Simple, biologically informed models, but not convolutional neural networks describe target detection in naturalistic images

Ingo Fruend*

Department of Psychology, Centre for Vision Research & VISTA, York University, Toronto, ON

* ifruend@yorku.ca

Abstract

The first steps of visual processing are often described as a bank of oriented filters followed by divisive normalization. This approach has been tremendously successful at predicting contrast thresholds in simple visual displays. However, it is unclear to what extent this kind of architecture also supports processing in more complex visual tasks being performed in naturally looking images.

Here, we use a deep generative model for natural images to embed arc segments with different curvatures in naturalistic images. Specifically, these images contain the target as part of the image scene, resulting in considerable appearance variation of target as well as background. Three observers localized arc targets in these images, achieving an accuracy of 74.7% correct responses on average. Data were then fit by a number of biologically inspired models and also by a 5-layer convolutional neural network. Four models were particularly good a predicting observer responses, (i) a bank of oriented filters, similar to complex cells in primate area V1, (ii) a bank of oriented filters followed by tuned gain control, incorporating knowledge about cortical surround interactions, (iii) a bank of oriented filters followed by local normalization, (iv) the 5-layer convolutional neural network. A control experiment with optimized stimuli based on these four models showed that the observers' data were best explained by model (ii) with tuned gain control.

These data suggest that standard models of early vision provide good descriptions of performance in much more complex tasks than what they were designed for, while general purpose non-linear models such as convolutional neural networks do not.

Author summary

Early stages of visual processing are often described as a bank of oriented filters followed by divisive normalization. While this standard model successfully predicts contrast thresholds in simple visual displays, it is unclear to what extent it also supports more complex tasks performed in naturally looking images. One challenge here is that naturalistic targets are not simply superimposed on the image, but they form part of an image. We use a high-capacity image model to generate random naturalistic images constrained to contain a visual target as part of the image. Human target detection performance on these images is indeed well described by variants of the filtering and normalization approach, but a convolutional neural network model based on modern methods from artificial intelligence does so equally well. Yet, the converse is

not true; artificial stimuli constructed from the standard model drive human performance in meaningful ways, while artificial stimuli constructed from the convolutional neural network do not. Thus standard models of early vision indeed provide good descriptions of human performance in more natural stimulus conditions, while convolutional neural networks do not.

Introduction

Processing in higher levels of the visual system becomes successively invariant to the specific details of an image. For example, orientation tuning in primary visual cortex seems to be largely independent of stimulus contrast ([1–4], at least for large stimuli [5]), shape processing is invariant to rotations [6], and specifically object recognition seems invariant to many different image transformations, including contrast, rotation, scaling, and illumination [7–9]. The visual system derives invariance with seemingly no effort even in natural scenes, with lots of clutter and possible occlusions [10, 11] and despite often dramatic variations in appearance [12].

Over the years, a standard model for the early stages of visual processing has emerged, that consists of a bank of linear filters tuned to image properties like spatial frequency [13, 14] and orientation [15] followed by divisive gain control [16–21]. This theory can successfully explain the independence of orientation tuning and contrast [1] but so far, it is unclear if it can also account for higher levels of visual processing and for more complex invariances.

One challenge here is the difficulty of generating and manipulating stimuli with naturalistic variability in appearance. Modern generative image models based on artificial neural networks, such as Generative Adversarial Networks (GANs, [22]), can generate impressively realistic images (see examples in [23, 24]) and human visual performance is highly sensitive to image manipulations guided by these models [25].

Here, we ask human observers to detect edges that were embedded in artificial, natural looking images generated by a GAN [26]. It is well known that edges in natural images have a considerable degree of appearance variation [12] and the embedding method by Zhu et al. [26] is expected to (at least partly) reproduce this variability.

Artificial neural networks have recently attracted interest in the vision science community because they appear to successfully perform object recognition in photographic images [27–29]. Furthermore, artificial neural networks have been argued to make similar errors as humans in tasks that involve intermediate visual representations [30] and to develop internal representations that seem to resemble those within the primate ventral stream [31, 32]. We therefore compare our observer's detection performance to a number of variations of the standard model as well as an artificial neural network model.

Materials and methods

GAN training

We trained a Wasserstein-GAN [35] on the 60 000 32×32 images contained in the CIFAR10 dataset [36] using gradient penalty as proposed by Gulrajani et al, [37]. In short, a GAN consists of a generator network G that maps a latent vector $\boldsymbol{z}$ to image space and a discriminator network D that takes an image as input and predicts whether that image is a real image from the training dataset or an image that was generated by mapping a latent vector through the generator network (see Figure 1). The generator network and the discriminator network were trained in alternation using stochastic gradient descent. Specifically, training alternated between 5 updates of the

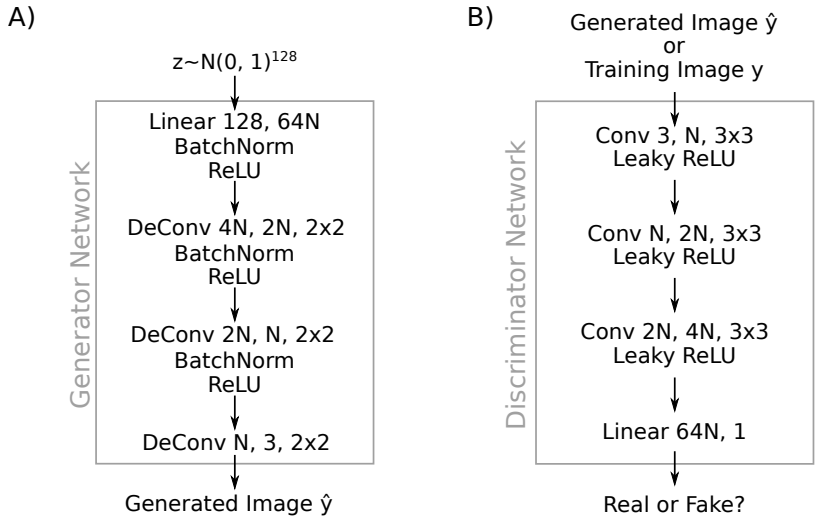

Fig 1. Architecture of the generative adversarial network. (A) Architecture of the generator network. Information flow is from top to bottom, network layers are: "Linear k, m"; Affine transformation from k features to m features, "Conv $k, m, n \times n$"; Convolutional layer from k channels to m channels using a kernel size of $n \times n$, "DeConv $k, m, n \times n$"; like convolution but up-sampling before the convolution to increase spatial resolution and image size, "BatchNorm"; Batch normalization [33], "ReLU"; rectified linear unit "$\mathrm{ReLU}(x) = \max(0, x)$" [34]. The generator network maps a sample z from an isotropic 128 dimensional Gaussian to a 32×32 pixel color image. (B) Architecture of the discriminator network. Architecture component not used in (A) is: "Leaky ReLU" [29]. The discriminator network receives as input either an image $\hat{y}$ generated by the generator network or a real training image y, and it decides if the input image is real or not.

discriminator network and one update of the generator network. Updates of the discriminator network were chosen to minimize the loss

$$\mathbb{E}_{\boldsymbol{z}}[D(G(\boldsymbol{z}))] - \mathbb{E}_{\boldsymbol{y}}[D(\boldsymbol{y})] + \lambda \|\nabla_{\boldsymbol{y}} D(\tilde{\boldsymbol{y}})\|,$$

and updates of the generator were chosen to maximize this loss. Here, the first term quantifies the false alarm rate of the discriminator (i.e. the likelihood that the discriminator D classifies a fake image $G(\boldsymbol{z})$ as real), the second term quantifies the hit rate of the discriminator and the third term is a penalty term to encourage the discriminator to be 1-Lipschitz (a stronger form of continuity). In accordance with Gulrajani et al, [37], we set $\lambda = 10$ for discriminator updates and $\lambda = 0$ for generator updates. In this equation, $\tilde{\boldsymbol{y}}$ denotes a random location between $\hat{\boldsymbol{y}} = G(\boldsymbol{z})$ and the training image $\boldsymbol{y}$. Networks with different numbers of hidden states (parameter N in Figure 1) were trained for 200 000 update cycles using an ADAM optimizer [38] with learning rate 10^{-4} and $\beta_0 = 0$, $\beta_1 = 0.9$. Specifically, we trained networks with $N = 40, 50, 60, 64, 70, 80, 90, 128$ (see Figure 1). Wasserstein-1 error [35] on a validation set (the CIFAR10 test dataset) was lowest with $N = 90$ in agreement with visual inspection of sample quality, so we chose a network with $N = 90$ for all remaining analyses.

Observers

Three observers with normal or corrected to normal vision gave their informed consent to participate in the study. All observers were naive to the purpose of this study. Observers o1 and o2 had some experience with psychophysical experimentation, while o3 did not. The study procedures were approved by the Ethics Committee of York University, Toronto, ON and adhered to the principles outlined in the original Declaration of Helsinki.

Stimuli

Stimuli were 64×64 pixel grayscale images presented at 8bit resolution on a Sony Triniton Multiscan G520 CRT monitor in a dimly illuminated room. The stimulus images were initially constructed at 32×32 pixel resolution and were subsequently upsampled for display using bilinear interpolation. The monitor was carefully linearized using a Minolta LS-100 photometer. Maximum stimulus luminance was 106.9 cd/m^2, minimum stimulus luminance was 1.39 cd/m^2. The background was set to medium gray (54.1 cd/m^2). At the viewing distance of 56 cm, the stimulus square subtended approximately 1 degree visual angle. One pixel subtended approximately 0.031 degree of visual angle. Two different types of stimuli were used, arc segments embedded in naturalistic images and model optimized stimuli to target different performance levels.

