Comparing pattern sensitivity of a convolutional neural network with an ideal observer and support vector machine

Fabian Reith
Brian A. Wandell
Psychology Department, Stanford University, Stanford, CA 94305, USA

Contact information:  
Fabian.H.Reith@gmail.com  
wandell@stanford.edu

Acknowledgements:  We thank David Donoho, Zhenyi Liu, Zheng Lyu, and Daniel Yamins for useful discussions.

Key words: Deep learning, Signal detection, ideal observer, convolutional neural networks, support vector machines, ResNet

Abstract

We investigate the performance of a convolutional neural network (CNN) at detecting a signal-known-exactly in Poisson noise. We compare the network's performance with that of a Bayesian ideal observer (IO) that reflects the theoretical optimum in detection performance and a linear support vector machine (SVM). For several types of stimuli, including harmonics, faces, and certain regular patterns, the CNN's performance matches the ideal. The SVM detection sensitivity is approximately 3x lower. For other stimuli, including random patterns and certain cellular automata, the CNN sensitivity is significantly worse than that of the ideal observer and the SVM sensitivity. Finally, when the signal position is randomized, so that the signal can appear in one of multiple locations, CNN sensitivity continues to match the ideal sensitivity.

Introduction

Deep convolutional neural networks - comprising a stack of computational layers connected by simple non-linearities - have become an important computational tool. The parameters of these
networks are established by training; with a sufficient number of examples, networks can be trained to detect and classify objects with an accuracy that far exceeds prior art. Much of the excitement in the field arises because the generalization can capture semantic categories, such as the texture of leather or a human face. Furthermore, region proposal networks can locate the position of objects within these semantic categories anywhere in an image (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2015b; Ren et al. 2015b).

There are many other imaging applications of convolutional and related related network learning methods, including denoising, image reconstruction, super-resolution and camera co-design (McCann, Jin, and Unser 2017; Ledig et al. 2016; Karras et al. 2017; Jain and Seung 2009; Jackson et al. 2017; Schlemper et al. 2017; Liu et al. 2019). Neural networks are also used as computational models and tools in visual neuroscience. In some cases direct comparisons are made between the network responses and neural responses (Yamins and DiCarlo 2016). In other cases, learning methods are used to discover the information present in a neural signal. In this application, we either measure or simulate neural data and use the data to train a CNN network or learn a support vector solution (Corinna Cortes and Vapnik 1995). This approach has applications in image quality (Farrell et al. 2014) and measuring the information available at different stages of the nervous system (W. S. Geisler 1989; Farrell et al. 2014; Cottaris et al. 2018).

In contrast to learning the parameters of a neural network, classical statistical methods permit one to derive the optimal formula for detecting a known signal in noise is straightforward (ideal observer, signal-known-exactly). This is the classic approach to measuring discriminability - called ideal observer theory. In many cases we cannot compute an ideal observer (IO), and for this reason, deeper networks are an important tool. But there are many cases in which we can compare a network with the IO in the hope of learning more about CNN performance. For example (Zhou, Li, and Anastasio 2019) used simple CNNs to approximate the ideal observer and the Hotelling observer (Yao and Barrett 1992). We can also compare CNN performance with another popular machine learning algorithm, the support vector machine (SVM). Such comparisons are not theorems about network performance, but they are useful estimates of how close the network performance comes with respect to the ideal.

Comparing a modern CNN with the theoretically optimal performance is valuable for assessing whether it is necessary to further develop the network to improve performance. If the CNN performs at the theoretical limit on a specific set of tasks, the research focus might shift to simplifying the networks for computation and power. We might try to establish an architecture that reaches the ideal performance using fewer training samples.

Detecting a signal in noise is an important task with applications in many fields. The types of objects of interest can be quite different, from tumors to cars. To detect a signal the CNN creates an array of internal representations of the signal class. It is quite likely that the CNN architecture has a specific range of signals where it is effective. Hence, exploring performance over a range of signals will identify the limits of the CNN architecture.
We report on a series of experiments that compare the performance of an ideal detector, an SVM, and a modern CNN (ResNet) in a range of detection and discrimination applications. We evaluate performance for detecting a pattern (signal-known-exactly) presented in Poisson noise (background-known-statistically), for discriminating between two patterns, and for detecting a pattern at one of N positions. We find that across patterns the CNN performance ranges from matching the IO to performing worse than the SVM. The detection and discrimination experiments we describe are typical of the methods used to assess imaging systems, for example in vision science, astronomy, and medical imaging (Wandell 1995; Starck and Murtagh 2001; Zhou, Li, and Anastasio 2019).

