**Brain Tumor Detection Using Deep Learning**



Student: Jenan Sarabi

Supervisor: Antonio Prgomet

EC Utbildning

Data Science 22- Examensarbete

2024-Feb

# Abstract

This study investigates the efficacy of Convolutional Neural Networks (CNNs) in detecting brain tumors from MRI images. The research explores various deep learning techniques, particularly focusing on CNN architecture, to enhance medical image analysis. By optimizing model configurations and hyperparameters such as dropout rates, batch sizes, and epochs, the study aims to improve CNN performance in accurately identifying brain tumors. Results indicate that dropout regularization plays a crucial role in preventing overfitting, thus enhancing model robustness. The findings contribute valuable insights into deep learning methodologies for brain tumor detection, ultimately advancing healthcare through more efficient and reliable diagnosis.

# Acknowledgments

I extend my heartfelt gratitude to Antonio for his unwavering support, guidance, and mentorship throughout this journey. His expertise and encouragement have been instrumental in shaping my understanding and proficiency in data science. I am truly grateful for his dedication and invaluable insights that have enriched my learning experience.

Additionally, I would like to acknowledge the unwavering support of my family. Their love, encouragement, and understanding have been a constant source of strength and motivation, enabling me to pursue my academic and professional goals with determination.

# Keywords

Magnetic Reasoning Imaging (MRI)

Machine Learning (ML)

Deep Learning (DL)

Artificial Intelligence (AI)

Convolutions Neural Network (CNN)

Residual Neural Network (ResNet)

**Skapas automatiskt i Word genom att gå till Referenser> Innehållsförteckning.**

Table of Contents

[Abstract 2](#_Toc159592663)

[Acknowledgments 3](#_Toc159592664)

[Keywords 4](#_Toc159592665)

[1 Introduction 1](#_Toc159592666)

[1.1 Project Purpose 1](#_Toc159592667)

[1.2 Questions steer the project: 1](#_Toc159592668)

[2 Teori 2](#_Toc159592669)

[2.1 Deep Learning 2](#_Toc159592670)

[2.1.1 Neuron 2](#_Toc159592671)

[2.1.2 Models: CNN 5](#_Toc159592672)

[3 Metod 7](#_Toc159592673)

[3.1 Data set 7](#_Toc159592674)

[3.1.1 Data Collection and Preparation: 7](#_Toc159592675)

[3.2 Models 8](#_Toc159592676)

[3.2.1 CNN model 8](#_Toc159592677)

[**3.2.2** ResNet 8](#_Toc159592678)

[3.2.3 Model Training: 8](#_Toc159592679)

[**3.2.4** ModelEvaluation**:** 8](#_Toc159592680)

[3.2.5 Prediction and Analysis: 9](#_Toc159592681)

[4 Reslust 10](#_Toc159592682)

[4.1 Analysis of Dropout Experiments in CNN Models 10](#_Toc159592683)

[4.2 Analysis of Dropout Experiments in ResNet Model 13](#_Toc159592684)

[5 Conclusion 15](#_Toc159592685)

[6 Resferences 16](#_Toc159592686)

[Appendix A 17](#_Toc159592687)

# Introduction

This research looks at how smart computer brains, like Convolutional Neural Networks (CNNs), get good at finding brain tumors in special pictures called MRI images. It helps doctors make better plans for treating patients and makes healthcare better overall.

Think about it - have you ever wondered how doctors spot brain tumors in those detailed MRI pictures? It's a big deal because it gives patients a fighting chance.

Now, imagine these super-smart computer brains (CNNs) jumping in, mastering the skill of finding brain tumors in images. They act like smart assistants for doctors, making healthcare even better.

## Project Purpose

The purpose of this research project is to develop a Convolutional Neural Network (CNN) model for identifying brain tumors in MR images. By exploring various deep learning techniques, particularly focusing on CNN architecture, the aim is to enhance skills in medical image analysis. Through experimentation with different model configurations, such as dropout rates and batch sizes, the goal is to optimize CNN’s performance in accurately detecting brain tumors. The overarching objective is to contribute to advancements in healthcare by improving the efficiency and reliability of brain tumor diagnosis using deep learning methodologies.

