|  |
| --- |
| Brain Tumour Classification Using Deep Learning |

# Abstract

Brain tumours are among the most dangerous diseases, and early identification of these tumours is important for successful treatments. Early detection and classification of brain tumours is an important research domain in the field of medical imaging and accordingly helps in selecting the most convenient treatment method to save patients’ life, therefore. In this research, we use deep learning to classify whether or not tumours are present in a brain tumour MRI imaging dataset, and if they are, we determine what form of tumour it is either Meningiomas, Pituitary, or Gliomas.

The goal of this research is to replace the manual diagnosis procedure with deep learning and machine learning algorithms in brain tumour diagnosis. In this paper, we employed multiple methods such as EfficientNet-B7, VGG-16 via Transfer Learning, proposed CNN model, Ensemble model, and XGBoost classifier to our dataset. As a result, in our work, we obtained a ***high accuracy of 97.63%*** for the ***EfficientNet-B7*** model during the ***experiment 1***. However, ***after increasing the size of the training dataset in experiment 2***, we were able to attain the greatest performance results for all models. However, the ***Ensemble model*** outperformed all other modes with the greatest ***accuracy score of 99.16%.***

Table of Contents:

[Abstract 1](#_Toc138437224)

[Acknowledgements 3](#_Toc138437225)

[1. Introduction 4](#_Toc138437226)

[1.1. Project Aim and Objectives 7](#_Toc138437227)

[1.2. Overview of This Report 7](#_Toc138437228)

[2. Literature Review 9](#_Toc138437229)

[3. Methodology 12](#_Toc138437230)

[4. Requirements 14](#_Toc138437231)

[4.1. Hardware Requirements 14](#_Toc138437232)

[4.2. System requirements 14](#_Toc138437233)

[4.3. Software requirements 14](#_Toc138437234)

[5. Analysis 15](#_Toc138437235)

[5.1. Data Engineering – Data Acquisition and Data Preparation 15](#_Toc138437236)

[5.2. Model Training 15](#_Toc138437237)

[5.3. Model Evaluation 15](#_Toc138437238)

[5.4. Hyper Tuning 17](#_Toc138437239)

[6. Design 18](#_Toc138437240)

[6.1. Overview of CNN Model 18](#_Toc138437241)

[6.1.1. The Convolutional Layer 18](#_Toc138437242)

[6.1.2. Activation Layer: 20](#_Toc138437243)

[6.1.3. Pooling Layers: 20](#_Toc138437244)

[6.1.4. Fully Connected Layers 21](#_Toc138437245)

[6.1.5. Architecture of proposed CNN Model 24](#_Toc138437246)

[6.2. Transfer Learning 26](#_Toc138437247)

[6.2.1. VGG Model - Architecture 27](#_Toc138437248)

[6.2.2. EfficientNet Model 29](#_Toc138437249)

[6.3. Ensemble Model - Bootstrap Aggregation 32](#_Toc138437250)

[6.4. Extreme Gradient Boosting 33](#_Toc138437251)

[7. Implementation 34](#_Toc138437252)

[8. Experimental Results and Discussions 37](#_Toc138437253)

[9. Project Management 42](#_Toc138437254)

[9.1. Project Schedule 42](#_Toc138437255)

[9.2. Risk Management 44](#_Toc138437256)

[9.3. Quality Management 44](#_Toc138437257)

[9.4. Social, Legal, Ethical and Professional Considerations 45](#_Toc138437258)

[10. Critical Appraisal 46](#_Toc138437259)

[11. Conclusions 47](#_Toc138437260)

[47](#_Toc138437261)

[11.1. Achievements 47](#_Toc138437262)

[11.2. Future Work 47](#_Toc138437263)

[12. Student Reflections 48](#_Toc138437264)

[Bibliography and References: 49](#_Toc138437265)

**Table of Figures:**

[Figure 1 - Sample brain tumour images from dataset 9](#_Toc121490362)

[Figure 2 - Iterative & Incremental Model (Iterative Development: A Starter’s Guide, 2021) 15](#_Toc121490363)

[Figure 3 - Confusion Matrix 18](#_Toc121490364)

[Figure 4 - Workflow of the Project and process 19](#_Toc121490365)

[Figure 5 – Overview of CNN model for Brain Tumour Classification (Mdpi.com, 2022) 21](#_Toc121490366)

[Figure 6 - ReLU Function and Output of ReLU layer for one feature (Working of Convolutional Neural Network - Javatpoint, n.d.) 22](#_Toc121490367)

[Figure 7 - Difference between Average and Max Pooling (Aljaafari, Nura, 2018) 23](#_Toc121490368)

[Figure 8 - Role of Weight and bias in Neural Networks (Ganesh, 2020) 25](#_Toc121490369)

[Figure 9 - Architecture of Proposed CNN Model 26](#_Toc121490370)

[Figure 10 - Proposed CNN Model 27](#_Toc121490371)

[Figure 11 - Overview of Transfer Learning implemented in Brain Tumour Classification (Mdpi.com, 2022) 28](#_Toc121490372)

[Figure 12 – A Standard VGG-16 Network Architecture (Younis, Qiang, Nyatega, Adamu, & Kawuwa, 2022) 30](#_Toc121490373)

[Figure 13 - VGG-16 Architecture (Tammina, 2019) 30](#_Toc121490374)

[Figure 14 – Model Scaling (Tan & Le, 2019) 31](#_Toc121490375)

[Figure 15 - Ensemble Model Architecture 34](#_Toc121490376)

[Figure 16 - XGBoost Classifier (How XGBoost Works - Amazon SageMaker, n.d.) 35](#_Toc121490377)

[Figure 17 - Proposed CNN Model Graph 37](#_Toc121490378)

[Figure 18 - EfficientNet - B7 Model - Training and Validation Accuracy & Loss Graph 37](#_Toc121490379)

[Figure 19 - Model Size vs ImageNet Accuracy (Tan & Le, 2019) 39](#_Toc121490380)

[Figure 20 - Project Schedule Chart 44](#_Toc121490381)

[Figure 21 - Project Schedule 45](#_Toc121490382)

# **Acknowledgements**

I would like to thank my supervisor for her advice and support over the course of my project. Her feedbacks and reviews were timely, and she was polite and supportive throughout the process. Her comments have enabled me to improve my work to a better level. I would also like to thank every member of the faculty who has guided me through his course, taught me well, and shared their valuable knowledge with me. Finally, I would like to thank my husband for all of his sacrifices, patience, and support in caring for me and my son during my studies, and for always encouraging me to do my best.

