This paper received the 2021 ASME Design Theory and Methodology Award

Proceedings of the ASME 2021
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC/CIE2021
August 17-20, 2021, Virtual, Online

IDETC2021-70961

TOWARD COMPUTER AIDED VISUAL ANALOGY SUPPORT (CAVAS): AUGMENT DESIGNERS THROUGH DEEP LEARNING

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

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

ABSTRACT

The goal of this research is to develop a computer-aided visual analogy support (CAVAS) framework that can augment designers' visual analogical thinking by providing relevant visual cues or sketches from a variety of categories and stimulating the designer to make more and better visual analogies at the ideation stage of design. The challenges of this research include *what roles a computer tool should play in facilitating visual analogy of designers*, *what the relevant and meaningful visual analogies are at the sketching stage of design*, and *how the computer can capture such meaningful visual knowledge from various categories through analyzing the sketches drawn by the designers*. A visual analogy support framework and a deep clustering model, called *Cavas-DL*, are proposed to learn a latent space of sketches that can reveal the shape patterns for multiple categories of sketches and at the same time cluster the sketches to preserve and provide category information as part of visual cues. The latent space learned serves as a visual information representation that captures the learned shape features from multiple sketch categories. The distance- and overlap-based similarities are introduced and analyzed to identify long- and short-distance analogies. Extensive evaluations of the performance of our proposed methods are carried out with different configurations, and the visual presentations of the potential analogical cues are explored. The evaluation results and the visual organizations of information have demonstrated the potential of the usefulness of the Cavas-DL model.

Keywords: Computational support for visual analogy making, visual similarity, unsupervised deep learning, design by analogy, sketching, fixation

1. INTRODUCTION

In engineering design, mental stimulation is useful to boost innovative solutions for ill-defined design problems. During conceptual design, designers, especially novices, usually struggle in choosing among various sources to gain insights when attempting to generate creative concepts. In our previous work, it has been shown that the shapes and structures, in addition to behaviors, of a design artifact tend to be more stimulating than the functions [1]. Researchers have observed that designers often search intensively for images from various websites for inspiration [2, 3]. Most existing design-dedicated analogy search tools and methods [4-6] require designers to initiate a search by entering keywords and use semantic-based approaches for fixation avoidance. Few computational tools exist to support design-by-analogy based on visual similarity analysis. The core research problem in this paper is to explore the roles of computational support for visual analogy and investigate how to learn visual features from raw image data, and discover potential short- and long-distance analogies relevant visual information based on visual similarities.

Sketching is an efficient way for designers to have their brief and ambiguous ideas taking shapes on paper [7]. The briefness accelerates the transformation of a rough thought into a reality. The ambiguity of an open-ended visual representation contributes to more possible interpretations. Sketching in conceptual design provides potentially meaningful clues for a designer to infer emerging design concepts [8, 9]. The inspiration of sketches mostly comes from the *shapes* and the relationships among them. Designers can manipulate given shapes in imagery and combine them into meaningful and even new concepts in a short time. Sketching can reflect premature design ideas in designers' minds, and it is also an ideal stimulant to facilitate

creative idea generation. Therefore, it is important to develop a computational tool to support designers in generating more creative ideas by stimulating their visual thinking process.

Research has been done to investigate visual analogy in the field of design. Goldschmidt and colleagues demonstrated that visual analogy is considered as an effective cognitive strategy to stimulate designers to create innovative concepts for solving ill-structured design problems [10-12]. For novel idea generation, the use of visual stimuli outperforms words [13, 14]. In design, shapes may represent semantic concepts and objects to reflect designers' understanding of the visual world. From a cognitive point of view, when making a visual analogy, designers can map shapes from high (geometric) dimensions to low (symbolic, conceptual) dimensions [15, 16]. At the low dimensions, they are capable of interpreting and detecting the similarities between shapes in the same or different categories. It means that designers can abstract perceptual information to some shape patterns which represent the shape features in a cognitive space [17]. In that space, they can manipulate and transform shapes by exploiting their domain knowledge. From an engineering design point of view, the high-dimensional geometric features signify the lower-dimensional semantic features [18, 19], meaning that the high-dimensional shape features can be reduced to a space of a low dimensionality that still preserves the underlying patterns, constraints, and configurations. It is more efficient to explore and exploit the low-dimensional design space to discover novel designs. In a similar spirit, computationally transforming *high dimensional* image sketches represented as *pixels* into *low dimensional* ones captured as *features* can, on the one hand, keep the underline shape patterns of the sketches, and on the other hand, allow the efficient computational shape analysis for preserving semantic meanings. An important question is: *how can a computation tool learn a low dimensional space, called latent space, which can capture the shape patterns of sketches from multiple categories*?

The precondition for making a visual analogy is a visual similarity existing between the source and target domains [2]. In most research on searching for visual stimuli, the magnitude of visual similarity is qualitatively determined by designers [19, 20]. A notion of distance is central to measure visual similarity. In the latent space, sketches are distributed based on their shape features. Clustering is an essential data analysis and visualization tool and provides a way to *group* sketches in the latent space based on the visual similarity. The traditional way of using a deep neural network for clustering images is not sufficient as it needs to do the training for extracting shape feature vectors first and then apply clustering algorithms on the extracted features into group images. Hence, the second research issue is: given a latent space for representing shape features from raw pixels, *how can a tool effectively cluster sketches into different shape groups based on their inherent shape patterns and analyze the short- and long-distance analogies based on the shape similarity?*

In this paper, unsupervised deep learning techniques are applied to build a model, called *CAVAS through deep learning,* or *Cavas-DL* for short, to learn a low dimensional latent space, in which shape patterns can be found to distill shape features of the sketches from multiple categories. A clustering layer is constructed to directly cluster images in the latent space during the training process. The distance- and overlap-based similarities are introduced to quantitatively measure visual relationships between one category and other categories in the latent space and determine short- and long-distance analogies for each category. Besides, the connections between different groups of categories are identified to explore how visual analogies can happen.

2. RELATED WORK

2.1 Computational tools for design by analogy

Design-by-analogy consists of two main steps: retrieving potentially inspirational information in the source domains and mapping the inspirational information from source domains to the target domain [21]. Designers often face difficulties when retrieving fitting inspirational sources. Therefore, using effective searching and retrieving tools has the potential to enhance design-by-analogy.

Biological systems provide a fruitful source of inspiration for engineering design. Vincent and Mann proposed Bio-TRIZ, which adds biological information and principles into the TRIZ database [22]. Chakrabarti et al. created an automated analogical tool called IDEA-INSPIRE that searches relevant ideas from a biological database to solve a given design problem [23]. Shu et al. used natural language analysis to correlate functional basis terms with useful biological keywords [24]. DANE (Design by Analogy to Nature Engine) was proposed by Goel et al. to search and retrieve the functioning of biological systems in the Structure-Behavior-Function library [25]. Nagel et al. put forward a computational method to generate biologically inspired concepts based on function-based design tools [26]. AskNature is a web-based tool to interactively classify biological information in the Biomimicry Taxonomy [27].

Patent databases can offer enormous cross-domain technical knowledge to inspire designers. Murphy proposed a search methodology to identify inspiring patents which have functional similarity with design problems [28]. A computation method was put forward for clustering patents based on their functional and surface similarity; then, designers can automatically retrieve analogical stimuli from these patents [29]. As many patent retrieval computational tools focus on mining patents generally, Song and Luo proposed a data-driven method to retrieve patents precisely related to a specific product [30]. Fu et al. proposed a technological distance to measure the "near" and "far" analogical stimuli based on the relative similarity of clusters of patents [31].

While the research into searching and retrieving analogies from biological systems and patents is prolific, the foundation of most research is in linguistics and semantic transfer for analogical reasoning. There are few computational methods and tools that support and guide visual analogy. Luo and his colleagues put forward visual analogy support tools based on visual maps of technology domains or technical concepts to guide the search for inspirations across domains or assist the analogical inference from concepts to concepts [32-34].

However, the big difference between the visual cues in this paper with theirs is that our visuals are the images and graphics, whereas their visuals are the structures of relations among semantic constructs and design domains.

2.2 Visual analogy in engineering design

CAD, sketches, photographs, and line drawings are the major visual sources that promote analogical thinking [2]. In engineering design, many researchers used a large assortment of visual displays to stimulate designers to generate creative design concepts. Jin and Benami indicated that meaningfulness and relevance are the two overwhelmingly important creative properties of visual stimuli that influence design stimulation [1]. Yang et al. showed that the quality and realism of the design can be improved when sketching during concept generation [8]. Goldschmidt et al. demonstrated that visual stimuli are useful for both expert and novice designers to improve the quality of design and more effective for novice designers [11, 12]. Linsey et al. illustrated that designers often prefer visual representations to textual descriptions for idea generation, and photographs are growing in popularity due to easy retrieval from the Internet [35, 36]. McKoy et al. showed that novice designers can generate higher quality and more novel design concepts when being presented with sketches rather than text-based examples [37].

However, displays of visual representations are less effective in producing creative design than reasoning by visual analogy. Casakin et al. found that if no instructions or directions are provided to guide visual analogy, the quality of the design solutions is mostly diminished [38]. It is often said that designers think more visually in their working environment. Designers are more likely to take advantage of shapes and forms of visual displays as stimuli to tackle given design problems [10]. Shape emergence means unexpected or implicit shape features and relations appear only after the manipulation and transformation of explicit shapes [15]. Visual imagery may provide a theoretical foundation for shape emergence in design by linking shape perceptions and cognitive processes of visual reasoning. Therefore, designers often take advantage of visual imagery to reinterpret and reformate underlying shapes from the visual stimuli for the idea generation. The precondition for shape emergence is shape ambiguity, which refers to the existence of numerous interpretations of the visual representation [39].

