

BRAIN TUMOUR DETECTION USING A CNN

COMS4030A/COMS7047A- ADVANCED COMPUTATION AND MACHINE LEARNING



Tumi Jourdan ~ 2180153

Luca von Mayer ~ 2427051

Mohammad Zaid Moonsamy ~ 2433079

Shakeel Malagas ~ 2424161

17/05/2024

School of Computer Science and Applied Mathematics

TABLE OF CONTENTS

TABLE OF CONTENTS.....	1
INTRODUCTION.....	2
PROBLEM STATEMENT.....	3
DATASET.....	4
Dataset Contents and Objectives.....	4
Data Preprocessing.....	5
Data Augmentation.....	6
Data splitting.....	7
THE MACHINE LEARNING MODEL - CNN.....	7
Choice of Model and Machine Learning Algorithm.....	7
Breakdown of the Model Architecture.....	8
Conv2Ds.....	8
Max pooling.....	8
Flatten Layer.....	9
Dense.....	9
Dropout.....	9
Architecture Tuning.....	10
EXPERIMENTS.....	11
Optimizer Selection and Hyperparameter Configuration.....	11
Epochs, Batch size and Dropout Rate Configurations.....	13
Epochs.....	13
Batch Size.....	15
Dropout Rate.....	15
TESTING & RESULTS.....	15
Preliminary Assessment.....	16
k-Fold Cross Validation.....	17
DISCUSSION OF RESULTS.....	18
CONCLUSION.....	20
REFERENCES.....	21

INTRODUCTION

The rise of machine learning, especially with Convolutional Neural Networks (CNNs), has completely transformed how medical images are analysed. CNNs are particularly noteworthy for their potential in aiding the diagnosis of various medical conditions, including brain tumours. These tumours encompass a wide range of complexities, each presenting unique challenges in both diagnosis and treatment. Accurate classification is crucial for guiding treatment decisions and predicting patient outcomes. Presently, classification heavily relies on manual interpretation by experienced radiologists, which is time-consuming and subject to errors.

Brain tumours are varied, each with unique challenges for doctors trying to diagnose and treat them. In this study, our aim is to harness the capabilities of CNNs to develop a robust and efficient model for automatically classifying three specific types of brain tumours: gliomas, meningiomas, and pituitary tumours. By training the CNN on a comprehensive dataset of high-resolution brain MRI images, our objective is to achieve a classification accuracy on par with or exceeding that of human experts. This automated approach not only promises to streamline the diagnostic process but also holds the potential to uncover subtle patterns and correlations that may elude human perception.

Through this endeavour, we aspire to contribute to the ongoing efforts in medical imaging research and pave the way for the integration of advanced machine learning techniques into routine clinical practice. By combining the power of CNNs with the wealth of information embedded in medical images, we aim to develop a tool with the potential to empower healthcare professionals and enhance the quality of care for patients with brain tumours.

PROBLEM STATEMENT

The problem addressed in this study revolves around the accurate classification of brain tumours into three distinct categories: gliomas, meningiomas, and pituitary tumours. Given a dataset comprising high-resolution MRI images of these tumours, the objective is to develop a Convolutional Neural Network (CNN) model capable of automatically classifying each tumour type with high accuracy.

The three types of brain tumours - gliomas, meningiomas, and pituitary tumours - exhibit distinct visual characteristics on MRI scans, primarily due to their different locations within the brain. Meningiomas develop in the meninges, the protective membranes surrounding the brain and spinal cord. Gliomas originate from glial cells, which are non-neuronal cells that provide support and insulation to the brain's neurons. Pituitary tumours, on the other hand, grow specifically in the region of the pituitary gland, a small endocrine gland located at the base of the brain. These locational differences manifest as unique visual patterns on MRI scans, enabling radiologists to differentiate between the tumour types. By leveraging these visual distinctions, we aim to develop a CNN model capable of automatically classifying brain tumours based on their MRI appearance, thereby assisting in the accurate diagnosis and treatment planning for patients with these conditions.

Problem: To design a CNN model that can effectively distinguish between gliomas, meningiomas, and pituitary tumours when provided with MRI images, thereby aiding in the automated diagnosis of these brain tumours.

Brain tumours pose significant challenges in clinical diagnosis due to their varied appearance on medical imaging scans and the potential for misinterpretation. Traditional methods of tumour classification rely on manual assessment by experienced radiologists, which is time-consuming and subject to human error. By employing machine learning techniques, specifically CNNs, we aim to streamline this process and provide clinicians with a reliable tool for rapid and accurate tumour classification [Amin, J. et al] .

The successful development of a CNN model for brain tumour classification holds immense potential in clinical practice, offering benefits such as improved diagnostic accuracy, reduced reliance on subjective interpretation, and expedited treatment planning.

DATASET

Dataset Contents and Objectives

The dataset was obtained from [kaggle.com](#). The dataset did not require any ethics clearance and it can be found [here](#). The dataset utilised in this project, comprises a diverse collection of high-resolution MRI images depicting three distinct types of brain tumours: meningiomas, gliomas, and pituitary tumours. It contains three subfolders, each dedicated to one of the three distinct types of tumours. Within each subfolder, numerous images serve as data. A representative sample from each tumour class is showcased in Figure 1, while the initial data distribution of the image dataset for each tumour is depicted in Figure 2.

