Viktor Rydbergs Gymnasium Djursholm
Graduation project presented for Swedish Gymnasieexamen
17-01/2025

Impact of Dataset Size on Convolutional Neural Network Classification Performance

A study on how dataset size affects convolutional neural networks' ability to classify handwritten letters

Adrian Hansson, Daniel Padban, Filip Silwer

hanssonadrian7@gmail.com | danielpadban@gmail.com | filip.silwer@gmail.com

Supervisor: **David Höglund**

Abstract

During an era of rapid growth in the artificial intelligence sector, every model needs data to function and improve. This study examines how dataset sizes impact convolutional neural networks' performance in image recognition systems, using a dataset of handwritten letters. The objective of this study is to evaluate how data set sizes affect accuracy, F1-score, and cross-entropy loss, and model a general trend in the relation between dataset size and the outcomes. This report utilizes the EMNIST Letters dataset and convolutional neural networks, consisting of convolutional layers, pooling layers, and deep layers to research the impact of dataset size. The results showcase a logarithmic pattern for accuracy and F1-score, as well as a negative logarithmic pattern for cross-entropy loss, verified using regression.

Keywords: Machine Learning, Convolutional Neural Networks, Model Performance, Neural Network Optimization, Overfitting, Model Generalization

Table of contents

1. Introduction

This study investigates the impact of dataset size on the performance of image recognition models, specifically focusing on CNNs using the EMNIST Letters dataset. The EMNIST dataset, an extended version of the popular MNIST dataset for handwritten digit classification, provides a variety of handwritten characters. By examining how CNNs perform as the size of the training dataset varies, this study aims to gain insights into the optimal conditions for training image recognition models and how to overcome the limitations posed by small datasets. This understanding can guide the development of more effective AI systems, particularly in fields where large, labeled datasets are difficult to obtain. This research hopes to give a deeper understanding of how dataset size impacts the performance of CNNs for image recognition.

1.1. Purpose

The purpose of this paper is to evaluate how a provided amount of data size will impact the performance of CNN models. This study will test this by training a model for image recognition, using a CNN to identify letters from pictures with handwritten letters, and compare the results from different data sizes, in order to generalize results. Therefore the question of issue is: *How does dataset size impact convolutional neural networks' image classification performance?*

2. Background

2.1. Theoretical background

2.1.1. Artificial intelligence and image processing

The arrival of large language models (LLMs) brought attention to machine learning, but also their data usage. It is suggested that the stock of available training data for LLMs could run out by 2026[1]. The problem of data scarcity also exists in other areas of machine learning, such as the medical image processing domain[2].

In the field of image recognition, the amount of data fed into the model plays a crucial role in its performance. In recent years, deep learning models, particularly Convolutional Neural Networks (CNNs), have revolutionized the way machines recognize and classify images. These models require large amounts of labeled data to train effectively, as they learn to identify patterns and features in images through layers of processing. As such, the size and quality of the dataset directly influences the ability of CNNs to generalize well to new, unseen data.[3]

2.1.2. The learning process of neural networks

Neural networks are trained using backpropagation to calculate gradients and gradient descent to optimize the parameters iteratively. The training process is inherently stochastic, meaning that there is a degree of randomness in the training process.[4] The model itself is however deterministic, meaning that there is no randomness involved, the same input will always generate the same output if the model remains the same.[5]

[1] Villalobos P and others, 'Will We Run out of Data? An Analysis of the Limits of Scaling Datasets in Machine Learning', <https://arxiv.org/pdf/2211.04325 >, 4 June 2024.
[2] Amit Gangwal and others, 'Current Strategies to Address Data Scarcity in Artificial Intelligence-Based Drug Discovery: A Comprehensive Review', (*Computers in Biology and Medicine* Vol. 179, September 2024), <https://www.sciencedirect.com/science/article/abs/pii/S0010482524008199 > accessed 7 January 2025
[3]GeeksforGeeks, 'Impact of Dataset Size on Deep Learning Model' (*GeeksforGeeks,* 9 April 2024) <https://www.geeksforgeeks.org/impact-of-dataset-size-on-deep-learning-model/ > accessed 13 January 2025.
[4] Loshchilov I and Hutter F, 'DECOUPLED WEIGHT DECAY REGULARIZATION', (Jan 2019), <https://arxiv.org/pdf/1711.05101 >.
[5] Zhirong Wu, 'Deep Learning Deterministic Neural Networks' <https://3dvision.princeton.edu/courses/COS598/2014sp/slides/lecture05_cnn/lecture05_cnn.pdf> accessed 14 January 2025.

The training data is split up into batches, where each batch represents a subset of the total dataset. The optimization algorithm updates the model's parameters after every batch, using information about the loss from the predictions from the previous batch.

The process of updating the parameters after every batch is repeated until all of the training data has been fed to the neural network, which comprises one epoch. An epoch can be defined as one loop through the whole dataset, so the number of epochs a model is trained is equal to how many times the dataset has been fed to the optimization algorithm.[6]

2.1.3. EMNIST

The EMNIST dataset is an extension of the popular benchmark dataset MNIST and contains handwritten letters and digits. The EMNIST Letters version of EMNIST only contains handwritten letters.[7]

2.1.4. Impact of dataset size on model performance

The relation between dataset size and model performance has been well-known across various domains and specialized fields, such as geological or medical research, with larger datasets typically leading to improved performance.[8] Larger datasets provide more examples for the model to learn from, allowing it to capture a wider range of features and variances in the data. However, when the dataset is too small, the model is at risk of overfitting, which harms its ability to generalize to new data. In these cases, the model may perform well on the training data but poorly on test data or in real-world applications.[9]

[6] Jason Brownlee, 'Difference between a Batch and an Epoch in a Neural Network - MachineLearningMastery.com' (*MachineLearningMastery.com*9 August 2022) <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/> accessed 16 January 2025.

[7] Gregory Cohen and others, 'EMNIST: An Extension of MNIST to Handwritten Letters' <https://arxiv.org/pdf/1702.05373> accessed 16 January 2025.

[8] Daniel W and Fine E, 'An Evaluation of Training Size Impact on Validation Accuracy for Optimized Convolutional Neural Networks' (2018) 1 SMU Data Science Review 12 <https://scholar.smu.edu/cgi/viewcontent.cgi?article=1046&context=datasciencereview >

[9] 'Scaling Machine Learning Models with Large Datasets and Data Labeling' (*Sapien.io,* 2024) <https://www.sapien.io/blog/scaling-machine-learning-models-with-large-datasets-and-data-labeling > accessed 13 January 2025.

The problem of dataset size occurs in many practical applications, particularly in specialized fields such as medical imaging or object detection, where models are constrained by the limited availability of large datasets. In such cases, it becomes crucial to understand how the amount of data impacts the model's results and how to optimize the performance of AI models despite these limitations.

