

National College of Ireland

Project Submission Sheet – 2022

Student Name: Juan Pablo Andres Palm Bustos, Kalpesh J. Dhande,
Rahul S. Vaydande, Tsai Shih Yang
.....

Student ID: 20225776, 20185821,
20181663, 21101825
.....

Programme: MSCDAD_B **Year:** 2022
.....

Module: Data Mining And Machine Learning -2
.....

Lecturer: Anu Sahni
.....

Submission Due Date: 09-05-2022
.....

Project Title: Brain Tumor Classification using CNN
.....

Word Count:3,588.....

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the references section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature: Juan Pablo Andres Palm Bustos, Kalpesh J. Dhande,
Rahul S. Vaydande, Tsai Shih Yang
.....

Date: 09-05-2022
.....

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. Projects should be submitted to your Programme Coordinator.
3. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
4. You must ensure that all projects are submitted to your Programme Coordinator on or before the required submission date. **Late submissions will incur penalties.**
5. All projects must be submitted and passed in order to successfully complete the year. **Any project/assignment not submitted will be marked as a fail.**

Office Use Only
Signature:
Date:
Penalty Applied (if applicable):

Brain Tumour Image Classification using CNN

1st Rahul S. Vaydande
Msc. In Data Analytics
National College of Ireland
Dublin, Ireland
x20181663@student.ncirl.ie

2nd Juan Pablo Andres Palma Bustos
Msc. In Data Analytics
National College of Ireland
Dublin, Ireland
x20225776@student.ncirl.ie

3rd Kalpesh J. Dhande
Msc. In Data Analytics
National College of Ireland
Dublin, Ireland
x20185821@student.ncirl.ie

4th Tsai Shih Yang
Msc. In Data Analytics
National College of Ireland
Dublin, Ireland
x21101825@student.ncirl.ie

Abstract—A brain tumor is the growth of abnormal cells or masses in the brain. The most common method for detecting a brain tumor is to examine an MRI of the brain. During an MRI, a large number of pictures are generated. According to certain research, manually recognizing brain tumors has a higher mistake rate than brain tumors discovered using deep learning classification. Detecting brain tumors and their varieties is a challenging task in medical image processing. Deep learning and other cutting-edge image processing technologies can be utilized to solve these problems. One of the most effective ways of identifying photos is the convolution neural network. we are using a convolution neural network to classify the brain images in this study. We will categorize the brain scans as having no tumor or glioma, meningioma, or pituitary tumors. We will be utilizing the Brain Tumour Classification dataset from Kaggle for this project. The confusion matrix, accuracy, f1 score, recall score, and precision will all be used to evaluate this constructed model.

Index Terms—component, formatting, style, styling, insert

I. INTRODUCTION

Brain tumour is one of the deadliest cancer. The brain is an essential component of any living being. Every action taken by a living being is controlled by the brain. This means that the Tumour can influence the actions of the human body. Also, because it is the most lethal, detecting it early and accurately is always preferable. Also, if we can detect brain tumors in their early stages, we can provide proper treatment, potentially saving lives. A brain tumor is the abnormal growth of a mass or cell in the brain. There are numerous types of brain tumors. Brain tumors are classified into two types: benign (non-cancerous) and malignant (cancerous). Furthermore, these brain tumors are classified as glioma, meningioma, or pituitary tumors. According to some articles, approximately 20,000 people died in the United States due to brain tumors last year, and approximately two lakh fifty thousand people died worldwide due to brain tumors.

MRI is the most commonly used technique for detecting a brain tumor. The images generated by the MRI must then be manually analyzed in order to detect brain tumors. This may take some time because brain tumors must be manually detected by an expert doctor. Furthermore, the accuracy of manually detecting a brain tumor is dependent on the doctor's expertise and experience. Manual detection has a high risk of error. This procedure must be automated. The image processing and deep learning methods can be used to automate this process.

Deep learning algorithms, such as Convolution neural networks, are having a positive impact in the medical field by detecting diseases in their early stages. We MRI images in manual detection of brain tumors. We can use these images to train the deep learning model and automate the process of detecting brain tumors. There are numerous algorithms available for image classification. Convolution neural networks are one of the best algorithms in image classification as they have a tendency to extract features from images. Furthermore, recent studies show that CNN algorithms perform nearly as well as humans in detecting brain tumors.

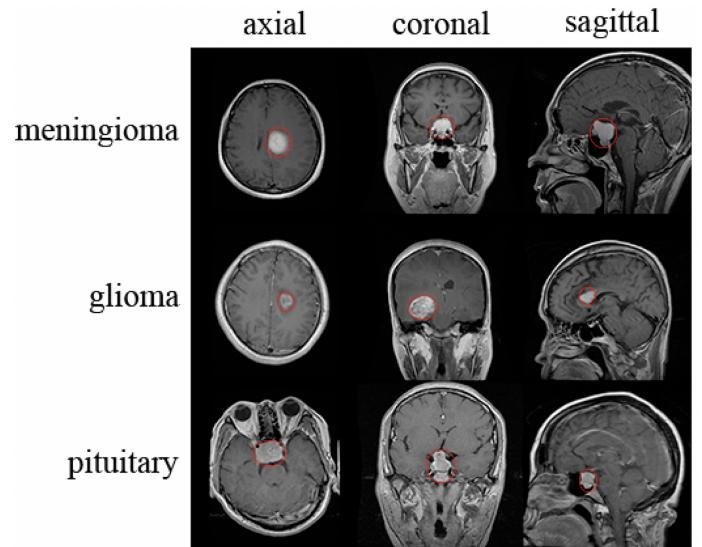


Fig. 1. Brain tumor types

In this project, we will build a model using a convolution neural network to classify brain images as having no brain tumors or having glioma, meningioma, or pituitary tumors. In this project, we will be using brain image classification data from Kaggle. The images were then classified using convolution neural network layers.

