Supplementary material

While the main aspects of employed paradigm, procedure, observers and DNNs were already mentioned earlier, this section aims at providing exhaustive and reproducible experimental details. Furthermore, Figure 6 examines how network uncertainty develops as a function of signal strength, and Figure 7 shows the classification accuracy of networks trained on distortions across all conditions. All data, if not stated otherwise, were analysed using R version 3.2.3 [68].

Paradigm & procedure

A schematic of a typical trial is shown in Figure 5. Prior to starting the experiment, all participants were shown the response screen and asked to name all categories to ensure that the task was fully clear. They were instructed to click on the category that they thought resembles the image best, and to guess if they were unsure. They were allowed to change their choice within the 1500 ms response interval; the last click on a category icon of the response screen was counted as the answer. The experiment was not self-paced, i.e. the response screen was always visible for 1500 ms and thus, each experimental trial lasted exactly 2200 ms (300 ms + 200 ms + 200 ms + 1500 ms). During the whole experiment, the screen background was set to a grey value of 0.454 in the [0, 1] range, corresponding to the mean greyscale value of all images in the dataset (41.17 cd/m^2).

On separate days we conducted twelve different experiments. The number of trials per experiment is reported in Table 1. For each experiment, we randomly chose between 70 and 80 images per category from the pool of images without replacement (i.e., no observer ever saw an image more than once throughout the entire experiment). Within each category, all conditions were counterbalanced. Random stimulus selection was done individually for each participant to reduce the influence of any accidental bias in the image selection. Images within the experiments were presented in randomised order. After 256 trials (colour, uniform noise and eidolon experiments), 128 trials (contrast experiment) and 160 trials (remaining experiments), the mean performance of the last block was displayed on the screen, and observers were free to take a short break. Ahead of each experiment, all observers conducted approximately 10 minutes of practice trials to gain familiarity with the task and the position of the categories on the response screen. Trials in which human observers failed to click on any category were recorded as an incorrect answer in the data analysis, and are shown as a separate category (top row) in the confusion matrices (DNNs, obviously, never fail to respond). Such a failure to respond occurred, on average, in only 1.91% of trials per experiment—one of the advantages of controlled laboratory studies (*SD* = 0.69%).

Apparatus

All stimuli were presented on a VIEWPixx LCD monitor (VPixx Technologies, Saint-Bruno, Canada) in a dark chamber. The 22" monitor (484×302 mm) had a spatial resolution of 1920×1200 pixels at a refresh rate of 120 Hz. Stimuli were presented at the center of the screen with 256×256 pixels, corresponding, at a viewing distance of 123 cm, to 3×3 degrees of visual angle. A chin rest was used in order to keep the position of the head constant over the course of an experiment. Stimulus presentation and response recording were controlled using MATLAB (Release 2016a, The MathWorks, Inc., Natick, Massachusetts, U.S.) and the Psychophysics Toolbox extensions version 3.0.12 [69, 70] along with the iShow library (`http://dx.doi.org/10.5281/zenodo.34217`) on a desktop computer (12 core CPU i7-3930K, AMD HD7970 graphics card "Tahiti" by AMD, Sunnyvale, California, United States) running Kubuntu 14.04 LTS. Responses were collected with a standard computer mouse.

Observers & pre-trained networks

Three observers participated in the colour experiment (all male; 22 to 28 years; mean: 25 years) and in the contrast experiment. Six observers participated in the opponent colour, high-pass filter, low-pass filter, phase noise and power equalisation experiments (three female, three male; 20 to 25 years; mean: 22 years). In the other two experiments, five observers took part (uniform noise experiment: one female, four male; 20 to 28 years; mean: 23 years; eidolon experiments: three female, two male; 19 to 28 years; mean: 22 years). Subject-01 is an author and participated in all but the eidolon experiments. All other participants were either paid 10 Euros per hour for their participation or gained course credit. All observers were students and reported normal or corrected-to-normal vision.

We used GoogLeNet [38], VGG-19 [39] and ResNet-152 [40] for our analyses. For all three networks, we used the pretrained implementations as provided by the TensorFlow-Slim framework (`https://github.com/tensorflow/models/tree/master/research/slim` cloned on May 2, 2017) and programmed in the TensorFlow library for machine learning [58]. The individual pretrained weights were also downloaded from the latter GitHub repository. We validated that our installation reproduced the classification accuracies provided on the website. The networks' input were 224×224 pixel RGB images. For greyscale images, we set all three

