Deciphering Cognition: An Investigation into CNN Architectures

Justice Tomlinson (justice.tomlinson@mail.utoronto.ca)
Department of Cognitive Science, University of Toronto
Toronto, ON M5S1A1 Canada

Shysta Sehgal (shysta.sehgal@mail.utoronto.ca)
Department of Cognitive Science, University of Toronto
Toronto, ON M5S1A1 Canada

Abstract

This study explores the intersection of cognitive science, machine learning, and computer vision, by leveraging convolutional neural networks (CNNs) and highlighting their similarity to human cognition both in terms of structure and function. It delves into the depth versus width trade-off in CNNs, mirrored in human cognition's efficiency and depth of processing, as seen in various cognitive tasks including memory and learning. Leveraging MNIST and CIFAR-10 datasets, this research conducts two depth-width trade-off experiments to compare the learning capacity of CNNs with respect to human visual cognition models. Through a controlled experimental setup in which CNN depth and width are incrementally manipulated, this study aims to offer a nuanced understanding of how computational models reflect and can enhance our grasp of human cognition and ultimately finds that more shallow models demonstrate superior learning efficiency in shorter training sessions, particularly on more simple datasets.

Keywords: computer vision; machine learning; CNNs

Introduction and Related Work

In the exploration of cognitive science and its interdisciplinary approach to studying the human mind, the application of machine learning algorithms, particularly convolutional neural networks (CNNs), presents a compelling avenue for understanding and mimicking human cognitive processes (Van Uden, 2019; Xu & Vaziri-Pashkam, 2021; Guo et al., 2008). CNNs and the brain both employ hierarchical structures and processing, where different layers capture basic features and subsequent layers combine these into more complex representations (Meunier et al., 2009; Friston, 2008). Additionally, just as the brain adapts synaptic strengths based on experience, CNNs adjust their weights during training to improve task performance, mirroring the brain's learning process (Li et al., 2021).

In doing so, particular interest has been invested in examining the tradeoff in width vs. depth processing. In neuroscience, CNNs have been used to analyze EEG signals, performing various classification and generative tasks related to abnormal psychology (Suchetha et al., 2021; Da Silva, 2017). One of the most notable impacts of reserach on CNNs is on our understanding of memory and learning, influencing a wide range of theories and research areas. One such impact is on our understanding of memory and learning, where the level of processing theory posits that the depth at which information is processed affects its subsequent recall. As famously discussed by Craik and Lockhart (1990), information processed at a deeper level, such as through semantic processing, is more likely to be retained than information processed at a superficial level, such as through sensory input alone. A classic example is Sporer's seminal study on facial similarity, where faces that were learned through a deep processing task that required more complex processing to encode the information were more effectively recalled. However, most participants took considerably longer to process these rounds that required deep processing (Sporer, 1991).

A pertinent problem in both cognitive and computational frameworks is analyzing the costs and benefits to having a deeper neural network rather than a wider neural network. He et al. (2015) demonstrated that networks with increased depth significantly improved performance on image recognition tasks. They argued thus that deeper networks may be able to achieve higher levels of feature abstraction compared to more shallow exercises. This is somewhat analogous to complex cognitive processes in humans where processing stimuli more deeply enhances performance in classification and generative tasks. On the computational side, Tan and Le (2019) determined that by doubling the depth, the computational cost for training roughly squares, while doubling the width or resolution increases the cost linearly. Thus, there seems to be long running historical evidence that while deeper processing allows systems to achieve higher accuracy, it also increases the amount of time and effort required to learn within a particular task.

This final project embarks on an experimental journey to delve into the intricacies of pattern recognition and image classification through CNNs. By designing and conducting a series of depth-width tradeoff experiments, this project aims to not only showcase the capabilities of CNNs in learning hierarchical representations but also to draw parallels and insights into how such computational models might mirror aspects of human visual cognition. Through this endeavor, this study seeks to contribute to a broader discourse on the interplay between artificial intelligence and cognitive science, shedding light on how machines' ability to 'see' and 'understand' images can inform and be informed by our understanding of human cognition. We hypothesize that increasing the depth will enhance learning more effectively than increasing width. Deeper networks will produce overall higher accuracy than more shallow, wider networks.

Materials and Methods

This section describes the methodology used in the study, which should enable full replication of the analysis and findings. The code and data is available at https://github.com/JusticeTomlinson/CompVisNet.

Models

The study employed three configurations of CNNs, characterized as follows:

- One-layer CNNs: This configuration comprises a single convolutional layer followed by a pooling operation and concludes with a fully connected layer.
- Two-layer CNNs: This setup consists of two convolutional layers, each succeeded by a pooling operation, and ending with a fully connected layer.
- Three-layer CNNs: The most complex configuration in the study, it includes three convolutional layers, each followed by a pooling operation, and culminates in a fully connected layer.

The allocation of nodes across layers was standardized to ensure consistency in model complexity. As an example, if there are three nodes:

- In one-layer models, all nodes were contained within the single convolutional layer.
- Two-layer models distributed the three nodes such that the majority were in the first layer, with the remaining nodes in the second layer. This means that there are two nodes in the first layer and one node in the second layer.
- For three-layer models, nodes were evenly divided across all three layers. This means there is one node per layer.

The uniform distribution of nodes allows for controlled comparison across models, evaluating the impact of depth while maintaining consistent representational capacity.

Training Setup

The following training parameters were employed for all models:

- Epochs: 5
- Batch size: 64
- Learning rate: 0.001
- Optimizer: Adam

Each model was trained using cross-entropy loss.

Datasets and Preprocessing

The study utilized two benchmark datasets: MNIST and CIFAR-10. Both datasets underwent specific preprocessing steps to standardize input dimensions and normalize pixel values. Data loading was implemented in Python using the PyTorch library, with a batch size of 64 for both training and testing sets on both datasets. The data was shuffled during training to promote model generalization.