The remainder of this paper is organized as section two, which includes the related work that we have completed. Section three contains the methodology that we have used. The fourth section is about putting our model into action. The fifth section contains the results of the brain tumor classification

model that we have developed, and the sixth section contains the conclusion of our work as well as work that can be implemented in the future to improve the model for the brain tumour classification.

II. RELATED WORK

Medical practice has changed significantly in recent years. Technology and digital data have an impact on doctors and patients. For example, people can look for information on the web or have an online consultation, and doctors can have complete historical data of patients before consulting. All these changes have affected the doctor-patient interaction but also how doctors work.

On the other hand, the accumulated information generated in hospitals stored electronically in medical records has provided a fertile field for Artificial Intelligence and Machine Learning appliances. This application has changed the way both doctors and researchers approach medical problem-solving.

A. Deep Learning in Medical Imaging

Recent researchers have critically evaluated how Medical Image Analysis has been developed for solving medical problems. The main conclusion is the convenience of Convolutional Neuronal Networks for many researchers [1] [2]. However, They also warn how dataset and overfitting are still the main issues for deep learning models, model complexity, and training time [1].

The problems in the accuracy of deep learning in medical imaging using a meta-analysis technique were exposed by Aggarwal et al. [3]. They found a high heterogeneity between studies and methodologies, terminology and outcomes. Sometimes with performance over 96% accuracy even when it has not been widely implemented in clinical practice. First of all, difficult to have large amounts of data when the illness is rare. Secondly, the need for explanation, the doctor's diagnosis of the disease is mainly through some specific characteristic of the disease in imaging [4].

Furthermore, the powerful conclusion is the evidence for the overestimation of the diagnostic accuracy of Deep Learning algorithms in medical imaging and how important it is to consider developing guidelines for standards in this field. For instance, quantifiable benefits, regulatory pathways that are lengthy and costly, patient safety considerations, data privacy, and economic factors like who paid for AI tools [5].

Moreover, [2] explains that some organisations collect databases. Still, no data covers the complete heterogeneity of collection protocols or differences between disciplines affecting the constant evolution of image capture systems limits the construction of large databases with similar characteristics, further exacerbating the problem of data availability that fits the actual context.

B. Deep Learning methods in Brain Tumor Detection: Convolution Neural Network

There is no rule for what methods may be used for Brain Tumor detection. Many studies have tried different

techniques such as Convolutional Neural networks, Support Vector Machine or others Machine and Deep Learning methods. However, the state-of-art shows that Convolution Neural networks (CNN) perform consistently. For example, using figshare pictures collection, which comprises around 300,000 photos of brain tumours classified as Meningioma, Glioma, and Pituitary, the model of CNN had greater accuracy than the machine learning technique, which was around 95% [6].

Baranwal et al. used a Support Vector Machine and a Convolution Neural Network to categorise brain tumours into three subtypes: Glioma, Meningioma, and Pituitary. The authors added some salt noise and shrunk the photos to minimise processing and increase the dataset, and CNN outperformed SVM [7].

On the other hand, authors have used combined methods of building a system that uses a deep convolutional neural network and a support vector machine to categorise different types of brain tumours [8]. They used Convolution neural network models like GoogleNet and ResNet-18 to extract the features and Support Vector Machine was used to classify the features derived from these models. In this approach, they first employed image min max-normalisation to normalise the pictures, then applied the data augmentation strategy to decrease the imbalanced class problem. The best accuracy they have obtained is 98 per cent using a combination of ResNet-18 and SVM.

Also, In [9] the author made uses the CNN model to extract key features from the images and further classify the images into a "Tumor Detected" and "Tumor Not Detected" categories. The three-layer CNN architecture linked to a fully connected neural network produced accurate results from MRI scans. The output reached an accuracy of 96.05% and an error rate of 0.029. This study was built using Keras, which Google's TensorFlow in the backend supports.

In [10]. This study provides a technique for identifying the kind of brain tumour by employing a deep learning model. For this classification, they employed the AlexNet model combined with the RPN region proposal network, which is based on Faster R-CNN. The dataset they are working with has been cleaned using the camshaft algorithm. This method aids in extending the picture window size and altering the orientation of particular images. They went through many steps in this. The first was to employ the AlexNet model as the basic model for image classification. Then they utilised a fully convolutional network (FCN) with three Conv layers and one proposal layer to create an area of interest (ROI). They then employed maximum suppression (NMS) to choose the high confidence zone while discarding the rest of the region of interest. The suggested approach showed a promising result compared to the image segmentation of a brain tumour.

