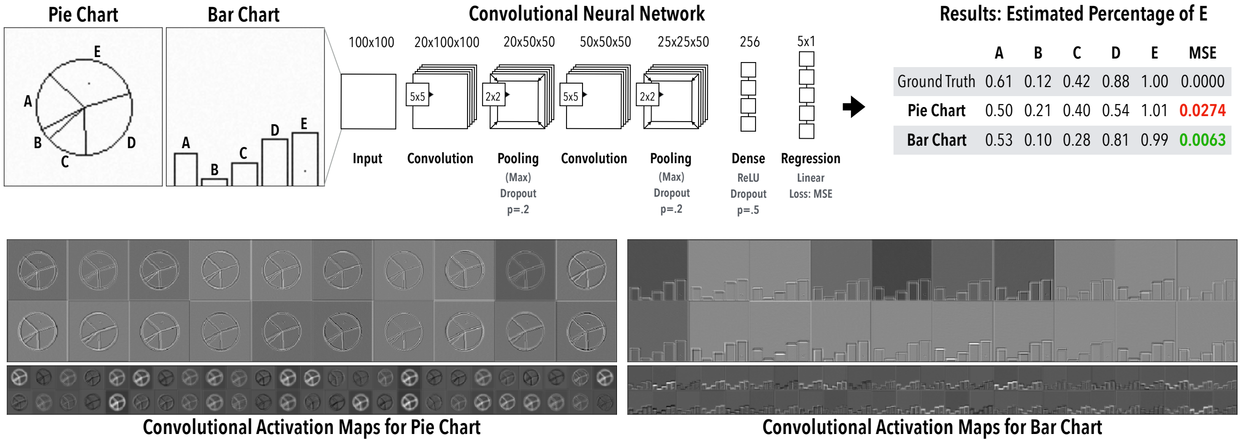
Evaluating 'Graphical Perception' with CNNs

Daniel Haehn, James Tompkin, and Hanspeter Pfister



	A	B	C	D	E	MSE
Ground Truth	0.61	0.12	0.42	0.88	1.00	0.0000
Pie Chart	0.50	0.21	0.40	0.54	1.01	0.0274
Bar Chart	0.53	0.10	0.28	0.81	0.99	0.0063

Fig. 1: **Computing Cleveland and McGill's Position-Angle Experiment using Convolutional Neural Networks.** We replicate the original experiment by asking visual cortex inspired machine learning classifiers to assess the relationship between values encoded in pie charts and bar charts. Similar to the findings of Cleveland and McGill [7], our experiments show that CNNs read quantities more accurately from bar charts (mean squared error, MSE in green).

Abstract—Convolutional neural networks can successfully perform many computer vision tasks on images, and their learned representations are often said to mimic the early layers of the visual cortex. But can CNNs understand graphical perception for visualization? We investigate this question by reproducing Cleveland and McGill's seminal 1984 experiments, which measured human perception efficiency of different visual encodings and defined elementary perceptual tasks for visualization. We measure the graphical perceptual capabilities of four classifiers on a) elementary perceptual tasks with increasing parametric complexity, b) the position-angle experiment that compares pie charts to bar charts, c) the position-length experiment that compares grouped and divided bar charts, and d) the bars and framed rectangles experiment where visual cues aid perception. We also study how feed-forward neural networks obey Weber's law, which defines the proportional relation between perceivable information and distribution density. We present the results of these experiments to foster the understanding of how CNN classifiers succeed and fail when applied to data visualizations.

Index Terms—Machine Perception, Deep Learning

1 Introduction

Convolutional neural networks (CNNs) have been successfully applied to a wide range of visual tasks, most famously to natural image object recognition [18, 29, 30], for which some claim equivalent or better than human performance. This performance comparison is often motivated by the idea that CNNs model or reproduce the early layers of the visual cortex, even though they do not incorporate many details of biological neural networks or model higher-level abstract or symbolic reasoning [13, 22, 34]. While CNN techniques were originally inspired by neuroscientific discoveries, recent advances in processing larger datasets with deeper networks have been the direct results of engineering efforts. Throughout this significant advancement, researchers have aimed to understand why and how CNNs produce such high performance [10, 28], with recent works targeting the systematic evaluation of the visual perception limits of CNNs [16, 26].

One fundamental application of human vision is to understand data visualizations. This is a task unlike natural image processing but includes the abstraction of real-world objects and their effects into data, represented with visual marks. As a field, visualization catalogues and evaluates human perception of these marks, such as in the seminal *graphical perception* experiments of Cleveland and McGill [7]. This work describes nine elementary perceptual reasoning tasks, such as position relative to a scale, length, angle, area, and shading density, plus orders their reasoning difficulty. But, with increasing research interest in the machine analysis of graphs, charts, and visual encodings, it seems pertinent to question whether CNNs are able to process these basic graphical elements and derive useful measurements from the building blocks of information visualization.

As such, we reproduce Cleveland and McGill's human perceptual experiments with CNNs, and discuss to what extent they have 'graphical perception'. To perform this evaluation, we parametrize the elementary perceptual tasks and experiments suggested by Cleveland and McGill [7], and define a set of regression tasks to estimate continuous variables. Against human perception, we pit four neural networks: a three-layer multilayer perceptron (MLP), the LeNet 2-layer CNN [21], the VGG 16-layer CNN [29], and the Xception 36-layer CNN [6]. As CNNs trained on natural images are said to mimic layers of the human visual cortex, we investigate whether using weights trained on natural images (via ImageNet [19]) or weights trained from scratch on elementary graphical perception tasks produces more accurate measurements and greater generalization.