Designers are prone to use sketches to represent rough ideas and obtain hints from the shapes of sketches [7]. The sketch is an informal visual representation that has the property of ambiguity; designers can perceive two or more different shapes from one single sketch. Therefore, sketches are an ideal source to serve as a visual stimulus. How to effectively support visual analogy from sketches remains a major research question in the design research community.

2.3 Deep learning models for sketch representation

Recent advances in deep neural network models drastically increased computers' ability to learn a common and general feature space for sketches and images [40]. Karimi et al. used a supervised learning method to learn the feature vectors of sketches given the category labels and then create clusters of visually similar sketches based on the learned feature vectors [41]. Shuo et al. introduced a supervised CNN-based approach for patent image vectorization to support visual design stimuli retrieval in design-by-analogy [42]. However, in our research, the goal is to learn a latent space that represents the object shape features by using only lines and curves in the sketches rather than having the labels of categories. Therefore, an unsupervised learning approach is needed. *Sketch-rnn* is an unsupervised learning model based on Variational AutoEncoder (VAE) for constructing stroke-based drawings of common objects; it can mimic how humans sketch and draw similar but unique objects [43]. *Sketch-rnn* uses a bi-directional recurrent neural network (RNN) as an encoder to capture the features of training data in a latent space and applies an autoregressive RNN as a decoder to reconstruct data. However, the performance of *sketch-rnn* to extract shape features of objects from multiple categories is not satisfactory.

In summary, a rich body of research on *design by analogy* has yet to be expanded by integrating the extensive work on *visual analogy* and advanced *deep learning* technologies. Our goal in this paper is to fill the gap of the three research areas by developing a computational method that can learn the visual similarity from sketches and provide highly effective visual stimuli to enhance the visual analogy of designers.

3. CAVAS: A VISUAL ANALOGY SUPPORT FRAMEWORK

Creative designers usually employ inspirational sources that are not directly linked to the design problem at hand, take advantage of incidentally presented cues, and tend to collect a wide range of ideas, sometimes seemingly irrelevant and highly dissimilar, that may lead to insights. Divergent thinking helps designers imagine the world from multiple perspectives, see problems in new ways and escape stereotypical thinking. There is significant anecdotal and experimental evidence [2, 12] for the importance of visual analogy to stimulate the originality and creativity of designers. Simply trying to think of or reason analogies and analogous domains is difficult even for experienced engineers. One of the main principles for enhancing analogical reasoning is to provide a variety of related effective cues.

3.1 Major functions

Following our previous work on the *generate-stimulate-produce* (GSP) model of creative stimulation [1], a process of computer-aided visual analogy support, called CAVAS, can be introduced as shown in Figure 1. A designer initiates his/her design process by starting sketching. When the designer carries out the design alone, as shown in Figure 1(a), the sketches the designer generated will be perceived by the designer hence visually stimulate the designer and lead to further cognitive processes, such as *association* or *analogy*. The results of the cognitive processes will be the production of more design operations, such as *sketching*, which then will generate more *sketches* as design entities. The GSP process keeps going on as design ideas become clearer and design concepts are solidified.

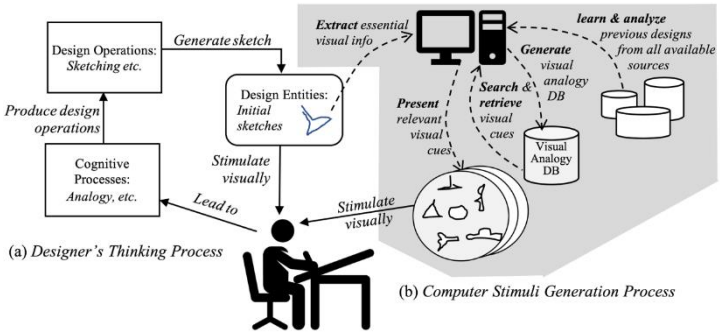

Figure 1: The proposed computer aided visual analogy support (CAVAS) in a human-computer interaction framework

The computer support in the proposed CAVAS is based on a human-computer interaction framework, in which the role of the computer is defined as "to augment the human designer's thinking and imagination capability by providing highly *relevant* and *stimulating* visual cues to the designer at the right timing during the early idea shaping stage of design." As shown in Figure 1(b), for a computer system, called the CAVAS system, to fulfill this role, it must possess the following six major functions, namely, *learn, analyze, generate, extract, search, retrieve*, and *present*.

***Learn* and *analyze* previous designs from all available sources:** The previous design materials such as sketches, CAD drawings, photographs, and line drawings in the open-source datasets are collected and converted into images. The visual patterns of these images can be learned and represented by the CAVAS system. The system can analyze the visual similarity between different domains based on the learned representations.

***Generate* visual analogy databases**: After learning and analyzing previous designs, the CAVAS system can generate visual knowledge in visual and textual formats, which captures the shape patterns of, and similarity relationships among, the visual components. The generated knowledge is stored in visual analogy databases, which can be reused and updated.

***Extract* essential shape information, *search*, and *retrieve* visual analogies:** The sketches drawn by designers are fed into the CAVAS system. The system can extract and represent the essential shape information from the sketches, search and retrieve the relevant visual cues in the visual analogy database.

***Present* relevant visual cues:** After the relevant visual cues are retrieved from the visual analogy database, the CAVAS system then presents the visual cues to designers in visually appealing ways so that the designers are stimulated to find appropriate source analogies from their memory and external databases. The visual cues should increase the chances for designers to retrieve relevant visual analogies.

3.2 Visual augmentation processes

Among the major functions in the CAVAS framework described above, ***learn*** and ***analyze*** functions are the key ones. Figure 2 shows the entire visual augmentation process, consisting of two main functions and six stages.

Stage 1: Sketches are collected as *previous designs*. In this research, the visual cues to be used as visual stimuli are identified based on shape similarities. In the eyes of a particular viewer, a sketch could bear a resemblance to an object, person, animal, texture, or place. This ability of *cross-domain transformation* of shapes can provide a degree of diversity, ambiguity, and uncertainty in the information gathering and idea generation process, making it possible for designers to seek inspiration from other domains.

Stage 2: Instead of identifying similar sketches in the enormously high dimensional pixel space, a *dimension reduction* approach is taken to transform images into a feature-based space where *shape features* are identified. Once this shape-feature based space, called *latent space*, is established, it becomes computationally feasible to analyze the sketches to provide relevant visual cues to the designers.

Stage 3: The inherent *shape patterns* of collected sketches can be discovered by analyzing and comparing their *shape features* in the latent space. A soft clustering approach is taken to cluster the sketches into different shape clusters based on their "distances" in the latent space. Each sketch is assigned different probabilities of belonging to multiple groups, preserving the ambiguity essential for supporting designers' visual analogy. It is assumed that 1) visually similar shapes should be clustered in the same group to represent one shape pattern and 2) the sketches of different categories, but belong to the same group, can be more effective in stimulating designers' analogical thinking due to the shape similarity.

Stage 4: As the clustering process converges, the size of each cluster becomes stable. A ratio is calculated based on dividing the number of cluster assignment changes by the total number of sketches. If it is smaller than the predefined threshold δ, then exit the learning process and jump to stage 5; otherwise, proceed to stage 2.

Stage 5: Two metrics are introduced to analyze the visual similarity between sketches. The first metric is called *distance-based similarity*, measuring the distances among centroids of different sketch categories in the latent space, shorter distance meaning higher similarity. The second metric is called *overlap-based similarity*, which measures the amount of overlap among cluster probability distributions of different sketch categories, larger overlap meaning higher similarity. These two metrics work together to deal with different scenarios and provide more accurate measurements for visual similarity.

Stage 6: Long- and short-distance analogies of each sketch category are identified based on visual similarity measures

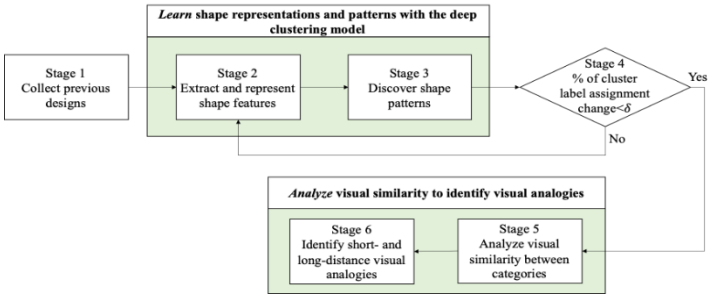

Figure 2: The process of learn and analyze functions in the CAVAS framework

mentioned above. Sketch categories with high visual similarity are classified as short-distance visual analogies, otherwise, as long-distance visual analogies. *Bridge categories* are identified to provide a way to discover valid long-distance visual analogies.

The proposed visual augmentation process is applied to sketches from Quickdraw [44] as a case study. Sections 4 presents the two main functions of the CAVAS framework.