2.1.5. Methods to combat data scarcity

Several strategies have emerged to address the challenges of data scarcity, including data augmentation, which artificially increases the size of the dataset by generating new samples by modifying existing ones.[10] One promising approach in image recognition is the use of transfer learning, particularly when dealing with limited data. Transfer learning allows a model pre-trained on a large dataset to apply the knowledge it has acquired to a smaller dataset. This method has been shown to improve performance significantly by leveraging the general features learned from a large, diverse dataset and applying them to a more specific task.[11]

The use of synthetic data is another way to combat data scarcity. Synthetic data is information that is artificially manufactured, sometimes using machine learning models, rather than generated from real-world events or existing datasets. The reason is often that gathering high quality data takes a lot of time and there is only so much information and datasets available.[12] However, the use of synthetic data has its own share of problems. The largest problem with synthetic data is that AI models fed by it will eventually become a closed, biased system. They will be trained on biased, repetitive datasets that are also made up of synthetic data which will lead to the data becoming further detached from reality. This is called AI hallucination, which could eventually harm users since the output data from the AI no longer aligns with reality.[13]

[10] Wang J and Perez L, ‘The Effectiveness of Data Augmentation in Image Classification Using Deep Learning’ <https://arxiv.org/pdf/1712.04621>, accessed 13 January 2025.

[11]Zehui Zhao and others, ‘A Comparison Review of Transfer Learning and Self-Supervised Learning: Definitions, Applications, Advantages and Limitations’ (2024) 242 Expert Systems with Applications 122807, <https://www.sciencedirect.com/science/article/pii/S0957417423033092>, accessed 13 January 2025.

[12] Cameron Hashemi-Pour, Kinza Yasar and Nicole Laskowski, ‘What Is Synthetic Data? Examples, Use Cases and Benefits’ (*Search CIO*2024) <https://www.techtarget.com/searchcio/definition/synthetic-data> accessed 29 January 2025.

[13] ‘Problems with Synthetic Data - AITUDE’ (*AITUDE*March 2023) <https://www.aitude.com/problems-with-synthetic-data/> accessed 29 January 2025.

2.2. Terms

An overview of important terms to understand the research.

Machine learning

Machine learning is the concept of automatically fitting a model to a dataset. A basic example would be linear regression, where k and m in $y = kx + m$ are adjusted to minimize the distance to each data point. More advanced versions, such as neural networks are based on the same principles, where weights (w) and biases (b) are adjusted for $wx + b$. Neural networks are however composed of multiple layers of neurons with activation functions that introduce non-linearity in between, where each neuron has its own weight and bias. Neural networks comprise a sub-field of machine learning called deep learning.[14]

Machine learning is divided into two ways to classify data. Supervised learning, which gives data predestined classifications. This allows machine predictions to be compared to the actual value of the class. On the other hand, unsupervised learning, meaning the machine itself catches common features when training on data. The data with common features then determines applicable classes.[15]

Neural Networks

Neural networks are machine learning models that consist of interconnected nodes or neurons that process data to enable tasks such as pattern recognition and decision-making. Neural networks are made up of a number of key components, such as neurons, connection, weights and biases, functions and learning rules. These components work together to enable the networks to perform complicated tasks and functions, much like a human brain.[16] They rely on being trained on pre-existing data to improve their accuracy and understanding over time, which leads to them becoming more and more “intelligent” over time.[17]

[14] Sloan M, ‘Machine Learning, Explained | MIT Sloan’ (*MIT Sloan*21, April 2021), <https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained >, accessed 13 January 2025

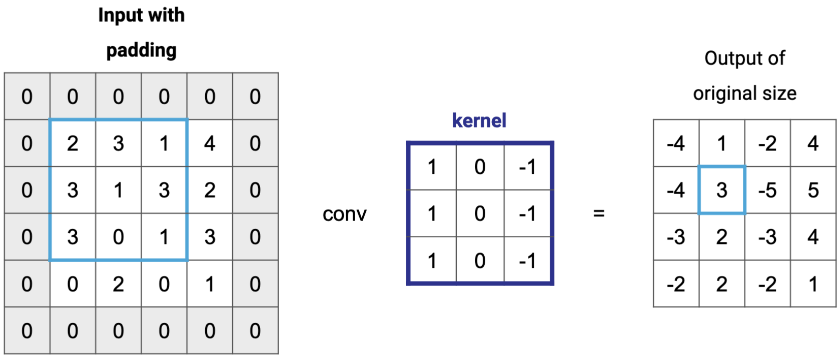
[15] Julianna Delua, ‘Supervised vs Unsupervised Learning’ (Ibm.com24 April 2024) <https://www.ibm.com/think/topics/supervised-vs-unsupervised-learning> accessed 29 January 2025.

[16] GeeksforGeeks, ‘What Is a Neural Network?’ (*GeeksforGeeks*17 January 2019) <https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/> accessed 29 January 2025.

[17] IBM, ‘Neural Network’ (*Ibm.com*6 October 2021) <https://www.ibm.com/think/topics/neural-networks> accessed 29 January 2025.

Convolutional Neural Networks (CNN)

Convolutional neural networks are a type of deep learning algorithm often used in computer vision tasks such as image recognition.[18] They work by using convolutional layers that apply small, learnable filters (also called kernels) to an input image. These filters slide across the image, detecting patterns such as edges, textures, and shapes. This allows the network to summarize features of the image into fewer dimensions. Valuable features of images are selected by optimizing the weights of the kernel during training. As the kernel slides over the input matrix, it performs element-wise multiplication. The products of each multiplication are summed, which in turn leads to reduced dimensionality. The process of zero-padding means that a layer of 0s is added around the input to avoid reducing the dimensions of the input. This allows the model to extract relevant features without losing information in the process.[19]



0	0	0	0	0	0
0	2	3	1	4	0
0	3	1	3	2	0
0	3	0	1	3	0
0	0	2	0	1	0
0	0	0	0	0	0

1	0	-1
1	0	-1
1	0	-1

-4	1	-2	4
-4	3	-5	5
-3	2	-3	4
-2	2	-2	1

Figure 1: Visualization of how a kernel summarizes data (with zero-padding) into latent spaces using weights.[20]

[18] Irhum Shafkat, 'Intuitively Understanding Convolutions for Deep Learning' (*Medium*June, 2018) <https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1 > accessed 7 January 2025

[19] O'Shea K and Nash R, 'An Introduction to Convolutional Neural Networks' (2015) <https://arxiv.org/pdf/1511.08458 >

[20] 'What Is a Convolution? How to Teach Machines to See Images | 8th Light' (*8th Light*2022) <https://8thlight.com/insights/what-is-a-convolution-how-to-teach-machines-to-see-images> accessed 7 February 2025.