- *Daniel Haehn, and Hanspeter Pfister are with the Paulson School of Engineering and Applied Sciences at Harvard University. E-mail: {haehn,pfister}@seas.harvard.edu.*
- *James Tompkin is with Brown University. E-mail: james_tompkin@brown.edu.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

We test these four networks across four scenarios presented by Cleveland and McGill [7]: 1) Nine elementary perceptual tasks with increasing parameteric complexity, e.g., length estimation with fixed x, then with varying x, then with varying width, including cross-network evaluations testing the generalizability of networks to unseen parameters; 2) The position-angle experiment, which compares judgements of bar charts to pie charts, 3) The position-length experiment, which compares grouped and divided bar charts, and 4) The bars and framed rectangles experiments, where visual cues aid ratio perception. We also investigate whether our CNNs can detect a proportional change in a measurement across scales, in relation to Weber's law.

With these experiments, we describe a ranking defining the ease with which our tested CNN architectures can estimate elementary perceptual tasks, as an equivalent to Cleveland and McGill's ranking for human perception. Further, we discuss the implications of our results and derive recommendations for the use of CNNs in perceiving visualizations. We accompany this paper with open source code and our input and results data, both to enable reproduction studies and to spur new machine perception systems more adept at graphical perception: `http://rhoana.org/perception`

2 Previous Work

Graphical Perception. Cleveland and McGill [7, 8] coin the phrase *graphical perception* to describe how different visual attributes and encodings are perceived by humans. They define *elementary perceptual tasks* as mental-visual stimuli to understand encodings in visualizations, and declare a ranking based on their perceptual difficulty. From these definitions, the authors propose and perform different experiments such as the *position-angle* experiment which compares bar charts and pie charts, the *position-length* experiment where users judge relations between encoded values in grouped and divided bar charts, and the *bars-and-framed-rectangles* experiment to evaluate Weber's law [12] using the proportional relation between an initial distribution density and perceivable change.

Heer and Bostock later reproduced the Cleveland-McGill experiments via crowd-sourcing on Mechanical Turk [14], with similar results. Harrison *et al.* [11] repeated the Cleveland-McGill experiments while observing viewer emotional states, again with similar results. Our experimental setup again reproduces the Cleveland-McGill experiments, but instead of judging human perception, we judge machine perception using convolutional neural networks. While we focus on Cleveland and McGill's work from the mid 1980s due to their repeated reproduction, many other works also investigate human perception to visual encoding [2, 5, 23, 24, 31–33].

Computational Visualization Understanding. Pineo *et al.* [25] create a computational model of human vision based on neural networks. Their simulations show that understanding visualization triggers neural activity in high-level areas of cognition, with the authors suspecting that this activity is supported by low-level neurons performing elementary perceptional tasks. Other work tries to parse infographics by finding higher-level saliency models [4], or by extracting text or key visual elements from visualizations [3, 17, 27]. However, these works do not investigate computational understanding of elementary perceptual tasks such as curvature, lengths, or position, which are the building blocks of visualization.

Visual-cortex-inspired Machine Learning. The human visual cortex allows us to recognize objects in the world seemingly without effort (though few remember their infancy). This visual system is organized in layers, which inspired computational classifiers based on multilayer neural networks. Fukushima and Miyake developed the early Neocognitron quantitative model [9], which ultimately led to the work of Hinton, Bengio, and LeCun [20] and today's GPU-powered *deep* neural networks. Such networks exist with many architectures. For this paper, we compare a set of networks with different architectures and depths, plus networks with weights trained on natural images and on the elementary perceptual reasoning tasks themselves.

3 Experimental Setup

First, we describe the commonalities across all of our experiments. We measure how different convolutional neural networks perceive low-level visual encodings, such as positions, angles, curvatures, and lengths. We formulate these measurement tasks as logistic regression problems: given a stimuli image of an elementary visualization, the networks must estimate the single quantity present or the ratio between multiple quantities present.

For each experiment, we use a single factor between-subject design, with the factor being the network used. This lets us evaluate whether different network designs are competitive against existing human perception results. We train each network in a supervised fashion with a mean-squared error (MSE) loss between the ground-truth labels and the network's estimate of the measurement from observing the generated stimuli images. Then, we test each network's ability to generalize to new examples with a separate data, created using the same stimuli generator function but with unseen ground-truth measurements (Section 3.2).

3.1 Networks

Multilayer Perceptron. As a baseline, we use a multilayer perceptron (MLP), but without the prior convolutional layers as is typical in network designs for solving visual tasks (Fig. 2). Our MLP contains a layer of 256 perceptrons, which are activated as rectified linear units (ReLU) (Fig. 2). We train this layer with dropout (probability $= 0.5$) to prevent overfitting, and then combine these ReLU units to regress our output measurement.

Convolutional Neural Networks. We compare different convolutional neural networks (CNNs) with both 'trained from scratch' weights and pre-trained weights on a database of natural images (1000-class ImageNet [19]). These networks are the traditional LeNet-5 with 2 layers, which was designed to recognize hand-written digits [21]; the VGG19 network with 19 layers, which was designed to solve the ImageNet object recognition challenge [29]; and the Xception network with 36 layers [6], which was also designed to solve the ImageNet object recognition challenge plus the 15,000-class JFT object recognition challenge [15]. Each of these networks has as its last layers an MLP architecture equivalent to our baseline, and so they act as earlier image and feature processors for this final regressor. Since the networks are of different architectures, the number of trainable parameters changes, with some networks having more capacity than others (Table 1).

For *VGG19* and *Xception*, we have two variants: the network trained from scratch on elementary perceptual tasks, plus the network using weights that were previously trained on the ImageNet object recognition challenge *except* for the MLP layer. This is intended to produce early-layer features which mimic human vision, and then to see whether they are more or less useful than networks trained from scratch.

Optimization. All network hyperparameters, optimization methods, and stopping conditions are fixed across networks (Table 1). We train for 1000 epochs using stochastic gradient descent with Nesterov momentum, but stop early if the loss does not decrease for ten epochs.