Embedding arc segments in naturalistic images

Stimuli in the main experiment consisted of arc segments that were embedded into naturalistic images by conditionally sampling from a GAN. In the 32×32 pixel image array, each arc segment connected one of four pairs of points. These were $(13, 20) - (20, 20)$ (top location), $(13, 20) - (13, 13)$ (left location), $(13, 13) - (20, 13)$ (bottom location), or $(20, 13) - (20, 20)$ (right location), where coordinates are in pixels from the bottom left corner of the image (see Figure 3). Furthermore, arc segments had one of eight curvatures $0, \pm 0.096, \pm 0.177, \pm 0.231, 0.25$, corresponding to orientations of $\pm n\pi/8,\ n = 0, \ldots, 7$ at their endpoints. These arcs were then rendered to a 32×32 pixel image using a background intensity of 0 and a foreground intensity of 1. The

differentiable histogram of oriented gradients (HOG) by Zhu et al, [26] was evaluated on each one of them.

In order to sample an image from the GAN that contained the respective arc segment, we follow Zhu et al, [26] and minimized the objective function

$$E(\boldsymbol{z}) = \sum_{xy\theta} \gamma_{xy} [H(G(z))_{xy\theta} - h_{xy\theta}]^2, \tag{1}$$

where the sum goes over all pixel locations x, y and 8 different equally spaced orientations θ, $H : \mathbb{R}^{32\times32} \to \mathbb{R}^{32\times32\times8}$ is the differentiable HOG operator, $G : \mathbb{R}^{128} \to \mathbb{R}^{32\times32}$ is the generator network of the GAN and $h = (h_{xy\theta}) \in \mathbb{R}^{32\times32\times8}$ is the differentiable HOG operator evaluated on the target arc image. Thus, the minimized error function was simply the squared difference between local histograms of oriented gradients in the generated image and in the target binary arc image. To make sure that only errors in the vicinity of the embedded arc could drive image generation, we weighted the error terms by weights γ_{xy} that were calculated by convolving the binary arc image with a kernel of the form

$$k = \frac{1}{10} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

This way error terms arising from pixels that were more than 2 pixels away from the embedded arc received a weight of zero and did not drive the search for an appropriate latent vector $\boldsymbol{z}_{\text{arc}}$. To find a latent vector $\boldsymbol{z}_{\text{arc}}$ that corresponded to an image that contained the respective arc segment, we started with a random latent vector $\boldsymbol{z}_0$ drawn from an isotropic Gaussian distribution and then used gradient descent on $E(\boldsymbol{z})$ as defined in Equation (1). While Zhu et al [26] aimed for real time performance and therefore trained a neural network to optimize equation (1) in one shot, we did not require real time performance and decided to simply minimize equation (1) using gradient descent. In Supplemental S1 Analysis, we show that the non-target areas of the constructed stimuli were statistically very similar to natural images from the CIFAR10 database.

We constructed a total of 134 400 stimuli (4 locations, 8 curvatures, 4200 exemplars, see Figure 2 for examples).

Optimizing stimuli to target different performance levels

In a second experiment, we tested observers' performance on stimuli that were optimized with respect to a candidate model (see Section "Model fitting" and Supplemental S2 Method for details of these models). Specifically, let $\mathcal{M} : \mathbb{R}^{32\times32} \to \mathbb{R}^4$ be a candidate model that maps a 32×32 pixel image I to a vector of probabilities $\mathcal{M}(I) =: \boldsymbol{p} = (p_{\text{left}}, p_{\text{right}}, p_{\text{top}}, p_{\text{bottom}})$. To construct a stimulus for which the model predicts a correct left-response with probability q, we minimized the error function

$$f_{\text{left}}(I) = [p_{\text{left}}(I) - q]^2 + \sum_{i \neq \text{left}} [p_i(I) - (1-q)/3]^2,$$

and similar for right, top, and bottom responses. For each location and for target performances of $q = 25\%, 50\%, 75\%, 95\%$, we constructed 40 stimuli using gradient descent on the image pixels.

Optimized stimuli were only computed for the four best models (see Section Prediction performance can not discriminate between different models).

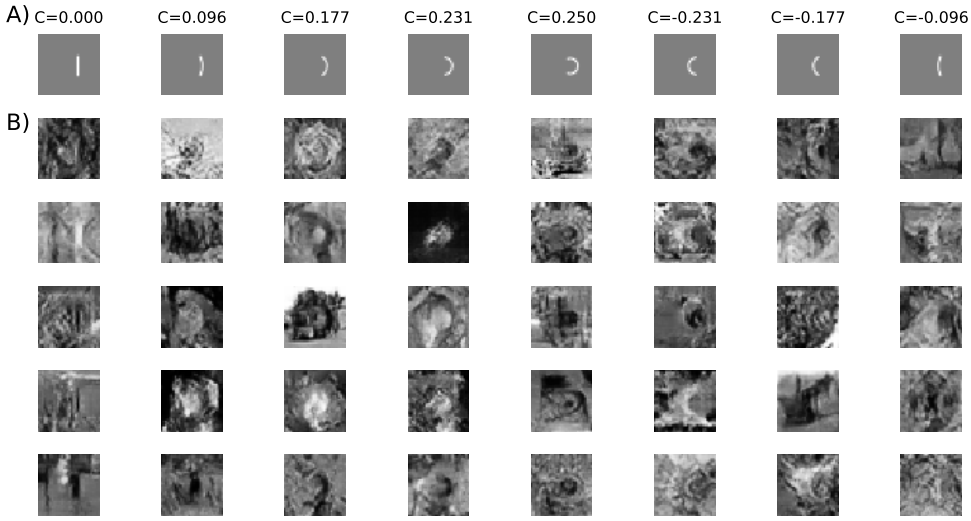

Fig 2. Embedding an arc segment in a natural image. (A) Arc stimuli embedded in the images. Only the "right" location is shown. The corresponding curvature values are shown above the images. (B) Natural images with embedded arc segments. All stimuli in one column correspond to the same arc segment.

Procedure

We performed two experiments that were the same in everything but the stimuli presented (see Section "Stimuli"). All other procedures were the same between both experiments.

Figure 3 shows the layout of a single trial. Each trial started with an 800ms fixation interval. During the fixation interval, a marker consisting of four dots (size 1 pixel) forming the corners of a 10×10 pixel (0.31×0.31 degree visual angle) square was visible on the screen and observers were instructed to fixate at approximately the center of the square. After the fixation interval, the target stimulus appeared on the screen for 100ms. On all experimental trials, the target stimulus was followed by a second interval showing only the fixation marker and observers had one second to indicate where they had seen the arc segment by pressing the corresponding arrow key on a computer keyboard. After

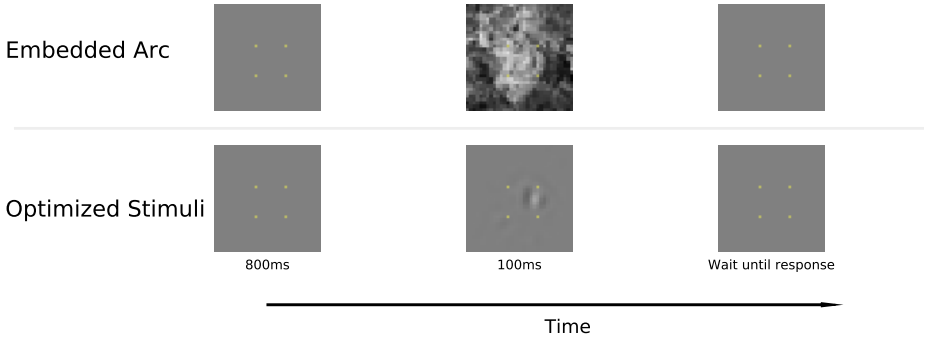

Fig 3. Example experimental display. Top row: Trial sequence in the embedded arc experiment. Bottom row: Trial sequence in the optimized stimuli experiment with an example stimulus optimized for the Feature Normalization model for observer o1. Time increases from left to right. In both cases, the correct target response would be "right".

the response, observers always received feedback: After a correct response, the fixation marker jumped up for a duration of 9 frames (150ms), after an incorrect response, the fixation marker jiggled randomly for a duration of 9 frames (150ms). If observers did not respond within one second, the message "no response" was flashed on the screen and the trial was marked as *invalid response* and was discarded from further analysis.

At the beginning of the experiment, each observer performed 20 trials of training, in which they saw the isolated arc segments and had an opportunity to familiarize themselves with the response format. After that observers performed two sessions of 400 trials each for training with the embedded arc stimuli. In the first training session, observers performed the task on unmasked stimuli. In the second training session, observers performed the task on masked stimuli; after 100ms, a second, unconditioned sample from the GAN appeared on the screen and remained visible for up to one second or until the observer made a response.

After the training sessions, each observer performed 8 sessions with 400 trials each with embedded arc segment stimuli, without masking. During these 8 sessions, performance did not change considerably as confirmed by plotting a rolling 20 trial response accuracy. We then performed two additional control experiments. In the first control experiment, the stimuli from one of the sessions from the main experiment were shown a second time. This allowed us to determine how consistent observers responded with their own responses [39]. In the second control experiment, we used stimuli that were optimized to target different performance levels for model observers. This second control experiment used the stimuli described in Section "Optimizing stimuli to target different performance levels". In the second control experiment, observers saw 10 trials for each combination of arc-location, target performance, and model.

