Contributions of Shape, Texture, and Color in Visual Recognition

Yunhao Ge(✉), Yao Xiao, Zhi Xu, Xingrui Wang, and Laurent Itti

University of Southern California, Los Angeles, USA
yunhaoge@usc.edu
https://github.com/gyhandy/Humanoid-Vision-Engine

Abstract. We investigate the contributions of three important features of the human visual system (HVS)—shape, texture, and color—to object classification. We build a humanoid vision engine (HVE) that explicitly and separately computes shape, texture, and color features from images. The resulting feature vectors are then concatenated to support the final classification. We show that HVE can summarize and rank-order the contributions of the three features to object recognition. We use human experiments to confirm that both HVE and humans predominantly use some specific features to support the classification of specific classes (e.g., texture is the dominant feature to distinguish a zebra from other quadrupeds, both for humans and HVE). With the help of HVE, given any environment (dataset), we can summarize the most important features for the whole task (task-specific; e.g., color is the most important feature overall for classification with the CUB dataset), and for each class (class-specific; e.g., shape is the most important feature to recognize boats in the iLab-20M dataset). To demonstrate more usefulness of HVE, we use it to simulate the open-world zero-shot learning ability of humans with no attribute labeling. Finally, we show that HVE can also simulate human imagination ability with the combination of different features.

1 Introduction

The human vision system (HVS) is the gold standard for many current computer vision algorithms, on various challenging tasks: zero/few-shot learning [31,35,40,48,50], meta-learning [2,29], continual learning [43,52,57], novel view imagination [16,59], etc. Understanding the mechanism, function, and decision pipeline of HVS becomes more and more important. The vision systems of humans and other primates are highly differentiated. Although HVS provides us a unified image of the world around us, this picture has multiple facets or features, like shape, depth, motion, color, texture, etc. [15,22]. To understand the

Y. Ge and Y. Xiao—Contributed equally.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19775-8_22.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13672, pp. 369–386, 2022.
https://doi.org/10.1007/978-3-031-19775-8_22

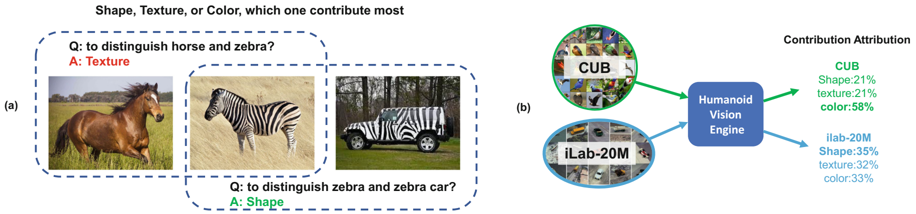


Fig. 1. (a): Contributions of Shape, Texture, and Color may be different among different scenarios/tasks. Here, texture is most important to distinguish zebra from horse, but shape is most important for zebra vs. zebra car. (b): Humanoid Vision Engine takes dataset as input and summarizes how shape, texture, and color contribute to the given recognition task in a pure learning manner (E.g., In ImageNet classification, shape is the most discriminative feature and contributes most to visual recognition).

contributions of the most important three features—shape, texture, and color—in visual recognition, some research compares the HVS with an artificial convolutional Neural Network (CNN). A widely accepted intuition about the success of CNNs on perceptual tasks is that CNNs are the most predictive models for the human ventral stream object recognition [7,58]. To understand which feature is more important for CNN-based recognition, recent paper shows promising results: ImageNet-trained CNNs are biased towards texture while increasing shape bias improves accuracy and robustness [32].

Due to the superb success of HVS on various complex tasks [2,18,35,43,59], human bias may also represent the most efficient way to solve vision tasks. And it is likely task-dependent (Fig. 1). Here, inspired by HVS, we wish to find a general way to understand how shape, texture, and color contribute to a recognition task by pure data-driven learning. The summarized feature contribution is important both for the deep learning community (guide the design of accuracy-driven models [6,14,21,32]) and for the neuroscience community (understanding the contributions or biases in human visual recognition) [33,56].

It has been shown by neuroscientists that there are separate neural pathways to process these different visual features in primates [1,11]. Among the many kinds of features crucial to visual recognition in humans, the shape property is the one that we primarily rely on in static object recognition [15]. Meanwhile, some previous studies show that surface-based cues also play a key role in our vision system. For example, [20] shows that scene recognition is faster for color images compared with grayscale ones and [36,38] found a special region in our brain to analyze textures. In summary, [8,9] propose that shape, color and texture are three separate components to identify an object.

To better understand the task-dependent contributions of these features, we build a Humanoid Vision Engine (HVE) to simulate HVS by explicitly and separately computing shape, texture, and color features to support image classification in an objective learning pipeline. HVE has the following key contributions: (1) Inspired by the specialist separation of the human brain on different features

[1,11], for each feature among shape, texture, and color, we design a specific feature extraction pipeline and representation learning model. (2) To summarize the contribution of features by end-to-end learning, we design an interpretable humanoid Neural Network (HNN) that aggregates the learned representation of three features and achieves object recognition, while also showing the contribution of each feature during decision. (3) We use HVE to analyze the contribution of shape, texture, and color on three different tasks subsampled from ImageNet. We conduct human experiments on the same tasks and show that both HVE and humans predominantly use some specific features to support object recognition of specific classes. (4) We use HVE to explore the contribution, relationship, and interaction of shape, texture, and color in visual recognition. Given any environment (dataset), HVE can summarize the most important features (among shape, texture, and color) for the whole task (task-specific) and for each class (class-specific). To the best of our knowledge, we provide the first fully objective, data-driven, and indeed first-order, quantitative measure of the respective contributions. (5) HVE can help guide accuracy-driven model design and performs as an evaluation metric for model bias. For more applications, we use HVE to simulate the open-world zero-shot learning ability of humans which needs no attribute labels. HVE can also simulate human imagination ability across features.