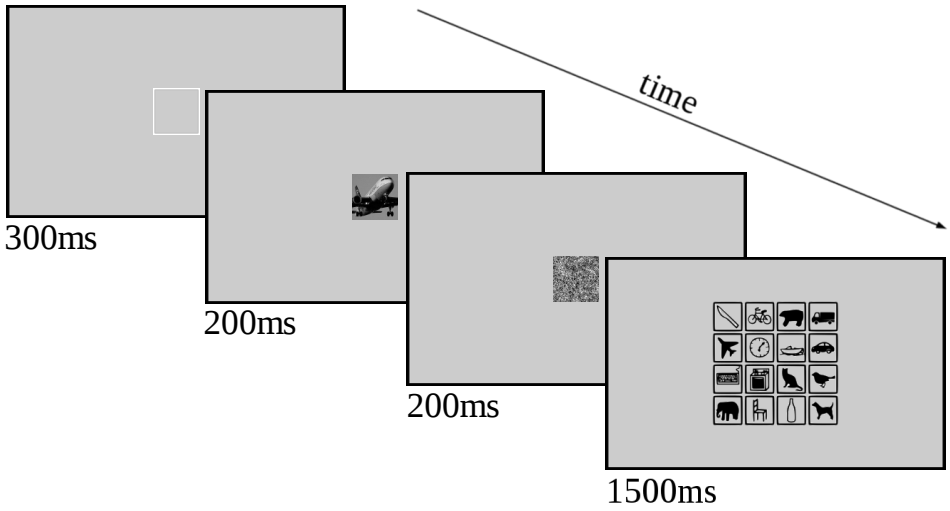


Figure 5: Schematic of a trial. After the presentation of a central fixation square (300 ms), the image was visible for 200 ms, followed immediately by a noise mask with $1/f$ spectrum (200 ms). Then, a response screen appeared for 1500 ms, during which the observer clicked on a category. Note that we increased the contrast of the noise mask in this figure for better visibility when printed. Categories row-wise from top to bottom: `knife`, `bicycle`, `bear`, `truck`, `airplane`, `clock`, `boat`, `car`, `keyboard`, `oven`, `cat`, `bird`, `elephant`, `chair`, `bottle`, `dog`. The icons are a modified version of the ones from the MS COCO website (`http://mscoco.org/explore/`).

Table 1: Numbers of trials in the respective experiments. C. = conditions; P. = practice trials & blocks; M.= main experiment trials and blocks. The per condition column reports the number of trials per category and distortion level. The duration is reported without breaks.

Distortion type	C.	P. blocks	P. total	M. blocks	M. total	Per C.	Duration
Colour	2	2	320	5	1280	40	47 min
Uniform noise	8	2	256	5	1280	10	47 min
Contrast	8	2	256	10	1280	10	47 min
Eidolon I	8	4	384	5	1280	10	47 min
Eidolon II	8	4	384	5	1280	10	47 min
Eidolon III	8	4	384	5	1280	10	47 min
Opponent colours	2	2	224	7	1120	35	41 min
Low-pass filtering	8	2	256	8	1280	10	47 min
High-pass filtering	8	2	256	8	1280	10	47 min
Phase noise	7	2	224	7	1120	10	41 min
Power-equalisation	2	2	224	7	1120	35	41 min
Rotation	4	2	256	8	1280	20	47 min

channels to be equal to the greyscale image's single channel. Images were fed through the networks using a single feedforward pass.

Categories and image database

The images serving as psychophysical stimuli were images extracted from the training set of the ImageNet Large Scale Visual Recognition Challenge 2012 database [50]. This database contains millions of labeled images grouped into 1,000 very fine-grained categories (e.g., over a hundred different dog breeds). If human observers are asked to name objects, however, they most naturally categorise them into so-called basic or entry-level categories, e.g. `dog` rather than `German shepherd` [71]. The Microsoft COCO (MS COCO) database [72] is an image database structured according to 91 such entry-level categories, making it an excellent source of categories for an object recognition task. Thus for our experiments we fused the carefully selected entry-level categories in the MS COCO database with the large quantity of images in ImageNet. Using WordNet's *hypernym* relationship (x is a hypernym of y if y is a "kind of" x, e.g., `dog` is a hypernym of `German shepherd`), we

mapped every ImageNet label to an entry-level category of MS COCO in case such a relationship exists, retaining 16 clearly non-ambiguous categories with sufficiently many images within each category (see Figure 5 for a iconic representation of the 16 categories). A complete list of ImageNet labels used for the experiments can be found in our online repository.[8] Since all investigated DNNs, when shown an image, output classification predictions for all 1,000 ImageNet categories, we disregarded all predictions for categories that were not mapped to any of the 16 entry-level categories. For each of those 16 categories we summed over the predictions of all ImageNet categories mapping to that particular entry-level category. Then the entry-level category with the highest summed prediction was selected as the network's response. This way, the DNN response selection corresponds directly to the forced-choice paradigm for our human observers.

Image preprocessing and distortions

We used Python for all image preprocessing (Version 2.7.11) and for running experiments through pre-trained networks (Version 3.5). From the pool of ImageNet images of the 16 entry-level categories, we excluded all greyscale images (1%) as well as all images not at least 256×256 pixels in size (11% of non-greyscale images). We then cropped all images to a center patch of 256×256 pixels as follows: First, every image was cropped to the largest possible center square. This center square was then downsampled to the desired size with `PIL.Image.thumbnail((256, 256), Image.ANTIALIAS)`. Human observers get adapted to the mean luminance of the display during experiments, and thus images which are either very bright or very dark may be harder to recognise due to their very different perceived brightness. We therefore excluded all images which had a mean deviating more than two standard deviations from that of other images (5% of correct-sized colour-images excluded). In total we retained 213,555 images from ImageNet.

For the experiments using greyscale images the stimuli were converted using the `rgb2gray` method [73] in Python. This was the case for all experiments and conditions except for the 'colour' condition of the **colour experiment**, as well as for the opponent colour experiment. For the **contrast experiment**, we employed eight different contrast levels $c \in \{1, 3, 5, 10, 15, 30, 50, 100\%\}$. For an image in the [0, 1] range, scaling the image to a new contrast level c was achieved by computing

$$new\_value = \frac{c}{100\%} \cdot original\_value + \frac{1 - \frac{c}{100\%}}{2}$$

for each pixel. For the **uniform noise experiment**, we first scaled all images to a contrast level of $c = 30\%$. Subsequently, white uniform noise of range $[-w, w]$ was added pixelwise, $w \in \{0.0, 0.03, 0.05, 0.1, 0.2, 0.35, 0.6, 0.9\}$. In case this resulted in a value out of the [0, 1] range, this value was clipped to either 0 or 1. By design, this never occurred for a noise range less or equal to 0.35 due to the reduced contrast (see above). For $w = 0.6$, clipping occurred in 17.2% of all pixels and for $w = 0.9$ in 44.4% of all pixels. See Figure 10 for example stimuli. For the **salt and pepper noise experiment**, used in DNN training experiments, we also scaled the greyscale image to a contrast level of 30% prior to adding noise in order to ensure maximal comparability with the uniform noise experiment. Salt and pepper noise, i.e. setting pixels to either black or white, was drawn pixelwise with a certain probability p, $p \in \{0, 10, 20, 35, 50, 65, 80, 95\}\%$. See Figure 14 for example salt-and-pepper stimuli at all conditions.