Contributions

- A modern CNN based on ResNet detects or discriminates certain spatial stimuli in the presence of Poisson noise (harmonics, faces, others) at an accuracy level that matches the classical ideal observer
- For these stimuli, the CNN requires 100,000+ training samples to reach ideal detection performance
- For other stimuli (certain textures), CNN detection and discrimination is substantially lower than optimal or even SVM performance
- In a task that identifies the location of a stimulus at one of multiple discrete locations, CNN performance can match the ideal observer

Methods

Image simulation

Harmonics and textures

The inputs to the CNN were image sensor data that were simulated as arising from a scene defined by its spatial-spectral radiance (e.g., a harmonic pattern at some contrast, frequency, phase and orientation). We modeled the scene as having a horizontal field of view of 10 deg, sampled at 512 rows and columns, and 31 wavelengths (400-700 nm with 10 nm spacing). We modeled the imaging lens modeled as diffraction limited (f/# = 4) with a focal distance of 3.9 mm. The monochrome sensor was ideal (no electronic noise) with a pixel size of 2.8 microns, approximately equal to the full-width half maximum of the diffraction limited lens (2.4 microns). In this configuration the 10 deg scene spans 238 sensor pixels and the Nyquist sampling frequency for the sensor is approximately 119 cycles/image. The sensor image data include only Poisson noise, which is the classic description of photon absorptions in an electronic device (Schottky 1918).
We implemented the sensors and optics using the open-source and freely available software, ISETCam (Farrell et al. 2003; Farrell, Catrysse, and Wandell 2012; Farrell and Wandell 2015). Signal detection performance was analyzed as input-referred: We assessed the stimulus contrast required to achieve a given performance level. Unless stated otherwise, the stimuli were presented on a uniform background with a mean level of about 300 photons. Detection was measured with respect to stimulus contrast, defined as the peak minus trough divided by twice the mean. Performance as a function of contrast was assessed using equally spaced logarithmic levels, ranging from $1e^{-5}$ to $2e^{-2}$ ($0.000010$ to $0.020$).

**Face stimuli**

Faces images were taken from the MIT-CBCL database\(^1\) (Weyrauch et al. 2004). We presented the 10 faces in this database on a uniform gray background. The faces were extracted from the images in the database and superimposed on a uniform gray background. We equated the contrast of the different faces by adjusting the images to a mean of 0.5 and a standard deviation to 0.7071. These values match the mean and standard deviation of a harmonic pattern with a contrast of one. The image was read into ISETCam where it was converted to a scene spectral radiance by assuming each pixel emitted an equal photon spectral radiance. The scene luminance was set so that the mean number of photons captured by each pixel was close to 300.

**Cellular automaton textures**

We generate complex textures that using a cellular automaton method (Wolfram 1983). We scale the scene resolution to 256x256, the resolution of the automaton we create, and slightly increase the lens field of view. This way, each pixel within the scene reaches exactly one pixel of the simulated sensor. For the textures the mean and standard deviation of the images were adjusted as we did for the face stimuli (scene radiance standard deviation of 0.7071; mean scene radiance set to create an average of 300 photons per pixel).

**Ideal observer**

The neural network was compared to an ideal observer with signal-known-exactly and background-known-statistically. The number of electrons at each position is given by a Poisson distribution (Snyder and Miller 1975), whose rate parameter $\lambda$ is equal to the intensity of the signal at each position in the image.

$$P(N) = \frac{exp(-\lambda)\lambda^N}{N!}$$

---

\(^1\) [http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html](http://cbcl.mit.edu/software-datasets/heisele/facerecognition-database.html)
The ideal observer chooses the more likely signal based on a maximum likelihood calculation. For a candidate signal in noise, \( \theta \), measured independently at each pixel, the likelihood is the product of the Poisson density scaled by the a priori likelihood of the signal.

\[
L(\theta) = P(\theta) \prod_{i=1}^{p} P(N_i|\theta)
\]

For computational simplicity it is usual to calculate the log likelihood.

\[
LL(\theta) = \log(P(\theta)) + \sum_{i=1}^{p} \log(P(N_i|\theta)) \quad \text{(Equation 1)}
\]

No training is necessary to implement the ideal observer. When there are \( N \) different signals, the system selects the most likely of these given the data. This algorithm performs optimally given the available information (Wilson S. Geisler 2011).

