Proceedings of the ASME 2022
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC/CIE2022
August 14-17, 2022, St. Louis, Missouri, USA

IDETC2022-89723

VISUAL REASONING FOR DESIGN BY ANALOGY: FUSE VISUAL AND SEMANTIC KNOWLEDGE

Zijian Zhang
Dept. of Aerospace & Mechanical Engineering
University of Southern California
Los Angeles, California 90089
zijianz@usc.edu

Yan Jin*
Dept. of Aerospace & Mechanical Engineering
University of Southern California
Los Angeles, California 90089
yjin@usc.edu
(*corresponding author)

ABSTRACT

Design by analogy is a design ideation strategy to find inspiration from source domains to generate design concepts in target domains. Recently, many computational methods were proposed to measure similarities between source domains and target domains to build connections between them. However, most existing methods only explore either visual or semantic cues of the concepts in source and target domains but ignore the integration of both modalities. In fact, humans have remarkable visual reasoning ability to transfer knowledge learned from objects in familiar categories (source domains) to recognize objects from unfamiliar categories (target domains). In this paper, we propose a visual reasoning framework to support design by visual analogy. The challenge of this research is how computation methods can mimic the process of humans' visual reasoning, which fuses visual and semantic knowledge. In the framework, the convolutional neural network (CNN) is applied to learn visual knowledge from objects in familiar categories. The hierarchy-based graph convolutional network (HGCN) is proposed to transfer learned visual knowledge from familiar categories to unfamiliar categories by their semantic distances. Finally, the unfamiliar objects can be reasoned and recognized based on the transferred visual knowledge. Extensive experimental results on one mechanical component benchmark dataset demonstrate the favorable performance of our proposed methods.

Keywords: Visual reasoning, visual similarity, deep learning, design by analogy, semantic knowledge.

1. INTRODUCTION

Designers often seek inspirational stimuli during ideation at the early stages of the design process. The visual analogy is considered as an effective cognitive strategy to stimulate designers to create innovative concepts for solving ill-structured design problems[1-3]. In our previous work[4, 5], it has been shown that visual similarity existing between the source and target domains is a precondition to make a visual analogy. A visual relationship might not be the only ideal criterion for making a visual analogy. For instance, a post-it note is visually similar to a map view of the state of New Mexico in that they're both squares, but this does not mean that they have some degrees of useful analogical similarity. For accessing more meaningful visual stimuli, semantic similarity should also be considered.

Shapes don't only refer to geometry but also carry semantics. In cognitive science, studies support the idea that people first perceive the shape and overall structure of an object and then comprehend the semantic details[6, 7]. As visual images are stored in memory both verbally and visually, verbal and visual descriptions in shape interpretation are possible for people to understand the images[8, 9]. One example is given a picture of an apple; humans can recognize the object's name, shape, color, texture, infer its taste, and think about how to eat it. In engineering design, shapes can arouse complex semantic content. That functions fit shapes or structures is one of the basic design principles well accepted in the design field, such as Structure-Behavior-Function (SBF) approach[10], SAPPhIRE model[11], a deep learning model[12], and a structure-function patterns approach[13]. Frequently, visual and language-based thinkings overlap and interconnect during design. Visual and semantic representations can help designers retrieve useful

analogies and increase the probability of successful designs[14-17]. However, in recent years, the majority of visual analogy research in the engineering design field focus on capturing and analyzing the visual information of an image[4, 5, 18, 19], few lines of work focus on utilizing semantics as a reasoning source to support visual analogy making.

Visual reasoning is possible as humans can interpret shapes' semantic meanings. Consider a "binding barrel" in Figure 1. Assume we have never heard of this category or seen visual examples in the past, and we would like to find a most functionally and geometrically similar object to the query from the support images. As the query image consists of a barrel and a binding screw that threads into the barrel, we can visually reason that it might be a type of fastener as it is very similar to a screw with threads and a crosshead. Even its shape is visually similar to a bike pump and a fire hydrant; however, we know that it semantically belongs to a mechanical component. Humans are capable of inferring unknown objects on a higher-level category (a binding barrel and a screw are different types of fasteners), considering visual and semantic information at the same time. This visual reasoning capability is helpful for design by visual analogy, as the relationships between a binding barrel (a target domain) with the four related objects (source domains) are built during this process. Also, before the visual inference, humans already have some prior knowledge of relevant objects (support images in Figure 1) and transfer the knowledge to comprehend and describe the unfamiliar object (the query image in Figure 1). Therefore, the main research problem in this paper: how to semantically weight visual knowledge transferred from the source domains and recommend semantically meaningful visual analogies.

Figure 1 A visual reasoning example

In this work, we propose a visual reasoning framework to infer the category of an unfamiliar object using visual knowledge from familiar objects in different categories and semantic knowledge of categories. Specifically, we first use a convolutional neural network (CNN) to learn visual features of familiar objects. Then, we build a hierarchy-based graph convolutional network (HGCN) in which each node corresponds to a category. Each node is represented by a word embedding. These nodes are linked via semantic relationship edges. The weights of the edges are determined by the similarities of the hierarchical semantics between these nodes. The HGCN is trained to transfer visual knowledge from familiar categories to unfamiliar categories. Finally, the category of the unfamiliar object can be inferred based on the transferred visual knowledge.

2. RELATED WORK

2.1 Computational methods in design by analogy

Searching for inspirational stimuli is an essential step in the initial stages of the design process. Many empirical studies have investigated the impact of external stimuli of inspiration on design ideation, such as their ability to promote the designers' imagination and boost the generation of novel concepts[14, 15, 20]. However, they can contribute to design fixation, which means designers could be stuck in mimicking external stimuli and unconsciously constrained in a limited set of ideas[21]. Design by analogy is a way to help designers to explore design space and alleviate design fixation. Inspiration can be drawn from analogies in source domains to generate creative ideas for a target domain.

The unlimited number of potential inspiration sources are around designers to search for. Therefore, databases, along with effective retrieval of analogies, have great potential to enhance design by analogy. Currently, many computational tools have been used to provide inspiration to designers and avoid design fixation. Based on the function, behavior, or shape of a device, analogies from nature, patents, and images are provided as potential sources of inspiration to the designer. In order to improve the efficiency of retrieving distantly related stimuli, computational methods and tools can be constructed to support a less time-consuming search for inspiration with different levels of semantic or visual distances to the problem domain[14].

Many computer-aided ideation methods or tools have been developed to retrieve analogies from a text-based database. Natural language processing is applied to conduct semantic similarity to filter the specific verbal stimulus to be retrieved. Shu et al. used natural language analysis to correlate functional basis terms with useful biological keywords[22, 23]. Murphy proposed a search methodology to identify inspiring patents which have functional semantic similarity with design problems [24]. Fu et al. created a computation method to cluster patents based on their functional and surface similarity, and then designers can automatically retrieve analogical stimuli from these patents [25]. The WordTree method can re-represent key functions of a design problem through the WordNet database, and analogous concepts can be identified[26].

