

Evaluating ‘Graphical Perception’ with CNNs

Daniel Haehn, James Tompkin, and Hanspeter Pfister

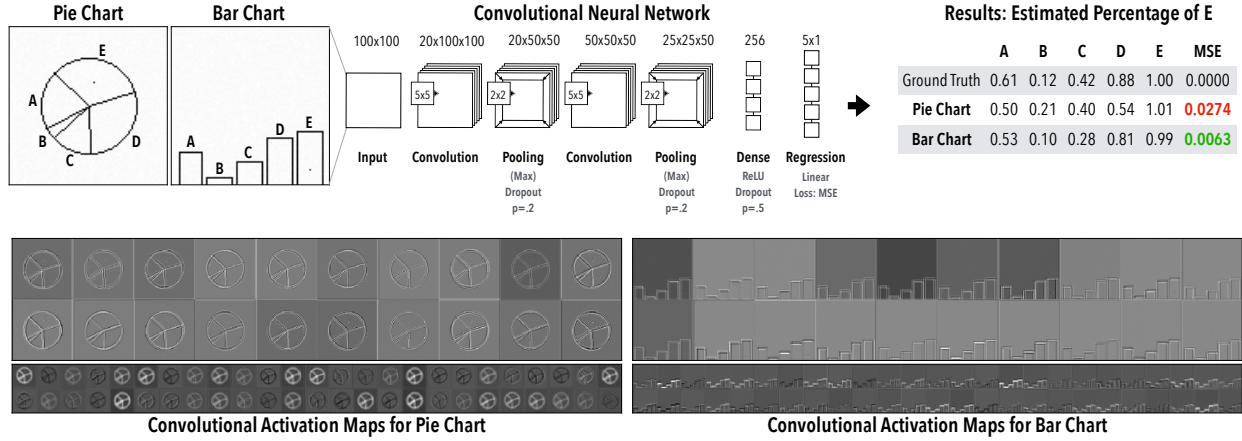


Fig. 1: **Computing Cleveland and McGill’s Position-Angle Experiment using Convolutional Neural Networks.** We replicate the original experiment by asking visual cortex inspired machine learning classifiers to assess the relationship between values encoded in pie charts and bar charts. Similar to the findings of Cleveland and McGill [6], 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 [16, 27, 28], 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 [12, 20, 32]. 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 [9, 26], with recent works targeting the systematic evaluation of

the visual perception limits of CNNs [14, 24].

One fundamental application of human vision is to understand data visualizations. This is a task unlike processing natural images, where real-world objects and their effects are abstracted into data and 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 [6]. 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 [6], 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 5-layer CNN [19], the VGG 19-layer CNN [27], and the Xception 36-layer CNN [5]. 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 [17]) or weights trained from scratch on elementary graphical perception tasks produces more accurate measurements

- 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 the Thomas J. Watson Sr. Center for Information Technology at 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.xxx/TVCG.201x.xxxxxxx

and greater generalization.

We test these four networks across four scenarios presented by Cleveland and McGill [6]: 1) Nine elementary perceptual tasks with increasing parametric complexity, e.g., estimating the length of a bar 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 judgments 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 CNNs obey Weber’s law, which defines a proportional dependency between an initial distribution and perceivable change.

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 data¹, both to enable reproduction studies and to spur new machine perception systems more adept at graphical perception.

2 PREVIOUS WORK

Graphical Perception Cleveland and McGill [6,7] 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 [11] 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 [13], with similar results. Harrison *et al.* [10] 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 [1, 4, 21, 22, 29–31].

Computational Visualization Understanding Pineo *et al.* [23] 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 perceptual tasks. Other work tries to parse infographics by finding higher-level saliency models [3], or by extracting text or key visual elements from visualizations [2, 15, 25]. 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 [8], which ultimately led to the work of Hinton, Bengio, and LeCun [18] and today’s GPU-powered *deep* neural networks. Such networks exist with many architectures. For this paper, we select the traditional LeNet-5 which was designed to recognize hand-written digits with 5 layers [19], the VGG19 network

with 16 convolutional layers [27], and the Xception network with 36 convolutional layers [5]. This allows us to compare architectures with different depths.

3 EXPERIMENTAL SETUP

We conduct quantitative experiments to measure how different convolutional neural networks perceive low-level visual encodings, such as positions, angles, curvatures, and lengths. We formulate measurement tasks as logistic regression problems, where the networks must estimate single quantities or ratio relations between multiple quantities. The output between 0 and 1 indicates the percent of the value to be measured in the individual experiment.

