





# Grounding Psychological Shape Space in Convolutional Neural Networks

Lucas Bechberger<sup>(✉)</sup>  and Kai-Uwe Kühnberger 

Institute of Cognitive Science, Osnabrück University, Osnabrück, Germany  
{lbechberger, kkuehnbe}@uos.de

**Abstract.** Shape information is crucial for human perception and cognition, and should therefore also play a role in cognitive AI systems. We employ the interdisciplinary framework of conceptual spaces, which proposes a geometric representation of conceptual knowledge through low-dimensional interpretable similarity spaces. These similarity spaces are often based on psychological dissimilarity ratings for a small set of stimuli, which are then transformed into a spatial representation by a technique called multidimensional scaling. Unfortunately, this approach is incapable of generalizing to novel stimuli. In this paper, we use convolutional neural networks to learn a generalizable mapping between perceptual inputs (pixels of grayscale line drawings) and a recently proposed psychological similarity space for the shape domain. We investigate different network architectures (classification network vs. autoencoder) and different training regimes (transfer learning vs. multi-task learning). Our results indicate that a classification-based multi-task learning scenario yields the best results, but that its performance is relatively sensitive to the dimensionality of the similarity space.

**Keywords:** Psychological similarity spaces · Conceptual spaces · Shape perception · Convolutional neural networks

## 1 Introduction

Shape information plays an important role in human perception and cognition, and can be viewed as a bootstrapping device for constructing concepts [18, 33, 40]. Based on the principle of cognitive AI [42, 44], which tries to base artificial systems on insights about human cognition, also artificial agents should be equipped with a human-like representation of shapes.

In this paper, we employ the cognitive framework of conceptual spaces [24], which proposes a geometric representation of conceptual knowledge based on psychological similarity spaces. It offers a way of neural-symbolic integration [23, 46] by using an intermediate level of representation between the connectionist and the symbolic approach, which are represented by artificial neural networks and entirely rule-based systems, respectively. The overall conceptual space is structured into different cognitive domains (such as COLOR and SHAPE),

which are represented by low-dimensional psychological similarity spaces with cognitively meaningful dimensions. Conceptual spaces have seen a wide variety of applications in artificial intelligence, linguistics, psychology, and philosophy [34, 70]. Typically, the structure of a conceptual space is obtained based on dissimilarity ratings from psychological experiments, which are then translated into a spatial representation through multidimensional scaling [14]. In this paper, we consider a recently proposed similarity space for the SHAPE domain [9–11].

The similarity spaces obtained by multidimensional scaling are not able to generalize to unseen inputs – a novel stimulus can only be mapped into the similarity space after eliciting further dissimilarity ratings [6]. In order to generalize beyond the initial stimulus set (which is necessary in practical AI applications), we have recently proposed a hybrid approach [8]: Psychological dissimilarity ratings are used to initialize the similarity space, and a mapping from image stimuli to coordinates in this similarity space is then learned with convolutional neural networks. Both our own prior study [8] and related studies by Sanders and Nosofsky [58, 59] used a classification-based transfer learning approach on relatively unstructured similarity spaces involving multiple cognitive domains. In contrast to that, the present study focuses on the single cognitive domain of SHAPE and investigates a larger variety of machine learning setups, comparing two network types (classification network vs. autoencoder) and two learning regimes (transfer learning vs. multi-task learning).

The remainder of this article is structured as follows: In Sect. 2, we provide some general background on convolutional neural networks, conceptual spaces, and the cognitive domain of shapes. We then describe our general experimental setup in Sect. 3, before presenting the results of our machine learning experiments in Sect. 4. Finally, Sect. 5 summarizes the main contributions of this article and provides an outlook towards future work. All of our results as well as source code for reproducing them are publicly available on GitHub [7].<sup>1</sup>

## 2 Background

Our work combines the cognitive framework of conceptual spaces [24] (Sect. 2.1) applied to the cognitive domain of SHAPE (Sect. 2.2) with modern machine learning techniques in the form of convolutional neural networks (Sect. 2.3), following a hybrid approach (Sect. 2.4). In the following, we introduce the necessary background in these topics.

### 2.1 Conceptual Spaces

A conceptual space as proposed by Gärdenfors [24] is a similarity space spanned by a small number of interpretable, cognitively relevant *quality dimensions* (e.g., TEMPERATURE, TIME, HUE, PITCH). One can measure the difference between two observations with respect to each of these dimensions and aggregate them into

---

<sup>1</sup> See <https://github.com/lbechberger/LearningPsychologicalSpaces/>.

a global notion of semantic distance. Semantic similarity is then defined as an exponentially decaying function of distance.

The overall conceptual space can be structured into so-called *domains*, which represent, for example, different perceptual modalities such as COLOR, SHAPE, TASTE, and SOUND. The COLOR domain, for instance, can be represented by the three dimensions HUE, SATURATION, and LIGHTNESS, while the SOUND domain is spanned by the dimensions PITCH and LOUDNESS. Based on psychological evidence [2,62], distance within a domain is measured with the Euclidean metric, while the Manhattan metric is used to aggregate distances across domains.

Gärdenfors defines *properties* like RED, ROUND, and SWEET as convex regions within a single domain (namely, COLOR, SHAPE, and TASTE, respectively). Concept hierarchies are an emergent property of this spatial representation, such as the SKY BLUE region being a subset of the BLUE region. Based on properties, Gärdenfors now defines full-fledged *concepts* like APPLE or DOG by using one convex region per domain, a set of salience weights (which represent the relevance of the given domain to the given concept), and information about cross-domain correlations. The APPLE concept may thus be represented by regions for RED, ROUND, and SWEET in the domains of COLOR, SHAPE, and TASTE, respectively.

This geometric representation of knowledge enables a straightforward implementation of *commonsense reasoning* strategies such as interpolative and extrapolative reasoning [17,61]. It also allows us to model *concept combinations* such as GREEN BANANA by restricting the region of the BANANA concept in the COLOR domain to the region representing GREEN and then updating the regions in other domains (such as TASTE) based on the aforementioned cross-domain correlations (e.g., by restricting it to the SOUR region). Moreover, conceptual spaces can be linked to the *prototype theory* of concepts from psychology [56], which states that each concept is represented by a prototypical example and that concept membership is determined by comparing a given observation to this prototype. In conceptual spaces, a prototype corresponds to the center of a conceptual region, which adds further cognitive grounding to the framework.

Conceptual spaces form an intermediate layer of representation that can act as a bridge between the symbolic layer and the connectionist layer [43]: *Connectionist approaches* make use of artificial neural networks and usually consider raw perceptual inputs (e.g., pixel values of an image), which can be interpreted as a very high-dimensional feature space (e.g., one dimension per pixel). These systems are often difficult to interpret and cannot model important principles such as compositionality. *Symbolic approaches* on the other hand are based on formal logics, but suffer from the *symbol grounding problem* [27], which means that the symbols they operate on are not tied to perception and action. Conceptual spaces can be used as an intermediate representation format which translates between these two approaches: Using a connectionist approach, raw perceptual input can be mapped onto the relatively low-dimensional and interpretable conceptual space. Points in this conceptual space can then be mapped to constants and variables from the symbolic layer, while conceptual regions correspond to

symbolic predicates. This way, the advantages of both classical approaches can be combined in a cognitively grounded way.