2 Related Works

In recent years, more and more researchers focus on the interpretability and generalization of computer vision models like CNN [23,46] and vision transformer [12]. For CNN, many researchers try to explore what kind of information is most important for models to recognize objects. Some paper show that CNNs trained on the ImageNet are more sensitive to texture information [6,14,21]. But these works fail to quantitatively explain the contribution of shape, texture, color as different features, comprehensively in various datasets and situations. While most recent studies focus on the bias of Neural Networks, exploring the bias of humans or a humanoid learning manner is still under-explored and inspiring.

Besides, many researchers contribute to the generalization of computer vision models and focus on zero/few-shot learning [10,17,31,35,48,54], novel view imagination [16,19,59], open-world recognition [3,26,27], etc. Some of them tackled these problems by feature learning—representing an object by different features, and made significant progress in this area [37,53,59]. But, there still lacks a clear definition of what these properties look like or a uniform design of a system that can do humanoid tasks like generalized recognition and imagination.

3 Humanoid Vision Engine

The goal of the humanoid vision engine (HVE) is to summarize the contribution of shape, texture, and color in a given task (dataset) by separately computing the three features to support image classification, similar to humans' recognizing

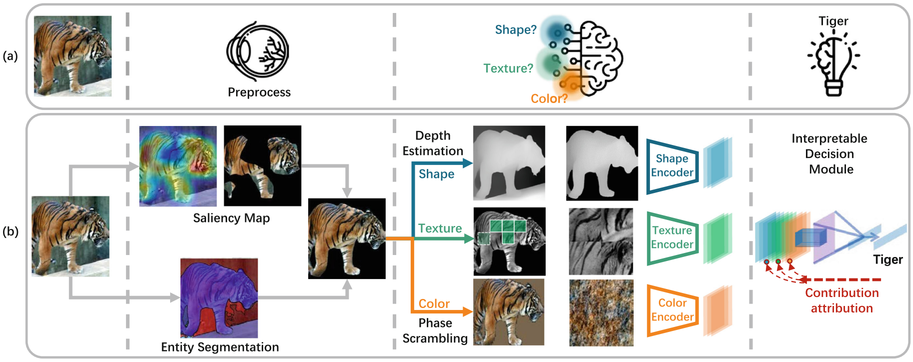


Fig. 2. Pipeline for humanoid vision engine (HVE). (a) shows how will humans' vision system deal with an image. After humans' eyes perceive the object, the different parts of the brain will be activated. The human brain will organize and summarize that information to get a conclusion. (b) shows how we design HVE to correspond to each part of the human's vision system.

objects. During the pipeline and model design, we borrow the findings of neuroscience on the structure, mechanism and function of HVS [1,11,15,20,36,38]. We use end-to-end learning with backpropagation to simulate the learning process of humans and to summarize the contribution of shape, texture, and color. The advantage of end-to-end training is that we can avoid human bias, which may influence the objective of contribution attribution (e.g., we avoid handcrafted elementary shapes as done in Recognition by Components [4]). We only use data-driven learning, a straightforward way to understand the contribution of each feature from an effectiveness perspective, and we can easily generalize HVE to different tasks (datasets). As shown in Fig. 2, HVE consists of (1) **a humanoid image preprocessing pipeline**, (2) **feature representation** for shape, texture, and color, and (3) **a humanoid neural network** that aggregates the representation of each feature and achieves interpretable object recognition.

3.1 Humanoid Image Preprocessing and Feature Extraction

As shown in Fig. 2 (a), humans (or primates) can localize an object intuitively in a complex scene before we recognize what it is [28]. Also, there are different types of cells or receptors in our primary visual cortex extracting specific information (like color, shape, texture, shading, motion, etc.) information from the image [15]. In our HVE (Fig. 2 (b)), for an input raw image $I \in \mathbb{R}^{H\times W\times C}$, we first parse the object from the scene as preprocessing and then extract our defined shape, texture, and color features I_s, I_t, I_c, for the following humanoid neural network.

Image Parsing and Foreground Identification. As shown in the preprocessing part of Fig. 2 (b), we use the entity segmentation method [39] to simulate the process of parsing objects from a scene in our brain. Entity segmentation is an open-world model and can segment the object from the image without labels. This method aligns with human behavior, which can (at least in some cases; e.g., autostereograms [28]) segment an object without deciding what it is. After we get the segmentation of the image, we use a pre-trained CNN and GradCam [45] to find the foreground object among all masks. (More details in appendix.)

We design three different feature extractors after identifying the foreground object segment: shape, texture, and color extractor, similar to the separate neural pathways in the human brain which focus on specific property [1,11]. The three extractors focus only on the corresponding features, and the extracted features, shape I_s, texture I_t, and color I_c, are disentangled from each other.

Shape Feature Extractor. For the shape extractor, we want to keep both 2D and 3D shape information while eliminating the information of texture and color. We first use a 3D depth prediction model [41,42] to obtain the 3D depth information of the whole image. After element-wise multiplying the 3D depth estimation and 2D mask of the object, we obtain our shape feature I_s. We can notice that this feature only contains 2D shape and 3D structural information (the 3D depth) and without color or texture information (Fig. 2(b)).

Texture Feature Extractor. In texture extractor, we want to keep both local and global texture information while eliminating shape and color information. Figure 3 visualizes the extraction process. First, to remove the color information, we convert the RGB object segmentation to a grayscale image. Next, we cut this image into several square patches with an adaptive strategy (the patch size and location are adaptive with object sizes to cover more texture information). If the overlap ratio between the patch and the original 2D object segment is larger than a threshold τ, we add that patch to a patch pool (we set τ to be 0.99 in our experiments, which means the over 99% of the area of the patch belongs to the object). Since we want to extract both local (one patch) and global (whole image) texture information, we randomly select 4 patches from the patch pool and concatenate them into a new texture image (I_t). (More details in appendix.)

