What Can You See? Modeling the Ability of V1 Neurons to Perform Low-Level Image Processing

Stone Yun University of Waterloo
Alexander Wong University of Waterloo
Email: {s22yun, a28wong}@uwaterloo.ca

Abstract

While not physiologically accurate, deep neural networks have a long history of being inspired by the brain. Of particular interest to computer vision researchers are the behaviour of neurons in the V1 Visual Cortex when responding to visual stimuli. Understanding how V1 neurons encode visual stimuli might offer insight on how to improve design of computer vision algorithms and "neural" representations of visual data. It has been known that neurons in the V1 cortex exhibit responses that can be modeled by 2D-Gabor filters. Knowing that, we wonder what kinds of functions a population of rate neurons with Gabor-like encoders would be able to perform on images. In this work we explore, via rate neuron modeling methods as described in the Neural Engineering Framework, what kinds of low-level image operations can be accurately encoded by a population of sparse Gabor encoders. Understanding what kinds of low-level image operations can be performed well by our simulated population of neurons could provide insight as to what kinds of feature extractions can plausibly be performed by the V1 visual cortex. We find that compared to the other operations tested such as Sobel filtering and high-pass filtering, our modeled V1 neuron population is better at low-pass filtering operations such as average filtering, as measured by the RMSE of decoding. The reasons for this are unclear for now and require further investigation.

1 Introduction

The primary visual cortex, or V1 cortex is the part of the brain that is responsible for processing visual sensory input coming from the retina via the lateral geniculate nucleus, a relay center for the visual pathway. It could be viewed as the part of the brain responsible for processing the "pixels" depicting the outside world that are rendered onto the retina. Being the first stage in visual processing, the V1 cortex is understood to be responsible for the initial stages of feature extraction and pattern recognition. Better understanding of the V1 cortex has greatly advanced research in both human and computer vision. Most notably, the works of Hubel and Weisel in [1, 2] dramatically shifted our understanding of visual processing and neuron organization. One of their key insights is that V1 neurons are organized into a hierarchical system to extract complex features from the retinal image [1, 2]. Furthermore, they also described how different cells can vary in the size of their receptive fields and may be selective to different kinds of simple or complex input patterns.

The research of Hubel and Weisel and many others studying the mechanisms of the V1 cortex would eventually inspire the convolutional neural network [3] and its precursor, the Neocognitron [4]. However, despite an explosion in design of brain-inspired image processing algorithms in recent years, not much research has been done on mathematically understanding what kinds of image processing the V1 cortex is actually good at. [5] demonstrates how having different densities of neurons assigned to different receptive field sizes effectively carries out an edge filtering computation of the image that is present on the retina. In this work, we wish to expand on this idea and explore what other kinds of low-level image processing operations a population of V1 cortex neurons may be good or bad at computing. To do this, we make use of the fact that V1 neurons have been observed to exhibit 2D-Gabor filter-like responses [6]. Thus, using the Neural Engineering Framework [7], we model a population of V1 neurons as a collection of rate approximate Leaky-Integrate-and-Fire (LIF) neurons with sparse Gabor encoders. The firing rate, $G[J]$, (or activation function) is described in Eq. 1 where J is the current induced when encoding the stimulus x, and τ_{ref}, τ_{RC} are parameters of the neuron.

$$G[J] = \begin{cases} \frac{1}{\tau_{ref} - \tau_{RC} \ln 1 - \frac{1}{J}}, & \text{if } J > 1 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

2 Methods

2.1 Neural Engineering Framework

In order to quantify how "good" our simulated V1 population is at performing a given image-processing operation, we make use of the tools defined in the Neural Engineering Framework (NEF) [7]. NEF describes a method for modeling the process by which an input signal x is non-linearly encoded into a population of neurons and represented by their collective firing rates, or in more physiologically accurate models, their simulated spike trains. The input stimulus is first encoded as a current as described by Eq. 2 where $\mathbf{e}, \alpha, J^{bias}$ are the neuron's encoding vector, gain and bias respectively. The encoded current can then be converted to a firing rate via the firing rate function (such as Eq. 1). Instead of using a firing rate value to approximate how often a given neuron is firing over time in the presence of a stimulus x, we can also directly simulate a neuron's spiking activity over time and get out a set of spike train's for each neuron. Each neuron's firing rate or alternatively, their simulated spike trains collectively form the population's activity matrix $\mathbf{A}$ (also referred to as a population code). This is illustrated in Eq. 4 where $\mathbf{E}$ is now a matrix containing the encoding vectors of each neuron in the population. As seen in Eq. 3, each column of $\mathbf{E}$ corresponds to the encoding vector of a neuron in the population.