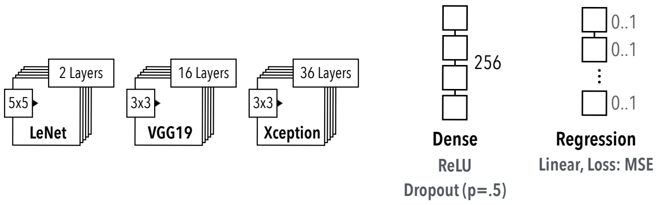
Environment. We run all experiments on Tesla X and Tesla V100 graphical processing units. We use the KERAS framework with a TensorFlow backend to train the networks, and use the scikit-image library to generate the stimuli.

3.2 Data

Image Stimuli and Labels. We create our stimuli visualizations as 100×100 binary images, rasterized without interpolation. We write a parameterized stimuli generator for each elementary task. The number of possible parameter values differs per experiment, and we summarize these in Table 2 and Section 4.1. Before use, we scale the generated images into an unbiased range: images to the range of -0.5 to 0.5. Then, we add subtle random noise (uniformly distributed between $0-$

Table 1: **Network Training.** We use different feature generators as input to a multilayer perceptron which performs linear regression. This results in different sets of trainable parameters. As a baseline, we also train the MLP directly on the visualization images without any additional feature generation.

Network	Trainable Parameters	Optimization
MLP	2,560,513	SGD (Nesterov momentum)
LeNet + MLP	8,026,083	Learning rate: 0.0001
VGG19 + MLP	21,204,545	Momentum: 0.9
Xception + MLP	25,580,585	Batchsize: 32
		Epochs: 1000 (Early Stopping)



(a) Feature Generation

(b) Multilayer Perceptron

Fig. 2: **Network Architecture.** The multilayer perceptron (MLP) in our experiments has 256 neurons which are activated as rectified linear units (ReLU). We use Dropout regularization to prevent overfitting. As output, we perform linear regression for continuous variables. The MLP can learn to represent the visualizations directly, but we also learn features generated by LeNet (2 conv. layers, filter size 5 × 5), VGG19 (16 conv. layers, filter size 3 × 3), or Xception (36 conv. layers, filter size 3 × 3) to test different model complexities.

0.05) to each pixel to introduce variation which prevents the networks from simply 'remembering' each different image.

Each stimuli image also has an associated ground truth label representing the parameter set which generated the image, e.g., the length in pixels of a bar. As before, we scale these labels to the range of 0.0 to 1.0, which represent the maximum and minimum values that this parameter can take.

Training/Validation/Test Splits. For each task, we use 60,000 training images, 20,000 validation images, and 20,000 test images. To create these datasets, we generate stimuli from random parameters and add them to the sets until the target number is reached, while maintaining distinct (random) parameter spaces for each set to ensure that there is no leakage between training and validation/testing.

3.3 Measures and Analysis

Cross Validation. For reproducibility, we perform repeated random sub-sampling validation, also known as Monte Carlo cross-validation, during our experiments. We run every experiment seperately twelve times, and randomly select (without replacement) the 60% of our data as training data, 20% as validation, and 20% as test.

Task Accuracy. In their 1984 paper, Cleveland and McGill use the midmean logistic absolute error metric (*MLAE*) to measure perception accuracy. To allow comparison between their human results and our machine results, we also use MLAE as a presentation metric:

$$\text{MLAE} = log_2(|\text{predicted percent} - \text{true percent}| + .125) \qquad (1)$$

In addition to this metric, we also calculate standard error metrics such as the mean squared error (*MSE*) and the mean absolute error (*MAE*). This allows a more direct comparison of percent errors. Please note that our networks were trained using MSE loss and not directly with MLAE.

Task Confidence Intervals. We follow Cleveland and McGill and present 95% confidence intervals. We approximate the value of the 97.5 percentile point of the normal distribution for simplicity with 1.96 as suggested by the central limit theorem [1].

Confirmatory Data Analysis. To accept or reject our hypotheses, we analyze dependent variables using analysis of variance (ANOVA) followed by parametric tests. JT: Which tests?

Training Efficiency. We use the training convergence rate as a measure of how easy or hard a particular task is for the network to learn to solve. This is defined as the MSE loss decrease per training epoch, which is an indicator of the training efficiency of the network with respect to the visual encoding.

Network Generalizability. We evaluate generalizability by asking a network previously trained upon one task parameterization to answer questions about the same type of task stimuli but with more complex parameterization, e.g., estimating bar length without and with changes in stroke width.

Further, some experiments compare different visual encoding types, e.g., bar plot vs. stacked bar plot. We train and evaluate individual networks for each parameterization, plus we also train and evaluate a networks on stimuli across the different types. This single decision-making software better mimics the judgements that a human would be able to make.

4 Experiment: Elementary Perceptual Tasks

Cleveland and McGill describe the mapping of graphical elements to quantitative variables as *elementary perceptual tasks* and introduce a list of ten different encodings in their paper [7, 8]. These tasks are the low-level building blocks for information visualizations and encode quantities. Cleveland and McGill did not explicitly test human perception of single instances of these encodings. However, we test how each classifier measures encoded values using the elementary perceptual tasks and create visualizations of these tasks as rasterized images with different parametrizations (Table 2).

4.1 Parametrizations

We generate multiple parameterizations for each elementary perceptual task and sequentially increase the number of parameters (Table 2). For instance, for *Position Common Scale* we first only vary the *y*-position which yields just 60 different parameters. We then include translation along the x-axis with a significant increase in variability. We then also add a variable spot size. This results in more complex datasets depending on the increase of variability. Table 2 shows the different settings. It is important to consider this variability when evaluating different classifiers with individual trainable parameters (Table 1). In theory, classifiers can memorize the images if the data set has a low variability. We also counteract such behavior by adding noise.

4.2 Hypotheses

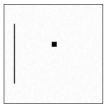
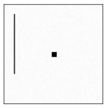
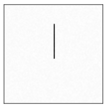
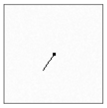
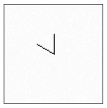
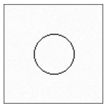
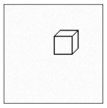
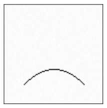
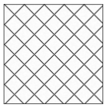
We proposed four hypotheses entering the elementary perceptual task experiment:

- **H1.1 Our convolutional neural networks are able to regress quantitative variables from graphical elements.** We parametrize different visual encodings (Table 2) and test whether the CNNs can measure them with an accuracy similar to humans.