As baseline, we use a standard multilayer perceptron (MLP) without the generation of additional feature maps through convolutions (Fig. 2). For our experiments, we use a series of single factor between-subject designs with the factor *network* (MLP, LeNet, VGG19, Xception, VGG19+ImageNet, Xception+ImageNet). We train each network in a supervised fashion, with each experiment including a training and a testing phase on data. without any overlap (Section 3.3). All parameters such as network hyperparameters, optimization methods, and stopping conditions are fixed (Section 3.2). However, since the networks are of different complexity, the number of trainable parameters changes (Table 1). We define a series of hypotheses prior to each experiment.

3.1 Measures and Analysis

Accuracy. In their 1984 paper, Cleveland and McGill use the midmean logistic absolute error metric (*MLAE*) to measure perception accuracy.

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

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 allow a more direct comparison of percent errors.

Efficiency. We use the convergence rate based on the decrease of loss per training epoch as an indicator for the efficiency of the classifier in combination with a visual encoding. For regression tasks the loss is defined as MSE.

Confidence Intervals. We follow the notion of Cleveland and McGill to compute the 95% confidence intervals but rather than performing bootstrapping, we approximate the value of the 97.5 percentile point of the normal distribution for simplicity.

Confirmatory Data Analysis. To accept or reject our hypotheses, we calculate these measures and treat them as continuous variables. We analyze these dependent variables using analysis of variance (ANOVA) followed by parametric tests.

3.2 Classifiers

Our classifiers are built upon a multilayer perceptron (MLP) which is a feedforward artificial neural network. We combine this MLP with different convolutional neural networks (CNNs) for preprocessing and feature generation. These include the traditional LeNet-5 network as well as the VGG19 and Xception networks.

Multilayer Perceptron. The multilayer perceptron in this paper has 256 neurons which are activated as rectified linear units (Fig. 2). We then add a dropout layer to prevent overfitting and compute linear regression with MSE loss.

Convolutional Neural Networks. We use CNNs to generate additional features as input to the MLP. We train the *LeNet* classifier, the *VGG19* network, and the *Xception* model specifically towards each visualization. For *VGG19* and *Xception*, we additionally generate features using previously trained weights on ImageNet to naively model human vision.

Optimization. All networks are optimized using stochastic

¹Code and data are available at: <http://rhoana.org/perception>

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

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

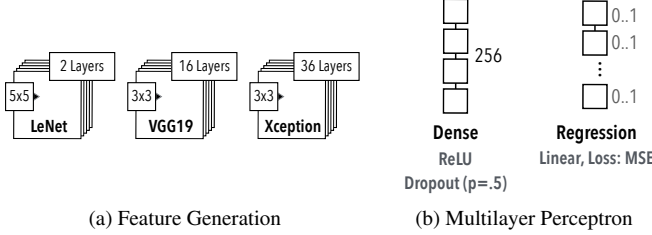


Fig. 2: **Classifier Architecture.** The multilayer perceptron (MLP) in our experiments has 256 neurons which are activated as rectified linear units (ReLU). We use Dropout regularization to prevent overfitting. As output, we perform linear regression for continuous variables. The MLP can learn the visualizations 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.

gradient descent with Nesterov momentum using fixed parameters (Table 1). We train for 1000 epochs 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 tensorflow and leverage the scikit-image and scikit-learn libraries.

3.3 Data

We create all visualizations as parametrized rasterized images without interpolation (except for the anti-alias experiment). The number of parameters differs per experiment as summarized in Table 2 and section 4.1.

Noise. We add subtle random noise (0.05) to each pixel to introduce additional variation. This prevents the classifiers of learning the actual pixels.

Training/Validation/Test Splits. We specify the size of each split set as follows: 60,000 training images, 20,000 validation images, and 20,000 test images. We then randomly add parametrized visualizations to the sets while guaranteeing that each set is disjunct from each other in terms of encoded variables. This eliminates leakage during training and evaluation. We also scale each set independently: images to the range of -0.5 to 0.5 and labels to the range of 0.0 to 1.0 .

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 separately multiple times (at least 4, up to 12) and randomly select (without replacement) the 60% of our data as training data, 20% as validation, and 20% as test. Our large data sample size of 100,000 guarantees that every single observation of our parametrizations will be selected at least once (excluding noise patterns). Finally, we average

the results over the runs.