# **Introduction**

Machine learning and Deep learning are both types of AI. In machine learning when something goes wrong, generally it requires human intervention and correction. Deep learning is considered as an evolution of Machine Learning.

In Deep Learning algorithms can enhance and increase their results through repetition without human intervention. For Deep Learning, we require a large dataset with diverse and unstructured data. The term deep learning refers to a technique of machine learning that layer’s algorithms and computing units, or neurons, into what is known as an artificial neural network. The structure of these deep neural networks is derived from that of the human brain. Like how our brains process information, the data passing through this web of interconnected algorithms moves in a non-linear pattern (AI vs. Deep Learning vs. Machine Learning: Beginner’s Guide. (n.d.)).

As described in a recent article (HealthITSecurity, 2022) Artificial Intelligence has a vital role in enhancing the outcomes in medical imaging for screenings, precision medicines and risk assessments. Different reasons and need of medical imaging include a cardiac event, fracture, neurological condition, or thoracic complications. AI can quickly diagnose and provide treatment options. Incorporating AI into medical imaging can enhance medical screenings, improve precision medicine, assess patient risk factors, and lighten the load on physicians. With this huge advantage, physicians can identify tumour conditions much faster and quicker, helping early involvement (HealthITSecurity, 2022).

A growth of abnormal brain cells is known as a brain tumour. The brain's anatomy is very complicated, with different parts responsible for different functions of the nervous system. Any part of the brain or skull, including the protective lining, the underside of the brain (skull base), the brainstem, the sinuses and nasal cavity, and many other areas, can develop a brain tumour. Depending on the tissue they originate from, more than 120 distinct types of tumors can develop in the brain. Few tumours originate in the brain tissue, spread outside of the brain or spine, and are life-threatening because they are hard to reach and treat with surgery (Brain Tumors and Brain Cancer. (n.d.).).

* Meningiomas form in the meninges, the protective lining of the brain , and they are benign (non-cancerous).
* Pituitary tumors develop in the pituitary gland, and they are benign (non-cancerous).
* Gliomas are composed of glial cells, and they are malignant (cancerous).

The severity of a brain tumour is determined by its grade. Utilizing the biopsy test, a pathologist will look at the growth under a magnifying instrument to decide its grade. Brain tumour grading is a classification system that shows how likely it is for the tumour to grow and spread and describes the cells in the brain. Brain tumour grading uses a scale from 1 (least aggressive) to 4 (most aggressive) (World Health Organization tumour grading system) (Brain Tumors and Brain Cancer. (n.d.).).

Generally, a brain tumour diagnosing involves a neurological exam, brain scans and a biopsy, if it can be done safely. In a neurological exam various tests are performed to evaluate neurological functions such as balance, hearing, vision, and reflexes (Brain Tumors and Brain Cancer. (n.d.).). The analysis of these neurological reports consumes huge time and performing the biopsy are invasive. Magnetic Resonance Imaging (MRI) and Computer tomography (CT) Scans are considered to be good for looking for tumours in the brain. But to rely on MRI data alone and detect the brain tumours was difficult for the physicians, hence the researchers planned on developing a deep learning system that relies only on the MRI Scan data for diagnosing and classification of different types of brain tumours.

In a recent article by Shania Kennedy (Kennedy, 2022) she was discussing about the findings from a recent research published in *JAMA Network Open* (Gao, et al., 2022) in which it shows that researchers have found that a deep learning tool may improve brain tumour classification and aid neuroradiologists in diagnosis. They used the MRI scan data collected between 2000 and 2019 from 37,871 patients. As a result of their research, they were able to achieve high accuracy, sensitivity, and specificity overall. In terms of accuracy and sensitivity it outperformed the neuroradiologists but achieved a similar specificity. The researchers further investigated the classification of brain tumours by the neuroradiologists with the help of model as a support tool and without the model’s assistance. Here, the researchers found that with the assistance of the model, neurologists’ average accuracy increased by 12 percentage points, from 63.5 percent without assistance to 75.5 percent with assistance (Kennedy, 2022).

The application of deep learning approaches in context to improve health diagnosis is providing efficient and great solutions. Proper brain tumour diagnosis involves detection, brain tumour location identification, and classification of the tumour based on malignancy, grade, and type. CNN is a supervised type of Deep learning, most preferable used in image recognition and computer vision. The Convolutional Neural Network (CNN) based multi-task classification is equipped for the classification and detection of tumours.

Our study is based on a dataset of MRI images of brain tumors, including both benign and malignant brain tumour images. We are implementing various types of CNN in this paper and find which performs well in classifying them accordingly. We are using the VGG-16 and EfficientNet-B7 models for classification into action by means of Transfer learning. The classification of the brain tumors is then done using a proposed CNN model. We are also calculating the XGBoost model's accuracy by feeding the XGBoost classifier with the features extracted from the proposed CNN model as its input. In addition, all three CNN models are combined to create an ensemble model, and finally the accuracy scores of each model are compared to determine which one is the highest and most accurate.

**Figure 1 - Sample brain tumour images from dataset**

A close-up of a brain scan

Description automatically generated with low confidence

## **Project Aim and Objectives**

* The main purpose of this project is to determine which CNN model is the most efficient and accurate for brain tumour classification using MRI data.
* As a result, the major goal of this research is to establish which of the most suggested CNN models, such as EfficientNet-B7, VGG-16, proposed CNN, Ensemble model, and XGBoost classifier, performs well in brain tumour image classification.
* To accomplish our primary goal, we pursue the following objectives in our paper:
* Dataset collection
* Investigating and studying about 4 different CNN models for effective brain tumour classification.
* Developing 5 different CNN models and training them to predict if a patient has a brain tumour or not, and if so, what type of tumour it is (Meningiomas, Glioma, Pituitary).
* Use the model to evaluate and understand the outcomes, and then fine-tune them with hyper - parameters if necessary.

