Optimization and Evaluation of CNN for Brain Tumor Classification: A Replication Study

Ahmad Abdilrahim
Department of Microdata Analysis
Dalarna University
Borlänge, Sweden
v22ahmaa@du.se

Abu Abdullah Dhrubo
Department of Microdata Analysis
Dalarna University
Borlänge, Sweden
h21abudh@du.se

***Abstract*—Convolutional neural networks, in particular, are deep learning algorithms that have quickly taken the lead in medical image analysis. This study rigorously examined a 13-layer Convolutional Neural Network (CNN) architecture for classifying brain tumors using MRI images. This study focused on reproducing previously reported accuracy levels and investigating the influence of hyperparameters on model performance. Through our efforts, we achieved an accuracy of 87.54%, setting a benchmark for further analyses. Notably, hyperparameter tuning conducted on the Fashion MNIST dataset revealed optimized parameters that, when applied to the MRI dataset, improved accuracy to 89.28%. However, a noticeable difference between training and test losses hinted at potential overfitting. Our findings highlight the profound impact of hyperparameter tuning and the inherent challenges of model generalization.**

***Key Words*—Deep Learning, Brain Tumor, Classification, CNN, Hyper-parameter Tuning**

I. Introduction

Brain tumors, characterized by abnormal cellular growths in the brain, are categorized as benign (non-cancerous) or malignant (cancerous) [1]. Symptoms vary based on size and location, encompassing headaches, seizures, speech or memory difficulties, vision impairments, and mood or behavioral changes. Diagnosis typically entails imaging tests like MRI or CT scans, supplemented by biopsies to ascertain the cancerous nature of the tumor [2].

Accurate tumor classification is pivotal for effective treatment planning. Recent advancements in deep learning, particularly Convolutional Neural Networks (CNNs), have shown promise in automating this task using MRI imagery [3] [4]. A 2022 study introduced a 13-layer CNN architecture, claiming high accuracies of 97.2% and 96.9% on different datasets, significantly advancing tumor identification and classification [5].

This study aims to validate the claims made by [5] by reproducing the 13-layer CNN architecture. It further delves into hyperparameter tuning on a smaller dataset to ascertain whether such optimizations maintain or enhance the model's performance accuracy in brain tumor classification when applied to a larger dataset. This endeavor also seeks to identify the most influential hyperparameters, comprehensively evaluating the CNN architecture and contributing to the broader understanding of deep learning's role in Brain Tumor Classification.

A. Problem Definition

Classifying brain tumors accurately from MRI images is pivotal for effective treatment planning. The principal issue stems from the discrepancies observed between the claimed accuracy levels of the 13-layer Convolutional Neural Network (CNN) architecture proposed by [5] and the results obtained upon re-produced. This study aims to validate the purported accuracy, optimize the model through hyperparameter tuning on a smaller dataset, and analyze the key hyperparameters' impact to understand their influence on model performance and the generalizability of optimizations when applied to a larger dataset.

B. Objective of the Study

This study endeavors to replicate and assess the 13-layer Convolutional Neural Network (CNN) architecture introduced by [5] for the classification of brain tumors using MRI images. The primary goals of this investigation include:

Model Replication and Verification: Reproducing the CNN model with the identical dataset and hyperparameters described by [5] and verifying the stated accuracy levels of 97.2% and 96.9%. Our independent tests, utilizing the specified model and hyperparameters, produced an accuracy of 88%. This establishes our benchmark as we were unable to achieve the previously claimed 97.2% accuracy.

Hyperparameter Optimization: Fine-tuning the model by examining various hyperparameter combinations to discern the most effective mix that enhances test accuracy. Upon identifying the optimum combination, we will apply it to the extensive MRI dataset to evaluate if there is an improvement in test accuracy relative to our benchmark.

Hyperparameter Influence Analysis: Identifying the key hyperparameters that have the most significant impact on test accuracy by comparing the configurations of the original model with the optimized variant.

