Brain Tumour Detection using Magnetic Resonance Imaging and Machine Learning

Madison Boem-20063662, Caroline Kim-20091781, Eric Venditti-20070722, Jade Watson-20052115

Abstract—Brain tumors are a life-threatening illness, with a rapid growth rate and short life expectancy in their most severe form. For this reason, early intervention, surgical plans, and treatment planning are crucial in improving a patient's quality of life. Currently, to confirm tumors in the brain, image methods such as computed tomography (CT), magnetic resonance imaging (MRI), and ultrasound images are used. These images are then manually segmented to diagnose if a patient has a tumor. This is a time-consuming process prone to human error. For this reason, this paper proposes the use of a convolutional neural network (CNN) to detect brain tumours from MRI images. Three research questions are proposed since this is a well-studied area. The first question inquires whether transfer learning can be effectively used to detect tumours in other parts of the body. The second question explores the effects of reducing pre-processing on training times, model accuracy and model loss. The systematic approach used showed that the combination of pre-processing steps that led to the fastest training time of 12 minutes was cropping and dilating the input images. The combination of cropping, dilating, and normalizing the input images yielded the highest accuracy at 95%. The third research question explored if computational complexity could be reduced while sustaining or improving prediction accuracy. Increasing the batch size improved the model accuracy by 12%, however increased the model run time by 20 minutes. Researching models for a computationally-efficient image classifier may help medical professionals perform life-saving procedures more efficiently.

Index Terms—Group 6, Life Sciences, Machine Learning, Convolutional Neural Networks, Image Classification.

1 Introduction

A brain tumor is a growth of abnormal cells in the brain [6]. There are two types of tumors: benign and malignant. Benign tumors are not cancerous and cannot grow or spread to other organs. Whereas, malignant tumors are cancerous and can spread to other organs of the body. An unexpected development of a malignant brain tumor threatens the life of a person. Even though brain cancer accounts for less than 2% of all cancers in humans, it can cause dramatic health consequences including death [5]. According to the cancer association, brain and nervous system cancer is the 10th leading cause of death in adults with a booming number 18,280 adults [5]. MRI's are used to capture brain tumors. To diagnose a patient with a brain tumor, a medical professional must manually segment the images into tumorous and non-tumorous. This task requires additional time that could be used to accelerate a patient's quality of life. In addition, this process is prone to human error, which may lead to the misclassification of an MRI and ultimately may prevent people from receiving necessary treatment. For this reason, the motivation of this paper is to apply machine learning techniques to classify brain tumors quickly. In particular, a convolutional neural network is proposed to assist medical professionals in classifying brain tumors by automatically parsing large datasets.

A machine learning process is followed to complete the image classification task: data pre-processing, model building and training, and model evaluation. During data pre-processing, image features are extracted using cropping, dilation, normalization, removing noise, erosion, and colouring processes. The model performance is evaluated and addressed using three research questions. The first question utilizes transfer learning to attempt to improve model accuracy. The second research question explores the effects of reducing pre-processing of the images. The third question experiments with model architecture and parameters to minimize model computational complexity.

Deep learning methods have been used extensively to detect brain tumors especially using the implementation of CNNs. For this reason, the main contribution of this research are to address the three questions proposed above. The findings of research question 2 show that the order and amount of pre-processing affects the accuracy of the model. Specifically, it is shown that the combination of cropping and dilating obtained the fastest training time and the combination of crop, dilate and normalize produced the largest accuracy. Furthermore, research question 3 indicates that varying the number of epochs has the largest effect on accuracy.

2 Related Work

The current industry standard for lesion detection is manual segmentation and diagnosis. The task of automated segmentation however, has been widely researched, tested, and reviewed within the medical community. Existing works on this topic include the testing of convolutional neural networks with other accepted segmentation methods, the improvement of existing deep learning models for segmentation with different stratifications of tumor data, and a systematic review of currently accepted

- *All group members are in Applied Mathematics and Engineering at Queen's University*
16mab11@queensu.ca, kim.caroline@queensu.ca, eric.venditti@queensu.ca, 16jw73@queensu.ca

methods for automated lesion segmentation. These related works are discussed hereinafter.

