Egyptian Journal of Pure and Applied Science

A precursory insight into the behavior of CNNs to analyze visual similarities of sketches

N. Muhammad[1], S. Khamis[1], A. Abdelfattah[1,2]

[1]*Mathematics Department, Faculty of Science, Ain Shams University, Cairo, Egypt.*
[2]*Faculty of Computer Science and Engineering, King Salman International University.*

ARTICLE INFO

Received 21 September 2021
Accepted 29 November 2021

Keywords

Feature visualization,
CNN interpretation,
Sketch recognition,
Visual similarities,
Learning transfer.

Correspondence

A. Abdelfattah

E-mail

ahabdelfattah@sci.asu.edu.eg

ABSTRACT

Artificial Neural Networks (ANNs) have been showing a great performance in Artificial Intelligence (AI)-related tasks in many fields for the last couple of decades or so. Nevertheless, there is also still an apparent lack of any clear understanding of its learning behavior, which is triggering the research of ANNs interpretation. In this article, we put forward a new technique to investigate a specific learning behavior of Convolutional Neural Networks (CNNs) using the problem of free-hand sketch classification as an interesting testbed. Our aim is to look into possible factors that contribute to the classification process and transfer of learning between different categories. We designed and analyzed two advanced experiments to deduce possible learning attributes that could be shared by both CNNs and humans. Our preliminary results show a huge potential of CNNs to detect visual similarities among hand-drawn sketches of different object categories, which seems to resemble that of humans to some extent.

1. Introduction

Hand-drawn sketches are one of the basic communication tools throughout history. What started as simple lines on stones by Homo sapiens 73,000 years ago [1], is still considered to be one of the widely used means of communications.

Humans tend to represent their ideas or plans by sketching a simple drawing or a chart. Sketches are considered as a universal language too, as neither prior knowledge is needed, nor specific syntax and semantics rules are applied. Moreover, the spread of devices with touch-screen increased the Human-Computer Interaction (HCI) using hand or finger gestures. Many people prefer to doodle something or use a gesture than actually typing. This can be seen in the simplest tasks like using a pattern instead of a password to unlock a device.

Many applications either use sketching as a main feature or include it as an additional, yet effective, feature: "Guess the sketch" is a general term for a lot of online games that mimic "Pictionary"; and texting services such as iMessage include this feature.

In educational platforms, which experienced a huge increase and development due to the COVID-19 pandemic, online classes and meetings would not have been plausible without the presence of handwriting or free sketching tools. In these platforms, students are also required to submit assignments online, which can include sketches of different kinds (e.g., biological drawings, circuits, free-hand, etc.).

The presence of a robust system for sketch classification, completion, and generation is, thus, vital. This, and a language for sketch understanding, will be even more vital soon when devices that sense human gestures and posture in 3D prevail. However, image-related applications were (and still are) more dominant in research areas than sketch-related ones. An account for human imprecision that characterize visually similar sketches, as well as features that are specific to hand-free sketching, must be taken into consideration too.

Image-related applications like image recognition [2], face recognition [3], image captioning [4], segmentation [5], and many other applications have been a hot research area in Artificial Intelligence (AI) for a while. This kind of research is urged mainly by the widespread of visual data (e.g., images and videos) along with the digital transformation taking place in most life aspects. Nevertheless, sketch-related applications have not gained an equal attention yet, particularly the training of machines to detect and recognize visually similar sketches of the same object sketched by different people.

Images versus sketches Among the key factors that facilitate image-related applications are: (i) the availability of huge, labeled image datasets; (ii) the increase in computers' capabilities in terms of processing units and memory capacities; (iii) the advances in Machine Learning (ML) models, especially the use of CNNs that acted as a breakthrough in the field; and (iv) the informative nature of images. While, in case of sketch-related applications, the following are limiting and challenging factors: (i) the lack of large, diverse, and representative datasets; (ii) the abstract nature of sketches, including their usual lack of colors, texture, or background information; (iii) the impossibility to have a unified form of a sketched object, as sketches are highly dependent to human factor and perspective; and (iv) the massive variation in human skill level of sketching (or abstracting), which makes many sketches misleading and uninterpretable even for humans. A good sketch-related application should embrace these variations, and generalize to different forms, which make it more challenging. This can partly be seen in Fig. 1. Likewise, the literature in Artificial Neural Networks (ANNs) interpretation focuses more on image recognition, since images are rich in details and textures that can easily be interpreted when visualized. The majority of feature visualization research is done on image recognition applications, with the exception of few attempts, such as [7] in which features of a Recurrent Neural Network (RNN) for text generation are visualized.

In this paper, a free-hand sketch classification problem is used to study the learning behavior of CNNs. Human results on the same problem are obtained that can be used as a guide in order to build a classification model that is cognitively inspired.

The rest of this paper is organized as follows. In Section 2 the related work to the research idea is presented. Section 3 gives a detailed description of the proposed technique and the conducted experiments. The preliminary results are discussed in Section 4. Finally, some conclusive remarks along with potential ideas for the future work are given in Section 5.

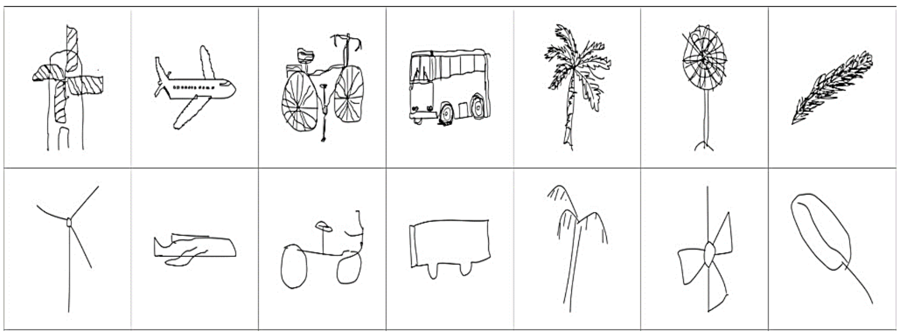


Fig. 1 An example of intra-class abstraction variations, with sample sketches adopted from [6]. The top row represents sketches with maximum number of strokes within each category, while the bottom row shows sketches with minimum number of strokes. Classes from left to right are windmill, airplane, bicycle, bus, palm tree, fan, and feather.

2. Related Work

The underlying research here incorporates sketch recognition and ANNs interpretation fields. Hence, this section gives an overview on some of the related work in both fields.

2.1 Sketch Recognition

Sketch recognition is a multidisciplinary problem that stands at a crossroad between humans and machines. However, most sketch recognition approaches in ML/ANN omit the human factor in their models. Some of the approaches used in sketch recognition are primitive shapes recognizers and objects recognizers, which are previewed in this part.

Primitive shapes recognizers: Primitive shapes recognizers are sketch recognition models that consider a sketch as a set of primitive shapes which, in turn, could be basic geometric shapes or concepts (i.e., meaningful constituents) of the sketch.

Forbus *et al.* have introduced CogSketch [8], which is an open-domain sketch understanding and reasoning system. The interface allows users to free-draw any sketch. Sketches are not restricted to certain domains (e.g., charts, electric circuits, etc.). Each constituent of a sketch is called a "glyph". Spatial and topological relations among glyphs are calculated and used in the recognition phase.

Another domain-independent system for sketch recognition is given in [9]. The system records each stroke, which is defined to be the series of coordinates, along with their time stamps, visited between a mouse-down and mouse-up events. Each stroke first undergoes an imprecise stroke approximation stage, in which a stroke is approximated to one of predefined geometric shapes. The second stage analyzes those primitives and some relations among them, using several heuristic reasoning rules.