For the **opponent colours experiment**, our aim was to produce images that would be perceived by human observers as having exactly the opposite colours of the original, while retaining the same luminance. Therefore, we converted images to a colour space in which we could invert the colours without affecting luminance values. One such colour space is the Derrington-Krauskopf-Lennie (DKL) colour space [74]. In order to account for the nonlinearity of our experimental display monitor, we measured the emitted luminance for RGB grey values between 0 and 255. From this we built a lookup table from RGB grey values to actual emitted luminance values $f_{monitor}$. To evaluate how much the human retina's long-, middle-, and short-wave receptors would be excited by the colours presented on the monitor, we measured the intensity of all emitted wave lengths between 390-780 nm for the RGB values (255 0 0), (0 255 0), (0 0 255), respectively. We then multiplied the respective emitted spectra between 390-780 nm with the corresponding cone sensitivities taken from the 2-deg LMS fundamentals proposed by [75] and summed over them. This resulted in a matrix C from RGB to cone activities (LMS space). Then we calculated a conversion matrix D of cone activities into the DKL colour space following the conversion example in [76]. An image was, consequently, converted from RGB to DKL by applying $f_{monitor}$ to it and subsequent multiplication with first C and then D. The DKL space has three channels reminiscent of the opponent colour process of the human visual system [76]. They are DKL_{lum}, a luminance channel, DKL_{L-M}, a channel representing the difference between long- and middle-wave receptor activation, as well as DKL_{S-lum}, a channel representing the difference between the activation of the short-wave receptor and the luminance. Since we wanted to keep the luminance unchanged, we multiplied the DKL_{L-M} and DKL_{S-lum} channels with the value '-1'. Subsequently, we converted the manipulated images back to RGB using the inverse matrices of D and C and then applied the inverse of $f_{monitor}$ to them. All resulting pixel values outside the range [0, 1] were clipped to 0 or 1. This only happened for 0.34% of pixels with a mean clipped away value of 0.004. This corresponds to the minimal colour intensity step as $0.004 \approx \frac{1}{255}$.

[8] `https://github.com/rgeirhos/generalisation-humans-DNNs`

For the low-pass and high-pass experiments we used the `scipy.ndimage.filters.gaussian_filter()` function. The **low-pass experiment**'s eight conditions differed in the standard deviation of the Gaussian filter. Standard deviations were 0 (original image), 1, 3, 7, 10, 15 and 40 pixels (Figure 11. We used constant padding with the mean pixel value over the testing images (0.4423) and truncation at four standard deviations. The **high-pass experiment** also had eight conditions. Standard deviations were 0.4, 0.45, 0.55, 0.7, 1, 1.5, 3 pixels and inf (original image) (Figure 11). The high-pass filtered images were produced by subtracting a low-pass filtered image as described above from the original image. However, many of the high-pass filtered images' pixels fell outside the [0, 1] range. To resolve this, we calculated the difference between the mean pixel value over all test images (0.4423) and the mean pixel value of the high-pass filtered image. That difference was added back to the image. This had the effect that images approached a uniform mean grey image of value 0.4423 for low standard deviations. For both experiments pixel values were clipped to the [0, 1] range, if lying outside after the filtering. This only happened for <0.001% of pixels with a mean clipped away value of <0.001 for the both filtering experiments.

We implemented the equalisation of the power spectra and phase noise in the Fourier domain. Conversion to frequency domain was accomplished by a fast Fourier transform through the application of the `fft2()` and then `fftshift()` functions of the Python package `scipy.fftpack`. This results in a matrix of complex numbers F, which represents both the phases and amplitudes of the individual frequencies in one complex number. F is organised in symmetric pairs of complex numbers with just their imaginary part differing in its sign and cancelling each other out when reversing the Fourier transform again. When transforming F to polar coordinates, the angle represents the respective frequency's phase and the distance from the origin represents its amplitude. Hence, we extracted the phases and amplitudes of the individual frequencies with the functions `numpy.angle(F)` and `numpy.abs(F)`, respectively. The **power equalisation experiment** had two conditions: original and power-equalised (Figure 13). For the power-equalised images, we first calculated the mean amplitude spectrum over all test images, which showed the typical $\frac{1}{f}$ shape [e.g. 77, 78]. Thereafter, we set all images amplitudes to the mean amplitude spectrum. Since the power spectrum is the square of the amplitude spectrum, the images were essentially power-equalised. There were seven conditions in the **phase noise experiment**. These were 0, 30, 60, 90, 120, 150 and 180 degrees noise width w (Figure 13). To each frequency's phase a phase shift randomly drawn from a continuous uniform distribution over the interval $[-w, w]$ was added. To ensure that the imaginary parts would later cancel out again, we added the same phase noise to both frequencies of each symmetric pair. After performing the respective manipulations, a F_{new} was calculated by recombining the new phases and amplitudes. Then we did an inverse Fourier transform using `ifftshift()` and then `ifft2()`. Finally we clipped all pixel values to the [0, 1] range. This was the case for 0.038% of pixels with a mean clipped value of about 0.003 for the phase noise experiment and for 0.013% of pixels with a mean clipped value of 0.005 for the power-equalisation experiment.