Research by Zavaliangos-Petropulu et. al. [2] conducted at the University of Southern California examined hippocampal volume as a biomarker in identifying post stroke dementia, and compared a new CNN based framework with two previously tested methods. The CNN framework Hippodeep, was tested as a neural network approach designed specifically for hippocampal segmentation in a stroke population. This was compared with a currently accepted hippocampal segmentation and labeling software known as FreeSurfer, and manual approaches to diagnosis. The ATLAS dataset was used as input for all three methods, which is an open source collection of T1-weighted MRI scans of stroke patients. This dataset included 229 scans of stroke patients and was normalized prior to analysis. This study concluded that Hippodeep is a reliable and accurate method for image segmentation in a stroke population, when compared to FreeSurfer software and manual segmentation methods. Both automated methods had good correlation with manually detected results, and this study confirmed their success and efficacy.

Rabsemen et al. [4] investigates the use of stratification as a means to improve currently used deep learning models for image segmentation. This work proposes the separation of data based on tumor grade, type, and size, as a method for investigating the influence of these factors on model performance. The 2018 BraTS dataset was used in this study, and tumors were separated into high-grade glioma, and low-grade for the training dataset. This work used data augmentation by flipping images, and performing rotations at random. Two separate networks were trained based on the stratified tumor data, and the predictions of these models were ensembled for the final classification. It was determined that the stratification of data based on tumor grade resulted in better performance in 64.9% of cases, with the use of ensemble learning and a convolutional neural network.

A systematic review conducted by Isin et al. [1] reviewed the field's leading deep learning methods for automated MRI segmentation. This review examined the various CNN architectures utilized in current and previous research for this task. Further steps are discussed and the potential to examine other imaging modalities such as Positron Emission Tomography are also examined. This review provides evidence to support the use of CNN models for automated tumor detection and classifies the use of such deep learning methods as "state-of-the-art." [1]

3 Methodology

3.1 Data Pre-Processing

There were some issues with the chosen image dataset that needed to be addressed before they could be considered a reasonable input for the CNN.

To get an idea, shown below in Figure 1 are two raw images with the tumorous classification.

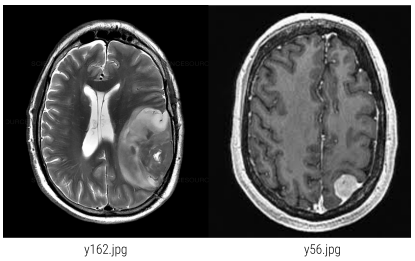

Fig. 1: y162.jpg and y56.jpg side-by-side comparison.

The most apparent difference is the size between these two images. y162.jpg has dimensions 1059x1200 pixels whereas y56.jpg has dimensions 211x239 pixels. Moreover, in these two images (along with most images in this dataset) there are large black borders surrounding the relevant brain scans. This is an issue because the CNN input layer in Keras (and general best practice) dictate that images be of the same size (height, width, channels). Additionally, the black borders are insignificant data that will contribute unnecessary complexity when fed into the CNN.

These issues were remedied in a two-step approach. First, a cropping technique was used to remove the maximum amount of border whilst retaining the full brain scan. Then, each image was resized to a logical dimension. Aiding in the first task were the Python packages cv2 and imutils. Automatically cropping each image was a process of detecting the four extreme corners of the brain scan, and those corners subsequently became the new image corners.

