SUPPLEMENTARY MATERIAL FOR

Do computer vision models differ systematically from human object perception?

RT Pramod[1,2] & SP Arun[1]

[1]Centre for Neuroscience & [2]Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore

CONTENTS

1) **Computational models**
2) **Residual error patterns for 100-PC based models**
3) **Generalization of the best model to novel experiments**
4) **Best model performance when fitted to individual experiments**

1. COMPUTATIONAL MODELS

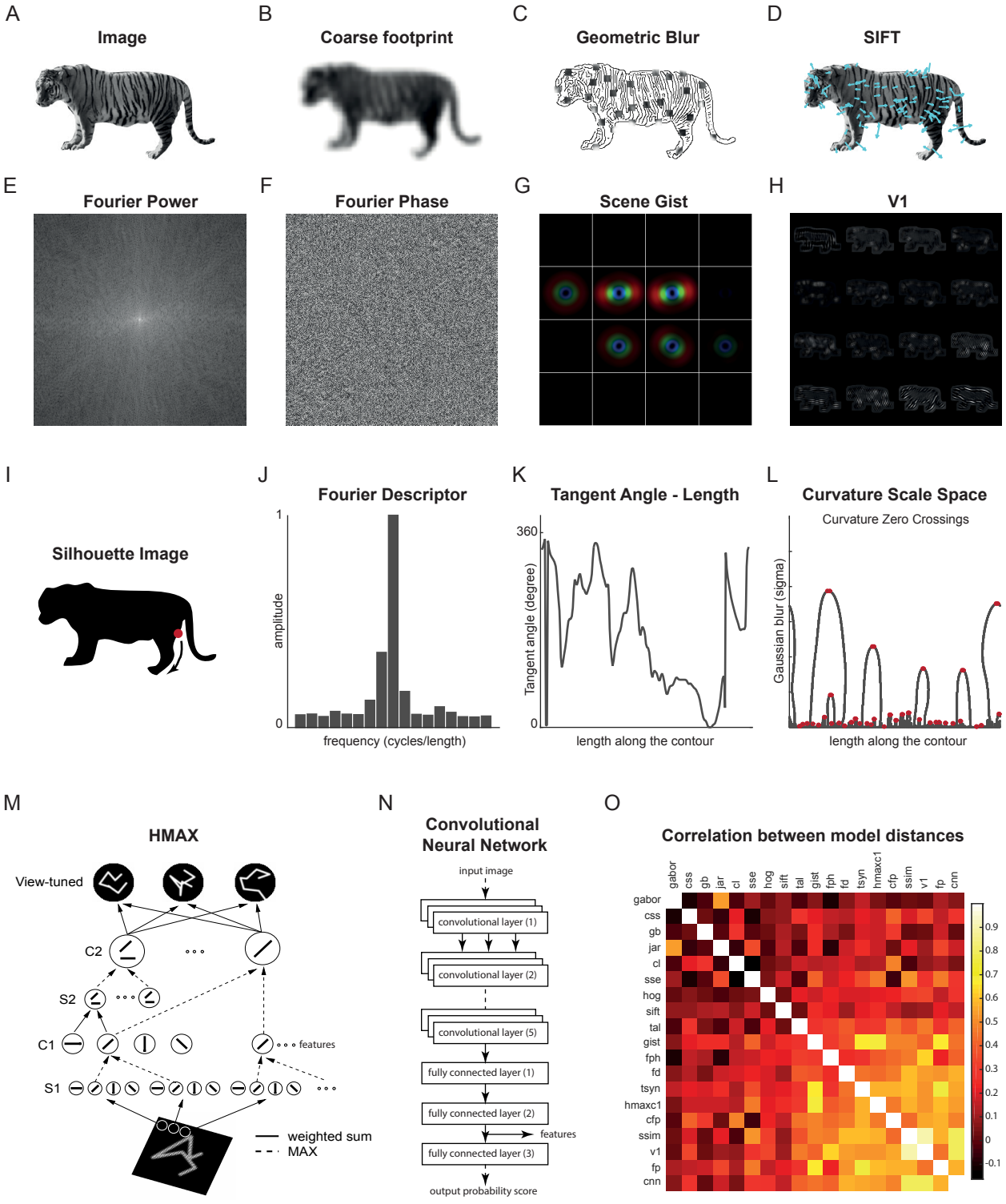

Figure S1. Visualization of the image representation for a subset of the models tested. (A) Original Image (B) Coarse footprint extracts a coarse description of the image. (C) Geometric blur representation. The square patches are the regions around the interest points overlaid on the edge image found using Canny edge detector. (D) SIFT representation. The blue arrows indicate the magnitude and direction of the gradient at key-points. (E) Fourier power represents the power in each spatial frequency (along both x & y) (F) Fourier phase represents the phase of each spatial frequency (along both x & y) (G) Scene Gist represents the output of a bank of Gabor filters operating on non-overlapping windows in the image. (H) V1 represents the image as a bank of oriented filters. The plots show outputs of 16 model V1 units. (I) Input image for boundary-based models (J) Fourier Descriptor (FD) represents the strength of frequency components along the contour of the object (K) Tangent Angle Length represents the

tangent angle of the boundary at regular locations along the entire length. (L) Curvature Scale Space represents a closed contour as a set of zero-crossings of curvature at multiple spatial scales. The plot shows the curvature zero crossing map with the scale space features marked in red. (M) HMAX model. The figure shows the alternating layers of simple and complex cells ultimately leading to view-tuned cells (from Riesenhuber and Poggio, 1999). (N) Convolutional Neural Network. An input image is passed through various convolutional and fully-connected layers of artificial neural units before computing the output probability score (likelihood of belonging to each category). We used the output of penultimate fully-connected layer as the feature vector. (O) Correlation between model distances. This plot shows the correlation between unweighted feature distances of all 26,675 pairs of objects between pairs of models.

DETAILED MODEL DESCRIPTIONS

All images in the dataset were scaled to a square frame of 140 pixels which was given as input to each model. In some models (CNN, TSYN), the input image is rescaled to another size before feature extraction in their standard implementation, and therefore we retained this rescaling step.