## Questions steer the project:

* How effective is the CNN model in accurately detecting brain tumors?
* How do adjustments in dropout rates, batch sizes, and epochs affect the CNN model's performance?
* What challenges arise in the development and deployment of the model?

# Teori

## Deep Learning

Deep Learning, a subset of machine learning, excels in recognizing patterns, particularly in images, using multiple layers in artificial neural networks. This method demands a substantial amount of data for effective training, allowing each layer to extract specific features and enhance the system's pattern understanding and identification capabilities.

### Neuron

The neuron is a key part of neural networks, like a basic building block. It works like human brain neurons, processing information with weighted inputs and using activation functions for output. Neurons are organized into layers in a network. Parameters include weights, summation, and activation functions.

A diagram of a neural network

Description automatically generated

Figure-1 Architecture of a Perceptron with two input neurons, one bias neuron, and three output neurons

#### Weight

Neurons multiply incoming signals by weights, and if a neuron has three inputs, it adjusts three weights during training.

#### Bias

A bias is an extra input always set to 1, ensuring neuron activity even when all other inputs are zero.

#### Activation Function

Also known as a transfer function, it helps neurons decide when to " activate" by using a threshold, simplifying classification into different groups.

#### Confusion Matrix Components

The confusion matrix is a table used in classification to evaluate the performance of a machine learning model. It consists of four components:

**True Positives (TP):**

Definition: The number of instances that are actually positive and were correctly predicted as positive by the model.

Interpretation: TP represents the model's ability to correctly identify positive cases.

**True Negatives (TN):**

Definition: The number of instances that are actually negative and were correctly predicted as negative by the model.

Interpretation: TN represents the model's ability to correctly identify negative cases.

**False Positives (FP):**

Definition: The number of instances that are actually negative but were incorrectly predicted as positive by the model.

Interpretation: FP represents the model's errors in predicting positive cases.

**False Negatives (FN):**

Definition: The number of instances that are actually positive but were incorrectly predicted as negative by the model.

Interpretation: FN represents the model's errors in predicting negative cases.

These components are typically arranged in a matrix format, which is known as the confusion matrix. The matrix allows a more detailed understanding of the model's performance, especially in terms of its ability to correctly classify instances belonging to different classes. The confusion matrix is widely used for evaluating the effectiveness of classification algorithms in various applications.

#### Overfitting

‘’Overfitting the Training Data Say you are visiting a foreign country, and the taxi driver rips you off. You might be tempted to say that all taxi drivers in that country are thieves. Overgeneralizing is something that we humans do all too often, and unfortunately machines can fall into the same trap if we are not careful. In Machine Learning this is called overfitting: it means that the model performs well on the training data, but it does not generalize well.’’

‘’ Overfitting happens when the model is too complex relative to the amount and noisiness of the training data’’ 27 -28 Hands on Machine learning

#### **Regularization**

Regularization simplifies models by constraining parameters, balancing between fitting training data and preventing overfitting for better generalization.

‘’To prevent overfitting in neural networks, regularization techniques are essential. In deep neural networks with tens of thousands or even millions of parameters, the model has extensive flexibility, allowing it to fit complex datasets. However, this flexibility also increases the risk of overfitting the training data. Regularization methods, such as early stopping and Batch Normalization, help mitigate this issue by imposing constraints on the model's parameters or reducing its complexity. Other effective techniques include ℓ1 and ℓ2 regularization, dropout, and max-norm regularization. ‘’

’’With four parameters I can fit an elephant and with five I can make him wiggle his trunk.’’

-John von Neumann, cited by Enrico Fermi in Nature 427

##### **2.1.1.8 Dropout**

Dropout is a regularization technique used in neural networks during training, where a fraction of neurons is randomly dropped out. This helps prevent overfitting by reducing the reliance on specific neurons and encourages the network to learn more robust features, leading to improved generalization performance.