Color Feature Extractor. To represent the color feature for I. We use phase scrambling, which is popular in psychophysics and in signal processing [34,51]. Phase scrambling transforms the image into the frequency domain using the fast Fourier transform (FFT). In the frequency domain, the phase of the signal is

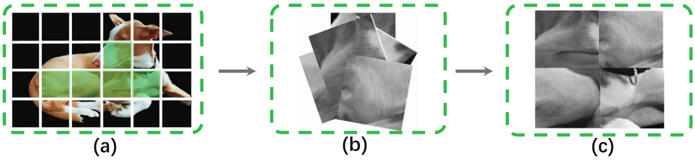


Fig. 3. Pipeline for extracting texture feature: (a) Crop images and compute the overlap ratio between 2D mask and patches. Patches with overlap > 0.99 are shown in a green shade. (b) add the valid patches to a patch pool. (c) randomly choose 4 patches from pool and concatenate them to obtain a texture image I_t. (Color figure online)

then randomly scrambled, which destroys shape information while preserving color statistics. Then we use IFFT to transfer back to image space and get $I_c \in \mathbb{R}^{H \times W \times C}$. I_c and I have the same distribution of pixel color values (Fig. 2(b)).

3.2 Humanoid Neural Network

After preprocessing, we have three features, i.e. shape I_s, texture I_t, color I_c of an input image I. To simulate the separate neural pathways in humans' brains for different feature information [1,11], we design three feature representation encoders for shape, texture, and color, respectively. Shape feature encoder E_s takes a 3D shape feature I_s as input and outputs the shape representation ($V_s = E_s(I_s)$). Similarly, texture encoder E_t and color encoder E_c take the texture patch image I_t or color phase scrambled image I_c as input, after embedded by E_t (or E_c), we get the texture feature V_t and color feature V_c. We use ResNet-18 [23] as the backbone for all feature encoders to project the three types of features to the corresponding well-separated embedding spaces. It is hard to define the ground-truth label of the distance between features. Given that the objects from the same class are relatively consistent in shape, texture, and color, the encoders can be trained in the classification problem independently instead, with the supervision of class labels. After training our encoders as classifiers, the feature map of the last convolutional layer will serve as the final feature representation. To aggregate separated feature representations and conduct object recognition, we freeze the three encoders and train a contribution interpretable aggregation module Aggr_θ, which is composed of two fully-connected layers (Fig. 2 (b) right). We concatenate V_s, V_t, V_c and send it to Aggr_θ. The output is denoted as $p \in \mathbb{R}^n$, where n is the number of classes. So we have $p = \text{Aggr}_\theta\left(\text{concat}(V_s, V_t, V_c)\right)$. (More details and exploration of our HNN are in appendix.)

We also propose a gradient-based *contribution attribution* method to interpret the contributions of shape, texture, and color to the classification decision, respectively. Take the shape feature as an example, given a prediction p and the probability of class k, namely p^k, we compute the gradient of p^k with respect to the shape feature V^s. We define the gradient as shape importance weights α_s^k, i.e. $\alpha_s^k = \frac{\partial p^k}{\partial V_s}, \alpha_t^k = \frac{\partial p^k}{\partial V_t}, \alpha_c^k = \frac{\partial p^k}{\partial V_c}$. Then we calculate element-wise product between V_s and α_s^k to get the final shape contribution S_s^k, i.e. $S_s^k = \text{ReLU}\left(\sum \alpha_s^k V_s\right)$. In other words, S_s^k represents the "contribution" of shape feature to classifying this image as class k. We can do the same thing to get texture contribution S_t^k and color contribution S_c^k. After getting the feature contributions for each image, we can calculate the average value of all images in this class to assign feature contributions to each class (class-specific bias) and the average value of all classes to assign feature contributions to the whole dataset (task-specific bias).

4 Experiments

In this section, we first show the effectiveness of feature encoders on representation learning (Sect. 4.1); then we show the contribution interpretation

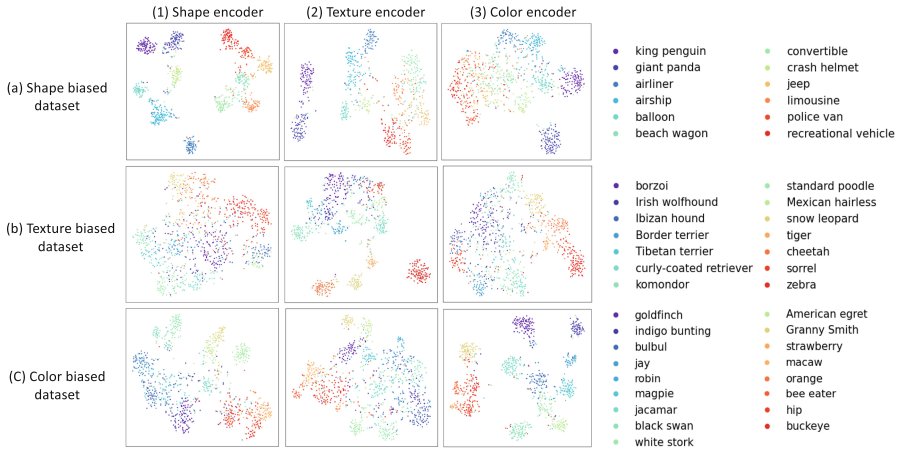


Fig. 4. T-SNE results of feature encoders on their corresponding biased datasets

performance of Humanoid NN on different feature-biased datasets in ImageNet (Sect. 4.2); We use human experiments to confirm that both HVE and humans predominantly use some specific features to support the classification of specific classes (Sect. 4.3); Then we use HVE to summarize the contribution of shape, texture, and color on different datasets (CUB [55] and iLab-20M [5]) (Sect. 4.4).