Pixel based models

1. *Sum-of-squared error (SSE):* This is the simplest of all models where pixel intensities are considered as features (Figure S1A). For a given pair of images, one of the images was linearly shifted in the x and y directions over the other image. The lowest sum of squared difference across all possible shifts was taken as the distance between the two images. (*Number of features = 19,600)*
2. *Coarse footprint (CFP):* This model has been previously used to explain image similarity driven by coarse structure of objects (Sripati and Olson, 2010, Mohan and Arun, 2012). The images were first shifted and scaled to a constant frame without changing the aspect ratio. Then, the images were low-pass filtered using a Gaussian filter. These filtered images were then normalized to have a total intensity of 1. An example coarse footprint image is shown in Figure S1B. Coarse footprint distance between any two images was calculated as the city-block distance between the images. (*Number of features = 19,600)*

Boundary based models

Boundary based models compute features on the digital boundary extracted from the image. Natural objects were converted to silhouette images before extracting boundary-based features.

1. *Curvature scale space (CSS):* This method was proposed to solve planar curve matching problem using descriptions at varying levels of detail (Mokhtarian and Mackworth, 1986, Mokhtarian, 1995). First, the external contour (a set of x-y co-ordinates) of the object is extracted and convolved with gaussian kernel with varying levels of standard deviation (σ) to get coarser representations of the object contour. Then, for each σ, curvature along the contour is calculated and points of inflections (curvature zero-crossing) are found. A plot of curvature zero-crossings along the curve for various values of σ is known as the curvature scale space image (Figure S1L). Local maxima in the curvature scale space image at a particular threshold of detail (σ_{th}) are considered to be features. That is, a set of co-ordinates (length, σ) form the feature representation. For any given pair of contours, the corresponding feature points are shifted to yield closest match and the Euclidean distance between the features is computed. (*Number of features = 6)*
2. *Curvature length (CL):* Curvature-Length representation of an object contour is obtained by computing the curvature (κ) at every point on the contour as a function of cumulative length of the contour. Curvature at a point on a digital curve (x(t), y(t)) is calculated as,

$$\kappa(t) = \frac{x'y'' - y'x''}{\left(x'^2 + y'^2\right)^{\frac{3}{2}}} \text{ where } t = [0,1]$$