A diagram of a diagram

Description automatically generated

Figure-2 Dropout regularization, at each training iteration a random subset of all neurons in one or more layers—except the output layer—are “dropped out”; these neurons output 0 at this iteration (represented by the dashed arrows)

### Models: CNN

Convolutional Neural Networks (CNNs) are advanced computer programs designed to understand images. They break down pictures into smaller parts, analyze them to recognize patterns, and then piece everything together to make sense of the whole image. They learn from lots of examples, getting better over time at tasks like identifying objects in photos or analyzing medical images. CNNs are used in many areas, from helping doctors diagnose diseases to powering facial recognition on your phone. They're like super-smart detectives for images, making sense of the visual world around us in ways that were once thought impossible.

CNNs consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. Here's a brief overview of each:

Convolutional Layers: These layers apply a set of learnable filters to input images. Each filter slides over the input image, performing element-wise multiplication and summation, which results in a feature map. These filters are designed to detect specific patterns or features, such as edges, textures, or shapes, at different locations in the image. By stacking multiple convolutional layers, CNNs can learn increasingly complex features.

Pooling Layers: Pooling layers downsample the feature maps produced by the convolutional layers, reducing their spatial dimensions. Common pooling operations include max pooling and average pooling, which take the maximum or average value within a window of the feature map, respectively. Pooling helps to make the learned features more invariant to small translations and distortions in the input image, while also reducing the computational burden.

Fully Connected Layers: After several convolutional and pooling layers, the resulting feature maps are flattened into a one-dimensional vector and passed through one or more fully connected layers. These layers act as a traditional neural network, with each neuron connected to every neuron in the previous layer. Fully connected layers combine the high-level features learned by the convolutional layers to make predictions or classifications.

(Source: Adapted from "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville)

CNNs are trained using a variant of gradient descent called backpropagation, where the network adjusts its internal parameters (weights and biases) based on the error between predicted and true labels. This training process enables the CNN to automatically learn discriminative features directly from raw pixel values, without requiring manual feature extraction.

Overall, CNNs have revolutionized the field of computer vision and have achieved state-of-the-art performance on various image-related tasks, including object recognition, image classification, segmentation, and more.

**ResNet**, also known as Residual Networks, is a significant advancement in convolutional neural networks (CNNs), as explained by Aurélien Géron." ResNet tackles the challenge of training very deep neural networks by introducing shortcuts, called skip connections. These shortcuts help gradients flow more smoothly during training, solving the vanishing gradient problem and enabling the training of deeper networks. ResNet architecture mainly consists of residual blocks, which contain convolutional layers and identity mappings. By using these residual connections, ResNet achieves remarkable performance in tasks like image classification and object detection. Géron's book provides detailed insights into ResNet's architecture, implementation, and applications in deep learning.

# Metod

The development of a Convolutional Neural Network (CNN) model designed to classify MRI brain images into two categories: tumor and normal. Below is a comprehensive overview of each stage in the process:

## Data set

The dataset used in this project is sourced from Kaggle and consists of MRI images of brain tumors. It is divided into two main folders: one containing images of normal brains and the other containing images of brains with tumors. Specifically, there are 155 images of healthy brains and 98 images of brains with tumors, making a total of 253 images in the dataset. This dataset offers a valuable resource for tasks such as classification and analysis related to brain tumor detection.

### Data Collection and Preparation:

The first step of the process is gathering MRI brain images categorized as either tumor or healthy from two separate directories labeled 'yes' and 'no'. These images are then resized to a standardized resolution of 128x128 pixels. This resizing step is crucial because the original data may contain images of varying sizes. Standardizing the size ensures consistency across the dataset, which aids the AI model in learning and identifying patterns effectively. After resizing, the images are converted into NumPy arrays, a format suitable for machine learning algorithms.

