2022 12th Iranian/Second International Conference on Machine Vision and Image Processing (MVIP)
Department of Electrical and Computer Engineering, Shahid Chamran University of Ahvaz, Iran, 23& 24 February 2022

2022 12th Iranian/Second International Conference on Machine Vision and Image Processing (MVIP) | 978-1-6654-1216-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/MVIP53647.2022.9738741

Exploring the Properties and Evolution of Neural Network Eigenspaces during Training

1st Mats Leon Richter
Institute of Cognitive Science
University of Osnabrueck
Osnabrueck, Germany
matrichter@uni-osnabrueck.de

2nd Leila Malihi
Institute of Cognitive Science
University of Osnabrueck
Osnabrueck, Germany
lmalihi@uni-osnabrueck.de

3rd Anne-Kathrin Patricia Windler
Institute of Cognitive Science
University of Osnabrueck
Osnabrueck, Germany
awindler@uni-osnabrueck.de

4th Ulf Krumnack
Institute of Cognitive Science
University of Osnabrueck
Osnabrueck, Germany
krumnack@uni-osnabrueck.de

We investigate properties and the evolution of the emergent inference process inside neural networks using layer saturation [1] and logistic regression probes [2]. We demonstrate that the difficulty of a problem, defined by the number of classes and complexity of the visual domain, as well as the number of parameters in neural network layers affect the predictive performance in an antagonistic manner. We further show that this relationship can be measured using saturation. This opens the possibility of detecting over- and under-parameterization of neural networks. We further show that the observed effects are independent of previously reported pathological patterns like the "tail pattern" described in [1]. Finally, we study the emergence of saturation patterns during training, showing that saturation patterns emerge early during training. This allows for early analysis and potentially increased cycle-time during experiments.

***Index Terms*—convolutional neural networks, logistic regression probes, saturation**

I. Introduction

The problem of opaqueness of neural networks is one of the key challenges in deep learning. This has lead to a primarily trial-and-error driven mode of development, based on the comparison of abstract metrics that capture model-agnostic concepts like predictive performance, demand in computational resources and capacity [3]–[9]. To move towards a more efficient, principle-based design process, a more profound understanding of the model's state is required. This understanding does not have to be necessarily complete in regard to fully understanding the relation of the input and output of the model. Comparative analysis methods based on SVCCA [10] are good examples of such non-holistic approaches. The information extracted from the model using SVCCA is highly aggregated, but allows for useful insights into the converged model and the training process, by comparing different layers within that model, as well as the states of one layer at different training epochs. Logistic regression probes [2] and saturation [1] aggregate a single layer to a number, which allows for easy and intuitive analysis, similar to measuring with a thermometer. While logistic regression probes measure the intermediate solution quality very directly by training logistic regressions on the output of a layer, saturation is more task agnostic. Richter et al. show that for visual classification tasks, the size of the subspace of the feature space[1] responsible for data processing varies significantly depending on the input resolution, leading to model and training inefficiencies [1]. While saturation is easy to define, the exact properties of this metric are yet to be discovered. In this paper we will address the following questions to gain a better understanding of hidden layer saturation:

- How do model capacity and problem difficulty influence the saturation value? Answer: We show that these two parameters have opposite effects on the saturation value.
- Is the organization of the inference process influenced by the capacity of individual layers or the capacity of the entire network? Answer: No, the tested models seem to be unable to shift processing to other layers, when some layers have substantially higher or lower capacity.
- How do saturation patterns evolve in the training phase? Answer: They converge at a similar pace as the loss of the model, with saturation increasing during training. The way saturation evolves also gives hints on the properties of the dataset, but it is not influenced by the model overfitting.

The remainder of the paper is structured as follows: Section II introduces the main methods applied in our work. Sections III, V and IV present the experiments we conducted to support our

[1]We refer to feature space as the space in which the data exists in a specific layer's output.

claims. Section VI concludes the paper with a summary of the results.