Intra-classifier Variability. We also evaluate classifiers previously trained with one visualization on the same type of visualizations with different parameters by decreasing and increasing the variability of the generated images.

Multi Classifiers. Some experiments compare different types of visual encodings. We first train and evaluate each classifier on one type of encoding but we also train the classifiers on a random selection of multiple different types. These scenarios afflict the optimization process to find global minima and result in networks with more flexible knowledge with the caveat of longer training times.

4 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 [6]. 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 Visual cortex inspired classifiers are able to connect graphical elements to their quantitative variables.** While much simpler models than their biological pendant, convolutional neural networks are heavily influenced by our biological knowledge of the visual system. Such classifiers therefore follow the same principles as human perception.
- **H1.2 Computed perceptual performance is dependent on classifier complexity.** We evaluate multiple classifiers with different numbers of trainable parameters. A more complex classifier (with higher number of parameters) will 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.
- **H1.4 Classifiers trained on perceptual tasks can generalize to more or less complex variations of the same task.** Recent research suggests that convolutional neural networks generalize extremely well. While the underlying reasons are mainly yet unknown, this property allows them to perform on variations of a similar perceptual task.

4.3 Results

some are good and some are bad.. why?

Computational Perception Ranking.

Cleveland McGills Ranking - can we observe something similar?

1. Position along a common scale e.g. scatter plot
2. Position on identical but nonaligned scales e.g. multiple scatter plots
3. Length e.g. bar chart
4. Angle & Slope (tie) e.g. pie chart
5. Area e.g. bubbles
6. Volume, density, and color saturation (tie) e.g. heatmap
7. Color hue e.g. newsmag

Cross-classifier variability.

Can a neural network generalize on simple perceptual tasks?

5 POSITION-ANGLE EXPERIMENT

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 [6]. 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.

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 POSITION-LENGTH EXPERIMENT

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 bar chart and we estimate one relation between two marked quantities: 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:

- **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 BARS AND FRAMED RECTANGLES EXPERIMENT

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 [6].

7.1 Hypotheses

We proposed two hypotheses entering the elementary perceptual task experiment:

- **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. ??).

7.3 Results

First run indicates that framed rectangles perform better but we don't really know it yet.

8 RESULTS AND DISCUSSION

General discussion..

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 truly different than infographics.

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

9 CONCLUSIONS

Future work: allow insights for infovis for machines

REFERENCES

- [1] J. Bertin and M. Barbut. *Semiologie graphique: les diagrammes, les reseaux, les cartes*. Mouton, 1967.
- [2] 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.
- [3] 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.
- [4] M. Carpendale. Considering visual variables as a basis for information visualisation. 2003.
- [5] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pp. 1800–1807. IEEE Computer Society, 2017.
- [6] 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.
- [7] W. S. Cleveland and R. McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229(4716):828–833, 1985.
- [8] 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.
- [9] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [10] 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.
- [11] 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
- [12] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [19] 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
- [20] 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.
- [21] 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.
- [22] T. Munzner. *Visualization Analysis and Design*. AK Peters Visualization Series. CRC Press, 2015.
- [23] 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
- [24] M. Ricci, J. Kim, and T. Serre. Not-So-CLEVR: Visual Relations Strain Feedforward Neural Networks. *ArXiv e-prints*, Feb. 2018.
- [25] M. J. Seo, H. Hajishirzi, A. Farhadi, and O. Etzioni. Diagram understanding in geometry questions. In *AAAI*, pp. 2831–2838. AAAI Press, 2014.
- [26] R. Shwartz-Ziv and N. Tishby. Opening the black box of deep neural networks via information. *CoRR*, abs/1703.00810, 2017.
- [27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.
- [29] A. Treisman and S. Gormican. Feature analysis in early vision: evidence from search asymmetries. *Psychological review*, 95(1):15, 1988.
- [30] 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
- [31] L. Wilkinson. *The grammar of graphics*. Springer Science & Business Media, 2006.
- [32] D. L. Yamins and J. J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356, 2016.

Table 2: **Elementary Perceptual Tasks.** Rasterized visualizations of the elementary perceptual tasks as defined by Cleveland and McGill [6] (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.

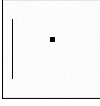
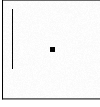
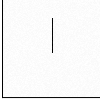
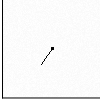
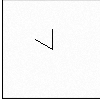
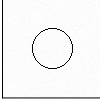
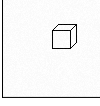

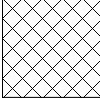
Elementary Perceptual Task	Permutations
 <i>Position Common Scale</i> Position Y + Position X + Spot Size	60 3,600 216,00
 <i>Position Non-Aligned Scale</i> Position Y + Position X + Spot Size	600 36,000 216,000
 <i>Length</i> Length + Position Y + Position X + Width	60 2,400 144,000 864,000
 <i>Direction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

Table 3: **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.