Pooling

Similar to convolution, pooling is used to compile data by sliding a filter over a matrix and aggregating the data using a predetermined method but it is different from convolution since this method does not contain any learnable parameters, it is only a process to reduce the dimensionality of the matrix. The point of pooling is to scale down the amount of information leaving only the most influential features.[21]

Activation function

Activation functions are used in neural networks to introduce non-linearity which allows the model to learn non-linear relationships. This can be compared to linear regression, which can only learn linear relationships.[22]

[21] GeeksforGeeks, 'CNN | Introduction to Pooling Layer' (2019) GeeksforGeeks <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/ > accessed 13 January 2025

[22] 'Activation Functions in Neural Networks [12 Types & Use Cases]' (*V7labs.com* 2021) <https://www.v7labs.com/blog/neural-networks-activation-functions > accessed 13 January 2025

Dropout

Dropout layers are used to create more robust models. Dropout works by zeroing random data points during training.[23] By zeroing elements in the input to the network, certain neurons are not activated, which helps the model avoid becoming dependent on specific neurons, in turn, helping the model generalize better.[24]

Feature scaling

Feature scaling is a preprocessing step often used for machine learning to bring all features to the same numerical range, making them compatible. One method for feature scaling is Z-score normalization. Z-score normalization rescales the range to have a mean of 0 and a standard deviation of 1.[25]

Overfitting

Overfitting occurs when a model becomes too optimized for the training data and fails to generalize to unseen data. A common cause of overfitting in learning models is the lack of information to train on, since fewer samples make the unique features of every input more influential, resulting in a worse ability to see general patterns.[26] Overfitting can also occur due to over-complex models that fail to learn the general patterns of the data.[27]

[23] 'Dropout — PyTorch 2.5 Documentation' (*Pytorch.org*2023) <https://pytorch.org/docs/stable/generated/torch.nn.Dropout.html> accessed 16 January 2025.

[24] Arunn Thevapalan, 'Dropout Regularization Using PyTorch: A Hands-on Guide' (*Datacamp.com*20 June 2024) <https://www.datacamp.com/tutorial/dropout-regularization-using-pytorch-guide> accessed 16 January 2025.

[25] 'Importance of Feature Scaling' (*scikit-learn*2025) <https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html> accessed 13 January 2025.

[26] Wohlwend B, 'Understanding Overfitting and Underfitting in Machine Learning' (*Medium,* 19 July 2023) <https://medium.com/@brandon93.w/understanding-overfitting-and-underfitting-in-machine-learning-b699e0ed5b28 > accessed 13 January 2025

[27] Shai Samana, 'Mastering Model Complexity: Avoiding Underfitting and Overfitting Pitfalls' (*Pecan AI*13 June 2024) <https://www.pecan.ai/blog/machine-learning-model-underfitting-and-overfitting/ > accessed 13 January 2025

Stochastic gradient descent

Gradient descent optimizes the model's parameters by using backpropagation through the neural network's layers to calculate the gradient (derivative) concerning the loss function, allowing the model to learn. This is a stochastic process, meaning that there is a degree of randomness since the optimization algorithm essentially practices a trial-and-error method.[28]

$$\theta = \theta - \eta \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)}) \qquad (1)^{29}$$

Equation 1 describes stochastic gradient descent, where θ denotes the parameter being updated, η denotes the learning rate and $\nabla_{\theta} J(\theta; x^{(i)}; y^{(i)})$ is the gradient of the sample or mini-batch with respect to the parameter and loss. This can be compared to standard gradient descent that calculated the gradient over the whole epoch instead.[30]

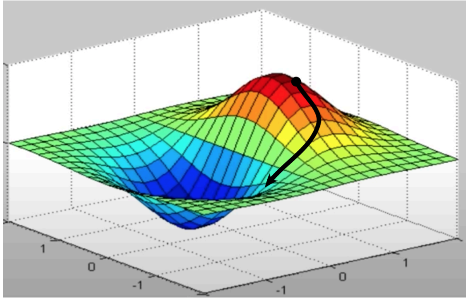


Figure 2: The model minimizes the loss every step (black arrow), finding the minima (blue area).[31]

[28] 'Gradient Descent', (*Questions and Answers in MRI,* 2022) <https://mriquestions.com/back-propagation.html#/> accessed 7 January 2025

[29] ibid.

[30] Sebastian Ruder, 'An Overview of Gradient Descent Optimization Algorithms *' <https://arxiv.org/pdf/1609.04747>.

[31] 'Gradient Descent', (*Questions and Answers in MRI,* 2022) <https://mriquestions.com/back-propagation.html#/> accessed 7 January 2025

Local minima

The goal of the parameter optimization is to find the global minimum of the loss function, however, local minima in the parameter space could cause sub-optimal weights and cause the gradient descent to get stuck.[32] By observing the optimization algorithm described in Equation 1, it is clear that if the gradient becomes too small, the parameter updates will also become small. This can lead to difficulties with getting out of the minima, since the gradient approaches 0 near the bottom of the minima and the parameter updates will therefore be too insignificant to change the gradient within a reasonable timeframe.

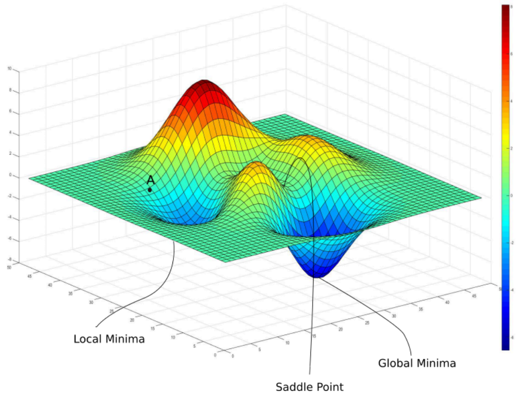


Figure 3: Graph of local and global minima.[33]

[32] Mishra M, 'The Curse of Local Minima: How to Escape and Find the Global Minimum' (*Medium*June, 2023) <https://mohitmishra786687.medium.com/the-curse-of-local-minima-how-to-escape-and-find-the-global-minimum-fdabceb2cd6a > accessed 13 January 2025

[33] 'Intro to Optimization in Deep Learning: Gradient Descent | DigitalOcean' (*Digitalocean.com*2025) <https://www.digitalocean.com/community/tutorials/intro-to-optimization-in-deep-learning-gradient-descent> accessed 13 January 2025.

Cross-Entropy

Cross-entropy is a loss function used in machine learning to measure the error of a classification model. It measures the difference between the discovered probability distribution of a classification model and the predicted values. The goal when optimizing the model is to minimize the cross-entropy loss.[34] The function can be described as:

$$L(y, \hat{y}_i) = \quad - \sum_{i=1}^{C} y_i log(\hat{y}_i) \tag{2}$$

C in Equation 2 represents the number of classes, log is the natural logarithm, y_i is a binary value (true or false) for the class being evaluated and $\hat{y}_i$ is the predicted probability (confidence level) for the class. The function can thereby evaluate the confidence level for the correct class, since non-correct classes are multiplied by 0.[35]

F1-Score

F1-score is a metric used for classification tasks that take into consideration false positives and negatives. It is the harmonic mean of the metrics recall and precision. Precision measures the accuracy of positive predictions by dividing the number of true positives by the number of true and false positives. Recall measures how well the model can predict actual positive values. It is calculated by dividing the number of true positives by the number of actual positives (true positives and false negatives). The range of F1-score is 0 to 1, where 0 is worst and 1 is best.[36]

[34] Kurtis Pykes, 'Cross-Entropy Loss Function in Machine Learning: Enhancing Model Accuracy' (*Datacamp.com,* 11 January 2024) <https://www.datacamp.com/tutorial/the-cross-entropy-loss-function-in-machine-learning> accessed 15 January 2025.

