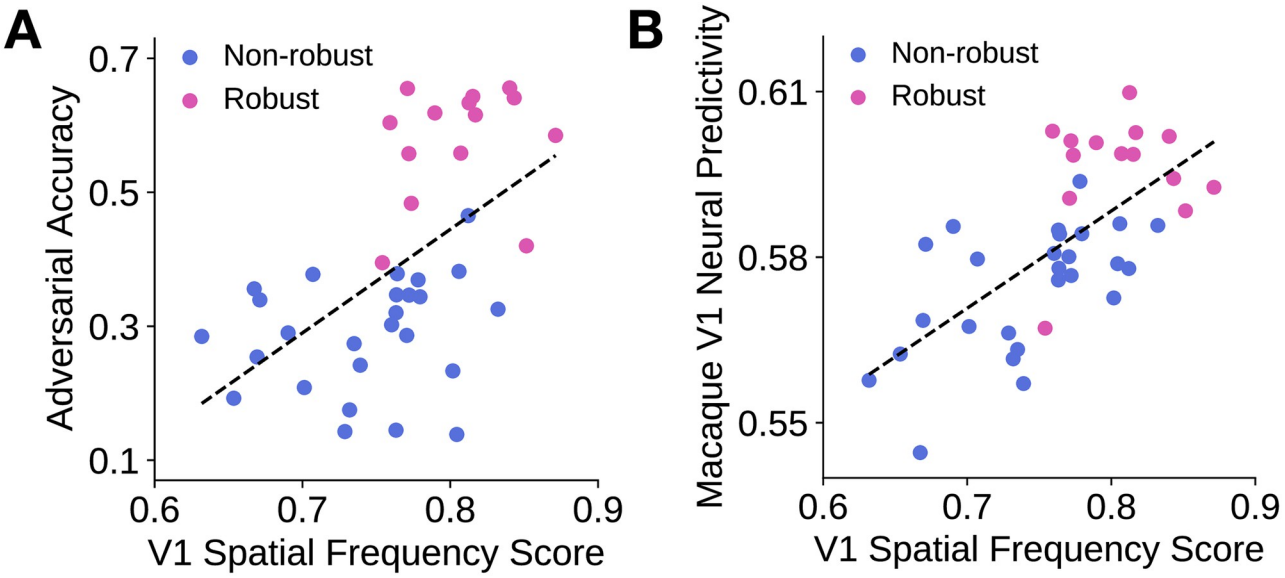

Fig 7. V1 spatial frequency score is somewhat correlated with adversarial accuracy and is correlated with V1 neural predictivity. Each model is represented by a dot in each subfigure, with blue denoting non-robust models and pink denoting robust models. Dashed line indicates the line of best fit through the data points. **A**. A model's adversarial accuracy is plotted against its V1 spatial frequency score, which denotes the similarity between a model's preferred spatial frequency distribution and that of macaque V1 cells. $R = 0.544$, $p < 0.001$. **B**. A model's maximum macaque V1 neural response predictivity is plotted against its V1 spatial frequency score. $R = 0.663$, $p < 0.001$.

https://doi.org/10.1371/journal.pcbi.1009739.g007

model is of V1. Of course, this metric can be combined with several other metrics concerning other phenomena of V1, as in work by Marques et al. [31]. This result was also robust to the linear regression procedure used to select the "V1-like" model layer, as shown in S2 Fig (right).

Discussion

Task-optimized CNNs are susceptible to human-imperceptible adversarial perturbations. In this work, we investigated properties of these CNNs related to robustness to these perturbations in order to gain more insights as to why these models are so brittle. The theory of Stringer et al. [26] suggested that the power law exponent of the eigenspectrum of a set of neural responses (to natural scenes) may be indicative of how prone a system is to small stimulus perturbations, where power law exponents larger than and closer to one would be indicative of a stimulus-neural response mapping that is less susceptible to small input perturbations [26]. We showed that the eigenspectra of mouse and macaque V1 neural responses obeyed the theory's predictions and both followed a power law with exponent at least one. Analyzing the models' eigenspectra, we found that they decayed more slowly relative to the neurophysiology. Moreover, when fixing a model architecture, models that were more robust to image perturbations had larger power law exponents than those of non-robust models. However, robustness was not correlated with power law exponent, somewhat consistent with the theory's predictions. Since a slow decay of the eigenspectrum suggests that substantial model response variance is related to the encoding of fine stimulus features, we performed in-silico electrophysiology experiments in order to assess the spatial frequency tuning of these models and found that models had a large proportion of neurons tuned to high spatial frequencies. Furthermore, robust models had preferred spatial frequency tuning distributions that were more like that of macaque V1 cells and also improved macaque V1 neural response predictions. Taken together, these results describe another way in which machine perception differs from human perception and also suggests

that one way in which our visual system achieves robustness to small image perturbations is by ignoring high spatial frequency information in an image.

Our result that a large proportion of artificial neurons are tuned to high spatial frequencies is consistent with other findings, providing additional evidence that there are differences in the image features used by humans and by machines for image classification. For example, Geirhos et al. [22] showed that when image classification-trained models were presented with ambiguous images, they tended to classify images based on the images' "texture" properties as opposed to their "shape" properties. This is in contrast to humans, who generally classified the ambiguous images based on their "shape" properties. Humans are, by definition, invariant to adversarial perturbations. Ilyas et al. [23] suggested that these perturbations are in fact "non-robust" features—features that are only weakly correlated with an image label, but still provide useful information for the model to learn a good image-label mapping. To show this, the authors constructed a dataset where only non-robust features were useful for the task. As humans are invariant to these non-robust features, the dataset appears to be completely misclassified (cf. Fig 2 in Ilyas et al. [23]). Models that were trained using this "non-robust" dataset attained non-trivial accuracy on a normal test set (i.e., test set images that were not adversarially perturbed), providing evidence that these non-robust features—features that humans are invariant to and presumably do not use—are used by models for image classification.

The model's preference for high spatial frequencies (relative to that in macaque V1) is also consistent with work that investigated model robustness to image corruptions through the lens of Fourier analysis on images. Yin et al. [25] found that even when models were trained on images that were strongly high-pass filtered, models were able to achieve non-trivial accuracy on ImageNet, indicating the fact that models can detect high-frequency image components that are both useful for the image-classification task and imperceptible to humans (cf. Fig 1 in Yin et al. [25]). They additionally found that adversarially training models and training models with Gaussian data augmentation both resulted in models that were less sensitive to high-frequency noise, but more sensitive to low to mid frequency noise, suggesting that training models in these ways results in a weaker dependence on high-frequency image components.