II. Related Work and Methodology

A. Logistic Regression Probes

Logistic regression probes, in this work abbreviated as *probes*, are a tool proposed by Alain and Bengio [2] used as a "thermometer-style"-scalar metric for analyzing the intermediate solution quality during the forward pass. They are obtained by training a simple logistic regression model on the same task as the original neural network, however, using the layer's output values as input data for training. Hence, the probe performance can be considered as a measure of the linear separability of the target classes in the layer's output representations. As the neural network's softmax layer and the logistic regression probe both minimize cross-entropy, both solve effectively the same task. Therefore, we can use the test accuracy of the logistic regression relative to the model's predictive performance to judge the intermediate solution quality. The logistic regression probe performance should increase monotonically from early to later layers of the network, approaching the predictive performance of the model towards the final layers. Such a development implies that all layers contribute qualitatively to the inference process. Logistic regression probe performance is visualized as a curve with individual measuring points (network layers) arranged in the same order as the data is flowing through the network during a forward pass. An example of this can be seen in figure 1. In this paper, we refer to the test accuracy of a probe computed on the output of layer l as p_l.

B. Saturation

The saturation s_l of a layer l is a simple scalar metric that was first introduced by Shenk et al. [11] and Richter et al. [1]. It can be computed for any layer in a neural network based on the layer's output values. It measures how many of the available dimensions in the output space Z_l of the layer l are relevant for the inference process:

$$s_l = \frac{\dim E_l^k}{\dim Z_l} \tag{1}$$

Saturation is computed by approximating the ratio of the dimensionality of the relevant eigenspace $\dim E_l^k$ of layer l and the extrinsic dimensionality of that layer's activation values $\dim Z_l$. The relevant eigenspace E_l^k is a subspace of Z_l in which the information is processed. This space is referred to as "relevant" because a projection of the data into the relevant eigenspace will not lead to a loss of predictive performance [1]. The relevant eigenspace can be considered the subspace in which the information processing is happening. The approximation of E_l^k is done using PCA [12], where the largest eigendirections are kept in order to explain 99% of the data's variance in the output of layer l. This technique allows computing saturation on-line during training. In contrast, evaluating logistic regression probes may take significant extra time, as it requires extraction of the activation values and the additional training of the probes.

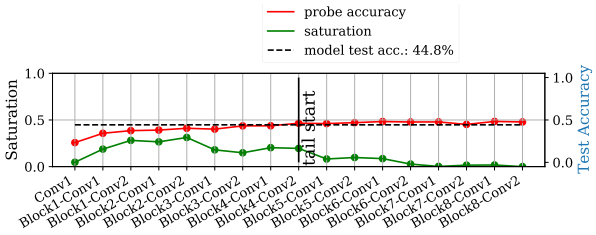


Fig. 1. An example of a tail pattern on a trained ResNet18 model. The tail is starting on the layer with the black border. Tail patterns can be identified by low saturation and stagnation of the logistic regression probe performance. Layers of the tail are no longer improving on the intermediate solution quality. For this reason, these layers can be considered a parameter-inefficiency. The model is trained on ImageNette at 32×32 pixel input resolution.

C. The Semantics of Saturation

A sequence of low saturated layers ($< 50\%$ of the average saturation of all other layers) is referred to as a "tail pattern" and indicates that these layers are not contributing qualitatively to the prediction (see figure 1). This is also visible when observing the performance of the probes, which stagnates in the tail-layers.

This suggests that solving a problem saturates the layer more than simply passing through information. However, this does not mean that the absolute saturation value is indicative of the activity within a layer. So far, saturation has been only explored as a quicker on-line computable alternative for logistic regression probes. As such, saturation has always been viewed relative to other layers within a neural network. In this work, we will explore how the absolute saturation value changes in different scenarios. We further explore how saturation evolves during training, and we will explain the low dimensionality of the tail pattern.