Working backwards, with cv2, this corner detection is only possible once the image contour is obtained. Hence, to find the brain scan contour, each image went under a series of pre-processing techniques. This is not to be confused with the pre-processing of the CNN input, as this preprocessing is solely for acquiring the brain contour. This preliminary pre-processing includes:

1) Greyscaling each image.
2) Removing Noise.
3) Thresholding the image (i.e. each pixel becomes a binary white/black). This is the fundamental step for contour detection.
4) Erosion and dilation (this helps smooth out the curve).

The transformation process is given below in Figure 2.

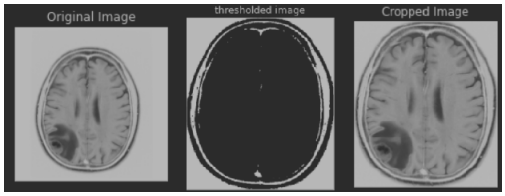

Fig. 2: Original image, thresholded image, and resulting crop.

Once the images were cropped, they needed to be resized.

The final dimensions were chosen to be the average width and height of each image in the dataset. The number of channels was 3.0 for each. These dimensions are shown in Table 6.

Once these issues had been addressed, other image pre-processing techniques were explored. These included dilation, erosion, normalization, denoising, and grey-scaling. An image summary of these techniques is shown below in Figure 3.

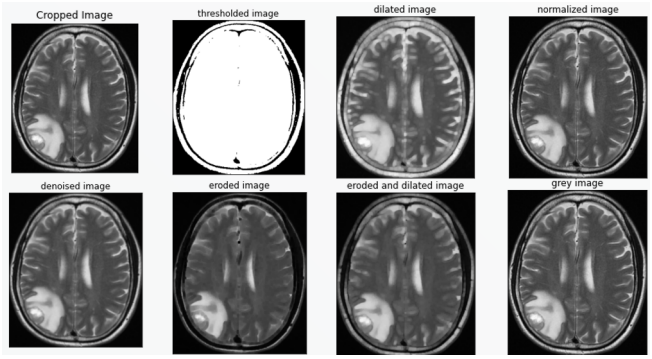

Fig. 3: Many of the image pre-processing techniques tested throughout this project.

3.2 Feature Engineering

The original dataset was slightly imbalanced, as 61% of the scans are tumorous and 39% are healthy. To remedy this, data augmentation was used.

To balance the tumorous and healthy scans, 9 new images were created for each healthy scan and 6 new images were created for each tumorous scan.

A number of augmentation techniques were used to generate the new images. These are given below:

- rotation_range: within a specified range (in degrees) each image underwent a random rotational shift
- width_shift_range: each image was shifted slightly in the width axis – defined as a fraction of total image width
- height_shift_range: each image was shifted slightly in the height axis – defined as a fraction of total image height
- shear_range: images underwent a shear operation, value is an angle in degrees for the counter-clockwise direction
- brightness_range: the brightness of each image was randomly altered range for randomly choosing a brightness shift
- horizontal_flip: will flip or not flip an image horizontally (with equal probability)
- vertical_flip: will flip or not flip an image vertically (with equal probability)
- fill_mode: when images leave the boundary (from the aforementioned shifts) this specifies how to deal with the previously occupied space

An example of an original and corresponding augmented image are given below in Figure 4.

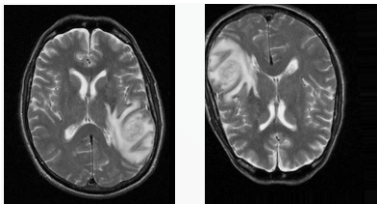

Fig. 4: Y71.png before and after data augmentation

3.3 Neural Network Architecture

Due to the high dimensionality of an image, a CNN introduces the idea of using hidden convolutonal layers to reduce channel dimensions. Each pixel of an image is a feature, and hence the model receives thousands of pixels as an input. The CNN converts the pixels in its field into a single value. Each hidden layer includes multiple nodes that represent a pixel of the image. The previous layer of each node is linked to the subsequent layer of nodes, however, nodes within the same layer are not connected. A function and a weight is associated to each node such that the output of the node is skewed toward a positive or negative value. This output indicates the importance of the node to the outcome classification. The node output is then passed to the fully connected layer that uses an activation function to determine the vector output. The proposed architecture of the CNN is illustrated below in Figure 5.