Furthermore, In this study [11], the author employed a concatenation strategy for the deep learning model and the likelihood of having brain tumour research. Pre-trained deep learning models DenseNet201 and Inception - v3 were employed to identify and categorise brain cancers. The pre-trained model Inception - v3 extracted features, then concatenated

for tumour categorisation. A SoftMax classifier was then used to do the classification. The suggested approach yielded 99.34 per cent and 99.51 per cent accuracies, respectively. Rehman and his team in this research [12]. They established a framework for tumour classification that employed a configuration termed tri-architectural Convolution Neural Network (VGGNet GoogLeNet, and AlexNet). Pituitary gland tumours, glioma tumours, and meningioma tumours were all included in this categorisation. The method mentioned above sliced the brain MRI to discover regions of interest. The data sets were also fine-tuned and frozen in preparation for further categorisation. The authors also investigated data augmentation strategies to improve the accuracy of the results. This study achieved a 98.69 per cent accuracy by increasing classification and detection with the VGG16 architecture.

C. Convolution Neural Network accuracy performance and consideration

In general overview, the performance of CNN was consistent over the literature review. The accuracy is much higher than the theoretical 50% baseline of random classification. However, this kind of analysis requires a more significant number of train images or parameter tuning [13].

Moreover, The transfer learning models can easily classify results with augmentation techniques. For example, in [14] using three popular CNN architectures as VGG16, ResNet and Inception models, which were trained on ImageNet database. The result using augmentation techniques was significant in overcoming the unbalanced dataset of normal and tumour images to improve the segmentation accuracy.

The number of layers used is also an important consideration. For example, in [15] the authors provided a self-designed CNN brain tumour classification method by using binary group data and also compared the classical Alexnet and MobileNet models. The result showed that their CNN model, which has 11 layers, gets a higher accuracy of 98.8%.

In [16] authors created a CNN-based framework for feature extraction. They also created a 5-layered architecture with all levels learnable and a customised 33-layered arrangement. Results revealed an accuracy of 81 per cent, which was further improved by another CNN-based classification model based on ELM (extreme learning machine), where the number of layers and design influence the outcome.

Finally, the autoencoder and K-means model can improve precise and clear segmented images to help save human efforts and time. These techniques could be an excellent method after detecting the image improving the performance by reaching a 95.55% accuracy using this technique [17].

III. METHODOLOGY

Many sources generate a large amount of data. It is vital to extract useful information from data in order to make better decisions. Several datasets are now available for the detection of Brain Tumour. This study is carried out using the Knowledge Discovery in Databases (KDD) approach. To

identify and diagnose brain tumor , KDD employs a pre-trained convolutional neural network to identify, analyze, and provide insights utilizing modified raw tumor images. Figure 2 depicts the KDD Methodology used in this study.

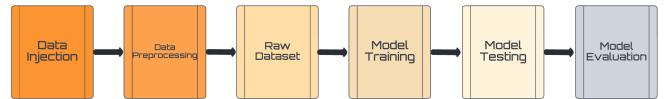


Fig. 2. KDD Process

IV. IMPLEMENTATION

A. Data Selection

As noted in the related works section, there has already been a substantial amount of study done in the subject of Brain Tumor detection utilizing various data sources. To obtain tumor images for our research, we will rely on such data sources. One of such popular dataset is publicly available dataset on kaggle¹ and the image sample is shown in the Figure 3.

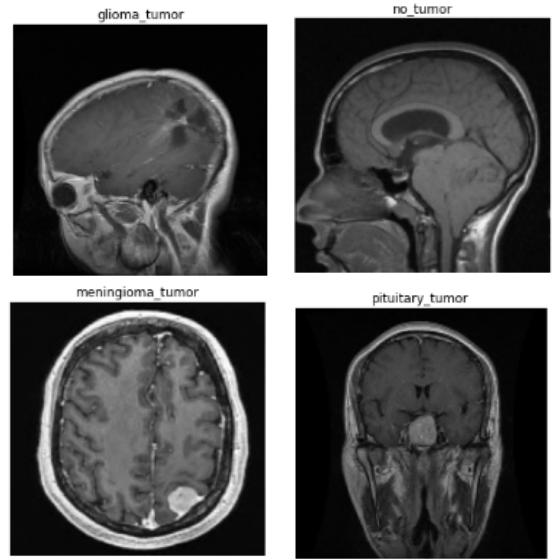


Fig. 3. Image Sample from Kaggle

B. Data Pre-processing

The second phase in the KDD's data mining technique is data pre-processing, which converts unstructured data into a structured format that is more efficient, useful, and understandable. In general, the dataset provides images which are stored in its respective class folder. As the study tries to conduct multi-class classification for categorizing brain tumor, the very first task is to map the images with their respective labels. The dataset will subsequently be classified/labeled as shown below.

¹<https://www.kaggle.com/datasets/sartajbhuvaji/brain-tumor-classification-mri>

- 1) Glioma Tumor
- 2) Meningioma Tumor
- 3) No Tumor
- 4) Pituitary Tumor

C. Data Transformation

Data transformation in KDD refers to the process of converting unprocessed data to processed data. This stage is crucial in data management because it brings the data to the point where it could be immediately put into the DL or ML Model. Using a random sample approach, the data will be separated into training and test sets in an 80/20 ratio. Image correction is the first transformation task in this phase, so we will be reshaping the images into 200x200 size. The next step is create a dataset by randomly rotating the images, flipping horizontally, height and width shifting and zooming in. The purpose of using this method is to improve the accuracy of our model by exposing it to different types of image.