III. Capacity and Problem Difficulty Behave Proportionally

In this section, we analyze the relationship between problem difficulty and model capacity in two experiments, exploring how this relationship is reflected in the saturation values. In our experiments, we train the entire VGG-network family (VGG11, 13, 16 and, 19) on Cifar10 [13] and reduce their capacity evenly over the entire architecture to observe how these reductions affect the saturation values. Our first hypothesis states that the average saturation s_μ increases proportionally with a reduction in capacity while the model performance decreases. We then move on to investigate how the problem difficulty changes the saturation emerging in a neural architecture. Since the relevant eigenspace is generally larger when the layer is contributing to the quality of the solution [1], we further hypothesize that more processing in a layer requires a larger relevant eigenspace. If this assumption holds true, the overall saturation level should increase with an increase in the difficulty of the task. If both working hypotheses are true, we can conclude that the difficulty of the problem and the capacity of the layers influence saturation in an antagonistic way.

A. Methodology

We test our working hypotheses by conducting two experiments. We first train the VGG-family [6] of networks on Cifar10. We further train 4 additional variants of each model, which have the respective number of filters (and thus capacity) reduced by a factor of $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$ and $\frac{1}{16}$. We choose Cifar10 for its manageable size, which allows for a larger number of model training runs to be conducted with our available resources, which is necessary for this experiment. We choose the VGG-family of networks for its architectural simplicity and because we can test different depths of convolutional neural networks by experimenting on the entire family of networks. The training itself is conducted using a stochastic gradient descent (SGD) optimizer with a learning rate of 0.1, which is decaying after 10 epochs with a decay factor of 0.1. The models are trained on a batch size of 64 for 30 epochs in total.

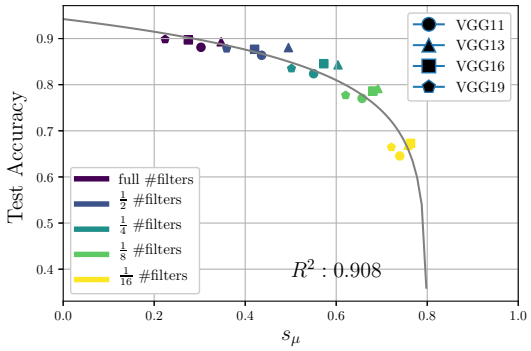
The second experiment is conducted on ResNet18. However, we are using a standardized input resolution of 224×224 pixels, to avoid artifacts caused by the input resolution. We train the model on multiple datasets of different difficulties (in ascending order of complexity): MNIST, Cifar10, TinyImageNet, and the ImageNet dataset [13]–[16]. While it is hard to precisely define the complexity of the task, we think the selected datasets can be regarded as increasingly difficult based on the number of classes and the complexity of the visual information provided as data points to the model. The resolution of MNIST binary images is 28×28 pixel. That is suitable for the 10-class classification problem. Cifar10's features are RGB images with a 32×32 resolution. That is suitable for the 10-class classification problem as well. TinyImageNet consists of RGB images of size 64×64 with 200 classes, and ImageNet is made up of RGB images of various sizes belonging to 1,000 classes.

B. Results

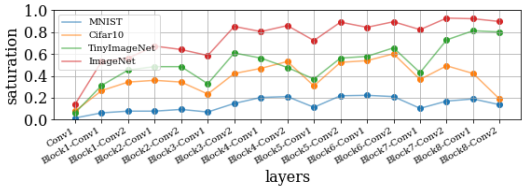
When the capacity of the model is reduced, the average saturation s_μ increases, and the predictive performance decreases. The exponential reduction in capacity is reflected in a logarithmic relation between the increasing s_μ and predictive accuracy measured on the test set (see figure 2). From these observations, we can conclude that reducing the capacity of the architecture results in an increase in saturation.

We further observe in figure 2 that saturation also increases with problem complexity. The saturation levels of all layers increase when the model is trained on a more difficult problem. The overall shape of the saturation curve only deviates slightly, with no tail pattern or similar anomalous shapes emerging. Since we know from the works of [1] that a resolution of 224×224 pixels results in an even distribution of the inference process for the trained model, we can conclude that less processing is required for less complex problems.

Combined with the insights gained from training the VGG-variants on Cifar10, we can conclude that for the pairs of dataset and model in our experiments a saturation "sweet spot" exists between s_μ = 0.2 and s_μ = 0.4, which yields the good predictive performance without being too excessively overparameterized.



