Brain Tumour Classification Using Machine Learning

ECE 470 Project Report

Andrew Rose	::	V00884894
Braidon Joe	::	V00822287
Ethan Janus	::	V00855202

University of Victoria
Victoria, BC

Table of Contents

Table of Figures

Abstract

In order to treat patients with brain tumours, accurate detection and classification of the tumour from MRI scans is vital so that medical professionals can administer proper treatment. In this paper, a machine learning model called a convolutional neural network (CNN) is described that classifies brain tumours into three different types.

Convolutional neural networks are well suited for image processing and the CNN in this paper is based on the LeNet-5 architecture. Images went through standard preprocessing before the model classified them. The optimal hyperparameters for the model were determined using a bespoke implementation of grid search designed to reduce computation time.

ReLU was determined to be the best activation function, and 2x2 was determined to be the best kernel size. The final training accuracy was 99.8% with a training loss of 0.011. The CNN achieved a tumour classification accuracy of 75.1% and loss of 5.84 on the testing dataset. The results were far better than random guessing, but further work could be done to increase the testing accuracy. The model overfitted the training data, so techniques to reduce overfitting such as dropout layers and data augmentation could be employed to increase the final testing accuracy.

Note: the grid search results described in this report are different from those described in the team’s slideshow. This is because, after the slideshow was presented, the training procedure was re-run with a different loss function. It is the results of that re-run that this report is based on, not the old results.

1. Introduction

Background

Proper detection and identification of brain tumours is the first step in treating patients with those tumours. Three types of brain tumours were analyzed by the machine learning model described in this paper: gliomas, meningiomas, and pituitary tumours.

A glioma is a brain tumour that begins in the glial cells of the brain or the spine [1]. Approximately 80% of malignant brain tumours are gliomas [1], and early detection is paramount to the patient's life. A meningioma occurs in the meninges, which are the membranous layers that surround the brain and spinal cord [1]. As such, meningiomas are typically located on the outer edge of the brain. Pituitary tumours occur in the pituitary gland, which is located at the base of the brain when looking at a side view MRI scan [1].

To distinguish brains with tumours from those without, and also to distinguish between different tumour types, a machine learning technique is needed that can process images with ease. Convolutional neural networks were the obvious choice, as their ability to detect patterns transposed to different positions within images makes them well-suited to image processing.

The LeNet-5 convolutional neural network architecture in Figure 1 was originally used to distinguish between 32x32 images containing handwritten digits [2]. Originally, it generated ten probability values corresponding to its estimated probabilities of each input image containing each of the ten digits from 0 to 9. Surprisingly, brain tumours can be handled by a similar architecture: instead of digits, the system classifies based upon tumour types.

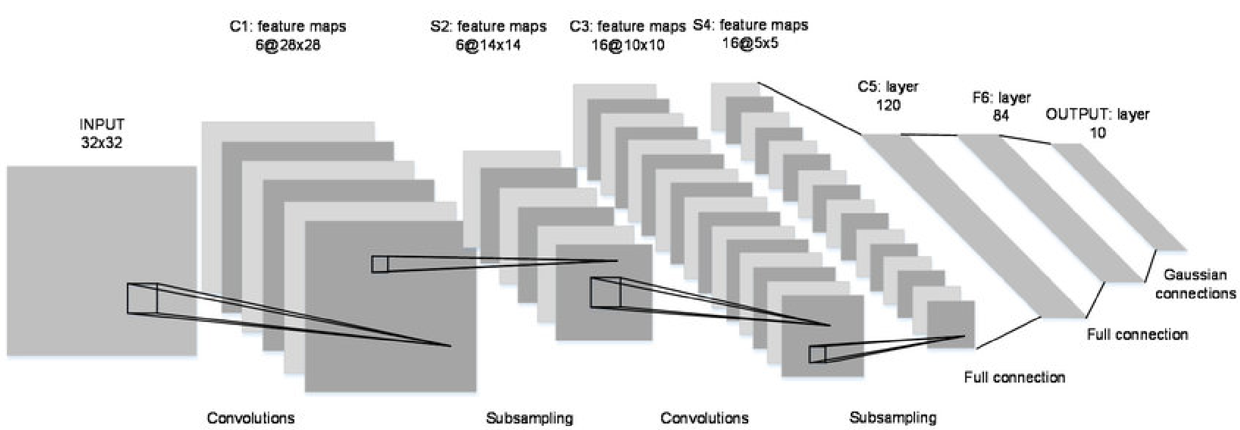

Figure 1: The classic LeNet-5 architecture. [3]

This project also implements grid search, a technique which runs several iterations of the same basic model architecture instantiated with different hyperparameters. The technique is so named because the ranges of multiple parameters form a "grid" of possible combinations so the parameters with the best performance can be selected and used.

