Neural Network-Based Image Prediction: Bridging the Gap Between Neuronal Responses and Visual Reconstructions

Jared Cohen

Computational Perception - 15-786 - Fall 2023

12/3/2023

# 

# INTRODUCTION

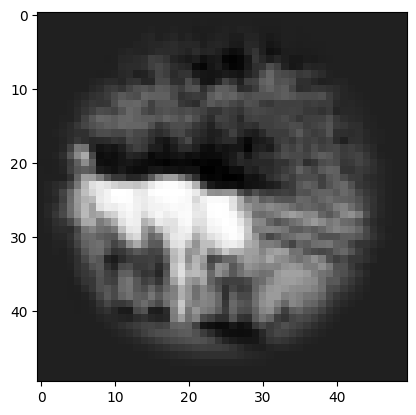
Our brain can process and comprehend a near infinite amount of stimuli due to a large number of neurons and their vast web of interconnections that enhance pattern recognition, stimuli compression, and stimuli analysis. This may be attributed to the emergent properties of multiple neurons interacting with each other (DiCarlo JJ). These neurons individually react to visual stimuli that vary in features such as orientation and shape (Gawne TJ). As stimuli progress through the neural layers, the interactions between neurons become increasingly complex and can result in the emergent property that allows the brain to recognize holistic structures, such as faces or objects.

This serves as inspiration for computational neuroscience, where one of the fundamental goals of the field is to understand and replicate the features and phenomena associated with our brain's interpretation of stimuli and higher-order cognitive functions. Researchers seek to emulate this intercellular communication and organization by creating artificial neural networks. These neural networks are effective at learning complex tasks and have been shown to recognize patterns (Kurogi S), adapt to changes (Kubo Y), and solve problems, mirroring the adaptive and cognitive abilities that humans have.

Individual neurons in artificial neural networks have been shown to activate/deactivate in the presence of specific patterns in the image. A neuron might respond strongly to the presence of edges with a specific orientation, while another might activate in response to the existence of textures or color gradients. It is the combination of weights and these preferred patterns that allow neural networks to respond to almost any input provided, similar to how the brain can interpret any visual stimuli (DiCarlo JJ).

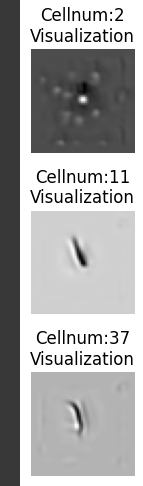
Motivated by the intricate responsiveness of individual neurons in artificial neural networks to specific visual patterns, this project endeavors to bridge the gap between learned patterns and image reconstructions. This project's central inquiry hopes to leverage artificial neural networks to regenerate an image based on the responses of a predefined model’s neurons.

# RESEARCH METHODOLOGY

**Image Pre-Processing**

A dataset comprising 34,000 images underwent pre-processing to optimize their suitability for analysis. All images were uniformly resized to dimensions of 50x50 pixels to standardize the quantity of information in the picture and reduce the training for each neural network. A circular mask of radius ~22 pixels was applied to the images to focus the model's attention on the central regions of interest within each image. The images were converted to grayscale, a single-channel representation of the image, to minimize computational load while maintaining the fundamental information necessary for pattern recognition and image interpretation. An example of a properly pre-processed image is shown.

**CNN to Convert Image to Response**

A convolutional model was generated and prepared to predict the neuronal responses from the processed image. This model contained a sequence of convolutional layers, batch normalization, soft plus activation functions, and factorized linear layers, carefully designed to extract hierarchical features from the input image (Klindt). The architecture is equipped with a total of 9 layers, each contributing to the transformation of the input image into a series of neuronal responses. The use of factorized linear layers and specific convolutional kernel sizes focuses on capturing both global and local features in the image and fosters a comprehensive representation for additional downstream processing. Average pooling was utilized to reduce spatial dimension and ensure information was properly transferred to the output.

The input of 2500 values, one for each pixel in the 50 by 50 grayscale image, was reduced to 302 through this neural network, where each element in the output represented the responses of a neuron. After the neural network was trained, its values were stored and reused to maintain each neuron’s preferred pattern across tests and runtimes.

It is possible to visualize each neuron’s preferred pattern, and this would show how neurons naturally arrange themselves to maximize the specificity for the large variety of given images. One technique utilized to determine a given neuron’s preferred pattern is to optimize or create an image that maximizes the neuron’s relative response. This result creates an image that often contains local areas of activation (white) and suppression (dark), and the relative orientation of the areas can form detectors for edges, eyes, gratings, and more, as shown.

