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

# ABSTRACT

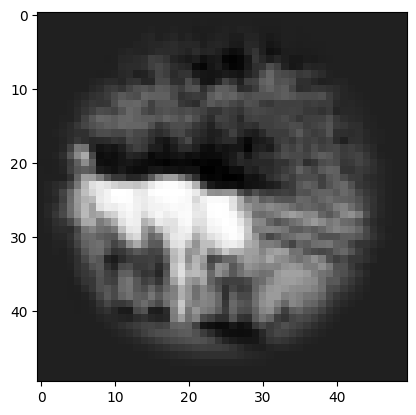
This project presents a novel approach to predicting images based on the responses of model neurons. Using the knowledge that neurons favor specific patterns in the image and that the neurons will activate in the presence of that pattern, we can attempt to recreate an image based on the relative activations of each neuron. The paper presents various approaches to achieve the results and discusses the successes and failures associated with each. Overall, the multi-layer perceptron model with picture loss function inputs provided the strongest results due to its ability to individually adjust each pixel in the predicted image and its ability to optimize its predictions to match the target image.

# 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.

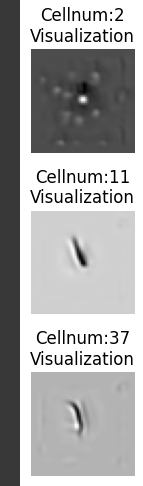
Individual neurons in artificial neural networks have been shown to activate/deactivate in the presence of specific patterns in the image. A neuron might respond strongly to the presence of edges with a specific orientation, while another might activate in response to the existence of textures or color 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.

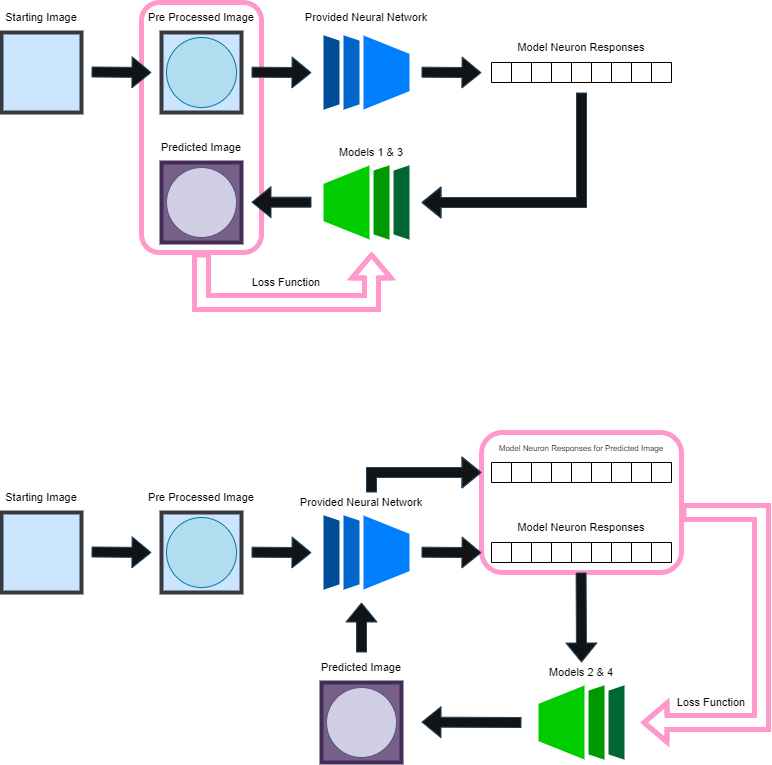
It is possible to visualize each neuron’s preferred pattern, and this would show how neurons naturally arrange themselves to maximize the specificity for the large variety of given images. One technique utilized to determine a given neuron’s preferred pattern is to optimize or create an image that maximizes 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**

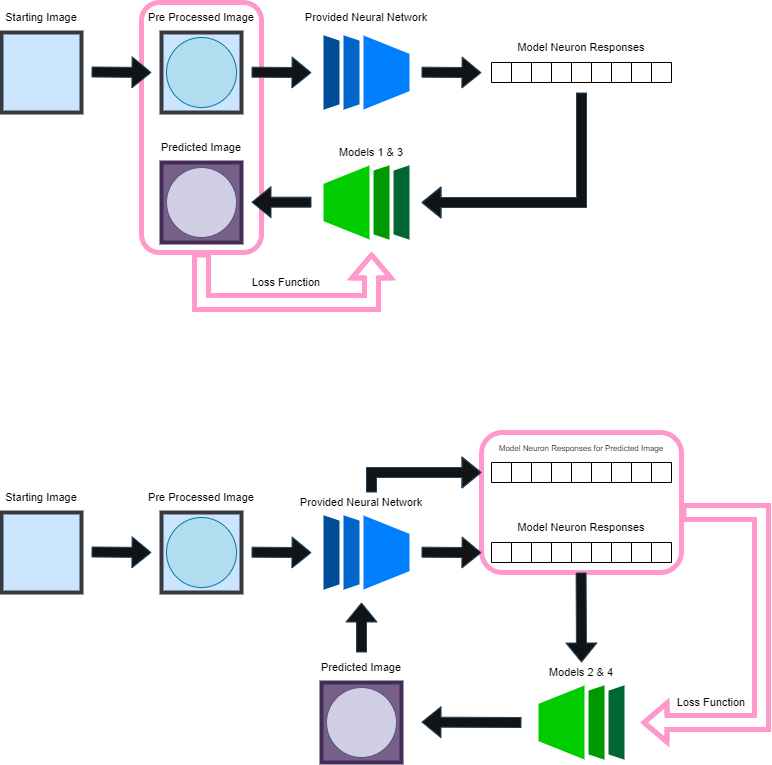
The information above does not rely on any personal work, as the data and network were provided by Tianqin Li and Professor Lee. The networks generated below are novel and self-chosen or self-designed to reach the project’s goals while managing limitations on computational power. Each network below received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

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.