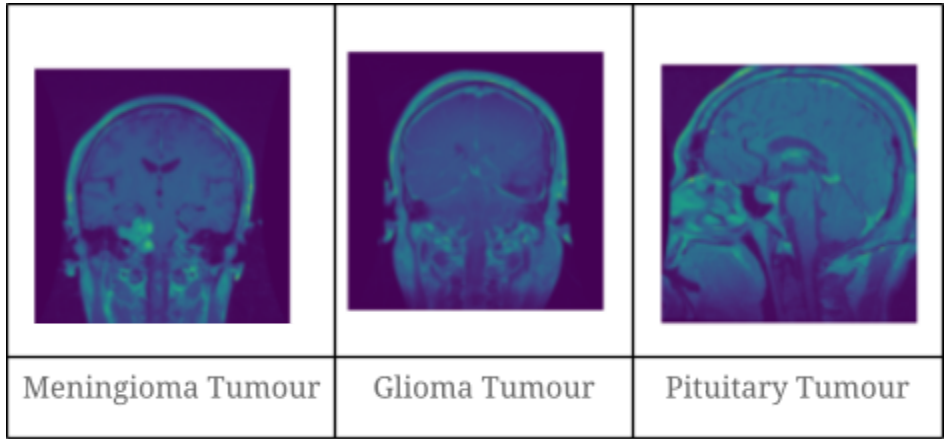


Figure 1

Each image in the dataset encapsulates intricate details and characteristics indicative of the corresponding tumour type, ranging from structural abnormalities to variations in tissue density and morphology. In our study, our main objective is how effective machine learning techniques, especially Convolutional Neural Networks (CNNs), can be in creating a model that can distinguish between different types of brain tumours based on MRI images. We would like to assess the performance of our CNN model on a multi-classification problem using a variety of evaluation metrics.

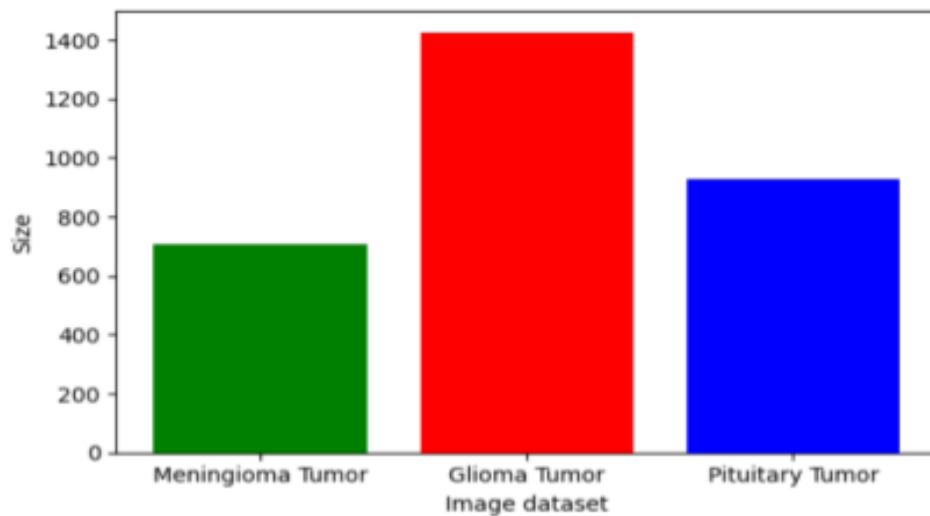


Figure 2

Data Preprocessing

In preparing the dataset for training our Convolutional Neural Network (CNN) model, several preprocessing steps were implemented to ensure the data's suitability for classification tasks. Initially, the dataset, comprising images of three distinct types of brain tumours—meningiomas, gliomas, and pituitary tumours—was organised into separate directories. This organisation facilitated the subsequent processing steps aimed at standardising the images for model training.

Each image underwent a series of preprocessing transformations to enhance its utility for classification. Notably, a cropping function was applied to isolate the brain region from the surrounding background, effectively removing extraneous elements that could potentially introduce noise into the training process. This transformation can be seen in Figure 3. Subsequently, all images were converted into an RGB format and then resized to a standardised resolution of 224x224 pixels, ensuring uniformity in the input data. Finally, pixel values were normalised to a range between 0 and 1 to facilitate convergence during model training.