Here is sample of images with different size

Image dimensions (width x height): 319 x 360

Image dimensions (width x height): 240 x 300

Furthermore, all images in the dataset have varying color channels, including 1 and 3 and 4 which provide comprehensive information for analysis. This RGB (Red, Green, Blue) color scheme enables a more detailed representation of the images, allowing the AI model to extract nuanced features during both training and evaluation phases.

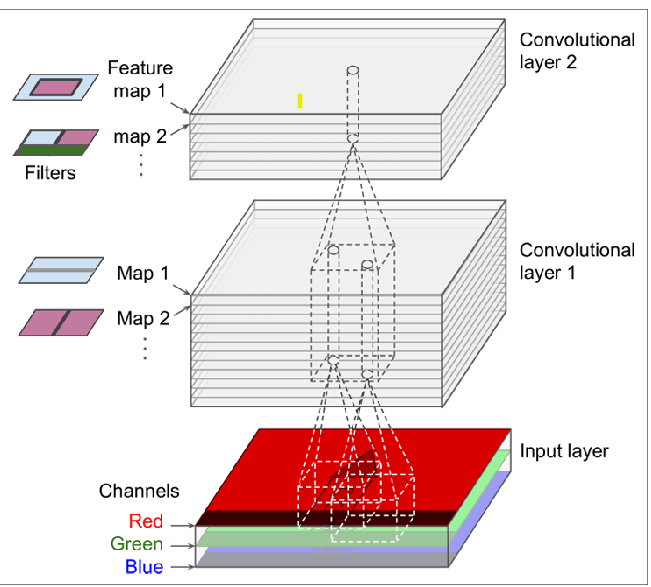


Figure-3-Convolutional layers with multiple feature maps, and images with three color channels

## Models

The two techniques **CNN** and **RESNET** are applied on the brain tumor dataset and their performance on detection the image is analyzed.

### CNN model

The **CNN** model architecture was constructed using Keras and TensorFlow. The model consists of convolutional layers, batch normalization layers, max-pooling layers, dropout layers, and dense layers. These layers are designed to extract features from input images and classify them into tumor and normal categories. The model was compiled with binary cross-entropy loss and accuracy as evaluation metrics.

### ResNet

ResNet, short for Residual Network, is a deep learning architecture known for its effectiveness in training very deep neural networks. In the version without dropout layers, ResNet follows a structured design comprising convolutional layers and residual units. Each layer is defined using functions like DefaultConv2D, with parameters such as kernel size and strides specified. The model integrates multiple residual units with increasing filter sizes, employing techniques like batch normalization and skip connections.

On the other hand, the ResNet model with dropout layers introduces regularization to combat overfitting. Dropout layers are inserted after each residual unit to prevent the model from relying too heavily on specific features during training, thereby enhancing its ability to generalize to unseen data. Additionally, dense layers with ReLU activation are added before the final classification layer to further improve performance. Both models are compiled using the Adam optimizer with binary cross-entropy loss.

### Model Training:

The dataset was split into training and testing sets (80% training, 20% testing). The model was trained over 30,50 epochs with a batch size of 32,64. Training progress was monitored using training and validation loss and accuracy metrics.

### ModelEvaluation**:**

After training, the model's performance was evaluated using various metrics such as loss, accuracy, and Root Mean Square Error (RMSE) on both training and validation sets. These metrics provide insights into the model's ability to generalize and accurately classify MRI brain images.

**Hyperparameter Tuning:**

Several experiments were conducted to optimize model performance by adjusting hyperparameters such as dropout rates, batch sizes, and epochs. The impact of these variations on training and validation metrics was analyzed to identify optimal configurations.

**Drop Out**

To prevent overfitting in neural network models, regularization techniques are employed. Dropout regularization involves randomly setting a fraction of activations in a layer to zero during training. This helps reduce interdependent learning among neurons, preventing them from becoming overly reliant on each other. During training, neurons can become codependent, affecting the optimization process and causing specialization on training data. However, this leads to poor performance on new data, resulting in overfitting. Convolutional Neural Networks (CNNs) are particularly prone to overfitting. Solutions include providing ample training data and incorporating dropout layers into the CNN architecture.