The work of [10] provides a language to describe sketched diagrams in a domain along with operations such as drawing, displaying, and editing. Given a LADDER domain description, this description is automatically transformed into a sketch recognition interface for that domain. LADDER is utilized in PaleoSketch [11], which is a sketch recognition and beautification system that builds up on the work provided in [9]. More primitive shapes are included and complex shapes are handled. If a stroke fails to be recognized it is considered to be a complex shape. The stroke is then divided to substrokes (at the points of highest curvature). Finally, the stroke is defined by the set of primitives recognized from its substrokes.

A different approach of sketch recognition is presented in [12], which is a logic-based sketch interpretation approach that enables a formal representation of sketches using description logics, and builds on the results of the experiments described and used in [13] and [14].

Object recognizers: Object recognizers classify a sketch regardless of its parts. A sketch is considered as a whole rather than as a composition of constituents.

In [6], **Eitz *et al.*** do not only provide one of the prominent datasets used for hand-free sketches, but they also record a classification accuracy of 56% using techniques such as HOG features and SVM to classify 250 objects. The human classification results on the same dataset are also provided with a recorded accuracy of 73.1%. Using Deep Learning (DL) on the same dataset of [6], a proposed method in [15] uses a CNN where a sketch is divided into 5 subsketches, each of them represents a channel of the input image. The recorded accuracy of this latter model is 74.9% which surpasses the human accuracy provided in [6] (as mentioned in [15]).

Using a deep RNN, a model is proposed in [16] that employs Gated Recurrent Unit (GRU). A sketch $S = (s_1, s_2, \ldots, s_N)$ is represented as strokes cumulatively accumulated over time, where s_1 represents the first stroke drawn by the user, s_N represents the full final sketch, and N is the number of strokes (or time stamps) of the whole sketch. Each stroke is first provided as an input to [2], then the resulted 4096-dimensional feature layer —of each stroke— is fed as an input to the GRU.

In [17], a loss function is introduced inspired by the Bayesian decision theory. The Bayesian risk of misclassification is calculated for every mini-batch during training. Then, it is backpropagated to a deep ANN to learn a feature vector that, as suggested, better represents sketches and is robust to intra-class variations and inter-class similarities.

2.2 Neural Networks Interpretation

One of the well-known drawbacks of DL methods is the lack of qualitative measures to its performance. In critical applications (e.g., medical, military, or driving), quantitative metrics such as loss value or accuracy are not enough. Such applications require the knowledge of the logical workflow of the model as well as the factors affecting the output. Consequently, the growth of DL interpretation research and techniques gained much attention, and terms like feature visualization [18, 19] and interpretable ML [20] became well known in the AI field.

This part gives an overview of the literature in two approaches: dataset-centric approach, which interprets parts of the network in terms of the training/testing data; and network-centric approach, which interprets the network independently of the dataset.

Dataset-centric approaches: One of the first approaches of interpretation by the means of feature visualization was done in [21] using the deconvolutional networks (deconvNets) technique. In the former paper, a set of experiments are performed (e.g., occlusion, heat maps), which aims at visualizing parts of the image that cause highest activations, if any. DeconvNets itself was earlier presented as a way of performing unsupervised learning [22]. Following the same path, an interactive tool is developed in [23] to visualize the activations of each layer of a trained CNN. Building up on the deconvNet approach, and using regularized gradient ascent, the tool visualizes the activations of every neuron in the network, the preferred stimuli for each neuron, and the top images causing highest activations.

Network-centric approaches: In a slightly different manner**, Simonyan *et al.*** in [24] generate an artificial image that maximally activates a neuron using gradient ascent. Starting with an arbitrary image (i.e., image of zeros or noise) the image is forward propagated through a network where the image pixels are updated with respect to a given set of weights.

3. Materials and Methods

CNNs are widely used in image-related applications. The improvement of its metrics like accuracy has long been pursued by research. In this section, we suggest a technique to investigate a *qualitative* measure of CNNs using the sketch recognition problem. The aim of this technique is to interpret and analyze CNNs behavior and challenge their learning capabilities rather than to increase the accuracy obtained in a typical recognition task. Also, since sketching is a human activity, the human factor is used as a reference and a guide in interpretation and analyses, extracting what we might consider as an aspect of behavioral similarities between CNNs and humans.

The unique nature of sketches requires distinctive approaches. Quantitative information describing a sketch like number of strokes, stroke length, curvatures, or number of points may not be sufficient for an efficient sketch recognition model (as shown in Fig. **1**). For this reason, we investigate the qualitative measures affecting the recognition process to build a recognition model especially designed for sketches, incorporating the human factor.

Our idea is twofold: 1) Study the 'learning transfer' behavior of CNNs when presented with new object classes and interpret misclassifications significance; and 2) Compare CNNs results to that of humans when it comes to similarity detection or visual analogy making (Visual analogy making is used here in a narrow sense. Analogy making is rather a broad term which refers to the ability to map and transfer knowledge between a source domain and a target domain).

We have designed two experiments to fulfill the aforementioned goals. The first experiment is called "Sketch Similarity: CNN", which studies the detection of visually similar sketches along with other properties of CNNs. The second experiment is called "Sketch Similarity: Humans", which studies similarity detection among humans and uses human-based results in analyzing CNNs behavior.

3.1 Sketch Similarity: CNN

This experiment aims at studying how CNNs convey previously acquired knowledge to new data. It also studies some of the misclassified sketches to find out whether or not they might indicate certain learning patterns of the network. Finally, some of the feature vectors produced by the CNN are visualized to view how different objects seem to be 'perceived' by the network. A detailed overview of the experiment setup and procedure is given in this part.

Setup TensorFlow [25] is used for building our suggested CNN. TensorFlow is an open-source end-to-end platform for ML introduced by Google Brain. The visualizations are produced using Lucid (https://github.com/tensorflow/lucid): a collection of infrastructure and tools for research in ANNs interpretability. As outlined in Fig. **2**, the CNN consists of 2 convolution layers, 2 pooling layers, a Fully Connected (FC) layer, and an output layer. Filters sizes at the first convolution layer are chosen to be relatively large of size 7 × 7 owing to the nature of sketches. The lack of texture and fine details make larger filters more efficient. In the training phase Adaptive Moment Estimation (ADAM) optimizer [26] is used. The weights are initialized using the Xavier initialization method [27].

Dataset We use subsets of the TU-Berlin [6] dataset to perform this experiment. To serve our particular purpose, our network is trained using one set of classes in the training phase yet tested on a different set of classes in the testing phase. For example, the network is trained on sketches of "Bicycle" objects, then tested on sketches of "Motorcycle" objects. Since we believe that our approach is novel and fresh, it seemed plausible for us to base our preliminary choice of the testing classes on what we thought would give more insights into visually analogous sketches of objects (These are arbitrary choices made by the authors of this paper as an initial step to serve the research idea, which would be supported or contradicted by the results) sketches of objects. An overall of 30 classes from the TU-Berlin dataset are used in this experiment: 15 classes for training and other (different) 15 for testing. Table 1 Show each of the 15 training classes along with their corresponding testing classes.

Preprocessing In this phase, the raw data is prepared to be compatible with the given architecture in terms of size, color range, number of channels, etc. The original TU-Berlin dataset contains 1111×1111 grayscale images of hand-drawn sketches of different objects with a white background and black sketch. Below is the list of augmentation and resizing steps executed.