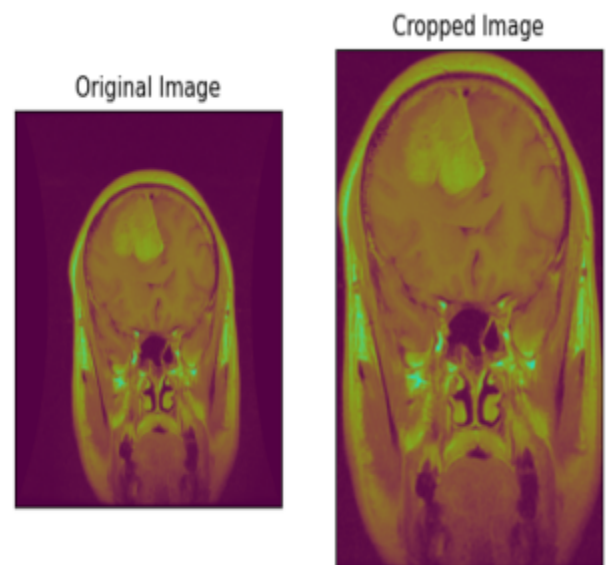
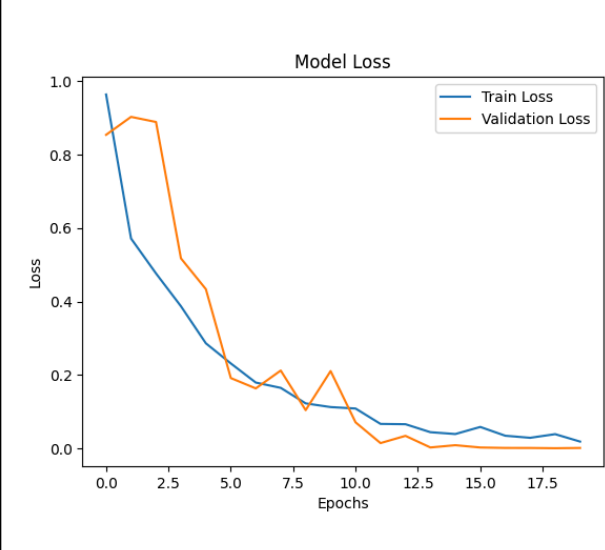
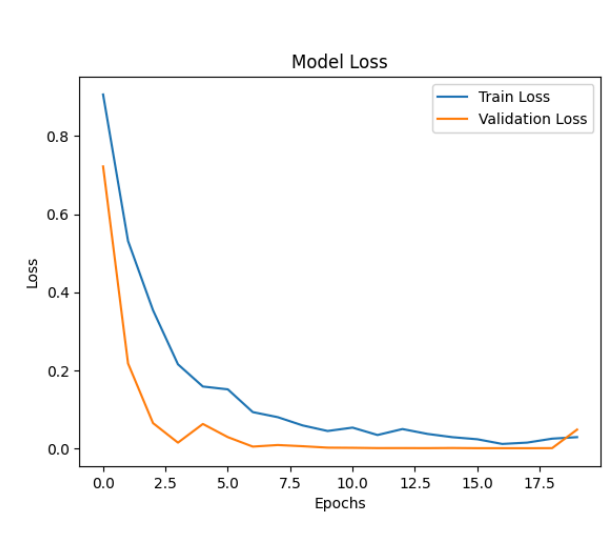


Figure 3

Data Augmentation

Upon visualising the class distribution in the dataset as shown in Figure 2, it is apparent that the tumour classes are imbalanced, with pituitary and meningioma tumours being underrepresented compared to glioma tumours. To address this class imbalance and prevent the model from being biased towards the majority classes, data augmentation techniques were employed.

The augmentation strategy involved oversampling the pituitary and meningioma tumour classes and ensuring all three tumour classes had equal representation in the final augmented dataset. This was achieved by generating additional synthetic samples to balance the existing pituitary and meningioma tumour image datasets. To verify the

	
No Data Augmentation	Data Augmentation
Table 1	

impact of data augmentation, an experiment was conducted by training two versions of the CNN model - one on the original imbalanced dataset and another on the augmented balanced dataset. The model loss during training was tracked for both scenarios, as presented in Table 1. In the case of no data augmentation, the validation loss diverges from the training loss and steadily increases after a few epochs, indicating potential overfitting to the majority classes. On the other hand, with data augmentation, the validation loss more closely follows the training loss and demonstrates better convergence, suggesting improved generalisation to all tumour classes.

These results highlight the effectiveness of data augmentation in mitigating class imbalance and enhancing the model's ability to learn robust features across all tumour types. The augmented dataset provides a more balanced representation, enabling the CNN to make accurate predictions even for the minority pituitary tumour class.

Data splitting

Following preprocessing, the dataset was divided into separate subsets to enable effective model training, validation, and evaluation. Initially, a conventional train-test split was performed, allocating 70% of the data for training, 10% for validation and reserving the remaining 20% for testing. This split ensured that the model was trained on a substantial portion of the data while retaining unseen samples for evaluation, thus assessing its generalisation ability.

Furthermore, to monitor the model's performance during training and fine-tune its parameters effectively, a validation set was created by extracting 10% of the whole dataset. This validation set served as an independent sample for evaluating the model's performance on unseen data during each training epoch, thereby facilitating early stopping and preventing overfitting. Moreover, batching was employed to streamline the training process, with a batch size of 25 chosen for both training and testing to balance computational efficiency and model convergence.

THE MACHINE LEARNING MODEL - CNN

Model Choice

For the task of classifying brain tumours from medical imaging data, a Convolutional Neural Network (CNN) model was selected as the primary machine learning algorithm. CNNs have demonstrated remarkable success in image classification tasks due to their ability to automatically learn hierarchical features directly from raw pixel data. In this study, the decision to employ a CNN was motivated by its capacity to effectively capture spatial patterns and intricate features present in MRI images of brain tumours. By leveraging the hierarchical, multilayer structure of CNNs, the model could discern subtle differences between different tumour types and their location in the brain, enabling accurate classification.

The CNN model was implemented using TensorFlow, a popular deep learning framework, with the Keras API providing a user-friendly interface for defining and

training neural network architectures. The model architecture comprised multiple convolutional layers followed by max-pooling layers to extract and downsample features from input images. Additionally, dense layers were incorporated to enable higher-level feature aggregation and classification. The model was trained using an Adam (Adaptive Moment Estimation) optimizer and optimised for multiclass classification using sparse categorical cross-entropy loss.

Machine Learning Algorithm and Model Architecture

Conv2Ds

This adds a convolutional layer with 32 filters of size 3x3. Convolutional layers are responsible for feature extraction by applying these filters to the input image. We start at 32 filters in the first layer to learn basic features such as edges. It takes the input shape of 224,224,3 as this is the 224x224 image size after preprocessing and 3 for the three colour channels RGB. It has the activation function of a ReLU as this introduces non-linearity to help the model learn more complex patterns. Padding of valid means there is no padding on the image, the output will be smaller than the input.