Motivation

Accurate detection and classification of brain tumours allows medical professionals to properly diagnose and treat them, but it's a difficult and time-consuming task for humans. A good machine learning model can process these images more accurately and more quickly.

Applying LeNet-5 to tumour classification requires only two modifications:

1. Increase the number of input nodes to work with higher-resolution images (since the number of input nodes equals the number of pixels in the input image).
2. Decrease the number of output nodes from ten to four, representing the three types of tumours in the data plus one node for the images without tumours.

With those tweaks made, the system can process tumour photographs, but the team made further optimizations to improve its performance. Central to those optimizations is the team's implementation of grid search, as discussed in detail in the Methodology section.

2. Related Work

Three research papers were investigated in order to better understand the field of identifying brain tumours with artificial intelligence and machine learning.

In "Brain Tumor Detection Using Shape features and Machine Learning Algorithms" [4], brain tumours were classified using a multi-layer perceptron and a C4.5 decision tree. The process began with preprocessing the MRI images using a sigma filter to remove noise. Next, adaptive thresholding occurred, in which pixels above a certain threshold were brought to the foreground and those below it were pushed to the background. This separated each tumour from the rest of its image. Region detection was then applied, where the tumours were sectioned off and made into its own image. Features were then extracted, which included Major Axis Length, Euler Number, Minor Axis Length, Solidity, Area, and Circularity. Finally, these features were used as input into the MLP and C4.5 algorithms for classification. MLP proved more effective than C4.5.

A neural network was used in "Brain Tumour Detection using Unsupervised Learning based Neural Network" [5]. Here, preprocessing included several steps. First, edge detection took place to determine where the brightness in an image changed sharply between two regions of the image. Histogram equalization was applied to improve images' contrast. Thresholding was applied, segmenting off regions of pixels which surpassed thresholds. Then, features were extracted, including contrast, correlation, energy, and homogeneity, using Independent Component Analysis (ICA). ICA, a type of dimensionality reduction, separates a signal with multiple variables into its additive subcomponents. For classification, an unsupervised technique called Self-Organizing Map (SOM) was applied. In the artificial neural network, newborn outputs compete with each other to be activated.

Finally, a convolutional neural network was used in "A deep learning model integrating FCNNs and CRFs for brain tumor segmentation" [6]. This paper's process included training Fully

Convolutional Neural Networks (FCNNs) with image patches, training Conditional Random Fields (CRFs) as Recurrent Neural Networks using image slices with parameters of FCNN's fixed, and fine-tuning the FCNNs and CRF-RNNs using image slices. Preprocessing involved determining the histogram gray values. Image slices were segmented using deep learning models with integrated FCNNs and CRF-RNNs from axial (rear), coronal (top), and sagittal (side) views of images, respectively. Results were obtained via a voting-based fusion strategy on the three different views. The resulting model could then segment brain tumours slice-by-slice, which is faster than using image patches.

It has been widely accepted for many years that convolutional neural networks are the best option for classifying images. This conclusion can be traced to a landmark paper which described the LeNet-5 network upon which this ECE 470 project is based [2].

When the original LeNet-5 paper was published, recent technological progress had enabled data scientists to abandon hybrid systems - which incorporated human-defined features to reduce computation complexity - in favour of neural networks which define their own features by finding subtly useful patterns humans could not discern. This paper demonstrated that with the requisite computing power available, convolutional neural networks yielded better results than every other model. By 1998, LeNet-5 had been deployed in banks across America, where it was reading "several million checks per day". [2]

Since 1998, computers have become far more powerful. Accordingly, neural networks can now process higher-resolution images using more advanced techniques such as grid search. This is what the team did, with the intent of optimizing the LeNet-5 algorithm for this new purpose.