- **H1.2 Computed perceptual performance is dependent on network architexture.** We evaluate multiple regressors with different numbers of trainable parameters. A more complex network (with a higher number of parameters) might perform better on elementary perceptual tasks.

- **H1.3 Some visual encodings are better than others for computations.** Cleveland and McGill order the elementary perceptual tasks by accuracy. We investigate whether this order is also relevant for computing graphical perception.

Table 2: **Elementary Perceptual Tasks.** Rasterized visualizations of the elementary perceptual tasks as defined by Cleveland and McGill [7] (color saturation excluded). We sequentially increase the number of parameters (e.g. by adding translation) for every task. This introduces variability and creates increasingly more complex datasets.

Elementary Perceptual Task		Permutations
	*Position Common Scale*	
	Position Y	60
	+ Position X	3,600
	+ Spot Size	216,00
	*Position Non-Aligned Scale*	
	Position Y	600
	+ Position X	36,000
	+ Spot Size	216,000
	*Length*	
	Length	60
	+ Position Y	2,400
	+ Position X	144,000
	+ Width	864,000
	*Direction*	
	Angle	360
	+ Position Y	21,600
	+ Position X	1,296,000
	*Angle*	
	Angle	90
	+ Position Y	5,400
	+ Position X	324,000
	*Area*	
	Radius	40
	+ Position Y	800
	+ Position X	16,000
	*Volume*	
	Cube Sidelength	20
	+ Position Y	400
	+ Position X	8,000
	*Curvature*	
	Midpoint Curvature	80
	+ Position Y	1,600
	+ Position X	64,000
	*Shading*	
	Density	100
	+ Position Y	2,000
	+ Position X	40,000

- **H1.4 Networks trained on perceptual tasks can generalize to more or less complex variations of the same task.** Recent research suggests that convolutional neural networks are able to generalize by interpolating between different datapoints. While the underlying reasons are mainly yet unknown, this property allows them to perform on variations of a similar perceptual task. We create visual representations of the elementary perceptual tasks with different variability (Table 2). We suspect that such variations are not a problem for successfully regressing the quantitative variables and that the networks are able to generalize.

4.3 Results

The results of the elementary perceptual tasks experiment demonstrates that our networks are able to measure visually encoded quantities using low-level building blocks (Fig. **??**). While the performance varies for the different encodings, we observe for all networks performance similar to the measured human performance by Cleveland and McGill [8]. These results confirm our initial hypotheses.

Accuracy. We report the average error across all tasks for our networks as *MLAE*: 2.943 ($SD = 0.857$) for the multilayer perceptron, for LeNet 2.125 ($SD = 0.38$), VGG19 on ImageNet 0.979 ($SD = 0.581$), for VGG19 trained from scratch 0.404 ($SD = 0.407$), Xception on ImageNet 1.627 ($SD = 0.462$), and for the re-trained Xception 1.504 ($SD = 0.493$). The average error for all classifiers is *MLAE*= 1.597 ($SD = 0.394$) and *MAE*=2.89 ($SD = 0.848$). VGG19 trained from scratch reaches the best performance overall. These values are in the same ballpark as the range of $8 - 14\%$ error (approx. MLAE=3) for a similar experiment by Cleveland and McGill testing human estimation of relations between elementary perceptual tasks [8]. We therefor carefully confirm that our networks are able to regress the graphical elements to quantities within a reasonable range of error and **accept H1.1**.

Regressor Complexity. The performance varies not only across different visual encodings but also very much across the different network architectures. We compare the average regression performances for our networks and report the effect of the regressor as statistical significant ($F_{5,48} = 20.392, p < 0.01$). Post hoc comparisons show that the differences between LeNet and the VGG19 network, independent of the used weights, are significant ($t_48 = 4.674, p < 0.01$). VGG19 from scratch and Xception (both versions) are significant, from scratch ($t_48 = 4.87, p < 0.01$) and for imagenet weights($t_48 = 5.621, p < 0.01$). However, differences between LeNet and both Xception networks are not significant. In any case, the variation of results also visually inspectable in Fig. 3 shows the variation of performance of the different networks. We therefor **partially accept H1.2**.

Ranking of Visual Encodings. Cleveland and McGill provide an ordering of elementary visual encodings based on theoretical arguments and experimental results. We re-create their ranking with MLAE values of our networks in Table 3. The rankings differ between the different networks and Cleveland and McGill's version. However, the ranking between networks using imagenet weights is identical. Since such a ranking see,s to heavily depend on the used regressor, the weights, and the initialization, we do not think a generalized ranking can be created and **reject H1.3**.

Table 3: **Ranking of Elementary Perceptual Tasks.** We create a ranking of the elementary perceptual tasks for all of our networks. We report midmean logistic absolute errors (MLAE) for each network averaged across multiple runs on the most complex parametrization of the individual stimuli. The VGG19 network performs best compared on all tasks. For human performance, we report the ranking of Cleveland and McGill [7]. The VGG19 * and Xception * networks use ImageNet weights and yield identical rankings.