Model fitting

We evaluated 7 different models for their ability to predict human responses on a trial-by-trial level. The first 6 models were based on a bank of orientated filters with parameters that mimicked spatial tuning properties of neurons in primary visual cortex [40]. These models differed between each others in two aspects. Firstly, models were either based on the filter outputs directly, or they were based on the energy of the filter outputs [41, 42] to mimic complex cell responses in primary visual cortex. Secondly, models were either directly using these filter outputs, or they applied spatially local or orientation tuned gain control to the filter outputs. The last model was a deep convolutional neural network [43], allowing for highly non-linear features. All seven models derive a set of (linear or non-linear) features and human responses were predicted by linearly decoding these features. Details of the seven models can be found in Supplemental S2 Method.

Before fitting the models to the observers' data, we split the data into a training set (80% of all trials, between 3200 and 3500 trials), a validation set (10% of all trials, between 400 and 440 trials), and a test set (10% of all trials, between 400 and 440 trials).

The biologically inspired models are ultimately logistic regression models with elaborate input features. As such, they could easily be fit using deterministic gradient based methods such as gradient descent or Fisher scoring iterations. This is however not the case for the deep convolutional neural network model. Such models are typically trained using stochastic gradient descent; the training data is split into a number of smaller "mini-batches" and gradient descent is performed using gradients of the likelihood of these small subsets of the data. This allows stochastic gradient descent to make more efficient use of the compute power and in addition, the stochasticity arising from taking the gradient on random subsets of the data allows the model to escape from local minima. To ensure that differences between models did not result from the

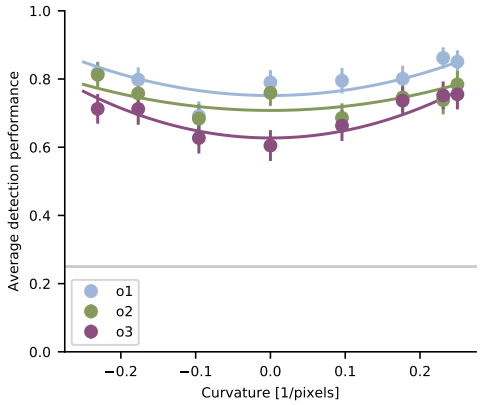

Fig 4. Performance for different curvatures. Mean fraction of correct responses is shown for different observers (color coded). Solid parabolas are least squares fits of the model $p(\text{correct}) \approx a + bC^2$. Error bars indicate 95%-confidence intervals determined from 1000 bootstrap samples. The horizontal gray line marks chance performance.

training method used, we decided to train all models using stochastic gradient descent on the negative log-likelihood of the training data. We used a batch size of 16 trials and a learning rate of 0.001 (except for the Energy model, for which a learning rate of 0.1 was used). Models were trained for up to 200 epochs, where one epoch consisted of one pass over all training data. We used early stopping to regularize the models (e.g. [44], chapter 7.8): After every epoch, we evaluated the negative log-likelihood of the validation data set. If this validation likelihood did not improve over 10 successive epochs, model training was interrupted early.

Results

Performance varies weakly with arc curvature

Observers' performance did not vary much with the curvature of the embedded arc. As shown in Figure 4, all observers performed at a level of approximately 75% correct (o1: $80.1 \pm 0.67\%$, o2: $74.6 \pm 0.73\%$, o3: $69.5 \pm 0.81\%$). For two out of three observers, more acute arcs were slightly easier to detect than straight arcs (o1: correlation between performance and squared curvature $r = 0.65$, $p < 0.08$ Wald test, o2: $r = 0.72$, $p < 0.05$, o3: $r = 0.90$, $p < 0.005$). However, these variations only covered a relatively small performance range.

Prediction performance can not discriminate between different models

We compared a number of different models for the trial-by-trial behaviour of observers in the arc detection task. These models consisted of a logistic regression applied to abstractions of features that are known to be computed at different levels by the visual system. As a first step of evaluating these models, we assessed how well they predicted the trial-by-trial responses of individual observers on a held out set of previously unseen trials from the same experiment.

For the first model, the features consisted of a bank of linear filters with tuning parameters that resembled those of simple cells in macaque area V1 (see Section "Model fitting" and Supplemental S2 Method). These features achieved an average prediction

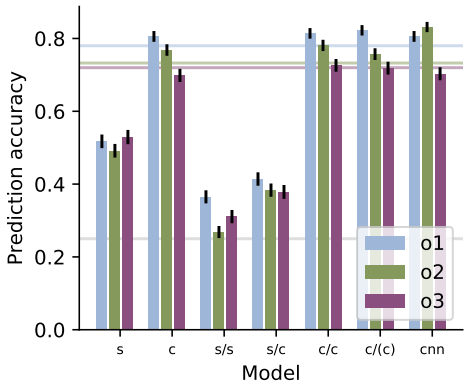

Fig 5. Predictive performance of evaluated models. (A) Prediction accuracy for different models on a held out test-set of trials. Error bars indicate standard errors on the test-set. Models on the X-axis correspond to the features used by the model *s*: linear filter outputs, *c*: "Energy" model, oriented energy features, *s/s*: linear filters normalized by locally pooled filter outputs, *s/c*: linear filters normalized by locally pooled energy, *c/c*: "Spatial Normalization" model, energy normalized by locally pooled energy, *c/(c)*: "Feature Normalization" model, energy normalized by global, oriented energy, *cnn*: "CNN" model, a 5-layer convolutional neural network. The light gray line indicates chance performance, the colored horizontal lines indicate double-pass consistency. (B) Like (A) but for cross entropy.

accuracy of 51.2%, which was significantly above chance ($p < 10^{-10}$, binomial test, see Figure 5A). However, the prediction accuracy of this linear model was still much below the internal consistency of the observers as determined by running one of the experimental sessions a second time.

The second model replaced the linear filters by energy detectors with the same tuning parameters, resembling complex cells in area V1. This model—which we will in the following refer to as the "Energy" model—achieved an average prediction accuracy of 75.7%, similar to the observers' internal consistency.

We furthermore explored a number of models that applied different kinds of normalization to either the linear filters or the energy filters. In general, these models used either a linear filter bank or a bank of energy detectors and normalized their responses by responses that were either pooled locally in space (*x/y* models in Figure 5) or in feature space (*c/(c)* model in Figure 5). These models generally replicated the observations made with the first two models: Normalized energy features achieved performances that resembled the internal consistency of the observers, while normalized linear features did not. Furthermore, we found that models based on local energy with normalization by locally pooled energy (the *c/c* model in Figure 5, hereafter referred to as the "Spatial Normalization" model) or by energy pooled in feature space (the *c/(c)* model in Figure 5, hereafter referred to as the "Feature Normalization" model) performed as good or slightly better than the simple energy model (o2: cnn - c = 6.3%, $p < .0005$, permutation test with 2000 samples, o3: c/c - c = 2.7%, $p < .0025$, all other tests n.s.). Yet, models based on normalized linear features performed worse than the model based on unnormalized linear features (accuracy differences > 10%, $p < .0005$ permutation test with 2000 samples).

We also explored a model with hierarchically constructed, highly non-linear features. This model was implemented as a 5-layer convolutional neural network and it achieved performances that were similar to the models based on energy detectors.

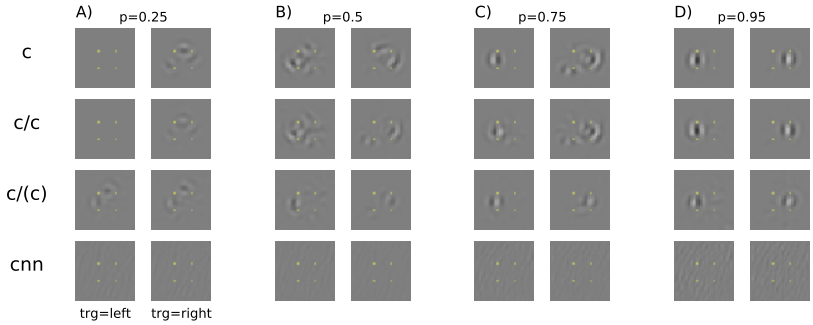

Fig 6. Optimized stimuli to target different performances of observer o1. (A) Stimuli targeting a performance of 25% correct responses. For reference, the target marker is shown as an overlay. Stimuli in the left column require a "left" response, stimuli in the right column require a "right" response. Different rows correspond to different models (*c*: Energy model, *c/c*: Spatial Normalization model, *c/(c)*: Feature Normalization model, *cnn*: CNN model). (B) Stimuli targeting a performance of 50% correct responses. Otherwise like (A). (C) Stimuli targeting a performance of 75% correct responses. Otherwise like (A). (D) Stimuli targeting a performance of 95% correct responses. Otherwise like (A).

Targeted stimuli reveal advantage of standard model

In order to discriminate between the different candidate models, we constructed artificial stimuli that directly targeted specific performance levels for each of the top models from Figure 5 (Energy, Spatial Normalization, Feature Normalization, CNN). Examples of these stimuli for observer o1 are shown in Figure 6 (stimuli for other observers were similar).