We observed that models that better corresponded to macaque primary visual cortex also had improved adversarial robustness, suggesting that building more "V1-like" models would improve model robustness. One strategy to develop models that are more "V1-like" is to explicitly optimize the artificial neural representations to be more like the representations obtained from V1, while simultaneously optimizing for task performance. One example of this strategy is work by Li et al. [32], who showed that models can be regularized by optimizing model representations to be similar to those computed using neural responses to natural scenes in primary visual cortex of mice. The authors showed that by incorporating this regularization into the objective function for image classification, model robustness can be improved. In a similar vein, Safarani et al. [33] showed that simultaneously training a model to perform image classification and to predict macaque V1 neural responses led to improvements in robustness to common image corruptions. Another strategy to improve model robustness could be to build into models known properties of the visual system, as humans are invariant to small image perturbations. For example, Dapello et al. [30] constructed a module based on known properties of primary visual cortex, such as the distributions of preferred orientation and of spatial frequency. The authors then showed that prepending this module to state-of-the-art CNN architectures and optimizing the whole model (except for the module) to perform ImageNet classification can improve adversarial robustness to small image perturbations. Since adversarial robustness was only improved for small image perturbations and their models were outperformed by adversarially trained models in larger perturbation size regimes, a simple smoothing of the inputs may not be sufficient to build more robust models.

The work described above, however, do not provide normative explanations for the characteristics observed in primary visual cortex (i.e., how these characteristics arose in the first place), as the V1 properties are either learned in a data-driven manner or hard-coded into the models. Our results suggest that such V1 properties as the power law exponent and preferred spatial frequency tuning distribution may arise in order to be robust to high-frequency noise or minute input perturbations. Furthermore, they could provide insight into objective functions or constraints leading to improved robustness and to the phenomena observed in primary visual cortex. Recall our observation that when a model is trained with robust optimization algorithms, the power law exponents in shallow and middle convolutional layers increases and is slightly closer to one (Fig 3). This suggests that it may be useful to explicitly optimize the eigenspectrum of the features in the shallow and the middle layers to have a power law exponent closer to one, while simultaneously optimizing for task performance. Nassar et al. [34] have made progress in this direction. They introduced a novel regularization term that explicitly penalizes eigenspectra which do not have a power law exponent of one and showed that this can improve the adversarial robustness of CNNs trained on a small dataset of handwritten digits. Important future work, we believe, would be to incorporate these regularization methods in a computationally tractable manner in large-scale image classification tasks, as higher performance on such tasks is associated with more quantitatively accurate models of the ventral visual stream [8, 11, 14].

We also observed that more robust models had preferred spatial frequency distributions more aligned with that of primary visual cortex. In particular, there was a larger proportion of artificial neurons in robust models than in non-robust models that preferred spatial frequencies in the middle (Figs 4B and 7A). The development of constraints or regularization methods to tune the preferred frequency distribution is, to our knowledge, an open problem. However, it may be the case that one does not need to explicitly constrain the convolutional filters to prefer particular spatial frequencies. Instead, one could alter the image statistics during training to bias models to learn convolutional filters that extract features across a larger extent of an image. In particular, it is known that infant visual acuity is poor early in development as a result of retinal immaturity and improves over time [35, 36]. This means that early in development, the visual cortex of infants effectively receives images of low spatial resolution as input, which increase in resolution over time. Training CNNs to perform face recognition with blurred inputs has been shown to result in convolutional filters of lower spatial frequencies [37]. We leave it to future work to investigate the implications of a developmental sequence of image resolutions during task-optimization for the preferred spatial frequency distribution of and the adversarial robustness of models.

Methods

Macaque V1 neural response dataset

We used a previously collected dataset of neural responses from macaque V1 [13]. We briefly describe the dataset here and refer the reader to the original publication for further details on the experiment [13]. Two macaques were presented with 1450 natural scenes and 5800 synthetically generated images at approximately 2˚ of visual angle. The synthetic images were generated such that their higher-order image statistics (as defined by features in various layers of VGG19) matched those of a natural image. Each stimulus was presented for 60 ms and a linear, 32-channel array was used to record spiking activity. Spike counts were obtained in the 40–100-ms time window post stimulus onset. In our analyses, stimuli that did not have at least two trials per neuron were removed, leaving a total of 6250 stimuli. 166 neurons were obtained for further analyses so that the neural response dataset was of dimensions 6250 stimuli × 166

neurons (after averaging across trials). The Spearman-Brown corrected, cross-trial correlation of each neuron was used in the noise-correction of the predictivity metric for each model layer. We found that the median of these values across neurons was 0.428.

Convolutional neural network architectures

A set of 26 models trained without robustness penalties and 14 models trained using various robust optimization algorithms [18–21, 28] were used in the analyses. We refer the reader to S1 Table for the complete list of models used in this work. Here we provide more information about the models.

Non-robust models. The non-robust models that we used include: AlexNet [2], VGGs [3], ResNets [6], wide ResNets [38], SqueezeNets [39], ShuffleNets [40], DenseNets [41], GoogleNet [42], Inception [43], MobileNet [44] and MNASNets [45]. All of these models were pre-trained on ImageNet (in a supervised manner) and accessed through the model zoo of PyTorch [46]. We additionally performed the analyses on a ResNet-50 that was previously trained using an unsupervised algorithm (SimCLR, [47]). This model's linear evaluation head (which was used for obtaining transfer learning performance on ImageNet) was kept for its robustness evaluation.

Robust models. The robust models we used are *somewhat* robust to adversarial perturbations. We consider these models as *somewhat* robust because even after training with these algorithms, there still exists perturbations that can fool these models, evidenced by the fact that they do not achieve the same accuracy on adversarially perturbed images as on unperturbed images (see S1 Table). These models were trained using four different algorithms. We briefly describe the algorithms below.