   where, x' and y' represent first derivative and x'' and y'' represent second derivative with respect to the normalized contour (t). Distance between two contours is calculated as the Euclidean distance between the curvature-length representations of the contours. We used gaussian windows with various levels of variance to smooth the contour before computing Curvature Length representation. Specifically, we used 101 levels of gaussian variance (0 to 1 in steps of 0.01) and chose the level of smoothening (= 0.23) that gave best correlation with observed dissimilarity. (*Number of features = 200)*
3. *Tangent Angle Length (TAL) representation:* Tangent Angle-Length representation is a simple representation of the object contour obtained by computing tangent angle at every point on the contour. The tangent angle, in the range [0, 360), is calculated at every point on the contour and represented as a function of cumulative length of the contour (Figure S1K). The Euclidean distance between two TAL representations is considered as the measure of dissimilarity. We used gaussian windows with various levels of variance to smooth the contour before computing Tangent Angle Length representation. Specifically, we used 101 levels of gaussian variance (0 to 1 in steps of 0.01) and computed correlations

between observed and model dissimilarities. However, the best correlation was obtained for gaussian variance of 0. (*Number of features = 200)*

4. *Fourier descriptor (FD):* Fourier descriptors are used to represent contours of objects in the x-y plane (Zahn and Roskies, 1972). An N-point digital contour of an object is transformed into a sequence of N complex numbers as, $z(n) = x(n) + jy(n)\ for\ n = 1,2 \dots N$. The discrete Fourier transform (DFT) of the sequence z(n) is computed and the resulting set of Fourier coefficients form the Fourier descriptors of the contour. Instead of considering all coefficients, only the first few coefficients can be used to match two contours. Specifically, the distance between two contours can be calculated as the Euclidean distance between the first K coefficients of the Fourier descriptor representation. In our experiments, we chose N = 500 and used a range of values for K (2 to 500 in steps of 2) and obtained the best correlation between model distances and perceived dissimilarities for K = 10. Fourier descriptor representation for the 'tiger' image is shown in Figure S1J. (*Number of features = 500)*

Feature based models

1. *Gabor Filterbank (Gabor):* The features in this model are the projections of an image onto a Gabor wavelet pyramid. The Gabor wavelet pyramid model used in our experiments is a simplified version used in a previous study (Kay et al., 2008) obtained from the Image Similarity Toolbox (https://github.com/daseibert/image_similarity_toolbox/). The Gabor wavelet filters span eight orientations (in multiples of π/8), four sizes (covering 100%, 33%, 11% and 3.7% of the image) and different shifts across the image such that the filters tile the entire image for each filter size. The vectors of filter responses are compared between images by computing Euclidean distance. (*Number of features = 1,3600)*
2. *Geometric Blur (GB):* Geometric blur computes local image properties at selected interest points (Berg and Malik, 2001,
3. Berg et al.,2005). These interest points were randomly selected from edges found by a Canny edge detector (Canny, 1986). Apart from local image properties, the relative locations of the interest points were also considered, thus incorporating global geometric properties of the image in the representation. Feature vector was formed by collecting low-pass filtered pixel values sampled regularly along radiating circles around the interest points. The amount of low-pass filtering was proportional to the distance between the pixel and the interest point. For a pair of images, the interest points were matched and the dissimilarity for each pair of points was computed by taking the weighted sum of the negative correlation between the corresponding feature vectors, the Euclidean distance between the points, and the change in circle orientation (
4. Berg et al.,2005). The total dissimilarity between the images was calculated by summing the dissimilarities for all pairs of interest points. We used an implementation of Geometric Blur found in Image Similarity Toolbox with default parameters, available at (https://github.com/daseibert/image_similarity_toolbox/). The interest points for the 'tiger' image are shown as small patches on the Canny edge image (Figure S1C).
5. *Scale Invariant Feature Transform (SIFT):* SIFT is a hugely popular algorithm in computer vision mainly used to describe local features in images (Lowe, 2004). It has been widely used in object recognition, object tracking and image stitching. We used the well-known VLFeat package (http://www.vlfeat.org/) to extract SIFT key-points and feature vectors. The feature vectors were remapped by pooling across all images and clustering them using Matlab's inbuilt k-means clustering algorithm with k = 15. In this way, every feature was assigned to a cluster and a histogram of cluster IDs was computed for each image. These histograms were compared using KL divergence to obtain distances between images. SIFT key-points for the 'tiger' image in Figure S1A is shown in Figure S1D. The magnitude and direction of the arrows indicate the magnitude and direction of the gradient respectively. To get Principal Components for SIFT model, we chose k = 100 and computed higher dimensional feature vectors. (*Number of features = 15 or 100)*
6. *Histogram of Oriented Gradients (HOG):* This is one of the widely used feature descriptors in computer vision for the purpose of object detection (Dalal and Triggs, 2005). In this representation, the image is broken down into overlapping blocks spanning the entire image space. Then, a normalized histogram of gradient orientations is calculated for every block. The computed histograms for all blocks of the image are concatenated to obtain the feature representation. We used the Matlab inbuilt function extractHOGFeatures with default parameters to compute HOG feature for an image. Distance between two HOG representations was calculated as the Euclidean distance. (*Number of features = 9,216)*
7. *Scene Gist (GIST):* Scene Gist was specially proposed for recognizing scenes rather than objects (Oliva and Torralba, 2001). However, this representation has been used previously to measure contextual influences on object recognition (Leeds et al., 2013). In this model, each image is represented as a weighted sum of bases, found through PCA on windowed Fourier transform of the image. The number of bases was chosen such that the image could be reconstructed with minimum error. The weights used for bases during