(a) Accuracy and saturation with varying model capacity



(b) ResNet18 saturaion curves for different datasets

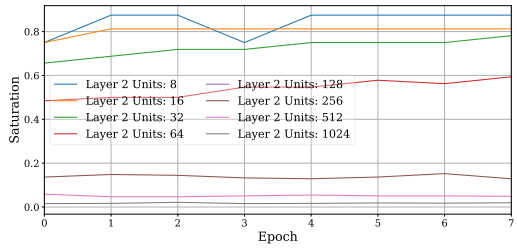
Fig. 2. Relationship of network saturation to model capacity and data complexity: (a) Reducing the number of filters and thus reducing the model capacity leads to an increase in the average saturation and a decrease in performance. (b) Training a model on more difficult datasets also increases the overall saturation level. This indicates that saturation can measure the load on a ResNet18 model.

IV. On the Emergence of Saturation Patterns

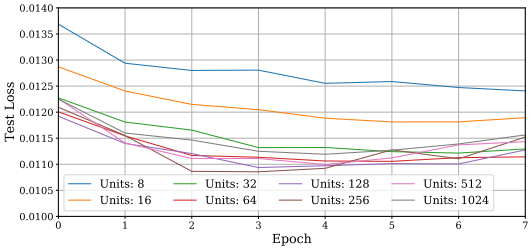
The tail pattern that we discussed earlier in this work allows for the identification of inefficiencies caused by mismatches between the neural architecture and the input resolution. However, since saturation can be computed live during training with little overhead [1], we think that it might be interesting to see how these patterns emerge during the training process.

A. Methodology

We first examine how the saturation levels evolve in a layer under different conditions. We train a set of neural networks with 3 fully connected layers. The first layer has 256 units, and the size of the second layer varies for each network, being in the range of 8, 16, 32, 64, 128, 256, 512, and 1024 units. We train these networks using the ADAM optimizer and a batch size of 128 on Cifar10 using the native resolution of the dataset. The training is conducted twice. Once using 8 epochs, which is enough for all models to converge, whereas the second experiment is run for 30 epochs, which results in the loss increasing again due to overfitting. The hidden layer saturation will be calculated after each epoch for observing the evolution of the architecture. We also calculate the cross-entropy loss of the model to observe a possible relationship between loss and saturation convergence. Based on these observations, we repeat the experiment on VGG11 and VGG19 as well as sparse (low capacity) versions of these models with $\frac{1}{8}$ of the original number of filters. We do this to understand if saturation patterns depend on the depth, architecture, and capacity of the network.

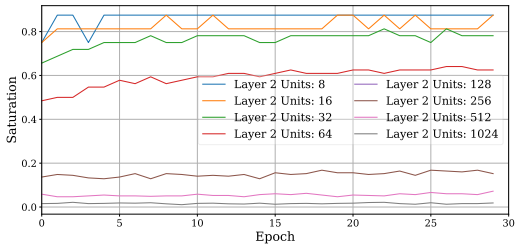


(a) Saturation of Layer 2 during training

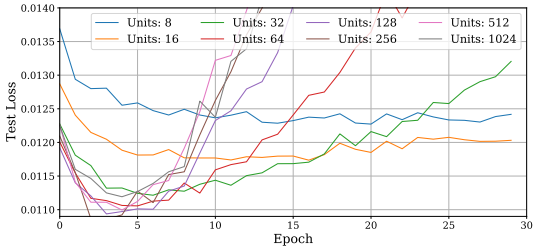


(b) Validation loss during training

Fig. 3. Saturation of a 3-layer MLP does not change substantially during training (a) while the loss is converging (b).



(a) Saturation of layer 2 while overfitting..



(b) Validation loss of an MLP while overfitting on the data.

Fig. 4. Saturation patterns are unaffected by overfitting, which indicates that overfitting is a process not affecting the overall dimensionality of the data inside the feature space.

B. Results

In figure 3, we observe that an increase in the number of units in the fully connected layer will result in an increased saturation. However, the saturation does not change substantially during training, indicating that the inference process is not changed or shifted substantially inside the layer.