D. Exploratory Data Analysis

Our dataset contains images and label, the only data analysis which can be performed is the analysis of the different classes of the brain tumor images. As seen from the fig, 4, the data for No_Tumour is less as compared to other classes indicating a class imbalance problem. But in this cases we will not be performing any oversampling techniques to increase the minority class count to avoid over-fitting.

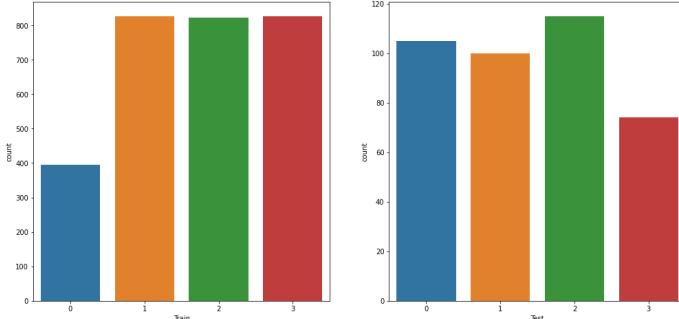


Fig. 4. EDA process

E. Model Design for Data Mining

In the this section, we have to consider the structure of the CNN model which has convolution, pooling, and dense layers we have used for our brain X-ray image classification.

- Convolution layer can extract features from the brain X-ray images by the filter and into the pixel value and extract features like curve, the edge from the image and generate a feature map.
- The pooling layer reduces the image dimensions and we use the max-pooling can select the maximum value in a region from the filter size.
- Dense layer which is the part of the final layer of CNN can capable of recognizing features and outputting a one-dimensional vector obtained as the result of the previous pooling layer.

TABLE I
SEQUENTIAL() FUNCTION DESCRIPTION

Default Model	
Layers	32,(3,3), relu (2,2)
Loss	64, (3, 3), relu (2,2)
Accuracy	128, (3, 3), relu
Val-loss	32, adam, epochs=10
Val-accuracy	0.1602
	0.9519
	5.1033
	0.7081

In our project, use the sequential() function to build the layers and this function is from the models and layers in "tensorflow.keras" package which we have imported. Our first layer uses Convolution layer 2D with 32 filters and take 3x3 pixels and the activation function use Relu with our input image size is 200 x 200 and the pooling layer we use maxpooling2D 2 x 2 to reduce the dimension and in the second Convolution layer use 64 filters and again with the maxpooling2D 2 x 2 and the final Convolution layer use 128 filters with maxpooling 2D 2x2 and use the layers. Flatten() to convert the matrix to a fully connected layer and the layers. Dense with Relu activation function to convert to 32 values with final layer.Dense with 4 values as our output classification value. The model summary can see Figure 5.

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 198, 198, 32)	896
max_pooling2d (MaxPooling2D)	(None, 99, 99, 32)	0
conv2d_1 (Conv2D)	(None, 97, 97, 64)	18496
max_pooling2d_1 (MaxPooling 2D)	(None, 48, 48, 64)	0
conv2d_2 (Conv2D)	(None, 46, 46, 128)	73856
max_pooling2d_2 (MaxPooling 2D)	(None, 23, 23, 128)	0
flatten (Flatten)	(None, 67712)	0
dense (Dense)	(None, 32)	2166816
dense_1 (Dense)	(None, 4)	132

Total params: 2,260,196
Trainable params: 2,260,196
Non-trainable params: 0

Fig. 5. Module Structure

We have applied the model.compile which defines the loss function after the CNN structure has build-up as follows:

```
1 loss function = tf.keras.losses.  
SparseCategoricalCrossentropy(from_logits=True),  
Optimizer = 'adam' and metrics = ['accuracy']
```

We can use our train_data with train_labels let model.fit to training our data with epochs setting as 10 and use test_data and test_labels for the validation_data and we can see the

model training processing status on each epoch step in Figure 6.

```
Epoch 1/10
96/98 [=====] - 53s 580ms/step - loss: 13.8861 - accuracy: 0.5881 - val_loss: 2.2702 - val_accuracy: 0.4695
Epoch 2/10
96/98 [=====] - 54s 590ms/step - loss: 0.5058 - accuracy: 0.8125 - val_loss: 2.0384 - val_accuracy: 0.7799
Epoch 3/10
96/98 [=====] - 55s 614ms/step - loss: 0.2048 - accuracy: 0.8868 - val_loss: 1.0759 - val_accuracy: 0.8236
Epoch 4/10
96/98 [=====] - 56s 626ms/step - loss: 0.1665 - accuracy: 0.9369 - val_loss: 4.0107 - val_accuracy: 0.6853
Epoch 5/10
96/98 [=====] - 56s 627ms/step - loss: 0.1012 - accuracy: 0.9627 - val_loss: 3.1866 - val_accuracy: 0.7107
Epoch 6/10
96/98 [=====] - 53s 590ms/step - loss: 0.0980 - accuracy: 0.9732 - val_loss: 5.5350 - val_accuracy: 0.7188
Epoch 7/10
96/98 [=====] - 52s 580ms/step - loss: 0.0641 - accuracy: 0.9777 - val_loss: 7.3565 - val_accuracy: 0.7110
Epoch 8/10
96/98 [=====] - 53s 587ms/step - loss: 0.0333 - accuracy: 0.9892 - val_loss: 6.8190 - val_accuracy: 0.7422
Epoch 9/10
96/98 [=====] - 52s 580ms/step - loss: 0.0560 - accuracy: 0.9847 - val_loss: 5.9277 - val_accuracy: 0.7183
Epoch 10/10
96/98 [=====] - 54s 580ms/step - loss: 0.1602 - accuracy: 0.9519 - val_loss: 5.3033 - val_accuracy: 0.7081
```