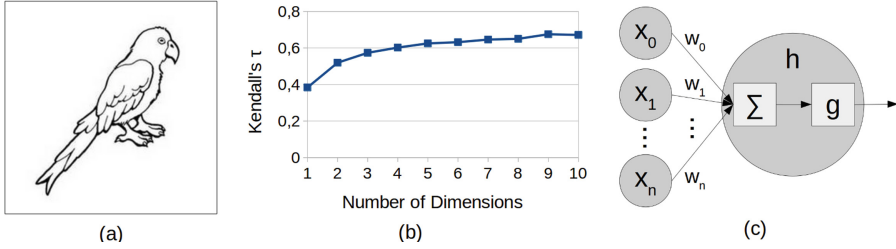
## 2.2 The Cognitive Domain of Shapes

Over the past decades, there has been ample research on shape perception in different fields such as (neuro-)psychology [4, 12, 13, 22, 30, 31, 41, 45, 50, 54, 66], computer vision [15, 47, 49, 71], and deep learning [3, 25, 38, 63]. Although so far no complete understanding of the shape domain has emerged, there exist some common themes that appear in multiple approaches, such as the distinction between global structure and local surface properties [3, 4, 12, 30], or candidate features such as ASPECT RATIO [4, 12, 15, 45, 47, 50, 66, 71], CURVATURE [12, 13, 15, 47, 50, 66, 71], and ORIENTATION [4, 15, 31, 45, 54, 66, 71].

In the context of conceptual spaces, Gärdenfors [24] mainly refers to the model proposed by Marr and Nishihara [45], which uses configurations of cylinders to describe shapes on varying levels of granularity. This cylinder-based representation can be transformed into a coordinate system by representing each cylinder with its length, diameter, and relative location and rotation. If the number of cylinders is fixed, one can thus derive a conceptual space for the SHAPE domain with a fixed number of dimensions. A related proposal for representing the SHAPE domain within conceptual spaces has been made by Chella et al. [15], who use the more powerful class of superquadrics as elementary shape primitives, allowing them to express many simple geometric objects such as boxes, cylinders, and spheres as convex regions in their similarity space.

Both existing models of the shape domain within the conceptual spaces framework define complex shapes as a configuration of simple shape primitives and follow therefore a structural approach [22]. The number of primitives necessary to represent a complex object may, however, differ between categories. Since two stimuli can therefore not necessarily be represented as two points in the same similarity space, it becomes difficult to compute distances between stimuli. Also the psychological plausibility of these approaches has so far not been established.

In order to provide a conceptual space representing the holistic similarity of complex shapes, Bechberger and Scheibel [9–11] therefore followed a different approach: As stimuli, they used sixty line drawings of everyday objects from twelve different semantic categories (such as APPLIANCE, BIRD, BUILDING, and INSECT), taken from different sources and adjusted such that they match in relative object size as well as object position and object orientation (see Fig. 1a). Six categories contained visually similar items (e.g., APPLIANCE and BIRD), while the other six categories were based on visually variable items (e.g., BUILDING and INSECT). Bechberger and Scheibel conducted a psychological study with 62 participants, where an explicit rating of the visual dissimilarity for all pairs of items was collected, using a five-point scale ranging from “totally dissimilar” to “totally similar”. In a small control experiment, Bechberger and Scheibel verified that the elicited ratings targeted shape similarity rather than overall conceptual similarity. Using the averaged dissimilarity ratings over all participants, they



**Fig. 1.** (a) Example stimulus from the study by Bechberger and Scheibel [11]. (Image license CC BY-NC 4.0, source: <http://clipartmag.com/cockatiel-drawing>) (b) Correlation of distances in the similarity space to the original dissimilarity ratings. (c) Artificial neuron as nonlinear transformation of a weighted sum.

then applied an optimization technique called *multidimensional scaling* (MDS) to obtain similarity spaces of different dimensionality. MDS represents each stimulus as a point in an  $n$ -dimensional space and ensures that geometric distances between pairs of stimuli reflect their psychological dissimilarity [14].

Their investigations showed that the resulting shape spaces fulfilled the predictions of the conceptual spaces framework: Distances had a high correlation to the original dissimilarities (see Fig. 1b), and visually coherent categories (such as APPLIANCE and BIRD) were represented as small and non-overlapping convex regions. Human ratings of the objects with respect to three psychologically motivated shape features – namely, ASPECT RATIO, LINE CURVATURE, and ORIENTATION – could be interpreted as linear directions in these spaces. Overall, their analysis indicated that similarity spaces with three to five dimensions strike a good balance between compactness and expressiveness. For instance, Fig. 1b shows that higher-dimensional spaces only marginally improve the correlation of distances to dissimilarities. We will use their four-dimensional similarity space as a target for our machine learning experiments.

Recently, Morgenstern et al. [49] have proposed a 22-dimensional similarity space for shapes obtained via MDS from 109 computer vision features on a dataset of 25,000 animal silhouettes. Predictions of their similarity space on novel stimuli were highly correlated with human similarity ratings ( $r = 0.91$ ), giving an indirect psychological validation to their approach. Moreover, Morgenstern et al. trained different shallow CNNs to map from original input images into their shape space. This relates their work quite strongly to our current study. In contrast to their work, we start from psychological data on complex line drawings and consider more complex network architectures.

### 2.3 Convolutional Neural Networks

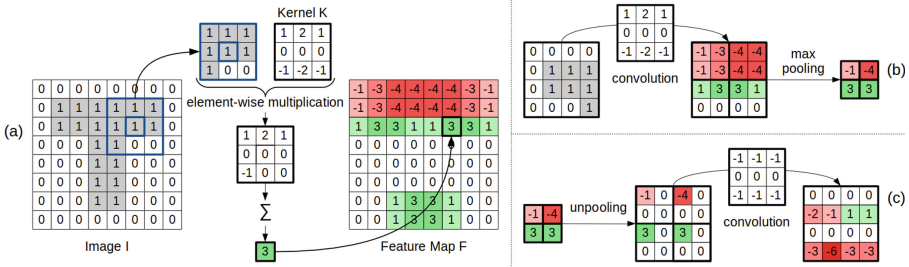
Artificial neural networks (ANNs) consist of a large number of interconnected units [48, Chap. 4]. Each unit computes a weighted sum of its inputs, which is then transformed with a nonlinear *activation function*  $g(\sum_i w_i \cdot x_i)$  (see Fig. 1c). Popular choices for the activation function include the so-called *Rectified Linear Unit* (ReLU, used for intermediate layers)  $g(z) = \max(0, z)$  as well as the *sigmoid unit*  $g(z) = \frac{1}{1+e^{-z}}$  (for binary classification output), the *softmax unit*  $g(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$  (for multi-class classification output), and the *linear unit*  $g(z) = z$  (for regression output).

The trainable parameters of an ANN correspond to the weights  $w_i$  of its connections. They are estimated by minimizing a given *loss function* which measures the network's prediction error. Popular loss functions include the *mean squared error* (which computes the average squared difference between regression output and ground truth) and the *cross-entropy loss* (which measures the difference between the probability distribution of the classification output and the ground truth). This loss function is minimized through *gradient descent*: One computes the derivative of the loss function with respect to each weight  $w_i$  and then makes small adjustments to the weights based on their derivatives. Instead of using the aggregated prediction error over all data points, one usually estimates it from a so-called *mini-batch*, i.e., a subset of examples [26, Chap. 8]. Training a neural network then consists of iterating over the dataset, where the network's weights are updated based on a new mini-batch in each iteration. Usually, multiple *epochs* (i.e., loops over the whole dataset) are needed until the optimization converges.