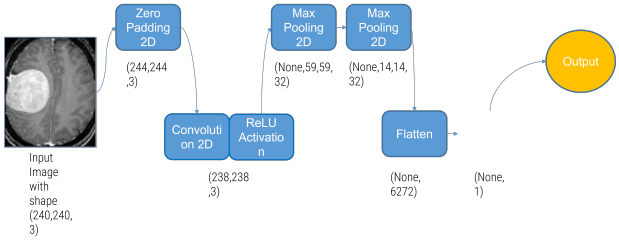

Fig. 5: Proposed architecture of the CNN.

There are four main operations in a CNN: convolution, non-linearity (ReLU), pooling, and classification (fully connected layer). Prior to the input image being passed through the convolutional hidden layer, zero padding is applied to preserve the original input size. This avoids losing information at the boundaries and thus weights in a filter will not differ significantly from its center values. The main hidden layer is the convolutional layer. It contains a set of filters to be learned through training. A filter is created when a collection of weights is multiplied by an array of input data. Since this is a systematic process, the filter is designed to detect specific features of an image such as a lesion. A ReLU activation function is used to add non-linearity to the network.

Two pooling layers are used to remove insignificant features by down sampling. To do so, max pooling characterizes the maximum active existence of a function [3] As a result, the image becomes smaller but the importance of the contents

remains the same. Since the model results in a binary output, a flatten layer is used to convert the data from a matrix to a one-dimensional array. This is then passed through a dense layer to make it linear. Overall, the model classifies the image into either a zero or one where a zero indicates the non-tumorous class and a one indicates the tumorous class.

3.4 Research Questions

3.4.1 How can transfer learning be effectively utilized?

This research question is centered around the idea of using transfer learning to solve a different, yet related task. In this case, the model is trained to distinguish between MRIs of brain scans and predict whether a tumor is present. The related task being explored is distinguishing between tumorous and healthy MRIs of organs in the body other than the brain. The ability to classify other MRIs provides the same assistance to medical professionals, essentially reducing the amount of manual analysis and subjectivity.

3.4.2 Can pre-processing be reduced and what are the implications?

This research question was posed to discover the effects that removing certain data pre-processing steps would have on the model's accuracy and overall run time. The comparison is done systematically, where only one variable (pre-processing step) is altered at a time and everything else is kept fixed.

This research question was divided into two separate experiments. The first focuses on many of the pre-processing steps mentioned in Section 3. By systematically adding, removing, or swapping the order of pre-processing steps, their benefits and drawbacks become apparent.

Below in Table 1 is an outline of the experiment with corresponding reasons regarding why each pre-processing step was selected.

Pre-processing Steps	Justification
1. No pre-processing	Use as a baseline for comparison
1. Cropping 2. Dilation	Reduce memory required for pre-processing
1. Cropping 2. Dilation 3. Normalization	Normalize pixel values for memory considerations
1. Cropping 2. Dilation 3. Denoising	Eliminate artifacts and noise from images to improve true tumour detection
1. Cropping 2. Dilation 3. Denoising 4. Normalization	Normalize pixel values after denoising to avoid losing information about artifacts and noise
1. Cropping 2. Dilation 3. Normalization 4. Denoising	Normalize pixel values before denoising for memory considerations

TABLE 1: Pre-processing step combinations and corresponding justification.

The second experiment hones in on one specific pre-processing technique, image dimensions. With cropping and normalization fixed, the image height and width were fluctuated. The experiment outline is given below in Table 2.