MNIST The MNIST dataset comprises 28x28 grayscale images of handwritten digits (0 through 9). For this study, the following preprocessing steps were applied to both training and testing images:

1. Resizing: Images were resized to 32x32 pixels to standardize input dimensions across different models.
2. Normalization: Pixel values were normalized to have a mean of 0.5 and a standard deviation of 0.5, transforming the original range from [0, 255] to [-1, 1].

CIFAR-10 The CIFAR-10 dataset consists of 32x32 color images spanning 10 classes, including animals and vehicles. The preprocessing applied to CIFAR-10 was similar to that of MNIST, with adjustments for color images: Pixel values for each color channel (red, green, and blue) were normalized to have a mean of 0.5 and a standard deviation of 0.5, thus standardizing the data range across all channels.

Model Evaluation and Performance Metrics

To comprehensively assess the performance of each CNN configuration, we utilized several key metrics throughout the training and testing phases. Specifically, the models were evaluated based on:

- Training Loss: The cross-entropy loss computed during the training process, providing a measure of how well the model fits the training data.
- Training Accuracy: The proportion of correctly classified images in the training set, evaluated at each epoch to monitor learning progress over time.
- Testing Accuracy: The proportion of correctly classified images in the test set, assessed after the completion of the training to evaluate the model's generalization to new data.

These metrics together allowed for a nuanced understanding of each model's learning dynamics, fitting accuracy, and generalization capability.

Results

This section showcases the findings from the comparative analysis of one-layer, two-layer, and three-layer CNNs using both the datasets. We did thorough evaluation of the models' performance over time and in relation to their complexity as measured by the number of parameters and nodes. We first report the findings on the CIFAR-10 dataset, then the MNIST dataset, and finally, we present the cross-dataset analysis.

CIFAR-10

For this dataset, model performance was evaluated over five epochs to understand how accuracy and loss evolved during the training process. We first averaged the model training performance per epoch for different layered models. For example, for different configurations of the one-layer models, we took the average accuracy and average loss on the first epoch, then the second epoch, the third epoch, the fourth epoch, and the fifth epoch. The line graphs for average accuracy and loss per epoch reveal that as the model depth increases, there is a noticeable trend: more complex models (with two and three hidden layers) generally show lower accuracy and higher loss across epochs compared to the one-layer models. Figure 1 illustrates this trend for the average training accuracy. This indicates that added complexity in terms of depth does not necessarily improve learning over time for image classification tasks. To understand this contextually, we must keep in mind how the node allocation was fixed across different layers.

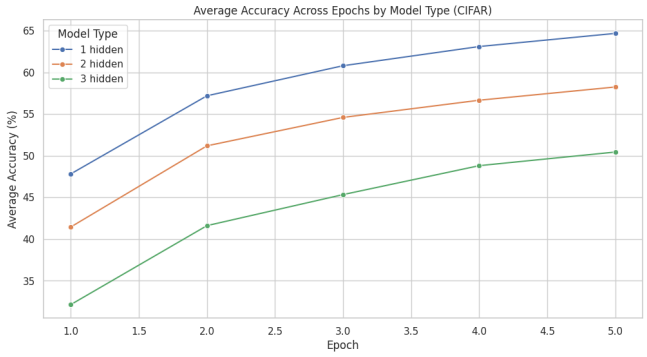

Figure 1: Average Accuracy across Epochs by Model Type on CIFAR-10

The heatmaps provide further insight into the models' efficiency, correlating the total number of parameters and nodes to the models' average training accuracy. From Figure 2, the data suggests that increasing the total number of nodes increases the average training accuracy for every model. However, with higher complexity, diminishing returns are observed. The one layer model with 48 nodes had the best training accuracy performance of 64.66%. The two layer model with 48 nodes (24 nodes in the first layer and 24 nodes in the second layer) had an average training accuracy of 61.12%, whereas the three layer model with the same number of nodes (16 nodes per layer) had an average training accuracy of 55.43%.

This may seem counter-intuitive; however, the heatmap on total number of parameters contextualizes these findings. There are more number of trainable parameters if the nodes are concentrated in one layer. However, the distribution of nodes across the layers leads to fewer number of total trainable paramters. We can observe from Figure 3 that one layer models with higher number of nodes had a lot more trainable

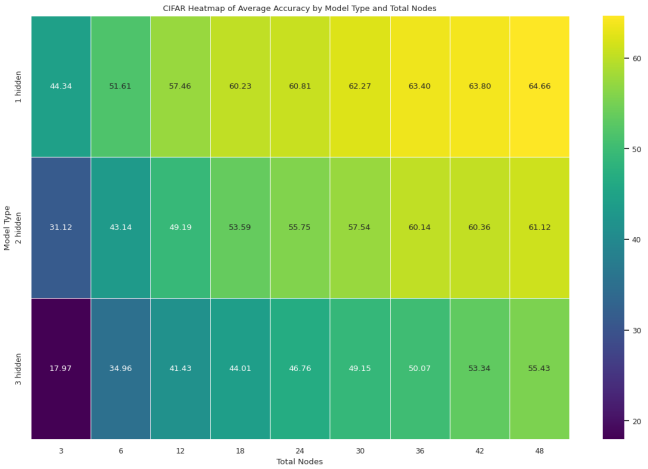

Figure 2: Heatmap of Average Accuracy by Model Type and Total Nodes on CIFAR-10.

parameters than the three layer models.