reconstruction were treated as features. We used an implementation of scene gist in Image Similarity Toolbox (https://github.com/daseibert/image_similarity_toolbox/). The feature vector for each image was normalized to sum to 1. The distance between each pair of images was calculated as the KL divergence between the corresponding normalized feature vectors. The scene gist representation of the 'tiger' image is shown in Figure S1G where blocks represent non-overlapping windows, colours represent scale and saturation represents orientation of the basis functions. (*Number of features = 320)*

8. *Fourier Phase (FPh):* The Fourier phase representation of an image is also computed by taking the 2-D Discrete Fourier transform. However, instead of computing the magnitude of Fourier coefficients, we calculate the angle (or phase) of the coefficients. The Euclidean distance between two such phase representations is considered as the distance measure. The Fourier phase representation for the 'tiger' image in Figure S1A is shown in Figure S1F. (*Number of features = 19,600)*
9. *Fourier Power (FP):* The Fourier power of an image is computed by taking the 2-D Discrete Fourier transform. The magnitude of the Fourier coefficients are calculated and normalized to have unit power. The Euclidean distance between two such normalized magnitude spectra is taken as the distance measure. The Fourier power representation for the 'tiger' image in Figure S1A is shown in Figure S1E. (*Number of features = 19,600)*

Statistical models

1. *Texture Synthesis (TSYN):* We tested a popular texture synthesis/analysis model (Portilla and Simoncelli, 2000). In this model, each image is passed through wavelet filters where it is initially split into high- and low-pass bands. The low-pass band is further split into a lower frequency band and various orientation sub-bands. Finally, the image is characterized by a set of statistics (central moments, range of pixel intensities, and correlation) computed on filter outputs or coefficients at adjacent spatial locations, orientations and scales. We used a MATLAB implementation of the Texture Synthesis algorithm available online (http://www.cns.nyu.edu/~lcv/texture/). We used wavelet filters corresponding to 5 scales and 4 orientations with a pixel neighbourhood of 9 pixels. All the statistics were concatenated to form a single feature vector for each image. (*Number of features = 3,027)*
2. *Structural Similarity Index (SSIM):* SSIM is used for measuring similarity between two images (Wang et al., 2004). This measure was designed to closely mirror human visual perception of distortions/degradations in an image. The SSIM between two image patches x and y of same size is given by,

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_2)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Where μ_x and μ_y are the averages and σ_x and σ_x are the variances of image patches x and y respectively. σ_{xy} is the covariance of x and y and c_1 and c_2 are constants. We used the Matlab implementation of SSIM with default values to get similarity measure between each pair of images. We used the structural dissimilarity index, defined as $\frac{(1-SSIM(x,y))}{2}$, to get distances between images.