Human	MLP	LeNet	VGG19 *	**VGG19**	Xception *	Xception
Position Common Scale						
1.	7. (3.84)	2. (1.36)	5. (1.02)	**3.** (-0.04)	5. (1.65)	1. (1.04)
Position Non aligned Scale						
2.	6. (3.61)	1. (1.35)	6. (1.09)	**5.** (0.26)	6. (1.71)	2. (1.06)
Length						
3.	1. (1.99)	8. (3.19)	4. (0.87)	**2.** (-0.14)	4. (1.59)	3. (1.11)
Direction						
3.	9. (4.65)	7. (3.07)	9. (2.84)	**8.** (0.92)	9. (3.46)	6. (1.57)
Angle						
3.	8. (4.61)	9. (3.33)	8. (2.31)	**9.** (0.99)	8. (2.60)	7. (1.72)
Area						
4.	2. (2.01)	5. (2.21)	1. (0.49)	**1.** (-0.17)	1. (0.80)	5. (1.38)
Volume						
5.	4. (2.38)	4. (1.91)	7. (1.16)	**7.** (0.87)	7. (2.03)	9. (2.10)
Curvature						
5.	3. (2.34)	3. (1.81)	2. (0.71)	**6.** (0.28)	2. (1.17)	4. (1.13)
Shading						
6.	5. (3.04)	6. (2.23)	3. (0.73)	**4.** (0.14)	3. (1.57)	8. (1.82)

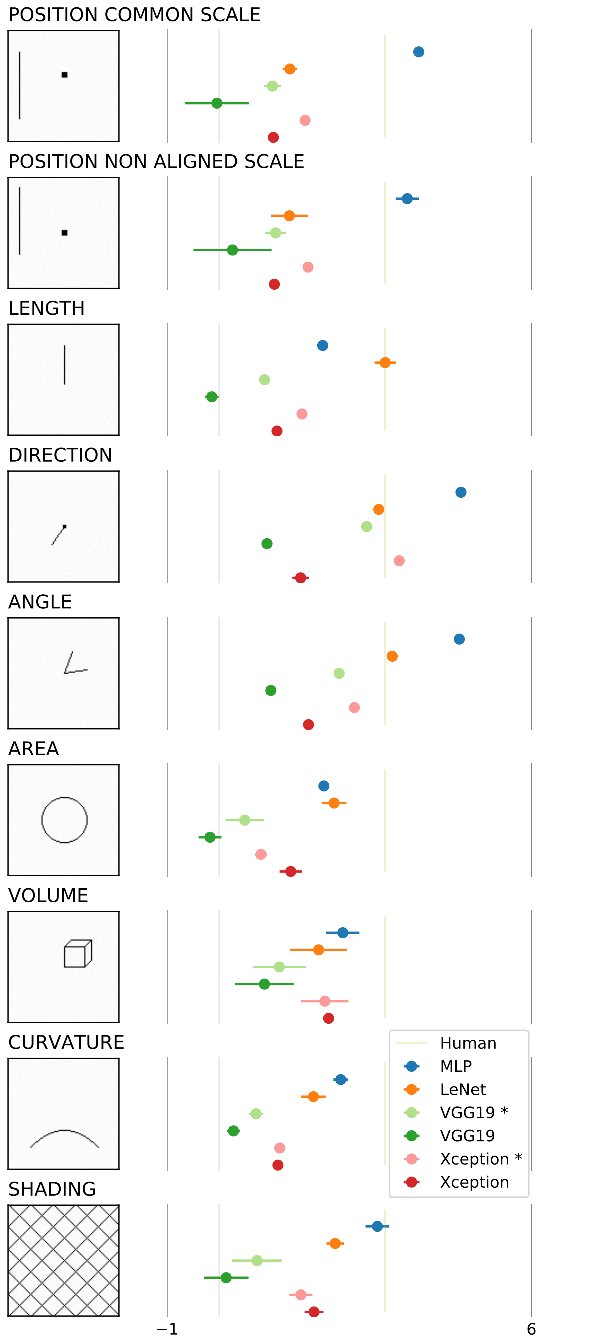


Fig. 3: **Computational results of Elementary Perceptual Tasks experiment.** Log absolute error means and 95% confidence intervals for computed perception of different classifiers on the *elementary perceptual tasks* introduced by Cleveland and McGill 1984 [7]. We test the performance of a Multi-layer Perceptron (MLP), the LeNet Convolutional Neural Network, as well as feature generation using the VGG19 and Xception networks on the most complex parametrizations. The * indicates networks which use ImageNet weights. We also compare against an approximate human baseline of 9% error reported by Cleveland and McGill.

Cross-classifier Variability and Network Generalizability. We measure regression performance across classifiers trained on the different parameterizations of the elementary perceptual tasks. Looking at the results in Fig. 4, we can observe that even slight variations of an encoding such as added translation in Y or X drastically increases the error. We think that the networks are not really learning concepts of perception but rather learn the variation of pixel values.

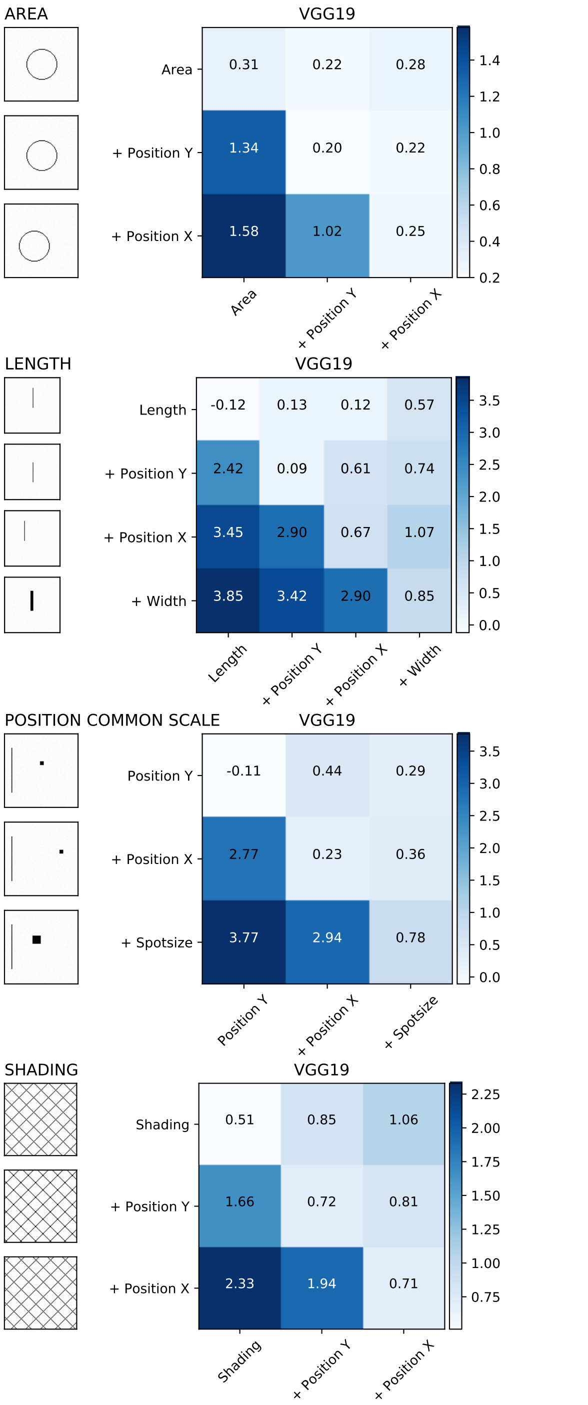
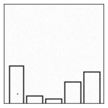
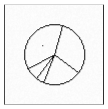