Further convolution layers use doubled filter numbers (64,128). This is due to the common practice of doubling the number of filters in each layer as this allows the model to learn more complex and abstract features with each layer, this is known as hierarchical feature learning. Some other popular CNN architectures, such as VGGNet and ResNet, also use this practice of progressively increasing the number of filters, which has been proven to be successful. It is also doubled to mitigate the loss that could be caused by reducing the spatial dimensions after the max pooling layers. Finally, doubling the number of filters in each layer allows the model to increase its representational power without an abrupt jump in the number of parameters, which helps in managing the computational resources and training time effectively.

Max pooling

This max pooling layer reduces the spatial dimensions, height and width, by a factor of two which reduces the computational load and ensures the detected features are more robust.

Flatten Layer

We then flatten the data from a 2D feature map into a 1D vector which is necessary before we can input this data into the fully connected dense layers in the next step.

Dense

This is the fully connected layer of 256 neurons. Its purpose is to combine the features learned by the convolutional layers to form more complex representations. It uses 256 neurons as this is double the last number of filters used, 128, as we had a final max pooling layer after that.

The final layer is also a dense layer with 3 neurons. One for each tumour type that we are trying to classify. This layer uses the softmax activation function which converts the outputs into probabilities adding up to 1. The model will then predict the output with the highest probability.

Dropout

This is a regularisation technique where 50% of the neurons are randomly dropped during training. This is to try to prevent overfitting by ensuring the model does not rely too heavily on particular neurons.

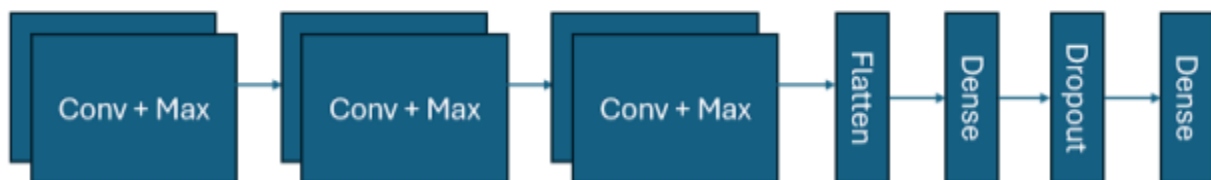


Figure 4: Final Architecture

Architecture Tuning

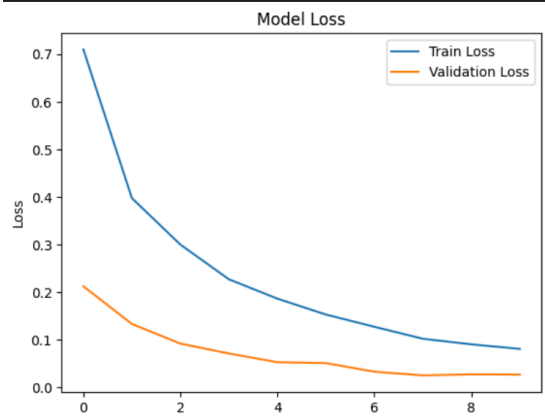
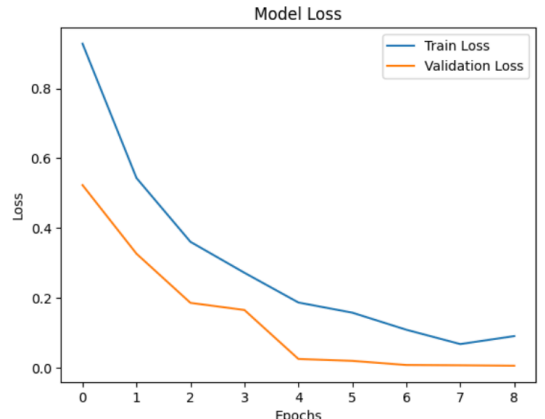
The above architecture is the final base model on which tuning was performed. This model uses Adam optimization. The performance of this model is an accuracy of 95%~96% with a learning time of ~1:00 min per epoch.

In research we found the VGG-16 model developed in the [paper](#) by Simonyan, K., & Zisserman, A. (2015). It is well known for its effectiveness and simplicity. Below is an image showing the different variations of the model:

ConvNet Configuration					
A	A-LRN	B	C	<u>D</u>	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 5: Adapted from Simonyan, K., & Zisserman, A. (2015)

The VGG16 model is a good baseline to compare our model to. Importing the trained body of the VGG16 from tensor flow and training the last layers on our personal data set, it achieved an accuracy of 95% in 10 epochs with a training time of 4 minutes per epoch. The only visible improvement is that the validation loss is less varied in the first 4 epochs.

	
VGG-16	Our Model
<i>Table 2</i>	

Adapting our network to be more akin to the VGG-16 network we added more layers. This of course bloated computation times , but also provided no further increase to accuracy. Considering that the accuracies were so similar, and yet our model trained in half the time, these optimizations became redundant.

EXPERIMENTS

Optimizer Selection and Hyperparameter Configuration

In the training process of the CNN model, an optimizer plays a crucial role in updating the model's parameters based on the computed gradients to minimise the loss function. For this project, we utilised the **Adam** (Adaptive Moment Estimation) optimizer, which is a popular choice for training deep learning models due to its adaptive learning rate and efficient convergence properties.

Adam is an extension of stochastic gradient descent (SGD) that combines the benefits of two other optimization algorithms: AdaGrad and RMSprop. It adapts the learning rate for each parameter individually based on the estimates of the first and second moments of the gradients. The first moment (mean) and second moment (uncentered variance) of the gradients are computed as exponentially decaying averages, which allows Adam to adapt to the changing landscape of the loss function effectively.