Adversarial training [18, 28]. In adversarial training, one seeks find model parameters that minimize the loss due to the worst-case perturbation:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}}[\max_{\boldsymbol{\delta}\in\Delta} \mathcal{L}(\boldsymbol{x}+\boldsymbol{\delta},\boldsymbol{y};\boldsymbol{\theta})], \tag{1}$$

where Δ is the set of "allowed" input perturbations, $\boldsymbol{\delta}$ is the perturbation, $\mathcal{L}$ is a loss function (e.g., cross-entropy loss), $\boldsymbol{x}$ is the input (e.g., an image), $\boldsymbol{y}$ is the output (e.g., image label) and $\boldsymbol{\theta}$ are the model parameters. The inner maximization of Eq (1) is performed using a few steps of projected gradient ascent. More details about this algorithm can be found in S1 Appendix.

Intuitively, this algorithm improves the robustness of models by training models using adversarial examples that are generated during each iteration of training. As additional steps are required to generate adversarial examples, this algorithm can be many times slower than generic training (i.e., supervised training). Adversarially trained models were generously released by Salman et al. [28]. In S1 Table, models trained using this algorithm are denoted as `robust_`*, where * is the name of the base architecture that was adversarially trained.

TRADES [19]. This algorithm seeks to improve the adversarial robustness of a model by adding a novel regularization term to the cross-entropy loss. The loss function $\mathcal{L}_{\text{TRADES}}$ is defined as follows:

$$\begin{aligned}
\mathcal{L}_{\text{cross-entropy}}(\boldsymbol{x}_i;\boldsymbol{\theta}) &= -\log\left(\frac{\exp(\boldsymbol{z}_i[c])}{\sum_{j=0}^{C-1}\exp(\boldsymbol{z}_i[j])}\right), \qquad i=\{1,\ldots,N\} \\
\mathcal{L}_{\text{robust}}(\boldsymbol{x}_i;\boldsymbol{\theta}) &= \max_{\boldsymbol{v}\in\mathcal{B}(\boldsymbol{x}_i,\varepsilon)} \mathsf{D}_{\mathsf{KL}}(f(\boldsymbol{v}) \parallel f(\boldsymbol{x}_i)), \qquad i=\{1,\ldots,N\} \\
\mathcal{L}_{\text{TRADES}}(\boldsymbol{X};\boldsymbol{\theta}) &= \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{\text{cross-entropy}}(\boldsymbol{x}_i;\boldsymbol{\theta}) + \beta\mathcal{L}_{\text{robust}}(\boldsymbol{x}_i;\boldsymbol{\theta}),
\end{aligned} \tag{2}$$

where N is the batch size, $C = 1000$ is the number of categories in ImageNet, $\boldsymbol{z}_i \in \mathbb{R}^C$ are the model outputs (i.e., logits) for image $\boldsymbol{x}_i$, $c \in [0, C-1]$ is the category index of the image (zero-indexed), $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ is a batch of N images, $\mathcal{B}(\boldsymbol{x}_i, \boldsymbol{\varepsilon})$ is a neighbourhood of $\boldsymbol{x}_i$ of radius $\boldsymbol{\varepsilon}$, $f(\boldsymbol{u}) \in \mathbb{R}^C$ is the vector of log-probabilities for image $\boldsymbol{u}$ belonging to each of the C categories, $\mathrm{D}_{\mathrm{KL}}(\cdot \parallel \cdot)$ is the Kullback-Leibler divergence between the two quantities, β is the regularization coefficient and $\boldsymbol{\theta}$ are the model parameters.

As in the work of Zhang et al. [19], the maximization in $\mathcal{L}_{\text{robust}}$ was performed using a few iterations of projected gradient ascent. During each iteration, a new image $\boldsymbol{v} = \boldsymbol{x} + \boldsymbol{\delta}$ is computed so as to increase the value of $\mathsf{D}_{\mathsf{KL}}(f(\boldsymbol{v}) \parallel f(\boldsymbol{x}))$. The steps are nearly identical to those used for the inner maximization in Eq (1). In particular, $\mathcal{L}(\boldsymbol{v}^{(t)}; \boldsymbol{\theta}) \coloneqq \mathsf{D}_{\mathsf{KL}}(f(\boldsymbol{v}^{(t)}) \parallel f(\boldsymbol{x}))$, and the `Project`$(\cdot, \cdot)$ function is that defined for ℓ_∞-norm constraints (i.e., perturbations are clipped to be within $[-\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}]$).

We trained a ResNet-50 architecture on ImageNet using the loss function defined in Eq (2). The loss function was minimized using stochastic gradient descent (SGD, [48]) with momentum for 80 epochs. We used a batch size of 128, momentum of 0.9 and an initial learning rate of 0.1, which was decayed by a factor of 10 at epochs 25, 45 and 65. The maximization in $\mathcal{L}_{\text{robust}}$ of Eq (2) was performed using three gradient ascent steps, with a step size of $\eta = 4/255 \times 2/3$ and the regularization coefficient was set to $\beta = 2$. The maximum perturbation size allowed was $\|\boldsymbol{\delta}\|_\infty \leq 4/255$. Finally, the weight decay was set to 0.0001. The model trained with this algorithm is denoted as `trades_robust_resnet50_linf_4` in S1 Table.

Input gradient regularization [21]. This algorithm seeks to improve the adversarial robustness of models by adding a regularization term to the cross-entropy loss. At a high-level, the regularization term penalizes the gradient of the loss function with respect to the input. Concretely, the loss function $\mathcal{L}_{\text{IGR}}$ is defined as follows:

$$\begin{aligned} \boldsymbol{z}_i &= \boldsymbol{x}_i + h \cdot \frac{\nabla_{\boldsymbol{x}_i} \mathcal{L}(\boldsymbol{x}_i; \boldsymbol{\theta})}{\|\nabla_{\boldsymbol{x}_i} \mathcal{L}(\boldsymbol{x}_i; \boldsymbol{\theta})\|_2}, \qquad i = \{1, \ldots, N\} \\ \mathcal{L}_{\text{IGR}}(\boldsymbol{X}; \boldsymbol{\theta}) &= \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\boldsymbol{x}_i; \boldsymbol{\theta}) + \frac{\lambda}{2h^2} \left(\mathcal{L}(\boldsymbol{z}_i; \boldsymbol{\theta}) - \mathcal{L}(\boldsymbol{x}_i; \boldsymbol{\theta})\right)^2, \end{aligned} \tag{3}$$

where $\mathcal{L}(\cdot; \boldsymbol{\theta})$ is the cross-entropy loss, $\lambda = 0.3$ is the regularization coefficient, $h = 0.01$ is the finite difference hyperparameter, $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ is a batch of N images and $\boldsymbol{\theta}$ are the model parameters.