## **Overview of This Report**

This paper is organised into 12 chapters, each of which refers to a specific series of activities, as indicated below. Appendix contains additional information and materials for reference.

Chapter 1 - This chapter summarizes about the project introduction as well as its goals and objectives.

Chapter 2 – This chapter discusses thorough research and analysis of prior relevant work for brain tumour classification, deep learning, and CNN algorithms.

Chapter 3 – In this chapter, we have explained about the project methodology (incremental and iterative approach) and the logic behind its selection for this project.

Chapter 4 – This chapter addresses about how the dataset is collected, its type, as well as all of the software, hardware, and system requirements for implementing the project.

Chapter 5 – In this we have discussed about the building blocks and the process flow of the deep learning project to be followed while performing image classification.

Chapter 6 – This chapter gives a complete analysis, design, and the architecture of all the CNN models implemented in our project.

Chapter 7 – In this section, we have discussed about the specifications and the parameters used to implement the 4 types of CNN models in brain tumour classification.

Chapter 8 – This chapter, discusses about the strength and weakness of all the 5 different models and about the results derived using the CNN models.

Chapter 9 - This chapter discusses project management issues such as project breakdown structure, risk identification, quality and risk management methodologies used, and social, legal, and ethical factors.

Chapter 10 – This chapter discusses the analysis and the results of the work, both positive and negative. Also, about the amount of knowledge gained during the course of this project.

Chapter 11 – This chapter examines the outcomes obtained in terms of the initial project objectives, as well as any additional work that may be required.

Chapter 12 - This chapter highlights my personal performance, challenges encountered and how they were resolved, and lessons learned.

# **Literature Review**

Machine learning, and in particular deep learning, has been argued that it has the potential to overcome several of the challenges associated with the detection and intervention of brain tumors in the literature. In data science and artificial intelligence, deep learning is considered as one of the most effective and great methods to train models through data and to develop valuable decision-making abilities. The goal of such models is to reduce the image while maintaining the information needed to make predictions in order to obtain the predicted network (Younis, Qiang, Nyatega, Adamu, & Kawuwa, 2022).

Deep learning is a subset of machine learning where artificial neural networks, algorithms inspired by the human brain, learn from enormous amounts of data. It is being using in various sectors and industries like banking, medical and pharmaceutical, Entertainment for various tasks like virtual assistants, voice recognition, language translations, facial recognition, chatbots and service bots, Image colorization, personalized shopping, and entertainment by proposing customer recommendation systems.

In medical and healthcare industry, it is being used in different aspects such as Clinical Decision Support systems, Smart Recording Robotic Surgery, Drug Discovery and Production, personalized medicine, ML in medical Imaging, tumour and cancer cells detections, Predictive approaches to treatments, Infectious disease outbreak prediction and Clinical research. With the help of Deep learning and ML algorithms is easier to achieve faster results which allows to start the treatment at the earliest (Tkachenko, 2021).

There has been a great deal of research into the detection and classification of brain tumors in recent years. A variety of techniques have been proposed for detecting and categorizing brain tumors on the basis of their characteristics. Among these methods are traditional machine learning (ML) methods as well as deep learning (DL). As part of this section, we will investigate existing approaches to the detection and classification of brain tumors (Ullah, et al., 2022).

Habiba, S. U, Andersson, K et al., (Habiba, et al.) worked on a deep learning-based classifier for brain tumour detection and classification using a transfer learning approach in InceptionV3 and DenseNet201 models and their proposed model has shown approximately 96.3% accuracy. Sethy, P.K., and Behera, S.K (Sethy, P.K., Behera, S.K.) studied tumorous brain MR images using deep learning methods using Alexnet, VGG16, and VGG19 pre-trained networks. In addition to deep fusion, Principal Component Analysis (PCA) was applied to reduce the feature vector dimensions to increase performance. Furthermore, the proposed approach avoids reproducing MR images, which generate anatomically incorrect images. Due to the small dataset, prejudice diagnosis was also avoided. Therefore, even if the proposed approach is data constrained, vgg16 with the fused features of fc6 and fc7 with linear SVM achieved the maximum value of accuracy is 0.97, sensitivity, specificity, and precision is 1, and F1 score of 97.92.

Another study by Nyoman Abiwinanda et al. (Abiwinanda, Hanif, Hesaputra, Handayani, & Mengko, 2018) trained a Convolutional Neural Network (CNN) to recognize three types of brain tumours, including Glioma, Meningioma, and Pituitary. They implemented the simplest possible CNN architecture, which consisted of three convolution layers, max-pooling layers, flattening layers, and a full connection from one hidden layer. Using simple architecture and without any prior region-based segmentation, the CNN trained on 3064 T-1 weighted CE-MRI images and achieved a training accuracy of 98.51% and validation accuracy of 84.19%.

Using CNN deep learning structure, Pan. Y et al. (Pan, et al., 2015) developed grading processes for brain tumour classification. Based on sensitivity and specificity, CNN improved grading performance by 18% over NN. Visualizations of kernels and output results at different layers reveal a close match between learned kernels and tumour features. It was also observed that a more complex CNN structure might not outperform the results of simple structured CNNs. paper, a new approach was presented to classify brain tumours.

Hassan Ali Khan et al., (Khan, Jue, Mushtaq, & Mushtaq, 2020) presented a different study for brain tumour classification using CNN. In this research paper, to increase the training data size they used image edge detection technique and data augmentation. To provide an efficient methodology they trained a simple CNN model from the scratch on the augmented and pre-processed dataset and compared the results with the VGG16, ResNet-50, and Inception-v3 models. The model average training time per epoch is 205 sec while the VGG-16 takes 456 sec, ResNet-50 takes 606 sec and Inception-v3 takes 375 sec average training time per epoch. These conclude that their proposed CNN model needs fewer computational specifications as it takes less execution time. Also, their model accuracy is much better than VGG-16, ResNet-50, and Inception-v3. For more accurate results and to avoid overfitting the neural networks need to be trained on a larger dataset.