### Support Vector Machine

Support vector machines (SVMs) were introduced by (Corinna Cortes and Vapnik 1995) under the slightly different name ‘Support-Vector Networks’. The widely used linear SVM uses training data to learn a support vector such that the value of the inner product between this vector and a data sample decides the classification (e.g., signal vs. noise). A linear SVM separating two classes implicitly defines a hyperplane separating the two classes. To solve nonlinear classification tasks it is possible to use a nonlinear kernel, which is an extension of the dot product, as described by (Aizerman, Braverman, and Rozonoer 1964).

We use the linear support vector classifier implementation by the python library Scikit-learn (Pedregosa et al. 2011), based on the libsvm implementation (Chang and Lin 2011). The SVM classifier optimizes for the hinge loss (Rosasco et al. 2004), which finds the maximum margin classification. the SVM is optimized via a SMO-type decomposition method proposed in (Fan, Chen, and Lin 2005). We set the maximum iterations performed to 1000, unless the convergence tolerance criterion of 0.001 is reached (Chang and Lin 2011).

### Image-based resnet

We used a ResNet network architecture because of its high quality (He et al. 2015b). The ResNet comprises multiple modules that each perform a convolution, batch normalization, and nonlinear operation (rectified linear unit). The network also includes skip connections.
Figure 1. The ResNet convolutional neural network architecture. (A) The input (signal plus noise) is processed through 18 ResNet modules. These are connected in series, along with certain connections that simply pass along the module input (skip connections). The final stage is a fully connected layer that provides a classification decision – signal present or absent. Dashed skip connections imply a resizing. The text in the modules describe: N x N conv is the kernel size; the next integer is the number of kernels; /N is the spatial sub-sampling (stride). In our implementation the last fully connected layer only has 2 output classes (signal vs. noise) rather than 1000. (B) The sixteen basic modules perform a standard set of operations: convolution, batch normalization, and half wave rectification (ReLu). The batch normalization adjusts the responses from the convolutional kernels. The weights are adjusted after accumulating errors across multiple forward passes through the network (batch size).

If not declared differently, the architecture was a ResNet-18 (He et al. 2015b), which has a good trade-off between speed and accuracy for our experiments. We use the PyTorch (Paszke et al. 2017) implementation.

There were a few minor adjustments to this implementation. We changed the first convolution layer to account for the fact that the sensor data are monochrome. We also replaced the average pooling layer through the PyTorch implementation of an adaptive average pooling layer (Lin, Chen, and Yan 2013). This allows the neural network to be more flexible to variations in image size. The network weights were randomly initialized by the default PyTorch initialization method, a method known as He Initialization (He et al. 2015a). This algorithm specifically addresses rectifier nonlinearities. The last layer of the ResNet-18, a fully connected layer, is replaced to accommodate the small output dimension (binary choice).

The data consists of one scene per class that has random Poisson noise. There is no inherent limit to the epoch size and this parameter can be set arbitrarily. We used 10,000 samples to define one epoch. The batch size was 32, and we used Adam (Kingma and Ba 2014) as the gradient-based optimization function.
The initial learning rate is 1e-3 and after 10 epochs, the learning rate is decreased to 1e-4. After another 10 epochs, the CNN is trained with a learning rate of 1e-5. The network's performance is tested on 5,000 data samples. Seeds are used to ensure the same random initialization of ResNet-18 on all experiments. Training data are generated with the same, specified, seeds to initiate the random number generator.

The ResNet-18 is trained using a parallel algorithm that allowed the server to use all available GPUs. While each neural network training runs on one specific GPU, each GPU is used by running multiple training and testing experiments in parallel. On the server used for training, there are six Nvidia GK210 graphics processors. On one GPU, training ResNet-18 with 300,000 data samples, generated in real-time, takes 1:18 hours.

**Network performance**

**Metrics**

The experiments are two-class classification problems. We vary the size of the signal contrast, position, or orientation and measure classification performance of the IO, ResNet-18, and SVM. We estimate the discriminability between the two classes using d-prime (d’) (Stanislaw and Todorov 1999; Green and Swets 1988). This metric is derived from the hit (true positive) and false alarm rates. Specifically, we calculate the z-scores (inverse of the standard normal cumulative distribution) for these rates and subtract the false alarm z-score from hit rate z-score.

\[
d' = Z(hit\ rate) - Z(false\ alarm\ rate)
\]

We manage extreme hit or false alarm rates (zero errors) by a small adjustment to the hit and false alarm rates (Knoke, Burke, and Burke 1980).

\[
hit\ rate = \frac{0.5 + \sum hits}{1 + \sum hits + \sum misses}
\]

\[
false\ alarm\ rate = \frac{0.5 + \sum false\ alarms}{1 + \sum false\ alarms + \sum correct\ rejections}
\]

Without this modification, a hit rate of 100% would result in a d’ of infinity, given the false alarm rate is not at 100% at well. The modified equations for false alarm and hit rates provides a finite and only slightly biased underestimate of the true d’ (Hautus 1995).