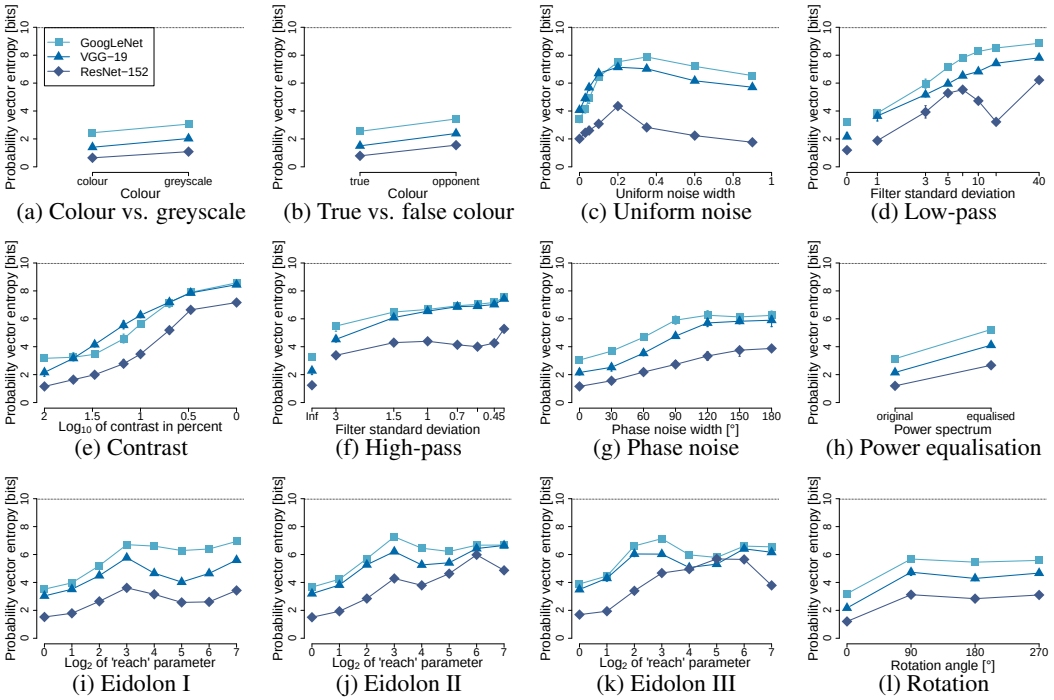
There were four conditions for the **rotation experiment**: 0 (original), 90, 180 and 270 degrees rotation angle. Rotation by 90 degrees was accomplished by first transposing the image matrix and then reversing the column order. Rotation by 180 degrees was done by reversing both, row and column ordering. Rotation by 270 degrees was implemented by first reversing the images columns and then transposing it.

For the **eidolon experiments**, all stimuli were generated using the eidolon toolbox for Python[9], more specifically its `PartiallyCoherentDisarray(image, reach, coherence, grain)` function. Using a combination of the three parameters reach, coherence and grain, one obtains a distorted version of the original image (a so-called eidolon). The parameters reach and coherence were varied in the experiment; grain was held constant with a value of 10.0 throughout the experiment (grain indicates how fine-grained the distortion is; a value of 10.0 corresponds to a medium-grainy distortion). Reach $\in \{1.0, 2.0, 4.0, 8.0, 16.0, 32.0, 64.0, 128.0\}$ is an amplitude-like parameter indicating the strength of the distortion, coherence $\in \{0.0, 0.3, 1.0\}$ defines the relationship between local and global image structure. Those two parameters were fully crossed, resulting in $8 \cdot 3 = 24$ different eidolon conditions. A high coherence value "retains the local image structure even when the global image structure is destroyed" [57, p. 10]. A coherence value of 0.0 corresponds to 'completely incoherent', a value of 1.0 to 'fully coherent'. The third value 0.3 was chosen because it produces images that perceptually lie—as informally determined by the authors—in the middle between those two extremes. See Figures 12 and 13 for example eidolon stimuli. The coherence levels of 1.0, 0.3 and 0.0 are refered to as eidolon experiment I, II and III throughout the paper.

Experimental modifications

Our psychophysical experiments were conducted in two batches and over an extended period of time. After completing the first batch of experiments (all experiments on the left half of Figure 3, i.e. a, c, e, g, i and k), we performed a number of modifications for the second batch of experiments. We here briefly list all the changes in which the second batch of experiments differed from the previously reported methods.

[9] `https://github.com/gestaltrevision/Eidolon`



(a) Colour vs. greyscale (b) True vs. false colour (c) Uniform noise (d) Low-pass

(e) Contrast (f) High-pass (g) Phase noise (h) Power equalisation

(i) Eidolon I (j) Eidolon II (k) Eidolon III (l) Rotation

Figure 6: Mean entropy of the probabilities for the 1000 ILSVRC classes for GoogLeNet, VGG-19 and ResNet-152. Dotted line indicates the maximum possible entropy. This is a measure of network 'uncertainty'.

Noise mask: In the human experiments, each experimental image was immediately followed by a $\frac{1}{f}$ pink noise mask (cf. Figure 5). In the second batch of experiments this noise mask was enhanced to improve its masking effect. This was done by multiplying each pixel value by four. Values greater than 1 or smaller than 0 due to the multiplication were then clipped to 1 or 0.

Cropping vs. downsampling: In the first experimental batch, humans saw 256×256 images. However, DNN classification was based on those images 224×224 centre crop. Thus, humans and DNNs saw slightly different images. Therefore, we used downsampling to 224×224 for the second batch of both, human and DNN experiments. As a consequence, the mean grey pixel value over all experimental images and hence the background grey value for presenting those images changed slightly from 0.454 to 0.442 in the [0, 1] range.