C. Research Hypothesis

Central to our investigation are the following hypotheses:

- Hyperparameter optimization, which encompasses adjustments to aspects like dropout rate, learning rate, kernel size, optimizer function, and batch size (maintained at 10 due to computational constraints), may yield enhanced model performance. Moreover, it is crucial to ascertain if such optimizations on a smaller dataset would generalize effectively to a larger dataset.
- Specific hyperparameters play a pivotal role in determining the accuracy of the model, and identifying these can lead to more effective and efficient model designs.

D. Contribution

This study meticulously re-produces and scrutinizes the 13-layer CNN architecture proposed by [5] to validate or challenge the claimed accuracy levels in brain tumor classification from MRI images. The endeavor into hyperparameter optimization seeks to enhance model performance and identify influential hyperparameters, contributing to a broader understanding of model optimization in CNN architectures for medical image analysis. Analyzing the generalizability of optimized hyperparameters from a smaller to a larger dataset provides insights into the scalability and real-world applicability of the CNN model in clinical settings. Furthermore, the study proposes avenues for future research, laying a groundwork for subsequent investigations to refine CNN architectures in brain tumor classification, thereby advancing the efficacy and accuracy of deep learning models in medical image analysis for better patient care and treatment planning.

II. Literature Review

The task of accurately classifying brain tumors from Magnetic Resonance Imaging (MRI) images is pivotal in the medical domain, contributing significantly towards effective treatment planning and patient care. The complexity and heterogeneity of brain tumors necessitate advanced techniques for precise classification. In recent years, Convolutional Neural Networks (CNNs) have emerged as a potent tool in automating such classification tasks, demonstrating notable success in image recognition and classification domains [6].

This paper from LeCun et al. [3] provides a thorough examination of deep learning's transformative impact across various domains, emphasizing the crucial role of neural networks, especially deep neural networks, in surmounting previous machine learning limitations and achieving remarkable advancements in fields like speech recognition, image analysis, and natural language processing. The authors highlight the significance of representation learning and the ability of deep networks to unravel complex structures within large datasets. Various deep learning architectures are explored, including convolutional neural networks (CNNs) for image-related tasks and recurrent neural networks (RNNs) for sequence data, focusing on the backpropagation algorithm as a fundamental training method. The text also addresses challenges such as vanishing gradients. It introduces memory-augmented networks, like long short-term memory (LSTM) networks, to enhance performance on tasks demanding an understanding of long-range dependencies. Despite these challenges, the authors note deep learning's unprecedented success in numerous applications, outperforming traditional methods and showcasing its potential to handle complex tasks requiring contextual and sequential understanding. However, the document underscores the necessity for ongoing research and innovation to address persistent challenges and propel the field forward. Deep learning, with CNNs at its forefront, has shown remarkable promise in medical image analysis. LeCun et al. [3] and Litjens et al. [4] expound on the capabilities of CNNs in handling image-based tasks by automatically and adaptively learning spatial hierarchies from data. The architectural depth and ability to learn feature representations from images make CNNs a suitable choice for medical image analysis, including brain tumor classification.

This study from Kibriya et al. [5] highlights the proposed CNN method's superior performance compared to other existing methods, achieving a high accuracy of 97.2% on a public dataset and 96.9% on a Kaggle dataset. The architecture of the proposed CNN is described as consisting of 13 layers, including convolutional layers, batch normalization layers, max-pooling layers, a dropout layer to prevent overfitting, a fully connected layer, a softmax layer, and a classification layer. The paper underscores the efficiency and lightweight nature of the proposed method due to its fewer layers and a smaller number of trainable parameters in comparison to existing deep CNN architectures. Moreover, the proposed CNN eliminates the need for manual lesion segmentation before classification, making it a robust and effective solution for real-time brain tumor classification. The authors conclude by expressing the promising performance of their proposed strategy and their intention to explore different databases with varied imaging modalities in future work to enhance the network's generalization capability.

This literature review underscores the prevailing efforts and advancements in employing CNNs for brain tumor classification from MRI images. It also establishes the necessity of the present study in validating and possibly enhancing the 13-layer CNN architecture proposed by Kibriya et al. [5], contributing further to the burgeoning field of deep learning in medical image analysis.

III. Methodology

This section outlines the methodology adopted in this research, including the description of the dataset, data preprocessing, CNN Architecture, model development and validation, and optimization of the parameters.