4. METHODS

4.1 Learn shape representations and patterns with deep clustering

As mentioned above, a dimension reduction approach is taken to learn about the low dimensional latent feature space of the given sketch datasets. Among various deep generative models for reconstructing images, variational autoencoder (VAE) is one of the most widely used techniques thanks to its good performance of generalizing and learning a smooth latent representation of the input images.

Ha and Eck [43] proposed a sequence-to-sequence VAE for generating sketch drawings for completing a user's stroke-based drawing sequence of common objects. In this model, the stroke-based sketch drawings are captured as a recurrent neural network (RNN) that can carry out conditional and unconditional sketch generation. Partly due to its stroke-based modeling approach, however, it has a key limitation: low quality of learning latent representations of sketches from *multiple* categories. The limitation made it inadequate for CAVAS, as visual relationships between multiple categories need to be learned.

To overcome this limitation, Chen et al. [45] replaced the RNN layers with CNN layers so that they can deal with pixel-based sketches (i.e., images), making it possible to learn from multi-category sketches and generate a wide variety of sketches based on the user's input.

Since the CAVAS framework considers visual analogies from multiple categories, our model must learn from multi-category sketches. Following [43], the CAVAS deep learning-based sketch generative model, called the *Cavas-DL* model, is defined as follows.

4.1.1 Shape feature learning

Given n sketches $\boldsymbol{x} = \{x_i \epsilon X\}_{i=1}^{n}$, X is the data space (i.e., the space of all the sketches, represented as images), Cavas-DL encoder $\boldsymbol{q}_\phi(\cdot)$ compresses $\boldsymbol{x}$ into n latent vector $\boldsymbol{z} = \boldsymbol{q}_\phi(\boldsymbol{x}) = \{z_i \epsilon Z\}_{i=1}^{n}$. Z is the *latent space*. The dimensionality of Z is typically much smaller (*e.g.,*128) than X ($e.g.,$ 2304).

Cavas-DL decoder $\boldsymbol{p}_\theta(\cdot)$ samples n sketches conditional on $\boldsymbol{x}' = \boldsymbol{p}_\theta(\boldsymbol{z}) = \{\boldsymbol{x_i}' \epsilon X\}_{i=1}^{n}$ given latent vector $\boldsymbol{z}$. The loss function of the model can be defined as:

$$L_r = E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}' | \boldsymbol{z})] \quad (1)$$

where ϕ and θ are the parameters to be trained in the encoder and decoder, respectively. The parameters are typically the weights and biases of the neural networks. $E_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}(\cdot)$ is the *reconstruction loss* that ensures the close resemblance between the generated sketches and the original sketches.

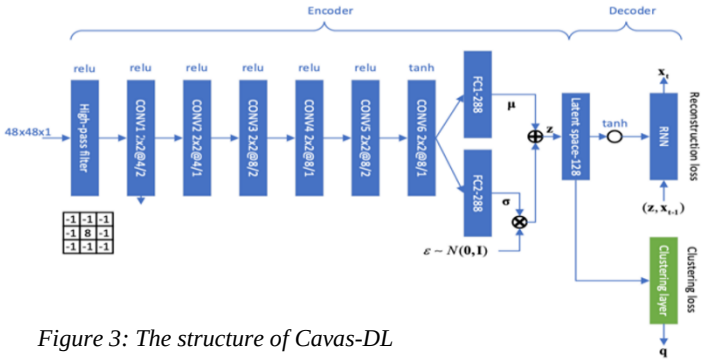

Figure 3: The structure of Cavas-DL

As shown in Figure 3, the Cavas-DL encoder $\boldsymbol{q}_\phi(\cdot)$ is implemented as a deep CNN that maps the black-and-white images in a space of 48x48 = 2304 dimensions into vectors in a latent space Z of 128-dimension.

4.1.2 Embedded clustering

To identify short- and long-distance analogies, sketches *sharing more shape features* should be grouped and separated from other groups. In ordinary situations, clustering of data points starts when the dimensional space of the data points is given and depends only on the settings of distance measures and clustering objectives. In Cavas-DL, however, clustering of sketches happens in the latent space Z that is being learned by training. The issue is how to devise a clustering process that not only perform the clustering in Z but also help the training of learning about Z hence the parameters of $\boldsymbol{q}_\phi(\cdot)$ and $\boldsymbol{p}_\theta(\cdot)$.

Xie et al. [46] proposed a deep embedded clustering (DEC) method to provide a way to simultaneously learn feature representations and clustering assignments using deep neural networks. The key idea of DEC is to iteratively refine clusters with an auxiliary target distribution derived from the current soft cluster assignment between the data points and the cluster centroids. This process gradually improves the clustering as well as the feature representation.

The DEC method is adopted in Cavas-DL. As shown in Figure 3, the clustering layer clusters all vectors in the latent space Z by simultaneously learning a set of K cluster centers $\{\mu_j \epsilon Z\}_{j=1}^{K}$ and mapping each latent vector z_i into a soft label q_i by student's t-distribution [47]. $\boldsymbol{q_i} = [q_{i1}, \dots, q_{ij}, \dots q_{ik}]$ is a *soft label* which quantifies the similarity between z_i and cluster center μ_j.

$$q_{ij} = \frac{\left(1 + \|z_i - \mu_j\|^2\right)^{-1}}{\sum_j \left(1 + \|z_i - \mu_j\|^2\right)^{-1}} \quad (2)$$

where q_{ij} is the jth entry of $\boldsymbol{q_i}$, representing the probability of z_i belonging to cluster j.

The clustering loss L_c is defined as a KL divergence between the distribution of soft labels Q measured by student's t-distribution and the *predefined target* distribution P derived from Q. The clustering loss is defined as

$$L_c = D_{KL}(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (3)$$

where the *target* distribution P is defined as

$$p_{ij} = \frac{q_{ij}^2/f_j}{\sum_j (q_{ij}^2/f_j)} \tag{4}$$

Raising q_{ij} to the second power and then dividing by the frequency per cluster, $f_j = \sum_i q_{ij}$, allows the target distribution P to improve cluster purity and put emphasis on confident labels. At the same time, this target distribution normalizes the contribution of each centroid on the clustering loss to prevent large clusters from distorting the latent space. This iterative strategy to minimize L_c works like self-training that labels the dataset to train on its *high confidence* predictions [48].

The total loss function of Cavas-DL, L_{rc}, is composed of two components: the reconstruction loss L_r in (1) and clustering loss L_c in (3). L_r is used to learn abstracted representations of the latent space in an unsupervised manner that can preserve shape features in sketch datasets. L_c is responsible for manipulating the latent space in order to cluster sketches based on shape similarity. The purpose of the loss function L_{rc} is to minimize reconstruction loss L_r and clustering loss L_r. A weighted sum method is used to optimize L_r and L_c, which is

$$L_{rc} = L_r + \tau L_c \tag{5}$$

where L_r is from (1) and L_c is from (3), and coefficient τ is set to be $0 \le \tau \le 1$.

4.1.3 Training

The shape feature mapping parameters ϕ and θ of Cavas-DL are pretrained by setting $\tau = 0$ to establish an initial latent space. After pretraining, the cluster centers are initialized by performing k-means on latent features of all sketches to get initial cluster centers $\{\mu_j \epsilon Z\}_{j=1}^{K}$. Based on (2) and (4), the initial distribution of soft labels Q and initial target distribution P can be obtained. After that, the deep clustering weights, cluster centroids, and target distribution P are updated as follows.

1) Update *weights* and *cluster centroids*. The gradients of L_c for each latent vector z_i and each cluster center u_j can be computed as:

$$\frac{\partial L_c}{\partial z_i} = 2\sum_{j=1}^{k}\left(1 + \|z_i - \mu_j\|^2\right)^{-1}(p_{ij} - q_{ij})(z_i - \mu_j) \tag{6}$$

$$\frac{\partial L_c}{\partial u_j} = 2\sum_{i=1}^{n}\left(1 + \|z_i - \mu_j\|^2\right)^{-1}(q_{ij} - p_{ij})(z_i - \mu_j) \tag{7}$$

Encoder and decoder parameter gradient $\frac{\partial L_r}{\partial \phi}$ and $\frac{\partial L_r}{\partial \theta}$ can be calculated by backpropagation when passing $\frac{\partial L_c}{\partial z_i}$ to the structure of the Cavas-DL model. The *parameters* of encoder and decoder, ϕ and θ, and the *cluster center,* μ_j, can be simultaneously updated by mini-batch stochastic gradient descent.

2) Update *target distribution*. In every epoch of training, the target distribution P serves as ground truth soft labels. The clustering layer is trained by predicting the soft assignment Q and then matching it to the target distribution P. At the end of the epoch, based on (4), the target distribution P is updated depending on the predicted soft label Q and used for the next epoch. After each epoch, the cluster label c_i assigned to z_i is obtained by

$$c_i = \arg\max_j q_{ij} \tag{8}$$

where q_{ij} can be obtained from (2). The training will stop when the cluster label assignment change (in percentage) between two consecutive epochs is less than a threshold δ.

4.2 Analyze visual similarity to identify visual analogies

The output of the clustering layer is a probability distribution of each latent vector z_i into each soft clustering label j. A clustering space can be introduced by any l-dimensional vector $\boldsymbol{\rho} \in \mathbb{R}^l$ that represents a probability distribution of clustering. Therefore, $\boldsymbol{\rho} = [p(\mathrm{c}_1|\rho), \dots, p(\mathrm{c}_k|\rho), \dots, p(\mathrm{c}_l|\rho)]$, $c_k(1 \le k \le l)$ represents the k-th cluster with $p(\mathrm{c}_k|\rho)$ indicating the probability of data ρ belong to k-th cluster.