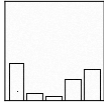
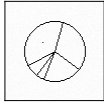
	Permutations
 Type 1: <i>Bar Chart</i> Perceptual Task: <i>Position</i>	878,520
 Type 2: <i>Pie Chart</i> Perceptual Task: <i>Angle</i>	878,520

Table 4: **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.

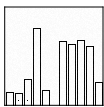
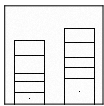
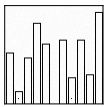
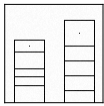
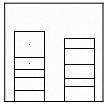
	Permutations
 Type 1: <i>Grouped Bar Chart</i> Perceptual Task: <i>Position</i>	$9.20E + 16$
 Type 2: <i>Divided Bar Chart</i> Perceptual Task: <i>Position</i>	$9.20E + 16$
 Type 3: <i>Grouped Bar Chart</i> Perceptual Task: <i>Position</i>	$9.20E + 16$
 Type 4: <i>Divided Bar Chart</i> Perceptual Task: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

Table 5: **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.

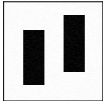
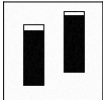
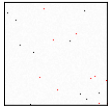
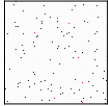
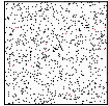
Permutations		
	<i>Bars</i> Perceptual Task: <i>Length</i> JND: <i>X%</i>	57,600
	<i>Framed Rectangles</i> Perceptual Task: <i>Position</i> JND: <i>X%</i>	57,600

Table 6: **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		
	<i>Base 10</i> JND: <i>X%</i>	10,000
	<i>Base 100</i> JND: <i>X%</i>	10,000
	<i>Base 1000</i> JND: <i>X%</i>	10,000