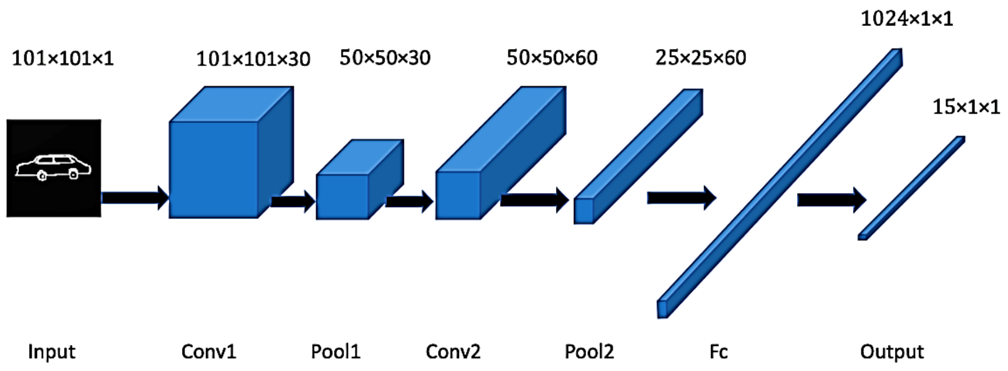


Fig. 2 The overall architecture of the proposed CNN

Table 1 Original classes and the corresponding new classes

Training	Testing	Training	Testing	Training	Testing
Tree	Palm tree	Car	Bus	Feather	Leaf
Violin	Guitar	Mushroom	Umbrella	Airplane	Flying bird
Key	Bottle opener	Ship	Sailboat	Saxophone	Smoking pipe
Parachute	Hot air balloon	Flower	Windmill	Axe	Hammer
Bicycle	Motorcycle	Lighter	Candle	Cell phone	Calculator

Data Augmentation: Data Augmentation is a series of transformations that are conventionally applied to the training data to increase the network's invariance to specific data and reduce overfitting. Each of the following 4 transformations is applied to one-eighth (12.5%) of each class' images, such that no overlapping of two or more transformations takes place. A representation of these transformations is shown in Fig. 3.

1. Zoom: Images are zoomed at the center 1.5 times of the original size but with maintaining the original size of the image. (i.e., 1111 × 1111).
2. Flipping: Images are flipped horizontally (i.e., mirrored).
3. Rotation: A 30◦ degree counterclockwise rotation.
4. Shifting: Pixels of images are shifted 100 pixels (to left and right, interchangeably).

Resizing: Since the original size of an image is relatively huge, the use of an available resizing method, such as PIL (Python Image Library https://python-pillow.org) or scikit-image (https://scikit-image.org) libraries, would lead to distorting the image due to the abstract nature of sketches, as shown in Fig. 4. We propose a resizing method especially developed to sketches to maintain and enhance the pixels representing an object.

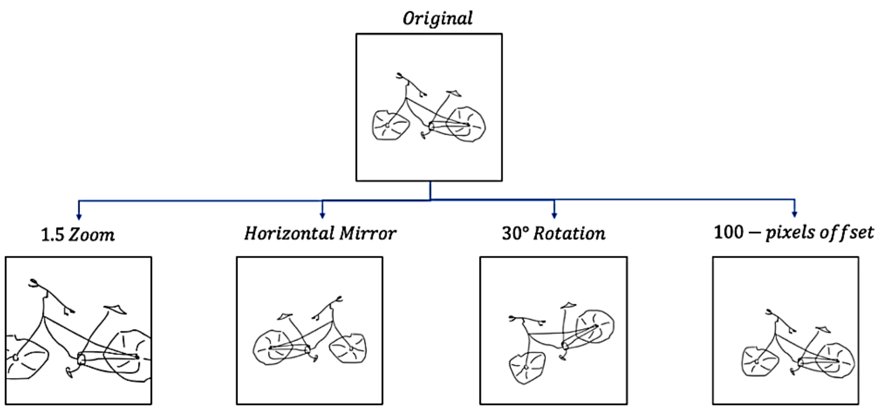


Fig. 3 A representation of the 4 transformations applied in data augmentation

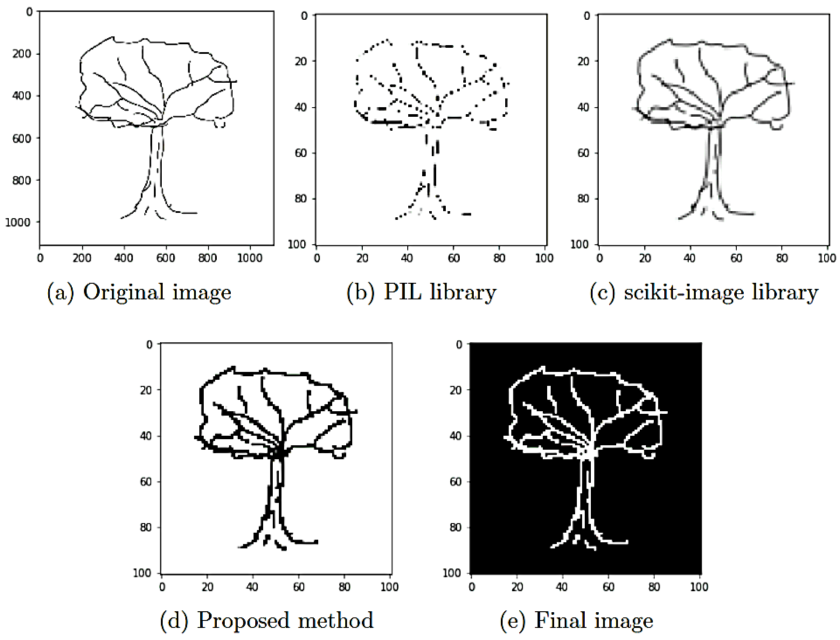


Fig. 4 An example showing the effect of applying different resizing methods on the same sketch

The resizing process is depicted in Fig. 5 and is explained as follows. Each image is resized to 101 × 101 to avoid overlapping or clipping of pixels, and each block of size 11 × 11 of the original images is transformed to a single pixel in the new resized image (note that 101 and 11 are the two non-trivial divisors of 1111). The new pixel takes a value of 1 if all the pixels of the original block have the value 1. Otherwise, the pixel takes the value 0. Although the original images are greyscale, we suggest that pixels of a sketch image are either background pixels or pixels representing the object itself, with no intermediate values. Thus, the pixel values of images are transformed to either 0s or 1s. As a final step, all background pixels are given the value 0, and what seems to be pixels representing the object are given the value 1 to emphasize their importance.

It is worth mentioning that a non-trivial 4% improvement in recognition accuracy is obtained when the proposed resizing and rescaling method is tested against available resizing methods with greyscale images.

Procedure The overall dataset is split into two parts: 15 classes used for training; other 15 for testing. The training dataset is split 80% for training and 20% for testing the performance of the CNN on the original classes. After training the CNN, the performance is tested on the corresponding classes. The results and analyses of this experiment are provided in section 4.1.

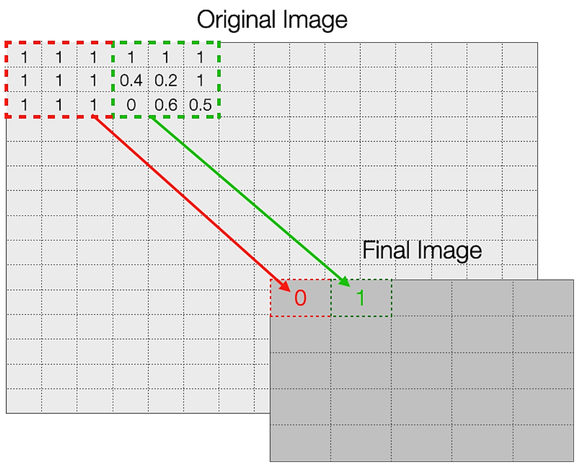


Fig. 5 An example of the overall resize method used on images: each block of 11×11 pixels in the original image is converted to a corresponding pixel in the resized image with either a value of 0 if all pixels represent background pixels in the original image, or 1 if at least one of them represents part of the sketch

3.2 Sketch Similarity: Humans

The second experiment aims at collecting information about how humans infer 'similarities' among different object classes. The results obtained in this experiment are also used in the analyses of 3.1 to analyze the similarities between the human's and the CNN's behavior, if any.

