UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG

SCHOOL OF COMPUTER SCIENCE AND APPLIED MATHEMATICS

COURSE TITLE: ADAPTIVE COMPUTATION AND MACHINE LEARNING

Project

Submitted By:	
Name	**Student Number**
Sebola ML	2929881
Nelufhangani N	2935736
Mpe P	2914238

Contents

List of Figures

1 Introduction

A brain tumor is an abnormal growth of cells in or around the brain (Zülch, 2013). These tumors can be benign (non-cancerous) or malignant (cancerous). They can originate from the brain tissue itself (primary brain tumors) or spread to the brain from other parts of the body (metastatic or secondary brain tumors). They can arise from the brain tissue itself or spread from other parts of the body (metastatic tumors) (Pichaivel et al., 2022). The impact of a brain tumor on an individual varies significantly depending on the tumor type, location, size, and growth rate. Symptoms often include headaches, seizures, etc., making early diagnosis and treatment crucial. Advances in medical technology and treatment approaches have improved the prognosis for many patients, but brain tumors remain a complex and serious health challenge (Bai et al., 2024). Detecting and classifying brain tumors using a Convolutional Neural Network (CNN) is an advanced application of deep learning in the medical field (Waghmare and Kolekar, 2021).

2 Dataset description and Objectives

The dataset used is a comprehensive collection of MRI (Magnetic Resonance Images) scans of 926 glioma tumors, 937 meningioma tumors, 901 pituitary tumors, and 396 images showing no tumor. The dataset was obtained from Kaggle-`https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri`.

- Glioma tumors are tumors that originate in the glial cells of the brain or spine (HOSOI, 1930).
- Meningiomas are tumors that arise from the meninges, which are the protective membranes surrounding the brain and spinal cord (Alzahrani, 2023).
- Pituitary tumors develop in the pituitary gland, which is a small gland located at the base of the brain (Alzahrani, 2023).
- No tumor is a category that refers to cases where tumors are absent in the brain.

2.1 Objectives

The objectives of the project are to:

1. perform data augmentation to improve the performance and reliability of your brain tumor classification model.
2. build a convolution neural network model to classify brain tumor images.
3. predict and categorise the brain tumor into one of the four classes.

3 Data Preprocessing

Data processing is a crucial step in preparing brain tumor images for classification using CNN, This section outlines the various preprocessing steps taken to ensure that the dataset is appropriate for training and validating the model.

3.1 Image Resizing

To standardise the input dimensions for the CNN, all brain tumor images were resized to 255×255 pixels. The image resizing ensures that each image has the same size, facilitating consistent input into the CNN and optimising computational efficiency.

3.2 Normalisation

Normalisation was used to scale the pixel values of the image to a range between 0 and 1. This step is essential in making the learning process more efficient and improving numerical stability.

3.3 Data Augmentation

Various data augmentation techniques were employed to improve the model's robustness and generalisation capability. These techniques artificially expand the dataset by creating modified versions of the original images making the dataset complex and large. The augmentation techniques used included:

1. **Horizontal and vertical flipping**: This transformation flips the images along the horizontal and vertical axes, effectively doubling the dataset size and introducing symmetry variations
2. **Random Rotations**: Images were randomly rotated within a specified range to make the model invariant to orientation changes
3. **Reflections**: This involves creating mirror images along the axes, further diversifying the dataset.
4. **Additional Transformations**: Other transformations, such as random cropping, scaling, and brightness adjustments, were also considered to introduce variability and prevent overfitting.

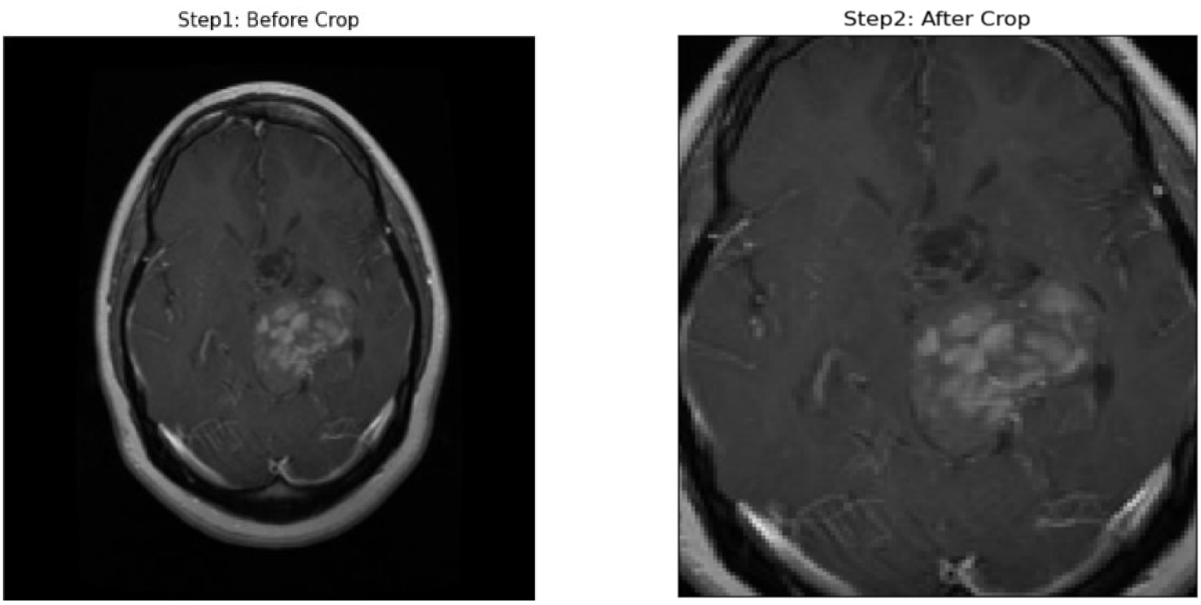

Figure 1: Image cropping.