**Novel Neural Networks**

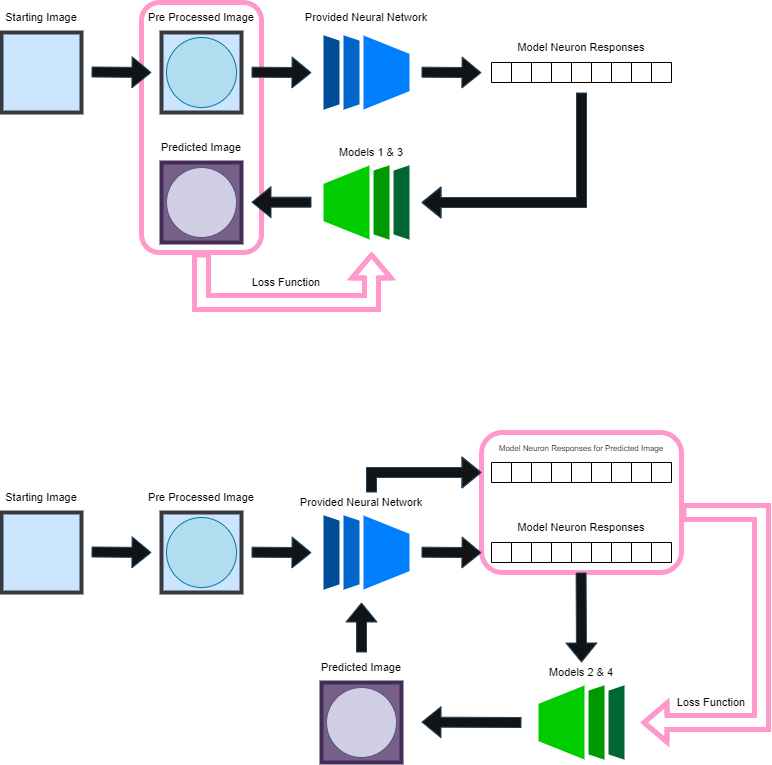
The information above does not rely on any personal work, as the data and network were provided by Tianqin Li and Professor Lee. The networks generated below are novel and self-chosen or self-designed to reach the project’s goals while managing limitations on computational power. Each network below received 25000 of the 34000 images as training data, with the rest being utilized for each network’s accuracy testing.

*Model Number and Its Respective Architectures - Table*

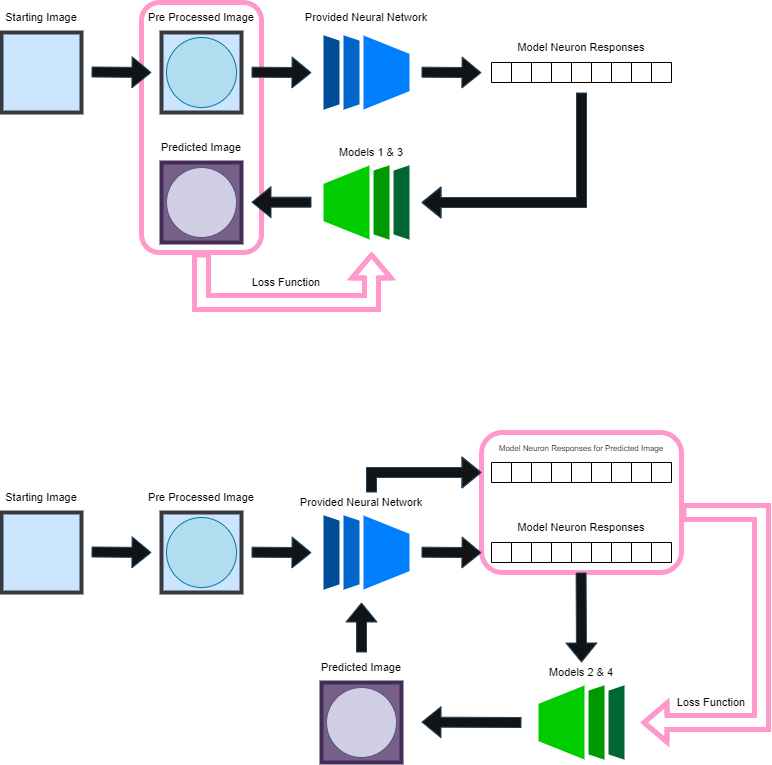
|  | Information Flow Architecture 1 (Pixel-based loss function) | Information Flow Architecture 2 (Neuron Response loss function) |
| --- | --- | --- |
| Neural Network Architecture 1 (Perceptron) | **Model 1** | **Model 2** |
| Neural Network Architecture 2 (CNN) | **Model 3** | **Model 4** |

It is important to mention that there are two information flows and two neural network architectures, resulting in four different models that are tested in this project.