Network-based models

1. *Jarrett et. al. model (Jar):* This is a biologically inspired hierarchical model with one stage of random filters with no learning involved (Jarrett et al., 2009). This model is described as $F_{csg} - R_{abs} - N - P_a$ in the literature where a first stage of random filters is followed by stages of divisive normalization, out-put nonlinearity and pooling. For a given image, the output of every unit in this model was converted into a histogram of pixel values. All such histograms were collected to form the feature representation. KL divergence between feature vectors of two images was computed to get distances. (*Number of features = 2624)*
2. *HMAX:* HMAX is a popular biologically inspired model of object representation with several layers of units (Riesenhuber and Poggio, 1999). These units are labelled as simple (S) and complex (C) because of their resemblance to the simple and complex cells in the primary visual cortex. These layers of units alternate between summing and winner-take-all (max) computations. We used an implementation of HMAX-C1 model provided in the Image Similarity toolbox (https://github.com/daseibert/image_similarity_toolbox/) to get feature vectors corresponding to images. Euclidean distance between two feature vectors was computed as the distance measure between images. A schematic of HMAX model is shown in Figure S1M. (*Number of features = 1728)*
3. *V1 model (V1):* We used a standard V1 model consisting of Gabor filters (Pinto et al., 2008). This model produced responses of a population of V1-like neurons using a bank of Gabor filters with output divisive normalization. The output of our implementation of the V1 model consisted of the activity of 48 (= 6 spatial frequencies x 8 orientations) model neurons in response to an image. The outputs of all the model neurons were concatenated to create a feature vector. The Euclidean distance between two feature vectors

was calculated as the distance between two images. Figure S1H shows the output of 16 model V1 neurons in response to the 'tiger' image. (*Number of features = 940,800)*

4. *Convolutional Neural Network (CNN):* Convolutional Neural Networks are a class of computational models that are inspired by the hierarchical nature of computations in the brain. These models have gained widespread acclaim in recent years with state-of-the-art performances in various visual tasks. There are various implementations of CNNs available in the literature with various parameter choices and learning techniques. In this study, we used a pre-trained CNN (VGG-16, http://www.vlfeat.org/matconvnet/pretrained/) with 3x3 convolutional filters and 16 weight layers (Simonyan and Zisserman, 2014). We used the outputs of the penultimate fully connected layer as features (Figure S1N). Hence, each image was represented by a collection of activities of 4,096 units. For each pair of images, the Euclidean distance between feature vectors was computed as distance. (*Number of features = 4,096, fully connected layer)*