Subsequently, NEF outlines how we can linearly decode the population code $\mathbf{A}$ to recover the originally encoded signal as $\hat{x}$ (see Eq. 5 where $\mathbf{D}$ are decoding parameters). Alternatively, if we wish for the neurons to apply a given function $f(x)$ we can also decode back the encoded transformation as $\hat{f}(x)$ (see Eq. 6 where $\mathbf{D}^f$ are the decoding parameters for computing $f(x)$). The error between our decoded $\hat{f}(x)$ and the ideal $f(x)$ quantifies how good our neuron population is at performing the desired operation. We will use root mean-squared error (RMSE) to describe decoding error.

Figure 1 shows the high-level process of encoding an image x into a population of neurons and then the process by which the image operation, $F(x)$ is computed. It is important to highlight that instead of learning the encoders $\mathbf{E}$ and decoders $\mathbf{D}^f$ via backpropagation and a loss function, the encoders are randomly generated Gabor filters (as dictated by findings in [6]) and the decoders are solved for using least-squares minimization to solve Eq. 6 over a set of sample inputs (ie. our training data). Thus, $\hat{F}(x)$ is the output of our neuronal layer and compared to the ideal function computation $F(x)$.

$$J = \alpha \langle \mathbf{e} \cdot \mathbf{x} \rangle + J^{bias} \tag{2}$$

$$\mathbf{E} = [\mathbf{e_1}^T, \mathbf{e_2}^T, ..., \mathbf{e_n}^T] \tag{3}$$

$$\mathbf{A} = G[\mathbf{J}] = G[\mathbf{E}^T x] \tag{4}$$

$$\hat{\mathbf{x}} = \mathbf{DA} \tag{5}$$

$$\hat{F}(\mathbf{x}) = \mathbf{D}^f \mathbf{A} \tag{6}$$

2.2 Image Processing Operations

In [5], the author primarily focuses on demonstrating the capability of V1 neurons to perform edge detection filtering. We would like to start with a simple V1 population model and simulate the computation of an expanded set of low-level image operations. These operations range from basic filtering such as low-pass filtering, to more complex inverse tasks such as image deblurring. We picked these as our initial operations since they are very common in various image processing applications. For example, the 3×3 Gaussian kernel with $\sigma = 0.85$ is often used because its fixed point approximation involves only powers of 2. The complete list of the operations we model is below. There are eight different operations in total when counting operation variants:

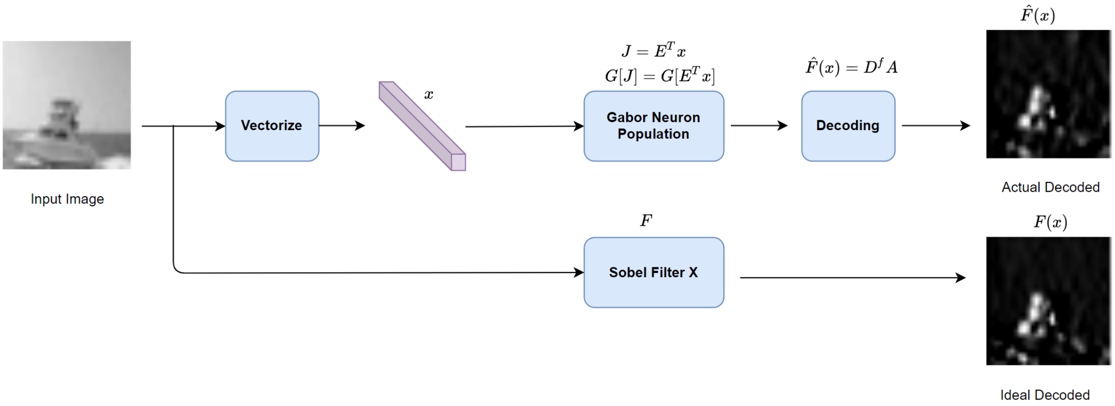

Fig. 1: Flow chart showing how we encode an image into the population of rate approximate Gabor neurons and then decode out the result of the neurons computing the desired low-level operation, $\hat{F}(x)$. The decoders $\mathbf{D}^f$ are computed via least-squares minimization of Eq. 6 over our entire training set. The computed decoders are then used for all testing inputs. We use the X-direction Sobel filter as an example.

- Gaussian blur with $\sigma = 0.85$, kernel width of 3.
- Averaging filter with varying square kernels of width {3, 5, 7}.
- 3×3 Sobel filtering in each of the X and Y directions.
- 3×3 High-pass filtering via unsharp filtering; we use the same Gaussian blur as listed above.
- Image deblurring, specifically the inverse of a 3×3 Gaussian blur with $\sigma = 0.85$.

This is by no means an exhaustive list of the low-level image operations that could be modeled. Computing the Laplacian, image sharpening, and image denoising are some other computations that we wish to explore in future work.