A model with the ResNet-50 architecture was trained to minimize Eq (3) using SGD with momentum for 100 epochs. We used a batch size of 128, momentum of 0.9 and an initial learning rate of 0.1, which was decayed by a factor of 10 at epochs 35, 70 and 90. Finally, the weight decay was set to 0.0001. The model trained with this algorithm is denoted as `igr_robust_resnet50` in S1 Table.

Adversarial training for free [20]. As mentioned previously, adversarial examples are generated on each iteration of adversarial training. Thus, assuming that K steps are used to generate the adversarial examples during each iteration, adversarial training can be $K + 1$ times slower than generic training. "Free adversarial training" was developed in order to reduce the total time required to train these models. At a high level, gradients with respect to the input are accumulated over multiple training steps circumventing the need to compute the gradient multiple times during each training step.

A model with the ResNet-50 architecture was trained using this algorithm for 23 epochs, where each batch of 256 images was repeated four times during each epoch. SGD with momentum of 0.9 was used and the weight decay was set to 0.0001. The initial learning rate of

0.1 was decayed by a factor of 10 every 30 epochs. The maximum perturbation size allowed was $\|\boldsymbol{\delta}\|_\infty = 4/255$ and the step size was also 4/255. The model trained with this algorithm is denoted as `free_robust_resnet50_linf_4` in S1 Table.

Computing power law exponents

Power law exponent of macaque V1 neural responses. In order to compute the eigenspectrum of the macaque V1 neural responses to natural scenes, we used cross-validated principal components analysis (cvPCA). It was developed by Stringer et al. [26] to compute the eigenspectrum of neural responses from mouse V1. This algorithm computes unbiased estimates of the eigenvalues (and hence the eigenspectrum) of the population neural responses. Briefly, the algorithm operates as follows:

$$\begin{aligned}
\boldsymbol{X}^{(1)} &= \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^\top && \text{(singular value decomposition)} \\
\widetilde{\boldsymbol{X}}^{(1)} &= \boldsymbol{X}^{(1)}\boldsymbol{V} && \text{(project data onto eigenvectors)} \\
\widetilde{\boldsymbol{X}}^{(2)} &= \boldsymbol{X}^{(2)}\boldsymbol{V} && \\
\boldsymbol{\lambda}_j &= \sum_{i=1}^{S} \widetilde{\boldsymbol{X}}_{ij}^{(1)} \widetilde{\boldsymbol{X}}_{ij}^{(2)}, \quad \text{for } j \in \{1, \ldots, C\} && \text{(compute eigenvalue)}
\end{aligned}$$

where S is the number of stimuli, N is the number of neurons, $\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)} \in \mathbb{R}^{S \times N}$ are the neural responses for the first and second half of the trials (and averaged across trials), $\boldsymbol{V} \in \mathbb{R}^{N \times C}$ are the C eigenvectors of the covariance matrix of $\boldsymbol{X}^{(1)}$ and $\boldsymbol{\lambda} \in \mathbb{R}^C$ are the cross-validated eigenvalues associated with each of the eigenvectors ($\boldsymbol{\lambda}_j$ is the jth eigenvalue).

The first step of the cvPCA algorithm computes the eigenvectors of the neural response covariance from one set of the trials. The second and third steps project the neural responses from each half of the trials onto each eigenvector. The final step computes the (scaled) variance of the neural responses when projected onto an eigenvector (that was computed using one half of the trials). Thus, each cross-validated eigenvalue is related to the amount of stimulus-related variance of the neural responses along the eigenvalue's corresponding eigenvector. The power law exponent was then determined as the negative slope of the line of best fit of the eigenspectrum in log-log space, similar to the procedure described by Stringer et al. [26]. We refer the reader to the original publication for a more detailed mathematical analysis of this method [26]. We ran this algorithm 20 times for the macaque V1 neural response dataset and averaged the eigenvalues computed from each of the 20 runs.

Power law exponents of artificial neural responses. The eigenspectrum of artificial neural responses to three random sets of 2816 images from the ImageNet validation set was computed for each model layer. Each image was first resized so that its shorter dimension was 256 pixels and then center-cropped to 224 × 224 pixels. Images were additionally preprocessed by normalizing each image channel (RGB channels) using the mean and standard deviation that was used during model training. Using these preprocessed images, we extracted activations from several layers of each CNN and computed their eigenspectra using principal components analysis (PCA). We did not use cvPCA, as we did for the macaque V1 neural responses, because artificial neural responses are deterministic. Similar to the procedure described by Stringer et al. [26], the power law exponent was estimated as the negative slope of the line of best fit of the eigenspectrum in log-log space over the principal component indices in the range of 10 to 999. For the analysis pertaining to the comparison of a model's robustness with its power law exponent, we summarized each model by the power law exponent of the model layer that best predicted the macaque V1 neural responses.

Spatial frequency tuning

Preferred spatial frequency tuning distribution of models. In order to assess the spatial frequency tuning of artificial neurons, we performed in-silico electrophysiology experiments. Specifically, we first generated Gabor patches of various orientations, spatial frequencies and phases according to the following equation:

$$\begin{aligned} \begin{bmatrix} x' \\ y' \end{bmatrix} &= \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ f(x, y; \sigma, \lambda, \psi, \theta, \gamma) &= \exp\left(-\frac{x'^2 + \gamma y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right), \end{aligned} \tag{4}$$

where $\sigma = 35$ determines the standard deviation of the Gaussian envelope in pixels, $\gamma = 1$ is the aspect ratio of the Gabor patch, λ is the number of pixels for one cycle of the sinusoid, ψ is the phase of the sinusoid in radians and θ is the orientation of the Gabor patch in radians. We generated Gabor patches with different orientations, phases and spatial frequencies using the following parameters: 10 orientations were evenly spaced between 0˚ and 172.5˚ and 10 phases were evenly spaced between 0˚ and 360˚. The following spatial frequencies were used (in units of cycles per image): 2.5, 3.5, 5, 7.1, 10, 14.1, 20, 28.3, 40, 56. This resulted in $10 \times 10 \times 10 = 1000$ Gabor patch stimuli. Prior to presenting the Gabor patch stimuli to the models, each stimulus was preprocessed by normalizing the RGB channels using the mean and standard deviation that the model was trained on.