A. Dataset

In this study, we have primarily used two distinct datasets. The brain tumor dataset consists of 3064 T1-weighted contrast-enhanced grayscale images collected from 233 patients diagnosed with three types of brain tumors: meningioma (708 slices), glioma (1426 slices), and pituitary tumor (930 slices). The dataset aims to provide a basis for developing and evaluating machine learning algorithms in the medical

MRI image analysis domain, particularly for brain tumor classification. The dataset was shared by Jun Cheng in 2017 on Figshare with the title ”brain tumor dataset” [7].

TABLE I
DATASET ATTRIBUTES

Attribute	Description
label	Indicates the type of brain tumor, with values 1 for meningioma, 2 for glioma, and 3 for pituitary tumor.
image	Contains the image data.

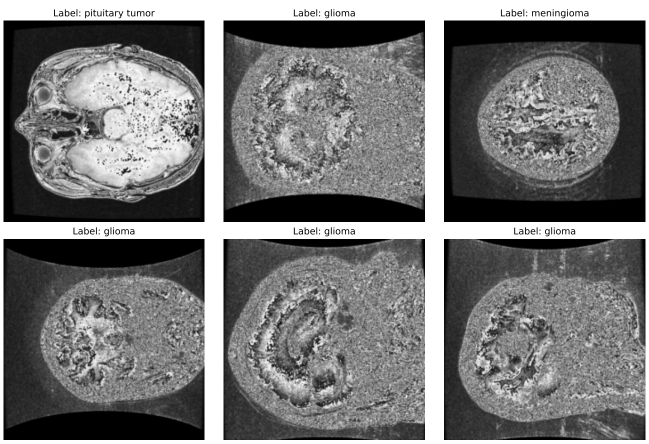

Fig. 1. MRI

The other dataset is the fashion-MNIST dataset is a dataset comprising 60,000 training examples and 10,000 testing examples. Each example is a 28x28 grayscale image associated with a label from 10 classes. It's designed to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It provides a more challenging problem than the often-used MNIST dataset. The dataset was introduced by Han Xiao, Kashif Rasul, and Roland Vollgraf in a 2017 paper titled ”Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms” [8].

B. Data Preprocessing

Data preprocessing is a critical initial step in our pipeline, aiming to prepare and enhance the quality of MRI images for the subsequent tasks of model training and evaluation. The dataset utilized in this study comprises MRI brain images, each encapsulated within a MATLAB (.mat) file. To facilitate the processing and loading of this data into our model, we implemented a custom dataset class, BrainTumorDataset, which inherits from PyTorch's Dataset class.

Within this class, the *__getitem__* method has been defined to load an MRI image and its corresponding label from a given .mat file. The image data is extracted, converted into a PIL Image format, and subsequently transformed using a series of pre-defined transformations. It is imperative to highlight that the images contained in the MRI dataset exhibit diverse intensity ranges. We applied a Min-Max Normalization technique to mitigate this variability and standardize the intensity levels across all images. This normalization adjusts the intensity values of each image to a standard range of [0, 255], which not only aids in reducing potential biases in the model due to intensity variations but also aligns the intensity range with that of conventional 8-bit images [5].

Following the normalization, the images undergo a resizing operation to transform them into a uniform size of 227x227 pixels. This dimension has been chosen to comply with the input size requirement of the subsequent convolutional neural network [5]. Additionally, the images are converted to PyTorch tensors, the required format for input into the neural network models. This series of transformations is encapsulated within the *transforming_img* object using PyTorch's *transforms.Compose*.

To bolster the robustness of our methodology, we extended our preprocessing to another dataset, FashionMNIST, to validate the generalizability of our findings. The FashionMNIST dataset undergoes similar preprocessing steps, including Min-Max Normalization and conversion to PyTorch tensors. However, given the inherent differences like the data (grayscale fashion item images versus MRI brain images), the resizing operation is omitted for FashionMNIST.

C. CNN Architecture

In this study, we precisely implemented a 13-layer Convolutional Neural Network (CNN) based on the architecture detailed in the work of [5]. The network was structured to comprise a series of convolutional layers, interspersed with max-pooling layers, and followed by fully connected layers towards the end.