In Cavas-DL, the inputs are sketches belonging to different categories, $\boldsymbol{x} = [x_{11}, \dots, x_{ij}, \dots, x_{st}]$, where x_{ij} means the j-th sketch belonging to i-th category, s is the number of categories and t the total number of sketches. In the latent space, latent vectors are $\boldsymbol{z} = [z_{11}, \dots, z_{ij}, \dots, z_{st}]$. In the clustering space, the probability distributions of latent vectors can be represented by a super matrix $\mathbb{Q}$, $\mathbb{Q} = [\boldsymbol{Q}_1, \boldsymbol{Q}_2, \dots, \boldsymbol{Q}_s]$. For matrix $\boldsymbol{Q}_i(1 \le i \le s)$, it includes n sketches. $\boldsymbol{Q}_i = [\boldsymbol{q}_{i1}, \dots, \boldsymbol{q}_{ij}, \dots, \boldsymbol{q}_{in}]$, $\boldsymbol{q}_{ij}(1 \le j \le n, n * s = t)$ represents a latent vector z_{ij} in the clustering space, i.e. $\boldsymbol{q}_{ij} = [p(\mathrm{c}_1|z_{ij}), \dots, p(\mathrm{c}_k|z_{ij}), \dots, p(\mathrm{c}_l|z_{ij})]$, where $P(\mathrm{c}_k|z_{ij})$ means the probability of z_{ij} belonging to the cluster c_k, $\sum_1^l P(\mathrm{c}_k|z_{ij}) = 1$. Soft clustering produces multi-clustering predictions for x_{ij}, while the ground truth category of x_{ij} is single labeled.

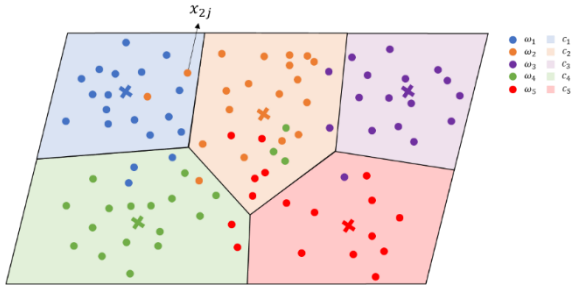

Figure 4: Sketches from five categories in a clustered 5D space

Figure 4 illustrates a clustered 5-dimensional space. Circle "o" indicates an input sketch, and cross "×" represents a centroid. The sketches of different categories are rendered with different colors. Solid lines indicate decision boundaries which are perpendicular bisectors of adjacent cluster centers, and the clusters are also rendered with different colors. As an example, it is assumed that all sketches come from five categories, $\omega_1, \omega_2, \omega_3, \omega_4$ and ω_5. Given the ground truth category of x_{2j} is ω_2, the probability distribution of corresponding latent

vector z_{ij} can be $\boldsymbol{q_{2j}} = [0.32, 0.21, 0.12, 0.19, 0.16]$. The cluster prediction of z_{ij} is c_1 which has a maximum probability of 0.32. However, sketches are clustered based on the shape similarity. Sketches from different categories can be clustered in the same group. Hence, the concept of the *sketch category*, which often indicates what a sketch is in the real world, is different from that of the *sketch group* (or *shape cluster*), which clusters sketches based on their shape similarities.

4.2.1 Sketch category and sketch group

We assume the number of clusters equals the number of sketch categories, and each cluster can represent one *shape pattern* which is composed of several *shape features*. In Figure 4, there are sketches from categories $\omega_1, \omega_2, \omega_3, \omega_4$ and ω_5, and there are five clusters c_1, c_2, c_3, c_4 and c_5. Each cluster has sketches from several categories, e.g., cluster c_2 contains sketches from the categories $\omega_2, \omega_3, \omega_4$ and ω_5. It means that each cluster presents a *shape pattern* that is obtained from learning *shape features* from multiple categories. In other words, different categories can share one common shape pattern.

Sketches of the same category can be clustered to different groups, e.g., some sketches in category ω_5 are clustered into the clusters c_2, c_4 and c_5. It means this category contains various shape features which are leaned by the Cavas-DL to form different clusters, i.e., shape patterns. For a given category, the cluster label of each sketch is determined by (8), and the number of sketches in each cluster can be counted. The probability of category i belonging to cluster k is o_{ik}, which indicates the ratio of how many sketches in category *i* belong to cluster *k* and can be computed in (9). The cluster probability distribution of each category is represented by $O_i = [o_{i1}, \dots, o_{ik}, \dots, o_{il}]$.

$$o_{ik} = \frac{n_{ik}}{N_i} \tag{9}$$

where n_{ik} is the number of sketches in category i which are located in cluster k, N_i is the total number of sketches in category i. l is the total number of clusters.

4.2.2 Similarity metrics

In this paper, the first visual similarity metric is a *distance-based similarity* that measures visual similarity based on the Euclidean distance between the category centroids in the latent space. The centroid of a category can be obtained by averaging all the latent vectors from the same category. A category centroid is different from a cluster centroid, which is the centroid of all sketches (maybe from different categories) are clustered in the same group. The distance-based similarity between category i and other categories can be computed as follow.

$$S_{ij}^{e} = 1 - \frac{E_{ij}}{\max_j E_{ij}} \tag{10}$$

where E_{ij} is the Euclidean distance between the centroids of category i and j, $\max_j E_{ij}$ is the longest Euclidean distance from the centroid of category i to centroids of other categories.

The second metric is an *overlap-based similarity* that measures visual similarity based on the amount of shape feature overlap between sketch categories. Shape feature overlap is defined as the amount of overlap between two cluster probability distributions. If two categories share more shape features, their sketches are more likely clustered into the same groups; their probability distributions are closer and have more overlapping regions. Hellinger distance is applied to measure the similarity of two cluster probability distributions, which is defined as:

$$H(O_i, O_j) = \sqrt{1 - \sum_{k=1}^{l} \sqrt{o_{ik} o_{jk}}} \tag{11}$$

where $\sum_{k=1}^{l} \sqrt{o_{ik} o_{jk}}$ is a measure of the area intersected by two cluster probability distributions.

The overlap-based similarity of other categories to category i can be defined as:

$$S_{ij}^{o} = 1 - \frac{H(O_i, O_j)}{\max_j H(O_i, O_j)} \tag{12}$$

where $\max_j H(O_i, O_j)$ is the longest Hellinger distance from category i to other categories.

4.2.3 Short- and long-distance visual analogies

The categories having high visual similarity are likely to be clustered in the same group. Sketch categories in the *same group* are considered visually *short-distanced*. The value of the similarity threshold determines how "short" the distance must be for two categories to be considered short-distanced. Given a designer is working on sketching in category *a,* and categories *a* and *b* are short-distanced, the Cavas-DL may provide a sketch of category *b* as a visual cue to stimulate the designer's visual analogy thinking. In this case, the designer's visual analogies are likely to be *short-distance* ones. On the other hand, if categories *a* and *b* belong to *different groups*, then the analogies are likely to be *long-distanced* ones.

Identifying long-distance visual cues requires relating sketch categories belonging to different groups, which can be time-consuming when the number of sketches and the number of categories are both large. To deal with this issue, a concept of *bridge category* is introduced. If a bridge category exists between two groups, the visual relationships between the categories in these groups can be established.

In Figure 5, the solid dots are categories clustered into two groups. The similarity of category *a* to category *b* can be represented by the similarity value S_{ab}^{o} or S_{ab}^{e}. That of category *b* to category *a* can be represented by the similarity value S_{ba}^{o} or S_{ba}^{e}. If S_{ab}^{o}, S_{ab}^{e}, S_{ba}^{o} and S_{ba}^{e} are all equal to or greater than a threshold ε, category *a* and category *b* can be classified in the same group and become *short-distance* analogies.

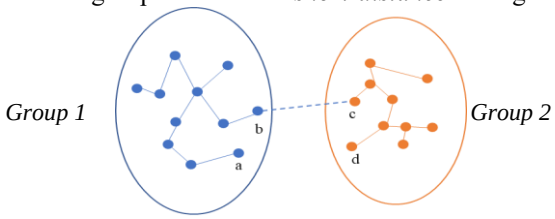

Figure 5: Visual relationships between two groups of categories

For category b from group 1, category c is the closest category in group 2, and for category c, category b is the closet category in group 1. The similarity of category b to c can be represented by the similarity values S_{bc}^{o} and S_{bc}^{e}. Category b is defined as a *bridge category,* if and only if S_{bc}^{o} or S_{bc}^{e} is equal to or greater than a threshold φ. In this case, there exists a visual relationship between categories b and c. As categories in group 1 are visually similar to category b and category b is visually similar to category c, other categories in group 1 can be visually similar to category c and then potentially visually similar to other categories, say category d, in group 2. If a bridge category is identified, it is possible to transfer shapes of categories between these groups based on visual similarities. The process to find a valid *long-distance* visual analogy follows:

$$Given\ a, b \in S\ \&\ c, d \in T;\ if\ b \sim c, then\ a \approx d$$

where S is a source domain of categories and T is a target domain of categories; $b \sim c$ means a visual relationship is built between categories b and c; $a \approx d$ means a possible long-distance visual relationship between categories a and d.