The update rule for the parameters in Adam is given by:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_t = \theta_{t-1} - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

where:

- m_t and v_t are the first and second moment estimates at time step t .
- β_1 and β_2 are hyperparameters that control the exponential decay rates for the moment estimates.
- g_t is the gradient at time step t .
- α is the learning rate.
- ϵ is a small value added for numerical stability.
- θ_t represents the updated parameters at time step t .

In our implementation, we used the following hyperparameters for the Adam optimizer:

Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.99)

The hyperparameters have the following meanings:

- **learning_rate (alpha):** It determines the step size at which the parameters are updated in the direction of the negative gradient.
- **beta_1:** It is the exponential decay rate for the first moment estimates. A value of 0.9 is a typical choice, which means that the past gradients have a high influence on the current update.
- **beta_2:** It is the exponential decay rate for the second moment estimates. A value of 0.99 is commonly used, indicating that the past squared gradients have a

significant impact on the current update.

By using the Adam optimizer with these hyperparameters, our model benefits from adaptive learning rates for each parameter, which helps in navigating complex loss landscapes and achieving faster convergence. The adaptive nature of Adam allows it to handle sparse gradients and noisy data effectively, making it a robust choice for training our CNN model on the brain tumour dataset.

It's important to note that while Adam is a popular and effective optimizer, the choice of hyperparameters may require tuning based on the specific problem and dataset. Experimenting with different learning rates and decay rates can help in finding the optimal configuration for achieving the best performance of the model.

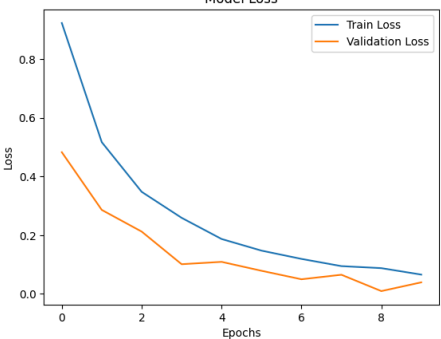
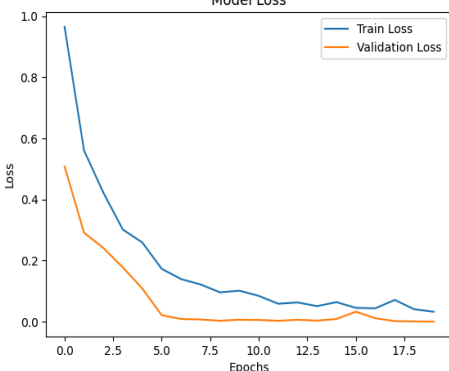
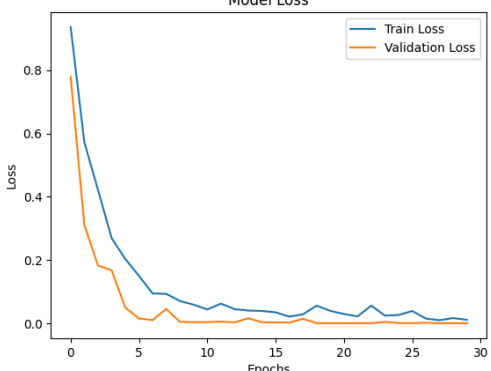
Epochs, Batch size and Dropout Rate Configurations

Epochs

During the training process, the model's accuracy and loss were closely monitored across multiple epochs for both the training and validation sets to evaluate convergence and generalisation.

As seen in Figure 6, after 10 epochs of training, the model exhibited promising performance. It achieved a training accuracy of approximately 93% and a validation accuracy of around 89%. Concurrently, the training loss decreased to about 0.2, while the validation loss settled at roughly 0.4. These results indicated potential for further improvement, as the accuracy curves had not reached a plateau, and the loss curves were still declining. Continuing the training for 20 epochs resulted in notable enhancements in the model's performance. The training accuracy surged to nearly 98%, with the validation accuracy climbing to approximately 93%. Moreover, the training loss decreased to approximately 0.1, and the validation loss stabilised around 0.2.

These observations suggested improved convergence, as the accuracy curves began to level off, and the loss curves showed signs of stabilisation. Further extending the training to 30 epochs yielded the most favourable outcomes. The model achieved an outstanding training accuracy of 99%, accompanied by a validation accuracy of around 96%. Additionally, the training loss decreased to approximately 0.05, while the validation loss dropped to roughly 0.2. This demonstrated exceptional convergence, with the accuracy curves flattening, and the loss curves maintaining stability.

		
10 Epochs	20 Epochs	30 Epochs
Table 3		

Analysing the results across different epoch settings, shown in Table 3, revealed a clear trend: increasing the number of epochs led to enhanced model performance. Both training and validation accuracies consistently improved, indicating effective learning and generalisation. However, it was crucial to avoid overfitting by monitoring the gap between training and validation accuracy. In this case, the model trained for 30 epochs struck the optimal balance between performance and generalisation, with high validation accuracy and stable loss curves.

We also experimented utilising an early stopping criteria based on the delta of the training loss. The stopping condition was such that after the delta drops below a tolerance of 0.001 and there is no change for 3 consecutive epochs, the training stops. It was observed that the model trained for 11 epochs before the aforementioned stopping criteria.

Based on these experiments, the optimal hyperparameter choice for the CNN model involves training for 30 epochs. This setting allows the model to learn meaningful features from the data while maintaining robust generalisation to unseen samples.