[35] GeeksforGeeks, 'Categorical CrossEntropy in MultiClass Classification' (*GeeksforGeeks,* 17 September 2024) <https://www.geeksforgeeks.org/categorical-cross-entropy-in-multi-class-classification/> accessed 29 January 2025.

[36] GeeksforGeeks, 'F1 Score in Machine Learning' (*GeeksforGeeks*27 December 2023) <https://www.geeksforgeeks.org/f1-score-in-machine-learning/> accessed 15 January 2025.

Random seed

A random seed initializes the random number generator which is used to initialize model parameters and control the training process. A manual random seed can be used for reproducibility.[37]

2.3. Previous work

2.3.1. SMU Data Science Review

A similar study, *An Evaluation of Training Size Impact on Validation Accuracy for Optimized Convolutional Neural Networks*, highlights that dataset size plays a pivotal role in influencing the performance of convolutional neural networks (CNNs) in image classification tasks. A one-way analysis of variance (ANOVA) revealed statistically significant differences (p-value < 0.0001) in validation accuracy across varying training set sizes. The results demonstrate that using larger portions of the dataset consistently improves performance, with diminishing returns observed at higher dataset percentages. For instance, a validation accuracy of 90% was achieved with just 40% of the original training data, while using 25% yielded significantly lower performance without overlapping quantiles in the box plot distributions. This underscores the importance of dataset size in achieving optimal results.[38]

In line with previous research, this study validates that transfer learning models benefit significantly from larger datasets. Pre-training on extensive datasets and fine-tuning for specific tasks enhances model generalization. However, when dataset size is constrained, techniques such as data augmentation become critical. By incorporating transformations like rotations, scaling, and reflections, the augmented training sets helped mitigate overfitting and slightly improved the generalization ability of the models. Despite these gains, results suggest

[37] 'Reproducibility — PyTorch 2.5 Documentation' (*Pytorch.org,* 2023) <https://pytorch.org/docs/stable/notes/randomness.html> accessed 14 January 2025.

[38] Daniel W and Fine E, 'An Evaluation of Training Size Impact on Validation Accuracy for Optimized Convolutional Neural Networks' (2018) 1 SMU Data Science Review 12 <https://scholar.smu.edu/cgi/viewcontent.cgi?article=1046&context=datasciencereview >

that datasets of at least 100,000 images are ideal for optimal CNN performance, consistent with findings in geological and medical domains.[39]

The statistical analysis further confirmed the sensitivity of CNNs to training set sizes. Models trained on reduced data (e.g., 10% or 25% of the original dataset) exhibited a noticeable drop in validation accuracy, reinforcing the hypothesis that insufficient data limits the model's ability to generalize. Nevertheless, neural networks demonstrated resilience compared to other algorithms, performing competitively even on smaller, well-representative datasets.[40]

2.3.2. Research on geological datasets

A study regarding geological datasets, *Impact of dataset size and convolutional neural network architecture on transfer learning for carbonate rock classification* by Dawson HL, Dubrule O and John CM, demonstrates that the size of a dataset has a significant impact on the accuracy of AI models in carbonate classification. Larger datasets, such as the 104k dataset used in the geological research, led to superior prediction performance with a top accuracy of 92% and a bottom of 88% depending on the model. In contrast, smaller datasets like the 7k dataset showed a top accuracy of 85% and a bottom of 66%, varying between models, despite achieving high training accuracies, which highlights the issues with overfitting and reduced generalization to unseen data. This calls attention to the importance of having sufficiently large datasets for training AI models to ensure reliability and decrease in result variance.[41]

The classification model in their study was based on convolutional neural networks (CNNs), much like the model used in this paper, employing transfer learning with architectures like Inception-v3, VGG19, and ResNet. The CNNs were pre-trained on large-scale datasets and specialized for carbonate classification.[42]

[39] Daniel W and Fine E, 'An Evaluation of Training Size Impact on Validation Accuracy for Optimized Convolutional Neural Networks' (2018) 1 SMU Data Science Review 12 <https://scholar.smu.edu/cgi/viewcontent.cgi?article=1046&context=datasciencereview >

[40]ibid.

[41] Dawson HL, Dubrule O and John CM, 'Impact of Dataset Size and Convolutional Neural Network Architecture on Transfer Learning for Carbonate Rock Classification' (2022) 171 Computers & Geosciences 105284 <https://www.sciencedirect.com/science/article/pii/S0098300422002333 > accessed 7 January 2025

[42] Dawson HL, Dubrule O and John CM, 'Impact of Dataset Size and Convolutional Neural Network Architecture on Transfer Learning for Carbonate Rock Classification' (2022) 171 Computers & Geosciences 105284 <https://www.sciencedirect.com/science/article/pii/S0098300422002333 > accessed 7 January 2025

For smaller datasets, the study highlights the utility of data augmentation techniques to artificially expand the training set and reduce overfitting. By introducing transformations such as rotations, reflections, and scaling, augmented datasets can help the model generalize better while maintaining the geological relevance of the images. However, the results indicate that transfer learning is still constrained by dataset size, emphasizing that datasets of at least 100,000 images are ideal for achieving optimal performance in geological image classification tasks.[43]

2.3.3. Research within the medical domain

An investigation, *Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain,* regarding dataset sizes in the medical domain reveals important findings about how neural networks (NNs) respond to changes in dataset size within medical applications. Neural networks showed moderate sensitivity to dataset size reduction, with performance dropping significantly in 42% of scenarios. This sensitivity occurs because NNs learn by adjusting a large number of weights during training, a process that relies on having sufficient data and so when datasets are too small, the network struggles to generalize effectively, leading to a decline in performance.[44]

Despite this sensitivity, neural networks performed better than some other models on smaller datasets in terms of average accuracy. This indicates that, while NNs benefit greatly from larger datasets, they can still deliver competitive results with smaller datasets if these datasets are well-representative of the original data distribution. The study also highlights that techniques like data augmentation or pre-training on larger datasets could help NNs overcome challenges with limited data.[45]

[43] ibid.

[44] Althnian A and others, 'Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain' (2021) 11 Applied Sciences 796 <https://www.mdpi.com/2076-3417/11/2/796 > accessed 7 January 2025

[45] Althnian A and others, 'Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain' (2021) 11 Applied Sciences 796 <https://www.mdpi.com/2076-3417/11/2/796 > accessed 7 January 2025

These results emphasize the importance of providing neural networks with enough high-quality data to maximize their performance. While NNs are powerful and flexible models, their dependence on dataset size makes it critical to ensure that datasets used for training are large enough or supplemented with strategies to mitigate overfitting and improve generalization.[46]

3. Methodology

The experiment consisted of training 110 models, each with the same architecture and training parameters such as learning rate, weight decay, and gradient clipping. The 110 models were grouped into 11 groups of 10 models, for each dataset size. Each dataset group contained the same set of random seeds for the PyTorch random generator. Train and test data from the training of each model was then collected and compiled into graphs. The experiment was performed using Python. The model was trained using a graphical processing unit (GPU) via kaggle.com, the GPU was an NVIDIA Tesla p100. A GPU was used to speed up the training process through multiprocessing.