2.3 Experiments

We use the CIFAR-10 dataset [8] and convert it to the YCrCb colour space before taking the Luminance channel, Y. This saves computation and furthermore, for many low-level image operations involving the structure of the image, the human visual system is known to be more sensitive to changes in the Y-channel. We use a population of 2500 neurons to cover the entire 32×32 image. Thus, each neuron's encoding vector is of size 1024. However, since we are using *sparse* Gabor encoders with a receptive field of K, there are only $K \times K$ non-zero elements in each encoding vector. As mentioned in [5], the V1 cortex often has bundles of thousands of neurons for each spatial location/region. Thus, in future works it would be more accurate to use a convolutional transform to model the neuron population encoding. However, due to computation and time constraints, we use a population of $N = 2500$ neurons where each neuron is randomly assigned a location of the image to attend to.

For our main experiments, there is an even distribution of neurons having receptive fields of size $K \in \{3 \times 3, 5 \times 5, 7 \times 7\}$ (we will call this our main model) and we compute the RMSE of decoding each of the 8 above-listed image operations from the neuron population code. Next, we seek to investigate the relationship between population size N and RMSE as well as the relationship between receptive field and RMSE. For the population size ablation, we compute the RMSE for each of the 8 operations for V1 neuron populations of size $N \in \{500, 1000, 1500, 2500\}$. These populations also have an even distribution of neurons among the three different receptive field sizes mentioned above. Finally, for our receptive field size ablation, we use a population size of $N = 1250$ neurons with all neurons having the same given receptive field size K which we ablate through the set of three receptive fields mentioned above. In all cases we use rate approximate neurons for our simulations. Future work should move to using spiking neurons as they are more biologically accurate.

3 Results and Discussion

Table 1 shows test RMSE for each of the 8 operations. Each population model is defined by the receptive field of the neurons. For example, $K3$ denotes a population where all neurons have receptive field of size 3×3. "Mixed" refers to our main model. As somewhat expected, having only larger receptive fields appears to increase decoding error. This would make sense since we have started off modeling very local operations. We observe the lowest error for performing the various low-pass filtering operations. However, it is currently unclear why. Further investigation is needed.

Figure 2 shows how $\log(\text{RMSE})$ changes as a function of population size N for each of the operations. RMSE decreases with larger neuron populations as expected which suggests that the differences in RMSE are inherent to the operation and not due to noise in an insufficiently large neuron population. Figures 4 and 3 show how RMSE changes as a function of receptive field size. We plotted the RMSE of the Sobel filters separately to better show the shape of the curves. It is a bit surprising that the error even increases with receptive field size for the 7×7 averaging filter. Though this may also have to do with the resolution of our input images being incredibly small. At $K = 7$, the receptive field covers almost a quarter of the image and thus local image statistics would likely be lost. It will be important to use larger input resolutions for our results to be more generalizable.

Finally, Table 1 also shows the decoding error of a population with neurons that have a single, fixed receptive field so we can compare to our main model. From [5], we know that the number of neurons for different receptive field sizes follow an empirical Gamma distribution. It would be interesting to see how decoding error changes with different distributions and to include much larger receptive fields in our model.

In this initial study, we have only modeled the sparse Gabor encoders of a single, small neuron population. In future works, it would be useful to observe how accuracy can improve with a few feed-forward connections between populations. Furthermore, we know from works such as [9], [10], [11] that neurons in the V1 cortex contain both lateral and feedback connections. Previous works have suggested that these connections may be significant in enabling the large receptive fields of V1 neurons and thus we should also model them if we wish to gain an accurate understanding of what kinds of low-level image operations are mathematically plausible for the V1 cortex.

4 Conclusion

We present an initial study on modeling the ability of V1 neuron populations to compute low-level image operations. Better mathematical understanding of what the V1 cortex is good/bad at computing could yield further insights for human and computer vision. Future work includes modeling a greater number of low-level image operations as well as improving the accuracy of our V1 population model such as using spiking neurons, modeling multiple neuron layers and including lateral and feed-back connections in these layers.

References

[1] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of Physiology*, vol. 148, 1959.

[2] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular in-

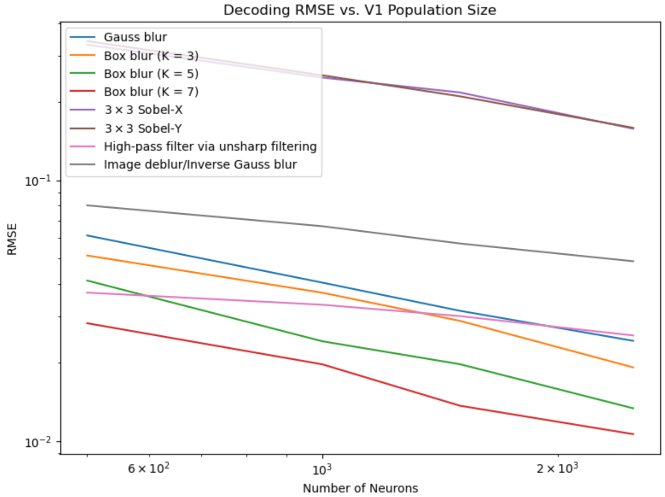

Fig. 2: $\log_{10}(RMSE)$ vs population size N. We would expect for all operations that RMSE decresaes as N increases.