JPEG vs. PNG: All images of the first batch of experiments, prior to showing them to human observers or DNNs, were saved in the JPEG format using the default settings of the `skimage.io.imsave` function. The JPEG format was chosen because the image training database for all three networks, ImageNet [50], consists of JPEG images. However, as the lossy compression of JPEG may introduce artefacts, we also examined the difference in DNN results between saving to JPEG and to PNG, which is lossless up to rounding issues. We therefore ran all those DNN experiments additionally saving them in the (up to rounding issues) lossless PNG format. We did not find any noteworthy differences for the colour, noise, and eidolon experiments. However, for the contrast experiment, the networks achieved on average better results for PNG images. We therefore tested three human observers additionally on the same stimuli (PNG instead of JPEG images). In this experiment, three of the JPEG experiment's five observers participated for maximal comparability.[10] We found human observers to be better for PNG images as well. In absolute terms, participants were 2.68% better on average. In order to disentangle the influence of JPEG compression and image manipulations, we used PNG images for all other experiments, that is for the false colour, phase noise, power equalisation, rotation, high-pass and low-pass experiments as well as for the DNN training experiments.

Python Version: The second batch used Python Version 3.5 instead of Python 2.7 for image preprocessing.

[10] A time gap of approximately six months between both experiments should minimise memory effects; furthermore, human participants were not shown any feedback (correct / incorrect classification choice) during the experiments.

Error bars & entropy

When showing *accuracy* in any of the plots, the error bars provided report the range of the data observed for different observers (not the often shown S.E. of the means, which would be much smaller). To produce a comparable measure of uncertainty for the DNNs, we computed seven runs with different subsets of the data, with each run consisting of the same number of images per category and condition that a single human observer was exposed to and report the range of accuracies observed in these runs. Seven runs are the maximum possible number of runs without ever showing an image to a DNN more than once per experiment.

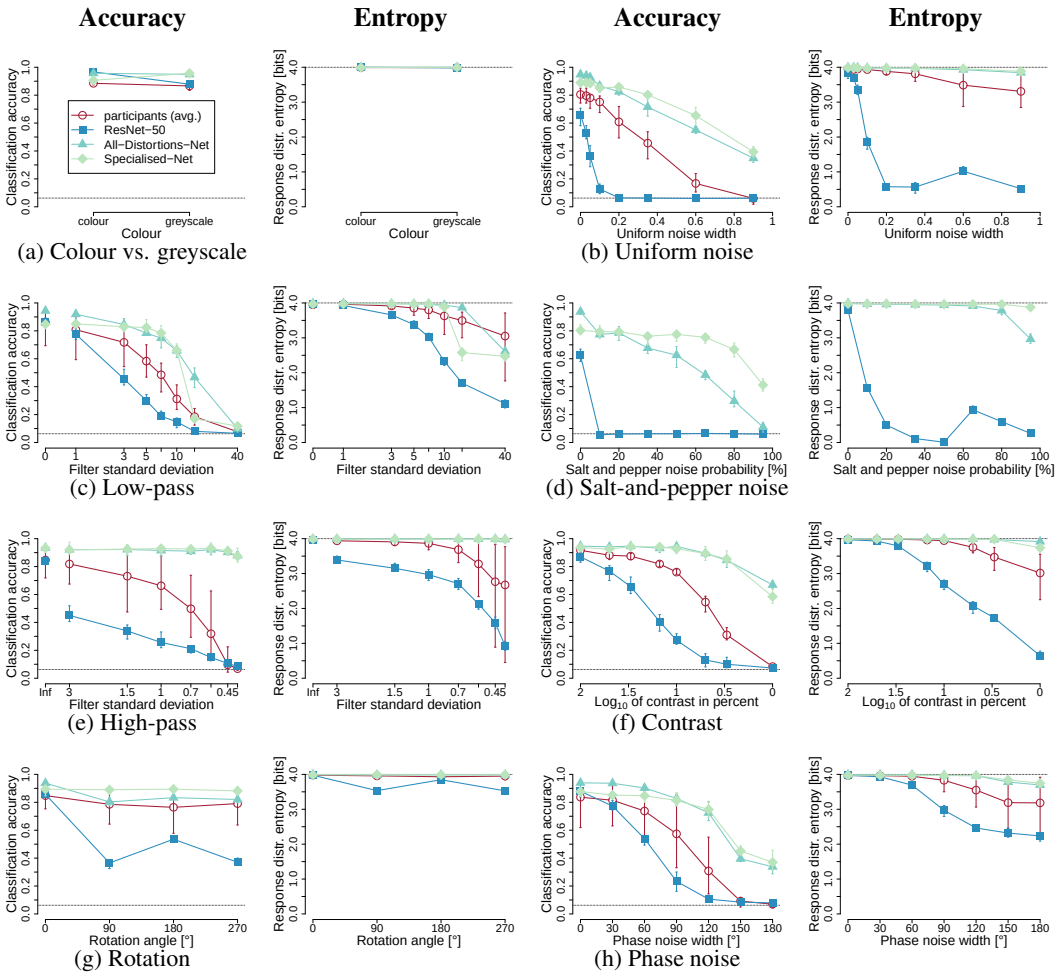
For all response distribution entropy results (Figures 3 and 7), we calculated the entropy as the average of individual participants' entropies: otherwise, if the entropy was calculated over the aggregated human trials, individual differences might cancel each other out, which would lead to a higher human response distribution entropy.