One of the main differences between humans and any computational model is that the computational model's knowledge is restricted to the amount and type of representation it is fed. While, in case of humans, it is almost impossible to control or accurately measure the knowledge of a human. For this purpose, and to have a semi-fair comparison, the humans are limited to a set of choices in the experiment as explained below.

Setup: A webpage is designed as an interface to facilitate performing the experiment and remotely collecting the data (due to the COVID-19 pandemic at the time). A total of 60 participants performed the experiment (19 male and 41 female). The age range of participants varies from 5 to 70 years old. This range is intended to avoid biased results based on experience or education. Instructions are given to the participants in both Arabic and English languages.

Procedure: This experiment consists of two identical parts. In each part, the participants are asked to divide a given set of 36 sketches of different object classes into at most 5 classes. The participants are directed to divide the given set based on how visually similar the sketches look from their perspective. In the second part, the same process is repeated with a different set of 36 sketches. Finally, the participants are asked to fill in their personal information and submit their answers. The results and analyses of this experiment are provided in section 4.2.

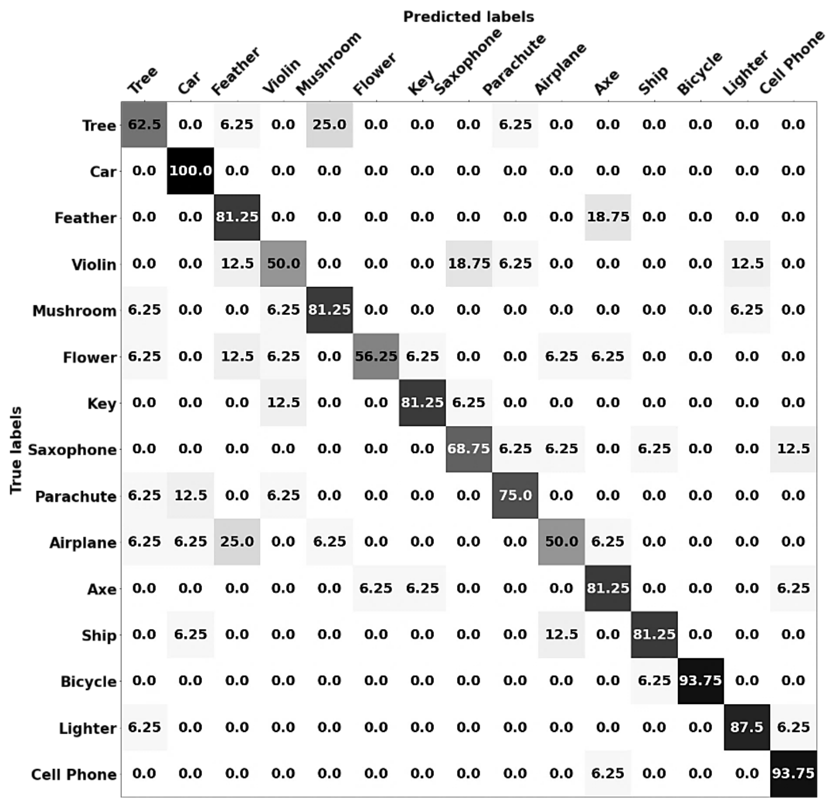
4. Results and Discussion

In this section, we present and discuss the outcomes of the two experiments introduced in section 3. The results obtained from the first experiment are given in section 4.1. In section 4.2, the results of the second experiment are presented and compared with the results of section 4.1. In section 4.3, we investigate the factors contributing to the decisions made by the CNN.

4.1 Sketch Similarity Results: CNN

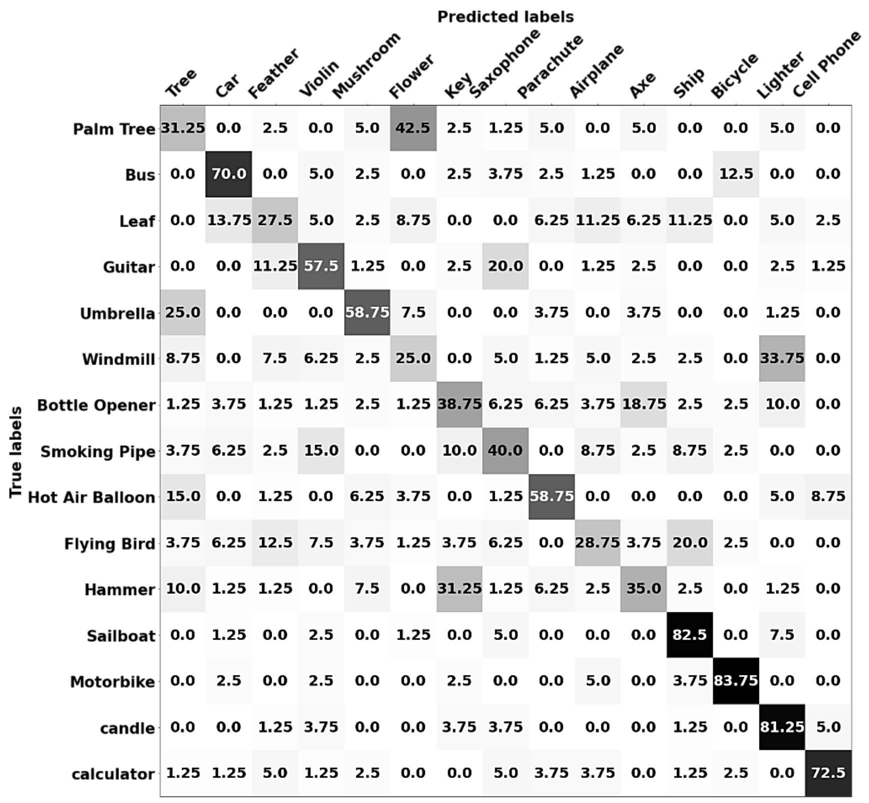
The performance of the proposed CNN on the 'originally trained' data is shown in Fig. 6, recording an overall recognition accuracy of 76% when testing on the testing data, i.e., 16 images of each of the original objects. Where, accuracy means the percentage of correctly recognized objects in the testing dataset.

Testing on the overall 80 sketches of each of the proposed one-to-one 'analogical objects', the CNN records an overall recognition accuracy of 58%. The detailed performance presented in Fig. 7 shows that 5 out of 15 objects record an accuracy greater than or equal to 70%, which approaches the accuracies obtained on the original objects. Whereas in case of a "Guitar" and a "Sailboat" the accuracy exceeds the original object's accuracy by 7% and 1%, respectively. This could be owing to the nature of the sketch itself. A "Guitar" could be considered as a simplified version of a "Violin" since the latter is often sketched including a violin bow. Similarly, a "Ship" could be considered as a complex shape of a "Sailboat". Hence, it seems that the network captures core features defining an abstract object rather than a decorated one.