According to Ullah et al. 's recent article, a comparative analysis of nine deep-learning models was performed with the help of transfer learning for the classification of brain tumours. In this paper, they have applied TL (transfer learning) to nine deep neural networks, i.e., Inceptionresnetv2, Inceptionv3, Xception, Resnet18, Resnet50, Resnet101, ShuffleNet, Densenet201, and Mobilenetv2, to classify brain tumours into glioma, meningioma, and pituitary. To all the transfer learning architectures that were used in this paper, the following parameters were applied: Stochastic gradient descent with momentum (SGDM) algorithm as the optimization algorithm, with a learning rate of 0.1 and a maximum of 14 epochs, and an accuracy of 30 and a verbose of false, respectively. Compared with other approaches that used deep learning models to extract deep features and SVM to classify brain tumours, Inceptionresnetv2 has confirmed the superiority of the best model. Additionally, they employed data augmentation techniques in order to increase the size of the training dataset. As a result, it is possible to further enhance the results of the model by training it on a larger dataset in the future (Ullah, et al., 2022).

In another recent study by Younis, et al., (Younis, Qiang, Nyatega, Adamu, & Kawuwa, 2022) aims to build a CNN model to detect brain tumours using MRI scans of the brain automatically. For faster and more convenient training of the network, they have chosen a pre-trained VGG-16 model. They aimed to discover a brain tumour using the VGG 16, CNN model architecture, and weights to training data. Accuracies and the correctness of the outcomes were evaluated. From their results, it was studied that the proposed network architecture was more appealing and performed well in detecting brain tumours. At the same time, numerous processing operations were also carried out to enhance the model’s efficiency. In this paper, the models yielded (Precision = 96%, 98.15%, 98.41%, and F1-score = 91.78%, 92.6%, and 91.29%) accuracy of CNN 96%, VGG 16 98.5%, and Ensemble Model is 98.14%. However, the dataset used on this paper, has 253 brain images from 155 different patient features and cases. To achieve higher precision and accuracy, ensembles could be further used on a comparatively larger dataset.

To classify 3260 T1-weighted contrast-enhanced brain magnetic resonance images into four categories (glioma, meningioma, pituitary, and no tumour), in a study by Nayak, et., proposed a CNN-based dense EfficientNet with min-max normalization. Their proposed network is a variation of EfficientNet with thick and drop-out layers added. In a similar manner, the authors increased the contrast of tumour cells by combining data augmentation with min-max normalization. The dense CNN model has the advantage of being able to accurately classify a limited database of images. Consequently, the proposed method delivers outstanding overall performance. The proposed model was 99.97% accurate during training and 98.78% accurate during testing, according to the experimental results. Their study concluded that the newly developed EfficientNet CNN architecture can be a useful tool for decision-making in the study of diagnostic tests for brain tumors due to its high accuracy and favourable F1 score (Nayak, Padhy, Mallick, Zymbler, & Kumar, 2022).

In a research paper by author Kim, two novel approaches were proposed to detect brain tumour using MRI images. By testing the concatenation of pre-trained models, they build on previous work on ensemble methods in the first proposed strategy: Classification algorithms or a stacked ensemble of those algorithms are used to merge and segment the features that have been extracted through transfer learning. They used a convolutional neural network with a specific module of layers used for classification in the second strategy. Outperforming a standard VGG-16 model, the first approach received an accuracy score of 0.98, while the second approach received a score of 0.863. In this paper they have also discussed about granular computing and the theory of circuit complexity (Kim D. , 2020).

In our paper, we are using 5 different models to predict the brain tumour classification by using ***VGG-16, EfficientNet-B7 (Transfer Learning), a simple proposed CNN,*** and an ***XGBoost classifier*** using the features extracted from the proposed CNN model. Finally, we are combining the three CNN models (VGG16, EfficientNet-B7 and proposed CNN) in a stacked architecture to design the ***Ensemble model*** to predict the brain tumour classification.

# **Methodology**

To achieve solutions in all industries in a faster and more efficient manner, it is imperative to follow a proper methodology in today's rapidly evolving world. There are many types of methodologies being followed by various businesses such as the Waterfall model, Iterative and Incremental Developmental model, Spiral developmental model, Agile model, Prototype model, Rapid Application Development (RAD) model, V-shaped model, Feature Driven model, etc. Nevertheless, the key to any business's success is identifying the right methodology based on their business needs and implementing it.

In our project, we have implemented the iterative and incremental methodology for classifying brain tumors. The incremental and iterative development process is closely associated with Agile project management, most notably the Scrum methodology. This is because it aligns with one of the key pillars: responding to change over following a set plan (What is Iterative and Incremental Development? (n.d.)).

The requirements of the project are divided into a number of PIs (Project Increments). Each PI focuses on one or more features, which are further divided into various tasks. To complete a feature, all of the tasks associated with that feature must be completed. These features can be developed incrementally and delivered over time. Core features such as dataset selection and analysis, loading, and pre-processing of the data are developed first, and the whole project is then developed by adding new features containing customized models designed specifically for the project in successive PIs.

As a result of this Incremental development model, researchers are able to make necessary modifications or changes early on in the project or business, rather than waiting until the end of the allotted time period has passed and the effort has been spent. A key aspect of iterative development is that improvements and changes are made continuously. This ensures that the end results can be delivered on time and be of a higher quality. The incremental/iterative methodology enables us to "deliver a better product at a faster and more efficient rate" (What is Iterative and Incremental Development? (n.d.)).

*Figure 2*, describes the detailed process of the Iterative and Incremental model used in our brain tumour classification project.

* As shown in Figure 2, we begin gathering all the requirements for our project after initial planning, including datasets, software requirements, and tools.
* As part of the requirement process, all the necessary libraries and packages for the dataset are loaded.
* As a next step in our process, we load and pre-process the data. During this process, the data will be cleaned, the images will be resized, and the dataset will be divided into training and testing sets.
* It is the next step in the process to conduct detailed research and analysis of various CNN models in order to apply them to the classification of brain tumors.
* Upon completion of the analysis, we will be able to finalize the CNN models required for our dataset.
* As a result, we can now begin designing the first CNN model, in order to classify our brain tumour data.
* Once we have designed the first CNN model, we can begin the implementation process and begin testing and evaluating the model. In case of any changes or modifications, or hyper-tuning the model can be performed in this stage. This helps us to enhance the model's accuracy and resolve errors in the early stages of the project.
* Similarly, the same process is iteratively implemented incrementally for the other CNN models and tested.
* Finally, the accuracy scores for all the CNN models are calculated and the best performing model for the brain tumour classification is obtained.