Image Dimensions	Acc	Loss	Time
(140, 140, 3)			
(240, 240, 3)			
(340, 340, 3)			
(440, 440, 3)			

TABLE 2: Fluctuating image dimensions to be tested

Given that data pre-processing on a large scale can be a time-consuming process, this question is important to analyze as it may reveal pre-processing steps that can be forgone.

3.4.3 Can Model Accuracy be Sustained with Lower Computational Complexity

While the previous research question is geared toward input complexity, this research question focuses specifically on the inherent model complexity. This question aims to yield a comparison of various model parameters and their effects on accuracy, loss, and total run-time.

This experiment is divided into three sections. The first fixes the number of convolutional layers and batch size then varies the number of epochs. This setup is shown below in Table 3.

Epochs	Acc	Loss	Time
5			
10			
15			
20			

TABLE 3: Varying the number of epochs.

The second experiment fixes the number of convolutional layers and epochs then varies the batch size. This setup is shown below in Table 4.

Batch Size	Acc	Loss	Time
32			
64			

TABLE 4: Varying the batch size.

The third experiment fixes the number of epochs and batch size then varies the number of convolutional layers. This setup is shown below in Table 5.

Convolutional Layers	Acc	Loss	Time
1			
2			

TABLE 5: Varying the number of convolutional layers.

The motivation of this research question is similar to the previous research question: reducing the time required to classify images is crucial on an enterprise scale.

4 Dataset

An open-source dataset from Kaggle was used for the project. The dataset was provided by Navoneel Chakrabarty in his attempt to classify brain tumors using

machine learning techniques. The dataset contained 253 T1-weighted images which include 98 non-tumorous scans and 155 tumorous scans. All the images are saved in a .jpg format and contain 3 channels. Due to the small amount of data provided, data augmentation was performed to increase the number of images the model was trained on. Three other datasets were considered, however eliminated due to accessibility issues.

A summary of statistics was performed to determine what image pre-processing techniques were relevant. Table 6 below shows the mean, maximum, and minimum dimension of the images.

	Min Height	Mean Height	Max Height	Min Width	Mean Width	Max Width
Healthy	168	342.23	1080	150	343.16	1920
Tumorous	173	413.70	1427	178	361.24	1275

TABLE 6: Dimension of Images

Figure 6 and 7 respectively illustrate examples of the non-tumorous and tumorous images used.

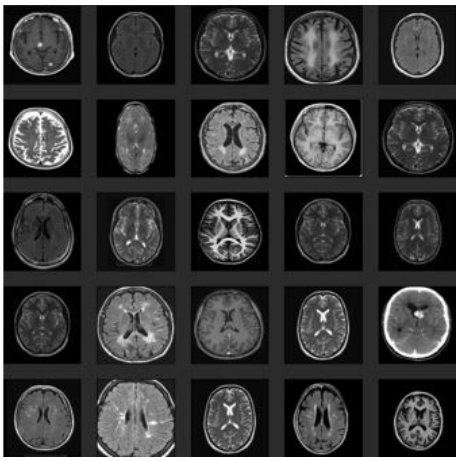

Fig. 6: Non-tumorous Set.

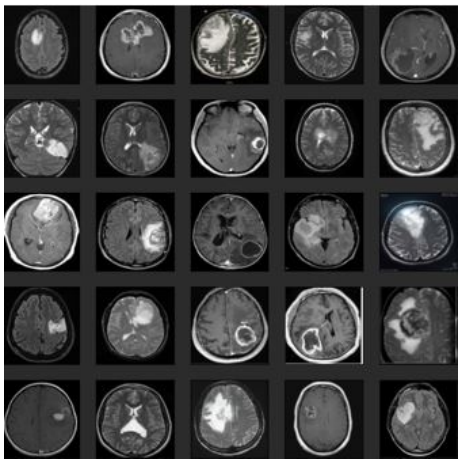

Fig. 7: Tumorous Set.