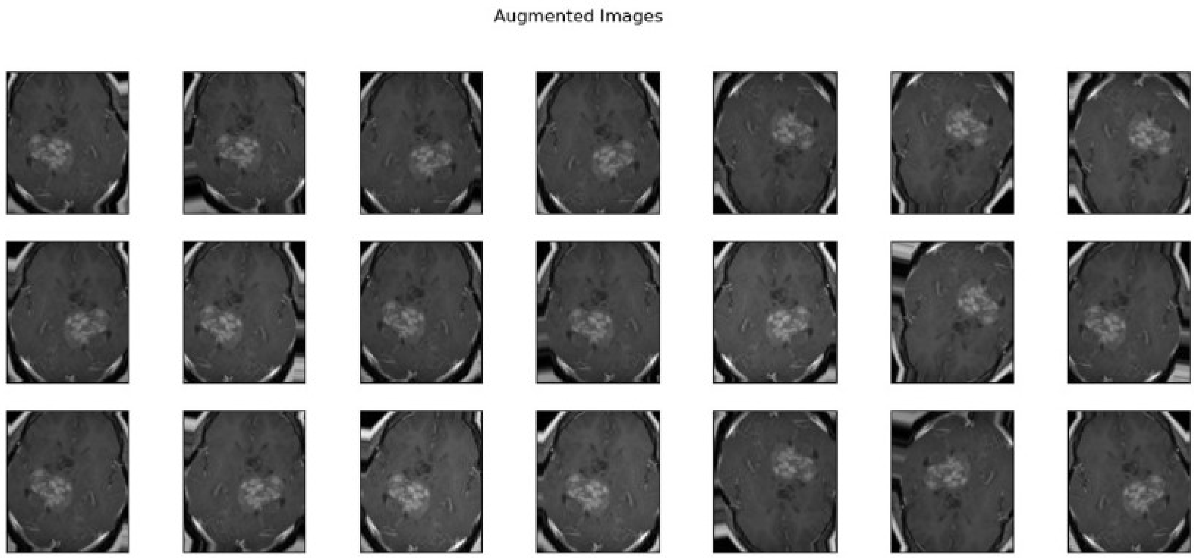

Figure 2: Images after augmentation.

3.4 Dataset Splitting

After augmentation, we had 66 289 images as our dataset. The dataset was split into training, testing, and validation after applying augmentation to the data set. This split ensures the model is trained on one subset of data, validated on another to tune hyperparameters and prevent overfitting, and finally tested to on a separate subset to evaluate its performance on the unseen data. The dataset was split as follows:

- **Training Set**: 80% of the data was allocated for model training which is 52964 images.
- **Validation Set**: 10% of the data was used for validating the model during training which is 6650 images
- **Test Set**: 10% of the data was reserved for the final evaluation of the model's performance which is 6675 images.

4 Model building and machine learning algorithm

The machine learning algorithm chosen in this project is CNN for its effectiveness in image recognition tasks (Wäldchen and Mäder, 2018). CNN is chosen due to its unparalleled ability to automatically extract intricate features from raw pixel data. CNN learn these features directly from the data through layers of convolutional, activation, pooling, and fully connected layers. This process allows CNNs to capture hierarchical representations of images, from basic features like edges to complex patterns like object shapes and textures.

4.1 Model selection

We thought about hyperparameter tuning (changing the number of convolutional layers and layers) while using 1 dense layer and 32 batch size to get the optimal model for this problem. This helps by choosing a better model that performs well based on the data. The number of combinations taken into consideration when selecting the model that best matches our dataset is displayed in the following figures.

```
Epoch 1/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 192s 114ms/step - accuracy: 0.4384 - loss: 1.1906 - val_accuracy: 0.5809 - val_loss: 1.0315
Epoch 2/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 190s 114ms/step - accuracy: 0.6386 - loss: 0.8565 - val_accuracy: 0.6633 - val_loss: 0.8062
Epoch 3/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 191s 115ms/step - accuracy: 0.7072 - loss: 0.7173 - val_accuracy: 0.6872 - val_loss: 0.8049
Epoch 4/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 190s 115ms/step - accuracy: 0.7524 - loss: 0.6233 - val_accuracy: 0.7212 - val_loss: 0.7134
Epoch 5/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 190s 114ms/step - accuracy: 0.7805 - loss: 0.5552 - val_accuracy: 0.7277 - val_loss: 0.6967
Epoch 6/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 191s 115ms/step - accuracy: 0.7964 - loss: 0.5172 - val_accuracy: 0.7203 - val_loss: 0.7188
Epoch 7/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 190s 114ms/step - accuracy: 0.8112 - loss: 0.4866 - val_accuracy: 0.7402 - val_loss: 0.6984
Epoch 8/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 195s 118ms/step - accuracy: 0.8195 - loss: 0.4623 - val_accuracy: 0.7198 - val_loss: 0.7666
Epoch 9/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 191s 115ms/step - accuracy: 0.8327 - loss: 0.4333 - val_accuracy: 0.7223 - val_loss: 0.7842
Epoch 10/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 197s 119ms/step - accuracy: 0.8391 - loss: 0.4148 - val_accuracy: 0.7292 - val_loss: 0.7628
Epoch 11/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 191s 115ms/step - accuracy: 0.8473 - loss: 0.3976 - val_accuracy: 0.7341 - val_loss: 0.7718
Epoch 12/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 188s 113ms/step - accuracy: 0.8526 - loss: 0.3777 - val_accuracy: 0.7266 - val_loss: 0.8424
Epoch 13/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 200s 120ms/step - accuracy: 0.8558 - loss: 0.3721 - val_accuracy: 0.7409 - val_loss: 0.8281
Epoch 14/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 266s 161ms/step - accuracy: 0.8648 - loss: 0.3493 - val_accuracy: 0.7340 - val_loss: 0.8818
Epoch 15/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 188s 113ms/step - accuracy: 0.8717 - loss: 0.3370 - val_accuracy: 0.7257 - val_loss: 0.8790
```

Figure 3: CNN model with 3 convolutional layers and 8 layer size.

```
Epoch 1/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 332s 199ms/step - accuracy: 0.4795 - loss: 1.1192 - val_accuracy: 0.6412 - val_loss: 0.8552
Epoch 2/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 329s 198ms/step - accuracy: 0.6776 - loss: 0.7727 - val_accuracy: 0.6780 - val_loss: 0.7692
Epoch 3/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 328s 198ms/step - accuracy: 0.7390 - loss: 0.6522 - val_accuracy: 0.7098 - val_loss: 0.7272
Epoch 4/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 332s 200ms/step - accuracy: 0.7896 - loss: 0.5458 - val_accuracy: 0.7338 - val_loss: 0.6839
Epoch 5/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 392s 236ms/step - accuracy: 0.8147 - loss: 0.4860 - val_accuracy: 0.7556 - val_loss: 0.6218
Epoch 6/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 331s 200ms/step - accuracy: 0.8442 - loss: 0.4176 - val_accuracy: 0.7472 - val_loss: 0.6797
Epoch 7/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 340s 205ms/step - accuracy: 0.8618 - loss: 0.3714 - val_accuracy: 0.7660 - val_loss: 0.6977
Epoch 8/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 342s 206ms/step - accuracy: 0.8872 - loss: 0.3158 - val_accuracy: 0.7680 - val_loss: 0.7021
Epoch 9/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 332s 200ms/step - accuracy: 0.8980 - loss: 0.2880 - val_accuracy: 0.7773 - val_loss: 0.7016
Epoch 10/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 327s 197ms/step - accuracy: 0.9085 - loss: 0.2583 - val_accuracy: 0.7716 - val_loss: 0.7558
Epoch 11/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 325s 196ms/step - accuracy: 0.9148 - loss: 0.2383 - val_accuracy: 0.7711 - val_loss: 0.7275
Epoch 12/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 329s 198ms/step - accuracy: 0.9239 - loss: 0.2180 - val_accuracy: 0.7689 - val_loss: 0.8252
Epoch 13/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 342s 206ms/step - accuracy: 0.9291 - loss: 0.1989 - val_accuracy: 0.7789 - val_loss: 0.7811
Epoch 14/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 383s 231ms/step - accuracy: 0.9377 - loss: 0.1794 - val_accuracy: 0.7635 - val_loss: 0.9530
Epoch 15/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 331s 200ms/step - accuracy: 0.9431 - loss: 0.1670 - val_accuracy: 0.7629 - val_loss: 0.8527
```