The two architectures explored in this project are provided and explained below.

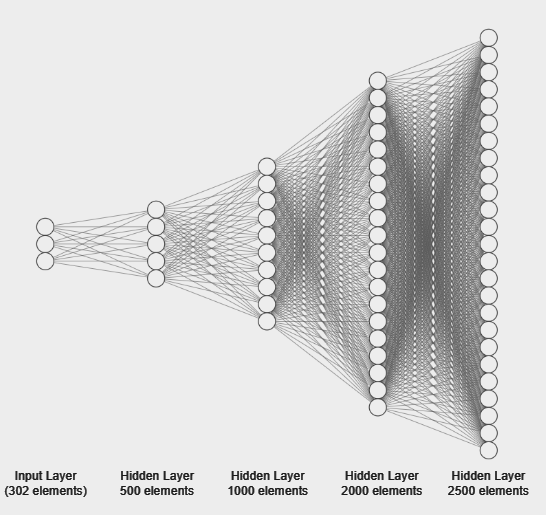
*Architecture 1 - Pixel Loss Driven*

In this architecture, the images are preprocessed as previously mentioned, and are then passed through the provided neural network to result in 302 neural responses. These are fed into a self-designed neural network (in this case, either models 1 or 3) to generate a predicted image, which is compared pixel-by-pixel to the original pre-processed image in the loss function. As training continues, the predicted image should look more like the unchanged pre-processed image.

*Architecture 2 - Perceptual Loss Driven*

In this architecture, the images are preprocessed and then passed through the provided neural network to obtain 302 neural responses. These responses are passed into a self-designed neural network (in this case, either model 2 or 4) to generate a predicted image. This image is then fed back into the provided neural network to generate a series of neural responses, which are then compared neuron-by-neuron with the pre-processed image’s neural responses in the loss function. As training continues, the predicted image’s neuronal responses should approach the pre-processed image’s unchanged neural response values. This may result in the predicted image to match the pre-processed image as well.

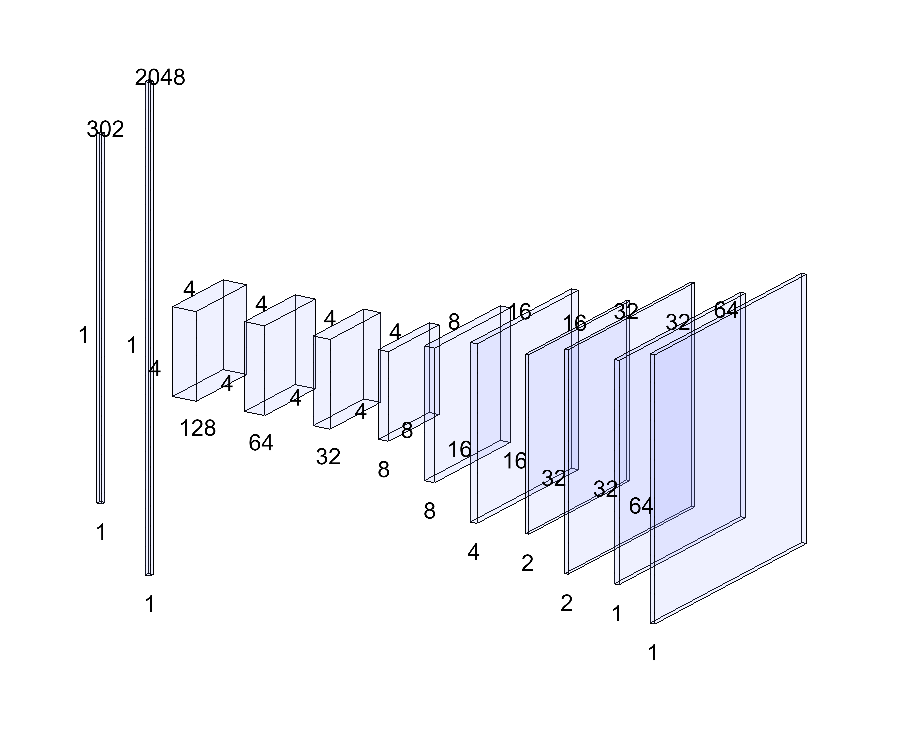
The two neural network architectures explored in this project are provided and explained below.