Fig. 6. Model training processing status

V. EVALUATION AND RESULTS

In Figure 7, it is possible to see the performance of accuracy and loss over the epoch increase. The first result is the distance between the training and validation accuracy (26.79 percentual points), even when both have their best performance at epoch 6. On the other hand, the loss remains stable after the epoch 1 for the training set, when the validation set has slightly increased over the epoch until the epoch when it reaches the peak and goes down.

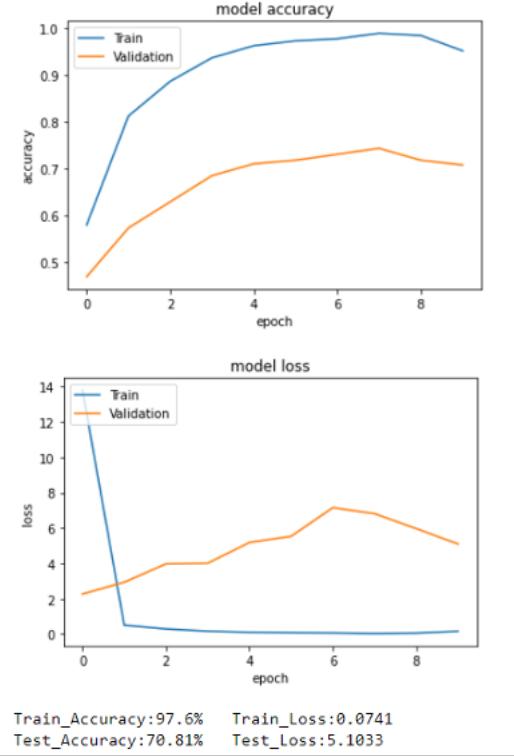


Fig. 7. Accuracy and Loss plots

On the other hand, we checked the confusion matrix as a summary for our prediction results of the classification problem. From there, we can interpret a problem trying to

distinguish between "no tumour" and "glioma tumour", being over predicted the "no tumour" (see Figure 8).

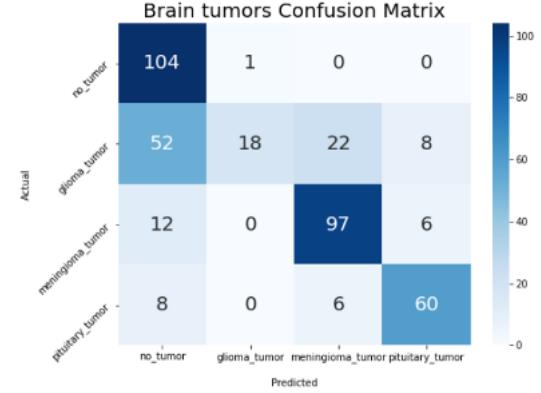


Fig. 8. Confusion Matrix of Brain tumors

A. Experiments

As part of the evaluation process, we run a series of experiments changing the filters on layers and finding better accuracy to understand our results better. Also, we changed the activation function. After that checking, the dense layer found better results compared with different epoch configurations, and we were able to decide on our final model.

1) Change filters: The first experiment was trying to change the filter in each Convolution layer. We chose 32,64 and 128, which compared with the default accuracy. As a result, we got better performance for the validation and test set. We got a score of 72,5% as the best score at this stage (See Figure 9).

	Conv	Dens	opt	epo	train_loss	train_acc	test_loss	test_acc
Default_model	[32, relu, 64, relu, 128, relu]	[32, relu]	adam	10	0.160200	0.951900	5.103300	0.708100
Model_1	[32, relu, 32, relu, 32, relu]	[32, relu]	adam	10	0.077284	0.972125	5.291972	0.611675
Model_2	[64, relu, 64, relu, 64, relu]	[32, relu]	adam	10	0.094854	0.970732	6.448600	0.700508
Model_3	[128, relu, 128, relu, 128, relu]	[32, relu]	adam	10	0.049040	0.985017	4.349739	0.725888
Model_4	[128, tanh, 128, tanh, 128, tanh]	[32, tanh]	adam	10	1.348114	0.286411	1.440734	0.291878
Model_5	[128, sigmoid, 128, sigmoid, 128, sigmoid]	[32, sigmoid]	adam	10	1.348256	0.288153	1.444807	0.187817
Model_6	[128, relu, 128, relu, 128, relu]	[64, relu]	adam	10	0.042390	0.989199	6.777133	0.703046
Model_7	[128, relu, 128, relu, 128, relu]	[128, relu]	adam	10	0.009916	0.997561	7.999988	0.705584
Model_8	[128, relu, 128, relu, 128, relu]	[32, relu]	adam	20	0.007373	0.996167	8.921905	0.713198

Fig. 9. Models of Filters adjustment

2) Change activation function: The second adjustment was to try different activation functions between tanh and sigmoid. As a result, the accuracy of trying these changes was inferior. For that reason, we decided to be consistent with the relu function (See Figure 10).