Figure 4: CNN model with 3 convolutional layers and 16 layer size.

```
Epoch 1/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 812s 489ms/step - accuracy: 0.5387 - loss: 1.0280 - val_accuracy: 0.7426 - val_loss: 0.6620
Epoch 2/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 844s 510ms/step - accuracy: 0.7954 - loss: 0.5229 - val_accuracy: 0.7489 - val_loss: 0.6314
Epoch 3/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 846s 510ms/step - accuracy: 0.8777 - loss: 0.3171 - val_accuracy: 0.7913 - val_loss: 0.5819
Epoch 4/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 777s 469ms/step - accuracy: 0.9237 - loss: 0.2068 - val_accuracy: 0.7896 - val_loss: 0.6877
Epoch 5/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 781s 471ms/step - accuracy: 0.9503 - loss: 0.1379 - val_accuracy: 0.7983 - val_loss: 0.7015
Epoch 6/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 774s 467ms/step - accuracy: 0.9653 - loss: 0.0964 - val_accuracy: 0.7905 - val_loss: 0.9611
Epoch 7/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 771s 465ms/step - accuracy: 0.9719 - loss: 0.0787 - val_accuracy: 0.8050 - val_loss: 0.9029
Epoch 8/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 779s 470ms/step - accuracy: 0.9751 - loss: 0.0688 - val_accuracy: 0.7992 - val_loss: 1.0405
Epoch 9/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 777s 469ms/step - accuracy: 0.9796 - loss: 0.0589 - val_accuracy: 0.8036 - val_loss: 1.0359
Epoch 10/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 771s 465ms/step - accuracy: 0.9830 - loss: 0.0487 - val_accuracy: 0.8102 - val_loss: 1.1267
Epoch 11/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 757s 457ms/step - accuracy: 0.9848 - loss: 0.0427 - val_accuracy: 0.8059 - val_loss: 1.2189
Epoch 12/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 756s 456ms/step - accuracy: 0.9831 - loss: 0.0487 - val_accuracy: 0.8090 - val_loss: 1.0650
Epoch 13/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 741s 447ms/step - accuracy: 0.9899 - loss: 0.0314 - val_accuracy: 0.8035 - val_loss: 1.1218
Epoch 14/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 738s 445ms/step - accuracy: 0.9873 - loss: 0.0389 - val_accuracy: 0.8030 - val_loss: 1.2671
Epoch 15/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 738s 445ms/step - accuracy: 0.9898 - loss: 0.0326 - val_accuracy: 0.8077 - val_loss: 1.4831
```

Figure 5: CNN model with 3 convolutional layers and 32 layer size.

```
Epoch 1/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 230s 137ms/step - accuracy: 0.4812 - loss: 1.1151 - val_accuracy: 0.6985 - val_loss: 0.7599
Epoch 2/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 230s 139ms/step - accuracy: 0.7411 - loss: 0.6612 - val_accuracy: 0.7634 - val_loss: 0.6097
Epoch 3/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 226s 137ms/step - accuracy: 0.8188 - loss: 0.4753 - val_accuracy: 0.8217 - val_loss: 0.4833
Epoch 4/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 229s 138ms/step - accuracy: 0.8665 - loss: 0.3556 - val_accuracy: 0.8300 - val_loss: 0.4759
Epoch 5/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 219s 132ms/step - accuracy: 0.8867 - loss: 0.2972 - val_accuracy: 0.8192 - val_loss: 0.4941
Epoch 6/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 218s 131ms/step - accuracy: 0.9030 - loss: 0.2540 - val_accuracy: 0.8256 - val_loss: 0.5388
Epoch 7/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 216s 130ms/step - accuracy: 0.9150 - loss: 0.2220 - val_accuracy: 0.8386 - val_loss: 0.4800
Epoch 8/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 219s 132ms/step - accuracy: 0.9272 - loss: 0.1979 - val_accuracy: 0.8513 - val_loss: 0.4728
Epoch 9/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 219s 132ms/step - accuracy: 0.9323 - loss: 0.1779 - val_accuracy: 0.8480 - val_loss: 0.4706
Epoch 10/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 228s 138ms/step - accuracy: 0.9405 - loss: 0.1558 - val_accuracy: 0.8578 - val_loss: 0.5167
Epoch 11/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 217s 131ms/step - accuracy: 0.9459 - loss: 0.1421 - val_accuracy: 0.8497 - val_loss: 0.5329
Epoch 12/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 225s 136ms/step - accuracy: 0.9519 - loss: 0.1297 - val_accuracy: 0.8492 - val_loss: 0.5583
Epoch 13/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 222s 134ms/step - accuracy: 0.9566 - loss: 0.1159 - val_accuracy: 0.8492 - val_loss: 0.6532
Epoch 14/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 221s 133ms/step - accuracy: 0.9583 - loss: 0.1107 - val_accuracy: 0.8392 - val_loss: 0.7801
Epoch 15/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 225s 136ms/step - accuracy: 0.9613 - loss: 0.1041 - val_accuracy: 0.8321 - val_loss: 0.7296
```

