

# Capstone Project Documentation

Authors: Laura Madrid, Lucas Noritomi-Hartwig, Keshav Worathur

## Problem

Brain tumours affect approximately 50,000 Canadians every year, according to the Brain Tumour Registry of Canada (BTRC) [1]. Manual diagnosis of brain tumours is time-intensive and requires specialized knowledge of the brain [2]. We seek to develop automatic methods for diagnosis of glioma, meningioma, and pituitary tumors from Magnetic Resonance Imaging (MRI) scans. This entails a multi-class classification problem to which we hope to apply machine learning techniques in new and insightful ways.

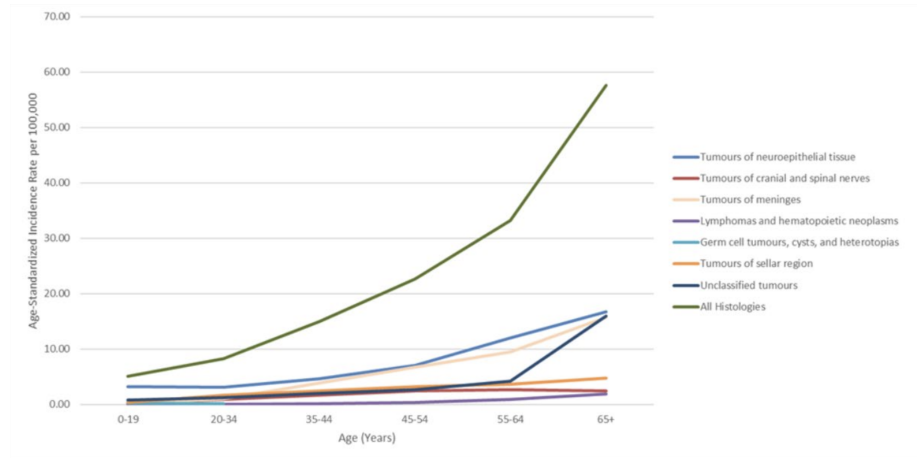


Figure 1: Rates

## Dataset

Our chosen dataset is the figshare brain tumour dataset [3]. This dataset is available at kaggle. The dataset contains 3,064 slices of MRI scans from patients at Nanfang Hospital, Guangzhou, China. The slices were taken in the sagittal, axial, and coronal planes.

Examples of the tumour scans are shown below:

The input to the model is a MRI slice resized to  $256 \times 256$  pixels. The labels are 1 for Meningioma, 2 for Glioma, and 3 for Pituitary.

