BRAIN TUMOR DETECTION USING CNN

Presented By:

G. Geetanath

R. Chethan Sai

N. Sai Teja

Git Hub: Brain-Tumour-Detection

Introduction:

Brain tumours pose a significant health risk, and early detection is crucial for effective treatment and prognosis. Convolutional Neural Networks (CNNs) have shown remarkable performance in various image recognition tasks, including medical image analysis. This project aims to develop a CNN-based system for accurate and efficient detection of brain tumours from MRI images.

Problem Statement:

The primary objective is to design and implement a CNN model capable of accurately detecting brain tumours from MRI scans. Building a detection model using a convolutional neural network in TensorFlow & Keras. The model should address the following specific challenges:

Classification: Differentiating between images with tumours and those without tumours.

Robustness: The model should be robust to variations in image quality, noise, and tumour characteristics.

Data Set Collection:

Data Set is collected from Kaggle. The dataset contains 2 folders: yes and no which contains 253 Brain MRI Images. The folder yes contains 155 Brain MRI Images that are tumorous and the folder no contains 98 Brain MRI Images that are non-tumorous.

Data Augmentation:

Since this is a small dataset, there was not enough examples to train the neural network. Also, data augmentation was useful in tackling the data imbalance issue in the data.

Before data augmentation, the dataset consisted of:

155 positive and 98 negative examples, resulting in 253 example images.

After data augmentation, now the dataset consists of:

1085 positive and 980 examples, resulting in 2065 example images.

Data Preprocessing:

For every image, the following preprocessing steps were applied:

Crop the part of the image that contains only the brain (which is the most important part of the image). Resize the image to have a shape of (240, 240, 3) = (image_width, image_height, number of channels):

Because images in the dataset come in different sizes. So, all images should have the same shape to feed it as an input to the neural network.

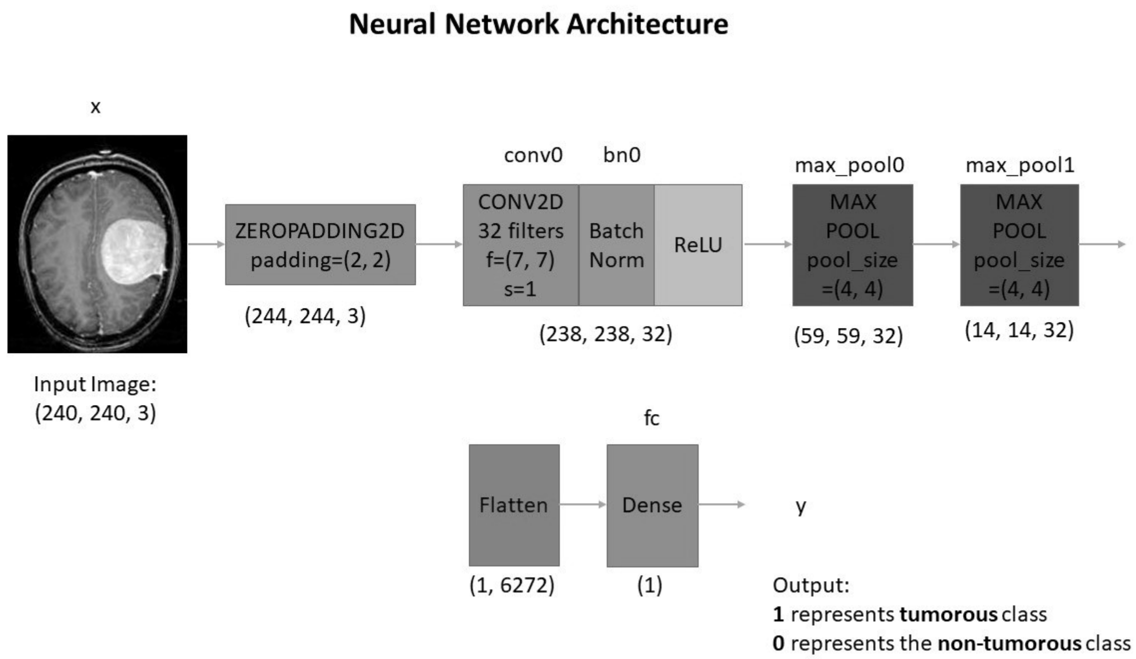
Apply normalization: to scale pixel values to the range 0-1.

Data Split:

The data was split in the following way:

1. 70% of the data for training.
2. 15% of the data for validation.
3. 15% of the data for testing.

Flow Diagram:



Understanding the architecture:

Each input x (image) has a shape of (240, 240, 3) and is fed into the neural network. And, it goes through the following layers:

- A Zero Padding layer with a pool size of (2, 2).

- A convolutional layer with 32 filters, with a filter size of (7, 7) and a stride equal to 1.
- A batch normalization layer to normalize pixel values to speed up computation.
- A ReLU activation layer.
- A Max Pooling layer with f=4 and s=4.
- A Max Pooling layer with f=4 and s=4, same as before.
- A flatten layer to flatten the 3-dimensional matrix into a one-dimensional vector.
- A Dense (output unit) fully connected layer with one neuron with a sigmoid activation (since this is a binary classification task).

The performance of CNN-based brain tumour detection model is evaluated using various metrics, including accuracy, Loss and F1 Score

- **Accuracy**:

In classification tasks, accuracy measures the proportion of correctly classified instances out of the total instances in a dataset. It is calculated as the ratio of correct predictions to the total number of predictions made by a model. High accuracy indicates a higher level of correctness in the model's predictions.