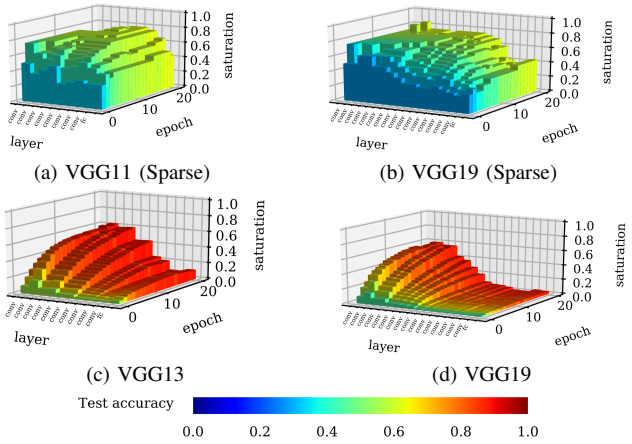
In figure 4, we can see that the increase in validation loss does not affect saturation. The fact that overfitting is not reflected in saturation values indicates that the changes to the way the data is processed when the model starts to overfit are subtle and thus are not reflected in changes to the relevant eigenspace and therefore saturation. This also means that saturation patterns in fully connected networks are independent of the training progress, which could allow for early detection of over and under-parameterization during training. However, it also means that overfitting and convergence of the model need to be taken into consideration, when analyzing saturation on fully connected neural networks.

In figure 5, we can see that saturation behaves substantially different in convolutional neural networks, which exhibit a converging behavior towards a final pattern. This converging behavior is independent of the position of the layer in the network, the number of layers, and the capacity of the network, as figure 5 illustrates.

Another interesting observation is that tail pattern seems to be observable rather early during training, which indicates that an on-line analysis during training allows the data scientist to detect inefficiencies early before training has concluded.

V. Predictability of Tail Patterns Regarding Complexity

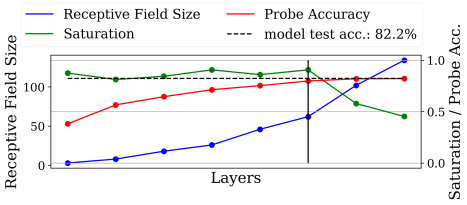
In the following, we examine how overall saturation affects the predictability of tail patterns. Richter et al. [17] show that



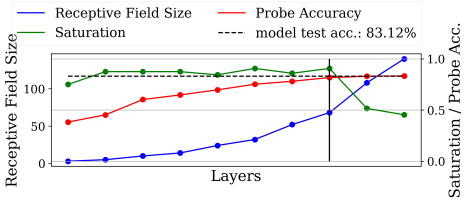
(a) VGG11 (Sparse)
(b) VGG19 (Sparse)
(c) VGG13
(d) VGG19

Fig. 5. Saturations of convolutional neural networks show a converging behavior regarding saturation similar to previous observations in figure 3.

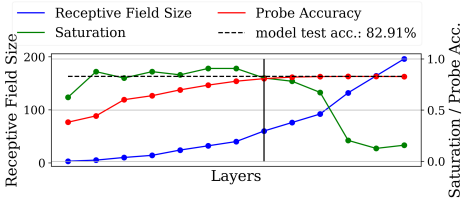
the tail patterns in sequential convolutional neural networks can be predicted by computing the receptive field of all convolutional layers. The receptive field can be considered the field of view of a convolutional layer. Everything contained in the area spanned by the receptive field can hypothetically influence the value on a single position on the output feature map. In section IV, we showed that changing the number of filters in a convolutional layer result the changing of the global saturation level. Because the required capacity in a layer no longer exists, the network can distribute the processing between additional layers of architecture to process the data. However, if the receptive field expansion is determining the number of unproductive layers (like the authors of [17] suggest), we will observe a tail pattern of unproductive layers starting at the border layer.