3) Change dense layer: In the next step, we are trying to define the amount of our dense layer filter deciding among 32, 64 or 128 (See Figure 11). As a result, the 32 dense

	Conv	Dens	opt	epo	train_loss	train_acc	test_loss	test_acc
Default_model	[32, relu, 64, relu, 128, relu]	[32, relu]	adam	10	0.160200	0.951900	5.103300	0.708100
Model_1	[32, relu, 32, relu, 32, relu]	[32, relu]	adam	10	0.077284	0.972125	5.291972	0.611675
Model_2	[64, relu, 64, relu, 64, relu]	[32, relu]	adam	10	0.094854	0.970732	6.448600	0.700508
Model_3	[128, relu, 128, relu, 128, relu]	[32, relu]	adam	10	0.049040	0.985017	4.349739	0.725888
Model_4	[128, tanh, 128, tanh, 128, tanh]	[32, tanh]	adam	10	1.348114	0.286411	1.440734	0.291878
Model_5	[128, sigmoid, 128, sigmoid, 128, sigmoid]	[32, sigmoid]	adam	10	1.348256	0.288153	1.444807	0.187817
Model_6	[128, relu, 128, relu, 128, relu]	[64, relu]	adam	10	0.042390	0.989199	6.777133	0.703046
Model_7	[128, relu, 128, relu, 128, relu]	[128, relu]	adam	10	0.009916	0.997561	7.999988	0.705584
Model_8	[128, relu, 128, relu, 128, relu]	[32, relu]	adam	20	0.007373	0.996167	8.921905	0.713198

Fig. 10. Models of activation function adjustment

layer option performs slightly better, so we run with this configuration.

	Conv	Dens	opt	epo	train_loss	train_acc	test_loss	test_acc
Default_model	[32, relu, 64, relu, 128, relu]	[32, relu]	adam	10	0.160200	0.951900	5.103300	0.708100
Model_1	[32, relu, 32, relu, 32, relu]	[32, relu]	adam	10	0.077284	0.972125	5.291972	0.611675
Model_2	[64, relu, 64, relu, 64, relu]	[32, relu]	adam	10	0.094854	0.970732	6.448600	0.700508
Model_3	[128, relu, 128, relu, 128, relu]	[32, relu]	adam	10	0.049040	0.985017	4.349739	0.725888
Model_4	[128, tanh, 128, tanh, 128, tanh]	[32, tanh]	adam	10	1.348114	0.286411	1.440734	0.291878
Model_5	[128, sigmoid, 128, sigmoid, 128, sigmoid]	[32, sigmoid]	adam	10	1.348256	0.288153	1.444807	0.187817
Model_6	[128, relu, 128, relu, 128, relu]	[64, relu]	adam	10	0.042390	0.989199	6.777133	0.703046
Model_7	[128, relu, 128, relu, 128, relu]	[128, relu]	adam	10	0.009916	0.997561	7.999988	0.705584
Model_8	[128, relu, 128, relu, 128, relu]	[32, relu]	adam	20	0.007373	0.996167	8.921905	0.713198

Fig. 11. Models of Dense layer adjustment

4) *Change Epochs:* Finally, we run our model using different epochs until the number 20. As is shown in Figure ?? it is possible to determine that is no accuracy improvement after the epoch 10. However, as we had a risk of overfitting, it is compulsory to go back at the history plots of accuracy and loss to find a better configuration for our model.

	Conv	Dens	opt	epo	train_loss	train_acc	test_loss	test_acc
Default_model	[32, relu, 64, relu, 128, relu]	[32, relu]	adam	10	0.160200	0.951900	5.103300	0.708100
Model_1	[32, relu, 32, relu, 32, relu]	[32, relu]	adam	10	0.077284	0.972125	5.291972	0.611675
Model_2	[64, relu, 64, relu, 64, relu]	[32, relu]	adam	10	0.094854	0.970732	6.448600	0.700508
Model_3	[128, relu, 128, relu, 128, relu]	[32, relu]	adam	10	0.049040	0.985017	4.349739	0.725888
Model_4	[128, tanh, 128, tanh, 128, tanh]	[32, tanh]	adam	10	1.348114	0.286411	1.440734	0.291878
Model_5	[128, sigmoid, 128, sigmoid, 128, sigmoid]	[32, sigmoid]	adam	10	1.348256	0.288153	1.444807	0.187817
Model_6	[128, relu, 128, relu, 128, relu]	[64, relu]	adam	10	0.042390	0.989199	6.777133	0.703046
Model_7	[128, relu, 128, relu, 128, relu]	[128, relu]	adam	10	0.009916	0.997561	7.999988	0.705584
Model_8	[128, relu, 128, relu, 128, relu]	[32, relu]	adam	20	0.007373	0.996167	8.921905	0.713198

Fig. 12. Models of Epoch layer

The Epoch 20 as Model_8 we can see on the history of accuracy plot (Figure 13) after epoch 6 which has test_acc

0.62 or epoch 11 which has test_acc 0.72 (Figure 14) showing like the over-fitting. We have decided to check the epoch 6 and epoch 11 on Model_8 and compare the confusion matrix then decide our final model (See Figure ?? and ??).

