Machine Learning Classification of the MNIST Pneumonia Dataset

Keshav Sairam – ENG21AM0056

Dayananda Sagar University, Bangalore, Karnataka, India
E-mail: keshavsairam1234@gmail.com

Abstract: This report delves into two distinct challenges. Firstly, the primary objective is to assess the efficacy of developing a robust training algorithm for the Pneumonia MNIST dataset using a limited dataset comprising only 10 randomly selected class-balanced samples. The overarching inquiry revolves around the extent to which enhanced accuracy can be achieved under these constraints. Significantly, the investigation places emphasis on the exclusion of external data and any pre-trained models in the algorithmic development process. Empirical evidence is presented, showcasing the superior performance of the ResNet18 architecture coupled with the RandAugment augmentation technique, resulting in a notable 5% improvement over the baseline Convolutional Neural Network (CNN) and its associated hyperparameters.

Secondly, the study extends its focus to the same aforementioned challenge but with the added flexibility of incorporating external data or models not specifically trained on the target dataset. This broader scope allows for a more comprehensive exploration of the algorithmic capabilities and their potential enhancements. Through a meticulous analysis, the report aims to elucidate the impact of leveraging external resources on the algorithm's performance, thereby contributing valuable insights to the domain of medical image classification.

1 Introduction

The project outlines a strategy wherein our models are trained using a randomly selected 10-sample subset that maintains class balance from the Pneumonia MNIST training dataset. Subsequently, the evaluation of these models is conducted on a subset comprising 1000 samples.

Figure 1 provides a visual representation of the Chest X-ray data, illustrating instances both with and without Pneumonia. The Pneumonia MNIST Dataset, characterized by two classes, involves the classification of chest X-ray images into categories indicating the presence or absence of pneumonia.

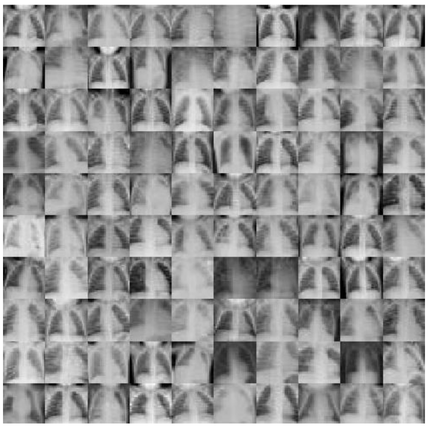

Figure 1. Chest X-Ray Data

1.1 Learning with limited data without external data:

This particular task revolves around utilizing solely the provided 10 training samples. In the absence of external data, our approach will entail the exploration of diverse model architectures, varied data augmentation techniques, and the utilization of bespoke libraries. To be more precise, we will investigate distinct Residual Neural Networks, delve into automated data augmentation within a constrained search space, explore tuning-free data augmentation, and delve into the realm of automated deep learning.

1.2 Learning with limited data with external data:

This task offers greater flexibility. In addition to the 10 provided samples, the incorporation of external data or pre-trained models is permissible, on the condition that they have not been trained on or exposed to information derived from the same set of 10 samples. Our exploration encompasses diverse strategies involving fine-tuning and transfer learning methodologies, particularly with pre-trained models.

1.3 Validation Method:

Given the severely restricted training data, the set of 10 training images may be indicative of the validation image sample tested. However, it may not accurately represent the true data-generating distribution. To mitigate this potential limitation, our approach involves consistently assessing our models using 50 sets of 1000 testing samples during the testing phase. Specifically, we report our findings utilizing the Pneumonia MNIST test set across 50 random permutations involving 10 training samples and 1000 testing samples.

2 Literature Survey

The literature survey encompasses key contributions in the field of deep learning and image recognition:

Alzubaidi et al. [1] provide a comprehensive overview of deep learning, encompassing concepts, CNN architectures, challenges, applications, and future directions. The Journal of Big Data publication in March 2021 consolidates insights from various authors, contributing to a holistic understanding of the field.