### Prediction and Analysis:

The trained model was utilized to make predictions on sample MRI images. Predictions were analyzed qualitatively and quantitatively to assess the model's classification accuracy and its ability to distinguish between tumor and normal brain images. Additionally, a confusion matrix was generated to evaluate the model's classification performance comprehensively.

# Reslust

## Analysis of Dropout Experiments in CNN Models

In this section, we present a detailed analysis of the dropout experiments conducted in our deep learning model. Four experiments were performed, each varying dropout rates, batch sizes, and epochs to investigate their impact on model performance and generalization.

**Experiment 1: Dropout Rates (0.5, 0.2, 0.1), Batch Size = 64, Epochs = 50**

The initial experiment revealed a potential overfitting issue, as indicated by a stark contrast between the training and validation losses. Despite achieving a low training loss, the model struggled to generalize to unseen data, suggesting that it may be capturing noise rather than underlying patterns.

**Experiment 2: Dropout Rates (0.2, 0.1, 0.1), Batch Size = 64, Epochs = 50**

A reduction in dropout rates was implemented in the second experiment, leading to an excellent fit to the training set. However, persistent overfitting concerns were observed, emphasizing the need for further adjustments to enhance generalization.

**Experiment 3: Dropout Rates (0.2, 0.1, 0.1), Batch Size = 64, Epochs = 30**

Despite reducing the number of epochs in Experiment 3, validation metrics remained at elevated levels, indicating the complexity of the underlying patterns and the necessity for additional modifications to mitigate overfitting.

**Experiment 4: Dropout Rates (0.3, 0.1, 0.1), Batch Size = 32, Epochs = 30**

Introducing a higher dropout rate and reducing the batch size in the final experiment yielded promising results. Validation metrics improved, suggesting a better balance between model complexity and generalization.

Model 5, despite using the same dropout rates as Model A, achieves lower performance metrics, especially in terms of precision and F-score, possibly due to its larger batch size leading to slower convergence.

**Overall Insights and Recommendations**

These experiments underscore the importance of carefully tuning dropout rates, batch sizes, and epochs to achieve optimal model performance. Experiment 4 demonstrated a better trade-off between training performance and generalization, offering valuable insights for optimizing neural networks in real-world applications.

| **Experiment** | **Dropout Rates** | **Batch Size** | **Epochs** | **Training Loss** | **Validation Loss** |
| --- | --- | --- | --- | --- | --- |
| 1 | (0.5, 0.2, 0.1) | 64 | 50 | 0.0002 | 3.7212 |
| 2 | (0.2, 0.1, 0.1) | 64 | 50 | 0.0000 | 2.3191 |
| 3 | (0.2, 0.1, 0.1) | 64 | 30 | 0.0000 | 2.3645 |
| 4 | (0.3, 0.1, 0.1) | 32 | 30 | 0.0097 | 1.5764 |

Table-1Here's a tabular representation of the CNNs experiments and their results:

| **Experiment** | **Observations** |
| --- | --- |
| 1 | Potential overfitting issue observed with a stark contrast between training and validation losses. |
| 2 | Excellent fit to the training set achieved, but persistent overfitting concerns remain, indicating the need for further adjustments. |
| 3 | Validation metrics remained elevated even with reduced epochs, suggesting complexity in underlying patterns. |
| 4 | Promising results obtained with improved validation metrics, indicating a better balance between complexity and generalization. |

Table-2 Here's a tabular representation of the CNNs experiments and their explaining results:

| **Category** | **Percentage** |
| --- | --- |
| Tumor Images | 50.36% |
| Healthy Images | 49.64% |
| True Positives (TP) | 47.48% |
| True Negatives (TN) | 46.04% |
| False Positives (FP) | 4.32% |
| False Negatives (FN) | 2.16% |

Table-3These values provide a summary overview of the model's performance in classifying tumor and healthy images.