Instead of using the complete dataset for training the network, one usually considers a split into three subsets: The *training set* is used to optimize the parameters  $w_i$  of the network, while the *validation set* is used to monitor its performance on previously unseen examples. This can for instance be used for *early stopping*, where the training procedure is terminated, once the performance on the validation set stops improving. The *test set* is then used in the end to judge the expected generalization performance of the network on novel inputs.

A final important aspect of training neural networks are *regularization techniques* [26, Chap. 7], which are used to counter-act *overfitting* tendencies (where the network memorizes all examples from the training set, but is unable to generalize to novel inputs from the validation or test set): This includes adding a so-called *weight decay* term to the loss function, which penalizes large weight values and is motivated by the observation that smaller weights often lead to smoother decision behavior. *Dropout* is another popular regularization technique, where on each training step a randomly chosen subset of neurons is deactivated in order to increase the network's robustness.

With respect to computer vision tasks such as image classification, convolutional neural networks (CNNs) are considered to be the most successful ANN variant [26, Chap. 9]. They make use of so-called *convolutional layers* which apply the same set of weights (represented as kernel  $K$ ) at all locations (see Fig. 2a). This and the relatively small *size of the kernel* (and thus the receptive



**Fig. 2.** (a) Two-dimensional convolution with a  $3 \times 3$  kernel. (b) Combination of convolution and max pooling. (c) Combination of unpooling and convolution.

field of each unit) drastically reduces the number of connections between subsequent layers. CNNs furthermore use so-called *max pooling* layers (see Fig. 2b) to reduce the size of the image by replacing the output at a certain location by the maximum of its local neighborhood. For a max pooling layer, one has to specify both the *pool width* (i.e., the size of the area to aggregate over) and the so-called *stride* (i.e., the step size between two neighboring centers of pooling).

Typical convolutional networks start from a very high-dimensional input (namely, images) and reduce the representation size in multiple steps until a fairly small representation is reached which can then be used for classification through a softmax layer. However, in some settings one is also interested in the opposite direction: Creating a high-dimensional image from a low-dimensional hidden representation. For instance, *autoencoders* [26, Chap. 14] are an important unsupervised neural network architecture and are commonly used for dimensionality reduction and feature extraction. Autoencoders are typically trained on the task of reconstructing their input at the output layer, using only a relatively low-dimensional internal representation. They consist of an *encoder* (which compresses information) and a *decoder* (which reconstructs the original input).

For the encoder, a regular CNN can be used, whose max pooling layers, however, create a loss of information [26, Sect. 20.10.6]: In Fig. 2b, we only keep the maximum value for each  $2 \times 2$  patch of the feature map. Since three out of the four values are discarded completely, it is impossible to accurately reconstruct them. In the decoder, one therefore needs to approximate the inverted pooling function with so-called *unpooling* steps. In most cases, one simply replaces each entry of the feature map by a block of size  $s \times s$ , where the original value is copied to the top left corner and all other entries of the block are set to zero [20] (cf. Fig. 2c). Using such an unpooling step followed by a convolution (which is together often called an *upconvolutional* layer) can be seen as an approximate inverse of computing a convolution and a subsequent pooling [20]. This allows us to increase the representation size inside the decoder in order to reconstruct the original input image from a small bottleneck representation.

## 2.4 A Hybrid Approach

A popular way of obtaining a conceptual similarity space is based on dissimilarity ratings [24], which are collected for a fixed set of stimuli in a psychological experiment. They are then converted into a geometric representation of the stimulus set by using MDS (cf. Sect. 2.2). The similarity spaces produced by MDS do not readily generalize to unseen stimuli: Mapping a novel input into the similarity space requires one to collect additional dissimilarity ratings and then to re-run the MDS algorithm on the enlarged dissimilarity matrix [6]. Artificial neural networks (ANNs) on the other hand are capable of generalizing beyond their training examples, but are not necessarily psychologically grounded.

In our proposed *hybrid approach* [8], we therefore use MDS on human dissimilarity ratings to “initialize” the similarity space and ANNs to learn a mapping from stimuli into this similarity space, where the stimulus-point mappings are treated as labeled training instances for a regression task. In general, ANNs require large amounts of data to optimize their weights, but the number of stimuli in a psychological study is necessarily small. We propose to resolve this dilemma not only through data augmentation (i.e., by creating additional inputs through minor distortions), but also by introducing an additional training objective (e.g., correctly classifying the given images into their respective classes). This additional training objective can also be optimized on additional stimuli that have not been used in the psychological experiment. Using a secondary task with additional training data constrains the network’s weights and can be seen as a form of regularization. This approach has, for instance, successfully been used by Sanders and Nosofsky [58, 59], who have fine tuned pretrained CNNs to predict the MDS coordinates on a dataset of 360 rocks. In contrast to their work, we focus on the single cognitive domain of shapes, use a considerably smaller set of annotated inputs, and consider a larger variety of machine learning setups.

## 3 General Methods

In this section, we describe both our data augmentation strategy for increasing the size and variability of our dataset (Sect. 3.1) and our general training and evaluation scheme for the machine learning experiments (Sect. 3.2).

### 3.1 Data Augmentation

The dataset of line drawings used for the psychological study by Bechberger and Scheibel [11] is limited to 60 individual stimuli. These stimuli are all annotated with their respective coordinates in the target similarity space and are thus our main source of information for learning the mapping task. Moreover, we used 70 additional line drawings which were not part of the psychological study by Bechberger and Scheibel, but which use a similar drawing style. Most applications of convolutional neural networks focus on datasets of photographs such as ImageNet [16]. In contrast to photographs, the line drawings considered in our



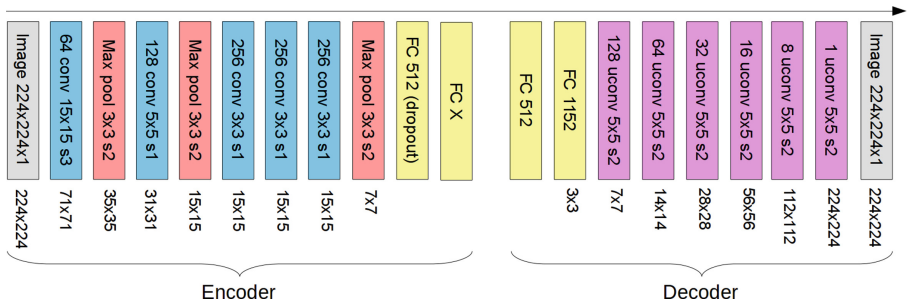
experiments do not contain any texture or background, since they only show a single object using black lines on white ground. Sketches have similar characteristics, so we used the sketch datasets TU Berlin [21] and Sketchy [60] as additional data sources. From the TU Berlin corpus, we used all 20,000 sketches, while for the Sketchy corpus we selected a subset of 62,500 images by first keeping only the sketches which had been labeled as correct by the authors and then randomly selecting 500 sketches from each of the 125 categories. TU Berlin contains 250 classes and Sketchy uses 125 classes, and both datasets overlap on a subset of 98 common classes. We used the full set of 277 distinct classes when training the network on its classification objective.