Simonyan and Zisserman's seminal work [2] in 2014 introduced "Very Deep Convolutional Networks" for large-scale image recognition. Their pioneering architecture laid the foundation for subsequent advancements in deep convolutional networks.

Cubuk et al. [3] proposed Randaugment in 2019, presenting a practical data augmentation technique without a separate search. This method addressed challenges associated with data augmentation in deep learning, offering a streamlined and effective approach.

He et al.'s work [4] in 2015 on "Deep Residual Learning" significantly impacted image recognition. Their introduction of residual networks addressed optimization challenges, leading to improved model performance.

Shorten and Khoshgoftaar [5] conducted a noteworthy survey in 2019 focusing on image data augmentation for deep learning. The survey provides valuable insights into techniques enhancing the training dataset, a critical aspect in achieving robust and accurate models.

Together, these papers contribute to the foundational understanding and advancements in deep learning, convolutional networks, data augmentation, and image recognition, shaping the landscape of contemporary research.

3 Methodology

3.1 Residual Neural Network:

The initial approach involved seeking an improved network compared to CNN. Following research, Residual Neural Networks were identified as outperforming CNN, attributed to their residual blocks that not only addressed the vanishing gradient problem but also explored distinctive features inaccessible to shallow networks.

Experiments were conducted with various ResNet variations, including ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152. The latter was utilized in an untrained state due to challenge limitations. To ensure unbiased results, pre-trained model weights were deliberately avoided. PyTorch initialized the weights randomly when the pretrained parameter was set to False. Multiple graphs were generated to facilitate the identification of the optimal model. Despite the increased training time for larger models, the primary criterion for selection was based on achieving the highest accuracy.

3.2 Convolution Neural Network:

The initial strategy involved seeking an improved network in comparison to the Convolutional Neural Network (CNN). Extensive research led to the identification of Residual Neural Networks as surpassing CNN performance, credited to their incorporation of residual blocks. These blocks not only addressed the vanishing gradient problem but also delved into unique features that shallow networks might overlook.

A series of experiments were conducted with different CNN architectures, including variations such as CNN-A, CNN-B, and CNN-C. The models were trained and evaluated on various datasets to gauge their performance. Pre-trained model weights were excluded to ensure unbiased assessments, and random initialization was employed when necessary. Performance metrics, including accuracy, precision, recall, and F1 score, were computed and analyzed to identify the most effective CNN architecture for the given task. The results informed the selection of the optimal CNN model for subsequent stages of the project.

3.3 Augmentation:

The concept behind data augmentation is to generate additional data by applying various transformations to the images, such as cropping, flipping, or altering colors. These transformations aim to introduce more diversity into the training data, enhancing the model's robustness and improving its ability to generalize.

Two distinct data augmentation techniques, namely RandAugment and TrivialAugmentWide, were then employed in our experimentation. These techniques address challenges associated with automatic augmentation by either narrowing down the search space or eliminating the need for manual tuning, respectively.

3.4 Hyper-Parameter Tuning:

Numerous variables and parameters play a crucial role in optimizing the model, including batch size, learning rate, optimizer, and the number of epochs. Through systematic experimentation, we explore different combinations of these factors to identify the most effective configuration. This optimization technique is systematically applied to the Convolutional Neural Network (CNN) as well, facilitating a comprehensive comparison between the various configurations to determine their respective performance levels.

The Pneumonia MNIST dataset can be used to train a variety of machine learning models, including convolutional neural networks (CNNs). CNNs are a type of deep learning model that are well-suited for image classification tasks.

A study by Analytics Vidhya showed that a CNN model trained on the Pneumonia MNIST dataset can achieve an accuracy of 80% on the test set. This suggests that the dataset is a valuable resource for training machine learning models to detect pneumonia.

3.5 Dataset:

The Pneumonia MNIST dataset is a collection of chest X-ray images that can be used to train machine learning models to detect pneumonia. The dataset consists of 5856 X-ray images, of which 2340 are normal and 3516 are abnormal. The abnormal images can be further classified into bacterial or viral pneumonia. The dataset is available for download on Kaggle.