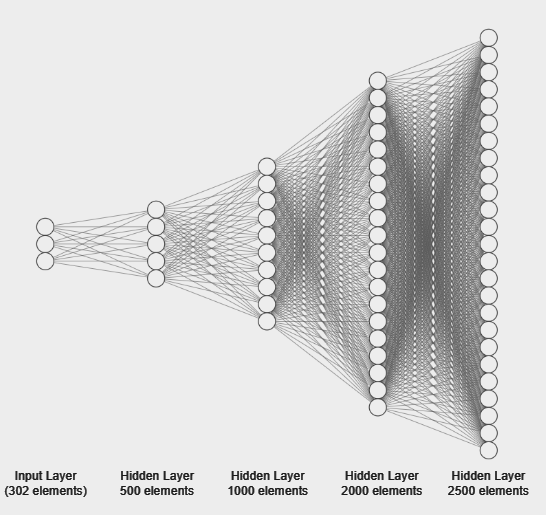
The two architectures explored in this project are provided and explained below.

*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.

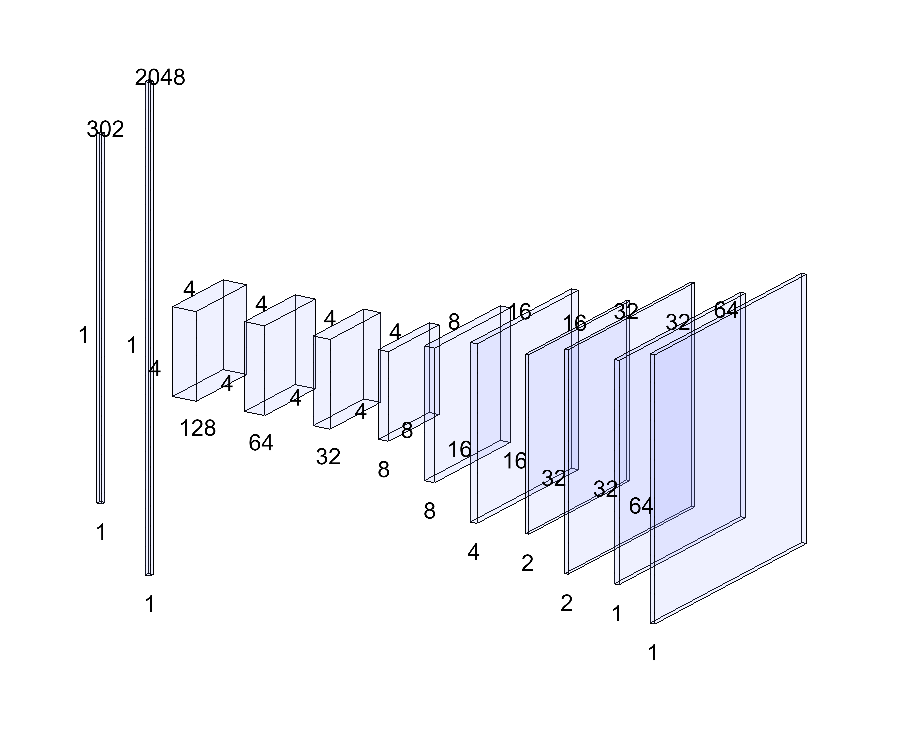


The two neural network architectures explored in this project are provided and explained below.

*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. 

*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** |

**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 |

1. Presentation of Predicted Image → The model’s normalized predicted image will be shown in grayscale
2. Presentation of Difference between Images → Subtracting the pixel values between the images will result in a picture that describes the error with the predicted image. Incorporating the maximum and minimum as the largest and smallest pixel values from all 3 images (predicted, target, difference) will result in the normalization of each photo. An ideal set of pictures would result in the ‘difference’ image being barely visible (not included in the report).

*Note: The numbers below are generalized for both images and responses*

1. Box Plots between Results→ Important Values (non-masked values and all neurons) will be placed in box plots to determine the similarity in data values. If there is a near-equal range between each quartile, there implies a similarity between the predicted and target data. This does not guarantee a near exact match between each data point, simply due to the limitations of the representation (not included in the report).
2. Histogram of Difference between Results → Important values (non-masked values and all neurons) between each image/neuron will be subtracted and placed on a histogram to create a distribution. If the distribution has a low standard deviation, a mean close to zero, and a near-zero skew value, the images can be considered nearly identical (not included in the report).
3. Correlation between Pixel Values between Images → Important values (non-masked values and all neurons) will be plotted and a correlation will be computed. If the correlation score is strongly positive (i.e. close to 1), the model's predictions closely align with the target images.
4. Distribution of Correlations → The correlation is a strong metric to determine the accuracy of the results. Each model will generate a distribution of correlations between the image it creates and the target image across all test images. This will be plotted (red) and will have its mean, skew, and standard deviation
5. Distributions of Mean, Standard Deviation, Skew → The difference between each value (pixels or neurons) in the predicted and target result can be used to generate a histogram for each result (as mentioned in #4) with its mean, standard deviation, and skew. These three metrics for determining a shape of distribution were created into its distribution when compared with all other test images/neurons to determine whether or not the model was able to generalize across the test data. The most important graph from these three is the Mean, where the relation between the predicted and target results is near-exact if the distribution has a low STD, near-zero-mean and near-zero-skew. [Blue = Mean Distribution, Yellow = Standard Deviation Distribution, Green = Skew Distribution].
6. Average Mean Squared Error → The Mean Squared Error is useful for determining the degree of dissimilarity or variance between the predicted and target results. This value was a determining metric for the model’s accuracy, where the prediction is better as the value approaches 0.