We used the following augmentation procedure to further increase the size of our dataset and the variety of inputs: For each original image, we first applied a horizontal flip with probability 0.5 and then rotated and sheared the image by an angle of up to  $15^\circ$ , respectively. In the resulting distorted image, we identified the bounding box around the object and cropped the overall image to the size of this bounding box. The resulting cropped image was then uniformly rescaled such that its longer side had a randomly selected size between 168 and 224 pixels. Using a randomly chosen offset, the rescaled object was then put in a  $224 \times 224$  image, where remaining pixels were filled with white. We used a uniform distribution over all possible resulting configurations for a given image, which makes smaller object sizes more likely since they have more translation possibilities than larger object sizes. Please note that we did not use the augmentation steps of horizontal flips and random shears and rotations on the line drawings from the psychological study, since the similarity space contains an interpretable direction which reflects the ORIENTATION of the object.

For each line drawing (both from the psychological study and additional ones), we created 2000 augmented versions, while the TU Berlin dataset and Sketchy were augmented with factors of 12 and 4, respectively. Overall, we obtained 120,000 data points for the line drawings from Bechberger and Scheibel, 140,000 data points for the additional line drawings, 240,000 data points for TU Berlin, and 250,000 data points for Sketchy.

### 3.2 Training and Evaluation Scheme

Sketch-a-Net [68, 69] was the first CNN specifically designed for the task of sketch recognition and is essentially a trimmed version of AlexNet [37], the first CNN that achieved state of the art results in image classification tasks. For our encoder network (see Fig. 3), we used Sketch-a-Net and treated the size of its second fully connected layer as a hyperparameter. Moreover, we did not use dropout in this layer and used linear units instead of ReLUs to allow the network to predict the MDS coordinates (which can also be negative) as part of its learned representation. Classification was realized with a softmax layer on top of the encoder (not shown). In the autoencoder setup, we additionally used a decoder network inspired by the work of Dosovitskiy and Brox [19], which uses two fully connected layers and 6 upconvolutional layers.



**Fig. 3.** Structure of our CNNs (“64 conv  $15 \times 15$  s3” = convolutional layer with 64 kernels of size  $15 \times 15$ , using a stride of 3, “max pool” = max pooling layer, “FC” = fully connected layer, “uconv” = upconvolutional layer; output image size shown next to the layers).

We furthermore applied binary *salt and pepper noise* (which sets randomly selected pixels to their minimal or maximal value) to the inputs before feeding them to our network. This additional noise further increases the variety of the network’s inputs and can be seen as an additional form of data augmentation. We chose salt and pepper noise rather than Gaussian noise, since the former is more adequate for our inputs, where most of the pixels are either black or white.

In our experiments reported below, we trained the overall network to minimize a linear combination of the classification error (softmax cross-entropy for the 277 classes), the reconstruction error (sigmoid cross-entropy loss with respect to the uncorrupted images<sup>2</sup>) and the mapping error (mean squared error for the target coordinates and the designated units of the second fully connected layer).

When evaluating the network’s overall performance, we used the following evaluation metrics: For the classification task, we report separate classification accuracies (i.e., percentages of correctly classified examples) for the TU Berlin and the Sketchy datasets. For the reconstruction task, we report the reconstruction error (i.e., the binary cross-entropy loss) and for the mapping task, we report the mean squared error (MSE), the coefficient of determination  $R^2$  (measuring the fraction of variance in the data explained by the model), and the mean Euclidean distance (MED) between the predicted point and the ground truth. We only used salt and pepper noise during training, but not during evaluation in order to avoid random fluctuations on the validation and test set.

Since the target coordinates used for learning and evaluating the mapping task are based only on 60 original stimuli, we decided to follow a five-fold *cross validation* scheme: We divided the original data points from each of the data sources into five *folds* of equal size and then applied the augmentation step for each fold individually. Therefore, all augmented images that were based on the same original data point are guaranteed to belong to the same fold, thus

<sup>2</sup> Since our autoencoder receives a corrupted image, but needs to reconstruct the uncorrupted original, it is a so-called *denoising* autoencoder [67].

preventing potential information leaks between folds. In our overall evaluation process, we rotated through these folds, always using three folds for training, one fold for testing, and the remaining fold as a validation set for early stopping (i.e., choosing the epoch with the lowest loss). We ensured that each fold was used once for testing, once as validation set, and three times as training set. The reported numbers are always averaged across all folds. By using this five-fold cross-validation technique, we implicitly trained five neural networks with the same hyperparameter settings, but slightly different data. Our averaged results therefore approximate the expected value of the neural network’s performance on unseen inputs and hence the generalizability of the learned mapping.

During training, we used the Adam optimizer [36] as a variant of stochastic gradient descent, with the initial learning rate set to 0.0001, the default parameter settings of  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 10^{-8}$ , and a mini-batch size of 128. We ensured that each mini-batch contained examples from all relevant data sources according to their relative proportions: When training only on the classification task, we took 63 examples from TU Berlin and 65 from Sketchy. When training on both the classification and the mapping task, we used 25 line drawings, 51 sketches from TU Berlin, and 52 examples from Sketchy. Whenever the reconstruction task is involved, we used 21 line drawings, 24 additional line drawings, 41 examples from TU Berlin, and 42 data points from Sketchy. We always trained the network for 200 full epochs and select the epoch with the lowest validation set loss (classification loss or reconstruction loss for the pretraining experiments, and mapping loss for the multi-task learning experiments) in order to compute performance on the test set.

## 4 Experiments

In this section, we report the results of the experiments carried out with our general setup as described in Sect. 3. With our experiments, we try to show that learning a mapping from line drawings into the SHAPE space of Bechberger and Scheibel [9–11] is feasible. Moreover, we aim to investigate the influence of different learning regimes on the network’s performance.

In Sect. 4.1, we train our network exclusively on the classification and reconstruction task, respectively, in order to identify promising settings for its various hyperparameters. This provides a starting point for our *transfer learning* experiments in Sect. 4.2, where we apply a linear regression on top of the pretrained CNNs. This is the perhaps most straightforward approach to solving the mapping problem. In Sect. 4.3, we then follow a more complex *multi-task learning* approach, where both the mapping task and the secondary objective (either classification or reconstruction) are optimized jointly. This is a computationally more costly approach, which may however also provide superior performance. Finally, in Sect. 4.4, we investigate how well the different approaches generalize to target similarity spaces of varying dimensionality.

**Table 1.** Selected hyperparameter configurations for the classification-based and the regression-based network, respectively.

Configuration	Encoder				Decoder	
	Weight decay	Dropout	Noise level	Rep. size	Weight decay	Dropout
$C_{\text{DEFAULT}}$	0.0005	True	10%	512	–	–
$C_{\text{SMALL}}$	0.0005	True	10%	256	–	–
$C_{\text{CORRELATION}}$	0.0010	False	10%	512	–	–
$R_{\text{DEFAULT}}$	0.0005	True	10%	512	0.0000	False
$R_{\text{BEST}}$	0.0000	False	10%	512	0.0000	False

#### 4.1 Pretraining

We first considered a default setup of the hyperparameters based directly on Sketch-a-Net [68, 69] and AlexNet [37]: We used a weight decay of 0.0005, dropout in the first fully connected layer, and a representation size of 512 neurons in the second fully connected layer. Moreover, we used 10% salt and pepper noise during training. For the decoder network, we used neither dropout nor weight decay. As evaluation metrics for the classification task, we considered the accuracies reached on TU Berlin and Sketchy, while for the autoencoder, the reconstruction error was used. In both cases, we also computed the monotone correlation of distances in the feature space to the dissimilarity ratings of Bechberger and Scheibel [11], measured with Kendall’s  $\tau$  [35]. Since a full grid search on many candidate values per hyperparameter was computationally prohibitive (especially in the context of a cross validation), we first identified up to two promising settings for each hyperparameter for both network types, before conducting a small grid search by considering all possible combinations of the remaining values. The most promising configurations selected in this grid search are shown in Table 1.