Figure 6: CNN model with 4 convolutional layers and 16 layer size.

```
Epoch 1/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 604s 363ms/step - accuracy: 0.5089 - loss: 1.0671 - val_accuracy: 0.6988 - val_loss: 0.7364
Epoch 2/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 630s 380ms/step - accuracy: 0.7990 - loss: 0.5197 - val_accuracy: 0.8210 - val_loss: 0.4943
Epoch 3/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 625s 377ms/step - accuracy: 0.8786 - loss: 0.3206 - val_accuracy: 0.8485 - val_loss: 0.4277
Epoch 4/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 619s 374ms/step - accuracy: 0.9160 - loss: 0.2259 - val_accuracy: 0.8557 - val_loss: 0.4396
Epoch 5/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 631s 381ms/step - accuracy: 0.9383 - loss: 0.1685 - val_accuracy: 0.8676 - val_loss: 0.4226
Epoch 6/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 645s 389ms/step - accuracy: 0.9552 - loss: 0.1224 - val_accuracy: 0.8752 - val_loss: 0.4257
Epoch 7/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 643s 388ms/step - accuracy: 0.9627 - loss: 0.1027 - val_accuracy: 0.8754 - val_loss: 0.4533
Epoch 8/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 646s 390ms/step - accuracy: 0.9703 - loss: 0.0822 - val_accuracy: 0.8593 - val_loss: 0.5429
Epoch 9/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 649s 392ms/step - accuracy: 0.9723 - loss: 0.0786 - val_accuracy: 0.8894 - val_loss: 0.5030
Epoch 10/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 646s 390ms/step - accuracy: 0.9771 - loss: 0.0645 - val_accuracy: 0.8850 - val_loss: 0.4989
Epoch 11/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 647s 391ms/step - accuracy: 0.9794 - loss: 0.0569 - val_accuracy: 0.8721 - val_loss: 0.5725
Epoch 12/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 640s 386ms/step - accuracy: 0.9818 - loss: 0.0504 - val_accuracy: 0.8831 - val_loss: 0.5354
Epoch 13/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 642s 388ms/step - accuracy: 0.9828 - loss: 0.0495 - val_accuracy: 0.8853 - val_loss: 0.6218
Epoch 14/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 644s 389ms/step - accuracy: 0.9848 - loss: 0.0419 - val_accuracy: 0.8796 - val_loss: 0.6393
Epoch 15/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 645s 389ms/step - accuracy: 0.9859 - loss: 0.0395 - val_accuracy: 0.8804 - val_loss: 0.6809
```

Figure 7: CNN model with 4 convolutional layers and 32 layer size.

```
Epoch 1/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1653s 995ms/step - accuracy: 0.5317 - loss: 1.0381 - val_accuracy: 0.7643 - val_loss: 0.6048
Epoch 2/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1632s 985ms/step - accuracy: 0.8616 - loss: 0.3622 - val_accuracy: 0.8441 - val_loss: 0.4207
Epoch 3/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1618s 977ms/step - accuracy: 0.9398 - loss: 0.1650 - val_accuracy: 0.8569 - val_loss: 0.4568
Epoch 4/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 5251s 3s/step - accuracy: 0.9694 - loss: 0.0846 - val_accuracy: 0.8707 - val_loss: 0.4993
Epoch 5/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1619s 978ms/step - accuracy: 0.9797 - loss: 0.0612 - val_accuracy: 0.8489 - val_loss: 0.6750
Epoch 6/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1646s 994ms/step - accuracy: 0.9826 - loss: 0.0493 - val_accuracy: 0.8590 - val_loss: 0.7551
Epoch 7/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 2766s 2s/step - accuracy: 0.9881 - loss: 0.0347 - val_accuracy: 0.8754 - val_loss: 0.6176
Epoch 8/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1666s 1s/step - accuracy: 0.9879 - loss: 0.0328 - val_accuracy: 0.8638 - val_loss: 0.7743
Epoch 9/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1635s 987ms/step - accuracy: 0.9889 - loss: 0.0327 - val_accuracy: 0.8742 - val_loss: 0.8959
Epoch 10/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1636s 988ms/step - accuracy: 0.9894 - loss: 0.0314 - val_accuracy: 0.8746 - val_loss: 0.8285
Epoch 11/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1636s 988ms/step - accuracy: 0.9930 - loss: 0.0208 - val_accuracy: 0.8816 - val_loss: 0.8478
Epoch 12/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1623s 980ms/step - accuracy: 0.9902 - loss: 0.0288 - val_accuracy: 0.8807 - val_loss: 0.7969
Epoch 13/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1671s 1s/step - accuracy: 0.9935 - loss: 0.0196 - val_accuracy: 0.8900 - val_loss: 0.7802
Epoch 14/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1741s 1s/step - accuracy: 0.9940 - loss: 0.0184 - val_accuracy: 0.8946 - val_loss: 0.7285
Epoch 15/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 1627s 982ms/step - accuracy: 0.9948 - loss: 0.0153 - val_accuracy: 0.8877 - val_loss: 0.8282
```

Figure 8: CNN model with 4 convolutional layers and 64 layer size.

```
Epoch 1/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 163s 95ms/step - accuracy: 0.3858 - loss: 1.2479 - val_accuracy: 0.6157 - val_loss: 0.9135
Epoch 2/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 165s 99ms/step - accuracy: 0.6263 - loss: 0.8882 - val_accuracy: 0.6526 - val_loss: 0.8358
Epoch 3/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 162s 98ms/step - accuracy: 0.6957 - loss: 0.7626 - val_accuracy: 0.7040 - val_loss: 0.7569
Epoch 4/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 150s 91ms/step - accuracy: 0.7309 - loss: 0.6906 - val_accuracy: 0.7266 - val_loss: 0.7045
Epoch 5/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 153s 92ms/step - accuracy: 0.7553 - loss: 0.6447 - val_accuracy: 0.7521 - val_loss: 0.6469
Epoch 6/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 153s 92ms/step - accuracy: 0.7822 - loss: 0.5859 - val_accuracy: 0.7739 - val_loss: 0.6181
Epoch 7/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 153s 92ms/step - accuracy: 0.8054 - loss: 0.5297 - val_accuracy: 0.8081 - val_loss: 0.5589
Epoch 8/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 155s 94ms/step - accuracy: 0.8221 - loss: 0.4914 - val_accuracy: 0.7763 - val_loss: 0.6396
Epoch 9/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 154s 93ms/step - accuracy: 0.8304 - loss: 0.4637 - val_accuracy: 0.8253 - val_loss: 0.5023
Epoch 10/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 154s 93ms/step - accuracy: 0.8437 - loss: 0.4341 - val_accuracy: 0.8277 - val_loss: 0.4785
Epoch 11/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 152s 92ms/step - accuracy: 0.8499 - loss: 0.4154 - val_accuracy: 0.8204 - val_loss: 0.4882
Epoch 12/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 157s 95ms/step - accuracy: 0.8552 - loss: 0.4023 - val_accuracy: 0.8168 - val_loss: 0.5103
Epoch 13/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 151s 91ms/step - accuracy: 0.8587 - loss: 0.3935 - val_accuracy: 0.8342 - val_loss: 0.4592
Epoch 14/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 155s 93ms/step - accuracy: 0.8688 - loss: 0.3672 - val_accuracy: 0.8368 - val_loss: 0.4637
Epoch 15/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 156s 94ms/step - accuracy: 0.8667 - loss: 0.3691 - val_accuracy: 0.8297 - val_loss: 0.4821
```