As the optimized stimuli in Figure 6 target higher performance levels, we find that structure in the images in the first three rows becomes more focused in the target region (i.e. left or right in Figure 6). Furthermore, images that target high performance levels (Figure 6D) have higher contrast than images that target chance performance (Figure 6A) for the models shown in the first three rows. While the first three rows of Figure 6 show stimuli for models that are—to some extent—inspired by knowledge about the early visual system, the model in the fourth row of Figure 6 is a convolutional neural network that has been optimized to correctly predict responses in the main experiment, without necessarily mimicking known properties of the visual system. Optimized stimuli for this model look quite different from stimuli for the remaining three models. In general, contrast for these images was low and did not change with the target performance level.

When observers repeated the experiment with these optimized stimuli, their performance was indeed correlated to the performances predicted by the model for the first three models (average correlation for Energy model $r = 0.81$, Spatial Normalization model $r = 0.82$, Feature Normalization model $r = 0.89$, see Figure 7A-C) but not for the CNN model ($r = -0.18$ for CNN model). However, humans tended to perform better than predicted by models based on oriented stimulus energy consistent with the idea that humans have access to additional, potentially more high-level image features than these simple models (see Figure 7A-C). On the contrary, human accuracy hardly correlated with the performance predicted by the convolutional neural network model ($r = -0.18$). Even for stimuli where the neural network model predicted near perfect accuracy (p=0.95), human accuracy was still close to chance (o1: 36.8±3.8% human accuracy ± s.e.m., o2: 25.0±3.4%, o3: 30.0±3.6%).

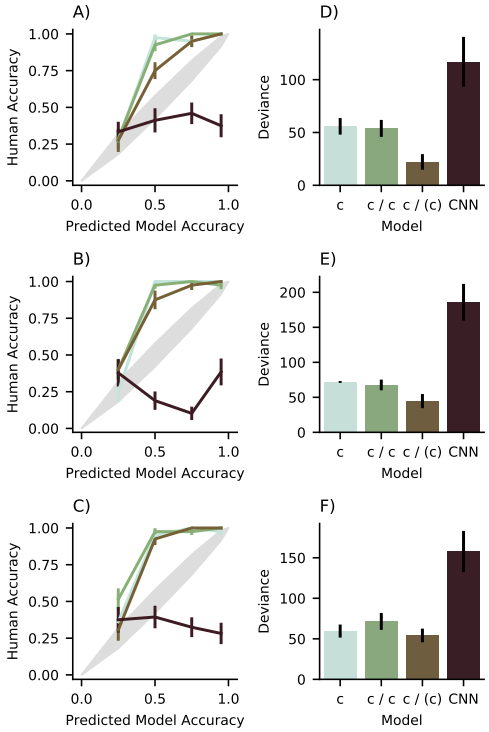

Fig 7. Human performance on optimized stimuli (A) Performance of observer o1 for stimuli that target different accuracies in the models. Error bars indicate standard error of mean. The diagonal gray line indicates equal performance between predicted and human (width of the line is ±s.e.m.). (B) and (C) same as (A) for observers o2 and o3. (D) Deviance between predicted and observed performance for stimuli optimized for different models of observer o1's trial by trial behaviour. (E) and (F) same as (D) for observers o2 and o3.

Table 1. Deviances relative to the predicted accuracies. Abbreviations are d.f.=Degrees of freedom, s.e.=standard error. Standard errors were determined using 1000 bootstrap samples.

Observer	Model	Deviance	d.f.	s.e.
o1	Energy	55.84	153	2.69
	Spatial Normalization	53.87	155	2.84
	Feature Normalization	22.13	160	2.72
	CNN	116.92	150	2.69
o2	Energy	71.98	160	2.70
	Spatial Normalization	67.69	160	2.53
	Feature Normalization	44.58	160	2.76
	CNN	185.81	152	2.72
o3	Energy	59.56	156	2.79
	Spatial Normalization	71.48	157	2.59
	Feature Normalization	54.17	155	2.72
	CNN	157.68	141	2.89

We quantified the differences between the different models by calculating the deviance between observed and predicted accuracy for each model. Deviance is a generalization of the well known sum-of-squares error to a setting with binomial responses [45]. Figure 7D-F shows deviances for four different models. For observer o1 (Figure 7D), the deviance between observed and predicted accuracy was lowest for the Feature Normalization model where the normalising signal was pooled over different orientations (see Table 1) and it was much larger for any of the other models (Feature Normalization vs any other model, $|t| > 2.8$, $p < 0.01$).

For the other observers (Figure 7B,E and C,F), a similar pattern emerged: While the predicted accuracies for the CNN model were largely unrelated to the observed human accuracy, the other models generated stimuli for which the predicted accuracy increased with human accuracy. However, and similar to observer o1, observers o2 and o3 showed higher accuracy than expected from the respective models. Looking at deviances, we found a similar pattern as well (see Table 1). For both observers, the CNN model was exceptionally bad (o2: $D = 185.81 \pm 2.72$, $|t| > 4.5$, $p < 10^{-5}$ comparing to other models, o3: $D = 157.68 \pm 2.89$, $|t| > 3.5$, $p < 0.001$ comparing to other models). Furthermore, the Feature Normalization model was better or equally good than the other biologically inspired models (o2, t-test Feature Normalization vs Energy, $t(318) = -2.96, p < 0.01$, t-test Feature Normalization vs Spatial Normalization, $t(318) = -1.92, p < 0.05$, o3, t-test Feature Normalization vs Energy, $t(309) = 0.46$, *n.s.*, Feature Normalization vs Spatial Normalization, $t(308) = 1.34$, *n.s.*).

This suggests that overall a model in which the output of the Feature Normalization model is used to derive a decision provides the best explanation for the edge localization task considered here. This model consists of energy filters with similar orientation tuning as complex cells in primary visual cortex, followed by biologically realistic gain control pooled over multiple different orientations.

Feature weighting for the Feature Normalization model

In order to understand how features from different image locations contributed to the observers' decisions, we visualized the readout weights of the Feature Normalization model. The readout weights map from 4 orientations at each image location to one of four different responses. We decided to visualize the readout weights for each combination of orientation and response separately (see Figure 8). Thus, in order to derive a decision signal for a "left" response, the model would match the four maps in the first row of Figure 8 and sum up the corresponding responses.

Figure 8 visualizes the weights for one observer (observer o2, other observers had similar patterns but they were more noisy and are shown in Supplemental S3 Figure). Three things are worth pointing out. (i) We found that for the left and right responses vertical orientations in the area of the target signal were strongly weighted and for the top and bottom responses, horizontal orientations in the area of the target signal were strongly weighted. In the following, we will refer to this weight pattern as ρ_1 (see Figure 9A). These weights correspond to signals that are aligned with a straight line interpolation between the start and end points of the respective arc. (ii) The model negatively weighted signals that corresponded to structure that was orthogonal to this straight line interpolation, but only towards the inside of the square formed by the four target markers (ρ_2, see Figure 9B). (iii) We find some weight associated with oblique directions (ρ_3, see Figure 9C). Specifically, locations in which curved arcs would connect to the corresponding target markers were positively weighted to arrive at a response.

In order to test the relevance of this visible structure in the readout weights, we created simplified models in which either of the above weight patterns were approximated by Gaussian blobs and the others were set to 0 (see Supplemental S4 Method for details). The predictive performance of these simplified models (and their

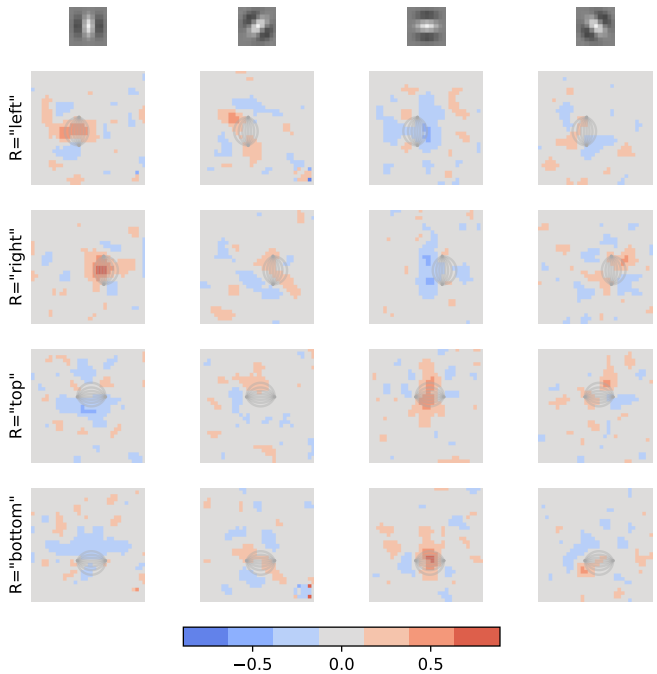

Fig 8. Readout weights for Feature Normalization model. Each column corresponds to one oriented energy feature, each row corresponds to one possible response. The orientation of the corresponding energy features is given by the small grating symbols above the columns. Color codes the weight with which the respective location contributed to the observer's decision. On each panel, the arcs that would be associated with the corresponding decision are superimposed in light gray.