2. RESIDUAL ERROR PATTERNS FOR 100-PC BASED MODELS

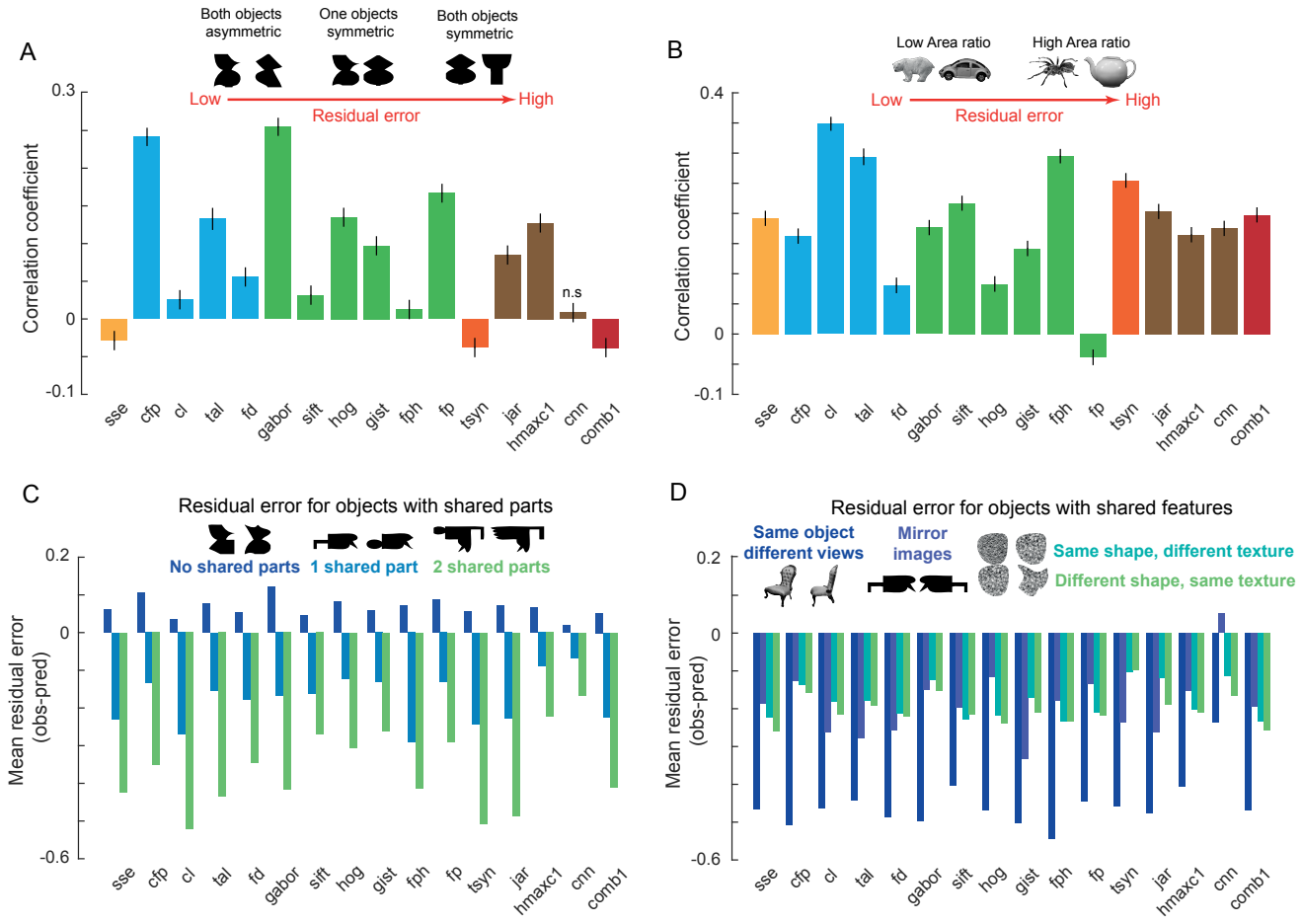

Figure S2. Residual error patterns for models after fitting to the perceptual data. We repeated the analysis in Figure 6 except that each model was fit to the perceptual data using the weighted sum of the feature differences along the first 100 principal components. (A) Correlation between symmetry strength and residual error across object pairs for each model. Error bars indicate standard deviation. All correlations are significant with $p < 0.05$. Non-significant correlations are indicated as "n.s". (B) Correlation between area ratio and residual error across object pairs for each model. (C) Average residual error across image pairs with zero, one or two shared parts. (D) Average residual error for objects pairs related by view, mirror reflection, shape and texture.

3. GENERALIZATION OF THE BEST MODEL TO NOVEL EXPERIMENTS

Although the best model (*comb2*) shows a good cross-validated prediction of the perceptual data, this may not accurately represent its ability to generalize to novel images. This is because the model is trained each time on 80% of the image pairs but these image pairs may contain all the images in the dataset. Because our dataset is based on 32 distinct experiments which contain non-overlapping sets of images, we asked whether the *comb2* model would predict the outcome of each experiment when it is trained on all other experiments. The resulting model performance, expressed as usual in terms of the percentage of the variance explained is shown in Figure S3. The best model was able to generalize to many new experiments, but not to all experiments. Specifically, the model generalized poorly to experiments containing similar natural objects, objects in multiple views, symmetric objects, broken objects and to sets of natural objects containing vehicles. This pattern was similar even when the best model was trained and tested exclusively on individual experiments (Supplementary Section S4). Note that although the experiments contained distinct images, they share many properties: for instance, several experiments contained symmetric objects, mirror images and objects in multiple views. Yet the best model generalized well to specific experiments and did not generalize to others. These patterns are similar to the residual error patterns reported in the main text.