For the transfer learning section, the dataset used was an open source collection of chest CT-Scans also gathered from Kaggle. This dataset was divided into two classifications: healthy and tumorous (which is further divided into the type of cancer, but this project will not differentiate between them). The dataset has 147 unique healthy scans and 645 total unhealthy scans.

5 Experiments and Results

The approaches to the three research questions are shown below in each respective subsection. The results and experiment setups are also outlined below. The limitations to the research questions are general due to the limited data set. There are some questions that remain unanswered, such as whether each image is unique or if there are repeated scans from the same patient. This could introduce bias in the dataset. Moreover, there should be more data pre-processing to decrease the risk of over-fitting. More specifically, it is unknown the types of brain tumours present in the dataset and whether they are located in similar regions of the brain, so this must be investigated to ensure our model is not over-fitting to a specific type of brain tumour.

5.1 Research Question 1

The type of transfer learning technique to use was evident based on the available data and the model's current functionality. The data in the target domain is labeled, data in the source domain is labeled, and the task of classifying the presence of a tumor is related and can be learnt simultaneously. Hence, the inductive transfer technique employed was multi-task learning.

When combining and simultaneously training the chest scans along with the brain scans, the model had a reasonable performance. Using a batch size of 32 and with 10 epochs, the model managed to retain 96.4% model accuracy, 86.82% validation accuracy, a model loss of 0.1216 and a validation loss of 0.3654.

5.2 Research Question 2

The baseline approach for this research question was not to pre-process any of the input images, build and tune the model, and use its results as a comparison for the five combinations of image pre-processing approaches outlined in Table 1. The baseline approach showed a model accuracy of 86.7% and model loss of 0.3042, with a training time of 12 minutes and 59 seconds.

The results for this experiment are shown below, in Table 7. The evaluation methods used were to compare model and validation accuracy, as well as model and validation loss. the experiment showed that the combination of pre-processing steps that led to the fastest training time of 12 minutes and 37 seconds was cropping, then dilating the images. However, the training times for all the combinations of pre-processing steps were within two minutes of this fastest training time, with the longest training time at 14 minutes and 38 seconds.

Moreover, there was a significant difference in the accuracy of the model when images were pre-processed versus not

Pre-processing step(s)	Acc	Loss	Time
1. No pre-processing	Model: 86.7% Validation: 76.3%	Model: 0.3042 Validation: 0.6508	0:12:5.9
1. Cropping 2. Dilation	Model: 90.7% Validation: 83.6%	Model: 0.2064 Validation: 0.5987	0:12:37.5
1. Cropping 2. Dilation 3. Normalization	Model: 95.5% Validation: 86.2%	Model: 0.1442 Validation: 0.3361	0:14:38.5
1. Cropping 2. Dilation 3. Denoise	Model: 90.0% Validation: 79.7%	Model: 0.2312 Validation: 0.6396	0:13:53.3
1. Cropping 2. Dilation 3. Denoise 4. Normalization	Model: 92.7% Validation: 87.9%	Model: 0.2053 Validation: 0.3175	0:13:39.2
1. Cropping 2. Dilation 3. Normalization 4. Denoise	Model: 94.1% Validation: 87.5%	Model: 0.1549 Validation: 0.2906	0:14:08.3

TABLE 7: Results when varying the pre-processing steps.

pre-processed. The combination of steps that led to the highest model accuracy (95.5%) and lowest loss (0.1442) was to crop, dilate, then normalize the images. Model accuracy did not improve when images were denoised before (92.7%) or after (94.1%) normalization, contrary to previous beliefs.

For the second part of this research question, the pre-processing techniques were fixed and image dimensions were changed instead. the pre-processing techniques included cropping and normalizing the images. The results are outlined below in Table 8.