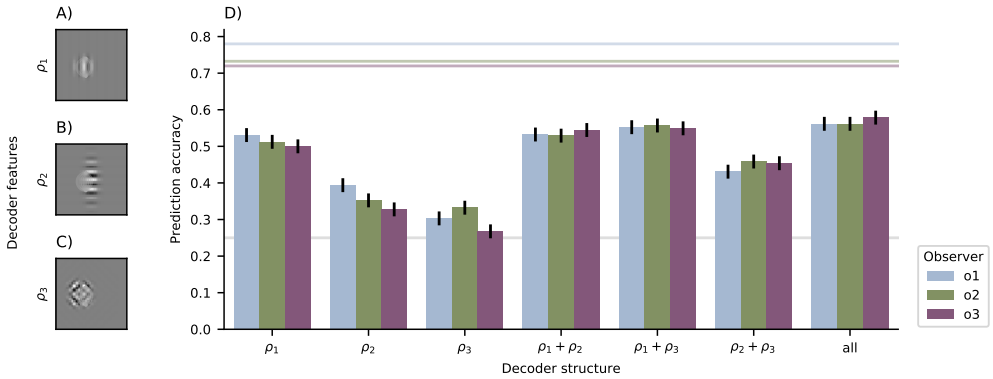

Fig 9. Performance of models with simplified decoder structure. (A) Schematic visualization of the decoder weights for weight pattern ρ_1 for response "left". The weight pattern was just the envelope of the pattern shown here, applied to the orientation channels visualized by the underlying grating. For reference, the superimposed lines indicate where the corresponding target was located. (B) Same as (A) for weight pattern ρ_2. Note that this pattern was associated with negative weights. (C) Same as (A) for weight pattern ρ_3. (D) Accuracy of prediction of human responses for the different weight patterns in isolation and combined. Similar to Figure 5, the horizontal lines indicate change performance (gray) and double pass consistency for the individual observers (colored lines).

Table 2. Performance of models with simplified decoder structure. Reported values are mean±s.e.m. on hold out test dataset.

Decoder structure	Accuracy		
	o1	o2	o3
ρ_1	0.531 ± 0.020	0.512 ± 0.020	0.500 ± 0.020
ρ_2	0.303 ± 0.018	0.332 ± 0.018	0.268 ± 0.017
ρ_3	0.394 ± 0.019	0.352 ± 0.019	0.328 ± 0.018
$\rho_1 + \rho_2$	0.532 ± 0.020	0.529 ± 0.020	0.545 ± 0.020
$\rho_1 + \rho_3$	0.552 ± 0.020	0.557 ± 0.019	0.549 ± 0.020
$\rho_2 + \rho_3$	0.431 ± 0.019	0.458 ± 0.020	0.451 ± 0.020
all	0.562 ± 0.019	0.562 ± 0.019	0.578 ± 0.019

combinations) was then evaluated on the test dataset from experiment 1 (see Figure 9). Weight pattern ρ_1 predicted human responses with an accuracy of about 50% (mean±s.e.m.=53.1±1.95% for observer o1, 51.2±1.96% for observer o2, 50.0±1.96% for observer o3). Combining weight pattern ρ_1 with either of the other two patterns predicted human responses with higher accuracy and combining all three components of the weight pattern predicted human performance even better (see Table 2). However, compared to the results presented in Section "Prediction performance can not discriminate between different models", these numbers are pretty low, suggesting that the detailed pattern of feature weights really matters for a full explanation of behaviour.

Discussion

By embedding edge segments in samples from a generative adversarial network, we were able to construct edge stimuli with considerable variation in appearance. Using these stimuli, we found that human performance at detecting these embedded edge segments is consistent with multiple variations of the standard model of early vision [21], as well as an artificial neural network model with no explicit knowledge about the visual system. We therefore constructed stimuli that, for each one of the models, would target specific levels of performance. When tested on these model specific stimuli, we found that the standard model generally performed better than the artificial neural network model.

On the model specific stimuli, none of the models really describes human behaviour. Although the biologically inspired models result in stimuli for which human accuracy increases with model accuracy, humans actually perform better for these stimuli than expected from the respective models. We believe that this implies that human observers have access to more complex interactions between signals at different locations than any of the models studied here. One class of such interactions might be effects related to surround suppression [46], where the output of a neuron with receptive field in one location is normalized by the output of neurons with neighbouring receptive fields [47, 48]. Our Feature Normalization model contains a normalizing mechanism that would result to some extent in surround suppression, yet recent evidence suggests that realistic surround suppression would likely be considerably more complex [49]. Furthermore, it might be that correlations between neural responses [50] or flexible assignment of neurons to different suppressive pools [51] could play a role. We believe that a detailed examination of the contributions of surround suppression to observers' performance in our experiment is beyond the scope of this paper and decided to restrict ourselves to better understood local gain control operations.

Many studies that aim to identify features that contribute to an observers decision have used white noise ([52–56] see [57] for an overview) or noise with relatively simple correlation structure [58]. The embedding technique used here can be interpreted as

"natural image noise"[1]. The convolutional neural network model fails to generalize to stimuli synthesized for this model. This could be interpreted as a failure to generalize from natural noise to less natural noise, which might seem less critical than a failure to generalize from artificial stimuli to natural conditions. However, looking at Figure 6, the images generated for the convolutional neural network generally have fairly low contrast and even for stimuli that should be recognized with high confidence, the image structures are very weak. Two explanations are possible for this failure to construct targeted stimuli for the convolutional neural network model. Firstly, it is known that convolutional neural networks can be fooled by small manipulations of an image that humans would simply ignore. These small manipulations are known as adversarial examples [59] and they are typically constructed by taking steps in image space in the direction of the networks gradients, similar to the construction of stimuli for experiment 2. It appears that the smoothing properties of our biologically inspired filter bank have made the variants of the standard model more robust to these adversarial-like stimulus examples. In fact, it has been shown that smoothing can—to some extent—reduce vulnerability to adversarially constructed images [60]. Secondly, it might be that the CNN model in our study does not learn the same features that humans use to solve the task. Baker et al [61] find that convolutional neural networks trained on a large database of natural images [62] use different image features than humans for object recognition. More specifically, Geirhos et al [63] report that such convolutional neural network models mostly rely on texture to perform classifications, while humans rely more on object shape. The CNN model here was not trained to perform large scale image recognition, but it was explicitly trained to predict human responses. However, if there was a correlation between the features used by our human observers and some subtle texture properties in the images from the main experiment, the CNN model might be biased to pick up these texture features. When constructing stimuli from the CNN model, these stimuli would only contain the subtle texture properties but not the features used by our human observers, resulting in near chance performance as observed in our control experiment.

Our models all use a single frequency channel (albeit with different orientations), while it is well known that the human visual system has multiple frequency channels [14, 40, 64, 65]. In this respect, our models are clearly a simplification. However, we note that the models don't seem to require this fine frequency resolution and are able to accurately predict human responses despite the limited frequency resolution. We find that adding orientation tuned gain control to the models tends to add to the model's ability to predict human responses, and to improve the model's ability to construct stimuli that target different performance levels. This suggests that gain control might have to some extent complemented the limited frequency resolution of our models.

Sebastian et al [66] report results on target detection in natural images. Their target stimulus was a grating patch of fixed size and orientation at the centre of a natural image patch. They elegantly "manipulated" properties of this background patch by selecting appropriate image patches from a large database of natural images. However, their target pattern was simply superimposed on the background image. Under natural viewing conditions, targets are usually not superimposed on the background, but they are part of the entire image as much as the background itself. Depending on the scene in which a target appears, that target may be partially occluded or be illuminated in different ways. All these factors will alter not only the appearance of the background, but also of the target itself (see [12] for a detailed discussion in the context of edge targets). Our embedding approach allows us to constrain part of the appearance of the target while still maintaining the fact that the target is part of the background scene.

[1] Note however, that in this case the noise also influences the appearance of the target, and is therefore not additive as assumed in previous studies.

Our method of constructing model specific stimuli that target specific performance levels has some resemblance to the construction of maximum differentiation stimuli [67]. Wang & Simoncelli [67] address the problem of comparing two competing models and they suggest constructing two classes of stimuli that each clip the accuracy of one model while maximizing accuracy of another model. Although it would be possible to generalize this procedure to comparisons between n models by clipping the accuracy $n-1$ models and maximizing the accuracy of the remaining one, this approach requires repeated evaluation of every model when constructing each one of the stimulus classes. In addition, clipping accuracy for $n-1$ related models will result in complex constraints on the generated images that can be computationally costly to satisfy. In this study, we compared 4 different models, with three of them being closely related to each others. We take an alternative approach to Wang & Simoncelli [67] by requiring models to construct stimuli that target a given level of human accuracy. This requires the models to also predict human responses at intermediate accuracy levels correctly while at the same time being less computationally demanding.

Conclusion

In conclusion, we have provided evidence that the standard model of early vision, combined with a flexible linear readout mechanism, is able to generalize to fairly complex target detection tasks in naturalistic stimuli, while convolutional neural networks provide less general descriptions of visual behaviour.

Code and data for this study will be made available upon publication at doi:10.5281/zenodo.2535941.