5. EXPERIMENTS

5.1 Datasets and implementation

The Cavas-DL model is evaluated based on the image datasets from Quickdraw, the largest sketch database built by Google [44]. It contains 345 categories of everyday objects. For the reason of computing time, sketches from 10 categories are chosen to test our proposed methods. The raw sequences from Quickdraw datasets are converted to monochrome png files of size 48x48, used as the input data for our deep neural network. These png files are images with binary pixels: strokes having value 1 and the rest value 0. Three datasets from ten categories are used for the experiments:

Dataset 1: Includes five categories: *van*, *bus*, *truck*, *pickup truck*, and *car*. All belong to automobiles and share some obvious shape features such as wheels and windows.

Dataset 2: Includes five categories: *speedboat*, *canoe*, *drill*, *pickup truck,* and *car*. Speedboat and canoe belong to boats and share some shape features such as V-shaped hulls. *Pickup truck* and *car* belong to automobiles. *Drill* doesn't share superficial shape similarity with other categories.

Dataset 3: Includes five categories: *television*, *canoe*, *drill*, *umbrella*, and *car*. Each of them doesn't share any superficial shape similarities with other categories.

Some examples of each dataset are listed in Table 1. The 15K sketches for each category are chosen. The sketches are divided into *training*, *validation,* and *testing* sets with sizes of 10K, 2.5K, and 2.5K, respectively.

For quantitatively verifying and demonstrating the improved performance of Cavas-DL, a comparison study between Cava-DL and the work of *sketch-pix2seq* proposed by Chen et al. [45] and its predecessor *sketch-rnn* by Ha and Eck [43] was conducted. For the sake of completeness, one of the traditional clustering algorithms, *k-means* is also included in the comparison. We show qualitative and quantitative results that demonstrate the benefit of Cavas-DL over other methods.

Table 1: Examples of each dataset

Dataset	Examples				
1	 van	 bus	 truck	 pickup truck	 car
2	 speedboat	 canoe	 drill	 pickup truck	 car
3	 television	 canoe	 drill	 umbrella	 car

The experiments on the four methods, namely, *Cavas-DL*, *sketch-pix2seq+k-mean, sketch-rnn+k-mean* and *k-mean*, are conducted using the three datasets described above. The parameters used for training *sketch-rnn* and *sketch-pix2seq* models are the same as the illustration in the papers [43, 45]. Cavas-DL is initialized by pretraining with $\tau = 0$, i.e., with the deep clustering detached. Then, the coefficient τ of clustering loss in (5) is set to 0.05, which is determined by a grid search in a list [0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0] and batch size to 100 for all datasets. The maximum number of epochs is set to $T = 50$. In each iteration, we train the encoder for one epoch using Adam optimizer with learning rate $\lambda = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$. The convergence threshold δ is set to 0.1%. The dimension of the latent space in these three models is 128, which is the same in the papers [43, 45]. K-means is performed to cluster sketches in the latent space of sketch-pix2seq and sketch-rnn. Besides, as a baseline for comparison, k-means also runs on the sketch datasets with the original dimensions of 48 x 48 = 2304, which is much larger than the latent space. k-means performs 20 times with different initialization and the result with the best objective value is chosen, where $k = 5$.

We evaluate all four clustering methods with unsupervised clustering accuracy (ACC). The ACC is defined as the best match between ground truth $\boldsymbol{y}$ and predicted cluster labels $\boldsymbol{c}$:

$$ACC(\boldsymbol{y}, \boldsymbol{c}) = \max_{m \in \mathcal{M}} \frac{\sum_{i=1}^{n} \mathbf{1}\{y_i = m(c_i)\}}{n} \tag{13}$$

where n is the total number of samples, y_i is the ground truth label, c_i is the predicted cluster label of the example x_i obtained by the model, and $\mathcal{M}$ is the set of all possible one-to-one mappings between predicted cluster labels to ground truth cluster. The best cluster assignment can be efficiently computed by the Hungarian algorithm [49].

5.2 Shape feature learning and clustering performance

As described in Section 3.1, to provide adequate visual cues to stimulate the designer's analogical thinking, the CAVAS system should learn from the given datasets the shape features and distinguish the shape patterns that go beyond the sketch categories. From feature learning and clustering perspectives, the major distinction of our Cavas-DL method is combining deep feature learning with deep clustering. Thanks to the dynamic property of Cavas-DL that simultaneously adjusts the processes

of feature learning and clustering, its improved performance in shape pattern identification is expected, and in fact, it has been reported that our proposed algorithm outperforms others significantly for both clustering and category information preservation [50].

In order to visualize the latent space of unsupervised modes and one supervised model on the three datasets, t-SNE [47] is used to reduce the dimensionality of Z from 128 to 2, and 7500 testing sketches are plotted from five categories of the three datasets for each method. The dimensionality reduction from 128 to 2 may cause significant information loss and generate misleading visualizations. t-SNE has a hyper-parameter, called perplexity, that balances the attention t-SNE gives to local and global aspects of the data and can affect the resulting plot. Its value is recommended to be between 5 and 50. If choosing different values between 5 and 50 significantly changes the interpretation of the data, then t-SNE is not the best choice to visualize or validate our hypothesis. To increase the robustness of our findings and reflect how multiple runs affect the outcome of t-SNE, we put forward the process to validate the visualization of a trained latent space, as shown in Figure 6.

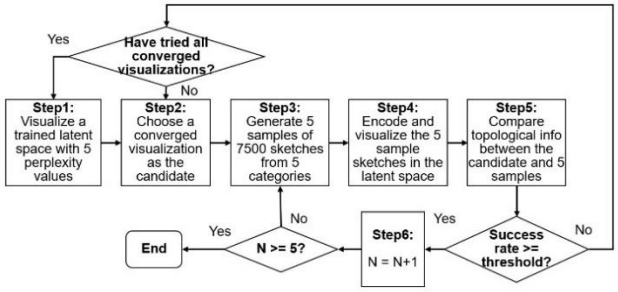

Figure 6: Process to validate visualization of a trained latent space

In **Step 1**, we set the initial value of the counter N as 0, which is used to record the times of sample generation. Then, we take advantage of t-SNE for visualizing a latent space with a list of perplexity values. In **Step 2**, we choose a converged visualization as the candidate. For example, in Figure 7, the latent space is visualized under different perplexity value settings. We can see the latent space visualizations in a list [30,40,50] are converged. There are two types of global geometry of the converged visualizations. One type can represent visualizations with perplexity values of 20, 30, 40. Another type can represent the visualization with a perplexity value of 50. We randomly choose one (perplexity value=30) as the candidate from the first type. If this type cannot be a valid visualization, we will try the other types. In **Step 3,** we randomly generate five samples of 7,500 sketches from five sketch categories in the QuickDraw dataset. In **Step 4**, the five sample sketches are encoded and visualized in the latent space. In **Step 5**, we compare the topological information of five samples in the latent space with the candidate. If over half of the visualizations of the five samples are similar to the candidate, it means this round of sample generation can validate that the candidate can represent these five sample sketches in the latent space. A success rate is used to illustrate how many samples are similar to the candidate. For example, in Figure 8, only sample_4 is different from the candidate, so the success rate is 0.8. A threshold is set to 0.6. If the success rate is no less than a threshold, then the next round of five sample generations will be started. The counter N will be increased by 1. Otherwise, we will go back to the second step to choose the other type of converged visualization as the candidate. If all the converged visualizations have been tried, then we go back to the first step. There will be five rounds of sample generation. If all of them can be successful, then the candidate will be chosen to visualize the trained latent space.

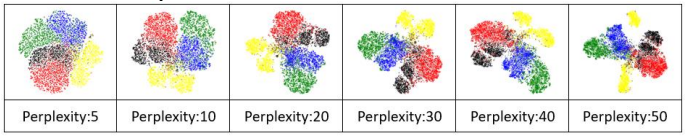

Figure 7: Visualizing a latent space with different perplexity values

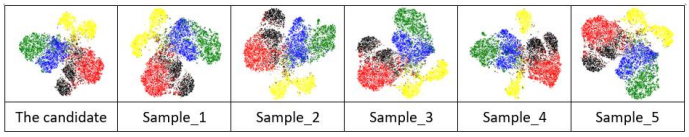

Figure 8: Visualization of five sample sketches in the latent space

After validating visualizations of the latent space of each model in three datasets, we can compare shape feature learning and clustering performance of different models. Firstly, we compare the learning and performance of unsupervised models. In Figure 9, the Cavas-DL performs the best in clustering since the sketches from different categories are more separated, and the sketches from the same category are denser together in all cases. For Dataset1, all sketches are from the same taxonomic category hence are hard to be separated into different clusters. The red, black, and green clusters are denser in Cavas-DL than the other two as the clustering loss L_c can force sketches from the same taxonomy to be gathered and push away sketches from different taxonomies. For Dataset2, sketches are from three taxonomic categories. Sketch categories belonging to the same taxonomy should be close to each other as they share more shape features and away from other taxonomies. This assumption can be confirmed by our model as well as sketch-pix2seq, as they both use CNN as an encoder that can discover and represent shape structures in the latent space. Car(red) is close to pickup truck(black), and speedboat(blue) is close to canoe(green) in the first Cavas-DL plot, while this cannot be easily detected in the third sketch-rnn plot; For Dataset3, all sketches are from different taxonomic categories. All three deep learning models can cluster each category. However, the clusters in the Cavas-DL plot are denser and have a larger margin with each other.