Figure 9: CNN model with 5 convolutional layers and 8 layer size.

```
Epoch 1/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 341s 204ms/step - accuracy: 0.4920 - loss: 1.1045 - val_accuracy: 0.7262 - val_loss: 0.7250
Epoch 2/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 330s 199ms/step - accuracy: 0.7742 - loss: 0.5984 - val_accuracy: 0.7965 - val_loss: 0.5916
Epoch 3/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 333s 201ms/step - accuracy: 0.8280 - loss: 0.4671 - val_accuracy: 0.8206 - val_loss: 0.5338
Epoch 4/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 341s 206ms/step - accuracy: 0.8634 - loss: 0.3730 - val_accuracy: 0.8505 - val_loss: 0.4331
Epoch 5/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 358s 216ms/step - accuracy: 0.8836 - loss: 0.3188 - val_accuracy: 0.8499 - val_loss: 0.4188
Epoch 6/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 352s 213ms/step - accuracy: 0.8974 - loss: 0.2820 - val_accuracy: 0.8626 - val_loss: 0.4192
Epoch 7/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 348s 210ms/step - accuracy: 0.9095 - loss: 0.2476 - val_accuracy: 0.8707 - val_loss: 0.4138
Epoch 8/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 343s 207ms/step - accuracy: 0.9209 - loss: 0.2171 - val_accuracy: 0.8741 - val_loss: 0.3811
Epoch 9/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 349s 210ms/step - accuracy: 0.9277 - loss: 0.1997 - val_accuracy: 0.8573 - val_loss: 0.4109
Epoch 10/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 382s 231ms/step - accuracy: 0.9314 - loss: 0.1880 - val_accuracy: 0.8705 - val_loss: 0.3816
Epoch 11/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 363s 219ms/step - accuracy: 0.9371 - loss: 0.1708 - val_accuracy: 0.8695 - val_loss: 0.4259
Epoch 12/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 354s 214ms/step - accuracy: 0.9419 - loss: 0.1625 - val_accuracy: 0.8762 - val_loss: 0.4032
Epoch 13/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 343s 207ms/step - accuracy: 0.9454 - loss: 0.1528 - val_accuracy: 0.8827 - val_loss: 0.3755
Epoch 14/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 356s 215ms/step - accuracy: 0.9468 - loss: 0.1456 - val_accuracy: 0.8774 - val_loss: 0.3958
Epoch 15/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 336s 202ms/step - accuracy: 0.9533 - loss: 0.1308 - val_accuracy: 0.8767 - val_loss: 0.4738
```

Figure 10: CNN model with 5 convolutional layers and 16 layer size.

```
Epoch 1/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 819s 488ms/step - accuracy: 0.5091 - loss: 1.0711 - val_accuracy: 0.7347 - val_loss: 0.6593
Epoch 2/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 784s 473ms/step - accuracy: 0.8179 - loss: 0.4877 - val_accuracy: 0.8373 - val_loss: 0.4855
Epoch 3/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 796s 480ms/step - accuracy: 0.8841 - loss: 0.3148 - val_accuracy: 0.8583 - val_loss: 0.4100
Epoch 4/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 795s 480ms/step - accuracy: 0.9120 - loss: 0.2392 - val_accuracy: 0.8445 - val_loss: 0.4358
Epoch 5/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 805s 486ms/step - accuracy: 0.9347 - loss: 0.1782 - val_accuracy: 0.8797 - val_loss: 0.3906
Epoch 6/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 788s 476ms/step - accuracy: 0.9475 - loss: 0.1447 - val_accuracy: 0.8815 - val_loss: 0.4551
Epoch 7/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 793s 478ms/step - accuracy: 0.9566 - loss: 0.1187 - val_accuracy: 0.8868 - val_loss: 0.4126
Epoch 8/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 788s 476ms/step - accuracy: 0.9647 - loss: 0.0958 - val_accuracy: 0.8953 - val_loss: 0.3953
Epoch 9/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 785s 474ms/step - accuracy: 0.9706 - loss: 0.0823 - val_accuracy: 0.8938 - val_loss: 0.4410
Epoch 10/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 786s 474ms/step - accuracy: 0.9752 - loss: 0.0716 - val_accuracy: 0.8941 - val_loss: 0.4438
Epoch 11/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 815s 492ms/step - accuracy: 0.9769 - loss: 0.0654 - val_accuracy: 0.9017 - val_loss: 0.4388
Epoch 12/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 819s 494ms/step - accuracy: 0.9802 - loss: 0.0568 - val_accuracy: 0.8823 - val_loss: 0.4939
Epoch 13/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 817s 493ms/step - accuracy: 0.9803 - loss: 0.0551 - val_accuracy: 0.9024 - val_loss: 0.4896
Epoch 14/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 807s 487ms/step - accuracy: 0.9844 - loss: 0.0466 - val_accuracy: 0.9012 - val_loss: 0.5328
Epoch 15/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 816s 493ms/step - accuracy: 0.9838 - loss: 0.0468 - val_accuracy: 0.8934 - val_loss: 0.6010
```

Figure 11: CNN model with 5 convolutional layers and 32 layer size.

Based on the provided information, Model on Figure 10 appears to be the best choice having 5 convolutional layers, 1 dense layer, 16 number of layers and 32 batch size. It achieves the highest accuracy on both the training set (accuracy = 0.9454) and the validation set (val_acc = 0.8827), indicating good generalisation performance. Additionally, it has the lowest validation loss (val_loss = 0.3755), suggesting that it is effectively minimising prediction errors on unseen data. Therefore, the model in Figure 10 outperforms the other models in terms of both accuracy and loss metrics, making it the recommended choice.

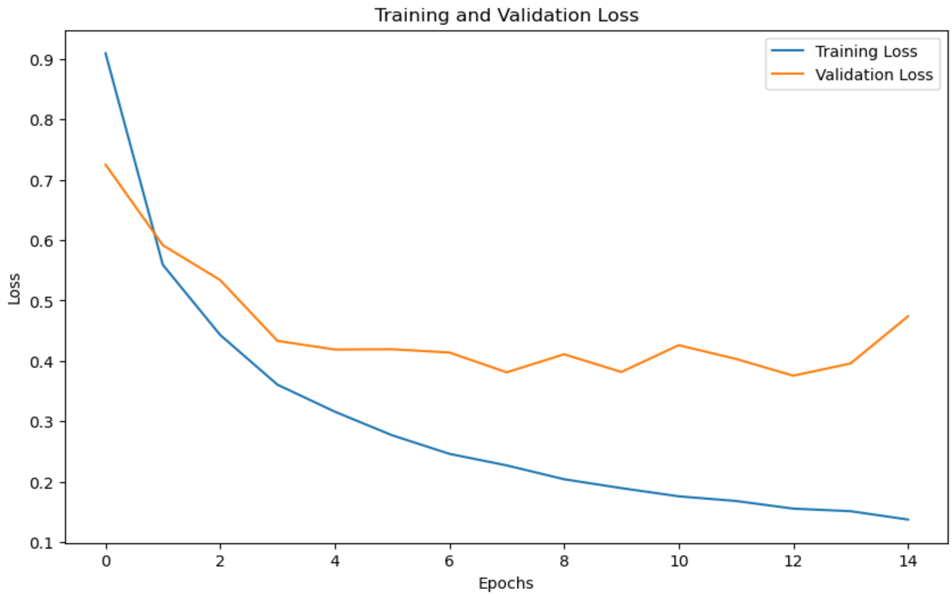

Figure 12: Training and validation loss plot through epochs.