Supporting information

S1 Analysis. Validation of embedded arc stimuli as "natural" Embedding arc segments clearly changes an image's statistics *locally*; otherwise the visual system wouldn't be able to recognize that arc segment. In order to verify that the embedded arc stimuli only affected the image's statistics locally, but left other parts of the image untouched, we analyzed quadrants of images (see Figure 10A). A large number of statistic signatures of natural images have been described. These include a characteristic scaling law of the images' power spectrum [68] or second order correlations between local wavelet statistics [69, 70]. Although these image properties are well known, they are only moderately meaningful on the 16×16 image quadrants used here. We therefore decided to adopt a different approach: We asked if a 4-layer version of the VGG network [28] could tell apart quadrants from natural images, from unconditioned GAN samples and from images that had an arc segment embedded in another segment, as well as images from a model that matches an image's second order statistics ([71], we used their code in the default configuration, but with 3 instead of 4 levels for the pyramid decomposition, due to the small image size) and images that matched the images power spectrum. Specifically, the VGG network's had four convolutional layers with max-pooling after every second layer, followed by two fully connnected readout layers (in the notation by Simonyan et al [28] the network's architecture was conv3_64, conv3_64, maxpool, conv3_128, conv3_128, maxpool, FC_64, FC_3, softmax). We selected 1824 image quadrants per category and randomly selected 400 of them as validation and test sets respectively. Thus, the remaining training set consisted of 8320 image quadrants. We trained the network for up 1000 epochs with a learning rate of 0.002 and a batch size of 128. We used early stopping to selected the network with lowest cross entropy on the validation dataset (after 302 epochs). We

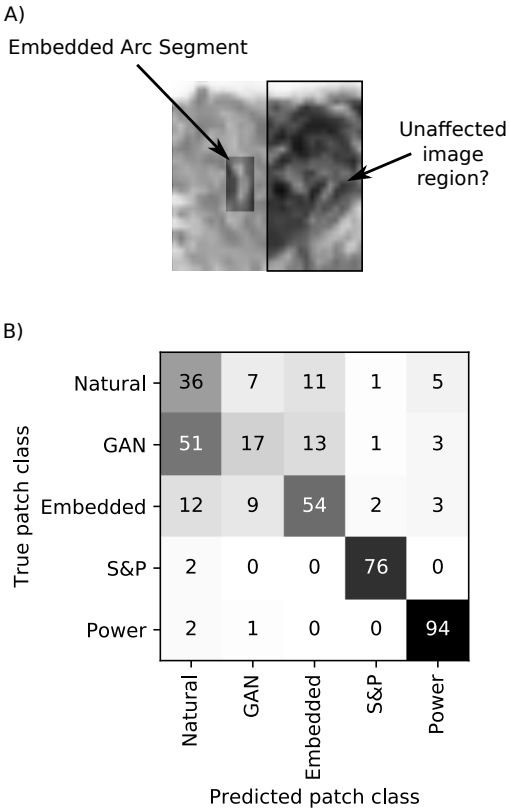

Fig 10. Do images with embedded arc segments have similar statistical properties as natural images? (A) The left side of the image contains an embedded arc segment, affecting the image's statistics. To understand if the effect of this manipulation also affected the rest of the image, we analyzed a quadrant from the supposedly unaffected side of the image (right side in this example). (B) Confusion matrix of the patch classification network. While natural images and GAN samples appear mostly natural to the network, samples from the texture model by Portilla & Simoncelli ([71], S&P) and images with matched power spectrum (Power) can be clearly told apart from natural images. Results for embedded arc images (Embedded) are somewhat in between.

report results for this network on the remaining test set of 400 image patches.

Overall, the network achieved a prediction accuracy of 69.2±2.3% (mean ± s.e.m.), significantly better than chance (20%). The full confusion matrix for the 5 classes reveals additional information (Figure 10B). While the network often confused GAN samples with natural images, it could easily tell apart images with the power spectrum matched to natural images from real natural images. The same was true of samples from the texture model by Portilla & Simoncelli [71]. Images with embedded arc images were somewhere in between; the network was clearly better at telling them apart from natural images than it was for the unconditioned GAN samples, but it confused embedded arc images more often with real natural images than samples from the Portilla & Simoncelli [71]-model or images with matched power spectrum. We therefore conclude that the images used in experiment 1 are reasonably well matched to the statistics of natural images and—more importantly—match those statistics better than alternative image models would.

S2 Method Details of model architectures

Simple cell filter bank The model consisted of a bank of linear, oriented filters. Filters were polar separable in the Fourier domain, such that the frequency response could be written as

$$H(r, \theta) = F(r)G(\theta), \tag{2}$$

where $r = \sqrt{f_x^2 + f_y^2}$ and $\theta = \arctan2(f_x, f_y)$ are the (absolute) frequency and orientation. We fixed

$$F(r) = \exp\left(-32\sqrt{\pi/2}(r-1)^2\right),$$

to cover approximately 1.6 octaves [72] and selected four different orientation response functions $G_n, n = 0, 1, 2, 3$ of the form (circularly wrapped)

$$G_n(\theta) = \exp\Big(-\frac{(\theta - \theta_n)^2}{2\sigma^2}\Big),$$

where $\theta_n = n\pi/4$ and

$$\sigma = 25\pi\Phi^{-1}(0.707),$$

to achieve an orientation bandwidth of 20–30 degrees [40]. Here, $\Phi : \mathbb{R} \to (0, 1)$ is the cumulative distribution function of the standard normal distribution. The filter H was subsequently converted to the spatial domain and only the real (even) part was retained. The resulting filter h_n was then pruned to only contain the central 7×7 coefficients.

All four filters were applied to each image with zero padding at the borders, resulting in a $4 \times 32 \times 32$ array. This array was treated as one long 4096 element vector $\boldsymbol{s}$ and was submitted to a 4-class logistic regression.

Complex cell filter bank The complex cell filterbank model used the same filterbank from equation (2). However, when converting H to the spatial domain the real (even) and imaginary (odd) parts were retained. These were pruned in the same way as for the simple cell filter bank and they were then both applied to the input image (with zero padding). This ultimately resulted in two 4096-dimensional feature vectors $\boldsymbol{\psi}_{\text{even}} = \boldsymbol{s}$ and $\boldsymbol{\psi}_{\text{odd}}$. From these two we constructed a local, oriented energy signal (e.g. [41, 42]) as

$$\boldsymbol{e} = \boldsymbol{\psi}_{\text{even}}^2 + \boldsymbol{\psi}_{\text{odd}}^2, \tag{3}$$

to be submitted to the subsequent 4-class logistic regression.

Simple cell filter bank with local gain control by simple cell outputs This model used the same filterbank model as in equation (2) and calculated the same feature vector $\boldsymbol{s}$ as above. In addition, a normalization signal $\boldsymbol{m}$ was computed by convolving the oriented real parts of the filter outputs $\boldsymbol{s}$ with filters rotated by 90 degree. In other words, if $\boldsymbol{s}_0$ denotes the oriented energy at orientation $n = 0$ and h_2 denotes the filter with orientation $o_2 = \pi/2$, then

$$\boldsymbol{m}_0 = \boldsymbol{s}_0 \star h_2,$$

with $\star$ denoting convolution. We then calculated a normalized vector of the form

$$\frac{\boldsymbol{s}}{1 + \boldsymbol{m}},$$

where the division was performed pointwise and we omitted the orientation index n for simplicity. The resulting 4096-dimensional feature vector was submitted to 4-class logistic regression.

Simple cell filter bank with local gain control by complex cell outputs This model was very similar to the model in the section above about gain normalized simple cell filters except that the normalization was calculated from the oriented energy signal rather than from the simple cell outputs. Thus vectors of the form

$$\boldsymbol{n}_0 = \boldsymbol{e}_0 \star h_2,$$

were used as a normalization signal and the vectors

$$\frac{\boldsymbol{s}}{1+\boldsymbol{n}},$$

were subsequently submitted to 4-class logistic regression.

Complex cell filter bank with local gain control by complex cell outputs This model was almost the same as the model in the previous paragraph except that the vectors

$$\frac{\boldsymbol{e}}{1+\boldsymbol{n}},$$

were submitted to 4-class logistic regression.

Complex cell filter bank with orientation tuned gain control by complex cell outputs We often think of the gain control signal as being tuned both in location and orientation, where similar orientations in similar locations contribute most to the normalization pool. We therefore used a model in which the gain control signal was pooled by a space-orientation separable filter of the form

$$w(x, y, \theta) = \phi(\frac{\theta}{\sigma_o})\phi(\frac{x}{\sigma_x})\phi(\frac{y}{\sigma_x}),$$

with orientation bandwidth $\sigma_o = \pi/2$ and spatial bandwidth $\sigma_x = 1$ pixel [21]. Here, $\phi : \mathbb{R} \to \mathbb{R}$ was the density function of the standard normal distribution. We then used

$$\frac{\boldsymbol{e}}{1+\boldsymbol{w}\star\boldsymbol{e}}$$

in our 4-class logistic regression.

Deep convolutional neural network The deep convolutional neural network constructed non-linear features in 5 convolutional layers. Each convolutional layer had 4 output channels and a learned kernel of size 3×3. The convolution operation was followed by rectification [34] and batch normalization [33]. The resulting non-linear features were then cast into one long vector of 4096 values and were linearly read out using logistic regression as in the other models. In total, the deep convolutional neural network had 17 060 parameters. Thus, fitting this network was really dependent on using effective regularization techniques. Batch normalization is known to have a regularizing effect [33] and we further used early stopping based on prediction error on a separate validation dataset (see for example [44], section 7.8). Although early stopping was used for the other models as well, it was particularly important for the deep convolutional neural network model.

S3 Figure Readout weights for observers o1 and o3

Figure 11 shows readout weights for observers o1 and o3. For both observers, the patterns of readout weights was more noisy than for observer o2. However, we found qualitatively similar structure.