As designers are skillful in making and using visual representations, they have a striking preference for visual stimuli. One of the reasons for the efficiency of visual stimulation in idea generation is that less cognitive effort is required when accessing, storing, and communicating pictorial information compared with textual information. Some methods have been proposed to retrieve visual analogies[27]. Ji et al. created a computational tool to provide image-based stimuli to improve creativity and enhance design communication[28]. Kwon et al. develop a method to use visually similar images to support concept generation for wind-turbine blades[19]. Jiang et al. introduced a supervised CNN-based approach for patent image vectorization to support visual design stimuli retrieval in design-by-analogy[18]. However, these methods assume that designers know what to search for and, thus, how to initiate the

image search by entering keywords. Recently, we put forward an unsupervised deep clustering method to retrieve visual stimuli based on vague ideas in designers' mind in sketch forms instead of keywords[4, 5].

2.2 Reasoning in visual analogy

Analogies are fundamental for human cognition and creativity in which information is transferred from source domains to target domains [29]. Reasoning by analogy is considered to be at the center of cognitive processes for solving creative problems [30]. In engineering design, prolific research has been related to provoke visual analogies by displaying a large variety of visual representations to designers [3, 15, 31, 32]. Designers can benefit more from reasoning by visual analogy than the use of visual display [33, 34]. However, little work has been carried out for developing computational methods and tools to support design by analogy based on visual reasoning.

Visual perception and cognition are two different but interactive mechanisms that operate in vision [35]. Visual perception is responsible for discerning what the stimulus input is based on the shape. Visual cognition is triggered by the perceptual event to understand and reason about the input, such as its physical properties and usage [36]. This suggests that visual perception may provide an indexical function for retrieving long-term memory in the human brain. The stored information and knowledge that applies to a particular object can be activated and manipulated by a cognition process such as reasoning, imagining, memorizing and so on.

Images can not only display shapes but also carry semantic content of the classes of perceptual inputs. Shapes can be reasoned because we can decipher their related semantic meanings. Much research in cognitive psychology has been done to study the relationship between visual and verbal descriptions in image interpretation. Visual images can evoke verbal-propositional memory traces in as little as 100 msec [37]. Human long-term memory can be characterized as a network of linked semantic concepts [38]. Humans first grasp the shape and spatial structure of an object and then understand the details [39]. The visual stimulus is linked to the higher-level knowledge, such as stored semantic meanings, abstract knowledge, and complex beliefs [40].

2.3 Graph Convolutional Network for Visual Reasoning

Recent graph neural models are showing strong performance in their ability to extract node features and learn structured relationship between nodes. To extend powerful convolutional neural network (CNN) to deal with graph-structured data, graph convolutional network (GCN) was introduced by Kipf and Welling for semi-supervised learning on graph structured data [41]. The graph convolutional operation aims to generate representations for vertices by aggregating its own feature and the features of its neighboring vertices.

Researchers have leveraged GCN for reasoning on pairwise relations to be beneficial to a variety of computer vision tasks. Some works have been proposed that leverage scene graphs for improving scene understanding [42] and generation[43] through visual reasoning. Automatic image captioning is another computer vision task supported by GCN to reason the visual relationships between objects in the image and understand the semantic information[44, 45]. Our work is most related to zero-shot learning, which builds GCN using semantic information from WordNet, ConceptNet, or Wikipedia texts to construct relationships between known and unknown objects and then understand unknown objects based on visual features learned from some known objects[46, 47].

In most of the previous works, one node in the GCN can only be able to obtain the nearest neighbors' information directly and needs a multilayer structure to acquire long-distance neighbors' information indirectly through graph propagations. However, feature representations of the dataset in GCN will become more similar as the depth increases if they come from the same cluster[48]. In our visual reasoning setting, it means visual classifiers sharing the same parent or grandparent category will be indistinguishable. Considering this limitation, we build a novel hierarchy-based graph convolutional network (HGCN) by adding semantic weights on the edges in the graph structure, which can directly obtain the short and long-distance neighbors' information using only one layer. In this way, while retaining the inherent advantages of GCN, the number of layers in the model can be reduced.

3. METHODS

The recent progress of deep learning has advanced design by analogy. Despite the success, the state-of-the-art models are notoriously data hungry, requiring tons of samples for parameter learning. In real cases, however, the visual phenomena follows a long-tail distribution where only a few categories are data-rich and the rest are with limited training samples[20].

Compared with machines, people are far better learners as they are capable of learning models from previous seen samples and accuratly infer a new category accordingly. An intuitive example is that a baby learner can learn to recognize a wolf that he/she has been able to successfully recognize a dog. The key mystery making the difference is that people have strong prior knowledge to generalize across different categories[21]. It means that people do not need to learn a new classifier (e.g. wolf) from scratch as most machine learning methods, but generalize and adapt the previously learned classifiers (e.g. dog) towards the new category.

In the design by analogy scenario, learning to make analogy refers to the mechanisms that learn how to transfer previous knowledge from source domains to target domains. The purpose of our proposed visual reasoning framework is to explore how machines can capture the learning to make analogy ability. Similarly, in the previous example of dog and wolf, we have a plausible explanation on the fast reasoning and learning of wolf that a human learner selects dog from the source domains and transfers its classification parameters for wolf classification. In this sense, visual reasoning provides effective and informative

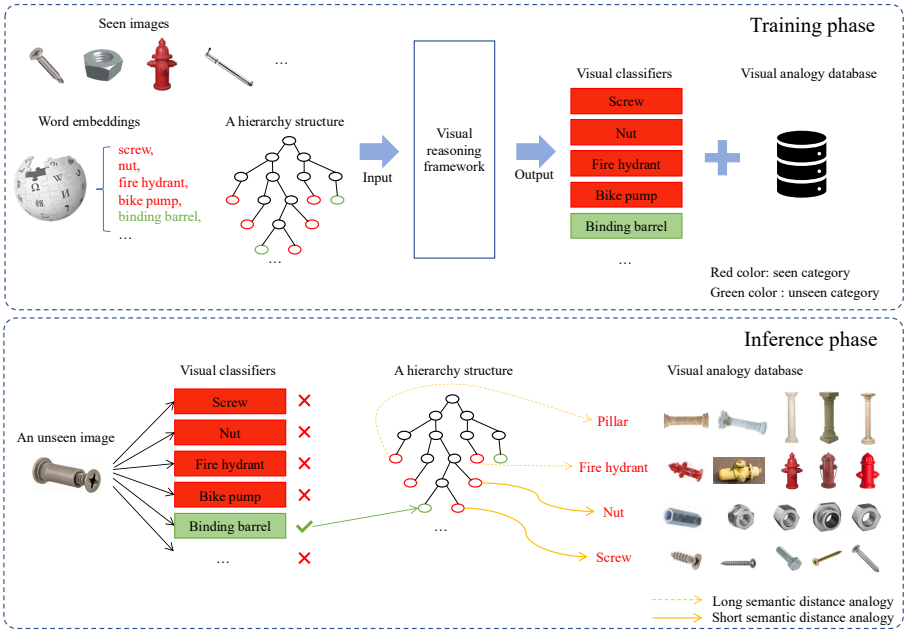


Figure 2 The phases of visual reasoning supported design by analogy

clue for generalizing image classifiers in a way of making visual analogy. In particular, when the samples in the target domain have a limited number and hard to learn the visual classifer, how to transfer the classification parameters from selected source domains is highly non-trivial.