Image Dimensions	Acc	Loss	Time
(140, 140, 3)	Model: 86.6% Validation: 80.2%	Model: 0.3236 Validation: 0.4038	0:02:26.4
(240, 240, 3)	Model: 92.1% Validation: 87.1%	Model: 0.2146 Validation: 0.2828	0:12:17.6
(340, 340, 3)	Model: 94.5% Validation: 88.4%	Model: 0.1475 Validation: 0.3196	0:13:21.4
(440, 440, 3)	Model: 95.0% Validation: 87.5%	Model: 0.1461 Validation: 0.3433	0:42:5.4

TABLE 8: Results when varying image dimensions.

The largest input image dimensions of 440x440 pixels led to the highest model accuracy of 95.0% and the lowest model loss at 0.1461. Using the smallest image dimensions of 140x140 pixels led to the fastest training time of 2 minutes and 26 seconds. This was significantly shorter than the time taken to train the model with the largest input image dimensions at 42 minutes and 5 seconds.

5.3 Research Question 3

The approach for this method was to decrease the number of epochs during training, using one convolutional layer and a batch size of 32. Then, to evaluate the effect of batch size on the model's performance, two batch sizes were compared while using one convolutional layer and ten epochs during training. Finally, to evaluate the effect of the number of convolutional layers on the model's performance, the number of convolutional layers was varied while using four epochs and a batch size of 32. The results are shown below respectively in Table 9, Table 10, and Table 11.

Epochs	Acc	Loss	Time
5	Model: 87.2% Validation: 83.6%	Model: 0.3050 Validation: 0.3867	0:7:26.9
10	Model: 95.0% Validation: 88.4%	Model: 0.1475 Validation: 0.3196	0:13:21.4
15	Model: 98.9% Validation: 82.3%	Model: 0.0695 Validation: 0.5006	0:20:15.9
20	Model: 99.8% Validation: 88.8%	Model: 0.0224 Validation: 0.3900	0:27:49.5

TABLE 9: Results when varying number of epochs.

Batch Size	Acc	Loss	Time
32	Model: 93.4% Validation: 85.8%	Model: 0.1765 Validation: 0.3844	0:13:50.6
64	Model: 91.5% Validation: 83.6%	Model: 0.2293 Validation: 0.3803	0:19:50.6

TABLE 10: Results when varying the batch size.

Convolutional Layers	Acc	Loss	Time
1	Model: 83.2% Validation: 81.9%	Model: 0.3889 Validation: 0.4016	0:5:32.1
2	Model: 85.5% Validation: 66.8%	Model: 0.3353 Validation: 0.8161	0:39:38.8

TABLE 11: Results when varying number of convolutional layers.

Using 20 epochs with one convolutional layer and a batch size of 32 led to a model accuracy of 99.8% and model loss of 0.0224. At just 5 epochs, the model accuracy decreased by 12% to 87.2% and model loss increased to 0.3050. However, a major trade off is that the training time was 20 minutes longer using 20 epochs compared to using 5 epochs.

The difference between using a batch size of 32 and 64 (with one convolutional layer and 10 epochs) did not amount to a significant difference in model accuracy or loss. However, training time did increase by 50% when using a batch size of 64, compared to a batch size of 32.

Similarly, the difference between using one versus two convolutional layers (with 4 epochs and a batch size of 32) did not amount of a significant difference in model accuracy or loss. However, training time increased by nearly 400% when using two convolutional layers, compared to just one.

6 Discussion

The results of this project contribute to understanding the classification of tumorous brain scans with deep learning. However, it is important to acknowledge the limitations of

this project as well. The limitations are general to all the research questions mainly due to the limited dataset and resources available. There are some questions about the image dataset that remain unanswered, such as whether each image comes from a unique patient or if there are repeated scans. This could introduce bias in the dataset. Moreover, there should be more data exploration to decrease the risk of over-fitting. Specifically, the types of brain tumours present in the dataset remain unknown, and it is unclear whether they are located in similar regions of the brain. This must be investigated to ensure the models are not over-fitting to a specific type of brain tumour. Furthermore, with more computational resources, more combinations of different epochs, batch sizes, etc. could have been used.