Prediction uncertainty

Figure 6 shows the entropy of the networks' predictions over the 1000 ILSVRC12 classes as a measure of the networks' 'uncertainty'. In principle, the more uncertain a network is in its predictions the more evenly it will distribute its softmax activations between classes and thus the higher the entropy will be. For all experiments, uncertainty roughly increases with distortion strength as reported by previous studies [59, 60]. However, for uniform noise and the eidolon distortions all networks become more certain again for some higher distortion levels. Furthermore, ResNet-152 also becomes more certain for stronger distortions in the low-pass experiment and is consistently more confident in its predictions than GoogleNet and VGG-19. Thus there are distortions for which all networks and especially ResNet-152 fail to capture that the input signal is becoming worse. Instead they limit their predictions to only a few classes (cf. Figure 3) with high certainty. This result is in line with previous reports stating that uncertainty in standard discriminative deep neural networks is not well represented [e.g. 79].

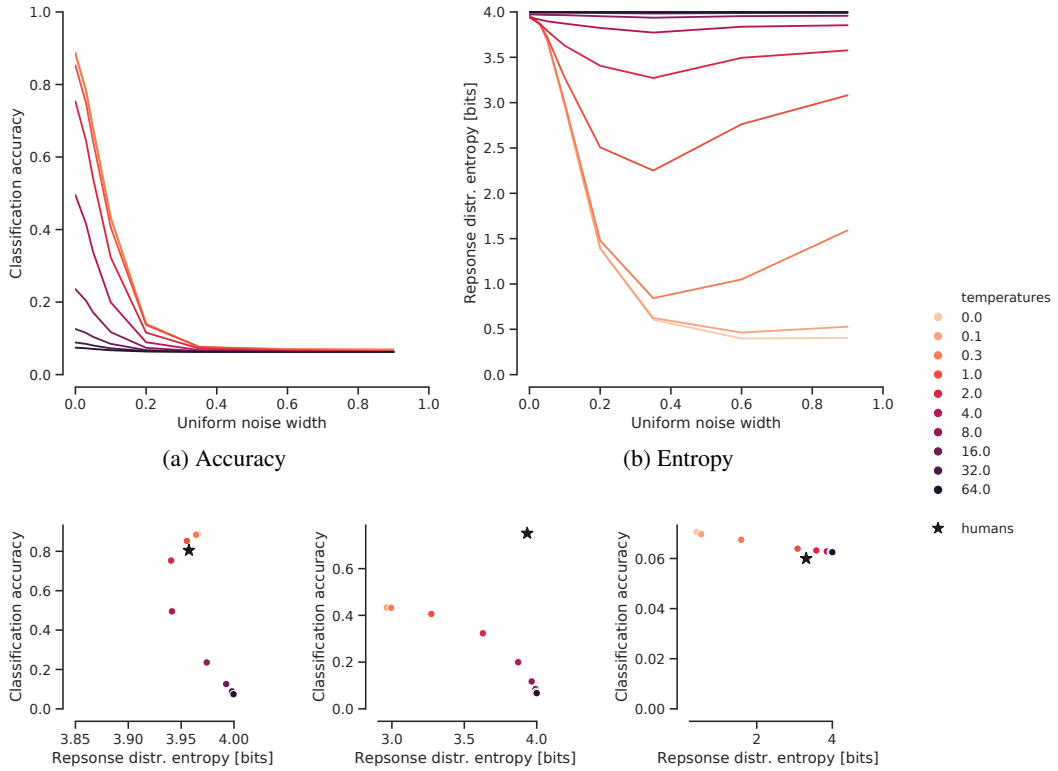
Network training results across all distortion conditions

While Figure 4 shows performance of networks trained on distortions for a single distortion level per manipulation, we here report the performance across all stimulus levels. In Figure 7, the performance of a vanilla ResNet-50 is compared against a network with the same architecture that is trained on a distortion directly (refered to as 'Specialised-Net'), as well as to a network that is trained on all distortions simultaneously (named 'All-Distortions-Net'). Object recognition accuracy shows a relatively consistent pattern across experiments: human performance is better than the performance of a vanilla ResNet-50. However, both an All-Distortions-Net and a Specialised-Net reach extremely high accuracies, with the Specialised-Net being either on par with or slightly better than the All-Distortions-Net. Interestingly, the response distribution entropy of those two networks is largely human-like, i.e. close to 4 bits of entropy (or no bias towards a certain category), even for conditions where the overall accuracy is low (e.g. for the difficult conditions of uniform and salt-and-pepper noise).



(a) Colour vs. greyscale (b) Uniform noise (c) Low-pass (d) Salt-and-pepper noise (e) High-pass (f) Contrast (g) Rotation (h) Phase noise

Figure 7: Classification accuracy and response distribution entropy for human observers, ResNet-50 as well as an All-Distortions-Net and a Specialised-Net. All networks are trained from scratch; the Specialised-Net in every plot is trained on a single distortion (models A1 to A9 in Figure 4) whereas the All-Distortions-Net is trained on a number of distortions simultaneously. This corresponds to models C1 and C2 in Figure 4: for subplot 7d, salt-and-pepper noise, performance of model C1 is shown. For subplot 7b, uniform noise, performance of C2 is shown. For all other plots, performance of the All-Distortions-Net is shown as the mean of performance for models C1 and C2.



(a) Accuracy (b) Entropy

(c) Uniform noise width = 0.0 (d) Uniform noise width = 0.1 (e) Uniform noise width = 0.9

Figure 8: Classifcation accuracy **(a)** and response distribution entropy **(b)** as well as the trade-off between accuracy and entropy **(c, d, e)** for different softmax temperatures when the decision of a ResNet-50 model is sampled from its distribution over classes (softmax output) rather than taking the argmax of the distribution (which is equivalent to sampling with temperature $\rightarrow 0$). While increasing the temperature does increase the response distribution entropy of ResNet-50, it simultaneously decreases the classification accuracy. For uniform noise with a width of 0.1 (d), increasing the temperature to match the response distribution entropy of humans reduces the accuracy of ResNet-50 below 0.1 whereas human accuracy is at 0.75.