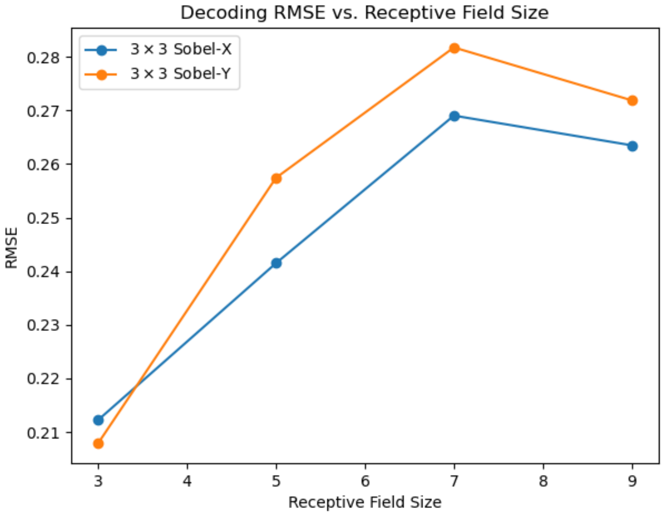

Fig. 3: $RMSE$ vs receptive field size K. To better show the change in RMSE we separated out the Sobel operations since their RMSEs were significantly higher.

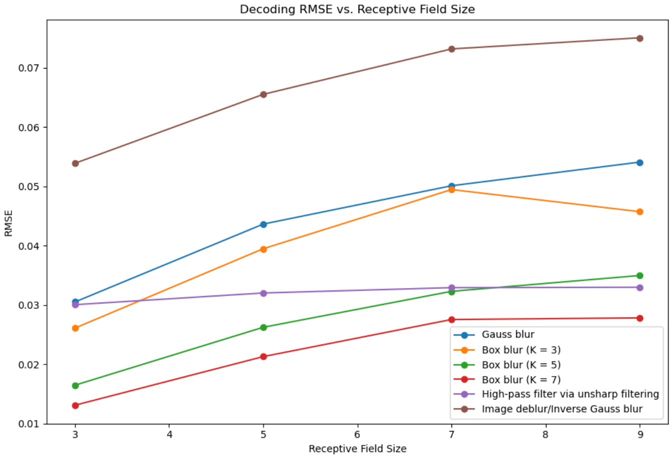

Fig. 4: $RMSE$ vs receptive field size K for the rest of the operations.

Table 1: Decoding RMSE for each of the 8 image operations using different V1 population models.

Operation	Mixed	K3	K5	K7
Gauss blur	0.0244	0.0197	0.0302	0.0374
Box blur (K = 3)	0.0221	0.0169	0.0254	0.0353
Box blur (K = 5)	0.0137	0.0105	0.0181	0.0242
Box blur (K = 7)	0.0101	0.0076	0.0134	0.0176
3×3 Sobel-X	0.1629	0.1431	0.1777	0.2056
3×3 Sobel-Y	0.1647	0.1463	0.1763	0.2037
High-pass filter via unsharp filtering	0.0256	0.0238	0.0263	0.0275
Image deblur/Inverse Gauss blur	0.0498	0.0456	0.0530	0.0606
Average across operations	0.0592	0.0517	0.0651	0.0765

teraction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, 1962.

[3] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[4] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.

[5] C. F. Stevens, "Novel neural circuit mechanism for visual edge detection," *Proceedings of the National Academy of Sciences*, vol. 112, no. 3, pp. 875–880, 2015. [Online]. Available: https://www.pnas.org/content/112/3/875

[6] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles," *Vision Research*, vol. 20, pp. 847–856, 1980.

[7] C. Eliasmith and C. H. Anderson, *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. Cambridge, MA: MIT Press, 2003.

[8] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[9] V. Piëch, W. Li, G. N. Reeke, and C. D. Gilbert, "Network model of top-down influences on local gain and contextual interactions in visual cortex," *Proceedings of the National Academy of Sciences*, vol. 110, pp. E4108 – E4117, 2013.

[10] S. Khan, A. Wong, and B. P. Tripp, "Task-driven learning of contour integration responses in a v1 model," in *NeurIPS 2020 Workshop SVRHM*, 2020.

[11] D. Linsley, J. Kim, V. Veerabadran, C. Windolf, and T. Serre, "Learning long-range spatial dependencies with horizontal gated recurrent units," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.