Predicted labels
True labels

	Tree	Car	Feather	Violin	Mushroom	Flower	Key	Saxophone	Parachute	Airplane	Axe	Ship	Bicycle	Lighter	Cell Phone
Tree	62.5	0.0	6.25	0.0	25.0	0.0	0.0	0.0	6.25	0.0	0.0	0.0	0.0	0.0	0.0
Car	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Feather	0.0	0.0	81.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	18.75	0.0	0.0	0.0	0.0
Violin	0.0	0.0	12.5	50.0	0.0	0.0	0.0	18.75	6.25	0.0	0.0	0.0	0.0	12.5	0.0
Mushroom	6.25	0.0	0.0	6.25	81.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.25	0.0
Flower	6.25	0.0	12.5	6.25	0.0	56.25	6.25	0.0	0.0	6.25	6.25	0.0	0.0	0.0	0.0
Key	0.0	0.0	0.0	12.5	0.0	0.0	81.25	6.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Saxophone	0.0	0.0	0.0	0.0	0.0	0.0	0.0	68.75	6.25	6.25	0.0	6.25	0.0	0.0	12.5
Parachute	6.25	12.5	0.0	6.25	0.0	0.0	0.0	0.0	75.0	0.0	0.0	0.0	0.0	0.0	0.0
Airplane	6.25	6.25	25.0	0.0	6.25	0.0	0.0	0.0	0.0	50.0	6.25	0.0	0.0	0.0	0.0
Axe	0.0	0.0	0.0	0.0	0.0	6.25	6.25	0.0	0.0	0.0	81.25	0.0	0.0	0.0	6.25
Ship	0.0	6.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	12.5	0.0	81.25	0.0	0.0	0.0
Bicycle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.25	93.75	0.0	0.0
Lighter	6.25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	87.5	6.25
Cell Phone	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.25	0.0	0.0	0.0	93.75

Fig. 6 A confusion matrix that shows the percentage values of the results obtained by the CNN on the original classes.

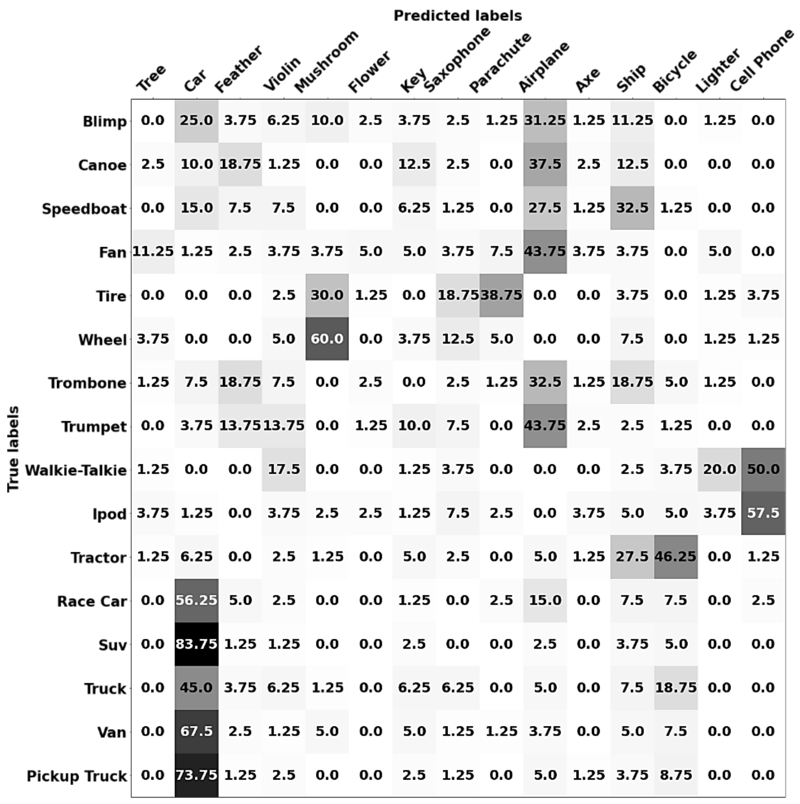

Predicted labels
True labels

	Tree	Car	Feather	Violin	Mushroom	Flower	Key	Saxophone	Parachute	Airplane	Axe	Ship	Bicycle	Lighter	Cell Phone
Palm Tree	31.25	0.0	2.5	0.0	5.0	42.5	2.5	1.25	5.0	0.0	5.0	0.0	0.0	5.0	0.0
Bus	0.0	70.0	0.0	5.0	2.5	0.0	2.5	3.75	2.5	1.25	0.0	0.0	12.5	0.0	0.0
Leaf	0.0	13.75	27.5	5.0	2.5	8.75	0.0	0.0	6.25	11.25	6.25	11.25	0.0	5.0	2.5
Guitar	0.0	0.0	11.25	57.5	1.25	0.0	2.5	20.0	0.0	1.25	2.5	0.0	0.0	2.5	1.25
Umbrella	25.0	0.0	0.0	0.0	58.75	7.5	0.0	0.0	3.75	0.0	3.75	0.0	0.0	1.25	0.0
Windmill	8.75	0.0	7.5	6.25	2.5	25.0	0.0	5.0	1.25	5.0	2.5	2.5	0.0	33.75	0.0
Bottle Opener	1.25	3.75	1.25	1.25	2.5	1.25	38.75	6.25	6.25	3.75	18.75	2.5	2.5	10.0	0.0
Smoking Pipe	3.75	6.25	2.5	15.0	0.0	0.0	10.0	40.0	0.0	8.75	2.5	8.75	2.5	0.0	0.0
Hot Air Balloon	15.0	0.0	1.25	0.0	6.25	3.75	0.0	1.25	58.75	0.0	0.0	0.0	0.0	5.0	8.75
Flying Bird	3.75	6.25	12.5	7.5	3.75	1.25	3.75	6.25	0.0	28.75	3.75	20.0	2.5	0.0	0.0
Hammer	10.0	1.25	1.25	0.0	7.5	0.0	31.25	1.25	6.25	2.5	35.0	2.5	0.0	1.25	0.0
Sailboat	0.0	1.25	0.0	2.5	0.0	1.25	0.0	5.0	0.0	0.0	0.0	82.5	0.0	7.5	0.0
Motorbike	0.0	2.5	0.0	2.5	0.0	0.0	2.5	0.0	0.0	5.0	0.0	3.75	83.75	0.0	0.0
candle	0.0	0.0	1.25	3.75	0.0	0.0	3.75	3.75	0.0	0.0	0.0	1.25	0.0	81.25	5.0
calculator	1.25	1.25	5.0	1.25	2.5	0.0	0.0	5.0	3.75	3.75	0.0	1.25	2.5	0.0	72.5

Fig. 6 A confusion matrix that shows the percentage values of the results obtained by the CNN on the one-to-one analogical classes.

An expansion of the results is shown in Fig. 8, which shows the different class accuracies obtained when a new set of objects, 80 sketches of each, are tested on the same CNN. Some of the new objects were chosen based on the misclassifications of human classification results provided in [6] (http:/cybertron.cg.tu-berlin.de/eitz/projects/classifysketch/human_classification/index.html). It can be seen from Fig. 8 that the CNN provides a relatively high accuracy rate on different objects that are thought to be "similar-to-car" (i.e., Bus, Race Car, SUV, Truck, Van, Pickup Truck, and Tractor). A "Car" has a 100% classification accuracy on the 16 testing sketches of the original dataset, and the overall "similar-to-car" objects–omitting the "Tractor"–are recognized as a car with an average accuracy of 66% tested on a total of 480 images; 80 for each object. The "Tractor" is classified by our CNN as a bicycle rather than a car half of the times (with an accuracy of 46%). Nevertheless, this is acceptable because "visual" similarity is still obvious (in contrast with "functional" or whatever other kind of similarity). This is further investigated in 4.3.