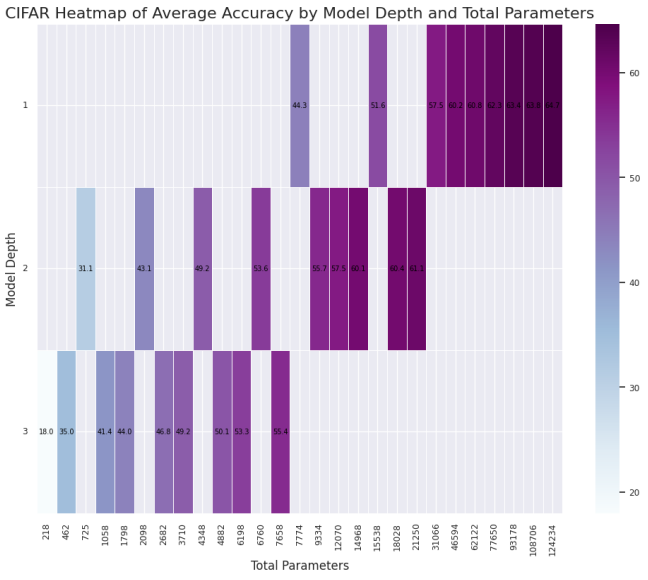

Figure 3: Heatmap of Average Accuracy by Model Type and Total Parameters on CIFAR-10.

Finally, the average test performance by model depth can be viewed in Table 1.

MNIST

We conducted similar analysis on the MNIST dataset. Higher training accuracies and lower losses were observed as compared to the CIFAR-10 dataset. This can be attributed to the fact that the CIFAR-10 dataset is more complex than the MNIST, and we kept the training paradigm fixed across both datasets despite the difference in their complexities to allow for a standardized comparison on the models across different datasets.

Table 1: Summary of Model Testing Metrics (CIFAR-10)

Model Type	Average Accuracy	Standard Deviation Accuracy	Average Loss	Standard Deviation Loss
1 hidden	60.714444	6.186813	1.130480	0.161384
2 hidden	57.247778	10.678360	1.212911	0.284231
3 hidden	50.480000	10.128981	1.384389	0.276265

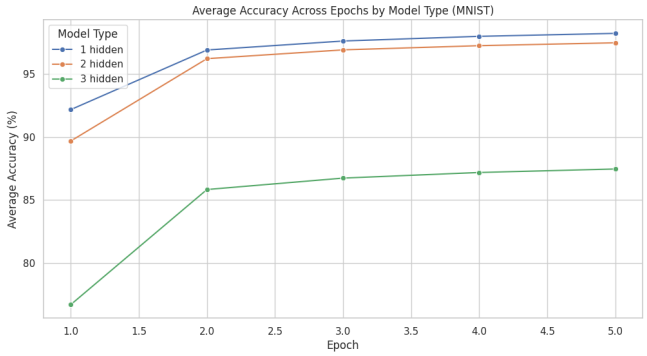

Figure 4: Average Accuracy across Epochs by Model Type on MNIST

We observe that both the one layer and two layer models with different configuration setups have superior performances to the three layer models. Additionally, there is a diminishing gap in the performance between one layer models and two layer models. However, one layer models still have better performance as illustrated in Figure 4.

Similar observations can be extracted with regards to the effect of total nodes and total parameters from the heatmaps for the MNIST dataset, which have been included in Appendix A.

Table 2 details the test performances for the MNIST dataset. In comparison to the previous dataset, we can observe that the one layer and two layer models had lower deviations from the average accuracy, whereas this deviation was greatly increased for the three layer models. This can be attributed to the three layer models with varying configurations to have certain outlier performances because, with greater nodes (but still fewer trainable parameters), the models eventually seem to be able to converge. However, with fewer nodes, the learning done by the model is insufficient. This does not seem to happen on CIFAR-10 because it is more complex and even with greater nodes, the three layer models might need more epochs to converge.

Cross-dataset Analysis

The examination of model performance across the MNIST and CIFAR-10 datasets provides a comprehensive view of how different architectures scale with respect to dataset complexity and image features.

As shown in Figures 5 and 6, we observed the best performances for one layer models with most nodes and most

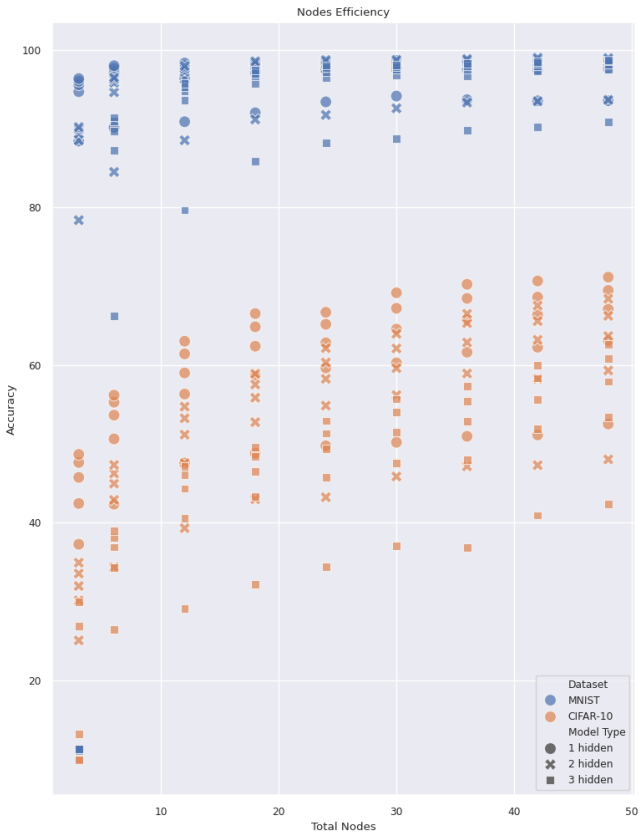

Figure 5: Nodes Efficiency

parameters; however, the increase in nodes and parameters after a certain point offered only diminishing returns. The performance is more varied and spread out for the CIFAR-10 dataset as compared to the MNIST dataset, indicative of its complexity along with potentiality for an alternative training paradigm for better convergence. We say this because the training paradigm used might not be optimal for this complex dataset and more epochs might needed for better performance results. This decision was solely made to directly evaluate model configurations, keeping everything else fixed.