The aforementioned data representations should determine the effectiveness of the models at generating an image from the target image’s neuronal responses. Data presentations that are not present in the report can be visualized on the Google Colab/Github file.

# 

# RESULTS

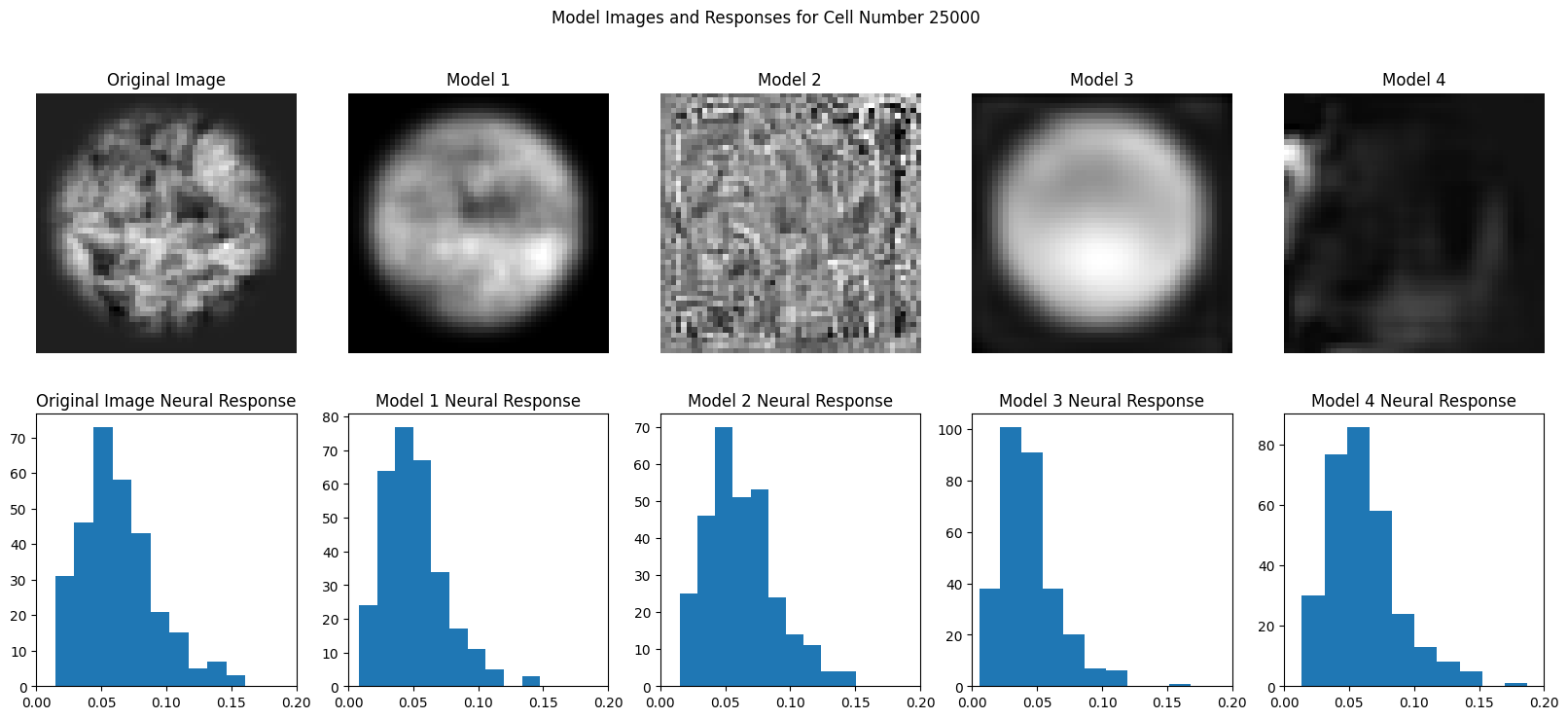
**Hyper Parameter Consideration**

|  | 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 | | | |

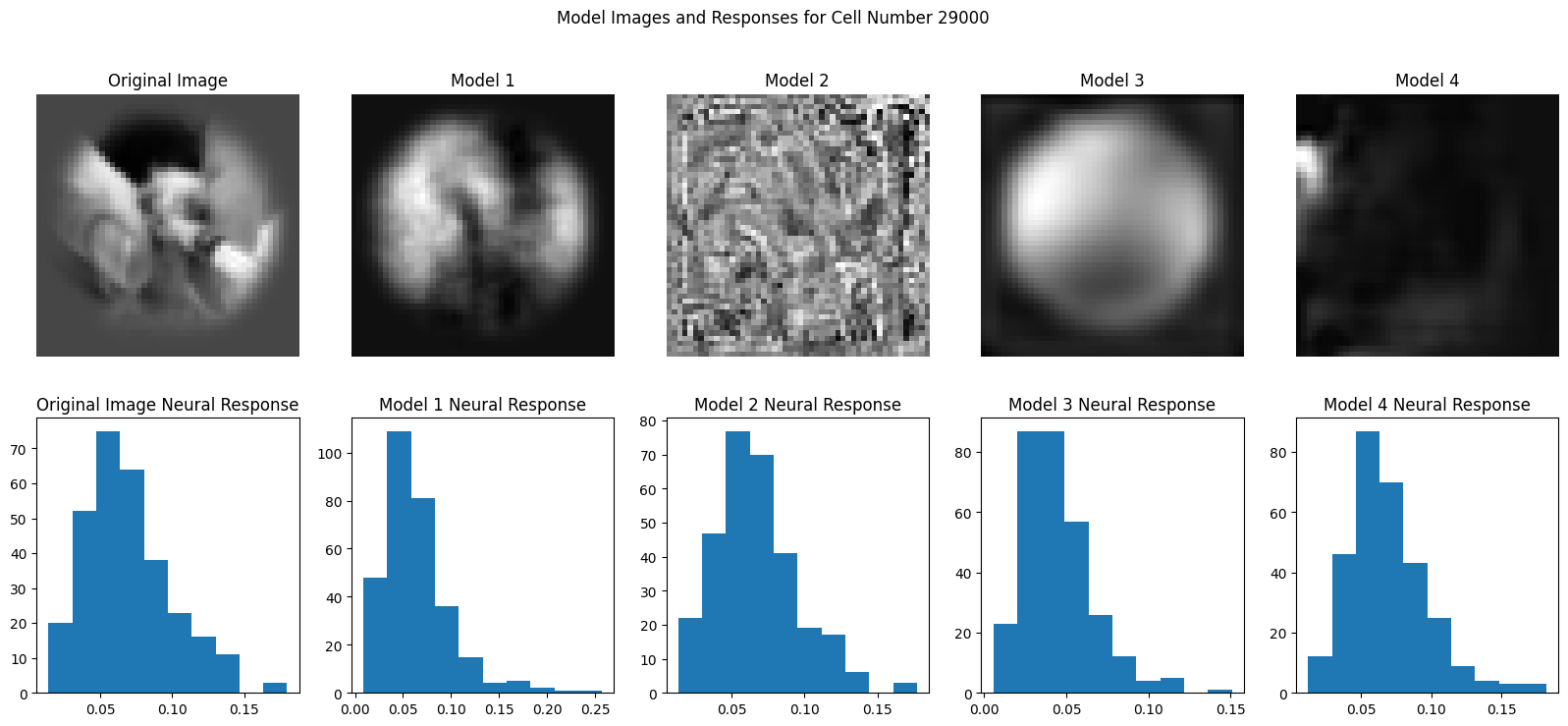
**Comparison of Models’ Results**

|  | 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*

****

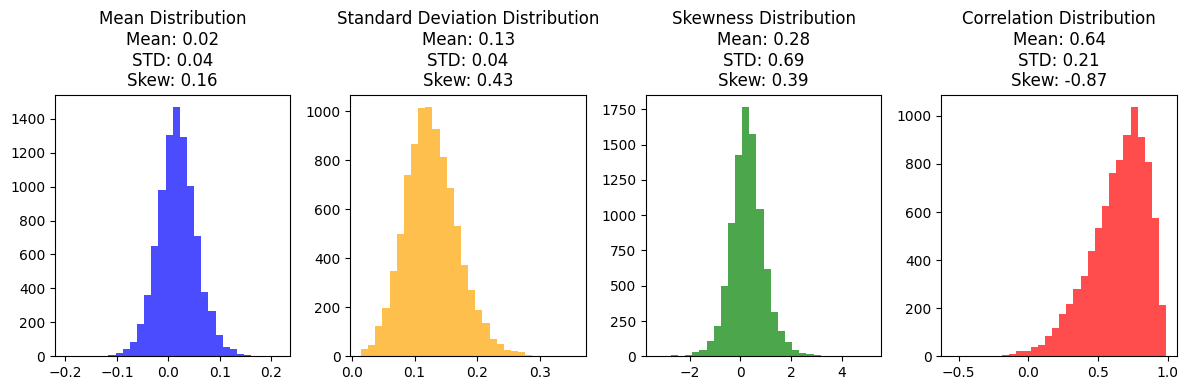
Presented from Left to Right are the respective models and their neural responses for Image Number 25000. Model one exhibits local variations that are similar to what is found in the original image – the white line emanating from the bottom right and moving toward the bottom center of the image is similar to the stem present in the original image. The region of darkness in the center is found in both images as well. Model 3 was able to generate the circle mask, but was limited in its clarity and pixel