One of the findings that we see interesting is the following: using black and white images (i.e., pixel values of zeros or ones) does increase the accuracy of the network by a range of 3–4% compared to greyscale images that are trained and tested on the same architecture.



True labels \ Predicted labels	Tree	Car	Feather	Violin	Mushroom	Flower	Key	Saxophone	Parachute	Airplane	Axe	Ship	Bicycle	Lighter	Cell Phone
Blimp	0.0	25.0	3.75	6.25	10.0	2.5	3.75	2.5	1.25	31.25	1.25	11.25	0.0	1.25	0.0
Canoe	2.5	10.0	18.75	1.25	0.0	0.0	12.5	2.5	0.0	37.5	2.5	12.5	0.0	0.0	0.0
Speedboat	0.0	15.0	7.5	7.5	0.0	0.0	6.25	1.25	0.0	27.5	1.25	32.5	1.25	0.0	0.0
Fan	11.25	1.25	2.5	3.75	3.75	5.0	5.0	3.75	7.5	43.75	3.75	3.75	0.0	5.0	0.0
Tire	0.0	0.0	0.0	2.5	30.0	1.25	0.0	18.75	38.75	0.0	0.0	3.75	0.0	1.25	3.75
Wheel	3.75	0.0	0.0	5.0	60.0	0.0	3.75	12.5	5.0	0.0	0.0	7.5	0.0	1.25	1.25
Trombone	1.25	7.5	18.75	7.5	0.0	2.5	0.0	2.5	1.25	32.5	1.25	18.75	5.0	1.25	0.0
Trumpet	0.0	3.75	13.75	13.75	0.0	1.25	10.0	7.5	0.0	43.75	2.5	2.5	1.25	0.0	0.0
Walkie-Talkie	1.25	0.0	0.0	17.5	0.0	0.0	1.25	3.75	0.0	0.0	0.0	2.5	3.75	20.0	50.0
Ipod	3.75	1.25	0.0	3.75	2.5	2.5	1.25	7.5	2.5	0.0	3.75	5.0	5.0	3.75	57.5
Tractor	1.25	6.25	0.0	2.5	1.25	0.0	5.0	2.5	0.0	5.0	1.25	27.5	46.25	0.0	1.25
Race Car	0.0	56.25	5.0	2.5	0.0	0.0	1.25	0.0	2.5	15.0	0.0	7.5	7.5	0.0	2.5
Suv	0.0	83.75	1.25	1.25	0.0	0.0	2.5	0.0	0.0	2.5	0.0	3.75	5.0	0.0	0.0
Truck	0.0	45.0	3.75	6.25	1.25	0.0	6.25	6.25	0.0	5.0	0.0	7.5	18.75	0.0	0.0
Van	0.0	67.5	2.5	1.25	5.0	0.0	5.0	1.25	1.25	3.75	0.0	5.0	7.5	0.0	0.0
Pickup Truck	0.0	73.75	1.25	2.5	0.0	0.0	2.5	1.25	0.0	5.0	1.25	3.75	8.75	0.0	0.0

Fig. 7 A (confusion) matrix showing the percentage of the results obtained by the CNN on miscellaneous analogical classes

4.2 Sketch Similarity Results: Humans

Some of the results obtained in this experiment are shown in Table 2 which shows the top 3 predictions obtained by humans for some of the object classes. Table 3 shows the top 3 predictions obtained by the CNN for the same object classes. To have a more comprehensible insight into the results, Table 4 shows the intersection between them. By intersection we mean the set of objects that are found in the top 3 predictions of a given object obtained by both humans and CNN, regardless of their order.

The bottom two rows in Table 4 show the misclassifications obtained by the CNN on the original data which also matches humans' decisions.

We can observe an underlying similarity in the predictions made by humans and the CNN. In Fig. 6, an "Airplane" is misclassified as a "Feather" with 25% accuracy, which comes next to being correctly classified as an "Airplane" with an accuracy of 50%. In case of humans, the top two predictions of an "Airplane" are also "Airplane" and "Feather". In Fig. 7, a "Hammer" is almost equally classified as a "Key" and an "Axe" by the CNN, which matches the first two responses made by humans.

It might be hard to comprehend when the *image* of each object is in mind, but in case of sketches the abstraction level seems to create visual similarities between what are originally different objects.

The network seems to effectively suggest a kind of visual similarity, generalized from outlining boundaries in sketches or silhouettes of 'seemingly-similar' objects

Table 2 Top three recognition predictions obtained by humans in 3.2

Object Class	Top 3 predictions		
	First Pred.	Second Pred.	Third Pred.
Palm Tree	Flower	Key	Lighter
Leaf	Airplane	Flower	Feather
Umbrella	Parachute	Mushroom	Tree
Windmill	Key	Flower	Axe
Bottle Opener	Key	Flower	Mushroom
Smoking Pipe	Saxophone	Key	
Hot Air Balloon	Mushroom	Parachute	Tree
Flying Bird	Feather	Airplane	
Hammer	Key	Axe	Parachute
Candle	Lighter		

Table 3 Top three recognition predictions obtained by the CNN in 3.1

Object Class	Top 3 predictions		
	First Pred.	Second Pred.	Third Pred.
Palm Tree	Flower	Tree	Lighter
Leaf	Feather	Car	Airplane
Umbrella	Mushroom	Tree	Flower
Windmill	Lighter	Flower	Tree
Bottle Opener	Key	Axe	Lighter
Smoking Pipe	Saxophone	Violin	Key
Hot Air Balloon	Parachute	Tree	Cell Phone
Flying Bird	Airplane	Ship	Feather
Hammer	Axe	Key	Tree
Candle	Lighter		

Table 4 The intersection of the top 3 predictions made by humans and the proposed CNN

Object Class	Intersection of top three predictions
Palm Tree	*{*Flower, Lighter*}*
Leaf	*{*Feather, Airplane*}*
Umbrella	*{*Mushroom, Tree*}*
Windmill	*{*Flower*}*
Bottle Opener	*{*Key*}*
Smoking Pipe	*{*Saxophone, Key*}*
Hot Air Balloon	*{*Parachute, Tree*}*
Flying Bird	*{*Airplane, Feather*}*
Hammer	*{*Axe, Key*}*
Candle	*{*Lighter *}*
Tree	*{*Mushroom*}*
Airplane	*{*Feather*}*

4.3 Classification vs. Visualization

One of the well-known methods used in CNN interpretation is using feature visualization, especially in image-based data. Lucid library is used to visualize some of the final and middle layers neurons.

The visualizations of the final layer neurons show the so-called “template” which the network has built for each object class as shown in Fig. 9. The figure shows the visualization produced by the network for some objects. Despite the fact that the *images* of these objects have great differences, the produced visualization of the sketches have an underlying similarity which, in many of our cases, matches the results obtained by both the CNN and humans. Tables 2 and 3 show the interchangeability of “Tree”, “Mushroom”, “Parachute”, “Hot Air Balloon” etc., in both misclassifications and similarity detection. In a similar manner, the visualizations shown in Fig. 10 also show an underlying similarity which can be seen despite the lack of rich details. It can also give an insight into why a “Sailboat” exceeded the recognition accuracy of the “Ship”, which is the “originally trained” object, as explained in Sec. 3.1.