Here we describe the method by which we obtained spatial frequency tuning curves of artificial neurons from model layers. The output of a convolutional layer is a matrix of dimensions $C \times H \times W$, where C is the number of channels (i.e., convolutional filters), and H and W are the height and width. Since the model layer is convolutional, each artificial neuron in each channel would have the same "tuning". Intuitively, each artificial neuron in the same channel would detect the same "features" as they are each associated with the same convolutional filter. As a result, one does not need to obtain the tuning for *all* artificial neurons in the output of a convolutional layer. Thus, for a particular channel (i.e., convolutional filter), we computed the tuning for the artificial neuron at the center of the activations matrix, which we denote as the "representative neuron" (i.e., the neuron at location $(\lfloor H/2 \rfloor, \lfloor W/2 \rfloor)$). The receptive field of the central artificial neuron would cover the Gabor patch stimuli, as the Gabor patch is placed at the center of the model's visual field.

To compute the value of the tuning curve for an artificial neuron at a particular spatial frequency, we averaged the activations of the artificial neuron to Gabor patch stimuli of all orientations and phases with that particular spatial frequency. This was performed for each of the desired spatial frequencies. Once the tuning curves were computed, artificial neurons were further sub-selected according to their tuning curves' peak-to-peak value. Artificial neurons were kept only if the peak-to-peak value of their tuning curves were greater than zero. The neuron's preferred spatial frequency was defined to be the frequency at which the tuning curve achieves its maximum value.

To mimic one electrophysiology experiment, we randomly sampled 150 representative neurons (with replacement) from the model layer's output and obtained each neuron's preferred spatial frequency, resulting in the model layer's preferred spatial frequency distribution. We performed 1000 in-silico electrophysiology experiments and therefore obtained 1000 preferred spatial frequency distributions for a particular model layer. Each of these distributions was then compared with that of macaque V1 cells and a score was computed for each distribution, resulting in 1000 scores (see below for the definition of the scoring function). Each model's

score was then defined to be the average score across the 1000 in-silico experiments (each of which is presented in Fig 7A) and the error was defined to be the standard deviation of the scores across the in-silico experiments.

Preferred spatial frequency tuning distribution of macaque V1. Using the online tool called "WebPlotDigitizer" (https://apps.automeris.io/wpd/), we extracted data from Fig 6 of De Valois et al. [29], which shows the preferred spatial frequency distribution of neurons in the foveal area of macaque V1. The extracted spatial frequency bins were as follows (in units of cycles per degree): 0.35, 0.5, 0.7, 1.0, 1.4, 2.0, 2.8, 4.0, 5.6, 8.0 (with 11.2 as the rightmost bin edge). The extracted cell counts for each spatial frequency bin were as follows: 0, 3, 3, 6, 17, 22, 18, 19, 9, 4.

V1 spatial frequency score. Here we describe the metric that we used to assess the similarity between a model's preferred spatial frequency distribution and that of macaque V1 cells. The score was defined to be one minus the maximum absolute difference between the two empirical cumulative distributions (similar to the Kolmogorov-Smirnov distance):

$$\text{score}(\boldsymbol{x}, \boldsymbol{y}) = 1 - \max_i |\boldsymbol{x}_i - \boldsymbol{y}_i|, \tag{5}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are the empirical cumulative distributions of two different samples and $\boldsymbol{x}_i$ is the value of the cumulative distribution at the ith bin (and correspondingly for $\boldsymbol{y}_i$). In our case, $\boldsymbol{y}$ would be the cumulative distribution for the preferred spatial frequency histogram of macaque V1 cells (obtained from the histogram in Fig 6 of De Valois et al. [29]) and $\boldsymbol{x}$ would be the model's cumulative distribution for its preferred spatial frequency histogram.

V1 neural response predictivity

In line with Cadena et al. [13], we first cropped each stimulus to the central 80 pixels and then resized the images to 40 × 40 pixels, as we also assumed that each models' field of view is 6.4 degrees. Each stimulus was then zero-padded up to the image size on which each model was trained. For example, the 40 × 40 pixels image would be zero-padded up to 224 × 224 pixels for most models. Each image channel was additionally normalized according to the mean and standard deviation used during model training. Model features were then extracted in response to each preprocessed stimulus.

In order for neural response prediction for each of the 40 models (and their representative model layers) to be more computationally tractable, we first projected the features of each model layer into a 1000-dimensional space using PCA prior to linear fitting. If the number of features was less than 1000 (e.g., there are 512 features in the average-pooling layer of a ResNet-18), the number of principal components used was equal to the number of features. ImageNet validation set images were used to compute the principal components so that the transformation was held constant across all 20 train-test splits during neural fitting [11]. These lower-dimensional stimulus features were then used as input to a partial least squares (PLS) regression procedure with 25 components, consistent with prior work [8, 11, 30, 31]. Each train-test split was generated by randomly selecting 75% of the stimuli to be in the train set and 25% of the stimuli to be in the test set. For the data shown in S1 and S2 Figs, we performed cross-validated ridge regression, where five-fold cross validation (using the train set) was used to obtain the best regularization coefficient from {0.01, 0.1, 1, 10}.

The noise-corrected predictivity metric for a neuron was defined to be the Pearson's correlation between the neuron's response predictions and the observed neural responses divided by the square-root of the Spearman-Brown corrected cross-trial correlation of the neuron's responses, consistent with prior work [14, 49].

Model robustness

We defined the robustness of a model to be its classification accuracy on the 50 000 ImageNet validation set images that have been perturbed using a set of white-box adversarial attacks and averaged across the set of adversarial attacks. This is referred to as the model's "adversarial accuracy". Adversarial perturbations were generated using projected gradient ascent (PGD, [18]). The algorithm used to generate adversarial images is the same as that used for the inner maximization in Eq (1) of adversarial training. For more details, we refer the reader to S1 Appendix.