3. Problem Formulation

The first problem in brain tumour classification is preprocessing the data. In this case, the raw images vary in size and were taken from 3 different angles (top, side, and rear of the subject's head). To resolve the size issue, the images can be cropped and scaled with only a minimal loss of data from the original image.

The second problem is selecting an appropriate machine learning algorithm for the data. In this case, the task involved classifying the data rather than predicting a value, so a multiclass classification algorithm was needed. Additionally, the algorithm had to be well-suited to image processing. Accordingly, it made sense to use a convolutional neural network with a final densely-connected layer with a softmax activation function, which is able to generate the probability of an input belonging to each of several classes.

4. Methodology

The input dataset for the model came in the form of JPG images of MRI scans of human brains [7]. The scans were variously taken from three angles: sagittal (side), coronal (top), and axial (rear) views of the brain are all present. Multiple views are useful for medical professionals since they give a better picture as to what the tumour actually looks like. For example, a tumour may be only properly visible at a certain angle. However, the variety of angles makes modelling more difficult. Training a model on only one view type would be easier, in terms of feature recognition in the CNN. The images were also formatted in colour - though the images did not actually have any meaningful colour content, appearing to human eyes to be in grayscale - and came in various heights and widths. Therefore, preprocessing was required to convert them to grayscale and normalize their dimensions before feeding the images into the CNN.

The data was divided into training and testing subdirectories. There were 2870 images used in training and 394 images used in testing. 10% of images in the training set were used for validation, so 2583 images from the testing folder were used for training the model, while the remaining 287 were used for validation. Within the testing and training folders, the images are further divided into four categories: Glioma Tumours, Meningioma Tumours, Pituitary Tumours, and No Tumour.

Preprocessing the raw images was required to prepare the training, validation, and testing datasets for efficient use in the CNN. The majority of the preprocessing was achieved through TensorFlow's tf.keras.preprocessing.image_dataset_from_directory() method. After loading in the file path to the images, this method was able to create labels for the 4 categories based on the names of the subdirectories. The categories were glioma_tumour, meningioma_tumour, pituitary_tumour, and no_tumour. The label mode selected was "categorical", since the labels have no particular order. The color_mode was set to grayscale, which allows the CNN to analyze one value for each pixel instead of 3 RGB values. Image resolution was set to 128x128, as this was found to increase the speed of the neural network without meaningfully impacting its ability to recognize features. Smart_resize was selected to automatically crop and resize the images to the 128x128 format without changing their original aspect ratios.

The final preprocessing step converted the pixel values of the images from integer values between 0 and 255 to floating point values between 0 and 1. This was achieved through the layers.experimental.preprocessing.Rescaling layer in TensorFlow. This was done to the training, validation, and testing datasets to ensure that all images were normalized for input into the CNN. Examples of the images after preprocessing are given in Figure 2.

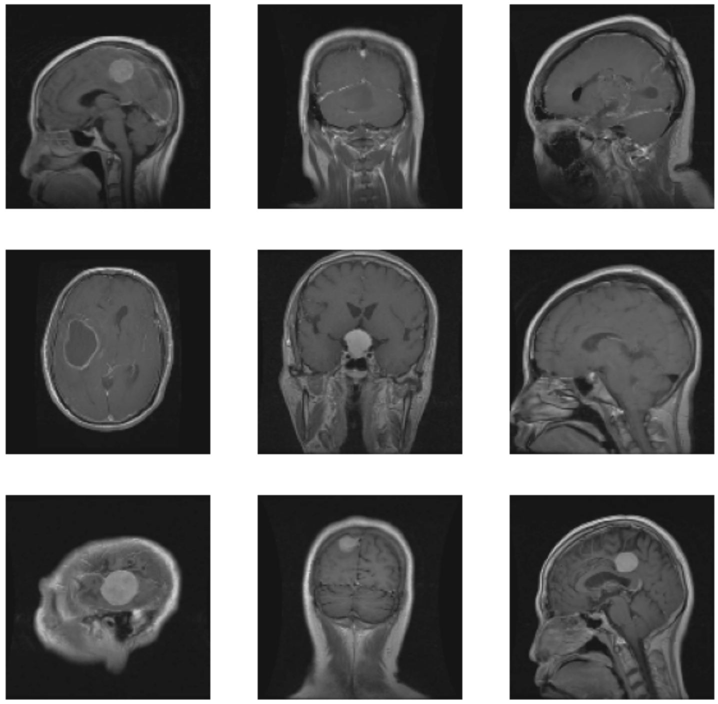

Figure 2: Sample brain scan images.