The initial layer of our network was a convolutional layer, which was responsible for extracting low-level features from the input images. This layer utilized filters with small receptive fields to scan through the input image and create feature maps. Following this, we applied a rectified linear unit (ReLU) activation function to introduce non-linearity to the model, enabling it to learn complex patterns from the data.

Subsequent layers in the network consisted of additional convolutional layers, designed to delve deeper and capture more intricate patterns and textures from the input data. To prevent the model from becoming too computationally expensive, and to reduce the spatial dimensions of the feature maps, we interspersed max-pooling layers at strategic points in the network. These layers worked by downsampling the feature maps, retaining only the most significant information, and discarding the rest.

As we progressed deeper into the network, the filters in the convolutional layers increased in number, allowing the model to detect a wider variety of features and patterns in the input data. This architectural choice was grounded in the understanding that lower layers capture basic features, while deeper layers amalgamate these basic features to recognize more complex patterns [5].

The final layers were fully connected, crucial in integrating the learned features and generating the final output. Before

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1         [-1, 32, 223, 223]             832
              ReLU-2         [-1, 32, 223, 223]               0
       BatchNorm2d-3         [-1, 32, 223, 223]              64
         MaxPool2d-4           [-1, 32, 74, 74]               0
            Conv2d-5           [-1, 64, 70, 70]          51,264
              ReLU-6           [-1, 64, 70, 70]               0
       BatchNorm2d-7           [-1, 64, 70, 70]             128
         MaxPool2d-8           [-1, 64, 23, 23]               0
           Dropout-9                [-1, 33856]               0
           Linear-10                    [-1, 3]         101,571
          Softmax-11                    [-1, 3]               0
================================================================
Total params: 153,859
Trainable params: 153,859
Non-trainable params: 0
----------------------------------------------------------------
```

Fig. 2. CNN Architecture for MRI

reaching these layers, the multidimensional feature maps were flattened into a one-dimensional vector, serving as input for the fully connected layers. Given our network's depth and complexity, we incorporated dropout layers to mitigate the risk of overfitting, randomly deactivating a subset of neurons during training to ensure that no single neuron became overly specialized.

The culmination of our network was a softmax activation function, transforming the output into probability scores corresponding to the different classes in our classification tasks. This enabled a clear and interpretable output, indicating the model's prediction for a given input image.

In designing and implementing this CNN architecture, we focused on capturing the intricate patterns inherent in the datasets, particularly the MRI dataset, with its nuanced and detailed imagery. The depth and complexity of our network were pivotal in achieving this, allowing the model to learn and make predictions with a high degree of accuracy.

D. Modeling and Validation

The modeling and validation were initiated with reproducing the 13-layer Convolutional Neural Network (CNN) architecture described in [5], utilizing the identical dataset and hyperparameter settings. This step was crucial for achieving their accuracy levels of 97.2%, However, when we attempted to reproduce their results, we fell short of reaching this accuracy. This discrepancy led us to reference the parameters from [5] in their study and establish our own benchmark due to our inability to replicate their results. Then, a comprehensive hyperparameter tuning was conducted on a smaller dataset (Fashion MNIST) to identify the most effective combination of learning rates, batch sizes, optimizer functions, dropout rates, and kernel sizes. The best-performing hyperparameters were then applied to the larger MRI dataset to assess the generalizability of the optimizations and their impact on model performance. Various metrics, including precision, recall, accuracy, and F1 score, were calculated for validation purposes, providing a holistic view of the model's performance and the effectiveness of the applied optimizations. This iterative process of modeling, hyperparameter tuning, and validation was instrumental in understanding the key factors influencing performance and ensuring the robustness of the model across different datasets.