For the classifier network, the best classification performance (with accuracies of 63.2% and 79.3% on TU Berlin and Sketchy, respectively) was obtained by our default setup  $C_{\text{DEFAULT}}$ . This is considerably lower than the 77.9% on TU Berlin reported for the original Sketch-a-Net [68], which, however, used a much more sophisticated data augmentation and pretraining scheme. A considerably higher correlation of  $\tau \approx 0.33$  (instead of  $\tau \approx 0.27$  for  $C_{\text{DEFAULT}}$ ) to the dissimilarity ratings could be obtained by disabling dropout and increasing the weight decay ( $C_{\text{CORRELATION}}$ ), however, at the cost of considerably reduced classification accuracies of 36.4% and 61.5% on TU Berlin and Sketchy, respectively. Since reducing the representation size barely affected classification performance, we also consider  $C_{\text{SMALL}}$ , which uses 256 units and otherwise default parameters.

For the autoencoder, we observed that completely disabling both weight decay and dropout in both the encoder and the decoder led to considerably improved reconstruction performance (reconstruction error of 0.08 for  $R_{\text{BEST}}$  in comparison to 0.13 for  $R_{\text{DEFAULT}}$ ). Also the correlation to the dissimilarities

**Table 2.** Results of our experiments on the four-dimensional target space. The respective best values for each configuration are shown in boldface.

Configuration	Task	Regressor	$\beta/\lambda$	$\tau$	MSE	MED	$R^2$
Any	Any	Zero Baseline	–	–	1.0000	0.9940	0.0000
$C_{\text{DEFAULT}}$	Transfer	Linear	–	0.2743	0.5567	0.6879	0.4409
		Lasso	0.05	0.2743	0.4775	0.6419	0.5216
	Multi-task	CNN	0.0625	<b>0.4141</b>	<b>0.4041</b>	<b>0.5920</b>	<b>0.5775</b>
$C_{\text{SMALL}}$	Transfer	Linear	–	0.2777	0.5373	0.6737	0.4575
		Lasso	0.02	0.2777	0.4737	0.6396	0.5246
	Multi-task	CNN	0.125	<b>0.4118</b>	<b>0.4182</b>	<b>0.6020</b>	<b>0.5567</b>
$C_{\text{CORRELATION}}$	Transfer	Linear	–	0.3292	0.7307	0.7825	0.2624
		Lasso	0.05	0.3292	0.5478	0.6815	0.4505
	Multi-task	CNN	2.0	<b>0.4534</b>	<b>0.4513</b>	<b>0.6115</b>	<b>0.5201</b>
$R_{\text{DEFAULT}}$	Transfer	Linear	–	0.2228	0.9709	0.9054	0.0168
		Lasso	0.02, 0.05	0.2228	0.8315	0.8739	0.1631
	Multi-task	CNN	2.0	<b>0.3533</b>	<b>0.6211</b>	<b>0.7297</b>	<b>0.3369</b>
$R_{\text{BEST}}$	Transfer	Linear	–	0.3019	1.0791	0.9362	-0.0886
		Lasso	0.02	0.3019	0.7376	0.8102	0.2605
	Multi-task	CNN	0.25, 0.5, 2.0	<b>0.4033</b>	<b>0.5494</b>	<b>0.6846</b>	<b>0.4213</b>
			0.0625	0.3893	0.5504	0.6851	0.4144

increased from  $\tau \approx 0.22$  to  $\tau \approx 0.30$ . Manipulation of all other hyperparameters did not lead to further improvements.

## 4.2 Transfer Learning

For our transfer learning task, we extracted the hidden representation of each network configuration for each of the augmented line drawings. We trained a linear regression from these feature spaces to the four-dimensional shape space by Bechberger and Scheibel [11]. In addition to the linear regression, we also consider a lasso regression (which introduces a weight decay term) with the following settings for the regularization strength  $\beta$ :

$$\beta \in \{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0\}$$

Table 2 contains the results of these regression experiments. As we can see, the linear regression performs considerably better than the zero baseline (which always predicts the origin of the target space) for the classification-based feature spaces, but not for the reconstruction-based feature spaces. Moreover, regularization helps to improve performance on all feature spaces. A lasso regression on  $C_{\text{SMALL}}$  slightly outperforms  $C_{\text{DEFAULT}}$ , hinting at an advantage of smaller representation sizes.  $C_{\text{CORRELATION}}$  does not yield competitive results, indicating that classification accuracy is a more useful selection criterion in pretraining than the correlation to human dissimilarity ratings.

Overall, transfer learning based on classification networks seems to be much more successful than transfer learning based on autoencoders, even when considering a lasso regressor. The reason for the relatively poor performance of  $R_{\text{BEST}}$

**Table 3.** Cluster analysis of the augmented images in the individual feature spaces (averaged across all folds) using the Silhouette coefficient and the Cosine distance (i.e., the Cosine of the angle between the feature vectors).

Configuration	$C_{\text{DEFAULT}}$	$C_{\text{SMALL}}$	$C_{\text{CORRELATION}}$	$R_{\text{DEFAULT}}$	$R_{\text{BEST}}$
0% Noise	0.6448	0.6347	0.5310	-0.0359	0.0818
10% Noise	0.6364	0.6263	0.5180	-0.0300	0.0768

and  $R_{\text{DEFAULT}}$  can be seen in Table 3, where we analyze how well the different augmented versions of the shape stimuli from Bechberger and Scheibel [11] are separated in the different feature spaces. We used the Silhouette coefficient [57], where larger values indicate a clearer separation of clusters. As we can see, the different augmented versions of the same original line drawing do not form any notable clusters in the reconstruction-based feature space. On the other hand, a relatively strong clustering can be observed for classification-based feature spaces under both noise conditions, indicating that the network is able to successfully filter out noise. We assume that this difference is based on the fact that the autoencoder needs to preserve very detailed information about its input (both local and global shape information) in order to create a faithful reconstruction, while a classification network only needs to preserve pieces of information that are highly indicative of class membership (rather global than local information).

### 4.3 Multi-task Learning

In our multi-task learning experiments, we trained our networks in the different configurations again from scratch, using, however, also the mapping loss as additional training objective. Instead of a two-phase process as used in the transfer learning setup, we therefore optimize both objectives at once. This allows the network to adapt the weights of its lower layers such that its internal representation becomes more useful for the mapping task, but comes at considerably higher computational cost. When training the networks, we varied the relative weight  $\lambda$  of the mapping loss in order to explore different trade-offs between the two tasks. We explored the following settings (where  $\lambda = 0.25$  approximately reflects the relative proportion of mapping examples in the classification task):

$$\lambda \in \{0.0625, 0.125, 0.25, 0.5, 1.0, 2.0\}$$

Table 2 also contains the results of our multi-task learning experiments. As we can observe, mapping performance is considerably better in the multi-task setting than in the transfer learning setting for all of the configurations under investigation. The best results are obtained for  $C_{\text{DEFAULT}}$ , which is followed closely by  $C_{\text{SMALL}}$ .  $C_{\text{CORRELATION}}$  performs again considerably worse than the other classification-based setups, although its best multi-task results are still superior to all transfer learning results. Moreover, both reconstruction-based setups are not able to close the performance gap to the classification-based networks

also under multi-task learning. These observations indicate that the multi-task learning regime is more promising than the transfer learning approach and that classification is a more helpful secondary task than reconstruction.