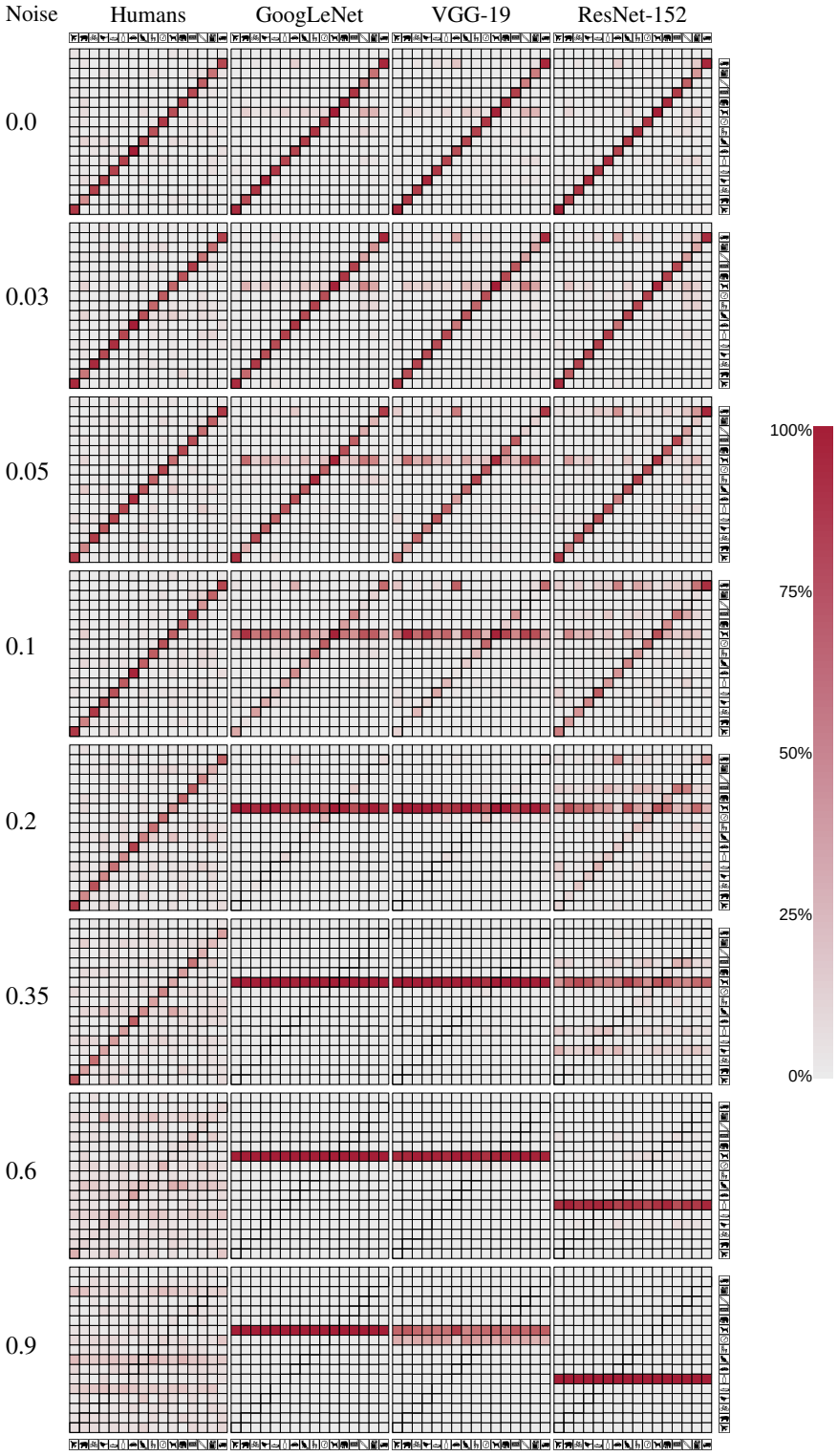
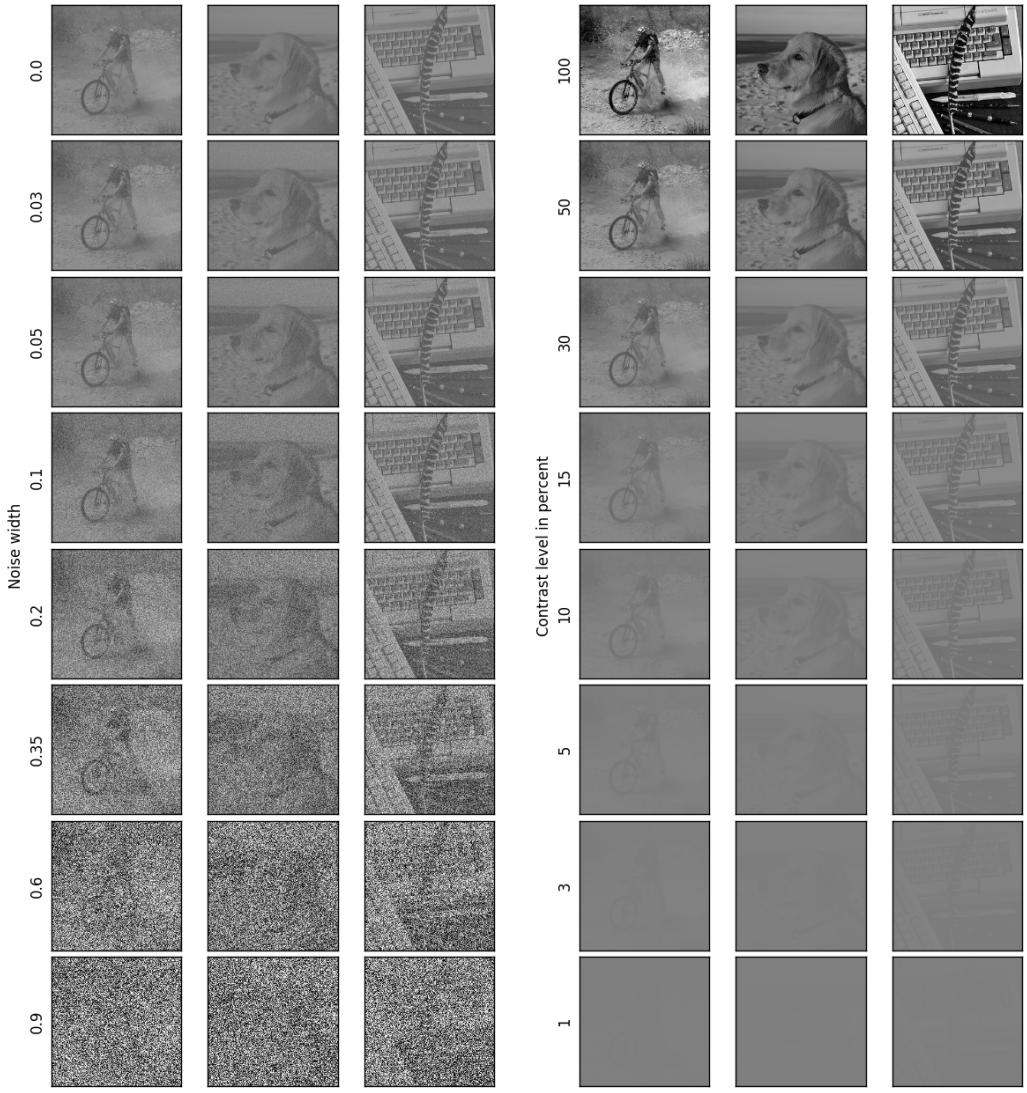