S4 Method Simplified models of readout weights We analyzed three different decoder features and their combinations to understand the readout process for the Feature Normalizationmodel. Specifically, those were ρ_1 horizontal/vertical structure aligned with the straight line version of the target, ρ_2 horizontal/vertical structure orthogonal to the straight line version of the target, ρ_3 oblique structure along the sides of the arcs. To describe these features, we can think of the decoding weights as a template tensor $\mathcal{T} \in \mathbb{R}^{4\times4\times32\times32}$, such that $\mathcal{T}_i$ denotes the templates for response i (i.e. one row in Figure 8). Each of these templates has 4 orientation bands, such that $\mathcal{T}_{ij}$ denotes the template for response i for features from the orientation band j. Note that

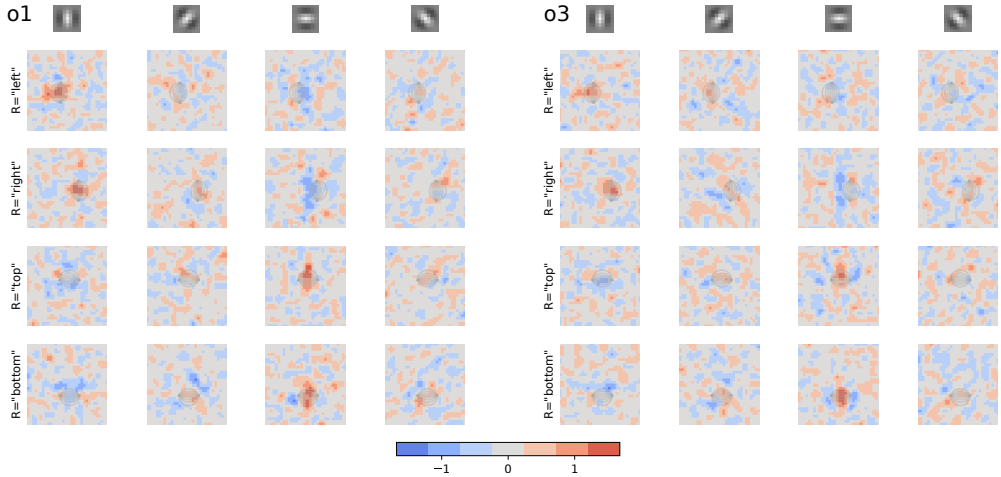

Fig 11. Readout weights for Feature Normalization model for observer o1 (left) and observer o3 (right). Details of the subfigures are the same as in Figure 8.

this template is sensitive to orientations of approximately $\pi j/4$. Components within orientation bands are approximated by Gaussian blobs of the form $u : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}$, where

$$u(\boldsymbol{x}, \sigma) = \exp(-\boldsymbol{x}^T \boldsymbol{x}/\sigma^2).$$

In all cases, the exact readout weights were determined using numerical optimizaton as outlined below, however the results remained qualitatively similar if the readout weights were selected using the starting values of the parameters (see below).

Horizontal/vertical structure aligned with the straight line version of the target For decoding features ρ_1, the template tensor $\mathcal{T}$ was zero except

$$\begin{aligned}
\mathcal{T}_{\text{left},0}(\boldsymbol{x}) &= gu(\boldsymbol{x} - \boldsymbol{m}, \sigma), \\
\mathcal{T}_{\text{right},0}(\boldsymbol{x}) &= gu(\boldsymbol{R}(\pi)(\boldsymbol{x} - \boldsymbol{m}), \sigma), \\
\mathcal{T}_{\text{top},2}(\boldsymbol{x}) &= gu(\boldsymbol{R}(\pi/2)(\boldsymbol{x} - \boldsymbol{m}), \sigma), \\
\mathcal{T}_{\text{bottom},2}(\boldsymbol{x}) &= gu(\boldsymbol{R}(-\pi/2)(\boldsymbol{x} - \boldsymbol{m}), \sigma).
\end{aligned}$$

Here, we denote by $\mathcal{T}_{ij}(\boldsymbol{x})$ the value of the template for response i at orientation band j and location $\boldsymbol{x}$. Furthermore, we used the rotation matrix

$$\boldsymbol{R}(\alpha) := \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}$$

The parameters $\boldsymbol{m}$, g, and σ were estimated by minimizing the squared difference between $\mathcal{T}$ and the observed decoding weights. The minimization was done using simplex search [73] starting at $\boldsymbol{m}_0^T = (16.5, 12)$, $\sigma_0 = 3$, and g_0 equal to the maximum of the original decoding weights.

Horizontal/vertical structure orthogonal to the straight line version of the target For decoding features ρ_2, the template tensor $\mathcal{T}$ was zero except

$$\begin{aligned}
\mathcal{T}_{\text{left},2} &= gu(\boldsymbol{B}(0)(\boldsymbol{x} - \boldsymbol{m}), 1), \\
\mathcal{T}_{\text{right},2} &= gu(\boldsymbol{B}(\pi)(\boldsymbol{x} - \boldsymbol{m}), 1), \\
\mathcal{T}_{\text{top},0} &= gu(\boldsymbol{B}(\pi/2)(\boldsymbol{x} - \boldsymbol{m}), 1), \\
\mathcal{T}_{\text{bottom},0} &= gu(\boldsymbol{B}(\pi/2)(\boldsymbol{x} - \boldsymbol{m}), 1),
\end{aligned}$$

with the rotation matrix

$$\boldsymbol{B}(\alpha) := \boldsymbol{R}(\alpha)\,\mathrm{diag}(1/\sigma_x, 1/\sigma_y).$$

The parameters $\boldsymbol{m}$, g, σ_x, and σ_y were estimated in the same way as in the previous section, but starting from $\boldsymbol{m}_0^T = (16.5, 13)$, $\sigma_{x,0} = 3$, $\sigma_{y,0} = 5$ and g_0 set to the minimum of the original decoding weights.

Oblique structure along the sides of the arcs For decoding features ρ_3, the template tensor $\mathcal{T}$ was

$$\mathcal{T}_{\text{left},1} = g\big(u(\boldsymbol{x} - \boldsymbol{c} + \boldsymbol{d}, \nu) + u(\boldsymbol{x} - \boldsymbol{c} - \boldsymbol{d}, \nu)\big),$$
$$\mathcal{T}_{\text{left},3} = g\big(u(\boldsymbol{x} - \boldsymbol{c} + \tilde{\boldsymbol{d}}, \nu) + u(\boldsymbol{x} - \boldsymbol{c} - \tilde{\boldsymbol{d}}, \nu)\big),$$

with analogous expressions for other orientations and responses. Here, the free parameters are $\boldsymbol{c}$, $\boldsymbol{d}$, g, and ν and for $\boldsymbol{d} = (d_1, d_2)^T$ we defined $\tilde{\boldsymbol{d}} := (d_2, d_1)^T$. Optimization in this case started with $\boldsymbol{c}_0 = (16.5, 12)^T$, $\boldsymbol{d}_0 = (4, 3)^T$, $\nu_0 = 2$, and g_0 was set to the maximum of the observed oblique feature maps.

Acknowledgments

The author would like to thank Richard F. Murray for generously lending out several pieces of lab equipment and for feedback on an earlier version of this manuscript. This work was supported by the York University Bridging Fund.

References

1. Finn IM, Priebe NJ, Ferster D. The Emergence of Contrast-Invariant Orientation Tuning in Simple Cells of Cat Visual Cortex. Neuron. 2007;54:137–152.

2. Nowak LG, Barone P. Contrast Adaptation Contributes to Contrast-Invariance of Orientation Tuning of Primate V1 Cells. PLoS one. 2009;4(3):e4781.

3. Skottun BC, Bradley A, Sclar G, Ohzawa I, Freeman RD. The Effects of Contrast on Visual Orientation and Spatial Frequency Discrimination: A Comparison of Single Cells and Behavior. Journal of Neurophysiology. 1987;57(3):773–786.

4. Bowne SF. Contrast Discrimination Cannot Explain Spatial Frequency, Orientation or Temporal Frequency Discrimination. Vision Research. 1990;30(3):449–461.

5. Mareschal I, Shapley RM. Effects of contrast and size on orientation discrimination. Vision Research. 2004;44:57–67.

6. Blais C, Arguin M, Marleau I. Orientation invariance in visual shape perception. Journal of Vision. 2009;9(2):14, 1–23.

7. Gauthier I, Tarr MJ. Visual Object Recognition: Do We (Finally) Know More Now Than We Did? Annual Review of Vision Science. 2016;2:377–396.

8. Pinto N, Cox DD, DiCarlo Jj. Why is Real-World Visual Object Recognition Hard? Plos Computational Biology. 2007;.

9. DiCarlo JJ, Zoccolan D, Rust NC. How Does the Brain Solve Visual Object Recognition? Neuron. 2012;73:415–434.

10. Balboa RM, Tyler CW, Grzywacz NM. Occlusions contribute to scaling in natural images. Vision Research. 2001;41:955–964.

11. DiMattina C, Fox SA, Lewicki MS. Detection natural occlusion boundaries using local cues. Journal of Vision. 2012;12(13):1–21.

12. Elder JH, Zucker SW. Local scale control for edge detection and blur estimation. IEEE Transactions on pattern analysis and machine intelligence. 1998;20(7):699–716.

13. Campbell FW, Robson JG. Application of fourier analysis to the visibility of gratings. Journal of Physiology. 1968;197(3):551–566.