accuracy. Models 2 and 4 were not able to properly generate anything that looked like the original image. The neural responses across all of the models were similar to the original image’s neural responses. This may be due to either the loss function designed to match the neural responses (2 and 4) or the images’ pixel values closely match the original image, which inherently results in a similar perceptual response (1 and 3)

****

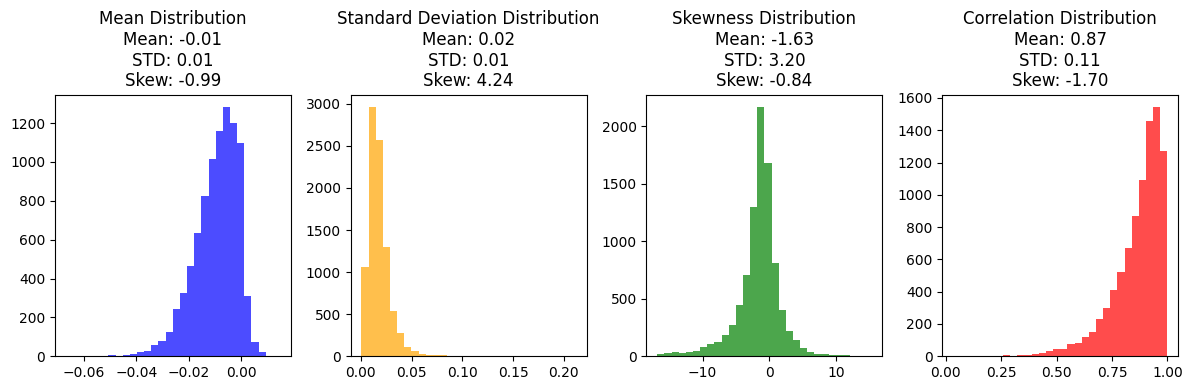
In these models, it is worth noting that Model 1 and 3’s image is not as accurate to the original image, though can generate local variations similar to the image (with the dark notch at the top. Interestingly, the models appear to be more similar to each other than the original image, which suggests that the generalizability completed during each neural network training may be correlated. The neural response histograms provided also are close to the neural responses from the original image. Specifically, Model 2’s neuron response histogram is almost identical to the original’s suggesting that the perceptual loss was successful.