A picture containing text, businesscard, vector graphics

Description automatically generated

Figure 2 - Iterative & Incremental Model (Iterative Development: A Starter’s Guide, 2021)

# **Requirements**

This section describes the dataset and other requirements needed to implement the MRI image classification of brain tumour using deep learning. The data used in our project is secondary data which was already gathered and is accessible on the online community platform called ***Kaggle***.

**Dataset Link:** <https://www.kaggle.com/datasets/masoudnickparvar/brain-tumor-mri-dataset> (Nickparvar, 2021)

**Dataset Description:**  The dataset consists of MRI images from patients with three different types of brain tumours are gliomas (brain cancer), meningiomas (inflammation in the membranes surrounding the brain and spinal cord), and pituitary tumour (secretion of hormones by an organ other than the adrenal gland) along with no tumour class images. The dataset consists of the images in two folders named ***training*** and ***testing***. Each directory contains 4 folders with brain MRI images such as Glioma, Meningioma, Pituitary, and no tumour. In the training directory there are 5712 images and 1311 images in the testing directory. The hardware and system requirements are listed below:

## **Hardware Requirements**

* NVIDIA graphics card with CUDA Compute Capability version 3.5 to 8.6. A minimum of 8 GB of GPU memory is recommended for optimal performance, particularly when training deep learning models.
* NVIDIA GPU driver version: Windows 461.33 or higher, Linux 460.32.03 or higher.
* A CPU with the Advanced Vector Extensions (AVX) instruction set. In general, any CPU after 2011 will contain this instruction set (Install TensorFlow with pip. (n.d.), 2022).

## **System requirements**

* Ubuntu 16.04 or higher (64-bit)
* macOS 10.12.6 (Sierra) or higher (64-bit) *(no GPU support)*
* Windows WSL2 - Windows 10 19044 or higher (64-bit) (System Requirements: Deep Learning. (n.d.), 2022)

## **Software requirements**

* Python 3.7–3.10
* pip version 19.0 or higher for Linux (requires manylinux2010 support) and Windows. pip version 20.3 or higher for macOS.

We used Google Colab to implement CNN models and execute all of the coding in our project because it supports GPU for deep learning.

# **Analysis**

In this section, we will cover the workflow and sequence of steps involved in our deep learning project. We can divide the process into 4 blocks of process, and they are Data Engineering, Model Training, Model Evaluation, and Hyper Tuning.

## **Data Engineering – Data Acquisition and Data Preparation**

There are 2 processes involved in the data engineering step for any data science project. They are data acquisition and data pre-processing. ***Data Acquisition*** is the initial step in any data science project, in which we collect the data and analyse it. For deep learning project, the most important concern is to get a dataset with large number of labelled data. The data can be collected from various sources and in different formats. The various sources available for data acquisition are Publicly available data repository, existing databases from the organizations, and web scraping/ APIs (Application Programming Interfaces). The most preferable and often used source is the publicly available datasets like Kaggle, UCI data repository, etc (Deep Learning Workflow, n.d.).

The steps involved in the ***data preparation or pre-processing*** block are data cleaning, scaling features (normalization or standardization), managing categorical data, labelling the data, and lastly splitting them into train, validation, and test sets. Before splitting the dataset, the pre-processing measures employed in our paper includes labelling all images with corresponding class names and resizing them to size of 128 x 128. Once the data is ready, we split the dataset into training set, validation set and testing set. In our paper, we have divided 70% of the data for training and 30% for testing set. Validation set will be separated within the training set, and we are using 20% of the data from the training set for validation purpose (Deep Learning Workflow, n.d.).

Table 1 - Train and Test Split Count for Experiment 1 and Experiment 2

## **Model Training**

During model training, we build our model and train them using the training dataset. In our paper, we built all the CNN models and then complied them using cross-entropy categorial for calculating the loss, Adam optimizer to optimize the loss and we have used accuracy as our metric to calculate the performance of the model.

## **Model Evaluation**

After training the model, we may evaluate it using the validation and test sets. This allows us to understand how well our model will perform in previously unseen or new data. In our paper, to evaluate all the CNN model's performance, we are using a confusion matrix, cross-entropy loss value, and a classification report, which includes the model's accuracy, precision, F1-score, and recall.

***Accuracy*** is used to assess how effectively a model performs. It is scaled from 0 to 1, with the value closest to 1 being the best. It can alternatively be defined as the proportion of successfully predicted classes to the total number of predictions. To make a better model evaluation, we may need to examine other metrics such as precision and recall, because if the dataset is imbalanced, then it may lead to incorrect model evaluation (Agrawal, 2021).

***Confusion Matrix*** is a matrix with the x-axis being the actual values and y-axis is the predicted values. It is also defined as a table that is used to evaluate classification model’s performance on the test dataset for which the true values are known. The confusion matrix is made up of four values and they are True Positive (TP), False Positive (FP), True Negative(TN), and False Negative(FN). Confusion matrix are very helpful for measuring Recall, Precision, and Accuracy (Agrawal, 2021).

Figure 3 - Confusion Matrix

|  |  |
| --- | --- |
| We predicted the value is negative and its true.  **TN** | The value predicted is positive and its false (Type 1 Error)  **FP** |
| **FN**  We predicted the value is negative and it is false (Type 2 Error) | **TP**  The value predicted is positive and its true. |





***Precision:*** It is the number of correctly predicted classes (TP) to the sum of all the predicted positive classes by the model (TP+FP). We can also say that it is considered only for the positive classes from the confusion matrix. ***Precision = TP / (TP+FP). For precision we think of “predictions” as our base.*** (Agrawal, 2021)

***Recall (Sensitivity):*** It is the true positive ratio (TP), meaning ratio of true positives to all the positives in our dataset. Therefore, ***Recall =***  ***TP / (TP + FN). For recall, we think of “truth” as our base.*** (Agrawal, 2021)

***F1 – Score:*** It is the single metric that combines both precision and recall and it is calculated by the weighted average of precision and recall. The higher the F1-score the better the model. The classifier will get high F1-score only when both precision and recall values are high. The value scales from values 0 to 1. (Agrawal, 2021)

Text

Description automatically generated with low confidence

## **Hyper Tuning**

Once model training and evaluation are finished, we usually need to iterate upon initial parameters. Hyper tuning is used so that the model’s performance can be increased further at the same time, we try to minimize the loss. In this we can explore with different hyper tuning parameters like learning rates, batch sizes and number of epochs. If the metric scores are lower in both the training and testing set, then the model is underfitting. This means that we may need different learning rate or a larger mode. There are various factors that can be considered for tuning the model (Deep Learning Workflow, n.d.).