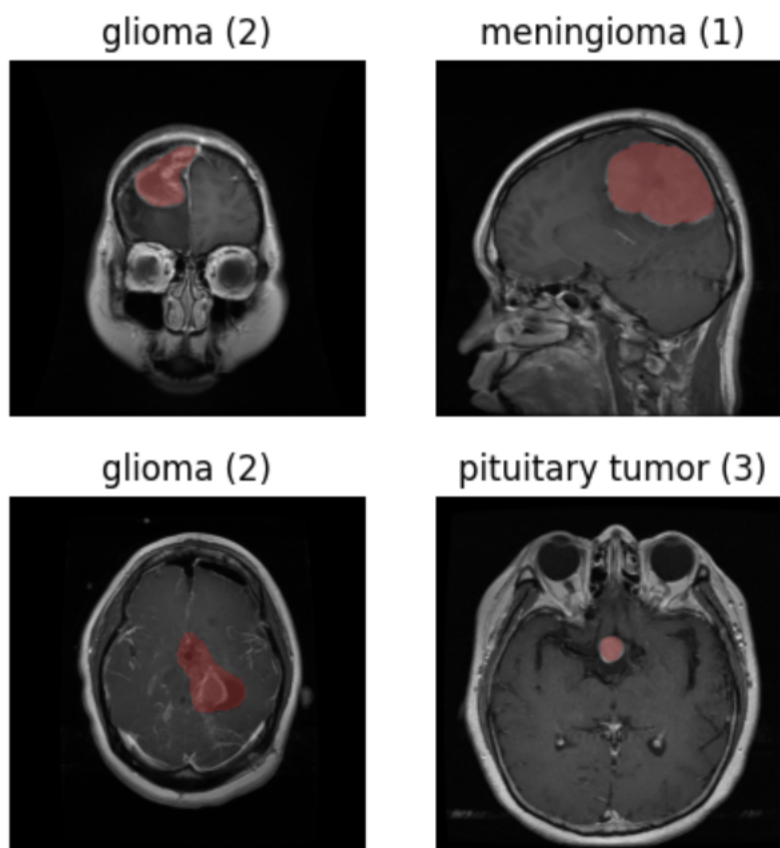


Figure 2: Examples

## Implementation Details

We implemented a custom Convolutional Neural Network (CNN) in tensorflow.

### Loading the Data

To load in the dataset, we created a custom pre-processing pipeline:

1. Resize the image from  $512 \times 512$  pixels to  $256 \times 256$  pixels.
2. Augment the image by performing a  $90^\circ$  rotation or be performing a flip over the horizontal axis.
3. Standardizing the pixels values in the input image.

We load the images into memory in batches of 16, to avoid exceeding RAM quota in Google Colab. We trained our model using the NVIDIA T4 GPU available in Colab.

### Model Implementation

Our model consists of four classification blocks, arranged in a sequence to down-sample the input image into a summary vector of size 2048. A diagram of a single classification block is shown below:

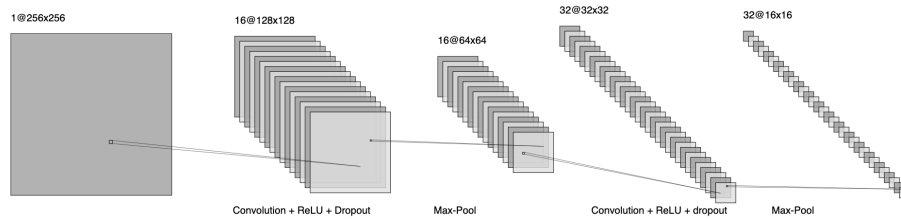


Figure 3: Architecture

The model outputs a **vector of probabilities**, where each probability represents the likelihood that a tumor class is present in the MRI scan. We take the class corresponding to the largest probability as the model's prediction.

### Novel Contributions

When a model and a neurologist differ in their opinions about a scan, how can we reconcile their differences? Our novel contribution is the use of saliency maps for interpreting our classifier's decisions. Our saliency maps are computed by taking partial derivatives of the predicted class with respect to each pixel of the original image.

Model: "sequential\_1"

Layer (type)	Output Shape	Param #
resizing_1 (Resizing)	(None, 256, 256, 1)	0
normalization (Normalization)	(None, 256, 256, 1)	131073
conv2d (Conv2D)	(None, 128, 128, 16)	416
dropout (Dropout)	(None, 128, 128, 16)	0
max_pooling2d (MaxPooling2D)	(None, 64, 64, 16)	0
conv2d_1 (Conv2D)	(None, 32, 32, 32)	4640
dropout_1 (Dropout)	(None, 32, 32, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_2 (Conv2D)	(None, 16, 16, 64)	18496
dropout_2 (Dropout)	(None, 16, 16, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 8, 8, 64)	0
conv2d_3 (Conv2D)	(None, 8, 8, 128)	73856
dropout_3 (Dropout)	(None, 8, 8, 128)	0
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 2048)	4196352
dense_1 (Dense)	(None, 3)	6147
Total params: 4430980 (16.90 MB)		
Trainable params: 4299907 (16.40 MB)		
Non-trainable params: 131073 (512.01 KB)		