Learning Curves The learning curves, formed by amalgamating the average accuracy per model depth per epoch per dataset (i.e., the average accuracy of one layer models on both datasets on epoch 1, then epoch 2, 3, 4, and 5 and so on for other models), indicate substantial performance increase in accuracy from epoch 1 to 2 and a steady incline later on. One

Table 2: Summary of Model Testing Metrics (MNIST)

Model Type	Average Accuracy	Standard Deviation Accuracy	Average Loss	Standard Deviation Loss
1 hidden	97.758889	0.513065	0.071508	0.017831
2 hidden	97.525556	2.590715	0.079317	0.087264
3 hidden	87.717778	28.723472	0.330241	0.742337

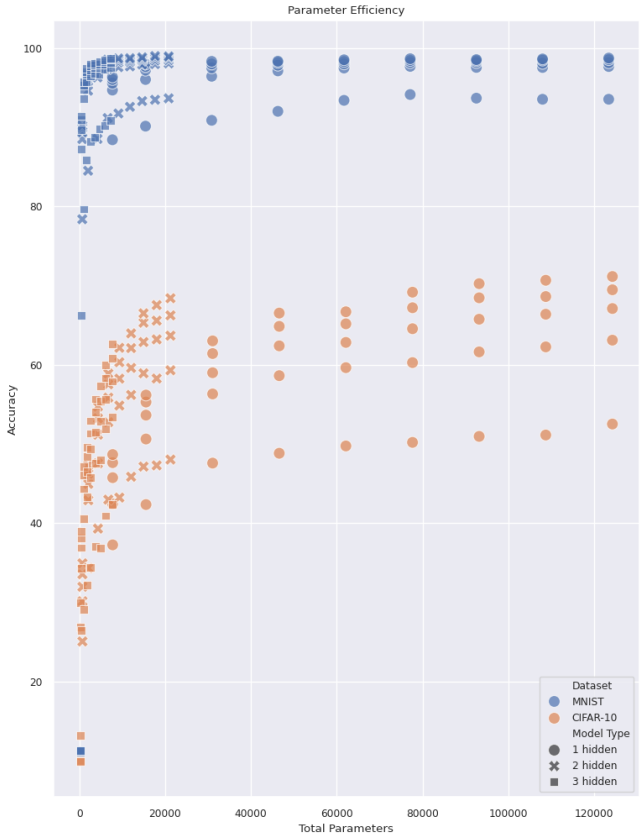

Figure 6: Parameter Efficiency

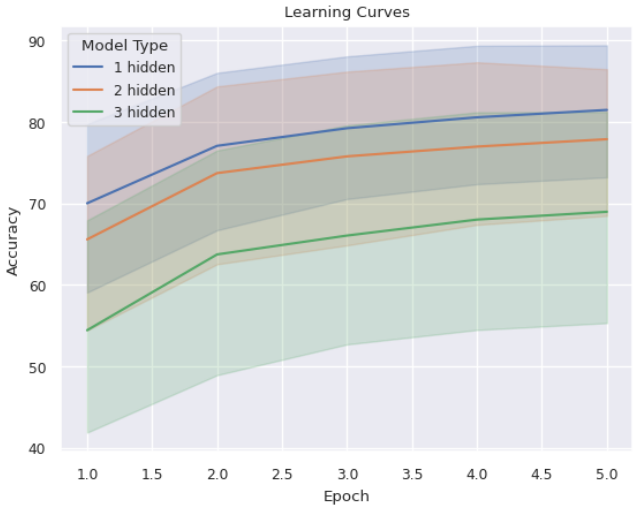

Figure 7: Learning Curves

layer models dominate, followed closely by two layer models. Three layer models lag behind the most, as seen in Figure 7.

Statistical Analysis To statistically examine the impact of model depth on learning efficiency, an Analysis of Variance (ANOVA) was conducted across the one-layer, two-layer, and three-layer models. The analysis revealed a significant effect of model depth on accuracy, $F(2, 267) = 9.07$, $p < 0.004$.

Further exploration through an Ordinary Least Squares (OLS) regression analysis assessed the combined influence of model depth and total nodes on accuracy. The regression results are summarized in Table 3.

The intercept, indicative of the baseline accuracy for one-layer models, was significant (coef = 72.4933, $p < 0.001$), suggesting that models with a single hidden layer start with a higher accuracy. The coefficient for two-layer models was not significant (coef = -7.1892, p = 0.280), indicating that moving from one to two layers does not significantly decrease the accuracy in the presence of node count. However, introducing a third layer significantly reduced accuracy (coef = -28.5854, $p < 0.001$), implying that additional depth beyond two layers might require more sophisticated training methods or additional data to be effective. The number of nodes alone had a positive but not statistically significant effect on accuracy (coef = 0.2119, p = 0.198), suggesting that simply increasing the number of nodes without considering layer depth might not be the most efficient strategy.

While our initial hypothesis posited that increasing model depth would unilaterally enhance learning efficiency, the OLS regression results challenge this assumption. The non-significant decrease in accuracy when adding a second hidden layer suggests that simply adding depth is not a guarantee of improved performance. In contrast, the significant decrease in accuracy with a third layer emphasizes inadequate learning when models become too complex without a corresponding increase in learning capability or data richness.