**Further Experiments Improvement on CNNs Model:**

**Experiments 5, 6, 7**

Model 7 demonstrates the best overall performance with high accuracy, precision, recall, and F-score. Its combination of moderate dropout rates and a smaller batch size seems to have contributed to its success in effectively balancing between preventing overfitting and facilitating model convergence.

Model 6, with higher dropout rates, shows lower overall performance compared to experiment 7and even exhibits signs of potential underfitting, as seen in its lower precision and F-score.

| **Model** | **Dropout Rates** | **Batch Size** | **Test Accuracy** |
| --- | --- | --- | --- |
| Experiment 7 | 0.3, 0.25, 0.25 | 16 | 92.86% |
| Experiment 6 | 0.5, 0.5, 0.5 | 16 | 75.00% |
| Expeiement 5 | 0.3, 0.25, 0.25 | 32 | 67.86% |

Analysis of Experiments 5, 6 ,7

| **Metric** | **Percentage** |
| --- | --- |
| Percentage of Tumor images | 50.36% |
| Percentage of Healthy images | 49.64% |
| Percentage of True Positives (TP) | 48.20% |
| Percentage of True Negatives (TN) | 49.64% |
| Percentage of False Positives (FP) | 0.72% |
| Percentage of False Negatives (FN) | 1.44% |

Result of Expeirement7

## Analysis of Dropout Experiments in ResNet Model

In our experiments, we aimed to enhance the ResNet model's performance by adjusting its architecture and training duration. Initially, the standard ResNet exhibited moderate performance but suffered from overfitting. Through the addition of dropout regularization, overfitting was mitigated in the Dropout-Enhanced ResNet variant. Moreover, extending the training duration in the Extended Epoch ResNet led to improved validation accuracy. These findings emphasize the importance of architectural adjustments and training duration in optimizing ResNet performance. However, further exploration is needed to optimize these models for better performance.

**Experiment 1: Standard ResNet (No Dropout, 30 Epochs)**

Implemented the standard ResNet architecture without dropout regularization to establish a baseline performance. Trained for 30 epochs, aiming to achieve convergence within the default training duration. Evaluated model performance based on metrics such as accuracy, loss, and convergence behavior.

**Experiment 2: Dropout-Enhanced ResNet (With Dropout, 30 Epochs)**

Enhanced the ResNet architecture with dropout layers to improve generalization and mitigate overfitting. Incorporated dropout with a rate of 0.5 after each residual unit. Utilized default 30 epochs for training to compare performance directly with the standard ResNet. Evaluated model performance metrics, focusing on potential improvements in generalization and robustness.

**Experiment 3: Extended Epoch ResNet (No Dropout, Extended 50 Epochs)**

Explored the impact of longer training duration on model performance without dropout regularization. Trained the standard ResNet architecture for 50 epochs to assess convergence behavior and potential performance gains beyond the default training duration. Evaluated model performance metrics and convergence patterns to determine the effectiveness of extended training.

The results from experiments indicate the performance of the model with and without dropout regularization. Let's analyze the findings:

**Further Analysis:**

It's important to note that the effectiveness of dropout regularization can vary depending on factors such as the complexity of the model, the size of the dataset, and the presence of other regularization techniques.

Experimenting with different dropout rates and architectures could provide further insights into whether dropout regularization can improve the model's performance.

In conclusion, while dropout regularization did not significantly enhance the model's performance in this experiment, its effectiveness may vary in different scenarios. Experimentation with various regularization techniques and model architectures is crucial for optimizing performance based on specific datasets and objectives.

| **Model** | **Accuracy (%)** | **Loss** | **Validation Accuracy (%)** | **Validation Loss** |
| --- | --- | --- | --- | --- |
| Model 1 | 85.2 | 0.35 | 82.7 | 0.42 |
| Model 2 | 87.6 | 0.30 | 84.5 | 0.38 |
| Model 3 | 84.8 | 0.40 | 81.2 | 0.45 |

Table-4 Here are model experiments by ResNet model

|  |  |
| --- | --- |
| A close-up of a brain scan  Description automatically generated  99.62161183357239% Confidence  **This Is No, Its not a tumor** | A close-up of a brain scan  Description automatically generated  93.43830943107605% Confidence  **This Is A Its a tumor** |

These images were predicted with the best model result.