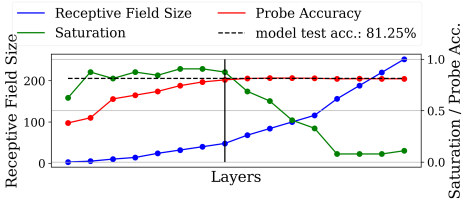
(a) VGG11 with $\frac{1}{8}$ filters per layer



(b) VGG13 with $\frac{1}{8}$ filters per layer



(c) VGG16 with $\frac{1}{8}$ filters per layer



(d) VGG19 with $\frac{1}{8}$ filters per layer

Fig. 6. Performance improvements of the logistic regression probes past the border layer are miniscule, even though the capacity of each layer is reduced to $\frac{1}{8}$ of the original capacity. This indicates that the networks are unable to shift processing to otherwise unused layers even if the capacity is limited. This is consistent with observation made by Richter et al. [17].

A. Methodology

We test the hypothesis by repeating the experiments conducted by [17] regarding the prediction of unproductive layers. The authors of [17] were able to predict unproductive layers by computing the border layer for VGG11, 13, 16, and 19 on Cifar10. We reduce the capacity of these models by reducing the filter size to $\frac{1}{8}$ of the original size to see whether a drastic loss in capacity changes how the inference is distributed. The models are trained for 30 epochs using the SGD-optimizer with a learning rate of 0.1, decaying by a factor of 0.1 every 10 epochs. The batch size is 64, each batch is channel-wise normalized, each image is randomly cropped during inference time as well as randomly horizontally flipped with a probability of 50%. The receptive field and the border layer are computed using the formulas provided by [17].

B. Results

Even though the capacity of the networks has been significantly reduced in every layer, the networks do not spread the inference process among significantly more layers.

Based on these results, we conclude that the inference dynamics of the tested networks did not change substantially by reducing their capacity. This means that the capacity of layers primarily interacts with the difficulty of the problem, while the presence and absence of tail patterns interact with the receptive field, as exemplified by [17].

VI. Conclusion

In this work, we explored the properties of the saturation metric in more detail and integrated this knowledge with insights from [1] and [17]. We find that saturation is influenced in two ways. First, while the saturation of layers relative to other layers in the network is indicative of mismatches between input resolution and architecture, the average saturation level provides insights into the interaction of the network's capacity and the complexity of the problem. This observation opens another way to detect inefficiencies based on over- or under-saturation of the network. Second, we find that these axes of analysis are independent, meaning that low capacity networks do not distribute their inference significantly different among layers than high-capacity networks of the same architecture. The saturation patterns converge similar to the model loss for convolutional neural networks and can therefore be detected early in the training, which results in a faster cycle time for experiments. In summary, saturation can be useful as a tool for both, optimizing neural architectures for specific problems and for performing a more principled investigation of information processing in deep neural networks.

References

[1] M. L. Richter, J. Shenk, W. Byttner, A. Arpteg, and M. Huss, "Feature space saturation during training," 2020.
[2] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," 2018.
[3] M. Tan, B. Chen, R. Pang, V. Vasudevan, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," *CoRR*, vol. abs/1807.11626, 2018. [Online]. Available: http://arxiv.org/abs/1807.11626
[4] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114. [Online]. Available: http://proceedings.mlr.press/v97/tan19a.html
[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385
[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.
[7] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.
[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567
[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Computer Vision and Pattern Recognition (CVPR)*, 2015. [Online]. Available: http://arxiv.org/abs/1409.4842

[10] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," 2017.
[11] J. Shenk, M. L. Richter, A. Arpteg, and M. Huss, "Spectral analysis of latent representations," 2019.
[12] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.
[13] A. Krizhevsky, "Learning multiple layers of features from tiny images," MIT and NYU, Tech. Rep., 2009.
[14] Y. LeCun and C. Cortes, "MNIST handwritten digit database," http://yann.lecun.com/exdb/mnist/, 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[15] Y. Le and X. Yang, "Tiny ImageNet Visual Recognition Challenge," 2015.
[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.
[17] M. L. Richter, W. Byttner, U. Krumnack, L. Schallner, and J. Shenk, "Size matters," 2021.