Given the mean number of absorbed photons is known in a detection task where Poisson noise is the only type of noise, d’ can also be calculated by a formula that only requires the mean
photon absorptions of both classes. In this formula, the sum of scaled Poisson random variables is approximated with normal density (W. S. Geisler 1984).

\[ d' = \frac{\sum_{i=1}^{n} (\beta_i - \alpha_i) \ln(\beta_i/\alpha_i)}{[0.5 \sum_{i=1}^{n} (\alpha_i + \beta_i) \ln^2(\beta_i/\alpha_i)]^{1/2}} \]

Our results show that the IO d', calculated via hit and false alarm rate, matches the theoretical d'. We also calculate the sensitivity of a discriminator.

In most analyses, we calculate how d' increases as the stimulus contrast, position shift, or angle changes. This produces a curve relating performance (d') to input stimulus parameter. In many analyses we summarize network sensitivity using an input-referred measure. Specifically, we calculate the contrast level, spatial shift or orientation angle needed to achieve d'=1.5. The contrast, phase shift or angle metric is calculated by linearly interpolating the performance curve.

Size of training data
SVM and ResNet-18 performance improves as training set size increases (Figure 2). The SVM performance asymptotes at a training set size of about 10,000. The ResNet-18 performance improves but reaches asymptote - in this case the maximum theoretical performance level - when the training set reaches 100,000 to 300,000 samples.

![Figure 2. Increase in detectability (d') of a harmonic image in Poisson noise as a function of training samples. The x-axis represents the number of samples used for training. The y-axis measures the discriminability between signal and noise for a harmonic contrast of 3.2e-4 (left) and 6.3e-4 (right). Irrespective of training set size, ResNet-18 was trained for 9375 iterations with a batch size of 32.](image)

The ResNet-18 performance is significantly better than that of the SVM after 10,000 training samples, continuing to rise up to the IO level at approximately 100,000 training samples. Based
on these experiments, we used a training set size of 10,000 samples for the SVM and 300,000 samples for ResNet-18.

Results

First, we consider the detection of harmonics. We measure discriminability as a function of spatial frequency, position (phase shift) and orientation. Second, we measure signal detection based on signal size (disks of various sizes). Third, we consider a collection of biological images (faces). Fourth, we measure texture signals that are not compact in space or spatial frequency (white noise, cellular automata). Fifth, we analyze the detection performance for targets that randomly appear in one of multiple positions.

Harmonics

Contrast

Detection measured by d' of a harmonic in Poisson noise increases with contrast for the IO, ResNet-18 and SVM performance. The harmonic images simulate a conventional CMOS image sensor with no internal noise and data obtained under relatively low light conditions (mean number of photons per pixel 332.1292). The ResNet-18 can be trained to achieve a performance that closely matches the ideal observer’s performance and the SVM performance is about half a log (4x) unit less sensitive (Figure 3).
Figure 3. Comparison of detection performance ($d'$) for a harmonic presented in Poisson noise. Performance increases as a function of contrast. The ideal observer (IO) and trained ResNet-18 perform at the same level. The SVM sensitivity is about 0.5 log10 units lower.

We repeated these calculations for a range of harmonic spatial frequencies, extending to the Nyquist limit of the sensor (Figure 4). The ResNet-18 contrast sensitivity (1 over contrast threshold) matched the performance of the ideal observer closely, and the SVM was systematically less sensitive. The SVM contrast sensitivity was an average of 63.39% lower (0.44 log10 units) compared to IO. The SVM’s biggest drop in contrast sensitivity is at a harmonic frequency of 7 (63.82%), while its lowest drop is at a harmonic frequency of 100 (62.41%). ResNet-18 contrast sensitivity was only slightly lower than the IO, by an average of 2.86% (0.013 log10 units).

Figure 4. Contrast sensitivity functions of the IO, ResNet-18 and SVM for spatial frequencies up to the sensor Nyquist frequency. The sensitivity is defined as the inverse of the contrast needed to achieve discrimination performance of $d' = 1.5$. Higher contrast sensitivity means performance is reached with less contrast.
Phase shift

Next we trained the ResNet-18 to detect slight changes in position (phase) of a harmonic signal (Figure 5). The shift measured in number of pixels is the phase divided by the frequency, so that the same phase shift for a 4 cycles/image harmonic is four times smaller than that phase for a 1 cycle/image harmonic.