For instance, if the values on training and evaluation sets shows a discrepancy then the model is overfitting. In this case, we can use the dropout layer parameter or add regularization to reduce model overfitting issue. A typical approach is to start with a smaller model and gradually increase our hyperparameters until training and validation performance diverge, indicating that we have overfitted to our data. When our results are adequate, we will be able to implement our model to use (Deep Learning Workflow, n.d.)!

Figure 4 - Workflow of the Project and process

**Data Preparation (pre-processing):**

* Labelling Images
* Resizing image 128 x 128
* Splitting the dataset into training and testing set

**Data Acquisition:**

* Public Dataset from Kaggle

**Model Evaluation:**

* Choosing evaluation metric
* Evaluating the model performance on validation and test set.

**Model Training:**

* Build CNN model
* Define model architecture and hyper parameters
* Train the model

**Hyper Tuning:**

* Tune hyperparameters
* Analyse the model’s behaviour
* Add regularization

Unsatisfactory

# **Design**

In our paper, we are using the 5 different models for classifying brain tumours such as Glioma, Meningioma, Pituitary and No Tumour.

1. Proposed CNN Model
2. CNN + XGBoost - The features extracted using the proposed CNN model as the input for the XGBoost classification algorithm.
3. Transfer Learning - EfficientNet-B7 and VGG-16.

Ensemble model = Proposed CNN model + EfficientNet-B7 model + VGG-16

## **Overview of CNN Model**

The Convolutional Neural Network (CNN or ConvNet) is a type of artificial neural network that can be used in various applications such as image, speech and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language processing, brain-computer interfaces, and financial time series. The advantage of using CNNs for image classification is that the convolutional layer reduces the high dimensionality of images without losing any information. Also, it is a feedforward network, meaning that information flows from its inputs to its outputs only in one direction. They are influenced by the visual cortex of the brain, which is composed of alternating layers of simple and complex cells (Hubel & Wiesel, 1959) (Hubel & Wiesel, Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex, 1962) (Rawat, W., & Wang, Z. ).

There are different variations of CNN architectures. In general, the main types of layers in the CNNs are Convolutional, pooling layers (or subsampling layers), and fully connected layers (or Classification Layers), and they are grouped into modules.Figure 5 – Overview of CNN model for Brain Tumour Classification

### **The Convolutional Layer**

The convolutional layers serve as feature extractors, and thus they learn the feature representations of their input images. Inputs are convolved with the learned weights in order to compute a new feature map, and the convolved results are sent through a nonlinear activation function. Greyscale images are represented as n x n size (height x width) and each pixel’s value is between 0 to 255. In the convolutional operation to extract the features from an image, a filter of size m x m slides over the entire image.

For example, when a 5 x 5 grey scale image convolves with a 3 x 3 filter, we get a 3 x 3 image. Initially, the 3 x 3 filter matrix gets multiplied by the first 3 x 3 size of our greyscale image, then we shift one column right up to the end, after that we shift one row, and so on. If we have N x N image size and F x F filter size, then after convolution result will be calculated as mentioned in the below equation.

(N x N) \* (F x F) = (N-F+1)x(N-F+1)

(5 x 5) \* (3 x 3) = (5 – 3 +1)x(5 – 3 +1) = 3 x 3

More formally, the kth output feature map Yk can be computed as

Yk = f(Wk ∗ x)

In the above equationYk = f(Wk ∗ x), the input image is denoted by x; the convolutional filter related to the kth feature map is denoted by Wk; the multiplication sign in this context refers to the 2D convolutional operator, which is used to calculate the inner product of the filter model at each location of the input image; and f(·) represents the nonlinear activation function (Yu, D., Wang, H., Chen, P., & Wei, Z., 2014) (Rawat, W., & Wang, Z. ). The convolutional layer also contains Nonlinear activation functions like Rectified Linear Units (ReLU) which are used for the extraction of nonlinear features. The ReLU activation is performed to make all the negative values zero.

A picture containing text, diagram, screenshot, plan

Description automatically generated

Figure 5 – Overview of CNN model for Brain Tumour Classification (Mdpi.com, 2022)

### **Activation Layer:**

***Rectified Linear Unit (ReLU)*** is an activation layer used in the hidden layers(convolution layers) for better computation performance and to avoid gradient vanishing gradient problems. This means the negative values in the filtered image are removed and replaced with zeros in this layer. By doing so, we can try to keep the values from adding up to zero. The ReLU layer only activates a node if the input exceeds a specified threshold. If the input is less than zero, the output is zero; however, if the information exceeds a certain threshold. It is proportional to the dependent variable. In Figure 6, the left side of the images represents the ReLU function and right side is table with the output of ReLU layer for one feature (Working of Convolutional Neural Network - Javatpoint, n.d.).

Figure 6 - ReLU Function and Output of ReLU layer for one feature (Working of Convolutional Neural Network - Javatpoint, n.d.)

Table

Description automatically generated

### **Pooling Layers:**

Pooling layer helps us to reduce the image stack to a smaller size. The steps performed in this layer are as follows:

1. Selecting a ***window size*** (usually 2 or 3)
2. Selecting the ***stride*** ***size***
3. ***Move*** your Window ***over*** your ***filtered images***.
4. Take the ***highest*** value from each ***window***.

The pooling layer is applied between two convolutional layers, its main function is to reduce the spatial resolution of feature maps and thus achieve spatial invariance to input distortions and translations (Rawat, W., & Wang, Z. ). When applying Fully Connected layers after Convolutional layers without applying average pooling or max pooling, it can be computationally expensive. Initially, average pooling was used as a common practice, in which the average value of all the pixels in a kernel of a feature map is selected and applied to the next layer.