In Figure 9, we also compare three unsupervised models mentioned above with a supervised model, which is a convolutional neural network (CNN) from official guides of QuickDraw[1,2]. For every dataset, the supervised model can more

[1] https://github.com/googlecreativelab/quickdraw-dataset

[2] https://github.com/zaidalyafeai/zaidalyafeai.github.io/tree/master/sketcher

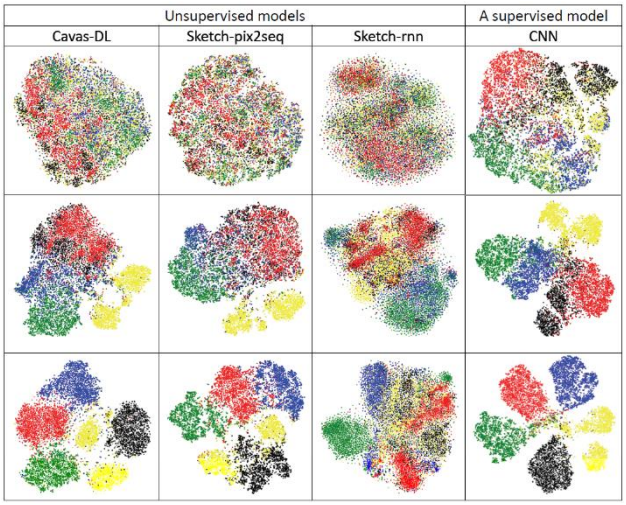

Figure 9: Clustered latent space of three datasets for each method (top row: Dataset1-van(blue), bus(green), truck(yellow), pickup truck(black), car(red); middle row: Dataset2-speedboat(blue), canoe(green), drill(yellow), pickup truck(black), car(red); bottom row: Dataset3-television(blue), canoe(green), drill(yellow), umbrella(black), car(red))

clearly separate each category in the latent space. The reason is the latent space of the supervised model is trained based on given category label information. Therefore, the supervised model can have better performance in categorizing sketches. From Dataset1 to Dataset3, the shape feature sharing become less and less, and the margins between sketch categories in the latent space of CNN become larger and larger. It infers that after training, shape features extracted by convolution layers are related to the given semantic information (category labels). When all sketches are from the same taxonomy, this relationship can hardly be built. When all sketches are from the different taxonomies, this relationship can be easily established. However, in this research, the goal is to learn a latent space that represents the shape patterns. Ideally, similar shapes from the same or different categories can be clustered in the same group, and different groups are distinguishable from each other. In other words, the purpose of the proposed Cavas-DL is to construct a relationship between shape features and shape patterns in a case that the shape pattern label of each sketch is hard or impossible to be collected and created. Therefore, even all sketch categories are from different taxonomies in Dataset3, Cavas-DL tries to keep relatively small margins to possibly build shape connections between these categories.

5.3 Performance of visual similarity analysis

After extracting shape features and discovering shape patterns from the given datasets, the CAVAS system should be able to analyze visual similarities between different sketch categories and identify relevant visual cues. In order to measure visual similarity, both *distance*- and *overlap*-based *similarities* are applied.

Euclidean distance. In Figure 10, the clustered latent space is presented to visually show Euclidean distances between centroids of 10 sketch categories. *Speedboat* and *canoe* in the green circle are from the same taxonomy, and *van*, *pickup truck*, *truck*, *car*, and *bus* in the red circle are also from the same taxonomy. *Pickup truck* and *speedboat* are close to each other; hence it is possible to build a visual relationship between two taxonomies through these two categories. *Drill*, *television*, and *umbrella* are from different taxonomies. Categories from the same taxonomy have shorter centroid distances and higher overlap magnitude; Categories from different taxonomies have longer centroid distances and lower overlap magnitude. After checking the dataset, one could see that there are two different kinds of drills in the dataset: *handheld drill* and *ground drill*. Therefore, *drills* are separated into two groups in the latent space. By exploring this latent space of ten categories, designers can have an overall view of the visual relationships between them.

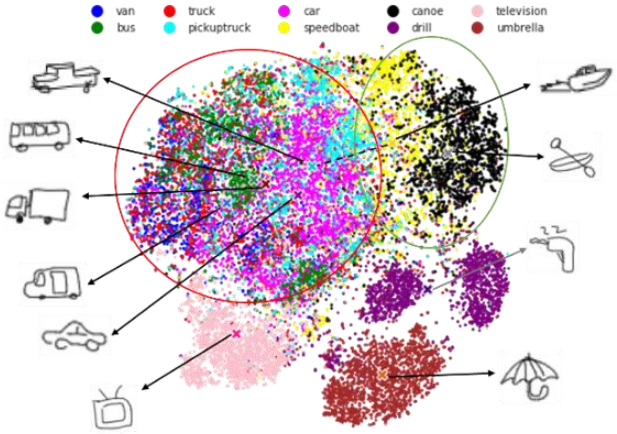

Figure 10: Sketches from ten categories in the latent space, cross "×" represents a category centroid

Hellinger distance. The cluster probability distributions in Figure 11 visually presents the amount of overlap between categories using Hellinger distance. In Figure 11, a cluster can accurately capture one shape pattern that represents shape features from the same or different taxonomies. For example, Cluster 1 captures most shape features from the automobile taxonomy; it also captures some shape features from *speedboat*. Cluster 5 captures most shape features from the boat taxonomy; it also captures some shape features from *pickup truck*. This is also why *speedboat* and *pickup truck* can be *bridge categories* to link boat and automobile taxonomies. Clusters 7 and 10 capture shape features from *umbrella* and *television*, respectively.

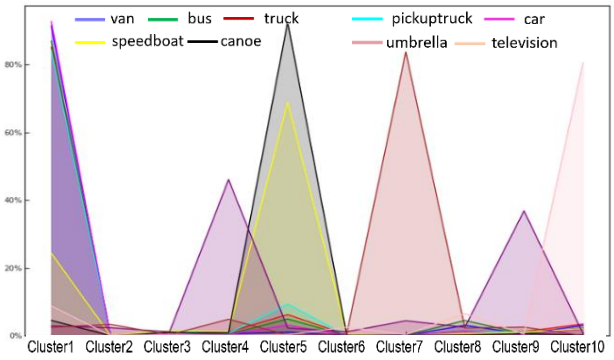

Figure 11: Cluster probability distributions of ten categories

Umbrella and *television* are distinct from other categories. One category may contain two or more shape patterns as it has variations. This case can be captured by different clusters. For example, Clusters 4 and 9 capture different shape patterns in *drill*, because there are two different types of drills in the dataset. Clusters 2, 3, 6, and 8 can barely capture some shape features from the ten categories. It was assumed that the clustering layer can learn and differentiate shape features from different categories, and one category can only represent one shape pattern. Therefore, the number of clusters is set equal to the number of categories during the experiment. However, the results showed that some clusters are not useful. Therefore, the optimal cluster number needs to be determined by iterating more experiments. We believe the *optimal cluster* number may relate to the number of taxonomies in the given dataset, as categories that belong to the same taxonomy would have a similar shape pattern.

Visual similarity. The distance- and overlap-based similarity matrices in Figure 12 and Figure 14 can quantify the visual similarity between each category based on Euclidean distance and Hellinger distance, respectively. As all distances from other categories to a given category are normalized by the maximum distance, these two matrices are asymmetric. The rows in these matrices are rearranged based on hierarchical clustering and accompanied by dendrograms describing the hierarchical cluster structure. The values in each cell represent the *similarity magnitude* of the row category to each column category. A larger value means higher similarity. The threshold ε is set to 0.5; if similarity values between several categories are all equal to or greater than 0.5, then these categories form a group. Categories in the same group are *short-distance* visual analogies; those in the different groups are *long-distance* visual analogies. The threshold φ is set to 0.5; a category is considered as a *bridge category* if the largest similarity value between this category with one category in another group is equal to or greater than 0.5.

In Figure 12, as threshold ε is set to 0.5, ten categories can form four groups based on distance-based similarity, which is shown in the dendrogram. *Van, bus, truck, pickup truck* and *car* are in the red group. *Bus* and *truck* have the highest similarity values. It implies they are tightly closed to each other in the latent space. The green group includes *speedboat* and *canoe*. The orange group contains *drill* and *umbrella*, which are from different taxonomies. The gray group contains *television*. The red group is entwined with the green group. It means shape transformation can happen between automobiles and boats as they share many shape features. The similarity value of *pickup truck* to *speedboat* is 0.6 and the value of *speedboat* to *pickup truck* is 0.67, which are above the threshold φ. They are bridge categories with a strong capability to connect two taxonomies. It means for making visual analogy, if the target domain is *boat*, a boat designer can try to make a visual connection with a source domain which is automobile through *speedboat*, vice versa.