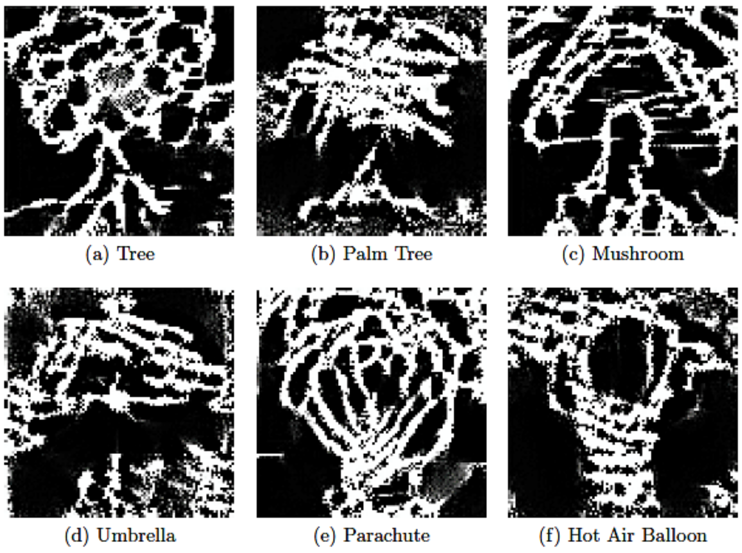


Fig. 8 Sample visualization for different objects

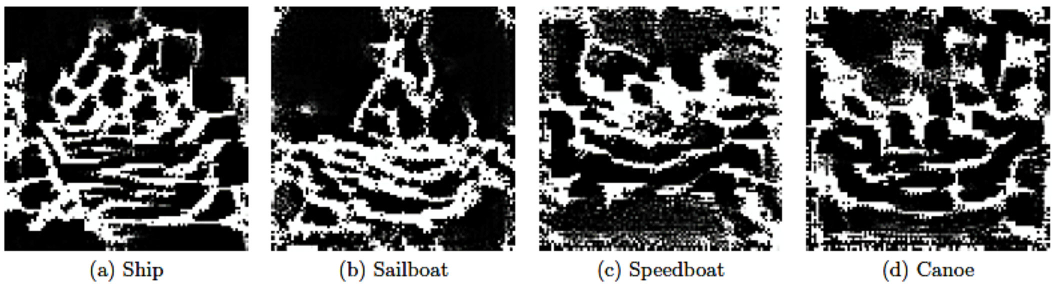


Fig. 9 Sample visualizations of different “similar-to-Ship” objects

In case of images, the produced visualizations include details like colors and textures which are visually informative of the features that a CNN learns. Despite the lack of these details in sketches, the produced visualizations can still give an interpretation of the features perceived by the CNN.

In Fig. 11 the top row shows the visualization of “Motor Bike”, “Bicycle” and “Tractor”, respectively. It can be seen (particularly from the high-density parts of the image) that what we recognize as ‘wheels’ play an important role in the classification process. The visualization of the “Tractor” shows a similar high-density distribution manner which can further be seen in the second row of the same figure. In the second row, a sample of the misclassified sketches (of the “Tractor” as a “Bicycle”) also shows an emphasis on the ‘wheels’ with a big relative size compared to other vehicles shown in Fig. **12.** The ability of the CNN to identify different sketches can be seen in Fig. **12**, which shows some "similar-to-car" sketches that vary in shape, number of strokes, and abstraction level, and are recognized as a "Car".

Figures **13** and **14** show some randomly selected visualization, adopted from the 2nd convolutional layer and the FC layer. Despite the lack of interpretable shapes, the figures still show an activation behavior towards some patterns.

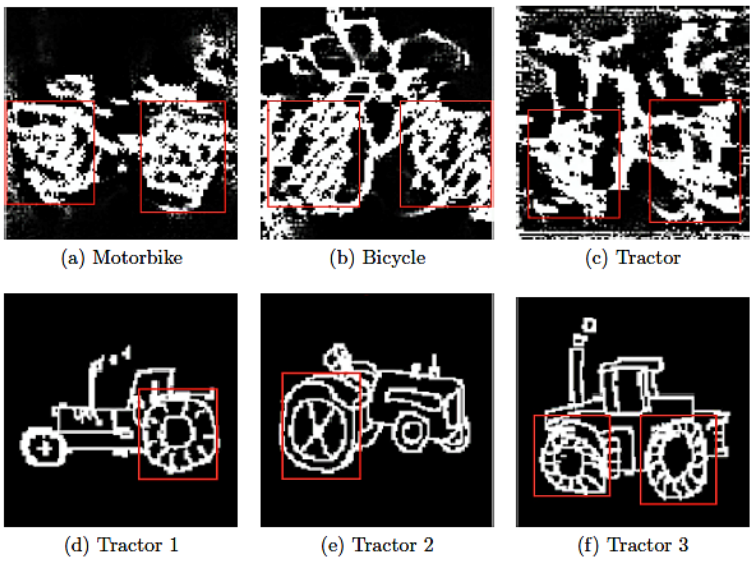


Fig. 11 visualization vs. Misclassification

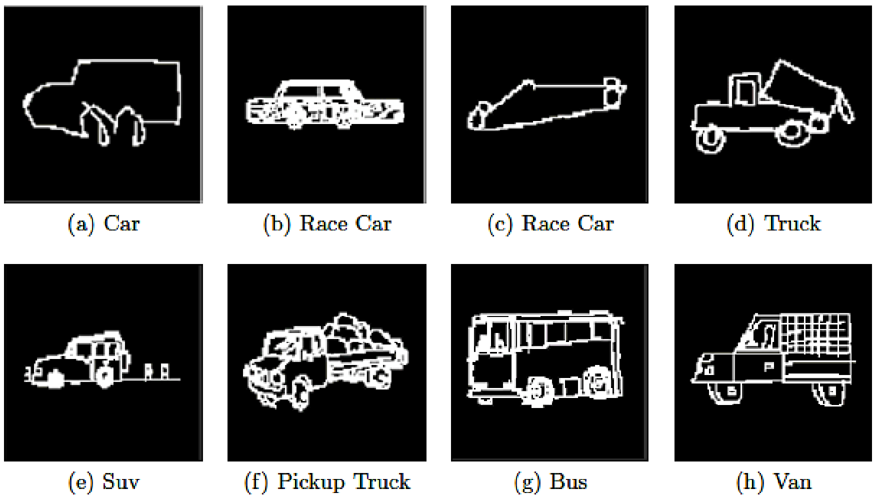


Fig 12 Sample of different objects that are classified as "Car"

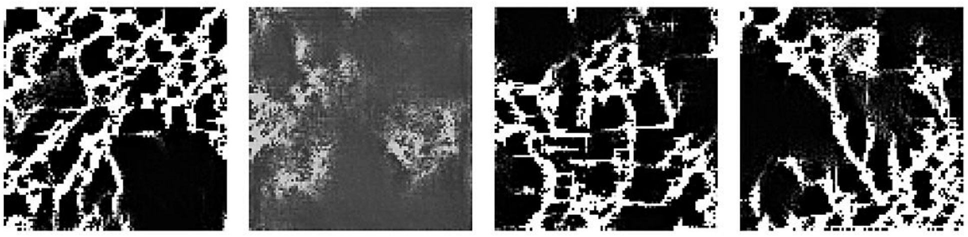


Fig 13 Randomly chosen visualizations from the 1024 FC layer

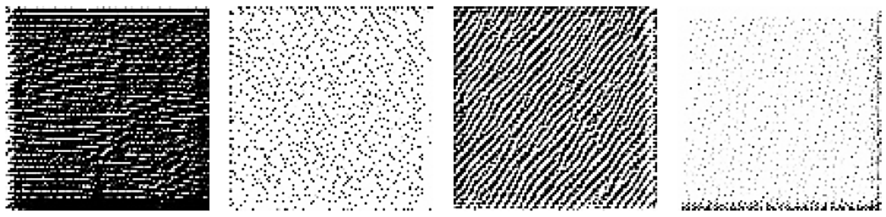


Fig 14 Randomly chosen visualizations from the 2nd convolutional layer

5. Conclusion and Future Work

In this paper, we investigate a new criterion of the learning process in CNNs: (visual) analogy making or transfer of learning. We address the question of how well a network behaves on classes of sketches of objects, and whether or not the network would be able to project its learning on new, visually similar classes. This can help in building a low-level recognizer that, for instance, classifies "vehicles" regardless of their specific type. Knowing this would help in correctly identifying a wide range of objects–not necessarily trained on–and reduce the size of datasets needed to train a recognition model.