4.1 Effectiveness of Feature Encoders

To show that our three feature encoders focus on embedding their corresponding sensitive features, we handcrafted three subsets of ImageNet [30]: shape-biased dataset (D_{shape}), texture-biased dataset (D_{texture}), and color-biased dataset (D_{color}). **Shape-biased dataset** containing 12 classes, where the classes were chosen which intuitively are strongly determined by shape (e.g., vehicles are defined by shape more than color). **Texture-biased dataset** uses 14 classes which we believed are more strongly determined by texture. **Color-biased dataset** includes 17 classes. The intuition of class selection of all three datasets will be verified by our results in Table 1 with further illustration in Sect. 4.2. All these datasets are randomly selected as around 800 training images and 200 testing images. The class details of biased datasets are shown in Fig. 4.

If our feature extractors actually learned their *feature-constructive* latent spaces, their T-SNE results will show clear clusters in the feature-biased datasets. "Bias" here means we can classify the objects based on the biased feature easily, but it is more difficult to make decisions based on the other two features.

After pre-processing the original images and getting their feature images, we input the feature images into feature encoders and get the T-SNE results shown in Fig. 4. Each row represents one feature-biased dataset and each column is bounded with one feature encoder, each image shows the results of one combination. T-SNE results are separated perfectly on corresponding datasets

Table 1. "Original" column means the accuracy of Resnet18 on the original images as our upper bound. Shape, texture and color columns represent the accuracy of feature nets. "all" means results of our HNN that combines the 3 feature nets. It approaches the upper bound, suggesting that the split into 3 feature nets preserved most information needed for image classification.

Accuracy	**Original**	Shape	Texture	Color	**All**
Shape biased dataset	**97%**	**90%**	84%	71%	**95%**
Texture biased dataset	**96%**	64%	**81%**	65%	**91%**
Color biased dataset	**95%**	70%	73%	**82%**	**92%**

(diagonal) but not as well on others' datasets (off-diagonal), which shows that our feature encoders are predominantly sensitive to the corresponding features.

4.2 Effectiveness of Humanoid Neural Network

We can use feature encoders to serve as classifiers after adding fully-connected layers. As these classifiers classify images based on corresponding feature representation, we call them *feature nets*. We tested the accuracy of feature nets on these three biased datasets. As shown in Table 1, a ResNet-18 trained on the original segmented images (without explicit separated features, e.g. Fig. 2 (b) tiger without background) provided an upper bound for the task. We find that feature net consistently obtains the best performance on their own biased dataset (e.g., on the shape-biased dataset, shape net classification performance is better than that of the color net or texture net). If we combine these three feature nets with the interpretable aggregation module, the classification accuracy is very close to the upper bound, which means our vision system can classify images based on these three features almost as well as based on the full original color images. This demonstrates that we can obtain most information of original images by our feature nets, and our aggregation and interpretable decision module actually learned how to combine those three features by end-to-end learning.

Table 2a shows the quantitative contribution summary results of Humanoid NN (Sect. 3.2). For task-specific bias, shape plays a dominant role in shape-biased tasks, and texture, color also contribute most to their related biased tasks.

4.3 Human Experiments

Intuitively, we expect that humans may rely on different features to classify different objects (Fig. 1). To show this, we designed human experiments that asked participants to classify reduced images with only shape, texture, or color features. If an object is mainly recognizable based on shape for humans, we could then check whether it is also the same for HVE, and also for color and texture.

Experiments Design. Three datasets in Table 1 have a clear bias towards corresponding features (Fig. 4). We asked the participants to classify objects in

Table 2. Contributions of features from HVE and humans' recognition accuracy.

(a) Contributions of features for different biased datasets summarized by HVE.

Contribution ratio	Shape	Texture	Color
Shape biased dataset	**47%**	34%	19%
Texture biased dataset	5%	**65%**	30%
Color biased dataset	11%	19%	**70%**

(b) Humans' accuracy of different feature images on different biased datasets.

Accuracy	Shape	Texture	Color
Shape biased dataset	**90.0%**	49.0%	16.8%
Texture biased dataset	33.1%	**40.0%**	11.1%
Color biased dataset	32.3%	19.7%	**46.5%**

each dataset based on one single feature image computed by one of our feature extractors (Fig. 5). Participants were asked to choose the correct class label for the reduced image (from 12/14/17 classes in shape/texture/color datasets).

Human Performance Results. The results here are based on 3270 trials, 109 participants. The accuracy for different feature questions on different biased datasets can be seen in Table 2b. Human performance is similar to our feature nets' performance (compare Table 1 with Table 2b). On shape-biased dataset, both human and feature nets attain the highest accuracy with shape. The same for the color and texture biased datasets. Both HVE and humans predominantly use some specific features to support recognition of specific classes. Interestingly, humans can perform not badly on all three biased datasets with shape features.

4.4 Contributions Attribution in Different Tasks

With our vision system, we can summarize the task-specific bias and class-specific bias for any dataset. This enables several applications: (1) Guide accuracy-driven model design [6,14,21,32]; Our method provides objective summarization of dataset bias. (2) Evaluation metric for model bias. Our method can help correct an initially wrong model bias on some datasets (e.g., that most CNN trained on ImageNet are texture biased [21,32]). (3) Substitute human intuition to obtain more objective summarization with end-to-end learning. We implemented the biased summarization experiments on two datasets, CUB [55] and iLab-20M [5]. Figure 1(b) shows the task-specific biased results. Since CUB is a dataset of birds, which means all the classes in CUB have a similar shape

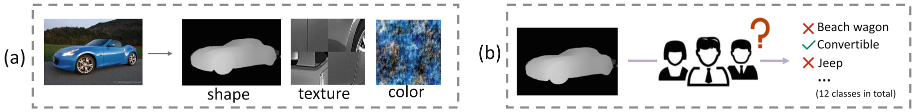


Fig. 5. Sample question for the human experiment. (a) A test image (left) is first converted into shape, color, and texture images using our feature extractors. (b) On a given trial, human participants are presented with one shape, color, or texture image, along with 2 reference images for each class in the corresponding dataset (not shown here, see appendix. For a screenshot of an experiment trial). Participants are asked to guess the correct object class from the feature image.