Interestingly, the significant interaction term for the three-layer models implies that distributing nodes across more layers does have a beneficial impact on accuracy, but only when these nodes are sufficiently numerous. This could point to the need for a balance between depth and breadth in neural network architecture, especially when considering the practical

Table 3: OLS Regression Analysis of Model Accuracy

Predictor	Coefficient	Std. Error	P-value
Intercept (1 layer)	72.49	4.69	<0.001
Model Depth (2 layers)	-7.19	6.64	0.280
Model Depth (3 layers)	-28.59	6.64	<0.001
Total Nodes (1 layer)	0.21	0.16	0.198
Interaction (2 layers:Total Nodes)	0.14	0.23	0.535
Interaction (3 layers:Total Nodes)	0.62	0.23	0.008

limitations of training data and computational resources.

These statistical findings shed light on why one-layer models consistently outperformed their more complex counterparts over the five epochs and why three-layer models lagged despite having similar total node counts. Such insights are vital for guiding future model architecture decisions and training paradigms.

Discussion and Conclusion

Our investigation into the depth versus width of CNNs has yielded some intriguing results that challenge the conventional emphasis on network depth as a primary driver of learning efficiency. Contrary to our initial hypothesis, our findings suggest that the learning efficiency of CNNs does not uniformly increase with network depth. Contrary to the views presented by He et al. (2015), who advocate for the superior capabilities of deeper networks, our results demonstrate that additional depth beyond may not provide the expected improvements in performance, especially under limited training conditions.

The limitations of this study, including the choice of datasets and the fixed duration of training, present opportunities for future research. For instance, the complexity of real-world tasks may necessitate deeper networks than those required for the MNIST and CIFAR-10 datasets. Furthermore, the potential of deeper networks might be more fully realized with longer training periods or with adaptive learning rates that reflect the intricacies of the learning task at hand.

Future work could explore the use of dynamic node allocation during training, where the network has the flexibility to shift nodes between layers depending on the learning demand. This approach would mimic neuroplasticity observed in human cognition, where neural resources are reallocated in response to task demands.

In conclusion, our research underscores the importance of tailoring network architecture to the specific demands of the learning task at hand, much like how human cognitive processing is adapted to task requirements. The interaction between model depth and total nodes on accuracy points towards a need for strategic consideration in neural network design. We believe that understanding the optimal architecture of CNNs not only propels the field of machine learning forward but also offers valuable insights into the mechanisms of human cognition. This work highlights the symbiotic potential between cognitive science and artificial intelligence, where insights from each field can mutually inform and refine the understanding of the other, pushing the boundaries of what machines can learn and, in turn, illuminating the complexities of the human mind.

References

Baldominos, A., Saez, Y., & Isasi, P. (2019). A survey of handwritten character recognition with mnist and emnist. *Applied Sciences*, *9*(15).

Da Silva, I. J., Vilao, C. O., Costa, A. H., & Bianchi, R. A. (2017). Towards robotic cognition using deep neural network applied in a goalkeeper robot. *2017 Latin American Robotics Symposium (LARS) and 2017 Brazilian Symposium on Robotics (SBR)*, 1–6.

Friston, K. (2008). Hierarchical models in the brain. *PLoS Computational Biology*, *4*(11). https://doi.org/e1000211

Guo, Y., Liu, Y., Bakker, E. M., Guo, Y., & Lew, M. S. (2018). Cnn-rnn: A large-scale hierarchical image classification framework. *Multimedia Tools and Applications*, *77*(8), 10251–10271.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Kriegeskorte, N. (2015). Deep neural networks: A new framework for modeling biological vision and brain information processing. *Annual Review of Vision Science*, *1*, 417–446. https://doi.org/10.1146/annurev-vision-082114-035447

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999–7019.

Lockhart, R. S., & Craik, F. I. (1990). Levels of processing: A retrospective commentary on a framework for memory research. *Canadian Journal of Psychology/Revue canadienne de psychologie*, *44*(1).

Meunier, D., Lambiotte, R., Fornito, A., Ersche, K., & Bullmore, E. T. (2009). Hierarchical modularity in hu-

man brain functional networks. *Frontiers in Neuroinformatics*, *3*. https://doi.org/571

Recht, B., Roelofs, R., Schmidt, L., & Shankar, V. (2018). Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, *2*(11).

Sporer, S. L. (1991). Deep—deeper—deepest? encoding strategies and the recognition of human faces. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*(2).

Suchetha, M., Madhumitha, R., & Sruthi, R. (2021). Sequential convolutional neural networks for classification of cognitive tasks from eeg signals. *Applied Soft Computing*, *111*.

Tan, M., & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *Proceedings of the 36th International Conference on Machine Learning*. https://doi.org/https://arxiv.org/abs/1905.11946

Van Uden, C. E. (2019). Comparing brain-like representations learned by vanilla, residual, and recurrent cnn architectures.

Xu, Y., & Vaziri-Pashkam, M. (2021). Limits to visual representational correspondence between convolutional neural networks and the human brain. *Nature Communications*, *12*(1). https://doi.org/2065