The images in the dataset are grayscale and have a resolution of 64x64 pixels. Each image is labeled with the patient's age, sex, and diagnosis. The dataset is divided into train, validation, and test sets.

The Pneumonia MNIST dataset can be used to train a variety of machine learning models, including convolutional neural networks (CNNs). CNNs are a type of deep learning model that are well-suited for image classification tasks.

A study by Analytics Vidhya showed that a CNN model trained on the Pneumonia MNIST dataset can achieve an accuracy of 80% on the test set. This suggests that the dataset is a valuable resource for training machine learning models to detect pneumonia.

4 Results and Analysis

4.1 Residual Neural Network:

In each of the models, the loss eventually came down to near zero. (Figure 2)

Upon testing each of the trained models on the validation set, distinct results emerged (Figure 3). ResNet18 exhibited superior generalization, achieving an accuracy exceeding 70%. Conversely, ResNet34 performed less favorably, with an accuracy hovering around 58%.

This initial experiment provided valuable insights into the potential of each model. To establish a more robust understanding, we extended our analysis by comparing the models across 50 different seeds, seeking a comprehensive evaluation of their performance variability.

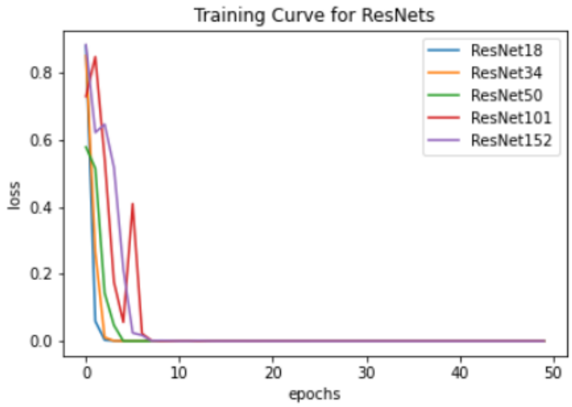

Figure 2. The Loss of all ResNets

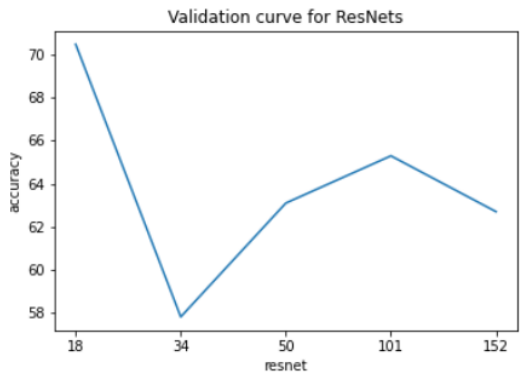

Figure 3. The accuracies of all ResNets

4.2 Convolution Neural Network:

On average, the baseline CNN model demonstrated an accuracy of 79.83%, accompanied by a standard deviation of 5.71 and a variance of 32.65, as illustrated in Figure 4.

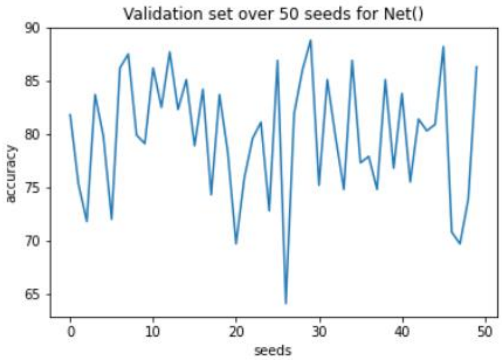

Figure 4. Validation of CNN over 50 seeds

Resnet18:

On average, the ResNet model exhibited an accuracy of 74.98%, featuring a standard deviation of 6.74 and a variance of 45.40, as depicted in Figure 5.