E. Optimizing the parameters

The conventional approach to optimizing hyperparameters is a grid search, which entails thorough searches throughout a specified fraction of the training algorithm's hyperparameter space. This technique exhaustively tests a predefined set of hyperparameter values to identify the most effective combination for a Convolutional Neural Network (CNN) [9]. In this instance, the hyperparameters under consideration include learning rates, batch sizes, optimizer types, dropout rates, and kernel sizes.

Grid search operates by constructing a Cartesian product of the provided hyperparameter values, resulting in a comprehensive set of configurations to be tested. Here, the learning rates are [0.0001, 0.00001, 0.000001], batch sizes are [10, 32, 64], optimizers include Stochastic Gradient Descent (SGD) and Adam, dropout rates are set at [0.3, 0.5, 0.6], and kernel sizes are [3, 4, 5]. The search space, therefore, consists of 162 different combinations, each of which is trained for 10 epochs, providing a thorough evaluation of the model's performance across a variety of settings.

In a grid search, the primary consideration is the balance between comprehensiveness and computational efficiency. While the exhaustive nature of grid search ensures that every possible combination is evaluated, leading to a reliable identification of the optimal parameters, it also demands substantial computational resources and time, especially as the number of hyperparameters and their potential values increase [9].

Moreover, the chosen ranges and values of hyperparameters are crucial, as they need to be sufficiently diverse to explore different model behaviors yet relevant to the problem at hand to ensure the practical applicability of the results. The granularity of the grid (i.e., how finely or coarsely the hyperparameter values are spaced) also plays a vital role, as finer grids provide more detailed insights but at the expense of increased computational load.

IV. Results

The overarching objective of this research was to investigate the performance of a 13-layer Convolutional Neural Network (CNN) for brain tumor classification using MRI images. The study aimed at reproducing the CNN model as presented in [5] and delving into hyperparameter optimization to potentially improve its efficacy.

A. Model Replication and Verification

Initially, the model was re-produced using the identical dataset and hyperparameters as prescribed by [5]. It was observed as seen in table III that the reproduced model achieved a training accuracy of 88.37% and a testing accuracy of 87.54% on the MRI image dataset. In comparison, the authors of [5] claimed an accuracy level of 97.2%. The training and testing

TABLE II
OPTIMIZED PARAMETERS MODEL PERFORMANCE OVER ITERATIONS

No	Train Accuracy %	Test Accuracy %	Train Loss	Test Loss
0	95.40	87.25	0.5974	0.6818
1	98.20	86.38	0.5700	0.6815
2	91.60	84.49	0.6364	0.7043
3	97.89	87.39	0.5730	0.6752
4	97.89	85.36	0.5745	0.6920
5	98.01	84.78	0.5714	0.7034
6	98.76	86.23	0.5635	0.6886
7	97.01	88.26	0.5816	0.6650
8	97.26	88.55	0.5790	0.6635
9	97.76	89.28	0.5734	0.6575

TABLE III
OUR EXPERIMENTAL RESULTS

Model_Dataset	Accuracy	Precision	Recall	F1 Score
Initial_MRI	0.8754	0.8892	0.8838	0.8804
Initial_FMNIST	0.8885	0.8913	0.8885	0.8894
Optimized_FMNIST	0.8982	0.8983	0.8982	0.8981
Optimized_MRI	0.8928	0.8912	0.9068	0.8951

losses were recorded at 0.6672 and 0.6744, respectively. For the performance metrics, the model scored a precision of 88.92%, a recall of 88.38%, an accuracy of 87.54%, and an F1 score of 88.04%.

The same architecture was applied to the Fashion MNIST dataset to evaluate the model's flexibility. The results showed a train accuracy of 89.59% and a test accuracy of 88.85%, with corresponding train and test losses being 1.5651 and 1.5726.