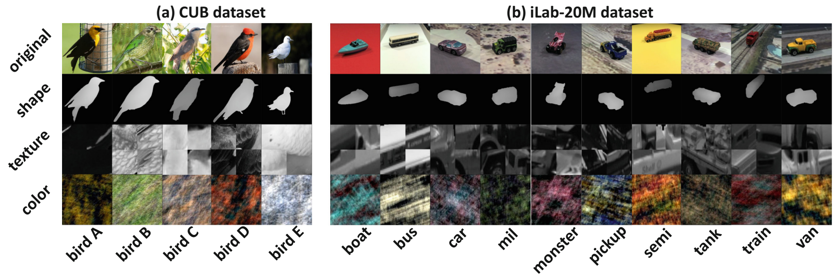


Fig. 6. Processed CUB and iLab-20M dataset examples

with feather textures, hence color may indeed be the most discriminative feature (Fig. 6 (a)).

As for iLab (Fig. 6 (b)), we also conduct the class-specific biased experiments on iLab and summarize the class biases in Table 3. It is interesting to find that the dominant feature is different for different classes. For instance, boat is shape-biased while military vehicle (mil) is color-biased. (More examples in appendix.)

5 More Humanoid Applications with HVE

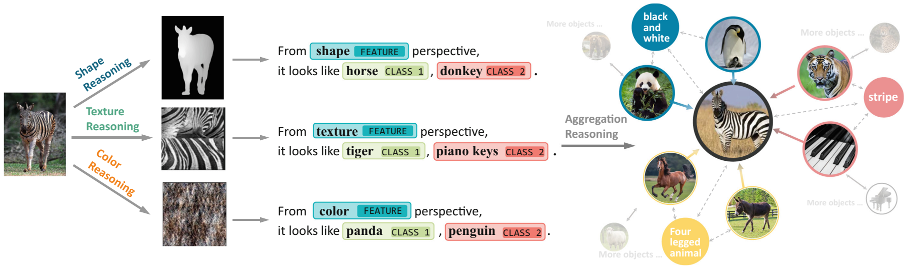
To further explore more applications with HVE, we use HVE to simulate the visual reasoning process of humans and propose a new solution for conducting open-world zero-shot learning without predefined attribute labels (Sect. 5.1). We also use HVE to simulate human imagination ability through cross-feature retrieval and imagination (Sect. 5.2).

5.1 Open-World Zero-Shot Learning with HVE

Zero-shot learning needs to classify samples from classes never seen during training. Most current methods [13,31,35] need humans to provide detailed attribute labels for each image, which is costly in time and energy. However, given an image from an unseen class, humans can still *describe* it with their learned knowledge. For example, we may use horse-like shape, panda-like color, and tiger-like texture to describe an unseen class zebra. In this section, we show how our HVE can simulate this feature-wise open-world image description by feature retrieval

Table 3. Class-specific bias for each class in iLab-20M

Ratio	Boat	Bus	Car	Mil	Monster	Pickup	Semi	Tank	Train	Van
Shape	**40%**	35%	**44%**	18%	36%	28%	**40%**	36%	31%	**40%**
Texture	32%	31%	**40%**	30%	34%	20%	31%	32%	34%	27%
Color	28%	34%	16%	**52%**	30%	**53%**	29%	32%	35%	33%


Shape Reasoning
Texture Reasoning
Color Reasoning
From shape FEATURE perspective,
it looks like horse CLASS 1 , donkey CLASS 2 .
From texture FEATURE perspective,
it looks like tiger CLASS 1 , piano keys CLASS 2 .
From color FEATURE perspective,
it looks like panda CLASS 1 , penguin CLASS 2 .
Aggregation Reasoning
black and white
stripe
Four legged animal
More objects ...
More objects ...
More objects ...
More objects ...

(a) Open-world Image Description (b) Reasoning for Zero-shot Learning

Fig. 7. The zero-shot learning method with HVE. We first describe the novel image in the perspective of shape, texture, and color. Then we use ConceptNet as common knowledge to reason and predict the label.

and ranking. And based on these image descriptions, we propose a feature-wise open-world zero-shot learning pipeline with the help of ConceptNet [49], like the *reasoning or consulting* process of humans. The whole process shows in Fig. 7.

Step 1: Description. We use HVE to provide feature-wise descriptions for any unseen class images without predefined attribute labels. First, to represent learnt knowledge, we use trained three feature extractors (described in Sect. 3.2) to get the shape, texture, and color representation image of seen class k. Then, given an unseen class image I_{un}, we use the same feature extractors to get its feature-wise representation. To retrieve learnt classes as descriptions, we calculate the average distance between I_{un} and images of other class k in the latent space on shape, texture, and color features. In this way, we can find the top K closest classes of I_{un} from the perspective of each feature, and we call these K classes "roots" of each feature. Now, we can describe I_{un} using our three sets of roots. For example, as shown in Fig. 7(a), for the unseen class zebra, we can describe its shape by $\{\text{horse}, \text{donkey}\}$, texture by $\{\text{tiger}, \text{piano keys}\}$, and color by $\{\text{panda}\}$.

Step 2: Open-world classification. To further predict the actual class of I_{un} based on the feature-wise description, we use ConceptNet as common knowledge to conduct reasoning. As shown in Fig. 7(b), for every feature roots, we retrieve their common attribute in ConceptNet, (e.g., stripe the is common attribute root of $\{\text{tiger}, \text{piano keys}\}$). We form a reasoning root pool R^* consisting of classes from feature roots obtained during image description and shared attribute roots. The reasoning roots will be our *evidence* for reasoning. For every root in R^*, we can search its neighbors in ConceptNet, which are treated as possible candidate classes for I_{un}. All candidates form a possible candidate pool P, which contains all hypothesis classes. Now we have two pools, root pool R^* and candidate pool P. For every candidate $p_i \in P$ and $r_i \in R^*$, we calculate the ranking score of p_i as: $\bar{S}(p_i) = \sum_{r_j \in R^*} \cos(\mathcal{E}(p_i), \mathcal{E}(r_j))$. where $\mathcal{E}(\cdot)$ is the word embedding in ConceptNet and $\cos(A, B)$ means cosine similarity between A and B.