Figure 5. (A) Position discrimination performance for IO, SVM and ResNet-18 for a 1 cycle/image harmonic. Harmonic signal is shift is specified in phase (radians). (B) Position shift sensitivity as a function of spatial frequency needed for a discrimination performance of $d'$ = 1.5.

The phase shift sensitivity is approximately constant until a frequency near 30 cycles/image. Beyond this frequency the phase shift sensitivity declines. This decline is presumably due to the reduced contrast of the sensor image which arises because of the diffraction limited optics used as part of the sensor image computation.
Orientation

IO and ResNet-18 discriminate the orientation of harmonics equally well at all spatial frequencies (Figure 6). This sensitivity is higher for higher frequency signals, presumably because the additional cycles aid the orientation estimate.

Figure 6: Comparison of IO, ResNet-18, and SVM on orientation discrimination performance for harmonics presented in Poisson noise (A). Panel B shows discrimination sensitivity as a function of spatial frequency. The orientation sensitivity (1/radians) is the inverse of the rotation angle needed for a discrimination performance of $d' = 1.5$.

Angle sensitivity increases logarithmically with frequency, reaching a maximum at spatial frequencies with reduced contrast because of the imaging optics.

Disks

Detection sensitivity grows nearly linearly with disk radius, and thus approximately as the square root of the disk area (Figure 7). Deviations from this are present for small disks which are blurred by the optics and very large disks that span nearly the whole sensor. ResNet-18 again
approximates IO performance for all disk sizes tested, and the SVM is about half a log unit lower.

**Figure 7.** Contrast sensitivity to disks for IO, ResNet-18 and SVM. Detection performance for disks with sizes from radius 1 to radius 100. Disk contrast sensitivity is shown for a performance level of $d' = 1.5$.

**Faces**

Disks and harmonic signals are very simple patterns, in contrast to many natural objects. We measured signal detection performance on an important biological structure, the human face. We also measured sensitivity to collections of faces (Figure 8).

The ResNet-18 contrast sensitivity is similar but slightly lower than the IO sensitivity. ResNet-18 contrast sensitivity is on average 5.87% worse compared to IO. This more than double the drop in sensitivity compared to harmonic signals. The SVM performance is about 1/3rd the sensitivity of the IO and ResNet-18 network.
Textures

Next, we investigate the contrast sensitivity to texture patterns. To organize our analyses, we use cellular automata as a means of generating texture patterns (Wolfram 1983). We study class 2 rules which converge to a structured repetitive pattern, and class 3 rules which remain random (Wolfram 2016). In particular, we generate textures with four different class 2 rules and four different class 3 rules.

Class 2 cellular automata

Class 2 automata converge to a repetitive texture pattern. We suspect that the CNN might learn to use filters learn to identify such repetitive patterns. To measure detection performance, we used experiments for four class 2 automata (rules 3, 57, 76 and 78). The contrast sensitivity for these patterns is slightly higher for the IO than ResNet-18, and substantially higher than SVM (Figure 9).

Slightly worse performance is achieved for the other two automata. At rule 3, IO has a contrast sensitivity of 1213.31, while ResNet-18 reaches 861.14 and SVM achieves 438.02. IO contrast sensitivity for the rule 57 automaton is the lowest. Here, IO reaches 824.34, ResNet-18 achieves 688.76 and SVM reaches 298.32.

Compared to IO, ResNet-18 performance drops by an average of 18.18%, while SVM performance drops by an average of 63.74%.
Figure 9. Contrast sensitivity for class 2 cellular automata. ResNet-18 sensitivity is slightly lower than IO; the sensitivity of SVM is around one third of IO sensitivity. The highest sensitivity is for rule 76, followed by rule 78. The one-dimensional patterns are easiest to detect.
Class 3 automata

Class 3 automata have a complex structure, without a predictable pattern. We examine four class 3 automata (rules 22, 30, 75 and 101). The limited number of convolutional filters in ResNet-18 coupled with the complex structure appears to limit the CNN sensitivity, which drops down to the level of the SVM performance for these texture patterns.

Figure 10. Contrast sensitivity for class 3 cellular automata. The IO sensitivity is substantially higher than either ResNet-18 or SVM. In some cases, the SVM sensitivity exceeds that of the ResNet-18. Unlike the class 2 cellular automata, these textures are dense and not space-invariant.