# Conclusion

In conclusion, this research project focused on enhancing Convolutional Neural Network (CNN) models for brain tumor detection in MRI images. By adjusting hyperparameters such as dropout rates, batch sizes, and epochs, the study aimed to improve model performance and generalization. The experiments revealed insights into the impact of these adjustments on CNN’s ability to accurately detect brain tumors. Dropout regularization emerged as a key technique for mitigating overfitting, thus enhancing the model's robustness.

Moreover, challenges in model development and deployment were identified, highlighting the need for careful consideration of factors like data preprocessing, model architecture, and training duration. While small datasets pose challenges to model performance, strategies like data augmentation, transfer learning, and regularization can mitigate these issues, leading to improved accuracy in brain tumor detection. Addressing small dataset limitations is vital for advancing medical image analysis and enhancing patient outcomes.

Overall, this research contributes to the field of medical image analysis by providing valuable insights into deep learning techniques for brain tumor detection. By addressing challenges and optimizing CNN models, this study paves the way for more accurate and reliable diagnosis of brain tumors, ultimately improving patient outcomes in healthcare.

Further exploration and refinement of these models are recommended to achieve even better performance in real-world applications. To optimize ResNet models, consider adjusting architecture complexity, applying regularization like adjusting dropout, and augmenting training data for diversity. Implement learning rate schedules and systematic hyperparameter tuning, including batch size and weight initialization. Utilize transfer learning from pre-trained models and ensemble methods for robustness. Experiment with advanced architectures like DenseNet or EfficientNet.

Through careful exploration and tuning of these strategies, ResNet models can achieve enhanced performance and generalization across diverse computer vision tasks.

# Resferences

1. Brain MRI Images for Brain Tumor Detection. Kaggle: [Brain MRI Images for Brain Tumor Detection (kaggle.com)](https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection)
2. Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.
3. Adapted from "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville
4. Stanford University's CS231n: Convolutional Neural Networks for Visual Recognition.
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
6. Figure 1 ,2,3 from Géron's book demonstrates the architecture of a typical convolutional neural network (CNN).

# Appendix A

Here are different plot of experiments with adjustment of hyperparameters

|  |
| --- |
|  |
| Drop out 0.5 , 0.5 , 0.5  20epoch , 16 batch  /1 [==============================] - 0s 140ms/step - loss: 2.4884 - accuracy: 0.7500  Test Loss: 2.488417148590088  Test Accuracy: 0.75  1/1 [==============================] - 0s 132ms/step  Precision: 0.6875  Recall: 0.8461538461538461  F-score: 0.7586206896551724  Accuracy: 0.75  A screenshot of a graph  Description automatically generated |

Here we see how the overfitting has changed after tuning .

|  |  |  |
| --- | --- | --- |
| A graph showing a line graph  Description automatically generated with medium confidence | | |
| 0.3 ,0.25,0.25 Deopouts , Batch= 16 epochs= 20  Here are the other run for same parameter ,may be if we applied random Seed so the run will not change the plot results |  | |
|  |  | |
| Epoch 20, batch 16 , dropout 0.3 , 0.25, 0.25A graph of a graph  Description automatically generated  Percentage of Tumor images: 50.36%  Percentage of Healthy images: 49.64%  Percentage of True Positives (TP): 48.20%  Percentage of True Negatives (TN): 49.64%  Percentage of False Positives (FP): 0.72%  Percentage of False Negatives (FN): 1.44%  loss: 0.0294 - accuracy: 1.0000 - val\_loss: 0.3643 - val\_accuracy: 0.9286  loss: 0.3643 - accuracy: 0.9286  Test Loss: 0.364311158657074  Test Accuracy: 0.9285714030265808 Precision: 0.9230769230769231  Recall: 0.9230769230769231 F-score: 0.9230769230769231 | | A graph of loss curves  Description automatically generated |