If the image perturbations can be as large as possible, one can easily distort the image so that it becomes completely unrecognizable by a human (e.g., by modifying the image so that it looks like noise). Thus, the sizes (i.e., norm) of the perturbations, $\boldsymbol{\delta}$, were bounded so that they are imperceptible to humans. In order to generate more variability in the adversarial accuracy across models, we used relatively small adversarial perturbations. Larger perturbations would reduce the adversarial accuracy of most CNNs to chance level. Specifically, the maximum sizes of the perturbations were defined as follows: $\|\boldsymbol{\delta}\|_\infty \leq 1/1020$, $\|\boldsymbol{\delta}\|_2 \leq 0.15$, $\|\boldsymbol{\delta}\|_1 \leq 40$. Adversarial examples were generated using projected gradient ascent for 20 steps. The step size was set to be $\varepsilon \times 2/20$, where ε is the maximum allowed size of the perturbation. This white-box adversarial attack method and these perturbation constraints were also used in previous work that compared adversarial accuracy with V1 neural response predictivity [30]. We used a Python package known as Foolbox [50] to evaluate the robustness of the CNNs.

Optimal stimulus visualization

As in prior work, we optimized the discrete Fourier transform (DFT) of the input to maximize the activations of a particular channel of a CNN layer. Concretely, we maximized the softmax of the average activation in the desired channel of the output of a convolutional layer:

$$g(\boldsymbol{X}, i) = \frac{\exp\left(\frac{1}{H \cdot W}\sum_{h=0}^{H-1}\sum_{w=0}^{W-1}\boldsymbol{X}[i,h,w]\right)}{\sum_{c=0}^{C-1}\exp\left(\frac{1}{H \cdot W}\sum_{h=0}^{H-1}\sum_{w=0}^{W-1}\boldsymbol{X}[c,h,w]\right)}, \quad (6)$$

where i is the index of the channel we would like to maximize the activations of, H and W are the height and width of the outputs of the convolutional layer, C is the total number of channels in the output of the convolutional layer and $\boldsymbol{X} \in \mathbb{R}^{C \times H \times W}$ is the output of the convolutional layer. Eq (6) was maximized using the Adam optimizer with a learning rate of 0.05. We used a Python package known as Lucent (a PyTorch adapation of Lucid [51]) to generate the optimal stimuli.

Supporting information

S1 Fig. Relationships between adversarial accuracy, V1 neural predictivity and power law exponent. In the main text, we showed results where the "V1-like" layer of a model was obtained via partial least squares regression, where we found that adversarial accuracy correlated with V1 neural predictivity and weakly correlated with power law exponent (recall that the power law exponent for a model was obtained from the model layer that best predicted the macaque V1 neural responses). Here, we show the same figures, but with results obtained via cross-validated ridge regression (where five-fold cross-validation was used to obtain the optimal regularization coefficient). Consistent with our partial least squares regression finding, adversarial accuracy was correlated with V1 neural predictivity ($R = 0.696$, $p < 0.001$).

However, when using ridge regression to determine the most "V1-like" model layer, adversarial accuracy was found *not* to correlate with power law exponent ($R = 0.159$, $p = 0.327$). Although there was no linear relationship between adversarial accuracy and power law exponent, we noticed that when comparing two models obtained by training a single architecture with and without robustness penalties, the robust model had higher power law exponents (as shown by the purple lines pointing to the upper right in the figure, indicating higher adversarial accuracy and higher power law exponent). This is consistent with the theory of Stringer et al. [26].
(TIFF)

S2 Fig. Relationships between adversarial accuracy, V1 spatial frequency score and V1 neural predictivity. In the main text, we showed the relationship between a model's adversarial accuracy, its V1 spatial frequency score and its maximum V1 neural predictivity when partial least squares regression was used to obtain a model's "V1-like" layer. Here, we show results pertaining to these relationships, obtained via cross-validated ridge regression. Qualitatively, the results are the same as those described in the main text. Here, we find that a model's adversarial accuracy and its V1 spatial frequency score was correlated ($R = 0.624$, $p < 0.001$). Furthermore, a model's maximum V1 neural predictivity was correlated to its V1 spatial frequency score ($R = 0.565$, $p < 0.001$).
(TIFF)

S3 Fig. V1 neural response predictivity as a function of maximum perturbation size allowed during adversarial training. Here we asked whether V1 neural response predictivity is related to the maximum allowed size of perturbation used during adversarial training. Using previously adversarially trained models [28], we found, for both CNN architectures, that as the maximum allowed perturbation size (using the ℓ_2-norm) for model training increased from zero, V1 neural predictivity increased. However, V1 neural predictivity plateaus when the ℓ_2-norm of the perturbation reaches and exceeds 0.5. Thus, just increasing the maximum allowable perturbation size during adversarial training (and hence robustness to larger image perturbations) is not enough to obtain further improvements in V1 neural predictivity.
(TIFF)

S4 Fig. Macaque V1 neural response predictivity as a function of model layer for all evaluated models. We present the neural predictivity for all 40 models as a function of its model layers. Consistent with other work [13, 14], we find that shallow to middle model layers best predict neural responses to V1 neural responses for all evaluated models.
(TIFF)

S5 Fig. Eigenspectra and power law exponents as a function of the fraction of stimuli used in the macaque V1 neural response dataset. As in Stringer et al. [26], we varied the fraction of stimuli that were used in the neural response dataset and computed their eigenspectra and their associated power law exponents. For each fraction of stimuli used, we randomly sampled stimuli ten times and computed the eigenspectrum and power law exponent for each subset of the neural responses, resulting in ten power law exponents and ten eigenspectra for each fraction of stimuli. We found that for all fractions of stimuli used, the power law exponents were greater than one and were more precise as more stimuli were used. We note that the power law exponent in the macaque dataset may not have converged to one yet, so more neurons may need to be recorded in the future to further verify the power-law-like behaviour of macaque V1 neural responses.
(TIFF)

S6 Fig. Changing the range of principal component variances used to fit the power law exponent does not affect the relationship between robust and non-robust models. In the main text, we fit the power law exponent using principal component variances from indices 10 to 999. Here we fit the power law exponent for each eigenspectrum using principal component variances from indices 1 to 1000. This small modification to the fitting procedure does not alter the relationship between the power law exponents of robust models and those of non-robust models. Specifically, we found that the power law exponents of robust models were higher than those of non-robust models, implying that the dimensionality of the internal representations of robust models is slightly lower than that of non-robust models.
(TIFF)