With the basic LeNet-5 architecture selected as a template for the team's final network, and with preprocessing done, the question remained of how best to optimize the architecture for this application. Educated guesses about the best hyperparameter values would have been unlikely to yield the best results, so grid search was selected as a means of finding the best values. But grid search, in its plainest, most thorough form, would have taken too much computation to iterate through for this project. Therefore, the team implemented a few ways to optimize the grid search algorithm for faster completion.

First, it was noticed that some activation functions were simply not applicable to the scenario. To flush out all the unworkable activation functions, a single-epoch model was created for each activation function, and the ones that had significantly lower accuracy than the rest had the model with the corresponding activation functions discarded.

To further reduce the amount of computation required, a check was implemented: to rank the performance of a model, the program looked at the loss value of the model. The loss value represents the total error between the predicted output values and the actual output values and thus would have to be minimized. The check that was implemented was based on the observation that the loss value is always reduced by a smaller and smaller value after each successive training iteration. The check was designed to see if it was theoretically possible for a given model to leapfrog the best model after the next training iteration. If a model was not capable of this under any circumstances, it was discarded. The condition the algorithm uses to check whether a selected model is worth pursuing is as follows.

min_loss_n < selected_loss_n - delta_selected_loss_n

where:
min_loss_n is the minimum loss value across all models after n iterations;
selected_loss_n is the selected model's loss value after n iterations;
delta_selected_loss_n is the amount by which the selected model's loss decreased after the latest iteration.

If the inequality is true, then the selected model is discarded. This check was not designed to remove models that could not possibly be the best model with total certainty, but rather to remove models that were very unlikely to be the optimal model. This reduced training time from a projected 12+ hours to approximately 4.5 hours.

That left the question of which values to iterate over. Since the computation time required for grid search grows exponentially with the number of parameters being searched over, the team decided to restrict its search to just two parameters for the purposes of this project. Candidates for these values initially included:

- The size of each convolution kernel [selected for final model]
- The activation function used for the convolution layers and the first densely connected layer [selected for final model]
- The number of filters in each convolutional layer
- The number of nodes in the first densely connected layer
- The pool size and stride of the pooling layers

The numbers of filters and densely-connected nodes were ruled out as grid search candidates for the following reasons:

- Testing multiple different increases to these numbers - even if replaced as a set instead of treating each individual filter bank's size as its own grid search parameter - would massively increase processing time, especially given that exponentially larger numbers would likely be required to cause significant improvements.
- The results would not likely be very interesting; they'd probably just improve performance (if paired with suitable anti-overfitting measures, another factor introducing further complexity that provides another reason not to use these hyperparameters) - the interesting thing would be to see where the improvements tapered off, which would require fine-resolution adjustments to the numbers of filters and nodes around higher values that the team was not equipped to handle.

Changing the pool sizes and stride sizes of the pooling layers was rejected on the grounds that:

- The pool sizes cannot be reduced beyond their original size of 2x2.
- Increasing pooling size and thus reducing the resolution of imagery the network handles internally, when the point is to classify more complex images than the original LeNet-5 network was built to handle, would be counterproductive.

The team was then left with the size of the convolution kernels and the activation function used throughout the network (with the exception of the final layer, which must retain its original softmax activation function to output useful values).

The original LeNet-5 architecture used 5x5 kernels and tanh activation functions. The model was ultimately set up to test:

- Every square kernel size between 2x2 and 18x18, inclusive.
- All available Keras activation functions that were proven usable in preliminary testing.

5. Results and Discussions

After ReLU was determined to be the optimal activation function, the performance of the CNN over 10 epochs was plotted.