*Neural Network Architecture - Multi-Layer Perceptron*

The Neural Networks in Models 1 and 2 utilize a sequential architecture with four fully connected layers, progressively increasing in size from 500 to 2500 neurons. Each layer is followed by a rectified linear unit activation function, introducing non-linearity to capture intricate patterns within the data. The final layer employs a sigmoid activation to normalize the pixel values. This structured configuration enables the model to transform the 300-element input neuron responses into a 2500-element output, which can be reshaped to match the 50x50 image.

*Neural Network Architecture - Convolutional Neural Network*

The architecture of neural networks 3 and 4 will utilize a mixture of convolution and single-layer perceptrons to generate the desired output. There is a fully-connected layer that converts the 302 inputs to 2048 outputs. These outputs are placed in 128 4x4 channels. Each channel is convolved, ReLUd, and upsampled to decrease the number of channels and increase the size of each channel. The kernel size is maintained at 3 (with channel number variation) to recognize and generate local and global patterns. After the convolutions and upsamples, we obtain a 64x64 pixel image, which is then compared to the upscaled original image in model 3’s loss function. In model 4, the predicted image is downsized to 50x50 and utilized to generate 302 neuron responses for its loss function. 

**Data Analysis Formatting**

The models’ effectiveness can be determined by various qualitative and quantitative methods that will use a combination of all of the test data or each photo’s results to generate inferences.

Each picture and its respective neuronal response gives information about the model’s successes and failures, and the data analysis will attempt to show how similar these results are.

|  | For Each Test Image per Model | For All Test Images per Model |
| --- | --- | --- |
| Pixel by Pixel | 1) Presentation of Predicted Image  2) Presentation of difference between Images  3) Box Plots between Images  4) Histogram of Difference between Images  5) Correlation between Pixel Values between Images | 6) Distribution of Correlations between Images  7) Distributions of Mean, Standard Deviation, Skew between Images  8) Average Mean Squared Error between Images |
| Neuron by Neuron | 3) Box Plots between Responses  4) Histogram of Difference between Responses  5) Correlation between Pixel Values between Responses | 6) Distribution of Correlations between Responses  7) Distributions of Mean, Standard Deviation, Skew between Responses  8) Average Mean Squared Error between Responses |