3.1. Load and prepare the EMNIST Letters Dataset

The EMNIST Letters dataset was accessed through PyTorch.

The data was then flipped and inverted so that each letter would be oriented correctly (readably). The pixel values of the images were then normalized according to Z-score normalization. The mean 0.1736 and the standard deviation 0.3248 were used.[47]

3.2. Selecting datasets of different sizes

The dataset sizes were selected based on divisibility with the maximum size. The maximum size chosen in our experiment was 12000 images, with the intent of providing diversification in the training data to yield a robust model.

[46] ibid.

[47] Xavier Spycy, ‘EMNIST-Classifier/Notebook.ipynb at Main · XavierSpycy/EMNIST-Classifier’, (*GitHub,* 2023), <https://github.com/XavierSpycy/EMNIST-Classifier/blob/main/notebook.ipynb >, accessed 7 January 2025.

$$epochs \ = \ \frac{updates \star batch\, size}{datasize} \tag{3}$$

$$base\ epochs_d \ = \ \frac{datasize_{max}}{datasize_d} \tag{4}$$

Where $datasize_d$ is the data size that is being tested and $datasize_{max}$ is the maximum size tested, which is 12000 images of the EMNIST Letters dataset in this experiment.

The value of the *batch size* variable was first tested at a value of 10. However, this led to an excess amount of *updates* which resulted in an unnecessarily long training time, a new value of 72 was therefore chosen to reduce the number of *updates* and training time. The dataset size varied, which affected how many training sessions, or epochs, were needed to reach the desired number of updates. The number of epochs was given by *Equation 1*, which leads to a difference in the number of epochs selected depending on the dataset size. Each model was trained on 144000 samples, however the amount of unique samples varied for each one.

Dataset ID	Size	Epochs	Base epochs
B1	12000	12	1
B2	6000	24	2
B3	4000	36	3
B4	3000	48	4
B5	2400	60	5
B6	2000	72	6
B7	1500	96	8
B8	1200	120	10
B9	1000	144	12
B10	800	180	15
B11	750	192	16

Table 1: Datasets used to evaluate model performance

3.3. Create a CNN model

A CNN model with 3 convolution blocks and 2 deep layers as the output section was used, with each convolution block consisting of:

1. Convolution layer
2. Sigmoid linear unit activation
3. Max pooling layer

Figure 4: A convolution-pooling block.

1. Convolution-pooling block 1
    - Channels: 32
    - Convolution padding: same
    - Convolution kernel size: 5
    - Max pooling kernel size: 2
2. Convolution-pooling block 2
    - Channels: 64
    - Convolution padding: same
    - Convolution kernel size: 5
    - Max pooling kernel size: 2
3. Convolution-pooling block 3
    - Channels: 128
    - Convolution padding: same
    - Convolution kernel size: 5
    - Max pooling kernel size: 2
4. Fully connected layer
    - Hidden size: 256
5. 1D dropout
6. Fully connected layer
    - Output size: 26 (Each output corresponds to one letter class)

Figure 5: The architecture of the model used in this experiment.

3.4. Training the model and collecting data

To ensure that the parameters were controlled, 10 runs were run for each dataset, where each run had a different random seed. The seeds followed the formula below. Since gradient descent and parameter initialization are inherently stochastic, results can vary depending on the specific values of each run. To mitigate this, the average values of all 10 runs for each dataset were used when compiling the results.

$$seed_i = 100 \star i + i \tag{5}$$

In *Equation 3, $seed_i$* was the random seed for run i in the dataset size, for PyTorch's random generator.

To train the model, Cross Entropy Loss[48] combined with the AdamW[49] optimizer with the learning rate 0.0004 and weight decay of 0.4 was used. To stabilize the training, gradient clipping was utilized, where all gradients above 1 would be clipped.

Each dataset was tested for 10 runs. Each run consisted of a number of training epochs of the training data. The number of epochs was calculated using *Equation 1*.

After each training epoch one test epoch was run, under the condition in *Figure 6*:

```
if ( epoch mod ( $base\ epochs_d$ ) = 0 ):
    run test epoch
else:
    pass
```

Figure 6: Pseudocode for the condition for test epochs.

[48] Mao A, Mohri M and Zhong Y, 'Cross-Entropy Loss Functions: Theoretical Analysis and Applications' <https://arxiv.org/pdf/2304.07288 >

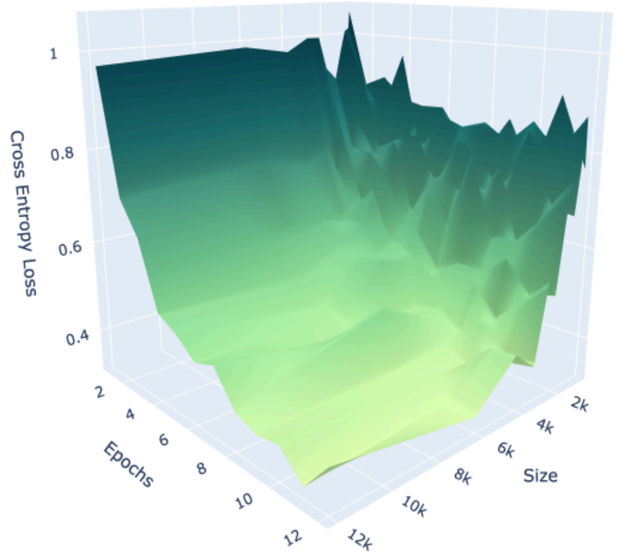
[49] Loshchilov I and Hutter F, 'DECOUPLED WEIGHT DECAY REGULARIZATION' <https://arxiv.org/pdf/1711.05101 >