As depicted in Figure 2, there two phases to use visual reasoning to support design by analogy. In the training phase, the seen images, their labels' word embeddings and a hierarchy structure including seen and unseen categories are the inputs to our proposed visual reasoning framework. The seen and unseen categories mean source and target domains in the design by analogy scenario. The outputs of the visual reasoning framework are learned visual classifiers and a visual analogy database. The seen visual classifiers are learned by seen images and other corresponding labels, which is introduced in section 3.2. The unseen visual classifiers are transferred from the seen visual classifiers based on the hierarchy structure, which is illustrated in sections 3.3 and 3.4. The visual analogy database includes seen images and their corresponding labels. In the inference phase, an unseen image can be a sketch/image in the target domain generated by a designer. The label of the unseen image will be predicated by our learned visual classifiers. According to our proposed semantic distance d_G, which is demonstrated in section 3.3, we can determine the long- and short- distance analogies in the hierarchy structure and visual analogies can be retrieved to stimulate the designer.

3.1 A visual reasoning framework

A rich body of research on computational methods for design by analogy only supports identifying analogies based on one modality, either linguistic or geometrical similarity. Based on a strong psychological cognition understanding of visual analogy, the process to identify similarities between source domains and target domains is fulfilled by reasoning. Our hypothesis for visual reasoning in design by analogy is that once an object in a target domain is perceived and cognized, related concepts, such as categories and attributes, will be activated and brought to the level of working memory, other objects in source domains can be energized from long-term memory based on similarities of the related concepts, and semantic information and knowledge of the objects in the source domains can be retrieved to working memory to understand and comprehend the object in the target domain. Therefore, computational tools can consider semantic and visual similarity at the same time when identifying visual analogies is needed. Advanced deep learning technologies, such as CNN and GCN, provide us with ways to figure out the semantic and visual relationships between source domains and target domains to realize the visual reasoning process. In this research, our goal is to propose a computational method to use visual and semantic knowledge to support visual reasoning for design by analogy. In order to approach to this goal, we set up a visual reasoning setting as follows.

Suppose we have source domain datasets $\mathbb{D}^s = \{\mathbb{V}^s, \mathbb{Y}^s\}$ which have N_s labeled images, where each image $V^s \in \mathbb{V}^s$ is associated with a label $Y^s \in \mathbb{Y}^s$. Similarly, there are target domains $\mathbb{D}^t = \{\mathbb{V}^t, \mathbb{Y}^t\}$consisting of $N_t$images from target categories $\mathbb{Y}^t$. Here, $\mathbb{Y}^s \cup \mathbb{Y}^t = \mathbb{Y}$, $\mathbb{Y}^s \cap \mathbb{Y}^t = \emptyset$. All the categories in source and targets domains are called seen and unseen categories, respectively. The processes of visual reasoning are as follows: images and their corresponding labels in the source domain datasets are used for training to learn visual features and classifiers of the seen categories by CNN; meanwhile, visual features of the images in the unseen categories can be recognized and extracted by the learned CNN; we assume there is a shared semantic hierarchy covering both seen and unseen categories. The visual classifiers of the unseen categories can be reasoned by building semantic relationships to the visual

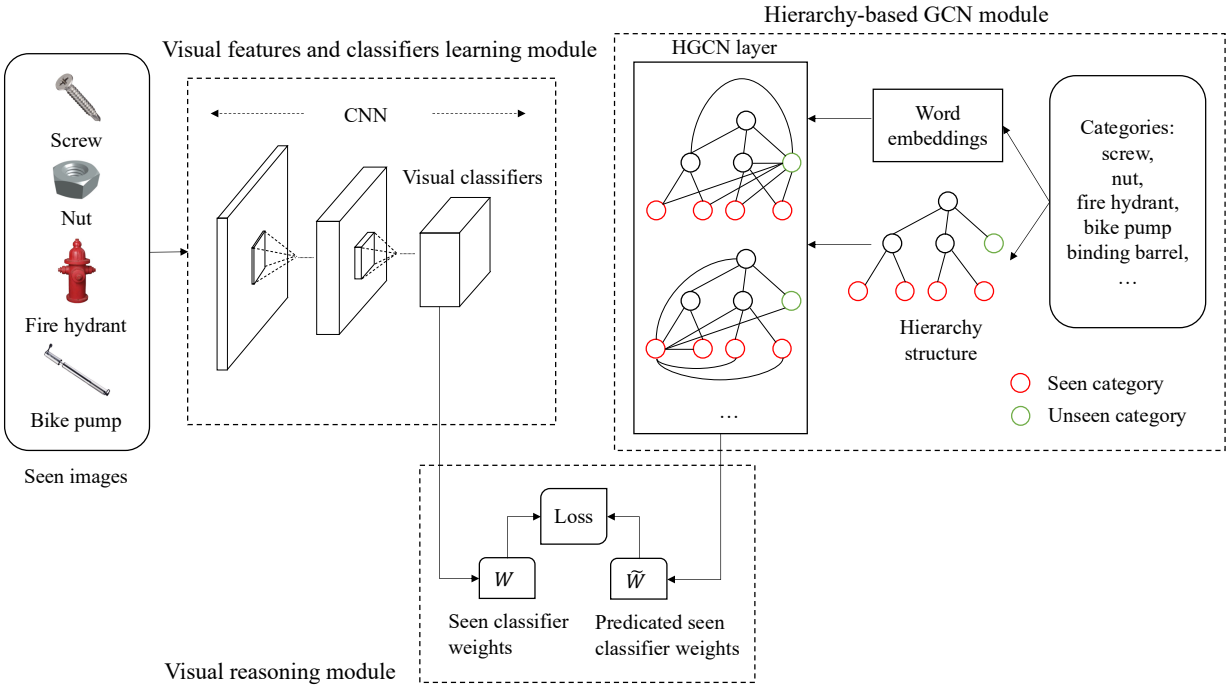


Figure 4 the visual reasoning framework

classifiers of the seen categories via a hierarchy relationship graph; finally, the extracted visual features of unseen images can be input into the reasoned unseen visual classifiers and predict the labels. Note that the labels of unseen images are only used for testing the performance of the reasoned visual classifiers of unseen images.