However, in recent models, Max Pooling is used widely to help overfitting by providing an abstracted form of presentation. As well, it reduces the computational cost by reducing the number of parameters to learn and provides basic translation invariance to the internal representation. In Max Pooling, the maximum pixel value of the kernel is selected.

A diagram of a pool

Description automatically generated with low confidence

Figure 7 - Difference between Average and Max Pooling (Aljaafari, Nura, 2018)

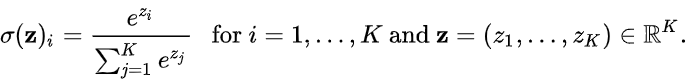
### **6.1.4. Fully Connected Layers**

The final layer is the fully connected (FC) layers and this is placed before the output layer. A fully connected layer combines the outputs of the topmost convolutional layer into a 1D feature vector. In this layer, the input image from the previous layers is flattened and fed to the FC layer. The flattened vector undergoes few more FC layers and in this stage the classification process starts to take place.

***Activation Functions:***As far as activation functions are concerned, there are a number of common ones including the ReLU, SoftMax function, tanH, and the sigmoid function. Every one of these functions has a specific application that can be applied. A CNN model for binary classification is generally better performed with the sigmoid and SoftMax functions. However, for a CNN model for multi-class classification, SoftMax is generally used as a function. A CNN model uses activation functions to determine whether or not a neuron should be activated as a result of a given input. (Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network., 2022, July 28)

ReLU is used in the hidden layers (convolutional layers) whereas ***Argmax and SoftMax are the activation layers*** that reside on the FC layers. In Argmax, the maximum value is derived from the output of FC layers, which might be great for the test data, but not for training our model. The cross-Entropy loss function needs the predicted probability, and that level of granularity is not achieved with Argmax.

Hence, we need SoftMax in which the raw outputs from the FC layers are represented in a normalized form; so that they become predicted probabilities. As mentioned in (contributors, 2022), prior to applying SoftMax, some vector components could be negative, or greater than one; and might not sum to 1; but after applying SoftMax, each component will be in the interval (0,1), and the components will add up to 1, so that they can be interpreted as probabilities. Furthermore, the larger input components will correspond to larger probabilities. and it is given by



In the above equation, z is the input vector, ezi is the standard exponential function for the input vector, K is the number of classes in the multi-class classifier and ezj is the standard exponential function for the output vector.

***Loss Function:*** The loss function is used to measure the error between the predicted values and the true values. The loss or the error is measured as numbers between 0 and 1. The closer the value to zero, the better the model. Always we try to achieve the model with the loss value closer to 0, the errors are minimized using any optimization method. In our paper, we have used the Cross-entropy loss metric to evaluate the model’s performance. It measures the difference between the discovered probability distribution of the model and the predicted distribution (Khan, Jue, Mushtaq, & Mushtaq, 2020). After running the data in the model, we obtain raw output. These raw outputs are run through the SoftMax function (as it produces values between 0 and 1). Then the cross-entropy is calculated by the following formula (Neural Networks Part 6: Cross Entropy. (n.d.)., 2022),

***bservedc* x *log(Predictedc)***

In the above equation of summation, M represents the number of output classes, in our case we have M=4 as we have 4 output classes (either glioma, meningioma, pituitary, and no tumour). To calculate the cross entropy for the class glioma, we expand the above equation as,

🡺 Cross Entropy (Glioma) = - ObservedGlioma x log (PredictedGlioma) - Observedmeningioma x log (Predictedmeningioma) - Observedpituitary x log (Predictedpituitary) – Observednotumour x log (Predictednotumour)

As it is for the class glioma, the equations can be further expanded as,

🡺 -1 x log (PredictedGlioma) – 0 x log (Predictedmeningioma) – 0 x log (Predictedpituitary) – 0 x log (Predictednotumour)

🡺 -1 x log (PredictedGlioma)

🡺 -1 x log(0.57) = 0.56

🡺 Therefore, Cross Entropy for class Glioma = 0.56

* The predicted probability is produced by the SoftMax function and using that to find the cross-entropy for each class.
* Similarly for all the other classes, the raw output is run through the SoftMax function and from the result (predicted probability), cross-entropy for each class is found separately.
* Now to derive the total error for the model, we need to sum all the cross-entropy values.
* To minimize the total error, backpropagation is used generally to adjust the weights and biases.
* Weights are values that are associated with each input (feature). It is vital as it gives the importance of that corresponding input in predicting the final output. Features with more weights have more importance compared to the features with lesser value or closer to zero. Weights describes the relationship between a feature and the target value. Bias is used for shifting the activation function towards left or right, we can compare this to y-intercept in the line equation. Summation Function is to bind the weights and inputs together and calculate their sum (Ganesh, 2020).

***Dropout Layer:*** The overfitting problem occurs when a particular model performs so well on the training data that it causes the model to have a negative impact on its performance when it is used on new data in the future. There is a possibility of overfitting in the training data when all of the features are connected to the FC layers.

In order to resolve this issue, a dropout layer is used. In this layer, a few neurons are removed from the neural network during training. This results in a reduction in the model's size. 30% of the nodes are dropped from the neural network after passing a dropout of 0.3. As a result of the dropout, a machine learning model is able to perform better as it prevents overfitting by simplifying its network and this has an effect on its performance. As a result, neurons are dropped from neural networks during the training process. (Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network., 2022, July 28).

Figure 8 - Role of Weight and bias in Neural Networks (Ganesh, 2020)

Diagram

Description automatically generated

### **6.1.5. Architecture of proposed CNN Model**

In our paper, we have implemented a simple CNN model with 2 sets of 2 convolutional layers followed by max pooling layer, then fully connected layers to classify the brain tumours accordingly.

After pre-processing the images (resizing the images) to 128 x 128 image size, it is used as the input for the proposed CNN model. In the first set, we have 64 filters convolutional layer having a filter size of 3 x 3 as the first Conv Layer . The second layer consists of 32 filters convolutional layer having a filter size of 3 x 3. In both of these layers, the stride and padding are set to same. The two convolutional layers are followed by a Max pooling layer with 2 x 2 filter to get the maximum summary of the image.