Figure 10 depict both training and validation losses decrease, showing that the model is learning and improving on both the training and validation datasets. After a few epochs, the validation loss starts to flatten and eventually increases slightly while the training loss continues to decrease. This divergence suggests that the model is beginning to overfit the training data. An accuracy of 0.9454 on the training set and 0.8827 on the validation set for a CNN model trained on brain tumor images is a good starting point, but it's not sufficient to determine whether the model is good enough for this specific application. The fact that the validation accuracy (0.8827) is slightly lower than the training accuracy (0.9454) suggests that there might be some degree of overfitting. Overfitting occurs when the model learns to perform well on the training data but fails to generalise to unseen data.

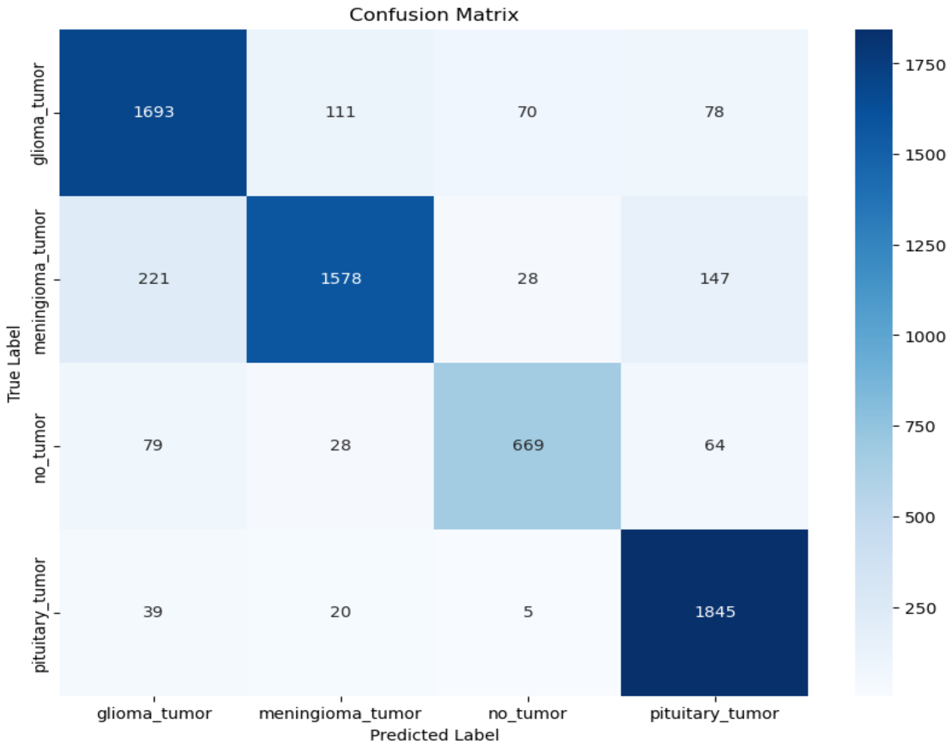

Figure 13: Training and validation loss plot through epochs.

The confusion matrix displayed in Figure 13 shows the performance of your CNN model in classifying four types of brain conditions: glioma tumor, meningioma tumor, no tumor, and pituitary tumor. The figure below shows the classification report. Based on both the figures we can see the model fits well which shows overfitting.

Classification Report:

	precision	recall	f1-score	support
glioma_tumor	0.83	0.87	0.85	1952
meningioma_tumor	0.91	0.80	0.85	1974
no_tumor	0.87	0.80	0.83	840
pituitary_tumor	0.86	0.97	0.91	1909
accuracy			0.87	6675
macro avg	0.87	0.86	0.86	6675
weighted avg	0.87	0.87	0.87	6675

Figure 14: Classification report for the best model.

4.2 Model adjustment

We must avoid overfitting in order to modify the adequate model we used in the previous section on Figure 10 and generalise it. We employed dropout, a regularisation approach that is frequently used to reduce overfitting, to stop overfitting. By randomly eliminating (i.e., setting to zero) a specific percentage of neurons in a layer during training, dropout helps to prevent this. Moreover, early stopping was also consider to ensures that the model does not continue to learn from the training data to the point where it starts to memorise it rather than generalize from it using a patience of 2, which determines to wait for an improvement in the monitored metric before stopping the training.

During training, the dropout technique works as follows:

- At each training iteration, each neuron (or node) in the layer to which dropout is applied has a probability P of being temporarily ”dropped out,” meaning its output is set to zero.
- This process is stochastic, meaning which neurons are dropped out varies from iteration to iteration.
- Typically, a dropout rate of around 0.25 to 0.5 is used, meaning each neuron has a 25% to 50% chance of being dropped out.

The following is our trained model with dropout included during training.

```
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 391s 232ms/step - accuracy: 0.3646 - loss: 1.2840 - val_accuracy: 0.4316 - val_loss: 1.2969
Epoch 2/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 475s 287ms/step - accuracy: 0.5229 - loss: 1.0908 - val_accuracy: 0.5205 - val_loss: 1.0686
Epoch 3/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 387s 233ms/step - accuracy: 0.5982 - loss: 0.9567 - val_accuracy: 0.6244 - val_loss: 0.8836
Epoch 4/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 390s 235ms/step - accuracy: 0.6287 - loss: 0.8805 - val_accuracy: 0.6472 - val_loss: 0.8295
Epoch 5/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 403s 243ms/step - accuracy: 0.6569 - loss: 0.8346 - val_accuracy: 0.6615 - val_loss: 0.8156
Epoch 6/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 420s 253ms/step - accuracy: 0.6772 - loss: 0.7952 - val_accuracy: 0.6638 - val_loss: 0.8054
Epoch 7/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 419s 253ms/step - accuracy: 0.6960 - loss: 0.7646 - val_accuracy: 0.7029 - val_loss: 0.7477
Epoch 8/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 416s 251ms/step - accuracy: 0.7120 - loss: 0.7323 - val_accuracy: 0.7165 - val_loss: 0.7250
Epoch 9/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 439s 265ms/step - accuracy: 0.7217 - loss: 0.7084 - val_accuracy: 0.7451 - val_loss: 0.6405
Epoch 10/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 430s 259ms/step - accuracy: 0.7349 - loss: 0.6837 - val_accuracy: 0.7026 - val_loss: 0.7465
Epoch 11/15
1656/1656 ━━━━━━━━━━━━━━━━━━━━ 431s 260ms/step - accuracy: 0.7397 - loss: 0.6766 - val_accuracy: 0.7153 - val_loss: 0.7271
```

Figure 15: CNN model with 5 convolutional layers and 16 layer size with dropout and patience of 2.