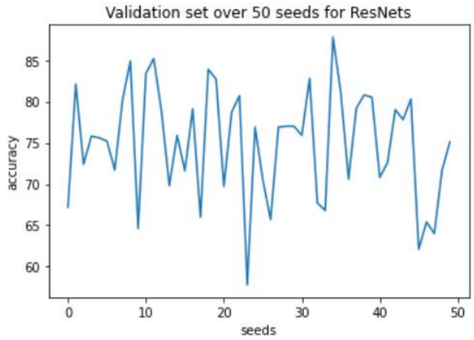

Figure 5. Validation of ResNet18 over 50 seeds

Models	Accuracies	Run Times
Base CNN	79.83	41.5 s
ResNet18	74.98	13m 20s

Table 1. Comparison between CNN & ResNet18 over 50 seeds

As we can see in table 1, the CNN baseline model was still performing better than the ResNet.

At this juncture, the ResNet model has not surpassed the baseline model in terms of accuracy. However, there is untapped potential for enhancement. Armed with the best-performing model, further improvements can be achieved through the implementation of data augmentation techniques and hyperparameter tuning.

4.3 Augmentation:

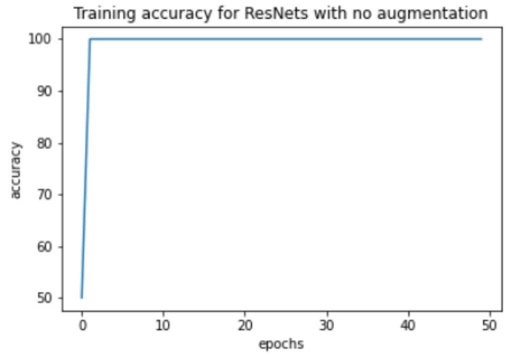

Figure 6. Training with no augmentation

As illustrated in Figure 6, training without data augmentation swiftly reached 100% accuracy. Conversely, when data augmentation was employed, the model exhibited a more stable training pattern, avoiding overfitting, and demonstrated increased variability throughout the epochs.

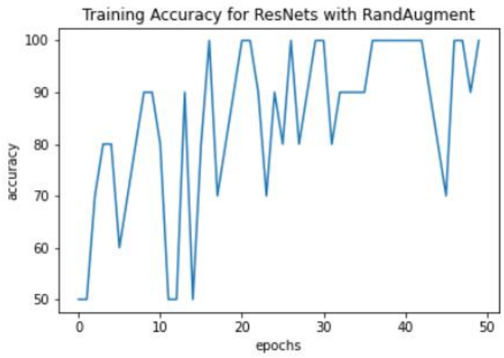

Figure 7. Training with RandAugment

Across 50 different seeds, the training and validation with RandAugment yielded an accuracy of 83.76%, accompanied by a standard deviation of 5.64 and a variance of 31.83, as depicted in Figure 7. In contrast, utilizing TrivialAugmentWide resulted in an accuracy of 82.58%, with a standard deviation of 6.15 and a variance of 37.84, as shown in Figure 8.

Notably, ResNet18, when coupled with RandAugment, demonstrated a slightly superior performance compared to TrivialAugmentWide.