14. Stromeyer CF, Julesz B. Spatial-Frequency Masking in Vision: Critical Bands and Spread of Masking. Journal of the Optical Society of America. 1972;62(10):1221–1232.

15. Wilson HR, McFarlane DK, Phillips GC. Spatial Frequency Tuning of Orientation Selective Units Estimated by Oblique Masking. Vision Research. 1983;23(9):873–882.

16. Legge GE, Foley JM. Contrast Masking in Human Vision. Journal of the Optical Society of America A, Optics and Image Science. 1980;70(12):1458–1471.

17. Foley JM. Human luminance pattern-vision mechanisms: masking experiments require a new model. Journal of the Optical Society of America A, Optics and Image Science. 1994;11(6):1710–1719.

18. Heeger D. Normalization of cell responses in cat striate cortex. Visual Neuroscience. 1992;9:181–197.

19. Albrecht DG, Geisler WS. Motion selectivity and the contrast-response function of simple cells in the visual cortex. Visual Neuroscience. 1991;7:531–546.

20. Mante V, Frazor RA, Bonin V, Geisler WS, Carandini M. Independence of luminance and contrast in natural scenes and in the early visual system. Nature Neuroscience. 2005;8:1690–1697.

21. Schütt HH, Wichmann FA. An image-computable psychophysical spatial vision model. Journal of Vision. 2017;17(12)(12):1–35.

22. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative Adversarial Nets. In: Gharamani Z, Welling M, Cortex C, Lawrence ND, Weinberger KQ, editors. Advances in Neural Information Processing Systems 27. 2672-2680. Curran Associates, Inc.; 2014.

23. Brock A, Donahue J, Simonyan K. Large Scale GAN Training for High Fidelity Natural Image Synthesis. arXiv preprint. 2018; p. 1809.11096.

24. Miyato T, Kataoka T, Koyama M, Yoshida Y. Spectral Normalization for Generative Adversarial Networks. International Conference on Learning Representations. 2018;.

25. Fruend I, Stalker E. Human Sensitivity to Perturbations Constrained by a Model of the Natural Image Manifold. Journal of Vision. 2018;18(11):20, 1–14.

26. Zhu JY, Krähenbühl P, Shechtman E, Efros AA. Generative Visual Manipulation on the Natural Image Manifold. In: Proceedings of European Conference on Computer Vision (ECCV); 2016.

27. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification with Deep Convolutional Neural Networks. In: F P, Burges C, Bottou L, Weinberger K, editors. Advances in Neural Information Processing Systems. vol. 25; 2012.

28. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In: International Conference on Learning Representations; 2015.

29. He K, Zhang X, Ren S, Sun J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImaImage Classification. In: International Conference on Computer Vision (ICCV); 2015.

30. Kubilius J, Bracci S, Op de Beeck HP. Deep Neural Networks as a Computational Model for Human Shape Sensitivity. Plos Computational Biology. 2016;12(4):e1004896.

31. Khaligh-Razavi SM, Kriegeskorte N. Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation. PLoS Computational Biology. 2014;10(11):e1003915.

32. Yamins DLK, Hong H, Cadieu CF, Solomon EA, Seibert D, DiCarlo JJ. Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proc Natl Acad Sci USA. 2014;111(23):8619–8624.

33. Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Bach F, D B, editors. Proceedings of the 32nd International Conference on Machine Learning. vol. 37 of 448-456; 2015.

34. Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. vol. 15 of JMLR; 2011.

35. Arjovsky M, Chintala S, Bottou L. Wasserstein GAN. In: Proceedings of the 34th International Conference on Machine Learning; 2017.

36. Krizhevsky A. Learning Multiple Layers of Features from Tiny Images. University of Toronto; 2009.

37. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A. Improved Training of Wasserstein GANs; 2017. arXiv:1704.00028.

38. Kingma DP, Ba L Jimmy. Adam: A Method for Stochastic Optimization. In: International Conference on Learning Representations; 2015.

39. Neri P, Levi D. Receptive versus perceptive fields from the reverse-correlation viewpoint. Vision Research. 2006;46(16):2465–2474.

40. Ringach DL, Shapley RM, Hawken MJ. Orientation Selectivity in Macaque V1: Diversity and Laminar Dependence. Journal of Neuroscience. 2002;22(13):5639–5651.

41. Morrone MC, Burr DC. Feature Detection in Human Vision: A Phase-Dependent Energy Model. Proceedings of the Royal Society of London Series B: Biological Sciences. 1988;235(1280):221–245.

42. Adelson EH, Bergen JR. Spatiotemporal energy models for the perception of motion. Journal of the Optical Society of America A, Optics and Image Science. 1985;2(2):284–299.

43. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521:436–444.

44. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge, MA: MIT Press; 2016.

45. Dobson AJ, Barnett AG. Introduction to Generalized Linear Models. 3rd ed. Boca Raton, FL9: Chapman and Hall; 2008.

46. Cavanaugh JR, Bair W, Movshon JA. Nature and Interaction of Signals From the Receptive Field Center and Surround in Macaque V1 Neurons. Journal of Neurophysiology. 2002;88:2530–2546.

47. Carandini M, Heeger DJ. Normalization as a canonical neural computation. Nature Reviews Neuroscience. 2012;13:51–62.

48. Coen-Cagli R, Dayan P, Schwartz O. Cortical Surround Interaction and Perceptual Salience Via Natural Scene Statistics. PLoS Computational Biology. 2012;8(3):e1002405.

49. Coen-Cagli R, Kohn A, Schwartz O. Flexible Gating of Context Modulation During Natural Vision. Nature Neuroscience. 2015;18:1648–1655.

50. Kohn A, Coen-Cagli R, Kanitscheider I, Pouget A. Correlations and Neuronal Population Information. Annual Review of Neuroscience. 2016; p. 237–256.

51. Guerrero-Colón JA, Simoncelli EP, Portilla J. Image Denoising Using Mixtures of Gaussian Scale Mixtures. In: 15th IEEE International Conference on Image Processing; 2008. p. 565–568.

52. Gold JM, Murray RF, Bennet PJ, Sekuler AB. Deriving behavioural receptive fields for visually completed contours. Current Biology. 2000;10(11):663–666.

53. Abbey CK, Eckstein MP. Classification image analysis: Estimation and statistical inference for two-alternativ force-choice experiments. Journal of Vision. 2002;2(5):66–78.

54. Morgenstern Y, Elder JH. Local Visual Energy Mechanisms Revealed by Detection of Global Patterns. Journal of Neuroscience. 2012;32(11):3679–3696. doi:10.1523/JNEUROSCI.3881-11.2012.

55. Neri P. Nonlinear characterization of a simple process in human vision. Journal of Vision. 2009;9(12):1. doi:10.1167/9.12.1.

56. Neri P, Levi D. Temporal dynamics of directional selectivity in human vision. Journal of Vision. 2008;8(1):22.

57. Murray. Classification images: A review. J Vis. 2011;11:2. doi:10.1167/11.5.2.

58. Wilder J, Fruend I, Elder JH. Frequency tuning of shape perception revealed by classification image analysis. Journal of Vision. 2018;18(8):9. doi:10.1167/18.8.9.

59. Szegedy C, Zaremba W, Sutzkever I, Bruna J, Erhan D, Goodfellow I, et al. Intriguing properties of neural networks. In: International Conference on Learning Representations; 2014.

60. Gu S, Rigazio L. Towards Deep Neural Network Architectures Robust to Adversarial Examples. In: International Conference on Learning Representations; 2015.

61. Baker N, Lu H, Erlikhman G, Kellman PJ. Deep convolutional networks do not classify based on global object shape. PLoS Computational Biology. 2018;14(12):e1006613.

62. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet Large Scale Visual Recognition Callenge. International Journal of Computer Vision. 2015;115(3):211–252.

63. 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.

64. Blakemore C, Campbell FW. On the existence of neurones in the human visual system selectively sensitive to orientation and size of retinal images. Journal of Physiology. 1969;203(1):237–260.

65. Goris RLT, Putzeys T, Wagemans J, Wichmann FA. A Neural Population Model for Visual Pattern Detection. Psychological Review. 2013;120(3):472–496.

66. Sebastian S, Abrams J, Geisler WS. Constrained sampling experiments reveal principles of detection in natural scenes. Proc Natl Acad Sci USA. 2017; p. E5731–E5740.

67. Wang Z, Simoncelli EP. Maximum differentiation (MAD) competition: A methomethod for comparing computational models of perceptual quantities. Journal of Vision. 2008;8(12):8, 1–13.

68. Field DJ. Relations between the statistics of natural images and the response properties of cortical cells. Journal of the Optical Society of America, A. 1987;4(12):2379–2394.

69. Wainwright MJ, Simoncelli EP. Scale Mixtures of Gaussians and the Statistics of Natural Images. In: Advances in Neural Information Processing Systems. vol. 12. Cambridge, MA: MIT Press; 2000. p. 855–861.

70. Schwartz O, Simoncelli EP. Natural signal statistics and sensory gain control. Nature Neuroscience. 2001;4(8):819–825.

71. Portilla J, Simoncelli E. A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. International Journal of Computer Vision. 2000;40(1):49–71.

72. Zhaoping L. Understanding Vision. Oxford; 2014.

73. Nelder JA, Mead R. A simplex method for function minimization. Computer Journal. 1965;7:308–313.