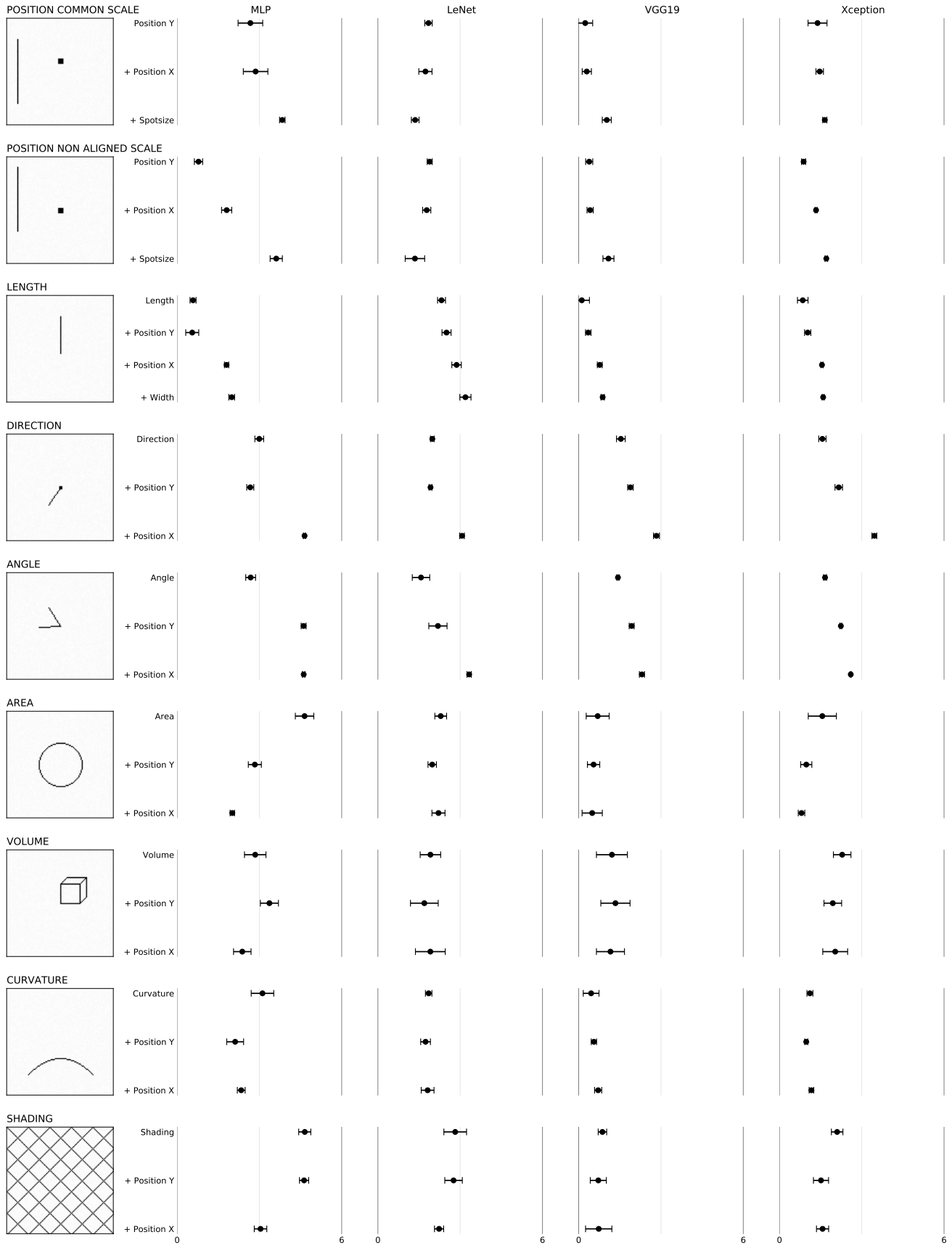


Fig. 3: **Computational results of Elementary Perceptual Tasks experiment.** Log absolute error means and 95% confidence intervals for computed perception of different classifiers on the *elementary perceptual tasks* introduced by Cleveland and McGill 1984 [6]. We test the performance of a Multi-layer Perceptron (MLP), the LeNet Convolutional Neural Network, as well as feature generation using the VGG19 and Xception networks trained on ImageNet.

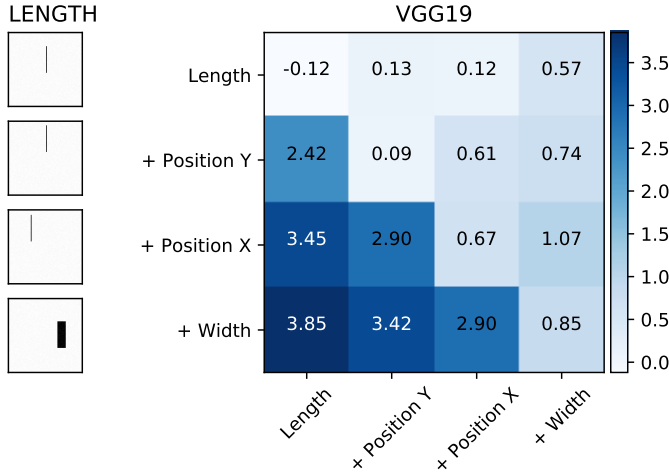


Fig. 4: **Cross-classifier variability for the perceptual task of measuring length.** We use predictions of LeNet classifiers trained on different parametrizations of the *curvature* elementary perceptual task and measure the mean logistic absolute error (MLAE). The lower score, the better. Classifiers trained on curves with variable position can generalize even if the axis of translation varies. However, classifiers trained on fixed positions of curves are not able to measure translated curves.

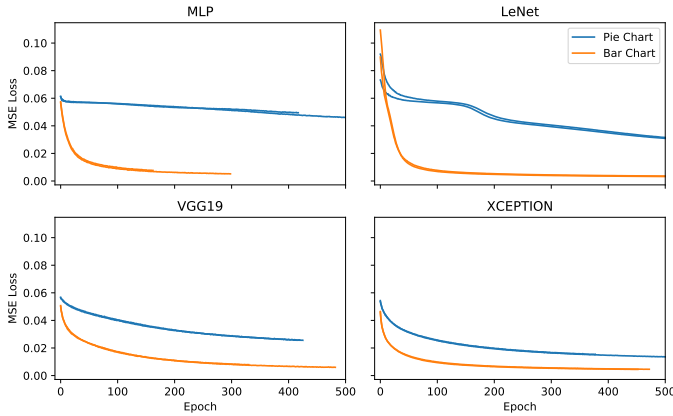


Fig. 5: **Classifier Efficiency of the Position-Angle experiment.** Mean Square Error (MSE) loss for the *position-angle* experiment as described by Cleveland and McGill [6] which compares the visualization of pie charts and bar charts. We report the MSE measure for both encodings of four different classifier on previously unseen validation data.