S7 Fig. Varying the image resolution does not change the relationship between the power law exponents of robust and non-robust models. Changing the image size by downsampling the image would remove high-frequency components and thus make spectral decay steeper. Therefore, we investigated how the power law exponent varies as a function of the image resolution. We used the 1250 natural scene stimuli (which are in grayscale) from the neural response dataset of Cadena et al. [13] and varied the image resolution (in pixels) before presenting them to the models. We fixed the model architecture to be ResNet-18 and ResNet-50 (using both robust and non-robust variations of them) and varied the input resolution from 40 pixels (the size used in neural response predictions) to 80 pixels (the size of the center crop prior to the downsampling used in the neural response fitting procedure). Specifically, the image transformations were as follows: (1) Center crop the original stimulus to 80×80 pixels and (2) resize the image to one of $\{40 \times 40, 50 \times 50, 60 \times 60, 70 \times 70, 80 \times 80\}$ pixels. Using the most V1-like model layer for each of the two models, we extracted activations to the images and randomly sampled 166 artificial neurons (same as the number of neurons in the macaque V1 neural response dataset) 20 times. Using the sub-sampled model response matrix (of dimensions 1250×166), we computed their eigenspectra, the power law exponents and the index at which cumulative principal component variance reached 75%. This resulted in 20 power law exponents and principal component indices and the mean and the standard deviation across the 20 values was reported. We found that the power law exponents for both robust and non-robust models was lower than that of macaque V1 neural responses and that they decreased as a function of image resolution indicating that increasingly fine stimulus features are encoded as more information is available in the stimulus (top row). These observations were corroborated by another metric that measures the dimensionality of the model or the biological responses. Shown on the bottom row, we found that the indices at which cumulative principal component variance reached 75% for the models were higher than that of macaque V1 neural responses (i.e., higher principal component index indicates relatively higher-dimensional responses).
(TIFF)

S1 Table. Model performance on ImageNet and V1 neural predictivity. This table lists all the models we used, their macaque V1 neural response predictivity and their top-1 accuracies on ImageNet validation set images which have not been perturbed (i.e., for a perturbation, $\boldsymbol{\delta}$, and for any norm, $\|\boldsymbol{\delta}\| = 0$) or were adversarially perturbed with different norm constraints on the perturbations: $\|\boldsymbol{\delta}\|_\infty \leq 1/1020$, $\|\boldsymbol{\delta}\|_2 \leq 0.15$, $\|\boldsymbol{\delta}\|_1 \leq 40$. For the models trained to be adversarially robust, the suffix corresponds to the norm constraint imposed on the size of the perturbation during model training. For example, `robust_resnet50_l2_3` corresponds to a ResNet-50 adversarially trained to be robust to perturbations, $\boldsymbol{\delta}$, of size at most $\|\boldsymbol{\delta}\|_2 \leq 3$ [28], `igr_robust_resnet50` corresponds to a ResNet-50 trained with input gradient

regularization (IGR, [21]) and `resnet50_simclr` corresponds to a ResNet-50 trained with the SimCLR unsupervised loss function [47].
(PDF)

S1 Appendix. Additional details on adversarial training. Here we provide additional details on adversarial training and on generating adversarial examples.
(PDF)

Acknowledgments

We are grateful to Cadena et al. [13] for publicly releasing their macaque V1 neural response dataset. We thank Tyler Bonnen for helpful comments on the manuscript. N.C.L.K. is indebted to the late Matthew Brennan for discussions during the early phase of this project.

Author Contributions

Conceptualization: Nathan C. L. Kong.

Data curation: Nathan C. L. Kong.

Formal analysis: Nathan C. L. Kong, Eshed Margalit, Justin L. Gardner, Anthony M. Norcia.

Investigation: Nathan C. L. Kong.

Methodology: Nathan C. L. Kong, Eshed Margalit, Justin L. Gardner, Anthony M. Norcia.

Project administration: Nathan C. L. Kong.

Resources: Anthony M. Norcia.

Software: Nathan C. L. Kong.

Supervision: Anthony M. Norcia.

Validation: Nathan C. L. Kong.

Visualization: Nathan C. L. Kong.

Writing – original draft: Nathan C. L. Kong.

Writing – review & editing: Nathan C. L. Kong, Eshed Margalit, Justin L. Gardner, Anthony M. Norcia.

References

1. Fukushima K. Neocognitron: A hierarchical neural network capable of visual pattern recognition. Neural Networks. 1988; 1(2):119–130. https://doi.org/10.1016/0893-6080(88)90014-7
2. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems; 2012. p. 1097–1105.
3. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: International Conference on Learning Representations; 2015.
4. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015. p. 3431–3440.
5. He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision; 2015. p. 1026–1034.
6. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 770–778.