**Model 1 Results**

*Pixel-by-Pixel Difference Distribution and Correlation*

**

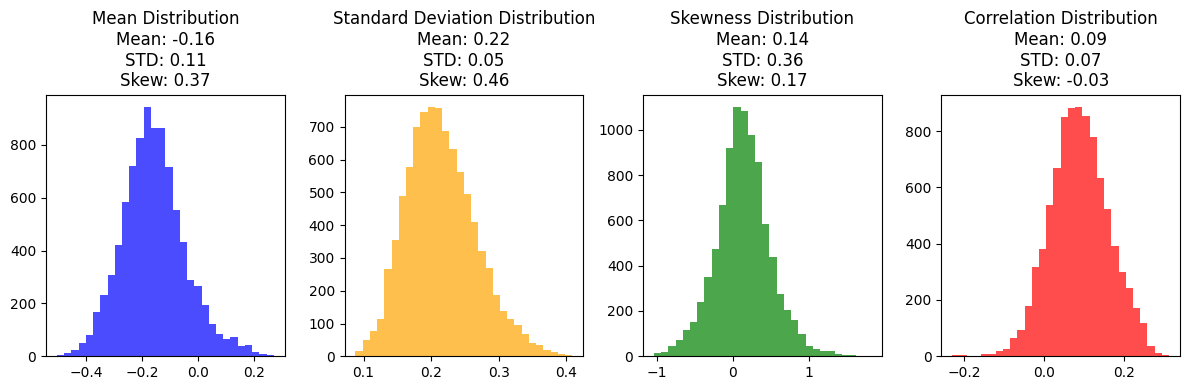
Provided here are the pixel-by-pixel differences’ distributions and the predictions’ correlation to the target for Model 1 for all 9000 images. The graphs appear to be regular, with a relatively low mean difference between the average pixel. Pixel values are between 0 and 1, so having a low mean and standard deviation suggests that the model is good at generating a strong result. The correlation is also extremely high here, with few models being negatively correlated.

*By-Neuron Difference Distribution and Correlation*

Provided here are the neuron-by-neuron differences’ distributions and the predicted’ correlation to the target for all 9000 image responses. The graphs again appear to be regular, with a relatively low mean difference between the average pixel. The correlation is also extremely high here, suggesting that there is a strong relationship between the neural results for the model and the target image.

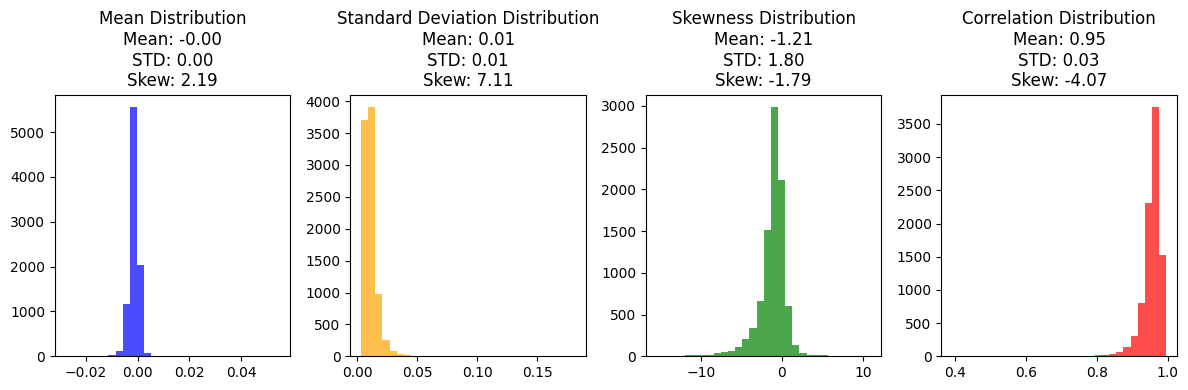
**Model 2 Results**

*Pixel-by-Pixel Difference Distribution and Correlation*

**

Provided here are the pixel-by-pixel differences’ distributions and the predictions’ correlation to the target for Model 2 for all 9000 images. The graphs again appear to be regular, with a significantly larger distance from the ideal 0.0, suggesting that the pixels do not align well. The correlation here is extremely poor, with no discernible image accuracy present

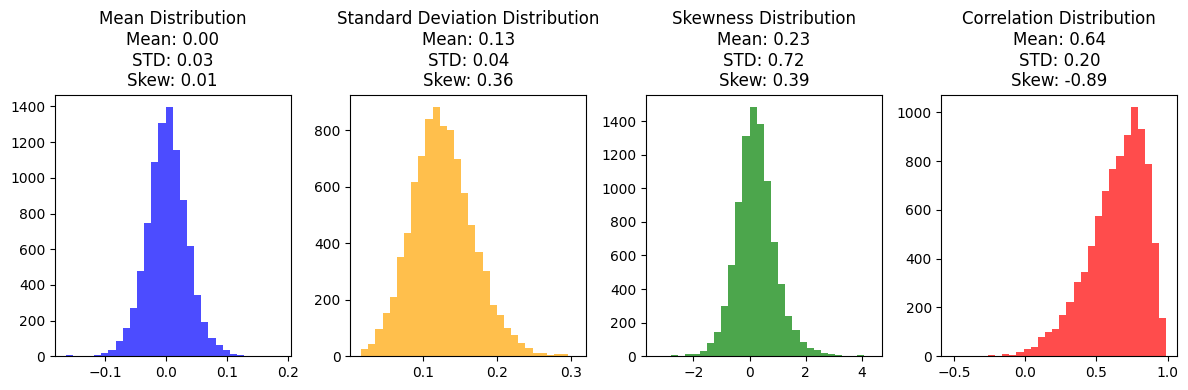
*By-Neuron Difference Distribution and Correlation*

**

Provided here are the neuron-by-neuron differences distributions and the predictions correlation to the target for all 9000 image responses. The graphs again show an extremely tight standard deviation, suggesting that the neuron’s responses are almost identical across the model. The correlation is also extremely high here, suggesting that there is a strong relationship between the neural results for the model and the target image.