Our visual reasoning framework is illustrated in . The proposed framework contains three main modules: visual features and classifiers learning module, hierarchy-based GCN learning module, and visual reasoning module.

3.2 Visual features and classifiers learning module

Before visual reasoning, humans have some basic visual knowledge of objects they have seen. Visual knowledge can help them recognize visual features from some unseen or unfamiliar objects. In recent years, many visual feature extraction, detection, and recognition issues can be addressed by CNN, which are affected by the structure of the human visual system. ResNet[22] is a type of CNN that uses a residual network to solve the problem that the CNN is difficult to train due to the increase of network layers. In this module, we use pre-trained ResNet-50 as a backbone and train the model with images from seen categories for learning visual features and classifiers as visual knowledge. The last fully connected (FC) layer includes the learned weights which are the seen visual classifiers. The outputs before the FC layer are the feature representations of the input images. The trained ResNet-50 model can be used for extracting visual features of objects in unseen categories.

3.3 Hierarchy-based GCN module

In traditional neural networks (such as multilayer perceptron with fully connected layers), there is no explicit relations between the data samples, and they are assumed to be independent. GCN aims to take the neighborhood relationships into consideration and create the feature representation of each node not only by its own features but also using its neighbors[23]. More specifically, given a graph with N nodes and S features per node, $X \in \mathbb{R}^{N\times S}$ denotes the feature matrix. Here each node represents one distinct category, and each category is represented by the word embedding of the category name. The connections between the categories in the knowledge graph are encoded in form of a symmetric adjacency matrix $A \in \mathbb{R}^{N\times N}$. GCN employs a simple propagation rule to perform convolutions on each layer of the model, which is shown below.

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \qquad (1)$$

Where $\widetilde{A} = A + I_N$ is the adjacency matrix of the undirected graph G with added self-connection of each node. I_N is an identity matrix. $\widetilde{D}_{ii} \in \sum_j \tilde{A}_{ij}$ is a degree matrix. $\widetilde{D}^{-\frac{1}{2}}$ is used to normalize rows in $\widetilde{A}$. $H^{(l)}$ represents the activations in the l^{th} layer and $W \in \mathbb{R}^{S\times F}$ denotes the trainable weight matrix for layer l with F corresponding to the number of the learned visual classifiers. For the first layer, $H^{(0)}$ = X. $\sigma(\cdot)$ denotes a nonlinear activation function, in our case a Leaky ReLU.

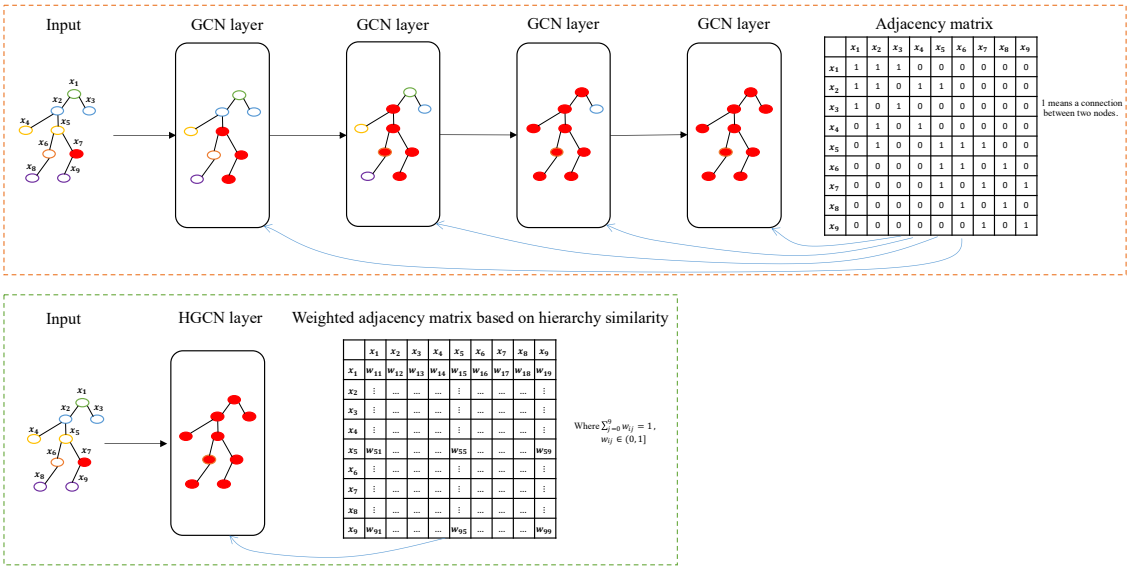


Figure 5 The comparison of GCN with HGCN. Take the node x_7 as an example. At the beginning of GCN, node x_7 only contains its own feature. After 1-layer GCN, node x_7 acquires the features of its one distance neighborhood nodes x_6 and x_9. At the same time, node x_6 is also updated by the features of its one distance neighbors, so do node x_9. And after 2-layer GCN, node x_7 gets the updated features of its one distance neighborhood nodes x_6 and x_9 again. Since the features of nodes x_6 and x_9 already contain the features of their one distance neighbors after the previous GCN, node x_7 indirectly obtains the features of the two distance neighborhood nodes. Thus, after 4 layers, node x_7 can merge features from all neighborhood nodes. For HGCN, we add virtual edges between node x_7 and nodes indirectly connected to it. Hence, after 1-layer HGCN, node x_7 can obtain the features of all nodes with paths to it.

However, feature fusion on one layer of GCN only considers the nearest neighborhood dependency. When the features of neighbors from k distances are required for further relation extraction, they can only be indirectly acquired through a multilayer GCN propagation, which has a high tendency of over-smoothing and makes nodes from the different classes indistinguishable[24]. To avoid this limitation and realize a long-distance feature fusion in one single layer, we propose a hierarchy-based graph convolutional network (HGCN). In the proposed model, we use the semantic similarity to construct a weighted adjacency matrix (WAM), which can directly figure out far neighborhood dependence with only one layer. The comparison between GCN and our proposed HGCN is shown in Figure 5. The main difference is we use a weighted adjacency matrix to propagate information in one shot. The method to obtain those weights is illustrated below.

For an acyclic graph $G(V, E)$, where V denotes the nodes and E denotes the edges specifying the hyponymy relations between semantic concepts. In other words, an edge $(u, v) \in E$ means that v is a sub-class of u. An example for such a graph with the special property of being a tree is given in Figure 6.