When taking a closer look at the optimal values for  $\lambda$ , we note that for both the  $C_{\text{DEFAULT}}$  and the  $C_{\text{SMALL}}$  setting, relatively small values of  $\lambda \in \{0.0625, 0.125\}$  have been selected. For the  $C_{\text{CORRELATION}}$  configuration, however, a relatively large mapping weight of  $\lambda = 2.0$  leads to the best mapping results, indicating that this configuration requires stronger regularization than others. Also for  $R_{\text{DEFAULT}}$ , a relatively large mapping weight of  $\lambda = 2.0$  yielded the best performance, while no unique best setting for  $\lambda$  could be determined for the  $R_{\text{BEST}}$  configuration, where different metrics are optimized by different hyperparameter settings – here,  $\lambda = 0.0625$  provides a reasonable trade-off.

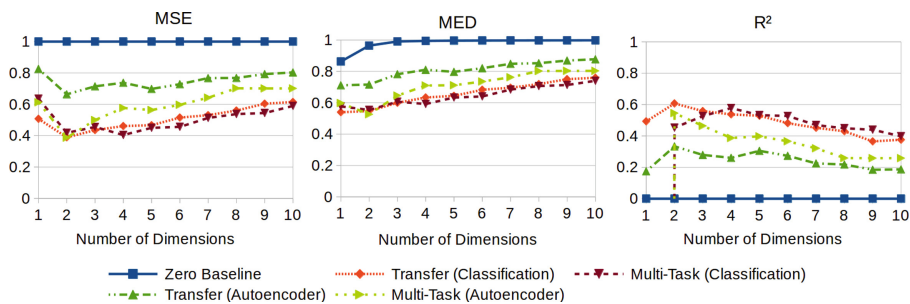
In all cases, the introduction of the mapping loss leads to a considerable increase in the correlation  $\tau$  to the dissimilarity ratings. This effect is, however, to be expected, since the mapping loss tries to align a part of the internal representation with the coordinates of the similarity space, which is explicitly based on the psychological dissimilarity ratings.

#### 4.4 Generalization to Other Target Spaces

So far, we have only considered a four-dimensional target space. In this section, we investigate how well the different approaches generalize to target spaces of different dimensionality. We considered the respective best setups for all combinations of classification-based vs. reconstruction-based networks and transfer learning vs. multi-task learning (cf. Table 2) and retrained them (using the same values of  $\beta/\lambda$ ) on all other target spaces (one to ten dimensions) of Bechberger and Scheibel [11], using again a five-fold cross validation.

Figure 4 illustrates the results of these generalization experiments for our three evaluation metrics. Both transfer learning approaches reach their peak performance for a two-dimensional target space, even though they have been optimized on the four-dimensional similarity space. Only with respect to the MED, performance is best on the one-dimensional target space. However, also the MED of the zero baseline is smallest for a one-dimensional space. If we consider the relative MED (by dividing through the MED of the zero baseline), then the best performance is again obtained on a two-dimensional target space. In all cases, classification-based transfer learning is clearly superior to reconstruction-based transfer learning.

The multi-task learners on the other hand do not show such a uniform pattern: While the reconstruction-based approach also obtains its optimum for a two-dimensional target space, the classification-based multi-task learner seems to prefer a four-dimensional target space. Moreover, both multi-task learners are more sensitive to the dimensionality of the target space than the transfer learning approaches: The classification-based multi-task learner considerably outperforms all other approaches on medium- to high-dimensional target spaces, while falling behind for a smaller number of dimensions. The reconstruction-based multi-task learner on the other hand performs quite poorly on high-dimensional spaces while



**Fig. 4.** Results of our generalization experiments to target spaces of different dimensionality for MSE, MED, and  $R^2$ .

becoming competitive on low-dimensional target spaces. Both multi-task learners use a mapping weight of  $\lambda = 0.0625$ , i.e., the smallest value we investigated. However, the size of the classification and reconstruction loss differed considerably, with a classification loss of around 1.3 to 1.6, compared to a reconstruction loss of 0.10 to 0.12 (both measured on the test set). The relative influence of the mapping objective on the overall optimization is thus considerably greater in the classification-based multi-task learner. One may therefore speculate that even smaller values of  $\lambda$  would have benefited the classification-based multi-task learner for smaller target spaces.

Overall, the results of this generalization experiment confirm the effects reported in our earlier study [8], where we also observed a performance sweet spot for a two-dimensional target space in a transfer learning setting. Again, we can argue that this strikes a balance between a clear semantic structure in the target space and a small number of output variables to predict. The observed sensitivity of the multi-task learning approach indicates that the target space should be carefully chosen before optimizing the multi-task learner.

## 5 Discussion and Conclusion

In this paper, we have aimed to learn a mapping from line drawings to their corresponding coordinates in a psychological SHAPE space. We have compared classification-based networks to autoencoders, investigating both transfer learning and multi-task learning. Overall, classification seemed to be a better secondary task than reconstruction, and multi-task learning consistently outperformed transfer learning. We found that the best performance in general was reached for classification-based multi-task learning, but that this approach was quite sensitive to the dimensionality of the target space. These results are mostly not surprising, given that multi-task learning allows for a finer-grained trade-off between tasks and that a reconstruction objective implicitly enforces also position and size information to be encoded.

We can compare our results to our earlier study [8] on a dataset of novel objects [29], where we used a lasso regression on top of a pretrained photograph-



based CNN. There, we achieved for a four-dimensional target space a MSE of about 0.59, a MED of about 0.73, and a coefficient of determination of  $R^2 \approx 0.39$ . These numbers are considerably worse than the ones obtained for classification-based transfer learning (see Sect. 4.2), indicating that the SHAPE space considered in the current study poses an easier regression problem. Moreover, we can compare our performance with respect to the coefficient of determination to the results reported by Sanders and Nosofsky [58], who reported a value of  $R^2 \approx 0.77$  for an eight-dimensional target space and a more complex network architecture, using a dataset of 360 stimuli. Our best results with  $R^2 \approx 0.61$  on a two-dimensional target space are considerably worse than this and clearly not good enough for practical applications. We assume that performance in our scenario is heavily constrained by the network size and the number of stimuli for which dissimilarity ratings were collected. This urges for further experimentation with more complex architectures, larger datasets, different augmentation techniques, and additional regularization approaches.

Overall, our present study has illustrated that it is in principle possible to predict the coordinates of a given input image in a psychological similarity space for the SHAPE domain. Although performance is not yet satisfactory, this is an important step towards making conceptual spaces usable for cognitive AI systems. Once a robust mapping of reasonably high quality has been obtained, one can use the full expressive power of the conceptual spaces framework: For instance, the interpretable directions reported by Bechberger and Scheibel [11] can give rise to an intuitive description of novel stimuli based on psychological features. Also categorization based on conceptual regions, commonsense reasoning strategies, and concept combination can then be implemented on top of the predicted coordinates in shape space (cf. Sect. 2.1).