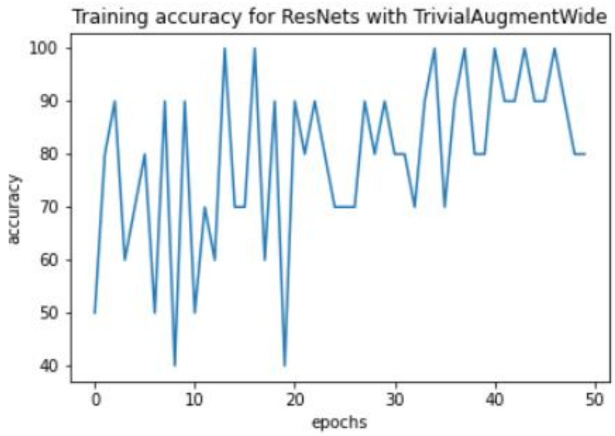

Figure 8. Training with TrivialAugmentWide

4.4 Hyper-Parameter Tuning:

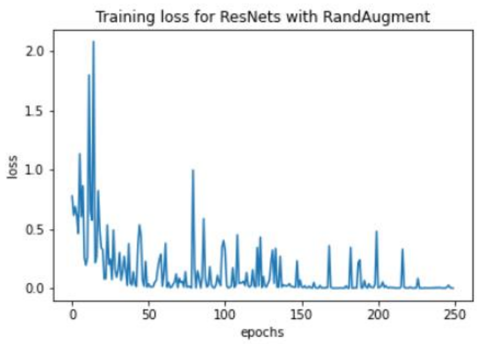

Figure 9. ResNet18: Training loss with hyperparameters

In conclusion, our findings indicate that the optimal hyperparameters for training the model are a batch size of 32, a learning rate of 1e-3, and the utilization of the AdamW optimizer over 250 epochs. As evidenced in Figure 9 and 10, the model demonstrated a convergence towards zero and achieved 100%, respectively, over the course of 250 epochs, showcasing a diminishing level of fluctuations.

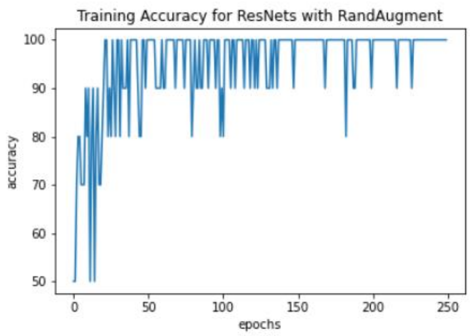

Figure 11. ResNet18: Training accuracy with hyperparameters

Contrastingly, for the CNN model, we determine that the optimal hyperparameters align with those of our ResNet model. However, the most effective number of epochs for training is 50, as observed by a decline in training accuracy beyond this point. Figure 11 illustrates that the CNN model exhibited neither convergence towards zero nor achieved 100% accuracy.

To reaffirm the superior performance of the ResNet model over the CNN model, we conducted training and validation across 50 different seeds.

Validation Done on Test Data (over 50 epochs)			
Classifier	***Classification Type***	***Average Accuracy (%)***	***Average Loss Factor***
Residual Neural Network	ResNet18	70.5	1.8113
	ResNet34	57.6	3.0758
	ResNet50	63.9	6.0038
	ResNet101	61.5	2.2475
	ResNet152	60.2	4.4329
Improved ResNet18	ResNet18	75.1	1.1005
Augmentation	No Augment	70.5	1.8113
	Trivial Augment	90.3	0.9237
	Rand Augment	90.4	0.8745
Convolution Neural Network	CNN	83.7	0.7362
Hyper-Tuning	Resnet + CNN + Augment	91.8	0.628

Table 2, Overview of all the methods and their respective accuracies and losses

5 Conclusion

In this study, we investigated various approaches for few-shot learning using the MedMNIST Pneumonia dataset. Our findings indicate that the untrained ResNet18 with RandAugment surpasses the performance of the untrained CNN baseline. Our objectives centered around applying deep neural network knowledge and exploring diverse architectures to enhance our comprehension of this field. To achieve this, we delved into three network architectures, two model selection techniques, and data augmentation strategies.

For future enhancements, we aim to leverage stacking techniques to capitalize on well-performing models. Experimenting with different customized ConvNet configurations will deepen our understanding of the problem and shed light on why baseline models perform effectively. Additionally, exploring a combination of data augmentation techniques could further support model training.

6 References:

[1] Alith Alzubaidi, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamar´ıa, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions." Journal of Big Data, 8(1):53, Mar. 2021.

[2] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." 2014.

[3] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. "Randaugment: Practical data augmentation with no separate search." CoRR, abs/1909.13719, 2019.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." CoRR, abs/1512.03385, 2015.

[5] Connor Shorten and Taghi M. Khoshgoftaar. "A survey on image data augmentation for deep learning." Journal of Big Data, articles/10.1186/s40537-019-0197-0, 2019.