3.5 Evaluation and analysis

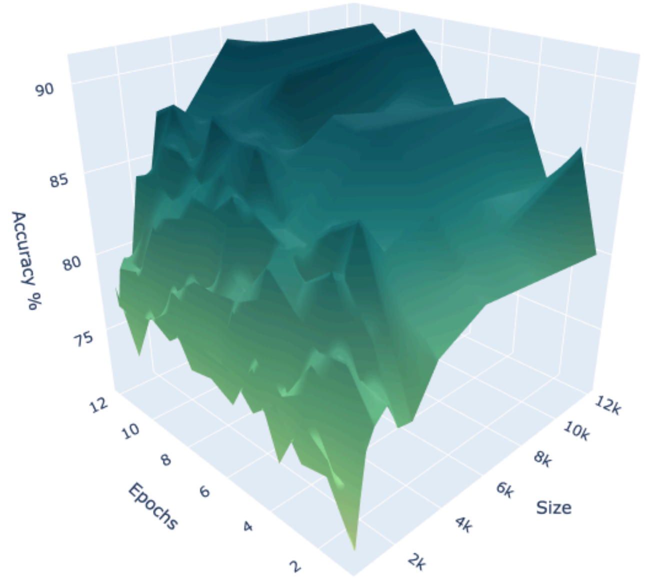
3 metrics were collected after each train and test epoch, the metrics collected were F1-score, accuracy and cross-entropy loss. The results were then compiled in 3 ways. The 3d graphs were based on the mean of all seeds for each dataset, from the train and test epoch results. This resulted in 2 (train and test) * 3 (metrics) * 110 run histories (10 (seeds) * 11 (dataset sizes)) being aggregated into 66 histories and then 6 graphs. The boxplots were designed to illustrate the variability in results between seeds. The last data point for each seed was used and then plotted as 6 box plots for the train and test versions of F1-score, accuracy, and cross-entropy. Lastly, regressions were made to analyze the relationship between dataset size and performance. The results were aggregated in the same way as for the 3d plots, except only the last epoch of each history was used. A log-linear regression was then made for the performance metrics with respect to dataset size.

4. Results

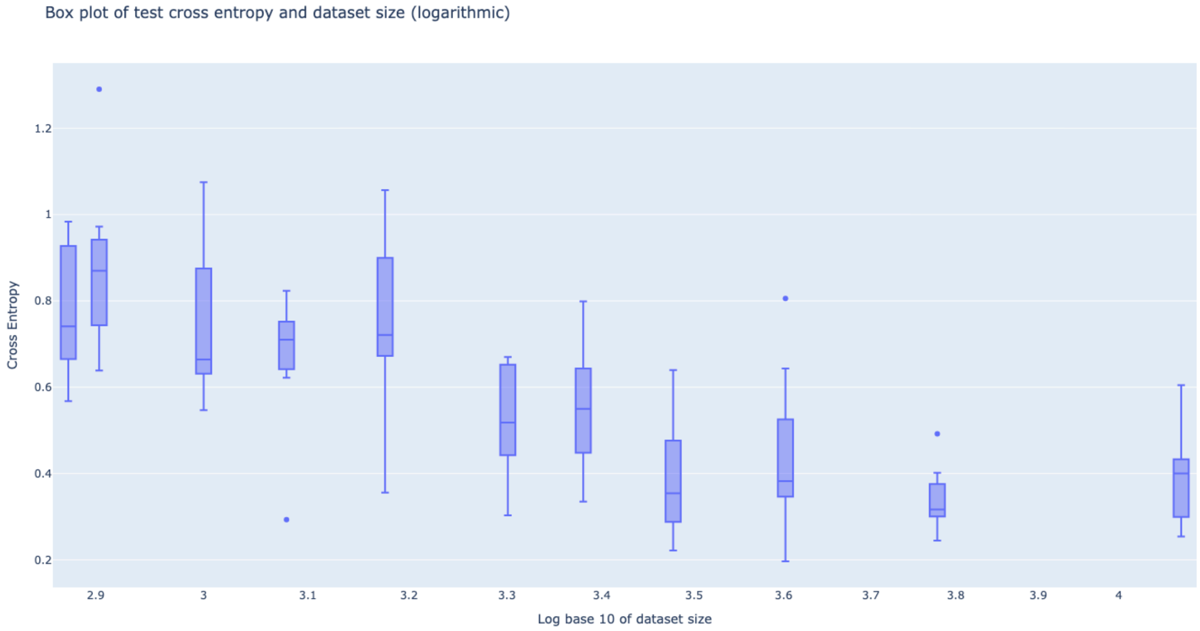
The following graphs are also available as interactive appendices along with 8 other interactive graphs.



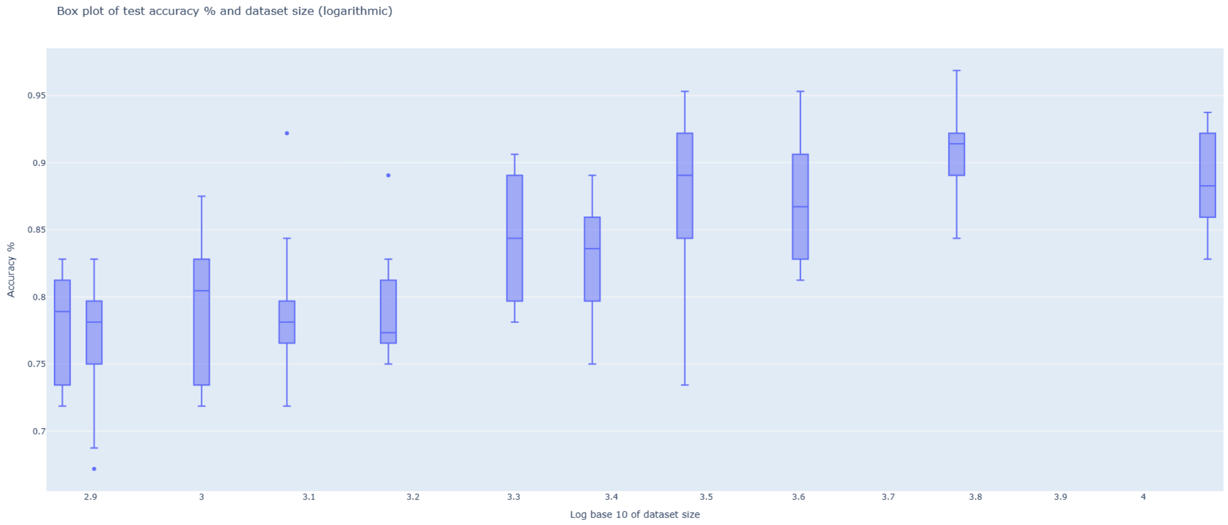
Graph 1: 3d surface plot showing test results, with epochs and dataset size on the x and y-axis, and cross-entropy loss on the z-axis. The graph displays a negative logarithmic pattern along the *Epochs* and *Size* axes. Available as an interactive plot in Appendix 8.



Graph 2: 3d surface plot showing test results, with epochs and dataset size on the x and y-axis, and accuracy on the z-axis. The graph displays a logarithmic pattern along the *Epoch* and *Size* axes. Available as interactive plot in Appendix 9.



Graph 3: Box plot showing test cross entropy (y-axis) and log base 10 of dataset size (x-axis). Available as interactive plot in Appendix 1.



Graph 4: Box plot showing test accuracy (y-axis) and log base 10 of dataset size (x-axis). Available as interactive plot in Appendix 3.

The results in *Graph 1* and *Graph 2* showcase that both dataset size and epochs have an impact on performance, following a logarithmic pattern for test accuracy, and a negative logarithmic pattern on test cross-entropy loss. The test F1-score (Appendix 7) followed a logarithmic pattern as well.