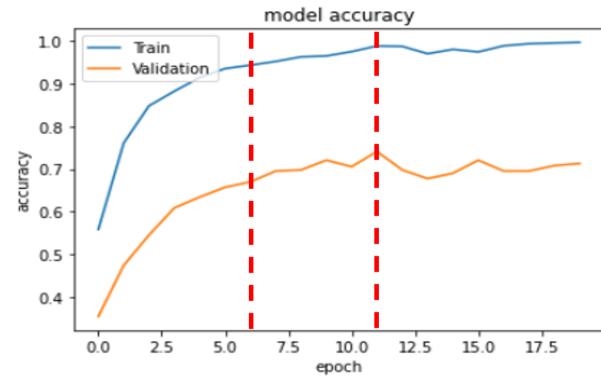


Fig. 13. Model_8 history of accuracy plot

The suspect of overfitting is visible in Figure 14, where the training accuracy and test accuracy was 0.96 and 0.62, and 0.99 and 0.72, corresponding to epoch 6 and 11, respectively.

	Conv	Dens	opt	epo	train_loss	train_acc	test_loss	test_acc
Epoch 6	[128, relu, 128, relu, 128, relu]	[32, relu]	adam	6	0.118795	0.963763	3.504443	0.624366
Epoch 11	[128, relu, 128, relu, 128, relu]	[32, relu]	adam	11	0.009741	0.997213	7.478468	0.725888

Fig. 14. Epoch 6 and Epoch 11 of model_8

Comparing both Confusion Matrix shows that the over-classification as "no tumour" is one of the main issues. However, as before, the model predicts no tumour in some glioma cases. Still, in the epoch 6 confusion matrix (See Figure 15), it is also over-predicted as no tumour in some meningioma cases, and most of these cases are well classified under epoch 11 (See Figure 16). The insight presented is a piece of partial evidence in favour of the model with epoch 11. Still, as an overview, the main issue in the overfitting problem is the correct classification of glioma, sometimes as no tumour and others as meningioma. That is consistent in both models.

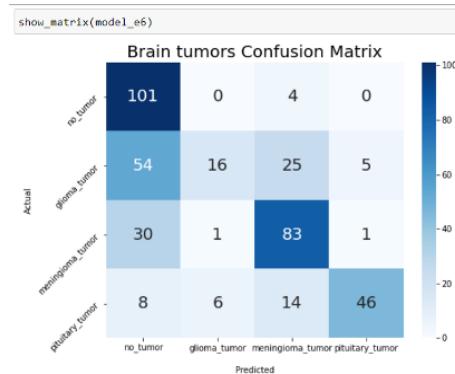


Fig. 15. Confusion matrix of Epoch 6 of model_8

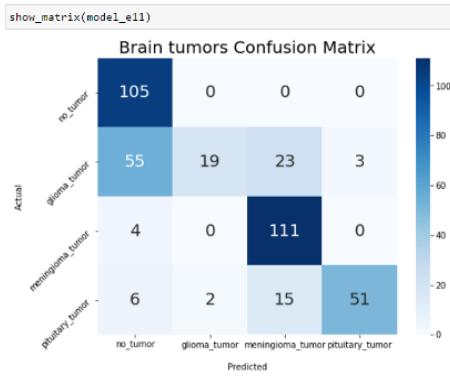


Fig. 16. Confusion matrix of Epoch 11 of model_8

Finally, we present the performance of our model with epochs 6 and 11 in the Figure 1818. It shows the precision, recall and f1-score, considering that it evident the problem with the category of glioma tumour and slightly better performance over the model with epoch 11.

```
test_pred=np.argmax(model_e6.predict(test_data),axis=1)
print(classification_report(test_labels,test_pred))

precision    recall   f1-score   support
0            0.62    0.98    0.76     105
1            1.00    0.17    0.29     100
2            0.68    0.96    0.80     115
3            0.90    0.61    0.73      74

accuracy                           0.62    394
macro avg                           0.80    394
weighted avg                         0.79    394
```

Fig. 17. Performance of Epoch 6 of model_8

```
from sklearn.metrics import classification_report

test_pred=np.argmax(model_e11.predict(test_data),axis=1)
print(classification_report(test_labels,test_pred))

precision    recall   f1-score   support
0            0.71    0.92    0.80     105
1            0.90    0.18    0.30     100
2            0.66    0.94    0.77     115
3            0.81    0.81    0.81      74

accuracy                           0.72    394
macro avg                           0.77    394
weighted avg                         0.76    394
```

Fig. 18. Performance of Epoch 11 of model_8

VI. CONCLUSIONS

Firstly, the use of a Convolutional Neuronal Network was fully documented and our best option for the brain tumour classification task. After a detailed evaluation process, we reached two models as the solution. The model's accuracy with epoch 6 is 0.62 in contrast with the 0.72 of the model with 11, which seems to be our best option.