Block randomization

A second means of producing texture patterns is to randomize the pixel positions in an existing image, say be creating and then scrambling blocks of pixels. We performed a series of experiments by block-wise scrambling the pixels in a one-cycle per image harmonic (Figure 11).
Figure 11. Performance for spatial randomization of frequency one harmonic signal. Panel (a) shows significant drop in ResNet-18 performance for randomization of all pixel locations of harmonic signal (equivalent to 1x1 block). Panel (b) displays detection sensitivity for various block sizes.

After reordering the pixels in the image, the IO performs at the same level, as expected from the computational formula (Equation 1). Similarly, the SVM adjust its critical vector and learn to detect the pattern with reordered pixels. The ResNet-18 sensitivity is substantially reduced by scrambling the pixels in the harmonic image. This scrambling creates a new texture pattern that does not repeat regularly across the image, and like the cellular automata in class 3, the ResNet-18 sensitivity is below the IO.

Column randomization

Block randomization scrambles pixels over two dimensions and decreases the ResNet-18 sensitivity compared to sensitivity for a one-cycle per image harmonic signal. By randomizing the columns, we ask whether one-dimensional randomization also reduces sensitivity. We explored this with the one-cycle harmonic, faces, and a Class 2, rule 3 automaton (Figure 12).
One-dimensional spatial randomization impacts sensitivity to certain stimuli. Randomization of columns slightly decreases ResNet-18 performance for a frequency one harmonic signal. A slightly more complex signal containing multiple faces exhibits a higher drop in performance when columns are randomized. A significant drop in performance is reached when columns of the rule 3 automaton signal are randomized.

In all cases, the column randomization reduced the ResNet-18 sensitivity. This reduction occurred both in a case when the original sensitivity was equal to the IO sensitivity (harmonic) and for the case when the ResNet-18 was somewhat lower than the IO (faces, Rule 3).

**Multiple target positions**

The ability to detect and localize a signal anywhere in a scene is one of the most important contributions of CNN technology (Ren et al. 2015a). We compare the CNN sensitivity with the ideal observer sensitivity to a simple stimulus (a Gabor patch) that might be presented at one of multiple possible locations (Figure 13). When there are $N$ different locations, the ideal observer selects the most likely of these locations, or no signal, given the image data.

Introducing position uncertainty reduces the sensitivity of both the networks and the IO. Although sensitivity declines, the ResNet-18 combined with the region proposal network continues to match the IO performance. Both methods are about half as sensitive when the target can appear in 16 locations rather than one location. The SVM sensitivity declines by a larger fraction, becoming about one-fourth as sensitive as the number of possible positions increases to 16 from one.
Summary and discussion

For many spatial patterns (harmonics, disks, faces) a ResNet-18 CNN can be trained to achieve the same contrast sensitivity as an ideal observer. Accuracy for detecting these stimuli remains comparable to ideal even when the stimulus position is uncertain. The SVM network detection sensitivity for these stimuli is substantially (2.5-3x) lower.

The CNN spatial sensitivity for other spatial patterns (cellular automata textures, block-scrambled, and column-scrambled images) is substantially lower than the ideal observer. For some of these patterns the CNN sensitivity is similar to the SVM, about 2.5-3x lower than the IO.

Across stimuli the relative sensitivity of the CNN with respect to the IO varied. On the other hand the SVM classifier had very similar relative sensitivity to the ideal observer for all stimuli.

Improving ResNet sensitivity

For the broad class of patterns the ResNet-18 sensitivity reaches the maximum sensitivity limit, and there is no possibility of improving sensitivity. Further research into CNN design might focus on two other objectives: (a) expanding the scope of spatial patterns that reach the performance limit, and (b) reducing the number of training examples required to reach this limit.
ResNet sensitivity is not invariant to block- or column-scrambling; and in the examples shown here we scrambled simple stimuli and reduced sensitivity. It follows that in some cases, say when we are trying to detect a specific texture, it may be possible to improve network sensitivity by reordering the image pixels to make the texture closer to a smooth signal.

The rate of training, requiring as many as 3e+5 samples to reach ideal observer performance, is a good topic for further research. This number of training samples may have been needed only because the network was more complex (18 modules) than required for this task and network simplification may reduce the training burden (Zhou, Li, and Anastasio 2019). In the Supplemental Materials we describe some initial experiments with other networks, but approaching this question will require a more systematic investigation.