- **Loss**:

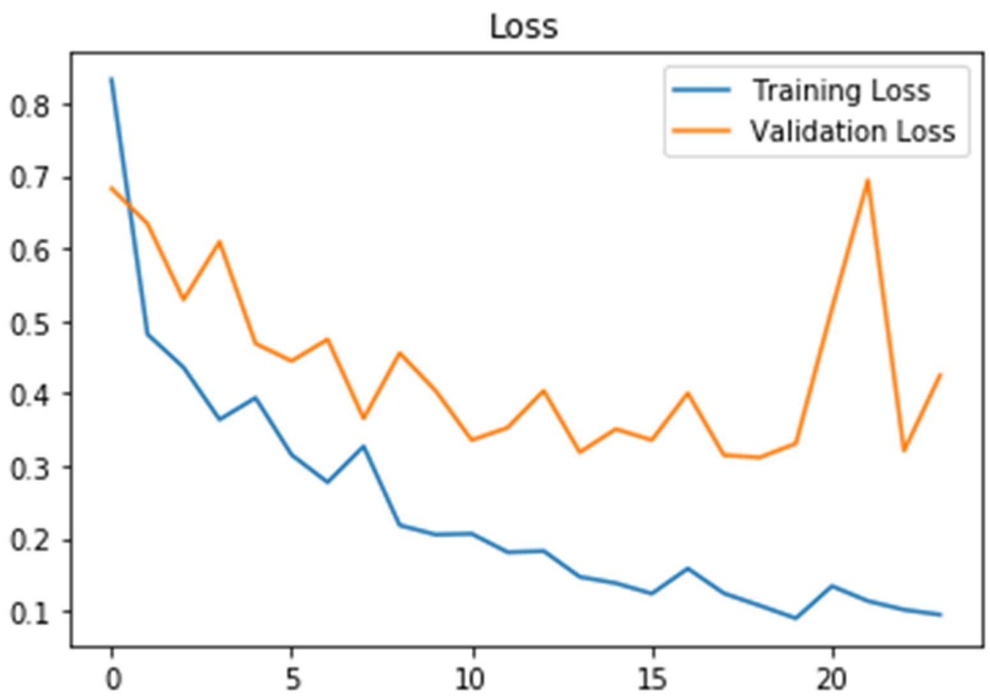
Loss, also known as cost or error, quantifies the disparity between the predicted outputs of a model and the ground truth labels. It serves as a measure of how well the model is performing during training. Lower loss values indicate better alignment between predicted and actual values.

- **F1 Score:**

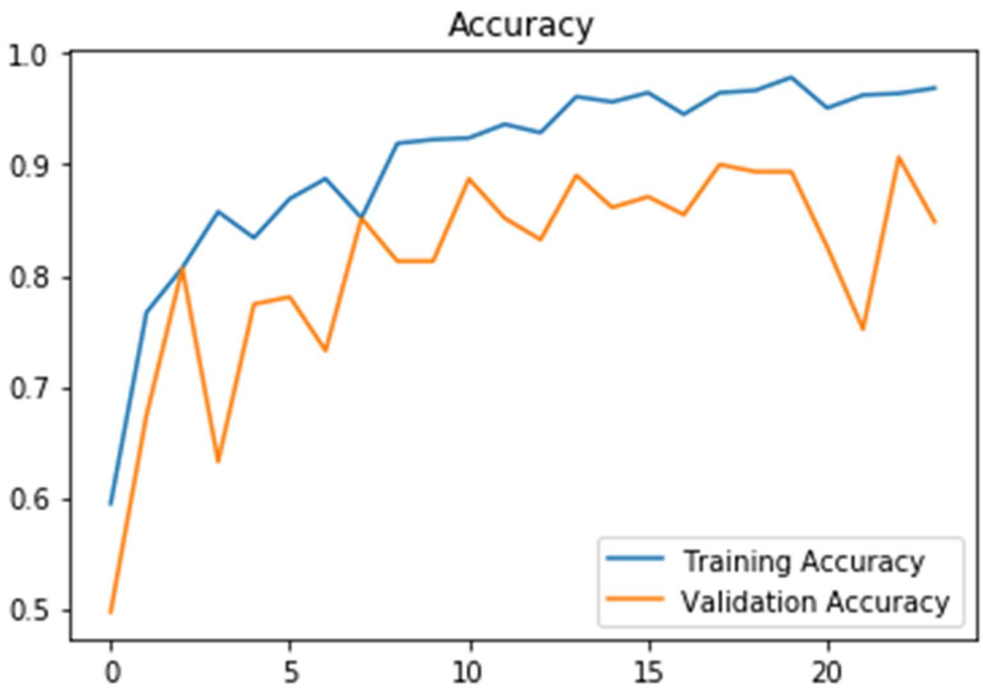
The F1 score is a metric commonly used in binary classification tasks that balances the trade-off between precision and recall. It is calculated as the harmonic mean of precision and recall, providing a single value that represents the model's performance. The F1 score ranges from 0 to 1, where higher values indicate better model performance in terms of both precision and recall.

Results:

Loss:



Accuracy:



The F1 score for the best model on the Testing data and Validation respectively:

```
F1 score: 0.9074074074074074
```

```
F1 score: 0.9074074074074074
```