Figure 4: Description

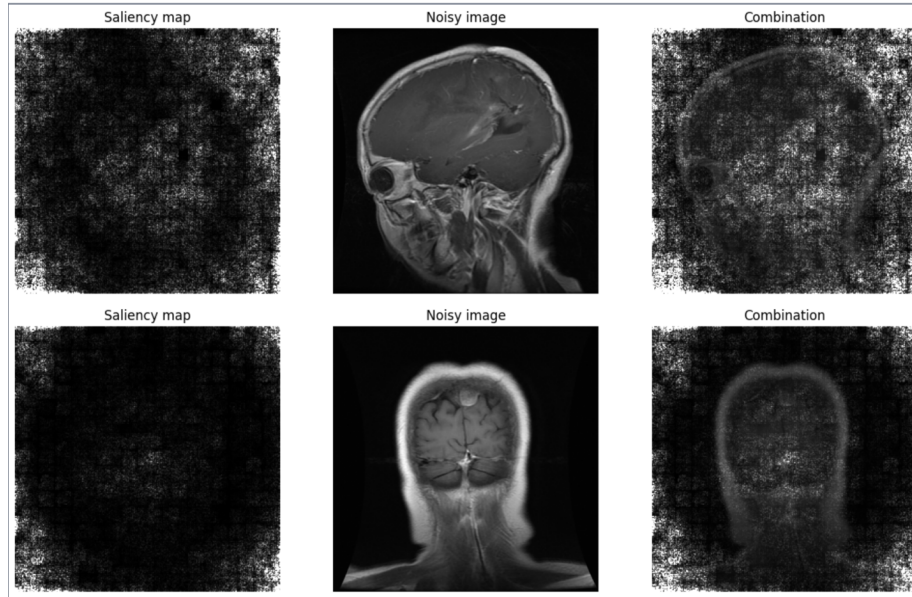


Figure 5: Saliency Maps

## Evaluation Results

When evaluating our model on a single test set, we obtained a test accuracy of 93.4%.

We also used subject-wise cross validation for evaluating the model:

- Split the dataset into 10 subsets, called **folds**, where each patient can appear in only one fold. The number of distinct patients in each fold is roughly equal.
- Set aside two folds for the test set, two folds for the validation set, and using the remaining six folds as the training dataset.
- Train the model and record the test accuracy.
- Repeat this process until every example has appeared in the test set. Average the test accuracies to produce an estimate of the model’s performance on unseen data.

We obtained an average test accuracy of 91.3%. This shows the model can differentiate between the different tumor classes. However, there is strong evidence the model may misclassify tumors when deployed in the real world.

## Individual Contributions

Laura Madrid explored novel extensions to our project, such as GANs for producing counterfactual images and the vanilla gradient method of producing saliency maps.

Lucas Noritomi-Hartwig selected the dataset and performed data pre-processing. He also researched novel extensions to our project and wrote the code for saliency maps.

Keshav Worathur researched related works pertaining to our problem and set up the project repository. He wrote code for the data pre-processing pipeline and trained the model.

## References

- [1] Brain Tumour Registry of Canada. <https://braintumourregistry.ca/>, 2019. Accessed: 2023-10-01.
- [2] E. S. Biratu, F. Schwenker, Y. M. Ayano, and T. G. Debelee, “A survey of brain tumor segmentation and classification algorithms,” *J Imaging*, vol. 7, Sept. 2021.
- [3] Milica M Badža and Marko Č Barjaktarović. Classification of brain tumors from mri images using a convolutional neural network. *Applied Sciences*, 10(6):1999, 2020.

## Workflow

The notebook `brain_tumor_classification.ipynb` contains code for \* Downloading the dataset. \* Visualizing the data. \* Pre-processing the data. \* Training the model. \* Producing saliency maps from the trained model.

The notebook downloads the dataset using the kaggle API, which requires a kaggle API key to be uploaded in the second cell of the notebook. After this initial input, the rest of the notebook can be run without interaction from the user. The data will be visualized followed by training of the model using the “one test set” evaluation method. Finally, the saliency maps are produced, providing insight into the model’s decisions.