On training set the training accuracy might be lower compared to the model without dropout since neurons are randomly dropped during training, making it harder for the model to overfit to the training data since we can see the accuracy starts with 0.3646 and end with 0.7397. Validation accuracy is expected to improve compared to the model without dropout, indicating better generalization as we can see that the validation accuracy is not far-off from the training accuracy. The model with dropout is applied patience of 2 to stop running to prevent overfitting and it stops at epoch 11. The following shows the plots for Training and validation loss plot through epochs with dropout.

Figure 16: Training and validation loss plot through epochs with dropout and patience of 2.

5 Results

The confusion matrix and classification report are used, and they are discussed below, to show how the trained model obtained above performs on the test dataset (unseen data) and further see the predictions obtained.

5.1 Confusion matrix

An effective tool for assessing a classification our model's performance is a confusion matrix as it offers a thorough analysis of the differences between the model's predictions and the actual results, revealing areas of the model's accuracy and inaccuracy. The confusion matrix obtained from our model is present at Figure below.

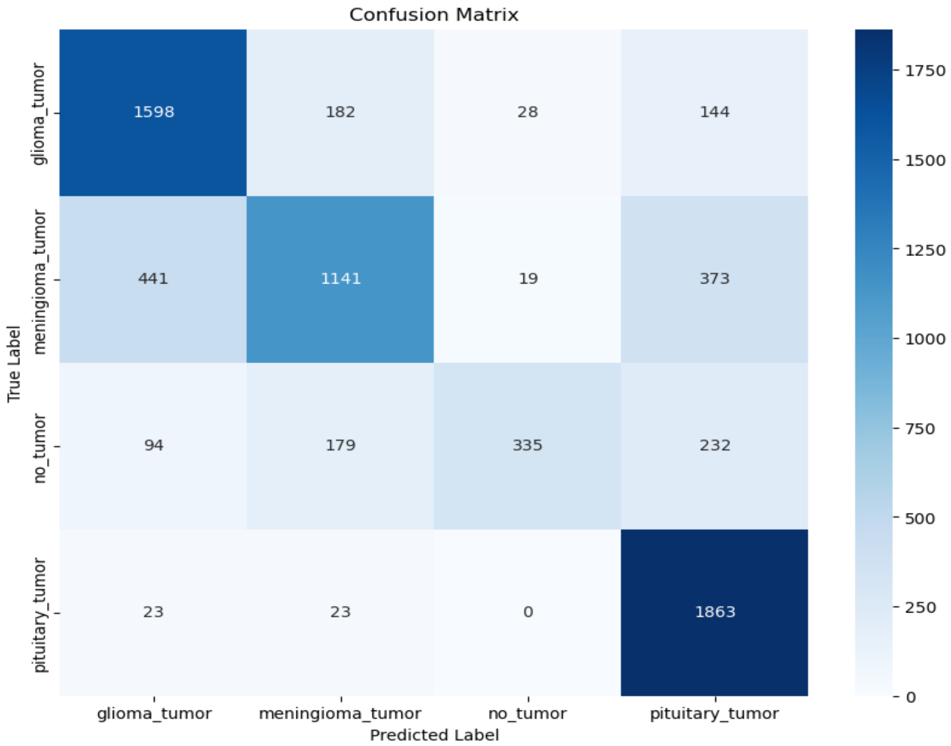

Figure 17: Confusion matrix for the best model applied dropout and patience of 2.

According to the results above the following results were obtained:

- **Glioma Tumor**- the model correctly identifies 1598 instances of glioma tumors, but there are significant misclassifications, particularly to meningioma tumors (182) and followed by pituitary tumors (144).
- **Meningioma Tumor** - The model has 1141 true positives for meningioma tumors, however there were more misclassifications for glioma tumors (441) and pituitary tumors (373).
- **No Tumor** - With only 335 accurate classifications and substantial misclassifications to other tumour kinds, particularly pituitary tumours (232), the model performs worse when it comes to the ”no tumour” class.
- **Pituitary Tumor**- This class has the highest number of true positives (1863) and relatively fewer misclassifications, indicating the model performs best in identifying pituitary tumors.

The performance of the classifier varies between classes. It struggles greatly to distinguish no tumours, but it works best with pituitary tumours. Notably, there is considerable confusion regarding meningioma and glioma tumours, and many cases of ”no tumour” are actually pituitary tumours. The class designated as ”no tumour” exhibits the lowest true positive rate, underscoring the necessity for enhanced identification of cases lacking tumours. However, overall we can confidently say the model perform well.

5.2 Classification report

To further see how our trained model its performs a classification report which offers a balanced view of both precision and recall, which is especially important in scenarios where one metric might be more critical than the other (e.g., in medical diagnoses, fraud detection). The classification report is presented in Figure 19.

Classification Report:

	precision	recall	f1-score	support
glioma_tumor	0.74	0.82	0.78	1952
meningioma_tumor	0.75	0.58	0.65	1974
no_tumor	0.88	0.40	0.55	840
pituitary_tumor	0.71	0.98	0.82	1909
accuracy			0.74	6675
macro avg	0.77	0.69	0.70	6675
weighted avg	0.75	0.74	0.73	6675

Figure 18: Classification report for the best model with dropout and patience of 2

The key observation made in the figure 19 are presented as follows:

- **Glioma Tumor**- The model performs well in detecting glioma tumors, with high recall (0.82) and a good F1-score (0.78).

- **Meningioma Tumor** - he recall is relatively low (0.58), indicating the model misses a significant number of meningioma tumors.
- **No Tumor** - While the precision is high (0.88), the recall is very low (0.40), suggesting that many actual no tumor instances are being misclassified.
- **Pituitary Tumor**- The model performs excellently in identifying pituitary tumors with a recall of 0.98 and a high F1-score (0.82).

The classification report revealed that the model has an accuracy of 74% on the unseen data (testing data).