Training and validation accuracy were plotted over ten epochs (Figure 3), as were training and validation loss (Figure 4):

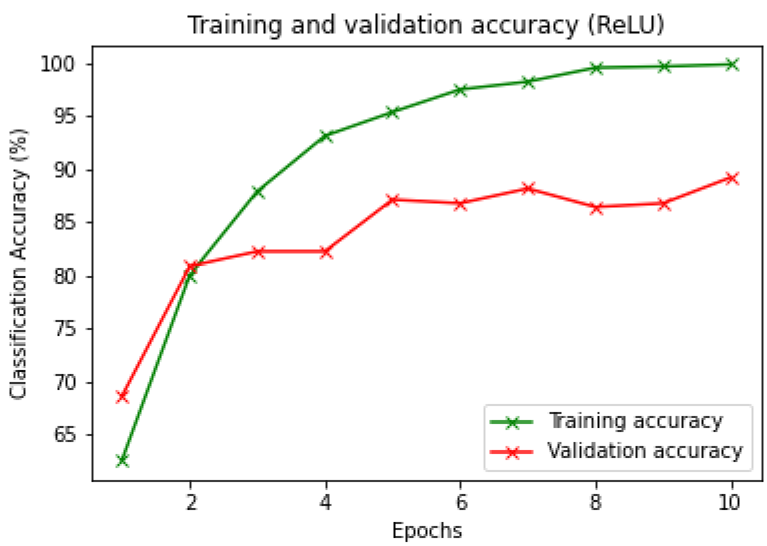

Figure 3: Training and validation accuracy.

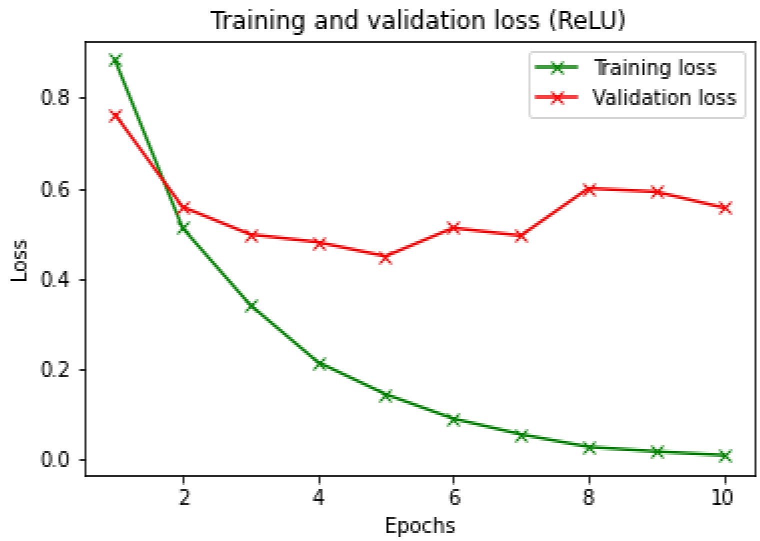

Figure 4: Training and validation loss.

From Figures 3 and 4, it can be seen that no meaningful improvement in validation accuracy occurred after five epochs of training. Additionally, after those five epochs, validation loss increased, hovering thereafter between 0.5 and 0.6. This all suggests that more than five epochs of training served only to overfit the model to the training data.

After completing the full 10 epochs, testing accuracy reached 75.1% and testing loss reached 5.84. Meanwhile, training accuracy reached 99.85%, with loss of 0.01055. Such a large gap between training and testing performance confirms that the model is severely overfitting.

The Future Work section describes further methods that could be implemented to reduce the overfitting and close the gap between the training and validation accuracy. These methods include data augmentation, grid-searching over filter counts and the number of densely-connected nodes in the second-last layer, trying different loss functions, and adding dropout layers.

After the training process was completed, the optimal activation function was found to be the rectified linear unit function (ReLU) and the optimal kernel size was found to be 2.

ReLU is one of a few similar activation functions that are all of the linear unit type. They all follow the same general shape:

$$f(x) = 0 \mid x \leq 0 \text{ and } f(x) = x \mid x > 0$$

and in the case of ReLU, that is the exact definition. (The others vary somewhat to the left of the y-axis.)