# 

# RESULTS

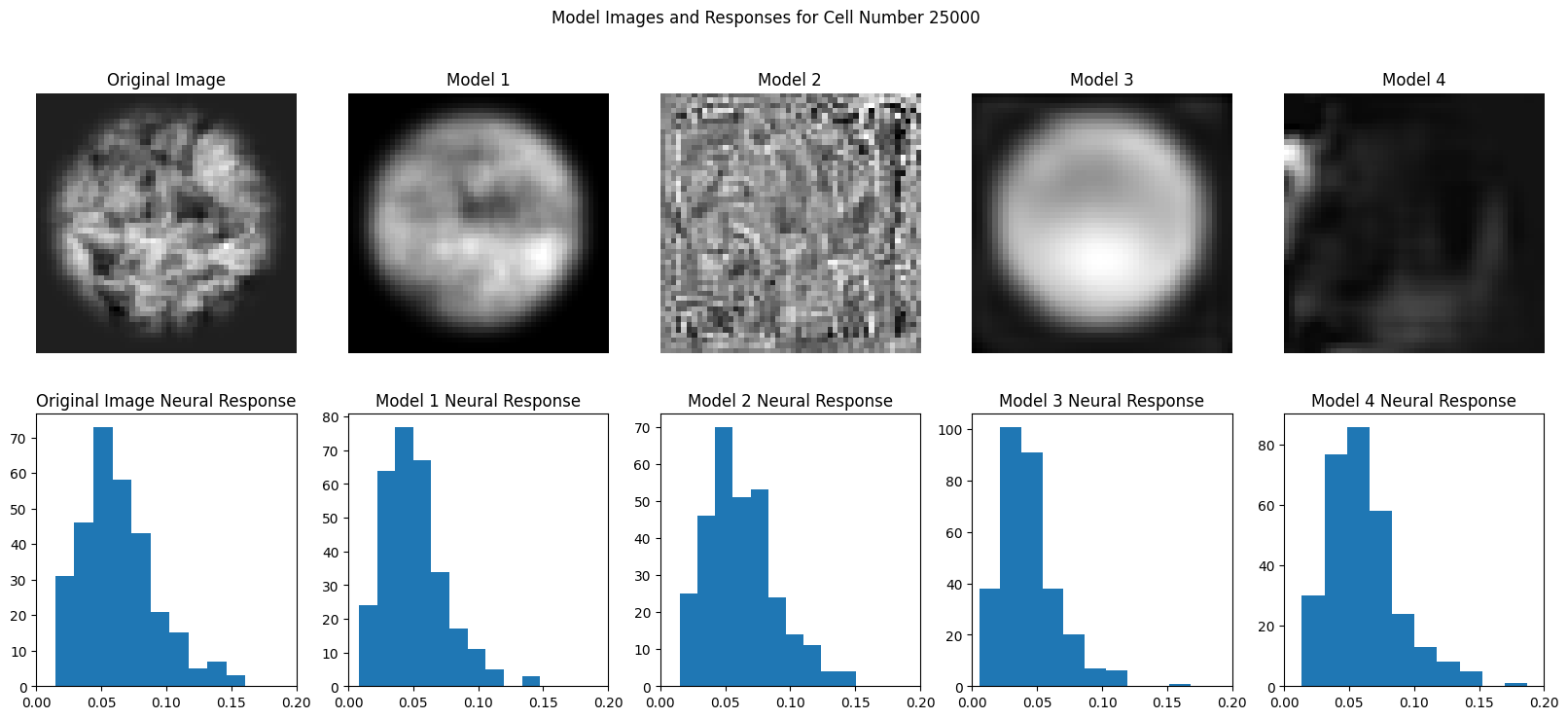
**Hyper Parameter Consideration**

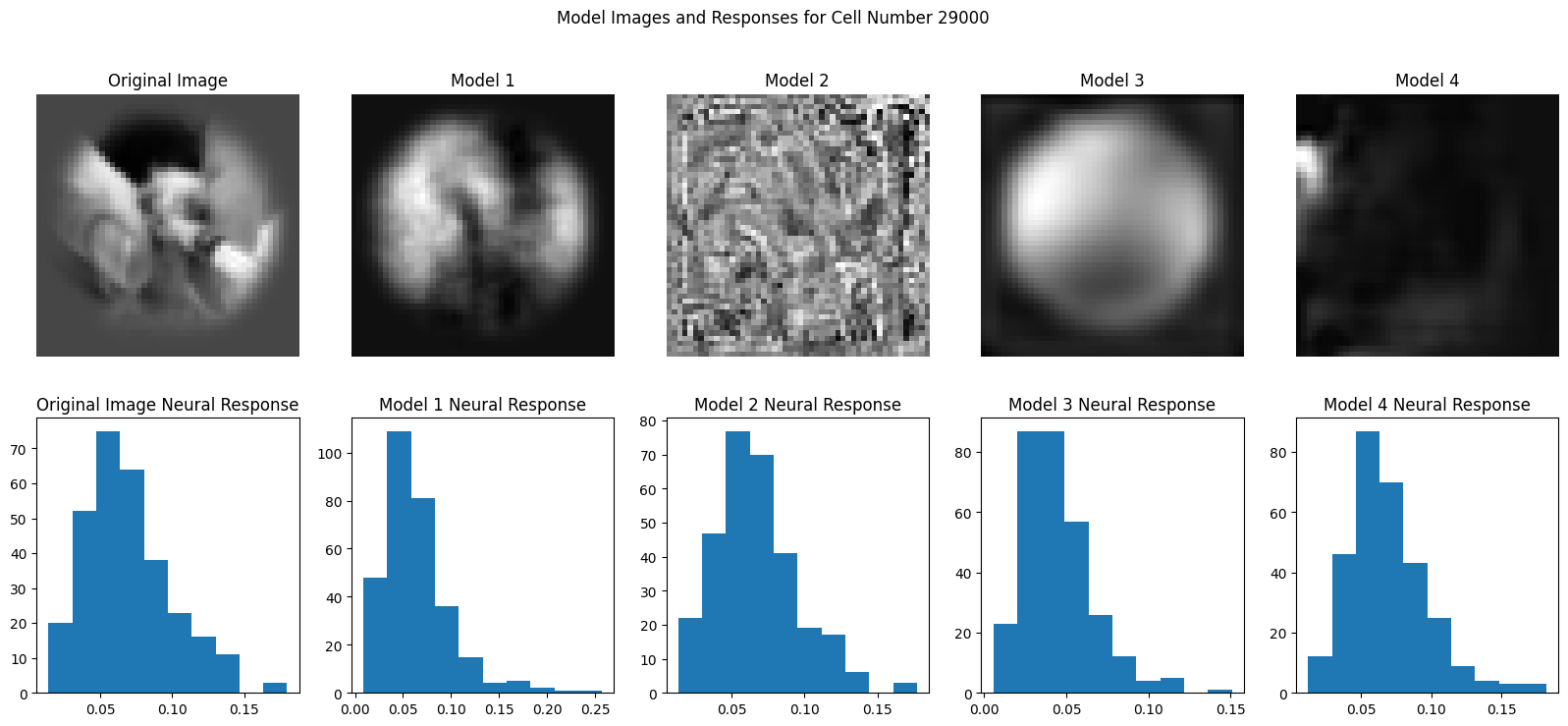
|  | Model 1 | Model 2 | Model 3 | Model 4 |
| --- | --- | --- | --- | --- |
| Completed Epoch | 100 | 10 | 100 | 40 |
| Batch Size | 100 | 100 | 1000 | 100 |
| Optimizer Algorithm | Adam (LR = .001) | | | |
| Loss Function | Mean Squared Error | | | |