B. Hyperparameter Optimization on Fashion MNIST

A hyperparameter tuning exercise was undertaken using the Fashion MNIST dataset. The varied parameters included learning rates, batch sizes, optimizer types, dropout rates, and kernel sizes. The experiment was set to run for 10 epochs for various combinations.

From the hyperparameter tuning testing, the most optimal combination yielded a test accuracy of 88.77%, which is the highest of all combinations. The best-performing hyperparameters were:

- Learning Rate: 0.0001
- Batch Size: 10
- Optimizer: Adam
- Dropout Rate: 0.3
- Kernel Size: 3

Upon integrating these optimized hyperparameters into the model and testing it again on the Fashion MNIST dataset, the model demonstrated a noticeable improvement. It achieved a training accuracy of 90.84%, test accuracy of 89.82%, and train and test losses of 1.5540 and 1.5643, respectively. This translated into a precision of 89.83%, a recall of 89.82%, an accuracy of 89.82%, and an F1 score of 89.81%.

C. Application of Optimized Hyperparameters on MRI Dataset

The study's ultimate test was to evaluate the MRI dataset using the model optimized with hyperparameters derived from the Fashion MNIST dataset. Intriguingly, this significantly enhanced the training accuracy, reaching 97.76%. However, the test accuracy stood at 89.28%. The recorded losses for this configuration were 0.5734 (training) and 0.6575 (testing). When we delve into the performance metrics, the precision was 89.12%, the recall was 90.68%, the accuracy was 89.28%, and the F1 score was 89.51% as seen in table II and III. To conclude the result, our hyperparameter tuning yielded the best parameter combination, improving accuracy by around 0.0174 (1.74%).

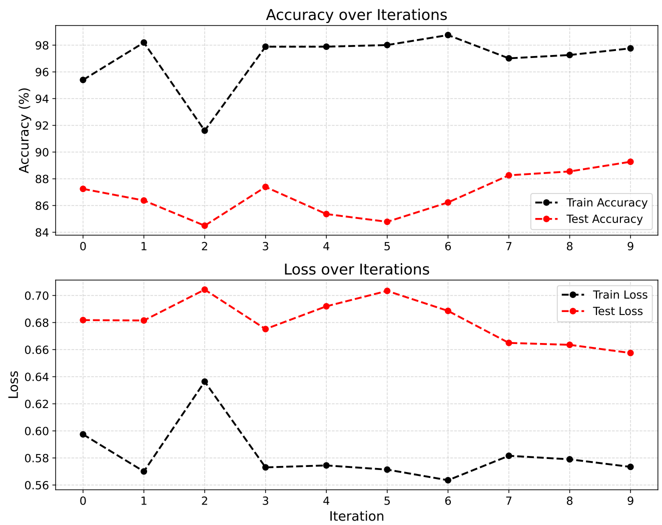

Fig. 3. Optimized CNN accuracy and losses on MRI dataset

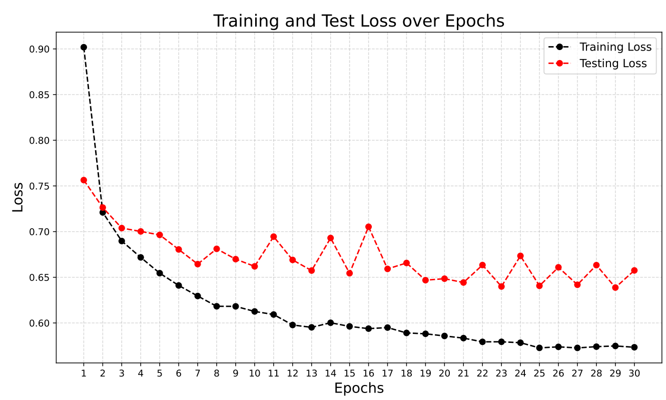

Fig. 4. Optimized CNN losses on 9th itr for MRI

The figure 4 above, shows that the model is overfitting while achieving slightly better accuracy than the initial train and test accuracy of the base model.

V. DISCUSSION

The results of our study highlight the inherent flexibility and resilience of the 13-layer CNN. Replicating the model proposed by [5], our observed accuracy was considerably lower than the reported 97.2%. This divergence underlines the imperative need for clarity and transparency in research methodologies. Various factors, including dataset specifics, computational environment, initialization procedures, and even random seeds, can influence replication results. Owing to the disparity in achieving the acclaimed 97.2% accuracy, we set our benchmark based on the 87.54% accuracy from our replication, which served as a reference for subsequent evaluations.

Our experimentation with hyperparameter tuning on the Fashion MNIST dataset underscored the sensitivity of the model's performance to hyperparameter configurations. Even minor tweaks in these parameters can trigger noticeable changes in the model's efficiency, emphasizing the cardinal role of hyperparameter optimization. This is especially true in fields like medical imaging, where the margin for error is minimal, and high accuracy is non-negotiable.

Intriguingly, our study further showcased that hyperparameter optimization, derived from a seemingly unrelated dataset (Fashion MNIST), could be generalized effectively to the MRI dataset. This phenomenon stresses the potential universality of certain hyperparameter configurations across distinct datasets, which is a significant observation in itself.

Upon analyzing the performance metrics post-optimization, specific hyperparameters have a more pronounced effect on the test accuracy. The optimized learning rate, kernel size, and the choice of the Adam optimizer were standout contributors. The adjusted dropout rate, set at 0.3, likely played a role in mitigating overfitting to some extent, even if not entirely. These findings suggest that while some hyperparameters are universally impactful across datasets, others require fine-tuning depending on the specificity and complexity of the data in question.