As shown in Figure 13, *van, bus, truck, pickup truck,* and *car* are different categories in the *automobile* group. It is more effective to build visual connections between them, but fewer changes to obtain visual inspirations. More efforts need to be made to construct a visual relationship between different categories in different groups (e.g., van and canoe). However, novelty is more likely to happen if the long-distance visual connection can be built [2, 10]. Bridge categories (*pickup truck* and *speedboat*) are valuable spots to draw a visual analogy for

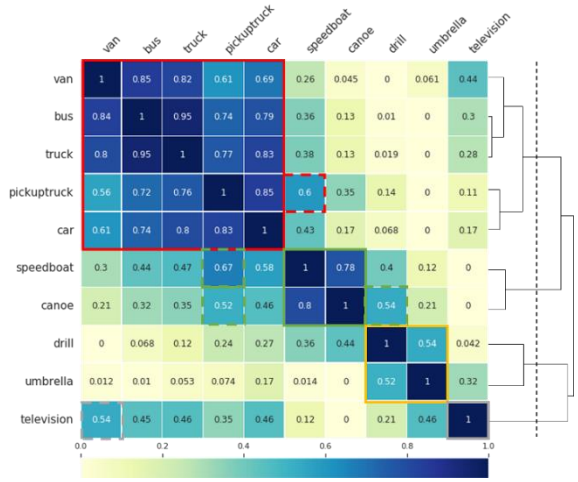

Figure 12: Distance-based similarity matrix with dendrograms (different groups are marked with solid squares with different colors; Some cells' values larger than threshold φ are marked with dashed squares to

both effectiveness and novelty at the same time. *Canoe* is the only category which can connect one group with the other two groups as the similarity values of canoe to *pickup truck* and *drill* are 0.52 and 0.54, respectively. It means it can lead visual connections to different directions. The similarity value of television to *van* in the red group is 0.54 which is above the threshold φ. It means *television* has a potential to make a visual relationship with *automobile*. It is easy to understand as a screen of a *television* is visually similar to a window of a van.

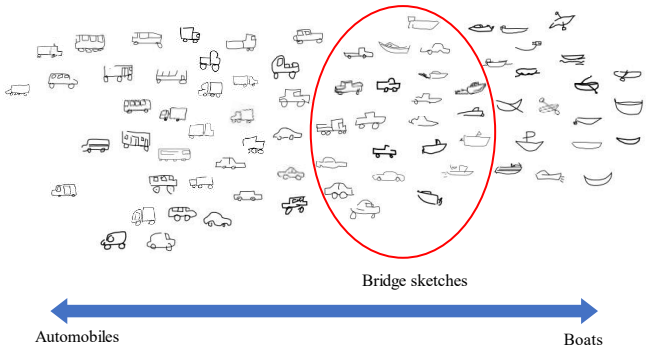

Figure 13: A possible visual analogy making via bridge categories

In Figure 14, as threshold ε is set to 0.5, ten categories form five groups based on overlap-based similarity, shown in the dendrogram. The categories in red and green groups are still the same. However, *drill* and *umbrella* are not classified in the same group. Basically, the similarity values between categories from the same taxonomy become larger, and the similarity values between categories for different taxonomy become smaller. For example, the lowest similarity value in the red group is increased

by 0.25. *Pickup truck* and *truck* have the highest overlap-based similarity. It makes more sense comparing with distance-based similarity, from which *truck* is more visually similar to *bus*. No bridge categories can be detected. Therefore, the visual relationships between categories in the same group become stronger. However, each group is distinct from other groups. It may be hard to build visual relationships between these groups based on overlap-based similarity. In other words, short-distance visual analogies can be easily identified, but long-distance visual analogies might be harder to be found.

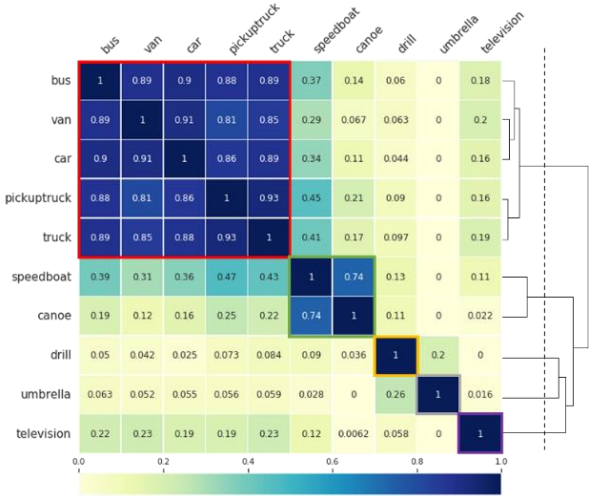

Figure 14: An overlap-based similarity matrix with dendrograms (different groups are marked with solid squares with different colors)

6. DISCUSSION

By visualizing the latent space of 3 different datasets with different levels of common shape feature sharing, we empirically validate two points: 1) If sketches are from the same taxonomy, they share many shape features. It is difficult for deep clustering models to separate them. The sketches in Dataset1 are from the same taxonomy; three models are struggling to cluster sketches. But our proposed model can somehow separate red and black points from others. The sketches in Dataset3 are from different taxonomies; it is easier for the three models to cluster sketches. Our proposed model can separate clusters with larger margin. 2) If a deep clustering model uses CNN layers to encode input sketches and simultaneously considers the clustering loss, its clustering performance can be improved, indicating the advantage of embedded clustering. Some of the sketches in Dataset2 come from the same taxonomy; our model can cluster points denser than the other two models. 3) Among three unsupervised models, Cavas-DL is the most similar to CNN regarding the sketch distributions in the latent space. It also suggests that Cavas-DL is better at differentiating sketches based on shape patterns and also retaining shape relationships between sketches in the same taxonomy.

After effectively encoding the source of analogies, potential targets need to be identified. During the visual analogy search process, designers qualitatively assess the similarity between visual materials. The moment to identify a bridge to connect or transfer one shape to another is often random and unpredictable. In order to quantify visual similarity, distance- and overlap-based similarity are introduced to analyze the visual relationships between categories and find useful analogies. Bridge categories are defined to guide the connection building of different shapes.

From the experiment of visual similarity analysis, one can see: 1) the distance- and overlap-based similarity metrics can confirm that categories from the same taxonomy share more shape features and have higher visual similarity than categories from different taxonomy; 2) distance-based similarity is less accurate than overlap-based similarity when finding visual relationships between categories from the same taxonomy as these categories share too many shape features, in these cases, the overlap-based similarity is more effective than distance-based similarity; 3) overlap-based similarity can make categories from different taxonomies more distinguishable, e.g., the visual similarity values between categories in the automobile taxonomy become larger. However, finding bridge categories become more difficult, e.g., the similarity values between speedboat with other categories in the automobile taxonomy become smaller, and it is not detected as a bridge category; 4) bridge categories can be useful to find the path to visually transform shapes from one taxonomy to another taxonomy. The path can potentially explain how to find long-distance visual analogies. For example, *pickup truck* is classified as a bridge category. A car designer can apply visual thinking to transfer the shape of a car to a pickup truck and then to a speedboat and retrieve some inspiring cues from speedboat design.

Both distance- and overlap-based similarities are useful when analyzing visual relationships between various categories in different scenarios. However, these two should work together to provide more convincing results. Being visually similar makes analogical inferences easy, and being categorically different makes the potential analogy across categories novel. One important finding is the detection of bridge categories allows both effectiveness and novelty to be obtained at the same time and may resolve the “analogical distance” dilemma as suggested from prior studies [31]: near-field stimuli are more effective, while far-field stimuli offer novelty. A bridge category is an analogy located in a “sweet spot” proposed by Fu et al. [31], which can offer a strategy to avoid visual fixation and find visual stimuli from long-distance analogies.

From a designer’s designing point of view, the visual presentation of the latent space shown in Figures 9 to 14 can be highly effective for the designer to choose potentially inspiring visual cues either systematically or randomly. The example, upon viewing the 2D distributions of sketches like Figures 9 and 10, a designer may intentionally choose a dataset with categories clearly from diverse taxonomies, or he/she may select the one that holds closely related sketches. Making a targeted selection, i.e., clicking a colored dot on the chart on the sketch map allows the designer to knowingly expand his/her thinking toward potentially fruitful directions. Besides, visual assistance like Figure 11 provides designers with a tool to explore the overlap-

similarity space that has the potential to offer multilayer expansions of thinking for the designer. Furthermore, the grouping matrix displays like Figures 13 and 14 allow designers to quickly access closely related groups of sketches which may impact designers' analogy making differently compared to single visual cue-based stimulation. Future human subject-based studies are needed to verify the effectiveness of these human augmentation strategies.

7. CONCLUSIONS

In this paper, a computer-aided visual analogy support (CAVAS) framework is proposed, and a deep learning based computational model Cavas-DL is introduced as a human design augmentation tool to assist human visual analogy making. The CAVAS framework extends the GSP creative stimulation model into the human-computer interaction context, and the Cavas-DL model has demonstrated the potential of sketch-and-image based visual analogy support. Through the model building and the experiment results, the following conclusions are drawn.

- A computer-aided visual analogy support framework CAVAS is introduced, and its key functional components and processes are identified and demonstrated for augmenting designers' visual analogical thinking processes.
- An unsupervised deep learning model Cavas-DL combines a CNN based shape feature extraction algorithm with a deep embedded clustering model and achieves the best feature capturing and clustering simultaneously.
- The visualization of the latent space of sketches can guide and assist designers' visual thinking, which has the potential to promote visual analogy making in conceptual design and boost idea generation.
- The distance- and overlap-based similarities introduced can be applied to identify short- and long-distance analogies based on visual similarity. The detection of bridge categories provides a way to find long-distance analogies for visual analogy-making processes.
- The extensive experiments conducted demonstrate the effectiveness and robustness of our computational tool, a major step toward computer aided visual analogy support.