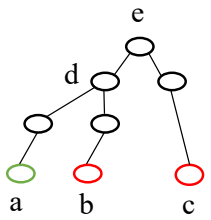


Figure 6 A toy hierarchy structure

Classes of interest (seen and unseen classes) are leaf nodes in the tree. The semantic distance d_G between two leaf classes is given as below.

$$d_G(u, v) = \frac{2 * height(lcs(u, v)) - height(u) - height(v)}{2 * \max_{\omega \in V} height(\omega) + 1} \quad (2)$$

where the height of a node is defined as the length of the longest path from that node to any of its descendants. The lowest common subsumer (lcs) of two nodes is the ancestor of both nodes that does not have any child being an ancestor of both nodes as well. One node can be its own ancestor. The semantic similarity s_G between semantic concepts can be calculated as:

$$s_G(u, v) = 1 - d_G(u, v) \quad (3)$$

where s_G is bounded between 0 and 1 as d_G is in the range (0, 1].

For example, the toy hierarchy in Figure 6 has a total height of 3, the lcs of the classes “a” and “b” is “d” and the lcs of the

classes “a” and “c” is “e”. It follows that $d_G(a,b) = \frac{2*2-0-0}{2*3+1} = \frac{4}{7}$, $s_G(a,b) = \frac{3}{7}$ and $d_G(a,c) = \frac{2*3-0-0}{2*3+1} = \frac{6}{7}$, $s_G(a,c) = \frac{1}{7}$.

The algorithm of constructing WAM is shown in Algorithm 1.

Algorithm 1 Calculate Weighted Adjacency Matrix (WAM) based on hierarchy similarity

Input: G: a graph represents the hierarchy structure of classes; N is the number of nodes in G
Output: WAM
initialize $WAM \in \mathbb{R}^{N\times N}$, all elements in WAM are zero
traverse every node i in G
  traverse every node j in G
    calculate the semantic similarity $s_G(i,j)$ using (2) and (3)
Normalization. Set $WAM_{ij} = softmax(s_G(i,j)) = \frac{\exp(s_G(i,j))}{\sum_{j=0}^{N}\exp(s_G(i,j))}$

After WAM is obtained, a new propagation formula for the fusion of hierarchy semantic information shown as follows:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{HWAM}\widetilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \quad (4)$$

where $\widetilde{HWAM} = HWAM + I_N$ and $\widetilde{D}_{ii} \in \sum_j \widetilde{HWAM}_{ij}$.
In this way, 1-layer HWGCN can integrate short and long-distance neighborhood information directly without multiple-layer propagations.

3.4 Visual reasoning module

In this work, we perform visual reasoning using semantic distances of seen and unseen categories embedded in the knowledge graph and the learned visual classifiers from the seen classes to predict the categories of unseen objects. More specifically, we need to infer the visual classifiers for the unseen categories.

The weights in the last layer of the trained ResNet-50 are interpreted as visual classifiers, which can determine the categories for the seen images. In order to predict a new set of weights for each unseen category, we parse a dependency tree as a graph structure on the seen and unseen categories. In the graph, a node is represented by the word embedding of each category’s name. And if there is a dependency between categories, there is an edge between corresponding nodes. The weight on each edge is determined by our proposed Algorithm 1. During the graph convolutional operation in the layer of HGCN, information of each node can be updated by fusing the feature of its short and long-distance neighborhood nodes. After training the HGCN, we can predict the visual classifiers of unseen categories based on the visual classifiers of seen categories. The loss function used to train HGCN is shown in Eq.(5).

$$\mathcal{L} = \frac{1}{2M}\sum_{i=1}^{M}\sum_{j=1}^{P}(W_{i,j} - \widetilde{W}_{i,j}) \quad (5)$$

where $\widetilde{W} \in \mathbb{R}^{M\times P}$ denotes the predicted visual classifiers of HGCN for the seen categories. M denotes the number of seen categories and P denotes the dimensionality of the weight vectors. The ground truth weights are obtained by extracting the last layer weights of the trained ResNet-50 and denoted as $W \in \mathbb{R}^{M\times P}$.

From the loss function, we can see that HGCN is trying to align the predicted and ground truth visual classifiers of seen categories. This information can be transferred and propagated in the graph and used to reason the visual classifiers of unseen categories.

4. EXPERIMENTS

4.1 Dataset

We evaluate the performance of our proposed methods on one benchmark dataset, which is called CADSketchNet[25]. It contains one computer-generated sketch for each representative image in the Mechanical Components Benchmark(MCB) dataset[26]. This results in 58,696 computer-generated sketches across 68 categories. Based on the hierarchy of the MCB dataset and the categories in the CADSketchNet. A modified hierarchy structure is shown in .The red dots in the Figure 7 are the 68 categories in CADSketchNet and they are all leaf nodes in the hierarchy structure. We randomly adopt 18 categories as unseen categories and the remaining 50 categories as seen categories. The number of sketches in each category can be found in [25]. The number of all nodes in the hierarchy structure is 232.

4.2 Implementation Details

We adopt the ResNet-50 model that has been pre-trained on the ImageNet 2012 dataset as the backbone. The pretrained ResNet-50 can learn some common-sense visual knowledge from 1000 categories. Then ResNet-50 is trained for 50 epochs using stochastic gradient descent with a learning rate of 0.001 and momentum of 0.9. The learning rate decays by 0.1 from 0.01 at every 10 epochs. The ResNet-50 is trained with the images from seen categories to learn visual features from mechanical component sketches. After the training, we can get visual classifiers of seen categories from the last layer of ResNet-50. Each visual classifier is represented by a weight vector that has 2048 dimensions. We extract word vectors to represent semantic information of our categories in the graph via the GloVe text model[27] trained on the Wikipedia dataset. Each category can be presented by a 300-dimensional vector. 232 categories in the hierarchy structure are used as the input of our proposed HGCN model. The HGCN model consists of one layer as illustrated in Eq. (4). For the layer, we make use of Dropout [28] operation with a dropout rate of 0.5 and leaky ReLUs with a negative slope of 0.2. Each predicted visual classifier of the HGCN model has 2048 dimensions which correspond to the dimensions of the learned visual classifiers of the ResNet-50. We perform L2-