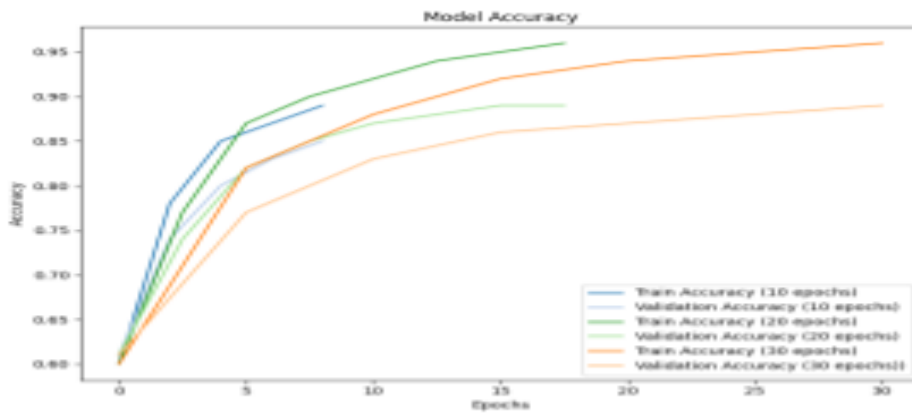


Figure 6

Batch Size

The batch size, which refers to the number of training inputs utilised in one iteration. It significantly influences the training dynamics and computational efficiency of the model. In this study, a batch size of 25 was chosen after initial experimentation and analysis. This choice struck a balance between computational efficiency and model convergence. A smaller batch size would provide a noisier estimate of the gradient but may have led to faster convergence due to more frequent updates. Conversely, a larger batch size would offer a more stable gradient estimate but may have required more memory and computational resources.

Dropout Rate

In this study, a dropout rate of 0.5 was selected based on empirical evidence and experimentation. This moderate dropout rate helped prevent overfitting while allowing the model to retain sufficient capacity to learn complex patterns from the data. A higher dropout rate could have led to more aggressive regularisation, which may be beneficial in scenarios with limited training data or high model complexity. Conversely, a lower dropout rate may have allowed the model to learn more nuanced features but risks overfitting, especially with larger datasets.

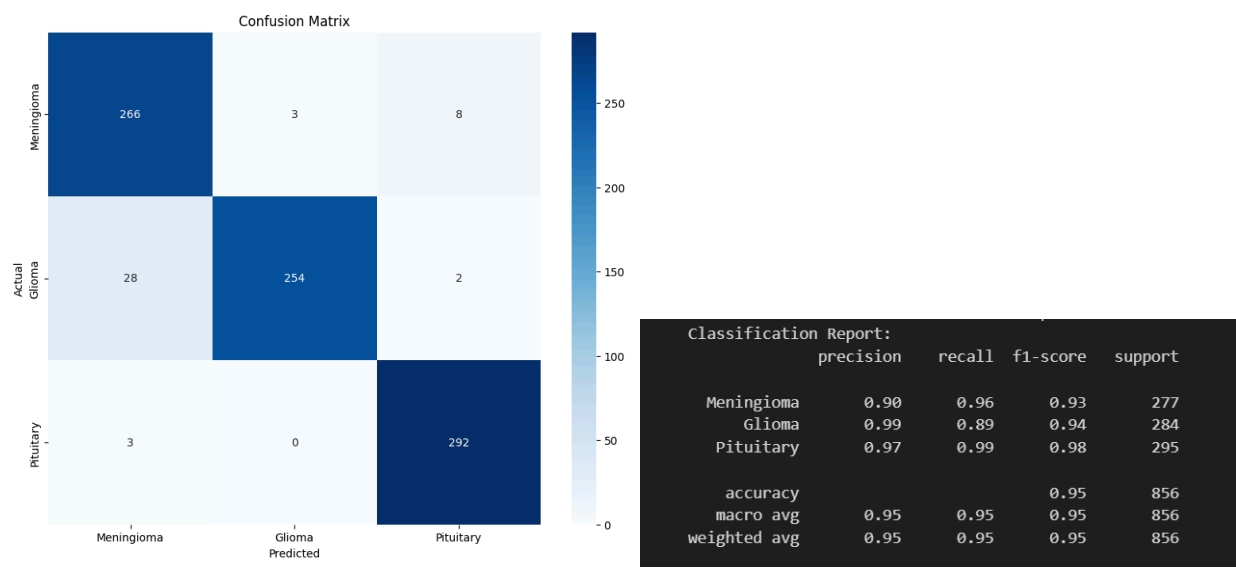
To conclude, the choice of a batch size of 25 and a dropout rate of 0.5 for the CNN model was informed by a balance between computational efficiency, model capacity, and regularisation requirements. These hyperparameters were selected and fine-tuned through iterative experimentation to achieve optimal performance and generalisation capabilities.

TESTING & RESULTS

In the testing phase of this project, two distinct methodologies were employed to comprehensively evaluate the model's performance. Initially, the dataset was partitioned into two subsets: a training dataset and a test dataset, with 70% of the data allocated for training, 10% for validation and the remaining 20% reserved for testing purposes. This initial testing approach provided an initial assessment of the model's performance on unseen data. However, given the moderate size of the dataset, a more robust evaluation method was deemed necessary. Therefore, K-fold cross-validation was implemented to assess the model's accuracy and other significant metrics.