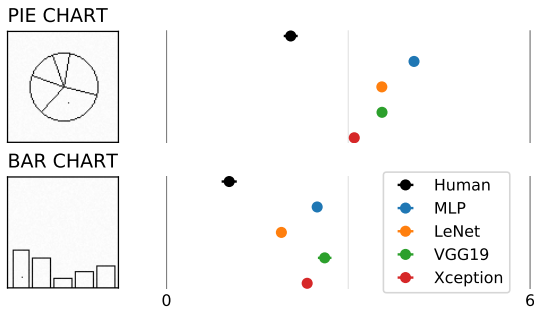


Fig. 6: **Computational results of the Position-Angle experiment.** Log absolute error means and 95% confidence intervals for the *position-angle* experiment as described by Cleveland and McGill [6]. We test the performance of a Multi-layer Perceptron (MLP), the LeNet Convolutional Neural Network, as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

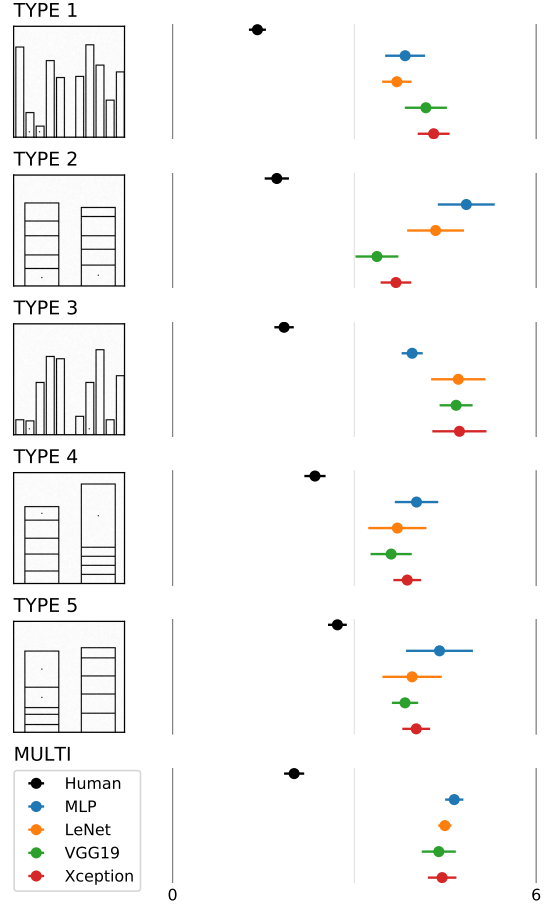


Fig. 7: **Computational results of the Position-Length experiment.** Log absolute error means and 95% confidence intervals for the *position-length* experiment as described by Cleveland and McGill [6]. We test the performance of a Multi-layer Perceptron (MLP), the LeNet Convolutional Neural Network,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

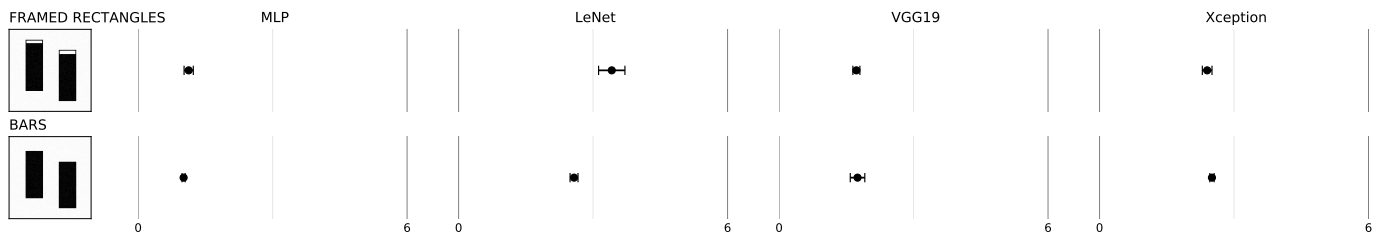


Fig. 8: **Computational results of the Bars-and-Framed-Rectangles experiment.** Log absolute error means and 95% confidence intervals for the *bars-and-framed-rectangles experiment* as described by Cleveland and McGill [6]. We test the performance of a Multi-layer Perceptron (MLP), the LeNet Convolutional Neural Network, as well as feature generation using the VGG19 and Xception networks trained on ImageNet.