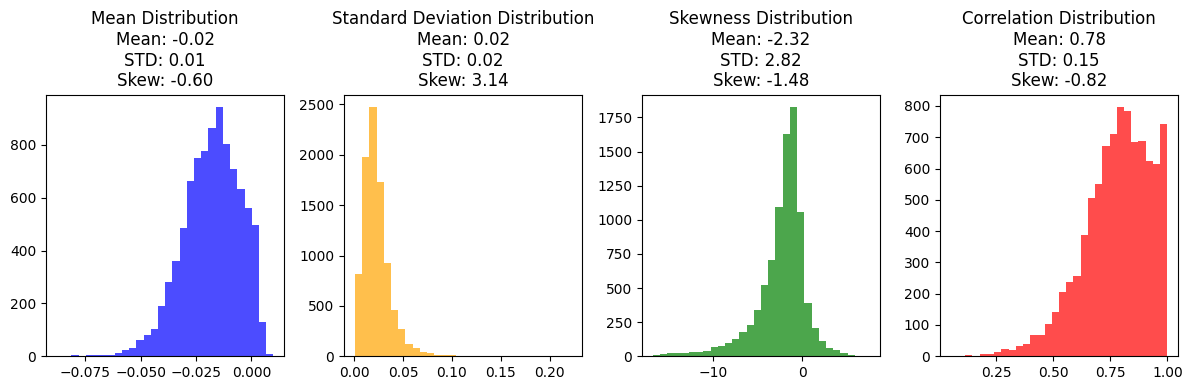
**Model 3 Results**

*Pixel-by-Pixel Difference Distribution and Correlation*

**

Provided here are the pixel-by-pixel differences’ distributions and the predictions' correlation to the target with respect to Model 3 for all 9000 images. The graphs appear to be regular, with a relatively low mean difference between the average pixel. The correlation is very similar to Model 1, here, with few models being negatively correlated, and suggests that this model is a strong representation of the original image, despite it not physically looking like it on the image.

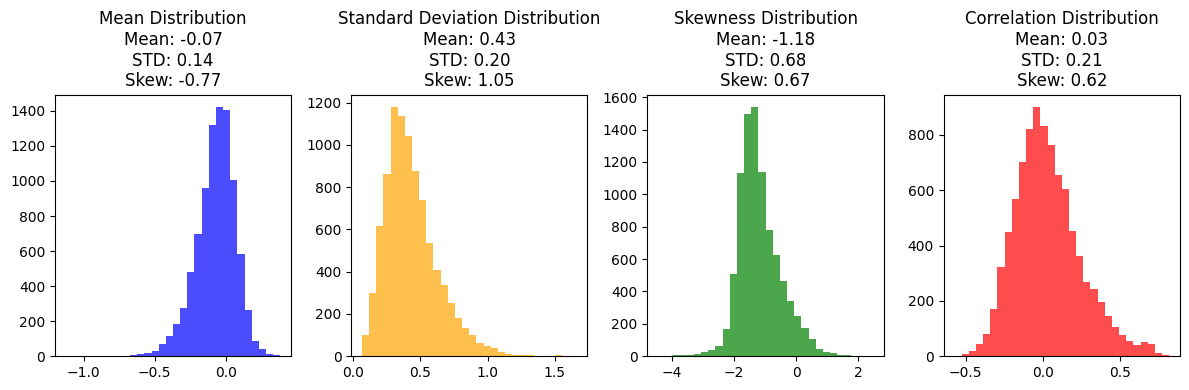
*By-Neuron Difference Distribution and Correlation*

**

Provided here are the neuron-by-neuron differences’ distributions and the predictions' correlation to the target for all 9000 image responses. These graphs better show the limitations of Model 3 when compared with Model 1, where the perceptual correlation is significantly lower and the perceptual per-neuron difference is not as accurate as Model 1. This may be due to the CNN's inability to generate local details.

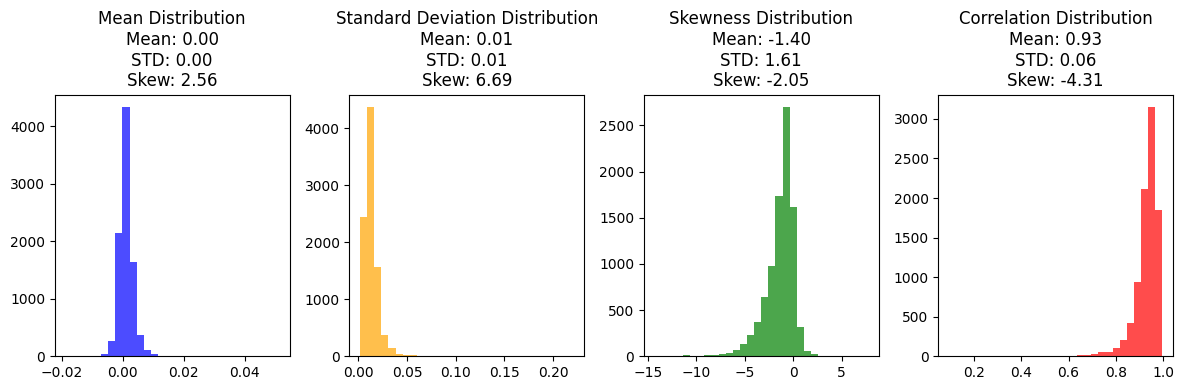
**Model 4 Results**

*Pixel-by-Pixel Difference Pixel-By-Pixel Correlation Distributions*

****

Provided here are the pixel-by-pixel differences distributions and the predictions correlation to the target with respect to Model 4 for all 9000 images. The graphs again appear to be regular, with a significantly larger distance from the ideal 0.0 than models 1 and 3, suggesting that the pixels do not align well between the target and the model’s prediction. The correlation here is extremely poor, with no discernible image accuracy present.