The approach presented in this article can of course also be generalized to other domains and datasets such as the THINGS data base and its associated embeddings [28] or the recently published similarity ratings and embeddings for a subset of ImageNet [55]. It can furthermore be seen as a contribution to the currently emerging field of research which tries to align neural networks with psychological models of cognition [1, 5, 6, 32, 38, 39, 49, 51–53, 58, 59, 64, 65].

## References

1. Attarian, I.M., Roads, B.D., Mozer, M.C.: Transforming neural network visual representations to predict human judgments of similarity. In: NeurIPS 2020 Workshop SVRHM (2020). <https://openreview.net/forum?id=8wNMPXWK5VX>
2. Attneave, F.: Dimensions of similarity. *Am. J. Psychol.* **63**(4), 516–556 (1950). <https://doi.org/10.2307/1418869>
3. Baker, N., Lu, H., Erlikhman, G., Kellman, P.J.: Deep convolutional networks do not classify based on global object shape. *PLOS Comput. Biol.* **14**(12), 1–43 (2018). <https://doi.org/10.1371/journal.pcbi.1006613>
4. Bar, M.: A cortical mechanism for triggering top-down facilitation in visual object recognition. *J. Cogn. Neurosci.* **15**(4), 600–609 (2003). <https://doi.org/10.1162/089892903321662976>

5. Battleday, R.M., Peterson, J.C., Griffiths, T.L.: Capturing human categorization of natural images by combining deep networks and cognitive models. *Nat. Commun.* **11**(1), 1–14 (2020)
6. Battleday, R.M., Peterson, J.C., Griffiths, T.L.: From convolutional neural networks to models of higher-level cognition (and back again). *Ann. N. Y. Acad. Sci.* (2021). <https://doi.org/10.1111/nyas.14593>
7. Bechberger, L.: lbechberger/LearningPsychologicalSpaces v1.5: machine learning study with CNNs on shapes data, September 2021. <https://doi.org/10.5281/zenodo.5524374>
8. Bechberger, L., Kühnberger, K.-U.: Generalizing psychological similarity spaces to unseen stimuli – combining multidimensional scaling with artificial neural networks. In: Bechberger, L., Kühnberger, K.-U., Liu, M. (eds.) *Concepts in Action*. LCM, vol. 9, pp. 11–36. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-69823-2\\_2](https://doi.org/10.1007/978-3-030-69823-2_2)
9. Bechberger, L., Scheibel, M.: Analyzing psychological similarity spaces for shapes. In: Alam, M., Braun, T., Yun, B. (eds.) *ICCS 2020. LNCS (LNAI)*, vol. 12277, pp. 204–207. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-57855-8\\_16](https://doi.org/10.1007/978-3-030-57855-8_16)
10. Bechberger, L., Scheibel, M.: Representing complex shapes with conceptual spaces. In: *Second International Workshop ‘Concepts in Action: Representation, Learning, and Application’ (CARLA 2020)* (2020). <https://openreview.net/forum?id=OhFQNQicgXy>
11. Bechberger, L., Scheibel, M.: Modeling the holistic perception of everyday object shapes with conceptual spaces (in preparation)
12. Op de Beeck, H.P., Torfs, K., Wagemans, J.: Perceived shape similarity among unfamiliar objects and the organization of the human object vision pathway. *J. Neurosci.* **28**(40), 10111–10123 (2008). <https://doi.org/10.1523/JNEUROSCI.2511-08.2008>
13. Biederman, I.: Recognition-by-components: a theory of human image understanding. *Psychol. Rev.* **94**(2), 115–147 (1987)
14. Borg, I., Groenen, J.F.: *Modern Multidimensional Scaling: Theory and Applications*. Springer Series in Statistics, 2nd edn. Springer, New York (2005). <https://doi.org/10.1007/0-387-28981-X>
15. Chella, A., Frixione, M., Gaglio, S.: Conceptual spaces for computer vision representations. *Artif. Intell. Rev.* **16**(2), 137–152 (2001). <https://doi.org/10.1023/a:1011658027344>
16. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255 (2009). <https://doi.org/10.1109/CVPR.2009.5206848>
17. Derrac, J., Schockaert, S.: Inducing semantic relations from conceptual spaces: a data-driven approach to plausible reasoning. *Artif. Intell.* **228**, 66–94 (2015). <https://doi.org/10.1016/j.artint.2015.07.002>
18. Diesendruck, G., Bloom, P.: How specific is the shape bias? *Child Dev.* **74**(1), 168–178 (2003). <https://doi.org/10.1111/1467-8624.00528>
19. Dosovitskiy, A., Brox, T.: Inverting visual representations with convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016)
20. Dosovitskiy, A., Tobias Springenberg, J., Brox, T.: Learning to generate chairs with convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015)

21. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? *ACM Trans. Graph.* **31**(4), 1–10 (2012). <https://doi.org/10.1145/2185520.2185540>
22. Erdogan, G., Jacobs, R.A.: Visual shape perception as Bayesian inference of 3D object-centered shape representations. *Psychol. Rev.* **124**(6), 740–761 (2017)
23. Garcez, A.D., et al.: Neural-symbolic learning and reasoning: contributions and challenges. In: *AAAI 2015 Spring Symposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches* (2015)
24. Gärdenfors, P.: *Conceptual Spaces: The Geometry of Thought*. MIT Press, Cambridge (2000)
25. 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: *International Conference on Learning Representations* (2019). <https://openreview.net/forum?id=Bygh9j09KX>
26. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press (2016). <http://www.deeplearningbook.org>
27. Harnad, S.: The symbol grounding problem. *Phys. D* **42**(1–3), 335–346 (1990). [https://doi.org/10.1016/0167-2789\(90\)90087-6](https://doi.org/10.1016/0167-2789(90)90087-6)
28. Hebart, M.N., Zheng, C.Y., Pereira, F., Baker, C.I.: Revealing the multidimensional mental representations of natural objects underlying human similarity judgements. *Nat. Hum. Behav.* (2020). <https://doi.org/10.1038/s41562-020-00951-3>
29. Horst, J.S., Hout, M.C.: The novel object and unusual name (NOUN) database: a collection of novel images for use in experimental research. *Behav. Res. Methods* **48**(4), 1393–1409 (2015). <https://doi.org/10.3758/s13428-015-0647-3>
30. Huang, L.: Space of preattentive shape features. *J. Vis.* **20**(4), 10–10 (2020). <https://doi.org/10.1167/jov.20.4.10>
31. Hubel, D.H., Wiesel, T.N.: Receptive fields of single neurones in the cat’s striate cortex. *J. Physiol.* **148**(3), 574–591 (1959). <https://doi.org/10.1113/jphysiol.1959.sp006308>
32. Jha, A., Peterson, J., Griffiths, T.: Extracting low-dimensional psychological representations from convolutional neural networks. In: *Proceedings for the 42nd Annual Meeting of the Cognitive Science Society* (2020)
33. Jones, S.S., Smith, L.B.: The place of perception in children’s concepts. *Cogn. Dev.* **8**(2), 113–139 (1993). [https://doi.org/10.1016/0885-2014\(93\)90008-S](https://doi.org/10.1016/0885-2014(93)90008-S)
34. Kaipainen, M., Zenker, F., Hautamäki, A., Gärdenfors, P. (eds.): *Conceptual Spaces: Elaborations and Applications*. SL, vol. 405. Springer, Cham (2019). <https://doi.org/10.1007/978-3-030-12800-5>
35. Kendall, M.G.: A new measure of rank correlation. *Biometrika* **30**(1–2), 81–93 (1938). <https://doi.org/10.1093/biomet/30.1-2.81>
36. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. *arXiv* (2014). <https://arxiv.org/abs/1412.6980>
37. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105. Curran Associates, Inc. (2012). <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>
38. Kubilius, J., Bracci, S., Op de Beeck, H.P.: Deep neural networks as a computational model for human shape sensitivity. *PLOS Comput. Biol.* **12**(4), 1–26 (2016). <https://doi.org/10.1371/journal.pcbi.1004896>
39. Lake, B., Zaremba, W., Fergus, R., Gureckis, T.: Deep neural networks predict category typicality ratings for images. In: Noelle, D.C., et al. (eds.) *Proceedings of the 37th Annual Conference of the Cognitive Science Society* (2015)