Studying and improving the performance of CNNs on given classes is already a widely handled research problem. Sketch recognition is a human oriented procedure. Drawing a sketch solely depends on a person. Sketch recognition systems that lack the incorporation of the human factor might not be of the best efficiency in many contexts involving human-based interaction, especially for a broad segment of people that lack artistic talents. Our experiments have shown that hand-drawn sketches could hold great differences than images in processes like recognition and classification. Hence, a cognitively inspired approach could better understand and guide humans in sketch classification, retrieval, or auto-completion systems. Instead of image-based similarity detection, employing these results into a sketch-based model would give meaningful suggestions to the user that meet their aims, and help reduce recognition ambiguity.

The work presented in this paper is already a continuation to ongoing research of building a model that is inspired by human cognition. As a possible future research, we support the continuation of this work in the same field, improving the performance of machine learning-based models such as our tested one. We suggest to study the building process of sketches more thoroughly in a cognitively inspired manner. Using LSTMs is a possible future approach to incorporate the temporal nature of sketches, and compare their learning behavior to that of CNNs and humans. Also, to tackle the stroke continuation problem in an approach that is based on the human behavior not only on mathematical properties of the stroke. Finally, we believe that employing Generative Adversarial Networks (GANs) in the sketch generation field might be promising and is worth pursuing, particularly when coupled with learning transfer.

6. Abbreviations

AI: Artificial Intelligence; **ANN:** Artificial Neural Network; **CNN:** Convolutional Neural Network; **DL:** Deep Learning; **FC:** Fully Connected; **GAN**: Generative Adversarial Network; **GRU:** Gated Recurrent Unit; **HCI:** Human-Computer Interaction; **HOG:** Histogram of Gradients; **ML**: Machine Learning; **RNN:** Recurrent Neural Network; **SVM**: Support Vector Machine.

7. References

1. **Henshilwood, C. S., d'Errico, F., van Niekerk, K. L., Dayet, L., Queffelec, A. and Pollarolo, L. (2018).** An abstract drawing from the 73,000-year-old levels at blombos cave, south africa. Nature, **562**(7725): 115 – 118.
2. **Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T. and Houlsby, N. (2020).** An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010. 11929.
3. **Wang, W., Yang, J., Xiao, J., Li, S. and Zhou, D. (2015).** Face recognition based on deep learning. In Human Centered Computing, Cham. Springer International Publishing, 812 – 820.
4. **Johnson, J., Karpathy, A. and Fei-Fei, L. (2015).** Densecap: Fully convolutional localization networks for dense captioning.

5. **Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N. and Terzopoulos, D. (2021).** Image segmentation using deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence.
6. **Eitz, M., Hays, J. and Alexa, M. (2012).** How do humans sketch objects? ACM Trans. Graph. (Proc. SIGGRAPH), **31**(4): 44:1 – 44:10
7. **Karpathy, A., Johnson, J. and Li, F. (2015).** Visualizing and understanding recurrent networks. *CoRR*, abs/1506.02078.
8. **Forbus, K. D., Usher, J., Lovett, A., Lockwood, K. and Wetzel, J. (2010).** Cogsketch: Sketch understanding for cognitive science research and for education. In Spatial Cognition, VII, 4 – 44.
9. **Yu, B. and Cai, S. (2003).** A domain-independent system for sketch recognition. In Proc. of the 1st Intel. Conf. on Computer Graphics and Interactive Techniques in Australasia and South East Asia, GRAPHITE '03, 141146.
10. **Hammond, T. and Davis, R. (2007).** Ladder, a sketching language for user interface developers. In ACM SIGGRAPH 2007 Courses, SIGGRAPH '07, 35es
11. **Paulson, B. and Hammond, T. (2008).** Paleosketch: Accurate primitive sketch recognition and beautification. In Proc. of the 13th Intel. Conf. IUI '08, Association for Computing Machinery, 110.
12. **Abdelghaffar, N., Abdelfattah, A. M. H., Taha, A. A. and Khamis, S. M. (2019).** Accentuating features of description logics in high-level interpretations of hand-drawn sketches. Ku¨nstliche Intell., **33**(3): 253 – 265.
13. **Abdelfattah, A. M. H., Zakaria, W., Abdelghaffar, N., Abdelmoneim, N.,** Schneider, S. and Kühnberger, K. **(2016).** A preliminary assessment of the role of conceptual salience in automatic sketching. In Proc. of the 4th Intel. Workshop on Artificial Intelligence and Cognition co-located with the Joint Multi-Conference on HLAI 2016, CEUR Workshop Proceedings 1895: 81 – 92.
14. **Abdelfattah, A. M. H. and Zakaria, W. (2017).** Employing a restricted set of qualitative relations in recognizing plain sketches. In KI 2017: Advances in AI, **10505**: 3 – 14.
15. **Yu, Q., Yang, Y., Liu, F., Song, Y.-Z., Xiang, T. and Hospedales, T. M. (2017).** Sketch-a-net: A deep neural network that beats humans. International Journal of Computer Vision, **122**(3): 411 – 425.
16. **Sarvadevabhatla, R. K., Kundu, J., and Venkatesh, B. R. (2016).** Enabling my robot to play pictionary: Recurrent neural networks for sketch recognition.
17. **Mishra, A. and Singh, A. K. (2018).** Deep embedding using bayesian risk minimization with application to sketch recognition.
18. **Olah, C., Schubert, L. and Mordvintsev, A. (2017).** Feature visualization. Distill.
19. **Nguyen, A., Yosinski, J. and Clune, J. (2019).** Understanding Neural Networks via Feature Visualization: A Survey. Springer International Publishing, Cham., 55 – 76.
20. **Molnar, C., Casalicchio, G. and Bischl, B. (2020).** Interpretable machine learning – a brief history, state-of-the-art and challenges.
21. **Zeiler, M. D. and Fergus, R. (2014).** Visualizing and understanding convolutional networks. In Computer Vision – ECCV 2014, Cham. Springer International Publishing. 818 – 833.
22. **Zeiler, M., Taylor, G. and Fergus, R. (2011).** Adaptive deconvolutional networks for mid and high level feature learning. In ICCV 2011, 2018 – 2025.
23. **Yosinski, J., Clune, J., Nguyen, A., Fuchs, T. and Lipson, H. (2015).** Understanding Neural Networks Through Deep Visualization. In Deep Learning Workshop, ICML.
24. **Simonyan, K., Vedaldi, A. and Zisserman, A. (2014).** Deep inside convolutional networks: Visualising image classification models and saliency maps. In Workshop at International Conference on Learning Representations.
25. **Abadi, M., Agarwal, A., Barham, P., Brevdo, E., et al. (2015).** TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
26. **Kingma, D. P. and Ba, J. (2015).** Adam: A method for stochastic optimization. CoRR, abs/1412.6980.
27. **Glorot, X. and Bengio, Y. (2010).** Understanding the difficulty of training deep feedforward neural networks. Proc. of the 13th Intel. Conference on Artificial Intelligence and Statistics, **9**:249 – 256.