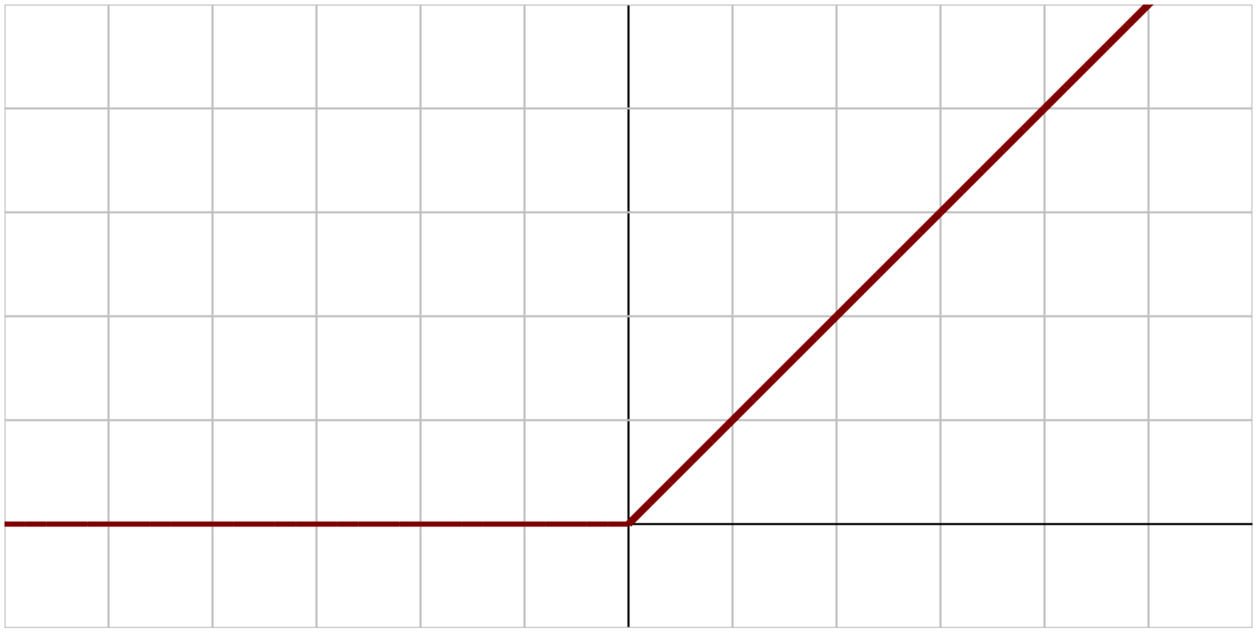

Figure 5: Plot of ReLU activation function.

The kernel size that gave the best results was 2x2. This meant that the model performed best when it was looking for sharp changes in contrast in the image rather than some sort of more general pattern. This is a rather counterintuitive result, since humans would most likely look at the whole image and try to find identifiable features in the tumour if there is one present.

After the testing process finished, the model had testing loss of 5.84 and testing accuracy of 75.1%. These results would need to be improved to be used in a practical context, where errors could kill medical patients, but the team still considers the project a success overall.

6. Conclusion

A convolutional neural network, based on LeNet-5 with custom grid search, was developed to classify brain tumours in MRI scans. The grid search selected the ReLU activation function and a kernel seize of 2x2. With these hyperparameters, the model achieved training accuracy of 99.8% and loss of 0.011. On the testing dataset, the model achieved accuracy of 75.1% and loss of 5.84. Further improvements are described in the Future Work section.

7. Future Work

The most important step to take to improve the model is to divide up the dataset by viewing angle. The data that the model was trained on was a random assortment of top, rear, and side views of people's heads. This made the network's task far more difficult, because it had to correctly identify tumours across three classes of images with different characteristics. Given this jumbled dataset, it's astonishing that the network reached 75% test accuracy. Nonetheless, there are many ways to deal with this issue, including:

1. Divide the existing data into three subsets (one for each viewing angle) and separately train one network per viewing angle. This is the best approach in the event that it is impossible to obtain matched trios of views for each head, and it would not require any modification to the team's existing LeNet-5-based network.
2. Obtain data which consists of labels matched with complete trios of MRI views. This would require reconsideration of the overall design of the network, since there are multiple ways to handle trios of photos. It might be best to train three networks and sum their probability estimates, possibly with the addition of weighting based on the three networks' relative accuracy, or to run all three photos through a single network.