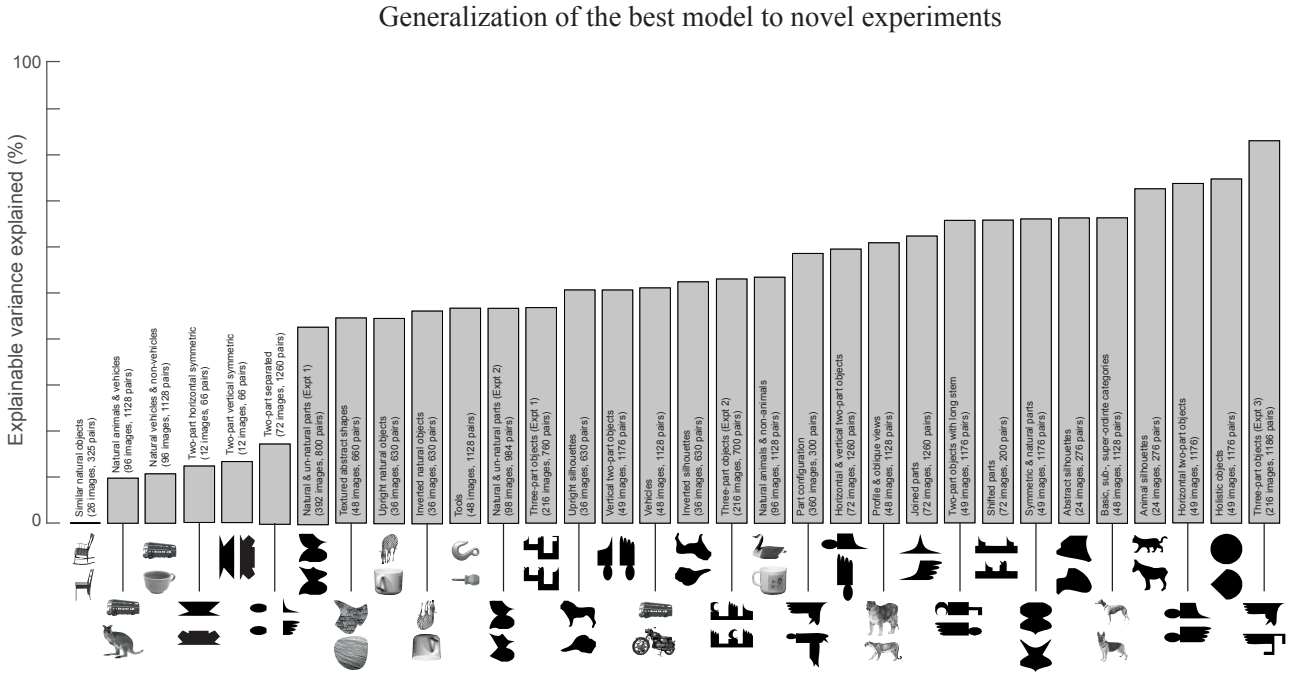

Figure S3. Generalization of the best model to novel experiments. Each bar represents the amount of variance explained by the best model (*comb2*) when it was trained on all other experiments and tested on the image pairs of that particular experiment. The text inside each bar summarizes the images and image pairs used, and the image centered below each bar depict two example images from each experiment.

4. BEST MODEL PERFORMANCE WHEN FITTED TO INDIVIDUAL EXPERIMENTS

In the previous section, we investigated the generalization capabilities of the best model (*comb2*) to novel experiments when trained on other experiments. Here, we set out to explore whether the trends observed previously hold even when the best model was trained to predict data from the same experiment. We considered 16 experiments which had perceptual data for at least 1000 image pairs and trained the model on 800 image pairs for each individual experiment. The model was then tested on a separate test set containing 200 image pairs. This procedure was repeated 10 times for each experiment and the average variance explained was computed (Figure S4). Interestingly, we saw similar trends as observed before. Thus, even the best model shows specific deviations from perception and these deviations are preserved when the model is trained on individual experiments.