Table 4. Open-world zero-shot accuracy and FID of cross-features imagination.

(a) Accuracy of unseen class for zero-shot learning. One-shot on Prototype and zero-shot on ours

Method	Fowl	Zebra	Wolf	Sheep	Apple
Prototype	19%	16%	17%	21%	74%
Ours	78%	87%	63%	72%	98%

(b) Cross-features imagination quality comparison. We compare HVE methods with three pix2pix GANs as baselines

FID (↓)	Shape input	Texture input	Color input
Baselines	123.915	188.854	203.527
Ours	**96.871**	**105.921**	**52.846**

We choose the candidate with the highest score as our predicted label. In our prototype zero-shot learning dataset, we select 34 seen classes as the training set and 5 unseen classes as the test set, with 200 images per class. We calculate the accuracy of the test set (Table 4a). As a comparison, we conduct prototypical networks [47] using its one-shot setting. More details are in the appendix.

5.2 Cross Feature Imagination with HVE

We show HVE has the potential to simulate human imagination ability. Humans can intuitively imagine an object when seeing one aspect of a feature, especially when this feature is prototypical (contribute most to classification). For instance, we can imagine a zebra when seeing its stripe (texture). This process is similar but harder than the classical image generation task since the input features modality here is *dynamic* which can be any feature among shape, texture, or color. To solve this problem, using HVE, we separate this procedure into two steps: (1) **cross feature retrieval** and (2) **cross feature imagination**. Given any feature (shape, texture, or color) as input, cross-feature retrieval finds the most possible two other features. Cross-feature imagination then generate a whole object based on a group of shapes, textures, and color features.

Cross Feature Retrieval. We learn a feature agnostic encoder that projects the three features into one same feature space and makes sure that the features belonging to the same class are in the nearby regions.

As shown in Fig. 8(a), during training, the shape I_s, texture I_t and color I_c are first sent into the corresponding frozen encoders E_s, E_t, E_c, which are the same encoders in Sect. 3.2. Then all of the outputs are projected into a cross-feature embedding space by a feature agnostic net $\mathcal{M}$, which contains three convolution layers. We also add a fully connected layer to predict the class labels of the features. We use cross-entropy loss $\mathcal{L}_{\text{cls}}$ to regularize the prediction label and a triplet loss $\mathcal{L}_{\text{triplet}}$ [44] to regularize the projection of $\mathcal{M}$. For any input feature x (e.g., a bird A shape), positive sample x_{pos} are either same class same modality (another bird A shape) or same class different feature modality (a bird A texture or color); negative sample x_{neg} are any features from different class. $\mathcal{L}_{\text{triplet}}$ pulls the embedding of x closer to that of the positive sample x_{pos}, and pushes it apart from the embedding of the negative sample x_{neg}. The triplet

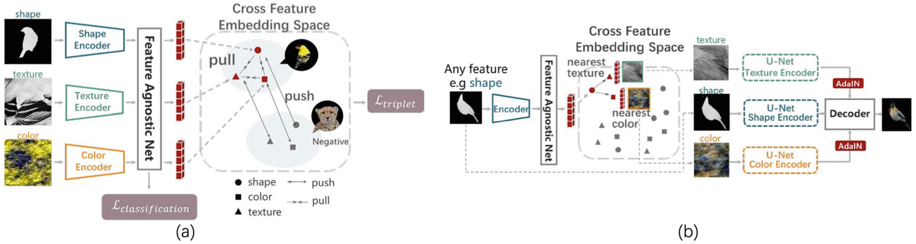


Fig. 8. (a) The structure and training process of the cross-feature retrieval model. E_s, E_t, E_c are the same encoders in Sect. 3.2. The feature agnostic net then projects them to shared feature space for retrieval. (b) The process of cross-feature imagination. After retrieval, we design a cross-feature pixel2pixel GAN model to generate the final image.

loss is defined as $\mathcal{L}_{\text{triplet}} = \max(\|\mathcal{F}(x) - \mathcal{F}(x_{\text{pos}})\|_2 - \|\mathcal{F}(x) - \mathcal{F}(x_{\text{neg}})\|_2 + \alpha, 0)$, where $\mathcal{F}(\cdot) := \mathcal{M}(E(\cdot))$, E is one of the feature encoders. α is the margin size in the feature space between classes, $\|\cdot\|_2$ represents ℓ_2 norm.

We test the retrieval model in all three biased datasets (Fig. 4) separately. During retrieval, given any feature of any object, we can map it into the cross feature embedding space by the corresponding encoder net and the feature agnostic net. Then we apply the ℓ_2 norm to find the other two features closest to the input one as output. The output is correct if they belong to the same class as the input. For each dataset, we retrieve the three features pair by pair (accuracy in appendix). The retrieval performs better when the input feature is the dominant of the dataset, which again verifies the feature bias in each dataset.

Cross Feature Imagination. To stimulate imagination, we propose a cross-feature imagination model to generate plausible final images with the input and retrieved features. The procedure of imagination is shown in Fig. 8(b). Inspired by the pixel2pixel GAN [25] and AdaIN [24], we design a cross-feature pixel2pixel GAN model to generate the final image. The GAN model is trained and tested on the three biased datasets. In Fig. 9, we show more results of the generation, which show that our model satisfyingly generates the object from a single feature. From the comparison between (c) and (e), we can clearly find that they are alike from the view of the corresponding input feature, but the imagination results preserve the retrieval features. The imagination variance also shows the feature contributions from a generative view: if the given feature is the dominant feature of a class (contribute most in classification. e.g., the stripe of zebra), then the retrieved features and imagined images have smaller variance (most are zebras); While non-dominant given feature (shape of zebra) lead to large imagination variance (can be any horse-like animals). We create a baseline generator by using three pix2pix GANs where each pix2pix GAN is responsible for one specific feature (take one modality of feature as input and imagine the raw image). The FID comparison is in Table 4b. More details are in the appendix.