**Detection at multiple locations**

An important value of a CNN is its ability to find signals at different locations. In our experiments comparing network and ideal observer performance across multiple positions, the region proposal network matches the IO sensitivity. Further experiments examining the sensitivity when position is unknown could be significantly expanded to include variations in the stimulus pattern, size and systematic analyses of position bias.

**Spatial contrast sensitivity of a CNN**

The ResNet-18 contrast sensitivity is lower than the IO sensitivity when the stimuli comprise fine textures that do not repeat regularly across the image. The block randomization and cellular automata examples all fit into this pattern. We hypothesize that the reduced sensitivity is due to the CNN design. The CNNs learnable weights are the values of the convolutional filters, which are relatively small and compact in image space. This formulation assumes that low level features of a signal can be summarized as local patterns that are at least partially shift-invariant. Such a decomposition may not be appropriate for center texture patterns.

In conclusion, we suggest that comparing the CNN with respect to an ideal observer is a useful way to think about the CNN’s spatial sensitivity. It is useful to know that the ResNet-18 CNN reaches ideal performance for certain stimuli, but not others. It would be useful to develop a method that predicts the spatial sensitivity profile of this - or any - CNN.
References

Aizerman, M. A., E. A. Braverman, and L. Rozonoer. 1964. “Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning.” In Automation and Remote Control, 821–37. Automation and Remote Control.

Chang, Chih-Chung, and Chih-Jen Lin. 2011. “LIBSVM: A Library for Support Vector Machines.” ACM Trans. Intell. Syst. Technol. 2 (3): 27:1–27:27.

Cortes, Corinna, and Vladimir Vapnik. 1995. “Support-Vector Networks.” Machine Learning 20 (3): 273–97.

Cottaris, Nicolas P., Haomiao Jiang, Xiaomao Ding, Brian A. Wandell, and David H. Brainard. 2018. “A Computational Observer Model of Spatial Contrast Sensitivity: Effects of Wavefront-Based Optics, Cone Mosaic Structure, and Inference Engine.” bioRxiv. https://doi.org/10.1101/378323.

Fan, Rong-En, Pai-Hsuen Chen, and Chih-Jen Lin. 2005. “Working Set Selection Using Second Order Information for Training Support Vector Machines.” Journal of Machine Learning Research: JMLR 6 (Dec): 1889–1918.

Farrell, Joyce E., Peter B. Catrysse, and Brian A. Wandell. 2012. “Digital Camera Simulation.” Applied Optics 51 (4): A80–90.

Farrell, Joyce E., Haomiao Jiang, Jonathan Winawer, David H. Brainard, and Brian A. Wandell. 2014. “27.2: Distinguished Paper: Modeling Visible Differences: The Computational Observer Model.” SID Symposium Digest of Technical Papers 45 (1): 352–56.

Farrell, Joyce E., and Brian A. Wandell, eds. 2015. “Image Systems Simulation.” In Handbook of Digital Imaging, 15:1–28. Chichester, UK: John Wiley & Sons, Ltd.

Geisler, Wilson S. 2011. “Contributions of Ideal Observer Theory to Vision Research.” Vision Research 51 (7): 771–81.

Geisler, W. S. 1984. “Physical Limits of Acuity and Hyperacuity.” Journal of the Optical Society of America. A, Optics and Image Science 1 (7): 775–82.

———. 1989. “Sequential Ideal-Observer Analysis of Visual Discriminations.” Psychological Review 96 (2): 267–314.

Green, David Marvin, and John A. Swets. 1988. Signal Detection Theory and Psychophysics. Peninsula Pub.

Hautus, Michael J. 1995. “Corrections for Extreme Proportions and Their Biasing Effects on Estimated Values Ofd”. Behavior Research Methods, Instruments, & Computers: A Journal of the Psychonomic Society, Inc 27 (1): 46–51.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015a. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.” arXiv [cs.CV], arXiv. http://arxiv.org/abs/1502.01852.

———. 2015b. “Deep Residual Learning for Image Recognition.” arXiv [cs.CV], arXiv. http://arxiv.org/abs/1512.03385.

Jackson, Aaron S., Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. 2017. “Large Pose 3D Face Reconstruction from a Single Image via Direct Volumetric CNN Regression.” In Proceedings of the IEEE International Conference on Computer Vision, 1031–39.
openaccess.thecvf.com.

Jain, Viren, and Sebastian Seung. 2009. “Natural Image Denoising with Convolutional Networks.” In Advances in Neural Information Processing Systems 21, edited by D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, 769–76. Curran Associates, Inc.