5. Analysis

The results showcase that the accuracy and F1-score follow the same pattern regarding both training steps and dataset size: The cross-entropy follows a negative logarithmic pattern. The graphs show that the average performance across random seeds is positively correlated to dataset size, with diminishing returns as the dataset size increases.

Graph 3 and *Graph 4* showcase the variability in the results. Despite the variability, a positive correlation can be observed. Since the x-axis is the base 10 logarithm of the dataset sizes, it is clear F1 (Appendix 2) and accuracy follow a logarithmic pattern concerning the dataset size, while the cross-entropy follows a negative logarithmic pattern. This can further be proven by making a log-linear regression of the performance metrics and the dataset size. The log-linear regression appears to resonate with the test results, since the regressions for F1 and accuracy both achieve an r-square value of 0.848, while the test cross-entropy regression has an r-square value of 0.801 (*regressions12000.txt* in Appendix 13), though it should be observed that when removing the last data point with 12k images, since it seems to be an outlier, the r squared value for accuracy and F1-score reaches 0.936, and the cross-entropy reaches 0.896. (*regressions6000.txt* in Appendix 13)

6. Discussion

6.1. Explanation of results

The positive correlation between accuracy and F1 with dataset size occurs because the model encounters more unique samples during training as the dataset size increases. This allows the gradient descent to optimize the parameters better and pick out the general features of each class, instead of the unique features from individual samples. By increasing the dataset size, the risk of overfitting is reduced, since the model encounters a more diverse set of samples during training. The benefits of increasing the dataset size do however seem to diminish as the dataset size increases. This suggests that at a certain point, the model encounters a set of samples that is sufficiently diversified to learn the patterns between classes. The diminishing returns can also be explained by the fact that the model cannot achieve more than 100% accuracy, the benefits of increasing the dataset size therefore need to diminish as the model approaches maximum accuracy, since dataset size is not capped, like accuracy.

6.2. Comparison to previous work

The SMU Data Science Review paper in 2.2.1. showcases similar results to our study, where an increased dataset size has a positive effect on performance, however, with diminishing returns. This is in line with our analysis, where the log-linear regression achieved r-square values of 0.841 and 0.801, that the dataset size has a logarithmic impact on performance and therefore has diminishing returns when increasing dataset size.

The results from the study found in 2.2.2 would also suggest a positive correlation for accuracy with respect to dataset sizes. When comparing their results to dataset sizes, their best-performing model for the 104k dataset achieved an accuracy of 92%, while it would provide a score of 62% for the 7k dataset. This would also align with our outcome.

6.3. Validity of results

In the case of this study, the use of a single model architecture could be a factor that limits the credibility of the results. Another experiment on another architecture that suggests a different trend in accuracy depending on data sizes and steps, would prove equally as much as this project. Nevertheless, that does not discredit the results for this specific model architecture, since all results per dataset size are calculated from means of ten tests based on ten random seeds.

Multiple random seeds were selected for each size to mitigate that the gradient descent is inherently a stochastic process, thereby improving the statistical significance of our research. However, an increased amount of random seeds would improve the credibility of the research, since the results showcased large variability across seeds. There is a possibility that the shortage of seeds had a significant effect on the outcome in this study, considering the last data point seems to be an outlier in the final dataset for accuracy, F1-score as well as cross-entropy loss. Increasing the seeds could therefore further validate the r-squared value for the regressions, since it would clarify whether or not the last data point is an outlier.

It should also be taken into consideration that this study only evaluates the performance of one dataset, EMNIST Letters. This leads to our results only being relevant when discussing the EMNIST dataset, and not other datasets that are available for evaluation. However, since the EMNIST dataset and its predecessor MNIST are commonly used as benchmark datasets, one could assume that the patterns discovered in this study would apply to other datasets as well. Despite this, it would be beneficial to expand this study to other architectures and datasets to further understand the impact of dataset size on model performance.

6.4. Conclusion

This study demonstrates a positive correlation between dataset size and model performance as measured by the F1-score and accuracy, with diminishing returns as dataset size decreases. The cross-entropy loss follows a reverse logarithmic curve, as mentioned in the analysis. Larger datasets allow the model to adapt better by being exposed to a higher degree of diversification in the training samples, allowing the model to generalize more effectively and therefore achieve higher performance. The flattening of the accuracy curve suggests that after a certain point, the training data becomes sufficiently diversified to allow the model to generalize to unseen samples.

7. Acknowledgements

We would like to thank David Höglund and Efstratios Zaloumis for their guidance throughout our research.

8. References

‘Activation Functions in Neural Networks [12 Types & Use Cases]’ (*V7labs.com,* 2021) <https://www.v7labs.com/blog/neural-networks-activation-functions > accessed 13 January 2025

Althnian A and others, ‘Impact of Dataset Size on Classification Performance: An Empirical Evaluation in the Medical Domain’ (2021) 11 Applied Sciences 796 <https://www.mdpi.com/2076-3417/11/2/796 > accessed 7 January 2025

Amit Gangwal and others, ‘Current Strategies to Address Data Scarcity in Artificial Intelligence-Based Drug Discovery: A Comprehensive Review’ (2024) 179 Computers in Biology and Medicine 108734 <https://www.sciencedirect.com/science/article/abs/pii/S0010482524008199 > accessed 7 January 2025

Bailly A and others, ‘Effects of Dataset Size and Interactions on the Prediction Performance of Logistic Regression and Deep Learning Models’ (2021) 213 Computer Methods and Programs in Biomedicine 106504 <https://www.researchgate.net/publication/355746542_Effects_of_Dataset_Size_and_Interactions_on_the_Prediction_Performance_of_Logistic_Regression_and_Deep_Learning_Models > accessed 13 January 2025

Cho Y, Cho S-H and Kim J, ‘Design of Handwriting-Based Text Interface for Support of Mobile Platform Education Contents’ (2021) 27 Journal of the Korea Computer Graphics Society 81 <https://www.researchgate.net/publication/356902070_Design_of_Handwriting-based_Text_Interface_for_Support_of_Mobile_Platform_Education_Contents > accessed 7 January 2025

Daniel W and Fine E, ‘An Evaluation of Training Size Impact on Validation Accuracy for Optimized Convolutional Neural Networks’ (2018) 1 SMU Data Science Review 12 <https://scholar.smu.edu/cgi/viewcontent.cgi?article=1046&context=datasciencereview >

Dawson HL, Dubrule O and John CM, ‘Impact of Dataset Size and Convolutional Neural Network Architecture on Transfer Learning for Carbonate Rock Classification’ (2022) 171 Computers & Geosciences 105284 <https://www.sciencedirect.com/science/article/pii/S0098300422002333 > accessed 7 January 2025

‘Dropout — PyTorch 2.5 Documentation’ (*Pytorch.org*2023) <https://pytorch.org/docs/stable/generated/torch.nn.Dropout.html> accessed 16 January 2025

EMNIST’ (*Westernsydney.edu.au,* 2023) <https://www.westernsydney.edu.au/icns/resources/reproducible_research3/publication_support_materials2/emnist > accessed 7 January 2025

GeeksforGeeks, ‘Categorical CrossEntropy in MultiClass Classification’ (*GeeksforGeeks,* 17 September 2024) <https://www.geeksforgeeks.org/categorical-cross-entropy-in-multi-class-classification/> accessed 29 January 2025

GeeksforGeeks, ‘CNN | Introduction to Pooling Layer’(*GeeksforGeeks,* 2024) <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/ > accessed 13 January 2025

GeeksforGeeks, ‘F1 Score in Machine Learning’ (*GeeksforGeeks,* 27 December 2023) <https://www.geeksforgeeks.org/f1-score-in-machine-learning/> accessed 15 January 2025

GeeksforGeeks, ‘What Is a Neural Network?’ (*GeeksforGeeks,* 17 January 2019) <https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/> accessed 29 January 2025.