Fig. 4: **Cross-classifier variability for perceptual tasks.** We use predictions of the VGG19 network trained on different parametrizations of *area*, *length*, *position common scale*, and *shading*. These are the top 4 encodings in the ranking for this network. We measure the mean logistic absolute error (MLAE) – the lower score, the better. Regressors trained on stimuli with variable position can generalize even if the axis of translation varies. However, if they are trained on fixed positions of the stimuli they are not able to measure translated encodings.

5 Experiment: Position-Angle

The position-angle experiment was originally performed by Cleveland and McGill to measure whether humans can better perceive quantities encoded as positions or as angles [7]. The actual experiment then compares pie charts versus bar charts since these map down to elementary position and angle judgement. We create rasterized images mimicking Cleveland and McGill's proposed encoding and investigate computational perception of our four classifiers.

Table 4: **Position-Angle Experiment.** We create rasterized visualizations of pie charts and bar charts to follow Cleveland and McGill's position-angle experiment. The experimental task involves the judgement of different encoded values in comparison to the largest encoded values. The pie chart and the bar chart visualize the same data point. In their paper, Cleveland and McGill report less errors using bar charts.

		Permutations
	Type 1: *Bar Chart* Perceptual Task: *Position*	878, 520
	Type 2: *Pie Chart* Perceptual Task: *Angle*	878, 520

5.1 Hypotheses

We proposed four hypotheses entering the elementary perceptual task experiment:

- **H2.1 Computed perceptual performance is better using bar charts than pie charts.** Cleveland and McGill report that position judgements are almost twice as accurate as angle judgements. This renders bar charts superior to pie charts and should also be the case for convolutional neural networks.
- **H2.2 Classifiers can learn position faster than angles.** We assume that understanding bar charts is easier than understanding pie charts. We suspect that our classifiers learn encodings of positions faster than of angles resulting in more efficient training and faster convergence.

5.2 Results

Bar charts are more accurate (Fig. 6) and networks converge faster (Fig. 5). This is great.

6 Experiment: Position-Length

This is the one where we estimate two selected bars compared to the longest one - very similar to the previous one but, not yet done. We basically test divided versus grouped barchart and we estimate one relation between two marked quanitities: what percent the smaller is of the larger.

There are five types: type 1-3 this is a position judgement along a common scale. (btw all classifiers seem to do that extremely well in the elementary tasks so we assume this will work well here too). Types 4-5 are length judgements and we know that the classifiers struggle with that quite a bit.

The setup from Cleveland McGill is first a classification task: which one is smaller? and then a regression task: how much smaller. So we have to see how to encode this.

6.1 Hypotheses

We proposed two hypotheses entering the elementary perceptual task experiment:

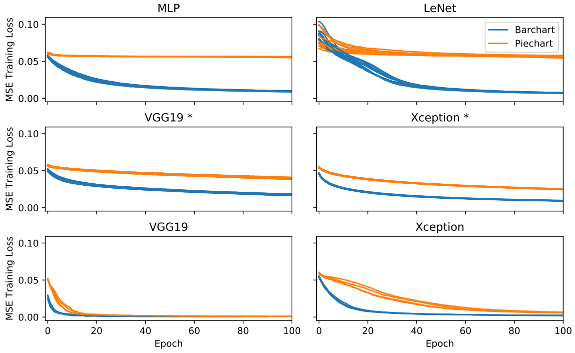


Fig. 5: **Classifier Efficiency of the Position-Angle experiment.** Mean Square Error (MSE) loss for the *position-angle experiment* as described by Cleveland and McGill [7] which compares the visualization of pie charts and bar charts. We report the MSE measure for both encodings of four different classifier on previously unseen validation data.

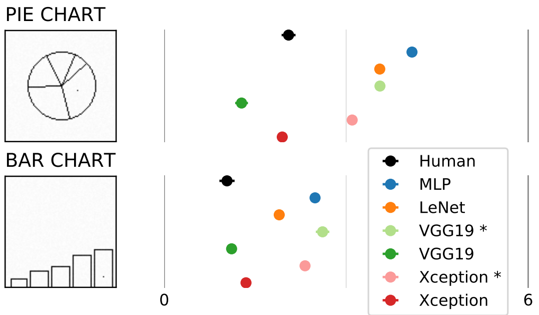


Fig. 6: **Computational results of the Position-Angle experiment.** Log absolute error means and 95% confidence intervals for the *position-angle experiment* as described by Cleveland and McGill [7]. We test the performance of a Multi-layer Perceptron (MLP), the LeNet Convolutional Neural Network, as well as feature generation using the VGG19 and Xception networks trained on ImageNet.