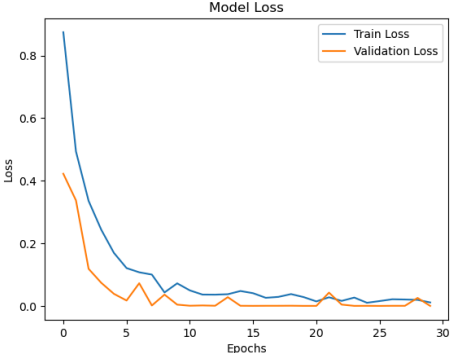
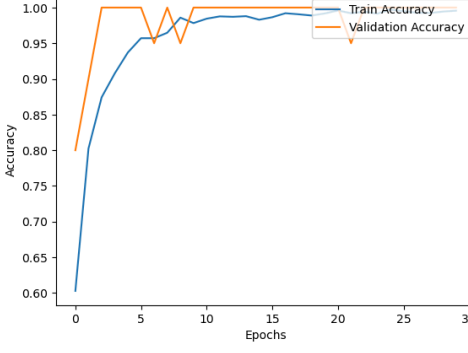
Preliminary Assessment

In the preliminary assessment, the model's performance was evaluated using the held-out test set, which comprised 20% of the original dataset. The classification report provided insights into the model's performance for each tumour type. The precision for meningioma, glioma, and pituitary tumours was 0.90, 0.99, and 0.97, respectively. The recall was 0.96 for meningioma, 0.89 for glioma, and 0.99 for pituitary tumours. The F1 score, which is the harmonic mean of precision and recall, was 0.93 for meningioma, 0.94 for glioma, and 0.98 for pituitary tumours. The support column indicates the number of instances of each tumour type in the test set.



The confusion matrix for the preliminary assessment showed the number of true positive, true negative, false positive, and false negative predictions for each tumour type. The model correctly predicted 266 instances of meningioma, 254 instances of glioma, and 292 instances of pituitary tumours. There were relatively few misclassifications, with 3 meningioma tumours misclassified as glioma, 8 glioma tumours misclassified as meningioma or pituitary, and 3 pituitary tumours misclassified as glioma. These results suggest that the model performed well in the preliminary assessment, achieving high precision, recall, and F1 scores for each tumour type.

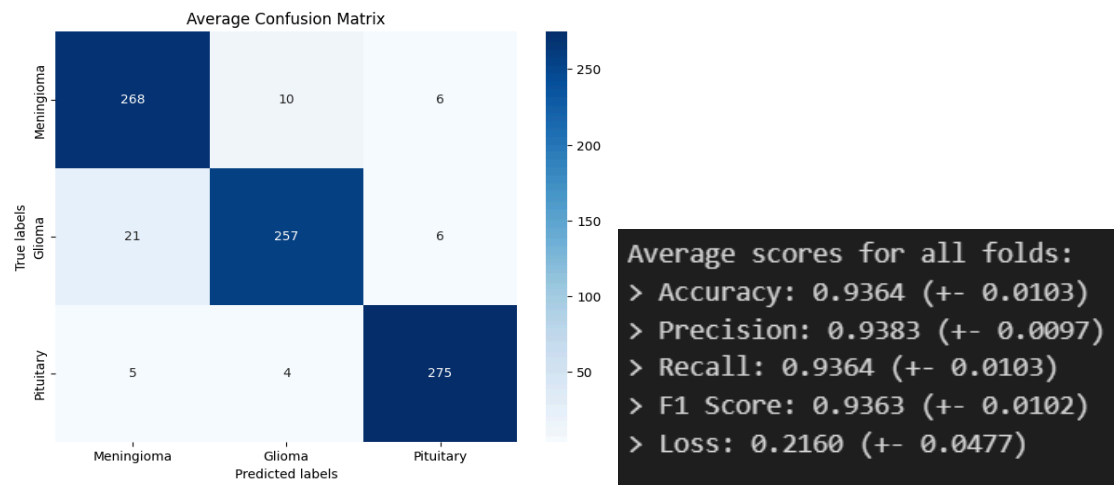
The following model loss and accuracy graphs were also obtained as part of the evaluation results, as can be seen in Table 4.

	
Model Loss	Model Accuracy
Table 4	

k-Fold Cross Validation

To obtain a more robust evaluation of the model's performance, k-fold cross-validation was employed. The dataset was split into multiple folds, and the model was trained and tested on different combinations of these folds. This approach helps to mitigate potential bias and provides a more reliable estimate of the model's generalisation capabilities.

The results of the k-fold cross-validation are summarised in the "Average scores for all folds" section of the output. The model achieved an average accuracy of 0.9364 (+- 0.0103) across all folds, indicating that it correctly predicted the tumour type in approximately 93.64% of the cases. The precision was 0.9383 (+- 0.0097), the recall was 0.9364 (+- 0.0103), and the F1 score was 0.9363 (+- 0.0102). These metrics demonstrate that the model maintained high performance across different folds of the dataset.



The "Average Confusion Matrix" provides a detailed breakdown of the model's predictions across all folds. The model correctly predicted 268 instances of meningioma, 257 instances of glioma, and 275 instances of pituitary tumours. There were relatively few misclassifications, with 10 meningioma tumours misclassified as glioma, 6 glioma tumours misclassified as meningioma or pituitary, and 4 pituitary tumours misclassified as glioma. The results of the k-fold cross-validation further validate the model's effectiveness in classifying brain tumours. The high average accuracy, precision, recall, and F1 scores, along with the low misclassification rates in the confusion matrix, indicate that the model is robust and can generalise well to unseen data.