GeeksforGeeks, ‘Impact of Dataset Size on Deep Learning Model’ (*GeeksforGeeks,* 9, April 2024) <https://www.geeksforgeeks.org/impact-of-dataset-size-on-deep-learning-model/ > accessed 13 January 2025

IBM, ‘Neural Network’ (*Ibm.com,* 6 October 2021) <https://www.ibm.com/think/topics/neural-networks> accessed 29 January 2025.

Irhum Shafkat, ‘Intuitively Understanding Convolutions for Deep Learning’ (*Medium,* June 2018) <https://towardsdatascience.com/intuitively-understanding-convolutions-for-deep-learning-1f6f42faee1 >

Loshchilov I and Hutter F, ‘Decoupled Weight Decay Regularization‘ <https://arxiv.org/pdf/1711.05101 >

Mao A, Mohri M and Zhong Y, ‘Cross-Entropy Loss Functions: Theoretical Analysis and Applications’ <https://arxiv.org/pdf/2304.07288 >

Speciale M, ‘Deep Learning and EMNIST: How to Use a Convolutional Neural Network for Image Recognition’ (*Medium,* 9 August 2023) <https://medium.com/@mspeciale/deep-learning-and-emnist-how-to-use-a-convolutional-neural-network-for-image-recognition-81acbcfa99eb> accessed 17 January 2025

Mishra M, ‘The Curse of Local Minima: How to Escape and Find the Global Minimum’ (*Medium*June, 2023) <https://mohitmishra786687.medium.com/the-curse-of-local-minima-how-to-escape-and-find-the-global-minimum-fdabceb2cd6a > accessed 13 January 2025

O’Shea K and Nash R, ‘An Introduction to Convolutional Neural Networks’ (2015) <https://arxiv.org/pdf/1511.08458 >

Reproducibility — PyTorch 2.5 Documentation’ (*Pytorch.org,* 2023) <https://pytorch.org/docs/stable/notes/randomness.html> accessed 14 January 2025.

Ruder S, ‘An Overview of Gradient Descent Optimization Algorithms *’ <https://arxiv.org/pdf/1609.04747>

‘Scaling Machine Learning Models with Large Datasets and Data Labeling’ (*Sapien.io,* 2024) <https://www.sapien.io/blog/scaling-machine-learning-models-with-large-datasets-and-data-labeling > accessed 13 January 2025

Shai Samana, ‘Mastering Model Complexity: Avoiding Underfitting and Overfitting Pitfalls’ (*Pecan AI,* 13, June 2024) <https://www.pecan.ai/blog/machine-learning-model-underfitting-and-overfitting/ > accessed 13 January 2025

Villalobos P and others, ‘Will We Run out of Data? An Analysis of the Limits of Scaling Datasets in Machine Learning’ <https://arxiv.org/pdf/2211.04325 >, 4 June 2024.

Wohlwend B, ‘Understanding Overfitting and Underfitting in Machine Learning’ (*Medium,* 19 July 2023) <https://medium.com/@brandon93.w/understanding-overfitting-and-underfitting-in-machine-learning-b699e0ed5b28 > accessed 13 January 2025

Xavier Spycy, ‘EMNIST-Classifier/Notebook.ipynb at Main · XavierSpycy/EMNIST-Classifier’ (*GitHub,* 2023) <https://github.com/XavierSpycy/EMNIST-Classifier/blob/main/notebook.ipynb > accessed 7 January 2025

‘What Is a Convolution? How to Teach Machines to See Images | 8th Light’ (*8th Light,* 2022) <https://8thlight.com/insights/what-is-a-convolution-how-to-teach-machines-to-see-images> accessed 7 February 2025

Wu Z, ‘Deep Learning Deterministic Neural Networks’ <https://3dvision.princeton.edu/courses/COS598/2014sp/slides/lecture05_cnn/lecture05_cnn.pdf> accessed 14 January 2025

9. Appendices

Appendix 1.

Box plot with test cross entropy.
https://html-preview.github.io/?url=https://github.com/daniel-padban/GA-Letters/blob/main/graphs/box_test_CE.html

Appendix 2.

Box plot with test F1-score.
https://html-preview.github.io/?url=https://github.com/daniel-padban/GA-Letters/blob/main/graphs/box_test_F1.html

Appendix 3.

Box plot with test accuracy.
https://html-preview.github.io/?url=https://github.com/daniel-padban/GA-Letters/blob/main/graphs/box_test_accuracy.html

Appendix 4.

Box plot with train cross entropy.
https://html-preview.github.io/?url=https://github.com/daniel-padban/GA-Letters/blob/main/graphs/box_train_CE.html

Appendix 5.

Box plot with train accuracy.
https://html-preview.github.io/?url=https://github.com/daniel-padban/GA-Letters/blob/main/graphs/box_train_accuracy.html

Appendix 6.

Box plot with train F1-score.

https://html-preview.github.io/?url=https://github.com/daniel-padban/GA-Letters/blob/main/graphs/box_train_f1.html

Appendix 7.

3D plot with test F1-score.
https://html-preview.github.io/?url=https://github.com/daniel-padban/GA-Letters/blob/main/graphs/test_f1_plot.html

Appendix 8.

3D plot with test cross entropy.
https://html-preview.github.io/?url=https://github.com/daniel-padban/GA-Letters/blob/main/graphs/test_CE_plot.html

Appendix 9.

3D plot with test accuracy.
https://html-preview.github.io/?url=https://github.com/daniel-padban/GA-Letters/blob/main/graphs/test_acc_plot.html

Appendix 10.

3D plot with train F1-score.
https://html-preview.github.io/?url=https://github.com/daniel-padban/GA-Letters/blob/main/graphs/train_f1_plot.html

Appendix 11.

3D plot with train accuracy.
https://html-preview.github.io/?url=https://github.com/daniel-padban/GA-Letters/blob/main/graphs/train_acc_plot.html

Appendix 12.

3D plot with train cross entropy.

https://html-preview.github.io/?url=https://github.com/daniel-padban/GA-Letters/blob/main/graphs/train_CE_plot.html

Appendix 13.

Repository containing all code and files used for this research paper.
https://github.com/daniel-padban/GA-Letters/