Appendix A

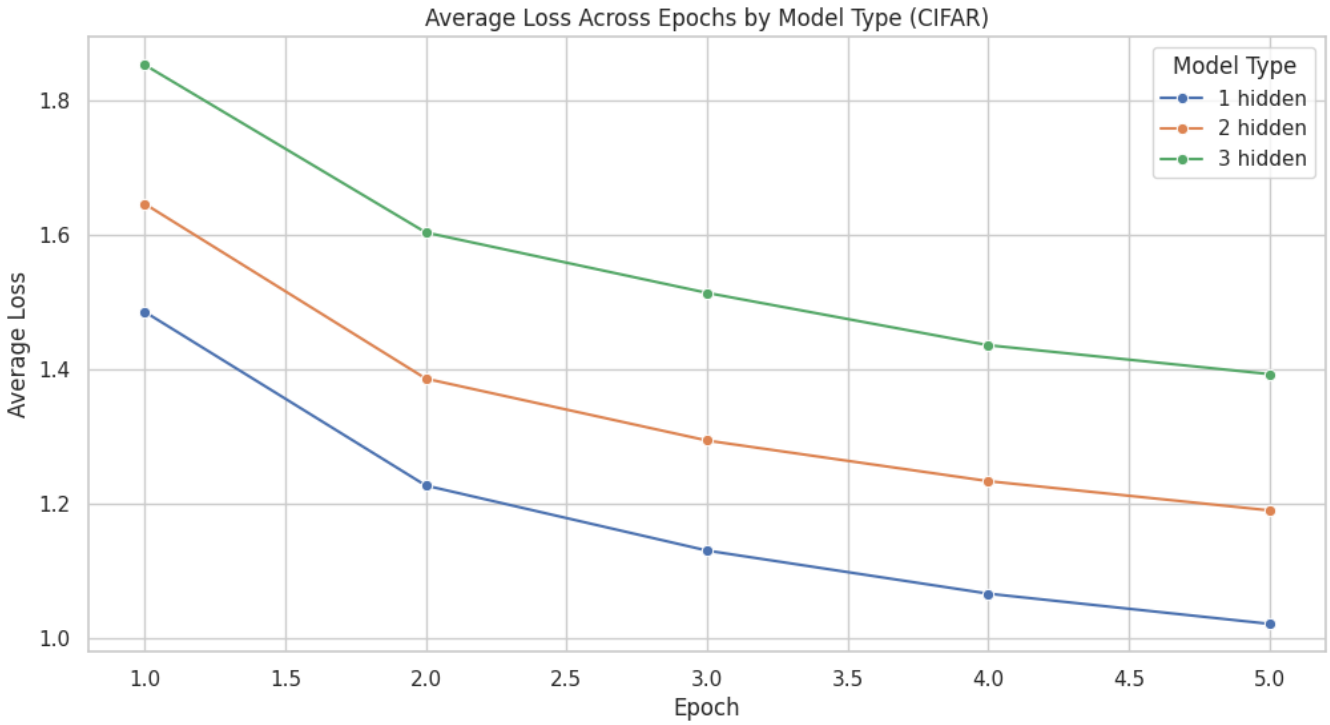

Figure A1: Average Loss Across Epochs by Model Type on CIFAR-10

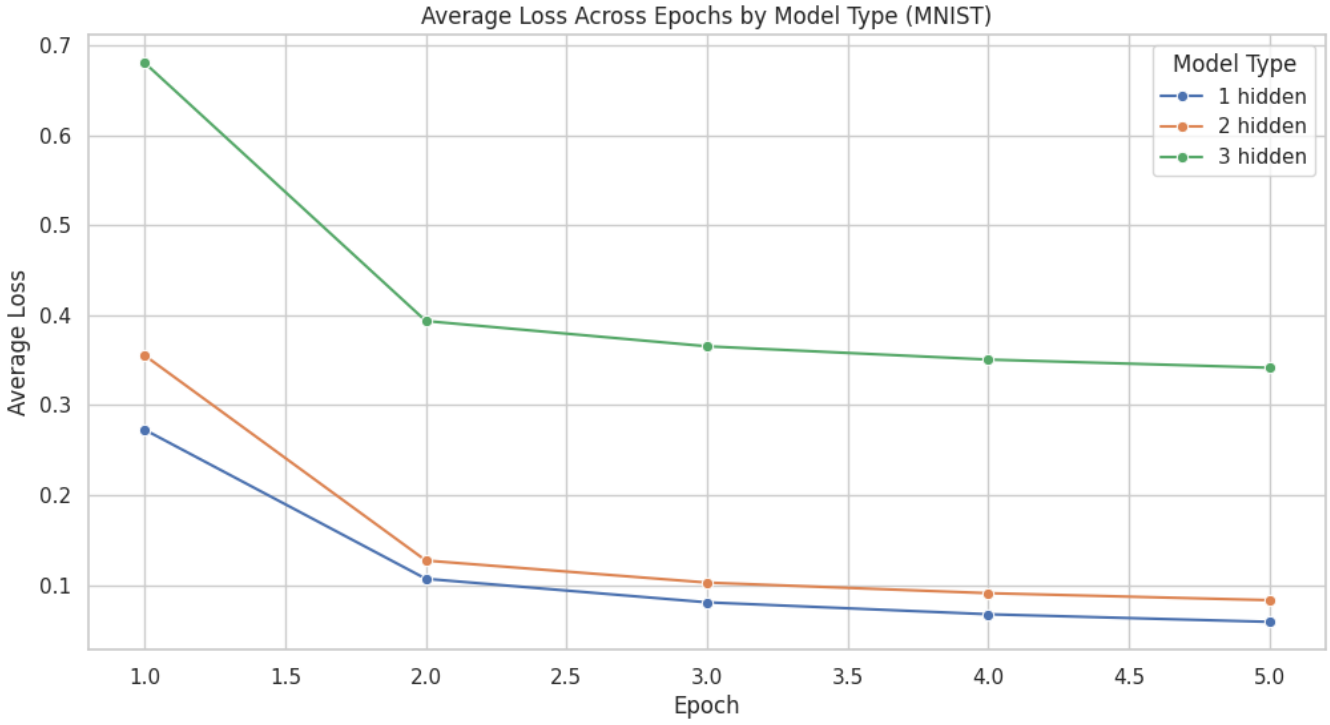