The results do however, contribute to a few conclusions. Firstly, it was found that cropping, dilating, and then normalizing the input images was the best combination of pre-processing steps in terms of model accuracy and loss. Cropping and dilating however was the best combination to minimize training time, while only decreasing model accuracy by 5% compared to the best combination (for model accuracy). Therefore, denoising input images may be a redundant pre-processing step for this task. Moreover, cropping the images to smaller image dimensions drastically reduced training times but compromised model accuracy. This should be an important design consideration when implementing such deep learning algorithms in practical applications.

Moreover, it was found that there is no clear advantage to increasing the number of convolutional layers with the dataset that was used. However, this could be due to the limitations of the dataset, as discussed above. Therefore, these research questions should be investigated with a diverse and larger dataset to confirm these findings.

7 Group Member Contributions

Name	Contribution
Madison Boem	Report: Related Work, Conclusion and Future Work Project: Research, Dataset Search
Caroline Kim	Experiment and Results, Discussion Project: Research, Dataset Search
Eric Venditti	Report: Methodology, Dataset, Experiment and Results Project: Coding, Running Scripts
Jade Watson	Report: Abstract, Introduction, Methodology, Dataset Project: Coding, Running Scripts

TABLE 12: Group Member Contributions

8 Replication Package

Please see attached zip folder submitted on OnQ.

9 Conclusion and Future Work

The proceeding discussion indicates the project limitations, and managing these would enable further success. Next steps would include this mitigation, collection of larger set of images, managing issues with compute resources, and further tuning of hyper-parameters. In accordance with related works by Rabsemen et al. [4] future steps could also include the stratification of data by tumor grade to seek higher accuracy. Among other things these steps could contribute to further success.

Convolutional neural networks remain a standard method in medical research of automated image segmentation and image classification. Currently, the annual Brain Tumor Segmentation (BraTS) Challenge propels this research and improves industry standards each year. The next steps in this project, which includes mitigating limitations, will be conducted in relation to this challenge. Utilizing the 2022 BraTS dataset would be beneficial to the project as it would allow for a more ideal comparison between the model of this project and others throughout the industry.

Models are not yet used in a clinical setting, but continued research and advancements brings the prospects of eventual use in tumor diagnosis and detection in patients. Manual detection by radiologists is time consuming and resource heavy, and early detection is advantageous for patient prognosis. This work can verify the benefits of the use of convolutional neural networks in brain tumor detection, and contribute to research on such to help medical professionals work with the best possible solutions.

References

[1] M. S. Ali Isin, Cem Direkoglu. Review of mri-based brain tumor image segmentation using deep learning methods. In *2th International Conference on Application of Fuzzy Systems and Soft Computing, ICAFS 2016, 29-30 August 2016, Vienna, Austria*. IEEE, 2016.

[2] E. H. A. Z. M. N. B. N. J. P. M. T. S.-L. L. Artemis Zavaliangos-Petropulu, Meral A. Tubi. Testing a convolutional neural network-based hippocampalsegmentation method in a stroke population. IEEE, 2020.

[3] J. Brownlee. A gentle introduction to pooling layers for convolutional neural networks. In *Machine Learning Mastery Deep Learning for Computer Vision*. IEEE, 2019.

[4] M. R. R. W. R. M. R. M. Michael Rebsamen, Urspeter Knecht. Divide and conquer: Stratifying training data by tumor grade improves deep learning-based brain tumor segmentation. In *frontiers in Neuroscience*. IEEE, 2019.

[5] C. Net. Brain tumor: Statistics. In *Brain Tumor Statistics*. IEEE, 2022.

[6] P. G. F. I. A. S. I. M. M. K. S. B. S. A. I. Tahia Tazin, Sraboni Sarker and H. Alshazly. A robust and novel approach for brain tumor classifcation using convolutional neural networks. In *Hindawi*. IEEE, 2021.