7. Khaligh-Razavi SM, Kriegeskorte N. Deep supervised, but not unsupervised, models may explain IT cortical representation. PLoS Computational Biology. 2014; 10(11):e1003915. https://doi.org/10.1371/journal.pcbi.1003915 PMID: 25375136
8. Yamins DL, Hong H, Cadieu CF, Solomon EA, Seibert D, DiCarlo JJ. Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proceedings of the National Academy of Sciences. 2014; 111(23):8619–8624. https://doi.org/10.1073/pnas.1403112111 PMID: 24812127
9. Cichy RM, Khosla A, Pantazis D, Torralba A, Oliva A. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. Scientific Reports. 2016; 6:27755. https://doi.org/10.1038/srep27755 PMID: 27282108
10. Güçlü U, van Gerven MA. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. Journal of Neuroscience. 2015; 35(27):10005–10014. https://doi.org/10.1523/JNEUROSCI.5023-14.2015 PMID: 26157000
11. Schrimpf M, Kubilius J, Hong H, Majaj NJ, Rajalingham R, Issa EB, et al. Brain-Score: Which Artificial Neural Network for Object Recognition is most Brain-Like? bioRxiv. 2018.
12. Seeliger K, Fritsche M, Güçlü U, Schoenmakers S, Schoffelen JM, Bosch S, et al. Convolutional neural network-based encoding and decoding of visual object recognition in space and time. NeuroImage. 2018; 180:253–266. https://doi.org/10.1016/j.neuroimage.2017.07.018 PMID: 28723578
13. Cadena SA, Denfield GH, Walker EY, Gatys LA, Tolias AS, Bethge M, et al. Deep convolutional models improve predictions of macaque V1 responses to natural images. PLoS Computational Biology. 2019; 15(4):e1006897. https://doi.org/10.1371/journal.pcbi.1006897 PMID: 31013278
14. Zhuang C, Yan S, Nayebi A, Schrimpf M, Frank MC, DiCarlo JJ, et al. Unsupervised neural network models of the ventral visual stream. Proceedings of the National Academy of Sciences. 2021; 118(3). https://doi.org/10.1073/pnas.2014196118 PMID: 33431673
15. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, et al. Intriguing properties of neural networks. In: International Conference on Learning Representations; 2014.
16. Kurakin A, Goodfellow I, Bengio S. Adversarial Machine Learning at Scale. In: International Conference on Learning Representations; 2017.
17. Ross A, Doshi-Velez F. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32; 2018.
18. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards Deep Learning Models Resistant to Adversarial Attacks. In: International Conference on Learning Representations; 2018.
19. Zhang H, Yu Y, Jiao J, Xing EP, Ghaoui LE, Jordan MI. Theoretically Principled Trade-off between Robustness and Accuracy. In: International Conference on Machine Learning; 2019.
20. Shafahi A, Najibi M, Ghiasi A, Xu Z, Dickerson J, Studer C, et al. Adversarial Training for Free! Advances in Neural Information Processing Systems 32. 2019;5:3358–3369.
21. Finlay C, Oberman AM. Scaleable input gradient regularization for adversarial robustness. Machine Learning with Applications. 2021; 3:100017. https://doi.org/10.1016/j.mlwa.2020.100017
22. Geirhos R, Rubisch P, Michaelis C, Bethge M, Wichmann FA, Brendel W. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations; 2019.
23. Ilyas A, Santurkar S, Tsipras D, Engstrom L, Tran B, Madry A. Adversarial Examples Are Not Bugs, They Are Features. In: Advances in Neural Information Processing Systems. vol. 32; 2019.
24. Tsipras D, Santurkar S, Engstrom L, Turner A, Madry A. Robustness May Be at Odds with Accuracy. In: International Conference on Learning Representations; 2019.
25. Yin D, Gontijo Lopes R, Shlens J, Cubuk ED, Gilmer J. A Fourier Perspective on Model Robustness in Computer Vision. In: Advances in Neural Information Processing Systems. vol. 32; 2019.
26. Stringer C, Pachitariu M, Steinmetz N, Carandini M, Harris KD. High-dimensional geometry of population responses in visual cortex. Nature. 2019; 571(7765):361–365. https://doi.org/10.1038/s41586-019-1346-5 PMID: 31243367
27. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2009. p. 248–255.
28. Salman H, Ilyas A, Engstrom L, Kapoor A, Madry A. Do Adversarially Robust ImageNet Models Transfer Better? In: Advances in Neural Information Processing Systems; 2020. p. 3533–3545.
29. De Valois RL, Albrecht DG, Thorell LG. Spatial frequency selectivity of cells in macaque visual cortex. Vision Research. 1982; 22(5):545–559. https://doi.org/10.1016/0042-6989(82)90113-4 PMID: 7112954

30. Dapello J, Marques T, Schrimpf M, Geiger F, Cox D, DiCarlo JJ. Simulating a Primary Visual Cortex at the Front of CNNs Improves Robustness to Image Perturbations. In: Advances in Neural Information Processing Systems; 2020. p. 13073–13087.
31. Marques T, Schrimpf M, DiCarlo JJ. Multi-scale hierarchical neural network models that bridge from single neurons in the primate primary visual cortex to object recognition behavior. bioRxiv. 2021.
32. Li Z, Brendel W, Walker E, Cobos E, Muhammad T, Reimer J, et al. Learning from brains how to regularize machines. In: Advances in Neural Information Processing Systems. vol. 32; 2019.
33. Safarani S, Nix A, Willeke K, Cadena SA, Restivo K, Denfield G, et al. Towards robust vision by multi-task learning on monkey visual cortex. arXiv preprint arXiv:210714344. 2021.
34. Nassar J, Sokol P, Chung S, Harris KD, Park IM. On 1/n neural representation and robustness. In: Advances in Neural Information Processing Systems; 2020. p. 6211–6222.
35. Dobson V, Teller DY. Visual acuity in human infants: a review and comparison of behavioral and electrophysiological studies. Vision Research. 1978; 18(11):1469–1483. https://doi.org/10.1016/0042-6989(78)90001-9 PMID: 364823
36. Kiorpes L. Understanding the development of amblyopia using macaque monkey models. Proceedings of the National Academy of Sciences. 2019; 116(52):26217–26223. https://doi.org/10.1073/pnas.1902285116 PMID: 31871163
37. Vogelsang L, Gilad-Gutnick S, Ehrenberg E, Yonas A, Diamond S, Held R, et al. Potential downside of high initial visual acuity. Proceedings of the National Academy of Sciences. 2018; 115(44):11333–11338. https://doi.org/10.1073/pnas.1800901115 PMID: 30322940
38. Zagoruyko S, Komodakis N. Wide Residual Networks. In: Proceedings of the British Machine Vision Conference; 2016. p. 87.1–87.12.
39. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. arXiv preprint arXiv:160207360. 2016.
40. Zhang X, Zhou X, Lin M, Sun J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2018. p. 6848–6856.
41. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 4700–4708.
42. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2015. p. 1–9.
43. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2016. p. 2818–2826.
44. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:170404861. 2017.
45. Tan M, Chen B, Pang R, Vasudevan V, Sandler M, Howard A, et al. Mnasnet: Platform-aware neural architecture search for mobile. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. p. 2820–2828.
46. torchvision.models—Torchvision master documentation; 2021. Available from: https://pytorch.org/vision/stable/models.html.
47. Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning; 2020. p. 1597–1607.
48. Bottou L. Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010. Springer; 2010. p. 177–186.
49. Nayebi A, Sagastuy-Brena J, Bear DM, Kar K, Kubilius J, Ganguli S, et al. Goal-Driven Recurrent Neural Network Models of the Ventral Visual Stream. bioRxiv. 2021.
50. Rauber J, Brendel W, Bethge M. Foolbox: A Python toolbox to benchmark the robustness of machine learning models. In: Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning; 2017.
51. Olah C, Mordvintsev A, Schubert L. Feature visualization. Distill. 2017; 2(11):e7. https://doi.org/10.23915/distill.00007