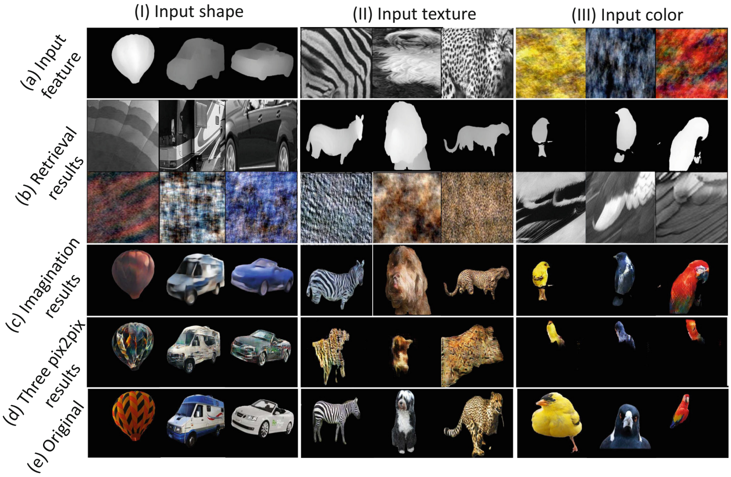


Fig. 9. Imagination with shape, texture, and color feature input (columns I, II, III). Line (a): input feature. Line (b): retrieved features given (a). Line (c): imagination results with HVE and our GAN model. Line (d): results of baseline 3 pix2pix GANs. Line (e): original images to which the input features belong. Our model can reasonably "imagine" the object given a single feature.

6 Conclusion

To explore the task-specific contribution of shape, texture, and color features in human visual recognition, we propose a humanoid vision engine (HVE) that explicitly and separately computes these features from images and then aggregates them to support image classification. With the proposed contribution attribution method, given any task (dataset), HVE can summarize and rank-order the task-specific contributions of the three features to object recognition. We use human experiments to show that HVE has a similar feature contribution to humans on specific tasks. We show that HVE can help simulate more complex and humanoid abilities (e.g., open-world zero-shot learning and cross-feature imagination) with promising performance. These results are the first step towards better understanding the contributions of object features to classification, zero-shot learning, imagination, and beyond.

Acknowledgments. This work was supported by C-BRIC (one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA), DARPA (HR00112190134) and the Army Research Office (W911NF2020053). The authors affirm that the views expressed herein are solely their own, and do not represent the views of the United States government or any agency thereof.

References

1. Amir, Y., Harel, M., Malach, R.: Cortical hierarchy reflected in the organization of intrinsic connections in macaque monkey visual cortex. J. Comp. Neurol. **334**(1), 19–46 (1993)
2. Andrychowicz, M., et al.: Learning to learn by gradient descent by gradient descent. In: Advances in Neural Information Processing Systems, pp. 3981–3989 (2016)
3. Bendale, A., Boult, T.: Towards open world recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1893–1902 (2015)
4. Biederman, I.: Recognition-by-components: a theory of human image understanding. Psychol. Rev. **94**(2), 115 (1987)
5. Borji, A., Izadi, S., Itti, L.: ilab-20m: A large-scale controlled object dataset to investigate deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2221–2230 (2016)
6. Brendel, W., Bethge, M.: Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019. OpenReview.net (2019). www.openreview.net/forum?id=SkfMWhAqYQ
7. Cadieu, C.F., et al.: Deep neural networks rival the representation of primate it cortex for core visual object recognition. PLoS Comput. Biolo. **10**(12), e1003963 (2014)
8. Cant, J.S., Goodale, M.A.: Attention to form or surface properties modulates different regions of human occipitotemporal cortex. Cereb. Cortex **17**(3), 713–731 (2007)
9. Cant, J.S., Large, M.E., McCall, L., Goodale, M.A.: Independent processing of form, colour, and texture in object perception. Perception **37**(1), 57–78 (2008)
10. Cheng, H., Wang, Y., Li, H., Kot, A.C., Wen, B.: Disentangled feature representation for few-shot image classification. CoRR abs/2109.12548 (2021). arxiv.org/abs/2109.12548
11. DeYoe, E.A., et al.: Mapping striate and extrastriate visual areas in human cerebral cortex. Proc. Natl. Acad. Sci. **93**(6), 2382–2386 (1996)
12. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021. OpenReview.net (2021). www.openreview.net/forum?id=YicbFdNTTy
13. Fu, Y., Xiang, T., Jiang, Y.G., Xue, X., Sigal, L., Gong, S.: Recent advances in zero-shot recognition: toward data-efficient understanding of visual content. IEEE Sig. Process. Mag. **35**(1), 112–125 (2018)
14. Gatys, L.A., Ecker, A.S., Bethge, M.: Texture and art with deep neural networks. Curr. Opin. Neurobiol. **46**, 178–186 (2017)
15. Gazzaniga, M.S., Ivry, R.B., Mangun, G.: Cognitive Neuroscience. The Biology of the Mind (2014) (2006)
16. Ge, Y., Abu-El-Haija, S., Xin, G., Itti, L.: Zero-shot synthesis with group-supervised learning. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021. OpenReview.net (2021). www.openreview.net/forum?id=8wqCDnBmnrT
17. Ge, Y., Xiao, Y., Xu, Z., Li, L., Wu, Z., Itti, L.: Towards generic interface for human-neural network knowledge exchange (2021)