Figure 9: Confusion matrices for additive uniform noise. Columns indicate correct category, rows the given classification decision. The top row of each confusion matrix indicates the fraction of failures to respond (i.e. if human observers failed to click on a category).



(a) Uniform noise (b) Contrast

Figure 10: Three example stimuli for different conditions of uniform noise and contrast experiments. The three images (categories `bicycle`, `dog` and `keyboard`) were drawn randomly from the pool of images used in the experiments. Best viewed on screen.

(a) Low-pass (b) High-pass

Figure 11: Three example stimuli for different conditions of low-pass and high-pass experiments. The three images (categories `bicycle`, `dog` and `keyboard`) were drawn randomly from the pool of images used in the experiments. Best viewed on screen.

(a) Eidolon I (b) Eidolon II

Figure 12: Three example stimuli for different conditions of eidolon I and eidolon II experiments. The three images (categories `bicycle`, `dog` and `keyboard`) were drawn randomly from the pool of images used in the experiments. Best viewed on screen.

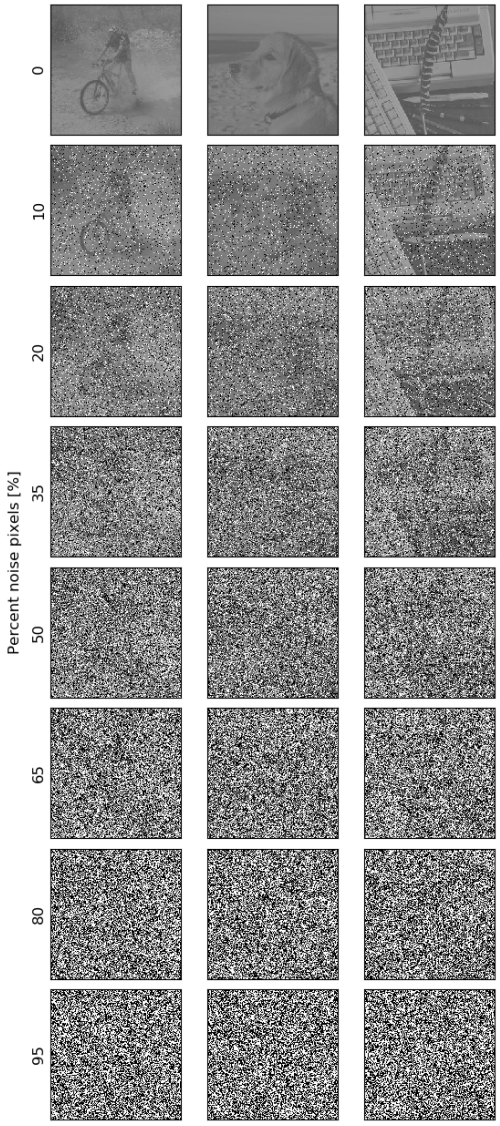
(a) Eidolon III

(b) Phase noise

(c) False colour

(d) Power equalisation

Figure 13: Three example stimuli for different conditions of Eidolon III, phase noise, false colour and power equalisation experiments. The three images (categories `bicycle`, `dog` and `keyboard`) were drawn randomly from the pool of images used in the experiments. Best viewed on screen.



(a) Salt and pepper noise

Figure 14: Three example stimuli for different conditions of salt and pepper noise. The three images (categories `bicycle`, `dog` and `keyboard`) were drawn randomly from the pool of images used in the experiments. Best viewed on screen. Salt and pepper noise was used in DNN training experiments with the conditions depicted in the figure above.