Yet, it is worth noting that while the optimized model showed improvements in training and test accuracy, nearing the figures posited by [5], the test accuracy was still not on par. This discrepancy could be indicative of overfitting [10]. Despite the model's adeptness at learning from the training dataset (evidenced by the high training accuracy), its generalization capabilities on unseen data might be compromised.

Conclusively, while our research vouches for the viability of the CNN model by [5], it also elucidates the significant influence of hyperparameters. Our results lay the groundwork for future explorations, with an emphasis on not just hyperparameter optimization but also strategies to circumvent overfitting, ensuring robustness in both training and test datasets. Future endeavors in this domain could also expand the range of hyperparameters explored and deploy more sophisticated optimization techniques for an all-encompassing model fine-tuning.

VI. Conclusion

This study delved into the replication and optimization of a 13-layer CNN proposed by [5], spotlighting the complexities and nuances of model performance, particularly in medical imaging. Our results clarify that achieving reported accuracies is contingent upon many factors, emphasizing the need for thoroughness and transparency in research methodologies. While our replication could not match the acclaimed 97.2% accuracy, the benchmark we established provided invaluable insights into the model's robustness and the transformative potential of hyperparameter tuning. Interestingly, optimized hyperparameters from an unrelated dataset (Fashion MNIST) proved beneficial for the MRI dataset, revealing the importance of certain hyperparameters across diverse datasets. However, the persistent gap between training and test accuracies suggests challenges with overfitting, a topic warranting deeper exploration. Ultimately, our study not only underscores the potential of the CNN architecture in question but also charts a roadmap for future research—highlighting the indispensability of hyperparameter optimization, the perils of overfitting, and the promise of iterative refinement in machine learning models.

VII. Limitations

For any machine learning study, selecting ranges and values for hyperparameters is pivotal, as they should be diverse enough to explore different model behaviors while remaining relevant to the problem at hand to ensure the practical applicability of the results. However, due to limited computational power, our ability to explore a broader or more granular range of hyperparameter values was constrained. Although finer grids in the hyperparameter space provide more detailed insights, they significantly increase the computational load. Our computational constraints necessitated a balance, which, in turn, may have impacted the comprehensiveness of our hyperparameter tuning process, limiting the number of parameter combinations we could explore.

References

[1] C. Clinic, "Brain tumor: Symptoms, signs causes," 2 2022. [Online]. Available: https://my.clevelandclinic.org/health/diseases/6149-brain-cancer-brain-tumor

[2] C. Net, "Brain tumor - diagnosis," 5 2023. [Online]. Available: https://www.cancer.net/cancer-types/brain-tumor/diagnosis

[3] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," vol. 521, no. 7553, pp. 436–444, 5 2015. [Online]. Available: https://www.nature.com/articles/nature14539

[4] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. v. d. Laak, B. v. Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 12 2017. [Online]. Available: https://doi.org/10.1016/j.media.2017.07.005

[5] H. Kibriya, M. Masood, M. Nawaz, and T. Nazir, "Multiclass classification of brain tumors using a novel cnn architecture," *Multimedia Tools and Applications*, vol. 81, no. 21, pp. 29 847–29 863, 2022.

[6] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights into Imaging*, vol. 9, pp. 611–629, 2018.

[7] J. Cheng, "Brain tumor dataset," 2017. [Online]. Available: https://figshare.com/articles/dataset/brain_tumor_dataset/1512427

[8] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," 2017.

[9] P. Liashchynskyi and P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for nas," *arXiv preprint arXiv:1912.06059*, 2019.

[10] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM Computing Surveys (CSUR)*, vol. 27, no. 3, pp. 326–327, 1995.