- **H3.1 Grouped bar charts are better computational perceivable than divided bar charts.** A grouped bar chart involves judging a position while a divided bar chart most likely (if not the bottom is looked at) requires length judgements. Classifiers are better at judging position than at judging length so grouped bar charts are easier to grasp in terms of computational perception.
- **H3.2 not yet** Any ideas?

6.2 Discussion

JT: Look at the relative difficulty of the tasks. In Cleveland and McGill, types 1-5 were post-ordered by their log error such that type 1 was easiest and type 5 was hardest. Is this still the case with our CNNs?

6.3 Results

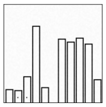
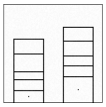
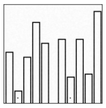
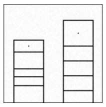
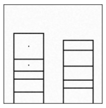
7 Experiment: Bars and Framed Rectangles

Visual cues can help converting graphical elements back to their real world variables. Cleveland and McGill introduced the bars and framed rectangles experiment which judges the elementary perceptual task of position along non-aligned scales [7].

7.1 Hypotheses

We proposed two hypotheses entering the elementary perceptual task experiment:

Table 5: **Position-Length Experiment.** Rasterized versions of the graphs of Cleveland and McGill's position-length experiment. The perceptual task involves comparing the two dot-marked quantities across five different visual encodings of either grouped or divided bar charts. We evaluate which type of bar chart performs better with our neural networks. The two marked values are chosen from a set of ten pairs which defines the dual regression task. Since the other 8 values are chosen randomly, the parameter space for images of this experiment is massive.

		Permutations
	Type 1: *Grouped Bar Chart* Perceptual Task: *Position*	$9.20E+16$
	Type 2: *Divided Bar Chart* Perceptual Task: *Position*	$9.20E+16$
	Type 3: *Grouped Bar Chart* Perceptual Task: *Position*	$9.20E+16$
	Type 4: *Divided Bar Chart* Perceptual Task: *Length*	$9.20E+16$
	Type 5: *Divided Bar Chart* Perceptual Task: *Length*	$9.20E+16$

- **H4.1 Classifiers can leverage additional visual cues.** The original bar and framed rectangle experiment shows how visual cues aid humans in mapping graphical elements to quantitative variables. This should be the same for feed-forward neural networks since they are inspired by the visual system.

- **H4.2 Weber's law can be transferred to computational perception.** Cleveland and McGill confirmed Weber's law based on the bar and framed rectangle experiment. For humans, the ability to perceive change within a distribution is proportional to the size of the initial distribution.

7.2 Weber-Fechner's Law

As identified by Cleveland and McGill, the bar and framed rectangle experiment is closely related to Weber's law. This psychophysics law states that perceivable difference within a distribution is proportional to the initial size of the distribution. Weber's law goes hand-in-hand with Fechner's law. We conduct an additional experiment based on the original illustrations of the Weber-Fechner law to investigate whether this law can be applied to computational perception of our classifiers (Fig. 7).

7.3 Results

First run indicates that framed rectangles perform better but we dont really know it yet.

8 Results and Discussion

General discussion..

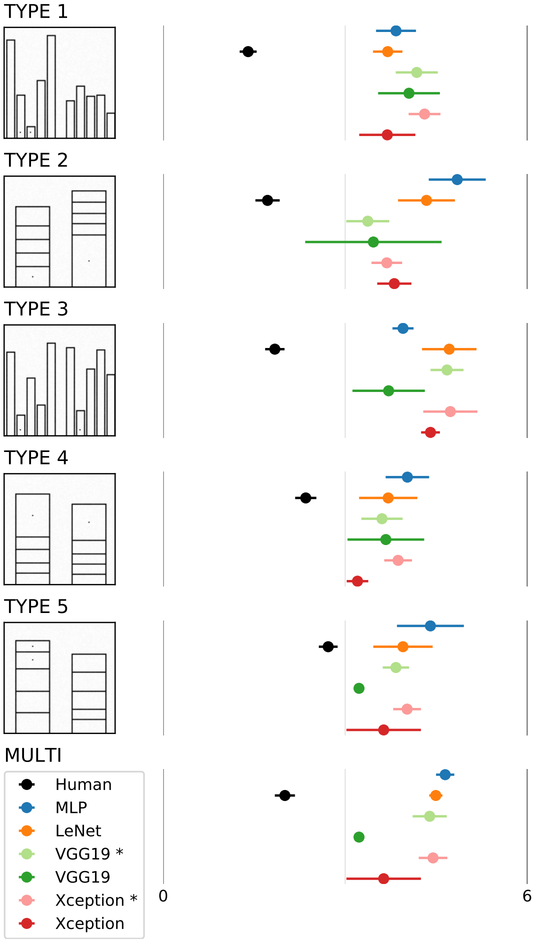


Fig. 7: **Computational results of the Position-Length experiment.** Log absolute error means and 95% confidence intervals for the *position-length experiment* as described by Cleveland and McGill [7]. We test the performance of a Multi-layer Perceptron (MLP), the LeNet Convolutional Neural Network, as well as feature generation using the VGG19 and Xception networks trained on ImageNet.

8.1 Classifiers

Transfer Learning using ImageNet. Classifiers trained on imagenet are tuned towards natural images. While VGG19 and Xception perform better than the shallower LeNet, their full performance only develops when training from scratch. This shows how natural images are truely different than infographics.

Anti-aliasing. Does it help? Not sure yet!

9 Conclusions

Future work: allow insights for infovis for machines

References

[1] I. Barany and V. H. Vu. Central limit theorems for Gaussian polytopes. *ArXiv Mathematics e-prints*, Oct. 2006.

[2] J. Bertin and M. Barbut. *Semiologie graphique: les diagrammes, les reseaux, les cartes*. Mouton, 1967.

[3] Z. Bylinskii, S. Alsheikh, S. Madan, A. Recasens, K. Zhong, H. Pfister, F. Durand, and A. Oliva. Understanding infographics through textual and visual tag prediction. *CoRR*, abs/1709.09215, 2017.