**Comparison of Models’ Results**

|  | Model 1 | Model 2 | Model 3 | Model 4 |
| --- | --- | --- | --- | --- |
| Unmasked Pixel-By-Pixel Correlation (image accuracy) | .64 | .09 | .64 | .03 |
| Neuron-By-Neuron Correlation (perceptive accuracy) | .87 | .95 | .78 | .93 |
| Average Mean Squared Error for Images | .012 | .106 | .012 | .619 |
| Average Mean Squared Error for Responses | .00067 | .00017 | .0013 | .00028 |
| Circular Mask Generated | Yes | No | Yes | No |

*Example Predicted Images*

****

****

# DISCUSSION

**Discussion of Results**

After training, Models 1 and 3 properly ignored the irrelevant pixels (such as the image mask surrounding the center pixels) present in the target image. This is due to the loss function, which relies on the original and the predicted images’ pixels to optimize the model. When utilizing image pixel values in the loss function, the neural network can easily understand the influence of each pixel and its relation to nearby pixels on the target image and thus can complete gradient descent with greater accuracy.

During training, one of the first features to form on the predicted image was the circle, as the mask promoted focus on the center set of pixels (and thus was easy to set a large quantity of weights and biases to 0). As the model continued to train, global regions appeared to turn light and dark, which represented the model learning how the neurons relate to the patterns featured on the target image. Patterns continued to get more and more local, with the range of locality limited by the underlying architecture.

Model 3 was mostly limited to large variations in pixel value color – generally light and dark regions were well described by the model, but few local features were present. This may be due to the limitations of the convolutional architecture utilized in the project, where the number of channels, kernel size, and size of each channel could play a role.

The input into all of the models was a 302-element list, and the output was a 50x50 (1&2) or 64x64 array (3&4). Because of the large dimensions associated with this function, there is a high chance that multiple predicted images will result in the same neuronal response. The starting image and starting parameters are important factors to drive which iso-perceptual picture will be chosen. Models 2 and 4’s final images show that this is true – the neural responses were very accurate on average, yet their respective images did not contain most or any of the features that were present in the original image – including the background created during pre-processing. The first few epochs of training established the overall pattern of the final predicted image, and additional epochs ensured that the local patterns were configured to maximize the neural response. This ‘wrong start’ is due to the inherent nature of the loss function, which relies on the original and predicted images’ neural responses to optimize the predicted image.