DISCUSSION OF RESULTS

The developed CNN model achieved a high accuracy of **95%~96%** in classifying brain tumours into three distinct categories: meningioma, glioma, and pituitary. This high level of performance can be attributed to several factors, including the carefully designed architecture of the model and the extensive hyperparameter tuning process.

The model's architecture, consisting of multiple convolutional layers followed by max-pooling and dense layers, proved to be effective in extracting relevant features from the MRI images. The convolutional layers allowed the model to learn hierarchical representations of the tumour patterns, while the max-pooling layers helped in downsampling the feature maps and reducing spatial dimensions. The dense layers at the end of the network enabled the model to learn complex relationships between the extracted features and the tumour classes.

During the training process, the model showed steady improvement in accuracy and a decrease in loss over the epochs which took around a minute to run each epoch. The model demonstrates high precision, recall, and F1-score for all three tumour classes, indicating its ability to correctly classify each type of tumour with minimal misclassifications. The high recall values of 0.96 0.89 and 0.99 for the tumours were observed. In medical diagnosis, particularly in conditions like brain tumours, high recall values are crucial because they indicate the model's capability to detect a vast majority of actual positive cases. A recall value of 0.96, for instance, suggests that the model correctly identified 96% of all actual tumour cases in the dataset. This means that the model is proficient at capturing the presence of tumours, minimising the chances of false negatives, which could lead to critical misdiagnoses or delays in treatment.

The confusion matrix provides further insights into the model's predictions. The matrix shows that the model correctly classified the majority of the samples for each tumour class, with only a few misclassifications between the classes. The accuracy and loss graphs illustrate the model's learning progress over the epochs. The train and validation accuracy curves converge towards higher values, indicating that the model is learning meaningful patterns from the data. The loss curves show a decreasing trend, suggesting that the model is minimising the error between its predictions and the actual labels.

However, achieving these impressive results required significant effort in tuning the hyperparameters of the model. The learning rate, optimiser parameters, and number of

epochs were among the most critical hyperparameters that greatly influenced the model's performance. Through extensive experimentation, we found the optimal combination of these hyperparameters that yielded the best results.

One particularly important hyperparameter was the number of epochs. We observed that training the model for too many epochs led to overfitting, where the model started to memorise the training data instead of generalising well to unseen examples. On the other hand, training for too few epochs resulted in underfitting, where the model failed to capture the underlying patterns in the data. By carefully monitoring the model's performance on the validation set and employing early stopping techniques (choosing to stop at a training accuracy of around 94%), we determined the optimal number of epochs that struck a balance between model complexity and generalisation ability.

In addition to the Adam optimizer, which proved to be highly effective, we also experimented with other optimizers such as Stochastic Gradient Descent (SGD), RMSprop, and Adagrad. While these optimizers yielded comparable results, the Adam optimizer consistently outperformed them, achieving slightly higher accuracy. The adaptive learning rate and momentum-based update rules of Adam allowed the model to converge faster and navigate the complex loss landscape more efficiently.

The k-fold cross-validation results underscore the robustness and generalisation capabilities of the model. By training and testing on different combinations of folds, potential bias is mitigated, resulting in a more reliable estimation of the model's performance. The model achieved an impressive average accuracy of 93.64%, indicating its ability to correctly predict tumour types in the majority of cases. Similarly high precision, recall, and F1 scores further validate its effectiveness across various folds of the dataset. The detailed breakdown provided by the confusion matrix reveals minimal misclassifications, reaffirming the model's reliability. Such high accuracy may be attributed to the model's ability to learn intricate patterns from the data, aided by the convolutional neural network architecture and careful fine-tuning of hyperparameters. Overall, these results demonstrate the model's positivity and its potential to accurately classify brain tumours, thereby offering valuable support to healthcare professionals in diagnosis and treatment decisions.

Finally, it is important to note that our CNN model's impressive performance is indeed influenced by various factors related to the dataset characteristics and thus our model might not obtain such results for different datasets and tasks.

CONCLUSION

In this study, we developed a Convolutional Neural Network (CNN) model for the accurate classification of brain tumours into three distinct categories: meningioma, glioma, and pituitary. By leveraging the power of deep learning and the wealth of information embedded in high-resolution MRI images, our model achieved a remarkable accuracy of 95%~96% in classifying these tumours.

The success of our model can be attributed to several key factors. Firstly, the carefully designed architecture of the CNN, consisting of multiple convolutional layers, max-pooling layers, and dense layers, allowed for effective feature extraction and hierarchical representation learning from the MRI images. Secondly, extensive hyperparameter tuning, including the selection of an appropriate optimizer (Adam), learning rate, and number of epochs, played a crucial role in optimising the model's performance and striking a balance between model complexity, run-time and accuracy.

The impressive results achieved by our model highlight the immense potential of CNNs in the field of medical image analysis, particularly in the context of brain tumour classification. The high accuracy, precision, recall, and F1-score across all three tumour classes demonstrate the model's reliability and its potential real world applications.

In conclusion, our study demonstrates the effectiveness of CNNs in accurately classifying brain tumours based on MRI images. The high performance achieved by our model underscores the immense potential of deep learning in revolutionising medical image analysis and assisting healthcare professionals in making informed decisions in this domain.

REFERENCES

1. Cheng, Jun (2017): brain tumor dataset. figshare. Dataset.
<https://doi.org/10.6084/m9.figshare.1512427.v5>
2. Amin, J., Sharif, M., Yasmin, M. and Fernandes, S.L., 2020. A distinctive approach in brain tumor detection and classification using MRI. *Pattern Recognition Letters*, 139, pp.118-127.
3. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations (ICLR 2015)*, 1–14.