A drawback of the Cavas-DL model is the need to balance the weight ratio of reconstruction loss and clustering loss. It means one needs to determine the weight of clustering loss in equation (5). Searching for an appropriate value for the weight can take time since the model needs to be trained many times. Besides, the threshold ε to determine short- and long-distance analogies and the threshold φ to determine bridge categories are set based on the distance-based and overlap-based similarity matrices and domain knowledge. Our ongoing work includes the investigation on how to determine the optimal cluster number for the clustering layer and how to use the learned sematic or functional meaning behind shapes to support visual analogy. One outstanding issue is to evaluate the effectiveness of the visual cues generated by Cavas-DL in stimulating designers' visual analogy making for generating more and better ideas. The tool Cavas-DL from this research has made it possible for us to conduct human subject-based design experiments to evaluate the effectiveness of computational support for visual analogy making in design.

REFERENCES

[1] Jin, Y., and Benami, O., 2010, "Creative patterns and stimulation in conceptual design," AI EDAM, 24(2), pp.191-209.

[2] Goldschmidt, G., 2001, "Visual analogy-a strategy for design reasoning and learning," Design knowing and learning: Cognition in design education, Elsevier, pp. 199-219.

[3] Mougenot, C., Bouchard, C., Aoussat, A., and Westerman, S., 2008, "Inspiration, images and design: an investigation of designers' information gathering strategies," Journal of Design Research, 7(4), pp. 331-351.

[4] Bouchard, C., Omhover, J.-f., Mougenot, C., Aoussat, A., and Westerman, S. J., 2008, "TRENDS: a content-based information retrieval system for designers," Design Computing and Cognition'08, Springer, pp. 593-611.

[5] Chakrabarti, A., Siddharth, L., Dinakar, M., Panda, M., Palegar, N., and Keshwani, S., "Idea Inspire 3.0—A tool for analogical design," Proc. International Conference on Research into Design, Springer, pp. 475-485.

[6] Han, J., Shi, F., Chen, L., and Childs, P. R., 2018, "A computational tool for creative idea generation based on analogical reasoning and ontology," AI EDAM, 32(4), pp. 462-477.

[7] Ullman, D. G., Wood, S., and Craig, D., 1990, "The importance of drawing in the mechanical design process," Computers & Graphics, 14(2), pp. 263-274.

[8] Yang, M. C., 2009, "Observations on concept generation and sketching in engineering design," Research in Engineering Design, 20(1), pp. 1-11.

[9] Kokotovich, V., and Purcell, T., 2000, "Mental synthesis and creativity in design: an experimental examination," Design Studies, 21(5), pp. 437-449.

[10] 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.

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

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

[13] Marshall, K. S., Crawford, R., and Jensen, D., "Analogy seeded mind-maps: A comparison of verbal and pictorial representation of analogies in the concept generation process," ASME DETC2016-60100.

[14] Malaga, R. A., 2000, "The effect of stimulus modes and associative distance in individual creativity support systems," Decision Support Systems, 29(2), pp. 125-141.

[15] Gero, J., and Yan, M., 1994, "Shape emergence by symbolic reasoning," Environment and Planning B: Planning and Design, 21(2), pp. 191-212.

[16] Oxman, R., 2002, "The thinking eye: visual re-cognition in design emergence," Design Studies, 23(2), pp. 135-164.

[17] Arnheim, R., 1997, Visual thinking, Univ of California Press.
[18] Chen, W., Fuge, M., and Chazan, J., 2017, "Design Manifolds Capture the Intrinsic Complexity & Dimension of Design Spaces," J. of Mechanical Design, 139(5), p.051102.
[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] Casakin, H., 2010, "Visual analogy, visual displays, and the nature of design problems: the effect of expertise," Environment & Planning B: Planning & Design, 37(1), pp.170-188.
[21] Linsey, J., Wood, K.L., & Markman, A.B., 2008, "Modality and representation in analogy" AI EDAM, 22(2), pp.85-100.
[22] Vincent, J.F., & Mann, D.L., 2002, "Systematic technology transfer from biology to engineering," Phil. Trans. of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 360(1791), pp. 159-173.
[23] Chakrabarti, A., Sarkar, P., Leelavathamma, B, Nataraju, B., 2005, "A functional representation for aiding biomimetic & artificial inspiration of new ideas" AIEDAM,19(2),113-132.
[24] Cheong, H, Chiu, I., Shu, L, Stone, R, McAdams, D.A, 2011 "Biologically meaningful keywords for functional terms of the functional basis" J. Mechanical Design, 133(2), 021007.
[25] Vattam, S., Wiltgen, B, Helms, M., Goel, A, & Yen, J., 2011, "DANE: fostering creativity in and through biologically inspired design" Design Creativity 2010, Springer, 115-122.
[26] Nagel, J.K, & Stone, R.B, 2012, "A computational approach to biologically inspired design" AI EDAM, 26(2), p161-176.
[27] Deldin, J.-M., and Schuknecht, M., 2014, "The AskNature database: enabling solutions in biomimetic design," Biologically inspired design, Springer, pp. 17-27.
[28] Murphy, J., Fu, K., Otto, K., Yang, M., Jensen, D., and Wood, K., 2014, "Function based design-by-analogy: a functional vector approach to analogical search," Journal of Mechanical Design, 136(10), p. 101102.
[29] Fu, K., Cagan, J., Kotovsky, K., and Wood, K., 2013, "Discovering structure in design databases through functional and surface-based mapping," J. of Mechanical Design, 135(3), p. 031006.
[30] Song, B., and Luo, J., 2017, "Mining patent precedents for data-driven design: the case of spherical rolling robots," Journal of Mechanical Design, 139(11), p. 111420.
[31] Fu, K., Chan, J., Cagan, J., Kotovsky, K., Schunn, C., and Wood, K., 2013, "The meaning of "near" and "far": the impact of structuring design databases and the effect of distance of analogy on design output," J. of Mechanical Design, 135(2), p. 021007.
[32] Luo, J., Sarica, S., and Wood, K. L., 2021, "Guiding data-driven design ideation by knowledge distance," Knowledge-Based Systems, 218, p. 106873.
[33] Sarica, S., Luo, J., and Wood, K. L., 2020, "TechNet: Technology semantic network based on patent data," Expert Systems with Applications, 142, p. 112995.
[34] He, Y, Camburn, B, Liu, H, Luo, J, Yang, M, Wood, K, 2019, "Mining and representing the concept space of existing ideas for directed ideation" J. of Mechanical Design, 141(12).
[35] Linsey, J S, Clauss, E, Kurtoglu, T, Murphy, J, Wood, K, Markman, A, 2011 "An experimental study of group idea generation techniques: understanding the roles of idea representation and viewing methods" J. of Mech Design, 133(3).
[36] Atilola, O., Tomko, M., and Linsey, J. S., 2016, "The effects of representation on idea generation and design fixation: A study comparing sketches and function trees," Design studies, 42, pp. 110-136.
[37] McKoy, F. L., Vargas-Hernández, N., Summers, J. D., and Shah, J. J., 2001, "Influence of design representation on effectiveness of idea generation," Proceedings of ASME DETC, Pittsburgh, PA, Sept, pp. 9-12.
[38] Casakin, H., 2004, "Visual analogy as a cognitive strategy in the design process: Expert versus novice performance," journal of Design Research, 4(2), p. 124.
[39] Stiny, G., "Emergence and continuity in shape grammars," Proc. CAAD futures, pp. 37-54.
[40] Bell, S., and Bala, K., 2015, "Learning visual similarity for product design with convolutional neural networks," ACM Transactions on Graphics (TOG), 34(4), p. 98.
[41] Karimi, P., Maher, M. L., Davis, N., Grace, K., 2019, "Deep Learning in a Computational Model for Conceptual Shifts in a Co-Creative Design System," arXiv:1906.10188.
[42] Jiang, S., Luo, J., Ruiz-Pava, G., Hu, J., Magee, C. L., 2021, "Deriving design feature vectors for patent images using convolutional neural networks," J. of Mech. Design, 143(6).
[43] Ha, D., and Eck, D., 2017, "A neural representation of sketch drawings," arXiv preprint arXiv:1704.03477.
[44] J. Jongejan, H. R., T. Kawashima, J. Kim, and N. Fox-Gieg, 2016, "The Quick, Draw! - A.I. Experiment."
[45] Chen, Y., Tu, S., Yi, Y., and Xu, L., 2017, "Sketch-pix2seq: a model to generate sketches of multiple categories," arXiv preprint arXiv:1709.04121.
[46] Xie, J., Girshick, R., and Farhadi, A., "Unsupervised deep embedding for clustering analysis," Proc. International conference on machine learning, pp. 478-487.
[47] Maaten, L, and Hinton, G., 2008, "Visualizing data using t-SNE," J. of machine learning research, 9(Nov), 2579-2605.
[48] Nigam, K., and Ghani, R., "Analyzing the effectiveness and applicability of co-training," Proc. of 9th Intl. Conference on Information & Knowledge Management, pp. 86-93.
[49] Kuhn, H, 1955, "The Hungarian method for the assignment problem" Naval Res. Logistics quarterly, 2(1-2), pp. 83-97.
[50] Zhang, Z., and Jin, Y., "An Unsupervised Deep Learning Model to Discover Visual Similarity Between Sketches for Visual Analogy Support," ASME IDETC2020-22394.