One of the underlying questions presented by the project was whether or not neurons can be utilized in the loss function to drive the predicted image closer to its target counterpart. As shown by Models 1 and 3, utilizing the images in the loss function will ultimately result in a prediction that closely resembles the target image. Inherently, that image will have similar neuronal response values to the target image. Therefore, it is unreasonable to optimize the image utilizing the neuronal responses. This, along with the wide variety of iso-perceptual images previously discussed, shows that a neuron-reliant loss function in these architectures is ineffective in recreating the images.

**Areas to Improve**

Several areas for improvement have been identified in this study, offering avenues for refining the proposed approach. The reliance on pre-processed images – including masks – may introduce unnecessary constraints, and eliminating the pre-processing steps may allow the model network to autonomously discern relevant features. It may also be possible to train the convolutional neural networks to further enhance the network's ability to focus on critical regions by implementing an automatic masking mechanism during training.

Expanding the dataset with a larger number of images is crucial to improve model generalization and robustness. This augmentation, coupled with an exploration of different hyperparameters – such as learning rate, changing of the loss function and optimizing function, size of epochs, and batch sizes, can significantly impact the models’ learning process and hopefully improve their accuracy.

Adjustments to the networks’ architectures, such as parallel convolution and upscaling, variations in kernel sizes, alterations in the number of hidden layers, and alterations in the number of weights in each hidden layer also present opportunities to enhance the model's capacity for capturing intricate patterns and details within the images.

Addressing these areas for improvement will contribute to the development of a more robust and effective neural network-based image generation system.

# CONCLUSIONS

In conclusion, it is possible to utilize a neural network algorithm to regenerate a series of images utilizing the neural responses of the original image. While convolutional networks provide an efficient algorithm to train, they do not result in a high accuracy with local/sharp pixel variations. Rather, a multi-layer perceptron can be used to generate a higher-accuracy image.

Applications of image generation given neural responses can be used in various fields across science and medicine. If the model is successfully expanded to include more neurons and is shown to be valid for biological neurons, an algorithm that can interpret and decipher neuronal signals can be instrumental in diagnosing neuropathologies and facilitating advancements in neuroscientific research (Glaser JI).

# ACKNOWLEDGEMENTS

I am truly grateful for the encouragement, mentorship, and support provided by both Tianqin and Tai Sing Lee, who have been instrumental in the successful completion of this project.

# 

# REFERENCES

Gawne TJ. The responses of V1 cortical neurons to flashed presentations of orthogonal single lines and edges. J Neurophysiol. 2015 Apr 1;113(7):2676-81. doi: 10.1152/jn.00940.2014. Epub 2015 Feb 11. PMID: 25673741; PMCID: PMC4416608.

Kubo Y, Chalmers E, Luczak A. Biologically-inspired neuronal adaptation improves learning in neural networks. Commun Integr Biol. 2023 Jan 17;16(1):2163131. doi: 10.1080/19420889.2022.2163131. PMID: 36685291; PMCID: PMC9851208.

Kurogi S. A model of neural network for spatiotemporal pattern recognition. Biol Cybern. 1987;57(1-2):103-14. doi: 10.1007/BF00318720. PMID: 3620538.

DiCarlo JJ, Zoccolan D, Rust NC. How does the brain solve visual object recognition? Neuron. 2012 Feb 9;73(3):415-34. doi: 10.1016/j.neuron.2012.01.010. PMID: 22325196; PMCID: PMC3306444.

David A. Klindt, Alexander S. Ecker, Thomas Euler, and Matthias Bethge. 2017. Neural system identification for large populations separating what and where. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 3509–3519.

Glaser JI, Benjamin AS, Chowdhury RH, Perich MG, Miller LE, Kording KP. Machine Learning for Neural Decoding. eNeuro. 2020 Aug 31;7(4):ENEURO.0506-19.2020. doi: 10.1523/ENEURO.0506-19.2020. PMID: 32737181; PMCID: PMC7470933.

# IMPORTANT LINKS

**Google Drive Link:** https://drive.google.com/drive/folders/1kehcCfz3nYfAp3CtbIXVeM5HZtZzKL2T?usp=sharing

**Photos Utilized for Training and Testing:** <https://www.ni.cmu.edu/~tai/cp_public/>

**Google Colab Link:** <https://colab.research.google.com/drive/1OtWWlJ1ZxKUERGJcRiCm5ClD0fjaaMKw?usp=sharing>

**Large Project Writeup (~15 pages):** https://docs.google.com/document/d/1G3wsihzUQ9NN8WO2HINKzYes\_vdiFfLUXEFCF40JTEE/edit