Figure A2: Average Loss Across Epochs by Model Type on MNIST

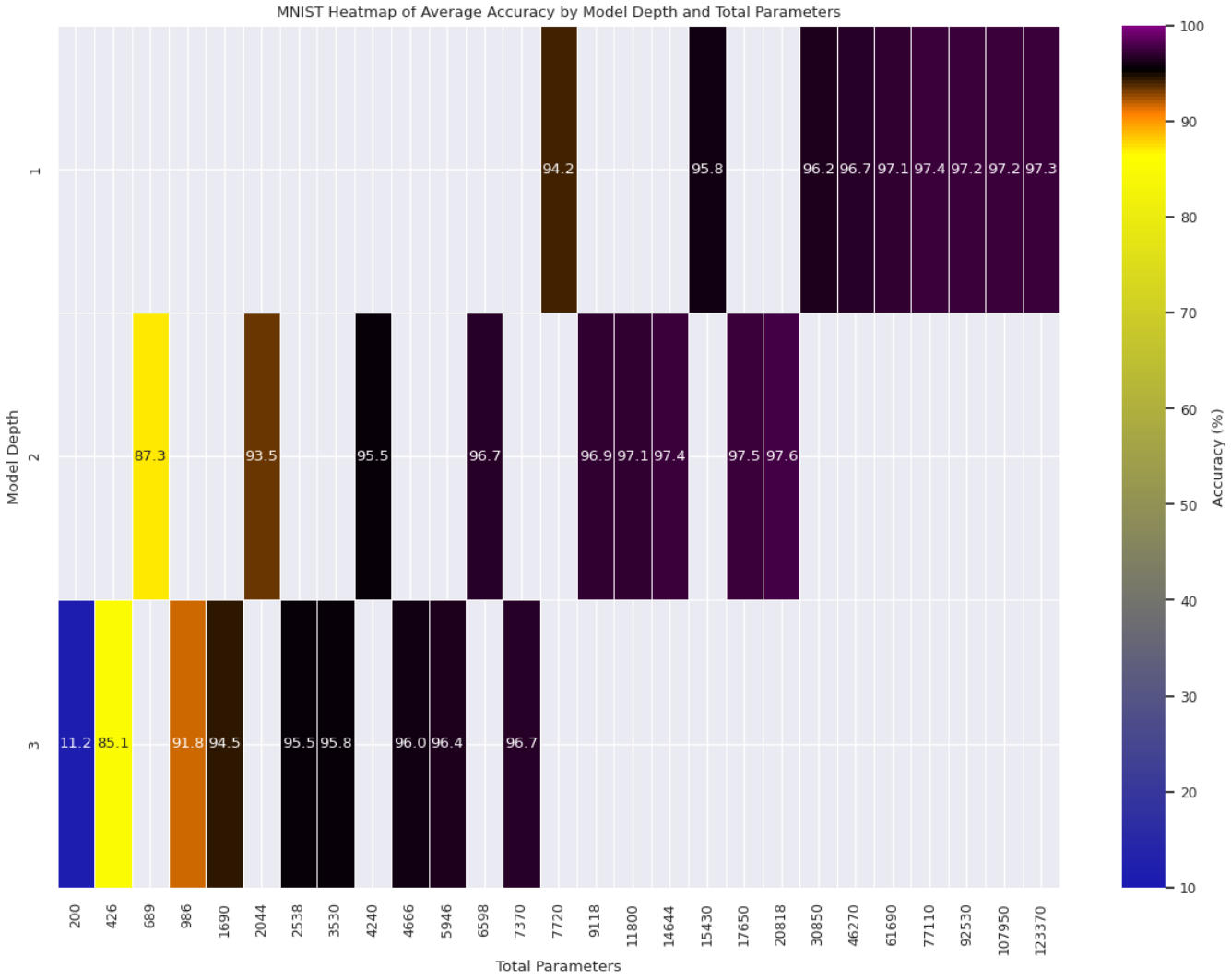

Figure A3: Heatmap of Average Accuracy by Model Type and Total Parameters on MNIST