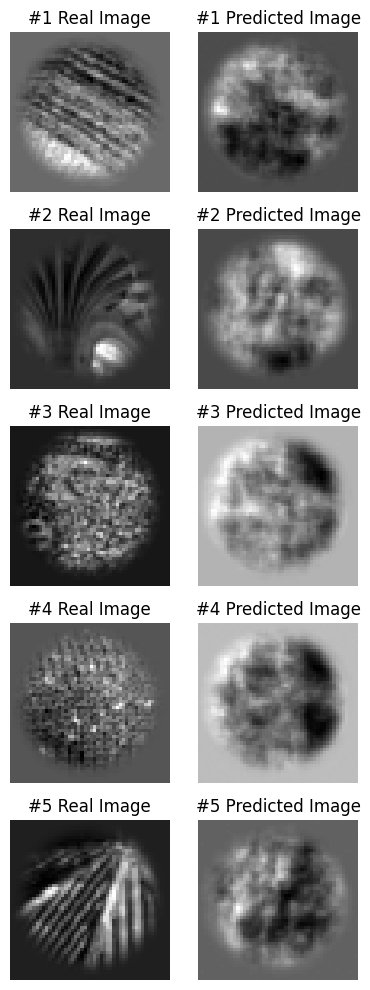
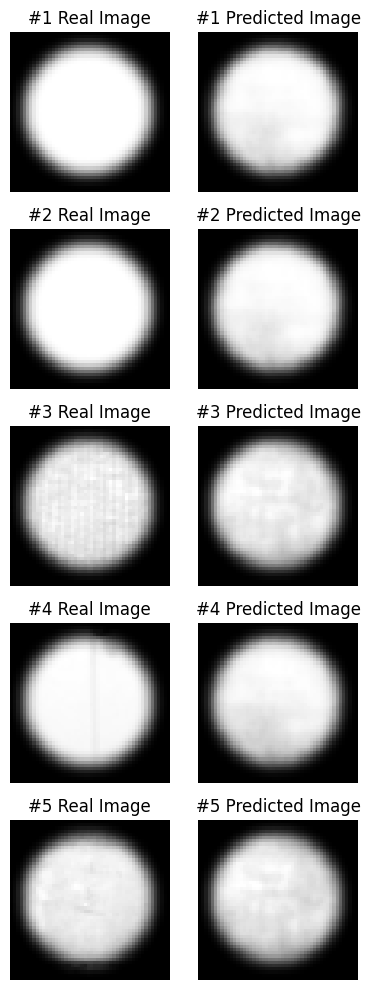
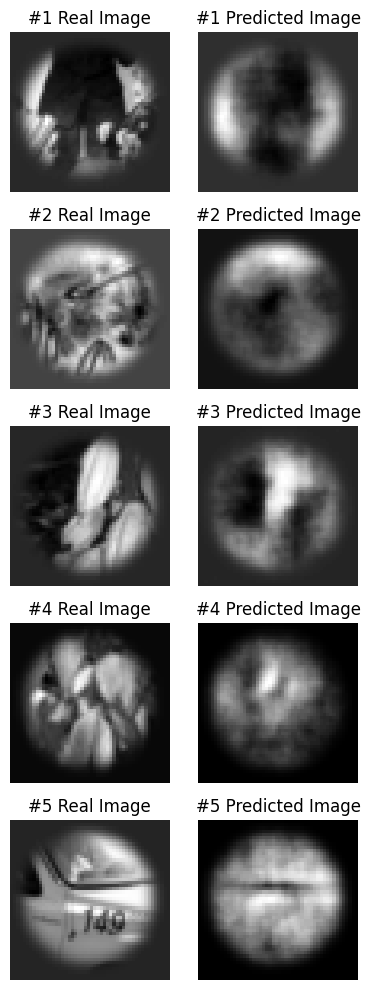
*By-Neuron Difference Distribution and Correlation*



Provided here are the neuron-by-neuron differences’ distributions and the prediction’s correlation to the target for all 9000 image responses. The graphs again show an extremely tight standard deviation, suggesting that the neuron’s responses are almost identical across the model. The correlation is also extremely high here, suggesting a strong relationship between the neural results for the model and the target image.

# 

# DISCUSSION

**Model 1: More Detail**

# 

Featured from Left to Right - Model 1: The Pictures with the best correlations (closest to 1), The Pictures with the correlations that closely match the mean (.64), the pictures with the worst correlation (closest to -1)