40. Landau, B., Smith, L., Jones, S.: Object perception and object naming in early development. *Trends Cogn. Sci.* **2**(1), 19–24 (1998). [https://doi.org/10.1016/S1364-6613\(97\)01111-X](https://doi.org/10.1016/S1364-6613(97)01111-X)
41. Li, A.Y., Liang, J.C., Lee, A.C.H., Barense, M.D.: The validated circular shape space: quantifying the visual similarity of shape. *J. Exp. Psychol. Gen.* **149**(5), 949–966 (2019)
42. Lieto, A.: *Cognitive Design for Artificial Minds*. Routledge (2021)
43. Lieto, A., Chella, A., Frixione, M.: Conceptual spaces for cognitive architectures: a lingua franca for different levels of representation. *Biolog. Inspired Cogn. Archit.* (2016). <https://doi.org/10.1016/j.bica.2016.10.005>
44. Marcus, G., Davis, E.: *Rebooting AI: Building Artificial Intelligence We Can Trust*. Pantheon (2019)
45. Marr, D., Nishihara, H.K.: Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. Roy. Soc. London Ser. B Biol. Sci.* **200**(1140), 269–294 (1978)
46. Maruyama, Y.: Symbolic and statistical theories of cognition: towards integrated artificial intelligence. In: Cleophas, L., Massink, M. (eds.) *SEFM 2020*. LNCS, vol. 12524, pp. 129–146. Springer, Cham (2021). [https://doi.org/10.1007/978-3-030-67220-1\\_11](https://doi.org/10.1007/978-3-030-67220-1_11)
47. Mingqiang, Y., Kidiyo, K., Joseph, R.: A survey of shape feature extraction techniques. *Pattern Recogn.* **15**(7), 43–90 (2008)
48. Mitchell, T.M.: *Machine Learning*. McGraw Hill, New York (1997)
49. Morgenstern, Y., et al.: An image-computable model of human visual shape similarity. *PLoS Comput. Biol.* **17**(6), 1–34 (2021). <https://doi.org/10.1371/journal.pcbi.1008981>
50. Ons, B., Baene, W.D., Wagemans, J.: Subjectively interpreted shape dimensions as privileged and orthogonal axes in mental shape space. *J. Exp. Psychol. Hum. Percept. Perform.* **37**(2), 422–441 (2011)
51. Peterson, J.C., Abbott, J.T., Griffiths, T.L.: Adapting deep network features to capture psychological representations: an abridged report. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 4934–4938 (2017). <https://doi.org/10.24963/ijcai.2017/697>
52. Peterson, J.C., Abbott, J.T., Griffiths, T.L.: Evaluating (and improving) the correspondence between deep neural networks and human representations. *Cogn. Sci.* **42**(8), 2648–2669 (2018)
53. Peterson, J.C., Battleday, R.M., Griffiths, T.L., Russakovsky, O.: Human uncertainty makes classification more robust. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2019)
54. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. *Nat. Neurosci.* **2**(11), 1019–1025 (1999)
55. Roads, B.D., Love, B.C.: Enriching ImageNet with human similarity judgments and psychological embeddings. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3547–3557 (2021)
56. Rosch, E., Mervis, C.B., Gray, W.D., Johnson, D.M., Boyes-Braem, P.: Basic objects in natural categories. *Cogn. Psychol.* **8**(3), 382–439 (1976). [https://doi.org/10.1016/0010-0285\(76\)90013-x](https://doi.org/10.1016/0010-0285(76)90013-x)
57. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987). [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7)

58. Sanders, C.A., Nosofsky, R.M.: Using deep-learning representations of complex natural stimuli as input to psychological models of classification. In: Proceedings of the 2018 Conference of the Cognitive Science Society, Madison (2018)
59. Sanders, C.A., Nosofsky, R.M.: Training deep networks to construct a psychological feature space for a natural-object category domain. *Comput. Brain Behav.* **3**, 229–251 (2020)
60. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: learning to retrieve badly drawn bunnies. *ACM Trans. Graph.* **35**(4), 1–12 (2016). <https://doi.org/10.1145/2897824.2925954>
61. Schockaert, S., Prade, H.: Interpolation and extrapolation in conceptual spaces: a case study in the music domain. In: Rudolph, S., Gutierrez, C. (eds.) RR 2011. LNCS, vol. 6902, pp. 217–231. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-23580-1\\_16](https://doi.org/10.1007/978-3-642-23580-1_16)
62. Shepard, R.N.: Attention and the metric structure of the stimulus space. *J. Math. Psychol.* **1**(1), 54–87 (1964). [https://doi.org/10.1016/0022-2496\(64\)90017-3](https://doi.org/10.1016/0022-2496(64)90017-3)
63. Singer, J., Hebart, M.N., Seeliger, K.: The representation of object drawings and sketches in deep convolutional neural networks. In: NeurIPS 2020 Workshop SVRHM (2020). <https://openreview.net/forum?id=wXv6gtWnDO2>
64. Singh, P., Peterson, J., Battleday, R., Griffiths, T.: End-to-end deep prototype and exemplar models for predicting human behavior. In: Proceedings for the 42nd Annual Meeting of the Cognitive Science Society (2020)
65. Sorscher, B., Ganguli, S., Sompolinsky, H.: The geometry of concept learning. *bioRxiv* (2021). <https://doi.org/10.1101/2021.03.21.436284>
66. Treisman, A., Gormican, S.: Feature analysis in early vision: evidence from search asymmetries. *Psychol. Rev.* **95**(1), 15–48 (1988)
67. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning - ICML 2008 (2008). <https://doi.org/10.1145/1390156.1390294>
68. Yu, Q., Yang, Y., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M.: Sketch-a-net: a deep neural network that beats humans. *Int. J. Comput. Vis.* **122**(3), 411–425 (2017)
69. Yu, Q., Yang, Y., Song, Y.Z., Xiang, T., Hospedales, T.: Sketch-a-net that beats humans. In: Xie, X., Jones, M.W., Tam, G.K.L. (eds.) Proceedings of the British Machine Vision Conference (BMVC), pp. 7.1–7.12. BMVA Press (2015). <https://doi.org/10.5244/C.29.7>
70. Zenker, F., Gärdenfors, P. (eds.): Applications of Conceptual Spaces. Springer, Heidelberg (2015). <https://doi.org/10.1007/978-3-319-15021-5>
71. Zhang, D., Lu, G.: Review of shape representation and description techniques. *Pattern Recogn.* **37**(1), 1–19 (2004). <https://doi.org/10.1016/j.patcog.2003.07.008>