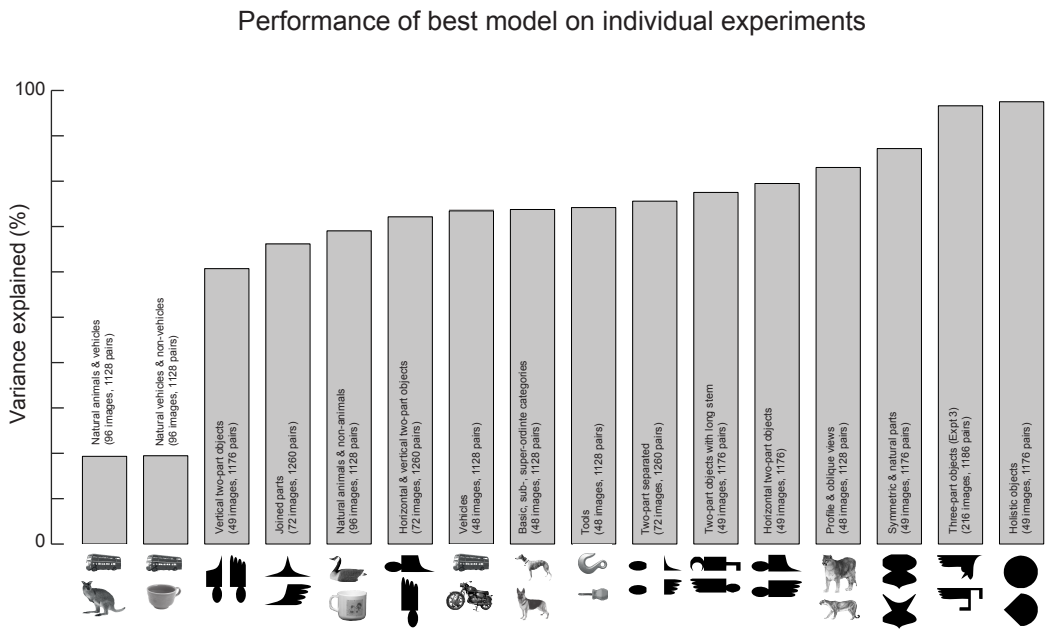

Figure S4. Best model performance on individual experiments. To investigate how well the best model (comb2) fits the data from each experiment, we selected 1000 image pairs from 16 experiments containing more than this number of pairs, and trained the model using 80-20 cross-validation as before. In the resulting plot, the bars represent the variance explained on the test set. The text inside each bar summarizes the images and image pairs used, and the image centered below each bar depicts two examples objects from each experiment.

SUPPLEMENTARY REFERENCES

Berg, A. C., Berg, T. L., and Malik, J. (2005). Shape matching and object recognition using low distortion correspondences. *Computer Vision and Pattern Recognition, 2005. CVPR 2005.*, 1:26–33.

Berg, A. C. and Malik, J. (2001). Geometric blur for template matching. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–607. IEEE.

Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. 1:886–893.

Gonzalez, R. C., Woods, R. E., and Eddins, S. L. (2004). *Digital image processing using MATLAB*. Pearson Education India.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? *IEEE 12th International Conference on Computer Vision*, pages 2146–2153.

Kay, K. N., Naselaris, T., Prenger, R. J., and Gallant, J. L. (2008). Identifying natural images from human brain activity. *Nature*, 452:352–355.

Leeds, D. D., Seibert, D. A., Pyles, J. A., and Tarr, M. J. (2013). Comparing visual representations across human fmri and computational vision. *J. Vis.*, 13(13):1–27.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Mohan, K. and Arun, S. P. (2012). Similarity relations in visual search predict rapid visual categorization. *J. Vis.*, 12(11):1–24.

Mokhtarian, F. (1995). Silhouette-based isolated object recognition through curvature scale space. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(5):539–544.

Mokhtarian, F. and Mackworth, A. (1986). Scale-based description and recognition of planar curves and two-dimensional shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1):34–43.

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175.

Pinto, N., Cox, D. D., and DiCarlo, J. J. (2008). Why is real-world visual object recognition hard? *PLoS Comput. Biol.*, 4(1):e27.

Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nat. Neurosci.*, 2(11):1019–25.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sripati, A. P. and Olson, C. R. (2010). Global image dissimilarity in macaque inferotemporal cortex predicts human visual search efficiency. *J. Neurosci.*, 30(4):1258–1269.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612.

Zahn, C. T. and Roskies, R. Z. (1972). Fourier descriptors for plane closed curves. *Computers, IEEE Transactions on*, 100(3):269–281.