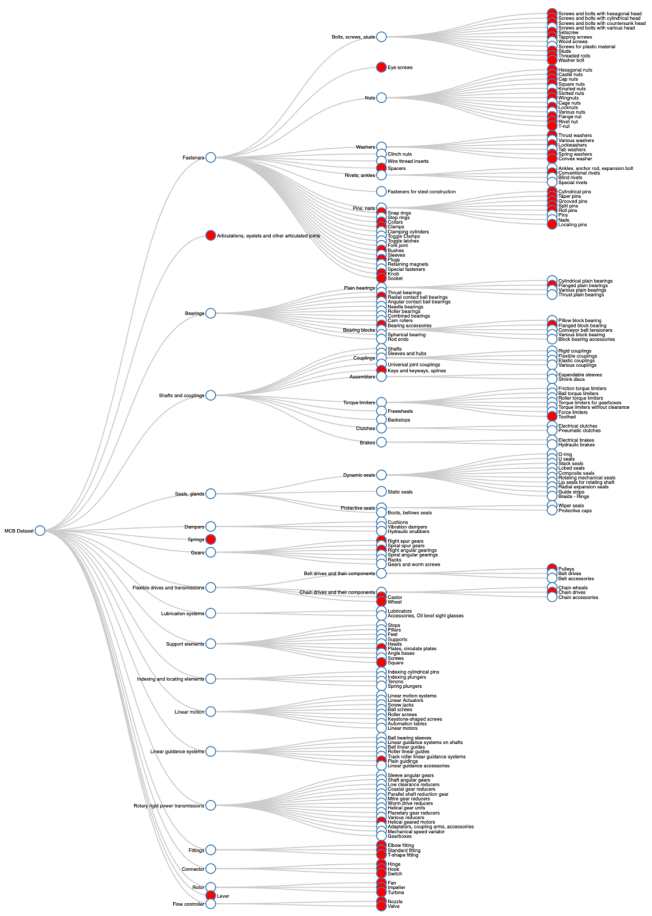


Figure 7 The hierarchy taxonomy of mechanical component categories

Normalization on the predicted visual classifiers of the HGCN and the ground truth visual classifiers produced by the ResNet-50 as it regularizes them into similar ranges. The loss function is the mean squared error between them, which is shown in Eq.(5). The model is trained for 3000 epochs with a learning rate of 0.001 and weight decay of 0.0005 using Adam[29]. All experiments are implemented with PyTorch[30] and training and testing are performed on a GTX 1080Ti GPU.

4.3 Performance comparison

Baseline method. We compare our proposed method with following methods. Devise[31] learns transformations of visual and semantic features to a common space. An unseen image's category can be determined by mapping the image to the common space and finding the nearest word-embedding in the space. ConSE[32] transforms image features into a semantic word embedding space through a weighted combination of several closest seen categories' semantic embeddings. The weights are predicted using pre-trained visual classifiers. ConSE assigns labels to unseen images according to the nearest categories in the semantic embedding space. GCNZ[33] is the approach most related to our proposed method. The main difference is our HGCN uses a hierarchy structure to determine a weighted adjacency matrix, which can quickly propagate information and achieve a better performance in visually reasoning unseen objects.

Quantitative results. Our metric is top-k accuracy, which is based on the percentage of assigning correct labels on unseen images out of top-k predictions. The processes to obtain the quantitative results are as follows. Let us assume, we have N unseen images, P unseen labels and Q seen labels. The ground truth label of an unseen image belongs to one of the P unseen labels. The Q seen visual classifiers can be learned based on section 3.2. Based on sections 3.3 and 3.4, the P unseen visual classifiers can be learned by transferring the learned seen visual classifiers through the taxonomy structure in Figure 7. An unseen image will be input to Q seen visual classifiers and P unseen visual classifiers to have P+Q predicted values. All predicated values will be ranked in descending order. If the predicated value of the ground truth label of the unseen image is among top k of the rank. It is a successful hit. We go through N unseen images and count the number of successful hits as M. The top-k accuracy is M/N. We set k to be 1, 2, 5, 7, and 10 in the experiments. Firstly, we perform evaluations on the task of 18 learned unseen visual classifiers and 0 seen visual classifiers. Secondly, we perform evaluations on the task of 18 learned unseen and 50 seen visual classifiers with the same metric and the same k settings. The results are shown in Table 1. We can observe that (1) Our model and GCNZ outperform the Devise and ConSE baselines by a large margin in two scenarios as these methods require a larger dataset to train and learn the connection between visual features and semantic features. (2) since the seen class classifiers are added to the classifiers in the second scenario, the performance of all models drops partly. (3) our model maintains comparable performance when comparing with GCNZ in two scenarios as our method can include hierarchy relationships between seen and unseen categories, which is useful to transfer learned knowledge to infer unknown objects. These observations further demonstrate the effectiveness of our proposed approach to visually reason unseen images.

Table 1 Top-k accuracy for the different models on the CADSketchNet dataset using visual classifiers of unseen categories and unseen categories combined with seen categories

Visual classifiers	Models	Hit@*k*(%)				
		1	2	5	7	10
Unseen categories	Devise	4.2	13.0	34.3	49.0	67.4
	ConSE	4.8	13.2	38.2	50.4	70.5
	GCNZ	6.3	33.6	49.4	57.1	71.5
	HGCN	**16.5**	**46.3**	**65.7**	**68.4**	**75.4**
Unseen + seen categories	Devise	3.6	5.0	11.5	13.7	15.9
	ConSE	5.6	9.4	11.6	16.3	23.6
	GCNZ	7.8	13.6	23.3	27.2	32.1
	HGCN	**13.9**	**16.9**	**38.6**	**43.9**	**50.4**

Qualitative results. Example images from unseen categories are displayed, and we compare the performance of our proposed HGCN with Devise, ConSE and GCNZ to predicate the top 5 categories from 18 unseen categories. For HGCN and GCNZ, we use the learned 18 unseen visual classifiers to classify

example images and obtain the top 5 highest probability among 18 unseen classes. For ConSE and GCNZ, we infer the word embedding of the example images and find the nearest 5 word embeddings from 18 unseen categories. We observe that HGCN and GCNZ generally provide coherent top-5 results and Devise and ConSE also offer similar top-5 results. Our proposed method can have better performance compared with other models. All models struggle to predict the "wingnut" and tend to predict detail features, such as threads and cylindrical shapes; however, HGCN does include the wingnut category in the top-5 results. The reason is HGCN takes advantage of semantic distances to weight visual features and considers visual and semantic similarity at the same time when predicting labels.