[4] Z. Bylinskii, A. Recasens, A. Borji, A. Oliva, A. Torralba, and F. Durand. Where should saliency models look next? In *European Conference on Computer Vision*, pp. 809–824. Springer, 2016.

[5] M. Carpendale. Considering visual variables as a basis for information visualisation. 2003.

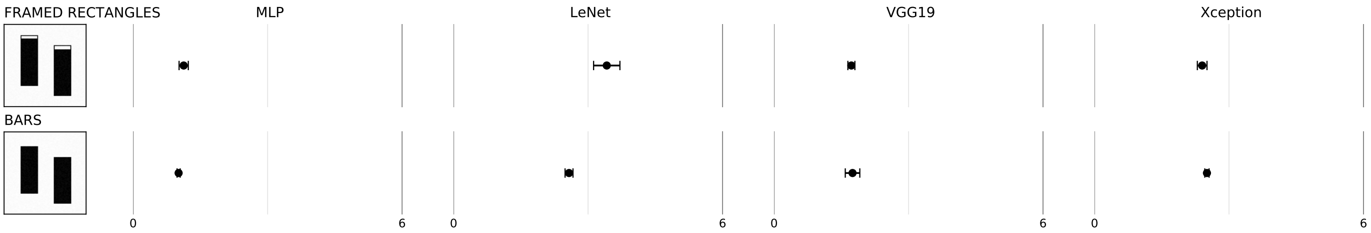


Fig. 8: **Computational results of the Bars-and-Framed-Rectangles experiment.** Log absolute error means and 95% confidence intervals for the *bars-and-framed-rectangles experiment* as described by Cleveland and McGill [7]. We test the performance of a Multi-layer Perceptron (MLP), the LeNet Convolutional Neural Network, as well as feature generation using the VGG19 and Xception networks trained on ImageNet.

Table 6: **Bars and Framed Rectangles Experiment.** Cleveland and McGill introduce the bars and framed rectangles experiment which measures the perceptual task of judging position along non-aligned scales. For humans, it is easier to decide which of two bars represent a larger height if a scale is introduced by adding framed rectangles. In this case, the right bar is heigher as visible with less free space when adding the frame. We evaluate whether such a visual aid also helps machines to perceive visually encoded quantities.

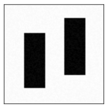
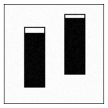
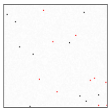
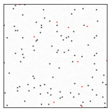
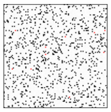
		Permutations
	*Bars* Perceptual Task: *Length* JND: $X\%$	$57,600$
	*Framed Rectangles* Perceptual Task: *Position* JND: $X\%$	$57,600$

Table 7: **Weber-Fechner Law.** The Weber-Fechner law states that the perceivable differences within a distribution is proportional to the initial size of the distribution. We create three different types of images initialized with 10, 100, and 1000 dots. We then mark randomly up to 10 dots in previously free pixels (here visualized in red). For humans, the difference is easily perceivable when the initial dot count is 10 but is hard when it is higher. We evaluate our classifiers on these rasterized visualizations.

		Permutations
	*Base 10* JND: $X\%$	$10,000$
	*Base 100* JND: $X\%$	$10,000$
	*Base 1000* JND: $X\%$	$10,000$

[6] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pp. 1800–1807. IEEE Computer Society, 2017.

[7] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical association*, 79(387):531–554, 1984.

[8] W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 1985.

[9] K. Fukushima and S. Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pp. 267–285. Springer, 1982.

[10] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[11] L. Harrison, D. Skau, S. Franconeri, A. Lu, and R. Chang. Influencing visual judgment through affective priming. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2949–2958. ACM, 2013.

[12] L. Harrison, F. Yang, S. Franconeri, and R. Chang. Ranking visualizations of correlation using weber's law. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1943–1952, Dec 2014. doi: 10.1109/TVCG.2014.2346979

[13] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

[14] J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 203–212. ACM, 2010.

[15] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015.

[16] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CoRR*, abs/1612.06890, 2016.

[17] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision*, pp. 235–251. Springer, 2016.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds., *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc., 2012.

[20] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[21] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. doi: 10.1109/5.726791

[22] D. Linsley, S. Eberhardt, T. Sharma, P. Gupta, and T. Serre. What are the visual features underlying human versus machine vision? *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pp. 2706–2714, 2017.

[23] J. Mackinlay. Applying a theory of graphical presentation to the graphic design of user interfaces. In *Proceedings of the 1st annual ACM SIGGRAPH symposium on User Interface Software*, pp. 179–189. ACM, 1988.

[24] T. Munzner. *Visualization Analysis and Design*. AK Peters Visualization Series. CRC Press, 2015.

[25] D. Pineo and C. Ware. Data visualization optimization via computational modeling of perception. *IEEE Transactions on Visualization and Computer Graphics*, 18(2):309–320, Feb 2012. doi: 10.1109/TVCG.2011.52

[26] M. Ricci, J. Kim, and T. Serre. Not-So-CLEVR: Visual Relations Strain Feedforward Neural Networks. *ArXiv e-prints*, Feb. 2018.

[27] M. J. Seo, H. Hajishirzi, A. Farhadi, and O. Etzioni. Diagram understanding in geometry questions. In *AAAI*, pp. 2831–2838. AAAI Press, 2014.

[28] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810, 2017.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[31] A. Treisman and S. Gormican. Feature analysis in early vision: evidence from search asymmetries. *Psychological review*, 95(1):15, 1988.

[32] D. Wigdor, C. Shen, C. Forlines, and R. Balakrishnan. Perception of elementary graphical elements in tabletop and multi-surface environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pp. 473–482. ACM, New York, NY, USA, 2007. doi: 10.1145/1240624.1240701

[33] L. Wilkinson. *The grammar of graphics*. Springer Science & Business Media, 2006.

[34] D. L. Yamins and J. J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356, 2016.