Secondly, the accuracy obtained for our best model is not as good as in previous experiences in the related work section.

However, the main goal at this stage was the experience of a crash course using Deep Learning methodologies, considering that our goal is fully achieved.

On the other hand, our limitations are related to performance and data. It is state-of-art in this field, where overfitting is still the main issue for deep learning models and the difficulty of having massive amounts of good data for training. Our project is not free of these difficulties in this field and part of our limitation.

Furthermore, implementation of more complex techniques for improving the model or adjustments for upgrading the classification of one of the categories in particular, as we saw in the related work, is a matter for future work, same as looking for more images for the training of our model.

Finally, this situation helps us to understand why the heterogeneity in this field and why sometimes, models with performance over 96% accuracy it has not been possible to be widely implemented in clinical practice. But, on the other hand, this technological solution can redefine the interaction and experience between doctor and patient, in a sensible area of our life as Health and in a difficult moment when you possibly face a brain tumour.

All this sociological consideration makes an effort to apply Deep Learning to the Health field, not just a data analytic problem. Moreover, it is why it is essential to consider developing guidelines for standards in this field, such as quantifiable benefits, regulatory pathways that are lengthy and costly, patient safety considerations, data privacy, and economic factors like who paid for AI tools [5].

REFERENCES

- [1] S. Garg and P. Singh, "State-of-the-art review of deep learning for medical image analysis," *Third International Conference on Intelligent Sustainable Systems [ICISS 2020]*, 2020.
- [2] A. Anaya-Isaza, L. Mera-Jimenez, and M. Zequera-Diaz, "An overview of deep learning in medical imaging," *Informatics in Medicine*, vol. 26, 2021.
- [3] R. Aggarwal, V. Sounderajah, G. Martin, D. Ting, A. Karthikesalingam, D. King, H. Ashrafian, and A. Darzi, "Diagnostic accuracy of deep learning in medical imaging: a systematic review and meta-analysis," *Digital Medicine*, vol. 4:65, 2021.
- [4] B. Han, "Application of deep learning in medical imaging," *2nd International Conference on Computing and Data Science (CDS)*.
- [5] K. Zhou, J. Duncan, and J. Prince, "A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises," *IEEE*, vol. 109, 2021.
- [6] T. Arumaitthurai and B. Mayurathan, "The effect of deep learning and machine learning approaches for brain tumor recognition," *10th International Conference on Information and Automation for Sustainability (ICIAS)*, pp. 185–190, 2021.
- [7] S. K. Baranwal, K. Jaiswal, K. Vaibhav, A. Kumar, and R. Srikanthaswamy, "Performance analysis of brain tumour image classification using cnn and svm," *Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 537–542, 2020.
- [8] H. Kibriya, M. Masood, M. Nawaz, R. Rafique, and S. Rehman, "Multi-class brain tumor classification using convolutional neural network and support vector machine," *Mohammad Ali Jinnah University International Conference on Computing (MAJICC)*, pp. 1–4, 2021.
- [9] C. Choudhury, C. Mahanty, R. Kumar, and B. Mishra, "Brain tumor detection and classification using convolutional neural network and deep neural network," *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, 2020.

- [10] R. Ezhilarasi and P. Varalakshmi, "Tumor detection in the brain using faster r-cnn," *2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2018 2nd International Conference on, 2018, pp. 388–392, 2018.
- [11] N. Noreen, S. Palaniappan, A. Qayyum, I. Ahmad, M. Imran, and M. Shoaib, "A deep learning model based on concatenation approach for the diagnosis of brain tumor," *IEEE Access*, vol. 8, pp. 55135–55144, 2020.
- [12] A. Rehman, S. Naz, M. Razzaq, F. Akram, and M. Imran, "A deep learning-based framework for automatic brain tumors classification using transfer learning," *Circuits, Systems, and Signal Processing*, vol. 39, pp. 757–775, 2019.
- [13] C. Someswararao, R. S. Shankar, S. V. Appaji, and V. Gupta, "Brain tumor detection model from mr images using convolutional neural network," *2020 Int. Conf. Syst. Comput. Autom. Networking, ICSCAN*, pp. 1–4, 2020.
- [14] R. M. Prakash and R. S. S. Kumari, "Classification of mr brain images for detection of tumor with transfer learning from pre-trained cnn models," *2019 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET*, pp. 508–511, 2019.
- [15] H. Chen, D. Chen, and L. Wang, "Cnn-based mri brain tumor detection application," *2021 Int. Conf. Comput. Eng. Appl. ICCEA*, pp. 464–467, 2021.
- [16] A. Pashaei, H. Sajedi, and N. Jazayeri, "Brain tumor classification via convolutional neural network and extreme learning machines," *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2018.
- [17] G. Raut, A. Raut, J. Bhagade, J. Bhagade, and S. Gavhane, "Deep learning approach for brain tumor detection and segmentation," *2020 Int. Conf. Converg. to Digit. World - Quo Vadis, ICCDW*, 2020.