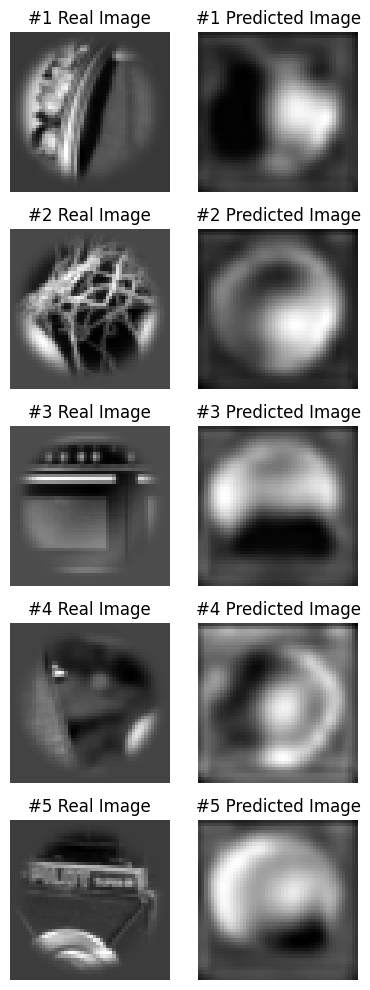
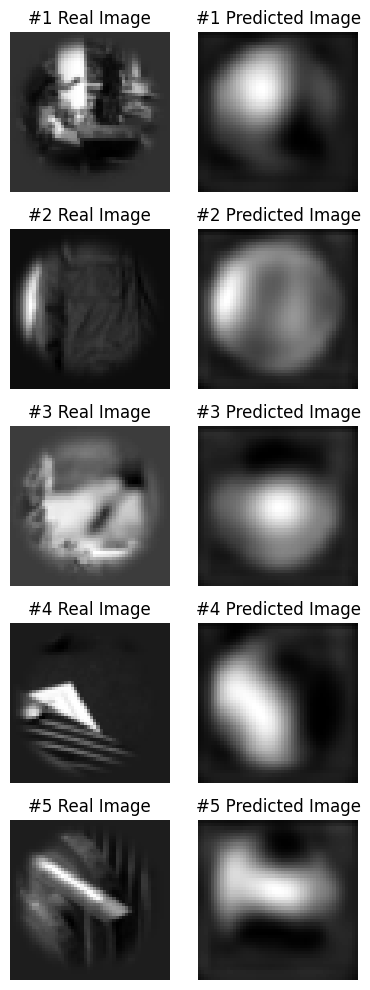
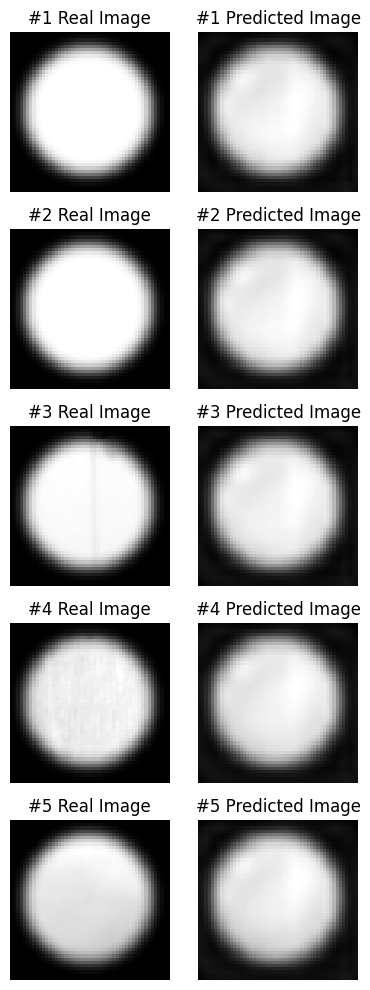
As previously stated, the Model 1 appears to be the most successful, with an extremely high correlation in both pixel-by-pixel and neuron-by-neuron comparisons. Looking into which images the model succeeds and fails is valuable for generalization.

Provided to the left are the top five images at which Model 1 was able to best correlate the pixel values. The target images themselves are solid white colors, which suggests that the model found it easiest to match one solid color as opposed to the more local variations found in most other images. This was also visualized often during training, where the model would show a clear preference for one color, and then learn more specific details as it was trained.

Provided in the middle are images that describe the average pixel-by-pixel correlation. In these images, global features are well represented, and even some local features are determinable (the line on the plane in the #5 real images). However, there are local elements that are not fully developed by the model.

On the right are the images that have the worst correlation and suggest that the model had trouble interpreting and including in its generalization. The worst correlations appear to be placed here because the model made the target image’s lighter regions dark. This is especially clear in #1’s image, where the dark spot at the bottom of the predicted image matches the light spot at the bottom of the target image.

Finding out the images that had the highest/lowest/most average perceptual correlation was not attempted due to time limitations. The generalizations suggested for Model 1 apply to Model 3, and are provided below



Featured from Left to Right - Model 3: The Pictures with the best correlations (closest to 1), The Pictures with the correlations that closely match the mean (.64), the pictures with the worst correlation (closest to -1)

**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.

For any of the four models, the random initial state of the image and the model parameters could result in the loss function reaching a local minimum during gradient descent. Ideally, the potential of the algorithm would be best presented when gradient descent reaches the global minimum.

One of the limitations of the multi-layered perceptron approach is the high quantity of weights and biases associated with turning 302 dimensions into 2500. This results in inefficient computations – where most weights are negligible values but still need to be computed and optimized – but more locally-defined predicted images. This is in direct contrast to the convolutional neural networks, where they were more efficient to train but less effective at generating results that mimicked the target 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.

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. 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.

It may also be helpful to normalize neuronal responses utilized during training to standardize the range of values. Normalization can play a crucial role in mitigating issues related to varying scales among neuron responses, ensuring a consistent and stable learning process, and fostering better performance when predicting images based on neuronal responses.

Additional statistics or modifications of currently available statistics may provide valuable insights into the validity and accuracy of these neural methods. In addition, running each neural network more than once will ensure that the algorithm’s results are reliable.

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 would like to express my sincere gratitude to Tianqin for dedicating time and effort to guiding me throughout this project. His valuable insights and assistance ensured that I completed complete and effective work while fostering a deeper understanding of the subject matter.

I extend my thanks to Professor Tai Sing Lee for his teaching on the fundamental concepts of computer vision and computational perception and for providing the opportunity to create my first series of neural networks.

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>

**Small Project Writeup (~7 pages):** https://docs.google.com/document/d/1G3wsihzUQ9NN8WO2HINKzYes\_vdiFfLUXEFCF40JTEE/edit