It also could be best to run some preliminary testing to evaluate which of those approaches (and possibly some other, additional approaches) performs best. Detailed investigation of these possibilities is beyond the scope of this report.

The next step is addressing the network's overfitting problem. Possible solutions to this issue include implementing dropout layers, a loss improvement threshold beyond which no more epochs are begun, and data augmentation. Data augmentation involves randomly modifying the training images by performing mirror flips, rotations, and other changes which essentially create additional training data without having to gather it.

Next, the team could add the numbers of filters and densely-connected nodes in the network to the grid search. It would take a lot of computing power to get this functional. Even if the three convolutional layers' filter counts are iterated through as one variable containing all three counts, two new dimensions are being added to the grid search, exponentially increasing training time. If the three filter counts are iterated through as separate variables, then *four* new dimensions are being added, which would massively increase training time. There's also the matter of determining what values to test, and what the relative values of the filter counts should be in the event that they are iterated through as a set instead of individually.

However, the team's timesaving tricks are present to cut down on the amount of extra training time by eliminating obviously-inferior solutions. Testing would be necessary to ascertain exactly how much more training time these new hyperparameters require, given the presence of the time-saving techniques. It may also be possible to reuse the ReLU function, which was previously determined to perform best for this task, without iterating over the available activation functions again, thus removing one dimension from the grid search. If the 2x2 optimal kernel size is reused, too, then only two variables need to be iterated through.

Finally, the team could explore different loss functions. The choice of categorical cross entropy was deliberate, since it is optimal for nominal data (as opposed to ordinal data) and since the four brain image types cannot be logically ordered, but there are other options. New loss functions could be iterated through with grid search, or a new one could be manually selected.

References

[1] *“Brain Tumors: Types of Brain Tumors”,* American Association of Neurological Surgeons. Accessed on July 26, 2021. [Online]. Available: https://www.aans.org/en/Patients/Neurosurgical-Conditions-and-Treatments/Brain-Tumors

[2] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791. http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

[3] Tra, Viet & Kim, Jaeyoung & Khan, Sheraz & Kim, Jongmyon, “Bearing Fault Diagnosis under Variable Speed Using Convolutional Neural Networks and the Stochastic Diagonal Levenberg-Marquardt Algorithm”, in Sensors, vol. 17, no. 12, Dec. 2017, doi: 10.3390/s17122834.

[4] D. George, H. Jehlol, and A. Oleiwi, "Brain Tumor Detection Using Shape features and Machine Learning Algorithms", International Journal of Scientific & Engineering Research, Volume 6, Issue 12, 2015. https://www.researchgate.net/profile/Dena-Nader/publication/330292229_Brain_Tumor_Detection_Using_Shape_features_and_Machine_Learning_Algorithms/links/5c377e2ca6fdccd6b5a19d42/Brain-Tumor-Detection-Using-Shape-features-and-Machine-Learning-Algorithms.pdf

[5] S. Goswami and L. K. P. Bhaiya, "Brain Tumour Detection Using Unsupervised Learning Based Neural Network," 2013 International Conference on Communication Systems and Network Technologies, 2013, pp. 573-577, doi: 10.1109/CSNT.2013.123. https://ieeexplore.ieee.org/abstract/document/6524466

[6] X. Zhao, Y. Wu, G. Song, Z. Li, Y. Zhang, and Y. Fan, “A deep learning model integrating FCNNs and CRFs for brain tumor segmentation,” Medical Image Analysis, vol. 43, pp. 98–111, Jan. 2018, doi: 10.1016/j.media.2017.10.002. https://www.sciencedirect.com/science/article/abs/pii/S136184151730141X?via%3Dihub#

[7] Sartaj Bhuvaji, Ankita Kadam, Prajakta Bhumkar, Sameer Dedge, and Swati Kanchan, “Brain Tumor Classification (MRI).” Kaggle, 2020, doi: 10.34740/KAGGLE/DSV/1183165. https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri

[8] “Activation function,” Wikipedia, 04-Jun-2021. [Online]. Available: https://en.wikipedia.org/wiki/Activation_function#/media/File:Activation_rectified_linear.svg. [Accessed: 27-Jul-2021].