18. Ge, Y., et al.: A peek into the reasoning of neural networks: interpreting with structural visual concepts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2195–2204 (2021)
19. Ge, Y., Xu, J., Zhao, B.N., Itti, L., Vineet, V.: Dall-e for detection: language-driven context image synthesis for object detection. arXiv preprint arXiv:2206.09592 (2022)
20. Gegenfurtner, K.R., Rieger, J.: Sensory and cognitive contributions of color to the recognition of natural scenes. Curr. Biol. **10**(13), 805–808 (2000)
21. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019. OpenReview.net (2019). www.openreview.net/forum?id=Bygh9j09KX
22. Grill-Spector, K., Malach, R.: The human visual cortex. Annu. Rev. Neurosci. **27**, 649–677 (2004)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
24. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1501–1510 (2017)
25. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
26. Jain, L.P., Scheirer, W.J., Boult, T.E.: Multi-class open set recognition using probability of inclusion. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8691, pp. 393–409. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10578-9_26
27. Joseph, K., Khan, S., Khan, F.S., Balasubramanian, V.N.: Towards open world object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5830–5840 (2021)
28. Julesz, B.: Binocular depth perception without familiarity cues: random-dot stereo images with controlled spatial and temporal properties clarify problems in stereopsis. Science **145**(3630), 356–362 (1964)
29. Khodadadeh, S., Bölöni, L., Shah, M.: Unsupervised meta-learning for few-shot image classification. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8–14 December 2019, Vancouver, BC, Canada, pp. 10132–10142 (2019). www.proceedings.neurips.cc/paper/2019/hash/fd0a5a5e367a0955d81278062ef37429-Abstract.html
30. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Adv. Neural. Inf. Process. Syst. **25**, 1097–1105 (2012)
31. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 951–958. IEEE (2009)
32. Li, Y., et al.: Shape-texture debiased neural network training. arXiv preprint arXiv:2010.05981 (2020)
33. Oliva, A., Schyns, P.G.: Diagnostic colors mediate scene recognition. Cogn. Psychol. **41**(2), 176–210 (2000). https://doi.org/10.1006/cogp.1999.0728

34. Oppenheim, A.V., Lim, J.S.: The importance of phase in signals. Proc. IEEE **69**(5), 529–541 (1981)
35. Palatucci, M., Pomerleau, D., Hinton, G.E., Mitchell, T.M.: Zero-shot learning with semantic output codes. In: Bengio, Y., Schuurmans, D., Lafferty, J.D., Williams, C.K.I., Culotta, A. (eds.) Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7–10 December 2009, Vancouver, British Columbia, Canada, pp. 1410–1418. Curran Associates, Inc. (2009). www.proceedings.neurips.cc/paper/2009/hash/1543843a4723ed2ab08e18053ae6dc5b-Abstract.html
36. Peuskens, H., Claeys, K.G., Todd, J.T., Norman, J.F., Van Hecke, P., Orban, G.A.: Attention to 3-D shape, 3-D motion, and texture in 3-D structure from motion displays. J. Cogn. Neurosci. **16**(4), 665–682 (2004)
37. Prabhudesai, M., Lal, S., Patil, D., Tung, H., Harley, A.W., Fragkiadaki, K.: Disentangling 3D prototypical networks for few-shot concept learning. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021. OpenReview.net (2021). www.openreview.net/forum?id=-Lr-u0b42he
38. Puce, A., Allison, T., Asgari, M., Gore, J.C., McCarthy, G.: Differential sensitivity of human visual cortex to faces, letterstrings, and textures: a functional magnetic resonance imaging study. J. Neurosci. **16**(16), 5205–5215 (1996)
39. Qi, L., et al.: Open-world entity segmentation. arXiv preprint arXiv:2107.14228 (2021)
40. Rahman, S., Khan, S., Porikli, F.: A unified approach for conventional zero-shot, generalized zero-shot, and few-shot learning. IEEE Trans. Image Process. **27**(11), 5652–5667 (2018)
41. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. arXiv preprint (2021)
42. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: mixing datasets for zero-shot cross-dataset transfer. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) (2020)
43. Schlimmer, J.C., Fisher, D.: A case study of incremental concept induction. In: AAAI, vol. 86, pp. 496–501 (1986)
44. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 815–823 (2015)
45. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)
46. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015). www.arxiv.org/abs/1409.1556
47. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
48. Snell, J., Swersky, K., Zemel, R.S.: Prototypical networks for few-shot learning. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA, pp. 4077–4087 (2017). www.proceedings.neurips.cc/paper/2017/hash/cb8da6767461f2812ae4290eac7cbc42-Abstract.html

49. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: an open multilingual graph of general knowledge. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
50. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: relation network for few-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1199–1208 (2018)
51. Thomson, M.G.: Visual coding and the phase structure of natural scenes. Netw. Comput. Neural Syst. **10**(2), 123 (1999)
52. Thrun, S., Mitchell, T.M.: Lifelong robot learning. Robot. Auton. Syst. **15**(1–2), 25–46 (1995)
53. Tokmakov, P., Wang, Y.X., Hebert, M.: Learning compositional representations for few-shot recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6372–6381 (2019)
54. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. Adv. Neural. Inf. Process. Syst. **29**, 3630–3638 (2016)
55. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD birds-200-2011 dataset (2011)
56. Walther, D.B., Chai, B., Caddigan, E., Beck, D.M., Fei-Fei, L.: Simple line drawings suffice for functional MRI decoding of natural scene categories. Proc. Natl. Acad. Sci. **108**(23), 9661–9666 (2011). https://doi.org/10.1073/pnas.1015666108, https://www.pnas.org/doi/abs/10.1073/pnas.1015666108
57. Wen, S., Rios, A., Ge, Y., Itti, L.: Beneficial perturbation network for designing general adaptive artificial intelligence systems. IEEE Trans. Neural Netw. Learn. Syst. (2021)
58. Yamins, D.L., Hong, H., Cadieu, C.F., Solomon, E.A., Seibert, D., DiCarlo, J.J.: Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proc. Natl. Acad. Sci. **111**(23), 8619–8624 (2014)
59. Zhu, J.Y., et al.: Visual object networks: image generation with disentangled 3D representations. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 31. Curran Associates, Inc. (2018). www.proceedings.neurips.cc/paper/2018/file/92cc227532d17e56e07902b254dfad10-Paper.pdf