The model performs differently in each class; it performs best when pituitary tumours are detected, but it has severe difficulties when there is ”no tumour” in an instance as it was also indicated in the confusion matrix. The model is confident when it predicts ”no tumour,” but it frequently fails to identify actual ”no tumour” instances, as evidenced by the high precision but low recall for the ”no tumour” class. The model has a 74% overall accuracy rate, which is respectable, but there is still much space for improvement, especially when it comes to evenly distributing recall among classes.

5.3 Classification results

The below is the some of the results that we got when using the test dataset on the train model to predict our class. The model successfully classified the images into one of the four classes.

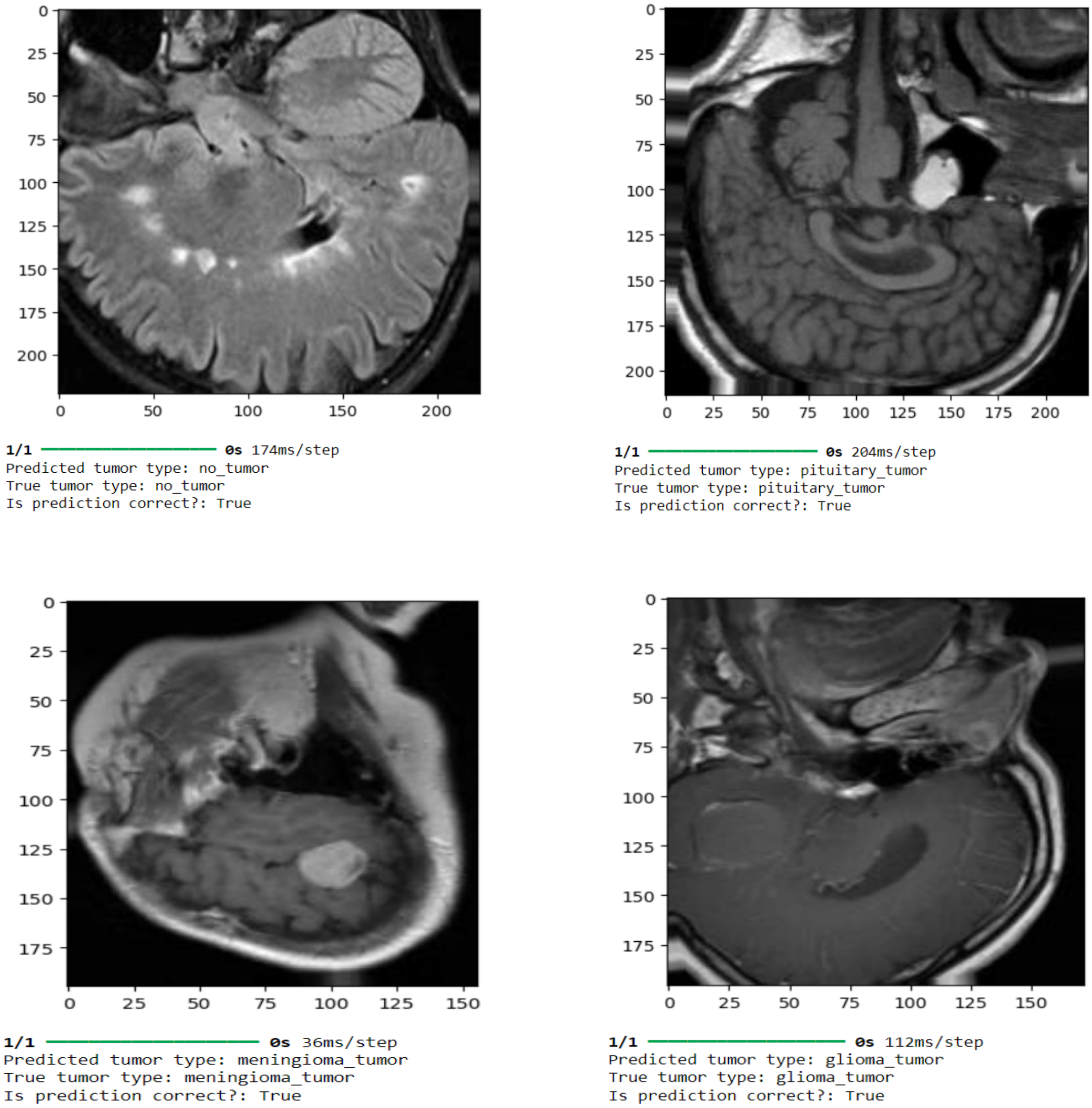

Figure 19: Brain tumor classification results.

6 Conclusion

In this report, we aimed to develop and evaluate a CNN for the detection and classification of brain tumors using MRI scans. The dataset utilized comprised MRI images of glioma tumors, meningioma tumors, pituitary tumors, and non-tumorous brains, Our objectives were to classify these MRI scans as either with tumor or without tumor and further categorize them into one of the four specific classes. The dataset consisted of 52964 images for training, 6650 for validation, and 6675 for testing, belonging to 4 classes after being cropped, and augmented. The best CNN model after training different models on the preprocessed data had an accuracy of 0.77 and a weighted average F1 score of 0.73. This report underscores the potential of CNNs in brain tumor classification, highlighting both their effectiveness and the need for ongoing refinement to enhance diagnostic accuracy and reliability across all tumor classes.

References

Alzahrani, S. M. (2023), 'Convattenmixer: Brain tumor detection and type classification using convolutional mixer with external and self-attention mechanisms', *Journal of King Saud University-Computer and Information Sciences* **35**(10), 101810.

Bai, F., Deng, Y., Li, L., Lv, M., Razzokov, J., Xu, Q., Xu, Z., Chen, Z., Chen, G. and Chen, Z. (2024), Advancements and challenges in brain cancer therapeutics, *in* 'Exploration', Wiley Online Library, p. 20230177.

HOSOI, K. (1930), 'Multiple gliomas of the brain', *Archives of Neurology & Psychiatry* **24**(2), 311–323.

Pichaivel, M., Anbumani, G., Theivendren, P. and Gopal, M. (2022), 'An overview of brain tumor', *Brain Tumors* **1**.

University, T. J. H., Hospital, T. J. H. and System, J. H. H. (2024), 'Brain tumors and brain cancer'. Accessed: 2024-05-27.
URL: *https://www.hopkinsmedicine.org*

Waghmare, V. K. and Kolekar, M. H. (2021), 'Brain tumor classification using deep learning', *Internet of things for healthcare technologies* pp. 155–175.

Wäldchen, J. and Mäder, P. (2018), 'Machine learning for image based species identification', *Methods in Ecology and Evolution* **9**(11), 2216–2225.

Zülch, K. J. (2013), *Brain tumors: their biology and pathology*, Springer-Verlag.