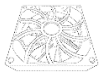
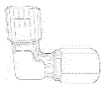
Test Image	Devise	ConSE	GCNZ	HGCN
	1. Screws and bolts with hexagonal head, 2. Screws and bolts with cylindrical head, 3. Cylindrical pins, 4. Threaded rods, 5. Grooved pins	1. Wheel, 2. Cylindrical pins, 3. Radial contact ball bearings, 4. Plugs, 5. Threaded rods	1. Radial contact ball bearings, 2. Right angular gearings, 3. Elbow fitting, 4. Grooved pins, 5. Chain drives	1. Radial contact ball bearings, 2. **Fan**, 3. Impeller, 4. Cylindrical pins, 5. Chain drives
	1. Cylindrical pins, 2. Threaded rods, 3. Screws and bolts with cylindrical head, 4. Screws and bolts with hexagonal head, 5. Grooved pins	1. Cylindrical pins, 2. Grooved pins, 3. Radial contact ball bearings, 4. Screws and bolts with cylindrical head, 5. Threaded rods	1. Radial contact ball bearings, 2. Chain drives, 3. Cylindrical pins, 4. Fan, **5. Elbow fitting**	**1. Elbow fitting**, 2. Right angular gearings, 3. Chain drives, 4. Radial contact ball bearings, 5. T-shape fitting
	1. Right angular gearings, 2. Radial contact ball bearings, 3. Impeller, 4. Grooved pins, 5. Elbow fitting	1. Radial contact ball bearings, 2. Threaded rods, 3. Chain drives, 4. Cylindrical pins, 5. Elbow fitting	1. Screws and bolts with cylindrical head, 2. Screws and bolts with hexagonal head, 3. Cylindrical pins, **4. Wheel**, 5. Grooved pins	**1. Wheel**, 2. Radial contact ball bearings, 3. Cylindrical pins, 4. Chain drives, 5. Screws and bolts with cylindrical head
	1. Screws and bolts with cylindrical head, 2. Cylindrical pins, 3. Screws and bolts with hexagonal head, 4. Threaded rods, 5. Grooved pins	1. Screws and bolts with cylindrical head, 2. Cylindrical pins, 3. Screws and bolts with hexagonal head, 4. Threaded rods, 5. Lock washers	1. Threaded rods, 2. Elbow fitting, 3. Right angular gearings, 4. Impeller, Lock 5. Washers	1. Lock washers, 2. Elbow fitting, 3. Impeller, 4. Grooved pins, **5. Wingnuts**

Figure 8 Test images from CADSketchNet and their corresponding top 5 labels predicted by learned 18 unseen visual classifiers four different models. The correct labels are shown in bold. Examples are randomly picked from 18 unseen categories.

5. DISCUSSION

Fusion of visual and semantic similarity: The key point of visual reasoning is to transfer the visual knowledge from seen categories (source domains) to understand unseen categories (target domains). To achieve this purpose, it is usually necessary to explicitly explore the connections between seen categories and unseen categories for knowledge transformation. GCNZ has powerful capabilities in exploiting category relationships. However, it is weak in coalescing visual and semantic information when learning the visual classifiers. In other words, some visual classifiers may be tightly clustered together in the feature space because of their high visual similarities. However, these visual classifiers may not be semantically related to each other. We propose HGCN, which can manipulate the feature representation of a visual classifier by fusing its neighbors' representation with different weights based on the hierarchy relationships using semantic distance d_G, which can be regarded as the "knowledge distance"[3] or "semantic distance"[4] to measure the proximity between the source and target domains. The effect of semantic distance d_G to the cross-category reasoning is pull images having both high visual and semantic similarities to the neighborhood of a certain visual classifier. In the qualitative results of the experiment, HGCN can include categories sharing the same parent or grandparent with the category of the test image. The reason is they have shorter knowledge distances. In Figure 8, for the test image "fan", HGCN predicts "impeller" label which shares the same parent "rotor" with "fan"; "T-shape fitting" label is predicated for the test image "elbow fitting", as they are children categories of "fittings"; "Chain drive" is predicated for the test image "wheel", as "chain drive" is a nephew of "wheel". However, GCNZ ranks visually similar categories higher and misses those semantically similar categories. The knowledge distance information provides the basis for guiding reasoning by fusing different degrees of visual knowledge from near and far distance categories.

Analysis of the number of layers in HGCN: We perform an empirical evaluation to verify our motivation which is applying multiple layers to the GCN could cause a drop in performance. Table 2 illustrates the performance when using one layer or multiple layers to GCNZ and HGCN for top k accuracy evaluation using unseen visual classifiers. The dimensions of one layer or multiple layers in HGCN are the same as the dimensions of GCNZ. For GCNZ, multiple layers perform better than one layer. The reason is one layer can only mix the features of a node and its one-distance neighbors' features. Meanwhile, multiple layers can integrate the features of neighbors from a long distance. However, for HGCN, multiple layers perform worse than one layer. The reason is HGCN utilizes a weighted adjacency matrix (WAM) to mix the features of a node and its short and long-distance neighbors' features in one shot and uses different weights to determine the magnitude to integrate information from its neighbors. Therefore, multiple layers can bring potential concerns of making categories indistinguishable through redundant propagations. We can see that to have better performance of GCNZ, some experiments need to be done to

find an optimal number of layers. This effort is not necessary for HGCN.

Table 2 Results for GCNZ and HGCN models with different sizes of layers when using unseen visual classifiers

Models	Hit@*k*(%)				
	1	2	5	7	10
GCNZ(6 layers)	6.3	33.6	49.4	57.1	71.5
GCNZ(one layer)	5.2	20.3	44.5	52.8	69.9
HGCN(6 layers)	14.4	41.7	61.4	66.3	74.8
HGCN(one layer)	**16.5**	**46.3**	**65.7**	**68.4**	**75.4**

Visual reasoning for design by analogy: Analogical reasoning applies the knowledge from a well-known domain (the source domain) to another less-known domain (the target domain). In this paper, visual reasoning is a type of analogical reasoning. Our proposed visual reasoning framework provides a way to transfer visual knowledge (visual classifiers) from the familiar (seen) objects to unfamiliar (unseen) objects using semantic knowledge (semantic embeddings and the hierarchy structure of different categories) to link these objects. Few researchers have developed a computational framework to support design by visual analogy through a semantic modality. With enormous amounts of labeled image data, deep learning methods have achieved impressive breakthroughs in various tasks. However, the need for large quantities of labeled images is still a bottleneck in the engineering design field. Our proposed framework can fulfill the need by learning the transferable visual knowledge from the seen dataset where ample labeled images are available and the semantic knowledge from seen and unseen categories to generalize to another dataset which includes labeled unseen images. By enlarging the image dataset, design by analogy can be empowered by exploring more domains.

6. CONCLUSION

In this paper, we propose a visual reasoning framework that unifies both visual and semantic modalities for design by analogy. In engineering design, many researchers have proven that a large assortment of visual displays can stimulate designers to make visual analogies and generate creative design concepts[3, 15, 34, 35]. The processes of visual reasoning are happening in designers' minds. However, our research has demonstrated the potential of using convolutional neural networks and graph neural networks to mimic the visual reasoning processes. Through the model building and the experiment results, the following conclusions are drawn.