Karras, Tero, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. “Progressive Growing of GANs for Improved Quality, Stability, and Variation.” arXiv [cs.NE]. arXiv. http://arxiv.org/abs/1710.10196.

Kingma, Diederik P., and Jimmy Ba. 2014. “Adam: A Method for Stochastic Optimization.” arXiv [cs.LG]. arXiv. http://arxiv.org/abs/1412.6980.

Knoke, David, Peter J. Burke, and Peter John Burke. 1980. Log-Linear Models. SAGE.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2012. “ImageNet Classification with Deep Convolutional Neural Networks.” In Advances in Neural Information Processing Systems 25, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, 1097–1105. Curran Associates, Inc.

Ledig, Christian, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, et al. 2016. “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.” arXiv [cs.CV]. arXiv. http://arxiv.org/abs/1609.04802.

Lin, Min, Qiang Chen, and Shuicheng Yan. 2013. “Network In Network.” arXiv [cs.NE]. arXiv. http://arxiv.org/abs/1312.4400.

Liu, Zhenyi, Minghao Shen, Jiaqi Zhang, Shuangting Liu, Henryk Blasinski, Trisha Lian, and Brian Wandell. 2019. “A System for Generating Complex Physically Accurate Sensor Images for Automotive Applications.” arXiv [cs.CV]. arXiv. http://arxiv.org/abs/1902.04258.

McCann, M. T., K. H. Jin, and M. Unser. 2017. “Convolutional Neural Networks for Inverse Problems in Imaging: A Review.” IEEE Signal Processing Magazine 34 (6): 85–95.

Paszke, Adam, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. “Automatic Differentiation in PyTorch,” October. https://openreview.net/forum?id=BJJsrmfCZ.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. “Scikit-Learn: Machine Learning in Python.” Journal of Machine Learning Research: JMLR 12 (Oct): 2825–30.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2015a. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” In Advances in Neural Information Processing Systems 28, edited by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, 91–99. Curran Associates, Inc.

———. 2015b. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” arXiv [cs.CV]. arXiv. http://arxiv.org/abs/1506.01497.

Rosasco, L., E. D. Vito, A. Caponnetto, and M. Piana. 2004. “Are Loss Functions All the Same?” Neural Computation. https://www.mitpressjournals.org/doi/abs/10.1162/089976604773135104.

Schlemper, Jo, Jose Caballero, Joseph V. Hajnal, Anthony Price, and Daniel Rueckert. 2017. “A Deep Cascade of Convolutional Neural Networks for MR Image Reconstruction.” In Information Processing in Medical Imaging, 647–58. Springer International Publishing.

Schottky, W. 1918. “Über Spontane Stromschwankungen in Verschiedenen Elektrizitätsleitern.” Annalen Der Physik 362 (23): 541–67.

Stanislaw, H., and N. Todorov. 1999. “Calculation of Signal Detection Theory Measures.” Behavior Research Methods, Instruments, & Computers: A Journal of the Psychonomic Society, Inc 31 (1): 137–49.
Starck, J., and F. Murtagh. 2001. “Astronomical Image and Signal Processing: Looking at Noise, Information and Scale.” *IEEE Signal Processing Magazine* 18 (2): 30–40.

Wandell, Brian A. 1995. *Foundations of Vision*. Sunderland, MA: Sinauer Associates.

Weyrauch, B., B. Heisele, J. Huang, and V. Blanz. 2004. “Component-Based Face Recognition with 3D Morphable Models.” In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, 85–85.

Wolfram, Stephen. 1983. “Statistical Mechanics of Cellular Automata.” *Reviews of Modern Physics* 55 (3): 601–44.

———. 2016. *A New Kind of Science (English Edition)*. Kindle. Wolfram Media, Inc.

Yamins, Daniel L. K., and James J. DiCarlo. 2016. “Using Goal-Driven Deep Learning Models to Understand Sensory Cortex.” *Nature Neuroscience* 19 (3): 356–65.

Yao, Jie, and Harrison H. Barrett. 1992. “Predicting Human Performance by a Channelized Hotelling Observer Model.” In *Mathematical Methods in Medical Imaging*, 1768:161–68. International Society for Optics and Photonics.

Zhou, Weimin, Hua Li, and Mark A. Anastasio. 2019. “Approximating the Ideal Observer and Hotelling Observer for Binary Signal Detection Tasks by Use of Supervised Learning Methods.” *IEEE Transactions on Medical Imaging*, April.

https://doi.org/10.1109/TMI.2019.2911211.