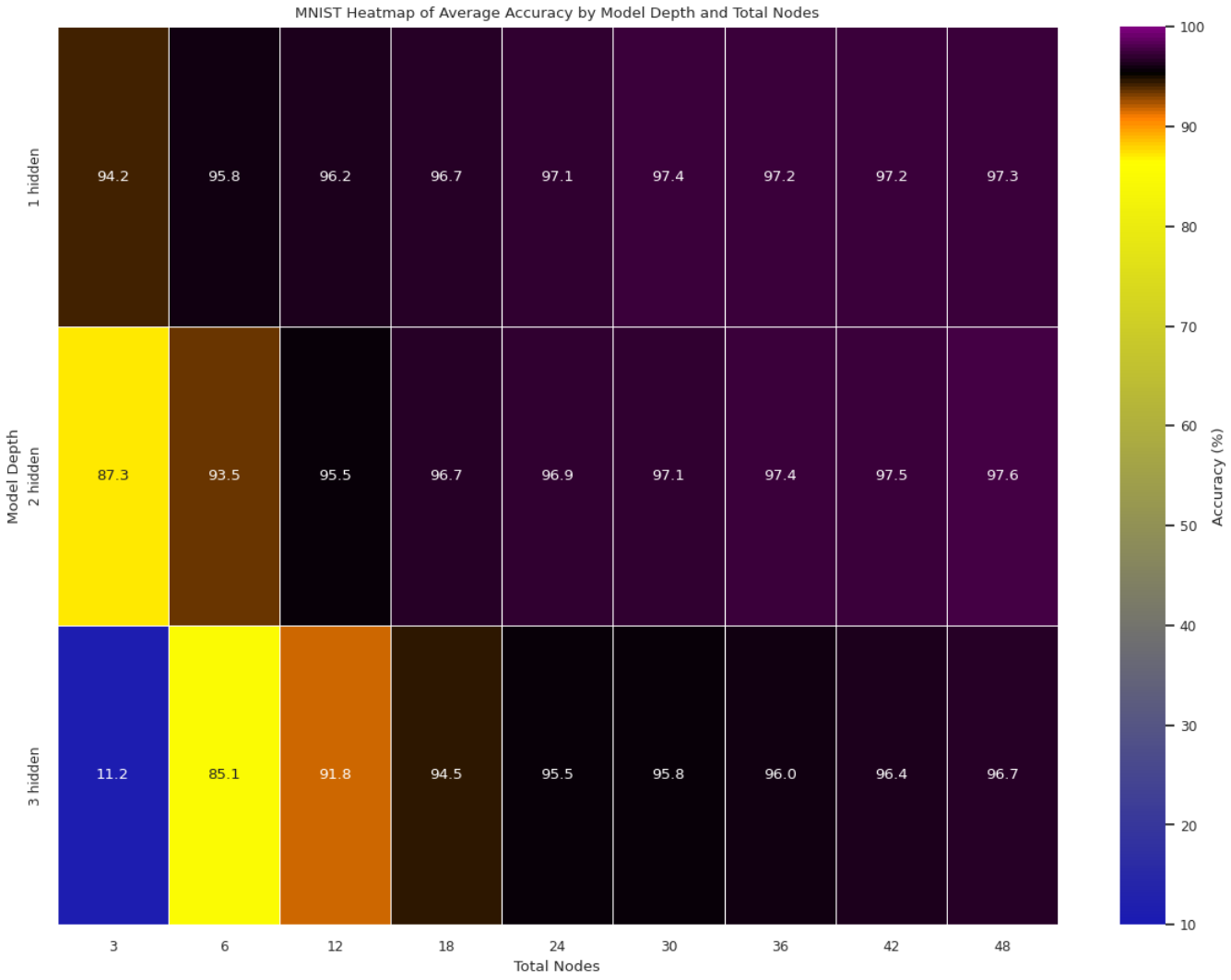

Figure A4: Heatmap of Average Accuracy by Model Type and Total Nodes on MNIST

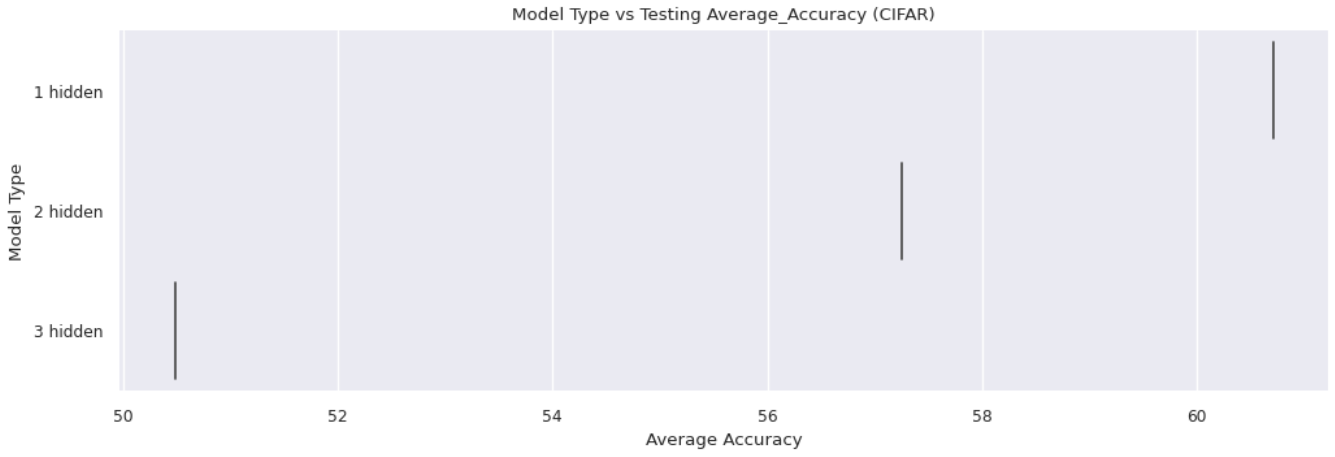

Figure A5: Model Type vs Average Test Accuracy for CIFAR-1O

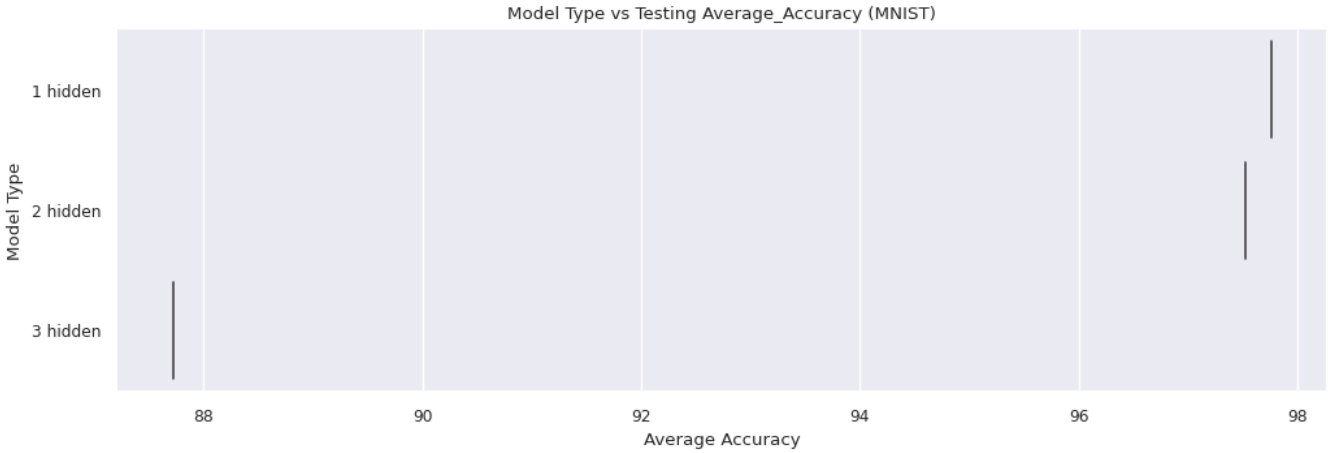

Figure A6: Model Type vs Average Test Accuracy for MNIST