1. The integration of CNN and HGCN are introduced to learn visual knowledge from source domains and transfer the visual knowledge to target domains based on their semantic distances.
2. Hierarchy weighted adjacency matrix is proposed to mix short and long-distance neighbors' information using only one layer in the HGCN, which can help distinguish neighbors based on semantic similarities.
3. The visual reasoning framework can be utilized to create more labels for engineering component images to support data-driven design by analogy.

A drawback of the proposed framework is the need to predefine the hierarchy structure of the categories every time we have a new dataset. In future work, we aim to investigate more semantic information about mechanical components, not only the category names but also attributes, such as functions. Meanwhile, advanced weighting mechanisms will be explored to further improve the performance of HGCN.

REFERENCES

[1] Goldschmidt, G., and Smolkov, M., 2006, "Variances in the impact of visual stimuli on design problem solving performance," Design Studies, 27(5), pp. 549-569.

[2] Goldschmidt, G., 2003, "The backtalk of self-generated sketches," Design issues, 19(1), pp. 72-88.

[3] Casakin, H., and Goldschmidt, G., 1999, "Expertise and the use of visual analogy: implications for design education," Design studies, 20(2), pp. 153-175.

[4] Zhang, Z., and Jin, Y., "An Unsupervised Deep Learning Model to Discover Visual Similarity Between Sketches for Visual Analogy Support," Proc. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, p. V008T008A003.

[5] Zhang, Z., and Jin, Y., "Toward Computer Aided Visual Analogy Support (CAVAS): Augment Designers Through Deep Learning," Proc. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers, p. V006T006A057.

[6] Cavanagh, P., 2011, "Visual cognition," Vision research, 51(13), pp. 1538-1551.

[7] Zwaan, R. A., and Taylor, L. J., 2006, "Seeing, acting, understanding: Motor resonance in language comprehension," Journal of Experimental Psychology: General, 135(1), p. 1.

[8] Carlesimo, G. A., Perri, R., Turriziani, P., Tomaiuolo, F., and Caltagirone, C., 2001, "Remembering what but not where: independence of spatial and visual working memory in the human brain," Cortex, 37(4), pp. 519-534.

[9] Mellet, E., Tzourio-Mazoyer, N., Bricogne, S., Mazoyer, B., Kosslyn, S., and Denis, M., 2000, "Functional anatomy of high-resolution visual mental imagery," Journal of Cognitive Neuroscience, 12(1), pp. 98-109.

[10] Goel, A. K., Rugaber, S., and Vattam, S., 2009, "Structure, behavior, and function of complex systems: The structure, behavior, and function modeling language," Ai Edam, 23(1), pp. 23-35.

[11] Chakrabarti, A., Sarkar, P., Leelavathamma, B., and Nataraju, B., 2005, "A functional representation for aiding biomimetic and artificial inspiration of new ideas," Ai Edam, 19(2), pp. 113-132.

[12] Dering, M. L., and Tucker, C. S., 2017, "A Convolutional Neural Network Model for Predicting a Product's Function, Given Its Form," Journal of Mechanical Design, 139(11), pp. 111408-111408-111414.
[13] Helfman Cohen, Y., Reich, Y., and Greenberg, S., 2014, "Biomimetics: structure–function patterns approach," Journal of Mechanical Design, 136(11), p. 111108.
[14] Linsey, J. S., Wood, K. L., and Markman, A. B., 2008, "Modality and representation in analogy," Ai Edam, 22(2), pp. 85-100.
[15] Jin, Y., and Benami, O., 2010, "Creative patterns and stimulation in conceptual design," AI EDAM, 24(2), pp. 191-209.
[16] Gonçalves, M., Cardoso, C., and Badke-Schaub, P., 2016, "Inspiration choices that matter: the selection of external stimuli during ideation," Design Science, 2.
[17] Toh, C. A., and Miller, S. R., 2014, "The impact of example modality and physical interactions on design creativity," Journal of Mechanical Design, 136(9).
[18] Jiang, S., Luo, J., Ruiz-Pava, G., Hu, J., and Magee, C. L., 2021, "Deriving design feature vectors for patent images using convolutional neural networks," Journal of Mechanical Design, 143(6), p. 061405.
[19] Kwon, E., Pehlken, A., Thoben, K.-D., Bazylak, A., and Shu, L. H., 2019, "Visual Similarity to Aid Alternative-Use Concept Generation for Retired Wind-Turbine Blades," Journal of Mechanical Design, 141(3).
[20] Zhu, X., Anguelov, D., and Ramanan, D., "Capturing long-tail distributions of object subcategories," Proc. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 915-922.
[21] Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J., 2017, "Building machines that learn and think like people," Behavioral and brain sciences, 40.
[22] He, K., Zhang, X., Ren, S., and Sun, J., "Deep residual learning for image recognition," Proc. Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778.
[23] Kipf, T. N., and Welling, M., 2016, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907.
[24] Li, Q., Han, Z., and Wu, X.-M., "Deeper insights into graph convolutional networks for semi-supervised learning," Proc. Thirty-Second AAAI conference on artificial intelligence.
[25] Manda, B., Dhayarkar, S., Mitheran, S., Viekash, V., and Muthuganapathy, R., 2021, "'CADSketchNet'-An Annotated Sketch dataset for 3D CAD Model Retrieval with Deep Neural Networks," Computers & Graphics, 99, pp. 100-113.
[26] Kim, S., Chi, H.-g., Hu, X., Huang, Q., and Ramani, K., "A large-scale annotated mechanical components benchmark for classification and retrieval tasks with deep neural networks," Proc. Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16, Springer, pp. 175-191.
[27] Pennington, J., Socher, R., and Manning, C. D., "Glove: Global vectors for word representation," Proc. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543.
[28] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., 2014, "Dropout: a simple way to prevent neural networks from overfitting," The journal of machine learning research, 15(1), pp. 1929-1958.
[29] Kingma, D. P., and Ba, J., 2014, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980.
[30] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A., 2017, "Automatic differentiation in pytorch."
[31] Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A., and Mikolov, T., "Devise: A deep visual-semantic embedding model," Proc. Advances in neural information processing systems, pp. 2121-2129.
[32] Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G. S., and Dean, J., 2013, "Zero-shot learning by convex combination of semantic embeddings," arXiv preprint arXiv:1312.5650.
[33] Wang, X., Ye, Y., and Gupta, A., "Zero-shot recognition via semantic embeddings and knowledge graphs," Proc. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6857-6866.
[34] Yang, M. C., 2009, "Observations on concept generation and sketching in engineering design," Research in Engineering Design, 20(1), pp. 1-11.
[35] Linsey, J. S., Clauss, E., Kurtoglu, T., Murphy, J., Wood, K., and Markman, A., 2011, "An experimental study of group idea generation techniques: understanding the roles of idea representation and viewing methods," Journal of Mechanical Design, 133(3), p. 031008.