The next set of layers have 32 and 16 filters convolutional layers with the same filter size of 3 x 3. And we applied a max-pooling layers on top of these convolutional layers to get most of the features of the image. Finally, we applied a fully connected dense layer along with the SoftMax output layer. The SoftMax layers helps to calculate the probability score for each class and classifies the final decision class whether the provided input image contains brain tumour or not and if what is the type of the tumour either glioma, meningiomas or pituitary.

Figure 9 - Architecture of Proposed CNN Model

Table

Description automatically generated

*Figure 10 - Proposed CNN Model*Diagram

Description automatically generated

## **Transfer Learning**

* It is a common assumption that machine learning techniques works well when the training and test data are drawn from the same distribution and feature space.
* In the event that the feature space and distribution change, most statistical models have to be rebuilt from scratch using newly collected training data.
* In many real-world applications, it would cost a lot of money and will be time consuming, and impossible as we would need to collect all the necessary training data and then rebuild the models.
* To overcome these issues, *knowledge transfer* or *transfer learning* between task domains would be beneficial and feasible solution (Pan & Yang, 2010).
* In machine learning, Transfer learning refers to the process of using a model that has been trained on one problem to solve another similar problem.
* In our paper, we have selected VGG-16 and EfficientNet-B7 *using* transfer learning to identify their performance in detecting and identifying meningioma, pituitary and glioma brain tumours.

Figure 11 - Overview of Transfer Learning implemented in Brain Tumour Classification (Mdpi.com, 2022)

A picture containing text, diagram, screenshot, plan

Description automatically generated

### **VGG Model - Architecture**

VGG was proposed by Karen Simonyan and Andrew Zisserman in 2013 (Simonyan & Zisserman, 2014)and presented the actual model based on the idea in the 2014 ImageNet Challenge.

* In VGG model, the convolution filters are 3x3 throughout the model with a stride of 1 pixel made it to be a standout when compared with other models. For example, the filters in the first layers was11x11 with stride 4 in AlexNet and 7x7 with stride 2 in ZFNet.
* Also, in AlexNet, there are totally 3 sets of convolution layers followed by pooling layers, also known as feature extractors.
* However, in VGG model there are totally 5 sets of convolution and pooling layers for feature extraction.
* Adding more layers is not all that we need to do, but we should also consider the additional 7x7 non-linear activation layers which will make the networks to converge faster, and this has a huge impact on the model's performance.

1st stack of Convolution Layers: In the first stack of layers, the input image is passed through 2 convolutional layers with 64 kernel filters, and this is followed by ReLU activation layer. Then it is followed by a max pooling layer with window size 2 and with a stride of 2 pixels. In this layer, the image size is reduced half. Thus, at this stack the size of the image becomes 112 x 112 x 64 (Team, 2021).

2nd stack of Convolution Layers: In the second stack’s configuration is similar to the previous stack, except for the size of filters are changed to 128, thus making the dimension as 112 x 112 x 128. After passing through the pooling layers, the resulting output size is reduced to 56 x 56 x 128.

3rd stack of Convolution Layers: In is there are three convolutional layers and a max pool layer, and the filter size applied here are 256, making the output size of the 3rd stack as 28 x 28 x 256.

4th stack of Convolution Layers: In is stack, the filter size is changed to 512, making the output size of the 4th stack to be 14 x 14 x 512 after applying max pooling with stride of 1 and ReLU activation.

5th (Final) stack of Convolution Layers: In the final stack of three convolutional layers, the filter size and the max pooling stride are same as the previous stack, making the output size of the final stack to be 7 x 7 x 512. The five stacks of convolutional layers to perform the feature extraction process, followed by three fully connected layers and a flattening layer in between.

Diagram

Description automatically generated

Figure 12 – A Standard VGG-16 Network Architecture (Younis, Qiang, Nyatega, Adamu, & Kawuwa, 2022)

Text

Description automatically generated

Figure 13 - VGG-16 Architecture (Tammina, 2019)

### **EfficientNet Model**

Generally, the models are typically too wide, deep, or extremely detailed. In the beginning, these characteristics are helpful to the model, but they quickly become saturated, and the model produced simply has more parameters, making it inefficient. They are scaled in a more principled manner in EfficientNet, with everything gradually increasing in performance. Chart, box and whisker chart

Description automatically generated

Figure 14 – Model Scaling (Tan & Le, 2019)

A ConvNet can be scaled in a number of different ways. By altering the network depth, ResNet, for instance, can be reduced or increased (ResNet-18 or ResNet-200). Similarly, the number of network channels (width) can be changed for MobileNets and WideResNet. As a result, they were concentrating on the architecture and network (convolutional layers - Fi operator). Model scaling, on the other hand, focused on increasing the image's height and width(resolution (Hi, Wi), as well as the width(Ci) and length (Li) of the network, without altering the baseline network's predefined Fi (convolutional layer) (Tan & Le, 2019).

* Width scaling - More feature maps are added in each layer.
* Depth scaling - More layers are added to the network.
* Resolution scaling - Increase the resolution the size of the input image, instead of downsizing it.
* Compound scaling, we extend all three dimensions at the same time at a specified ratio.

Tan and Le's research shows that when one of the three dimensions is scaled, the other two must also be scaled since they are interdependent, requiring a coordinated balance between the various scaling dimensions, as opposed to conventional single-dimension scaling. (Tan & Le, 2019).

Table 2 - EfficientNet-B0 to B7 – Structure

Table

Description automatically generated

In compound scaling method proposed by Tan and Le, they use a compound coefficient ϕ, to uniformly scales the width, depth, and resolution of the network in a logical manner as described below (Tan & Le, 2019):

Depth: d=αϕ

Width: ω=βϕ

Resolution: r=γϕ

s.t. α . β2 . γ2 ≈ 2

α≥1, β≥1, γ≥1

The depth, width, and resolution of the compound coefficient ϕ are uniformly scaled, while the constant α,β,γ were determined through a grid search as mentioned in the above equation (Tan & Le, 2019). Floating-point operations per second (FLOPS) in a convolutional operation is related to the d, w2, r2 , this means that FLOPS will be doubled when doubling network depth, however FLOPS will be increased by four times when doubling network width or resolution. Scaling a ConvNets with respective to the α≥1, β≥1, γ≥1 will increase the FLOPS by α . β2 . γ2. In the research paper, the constraint is α . β2 . γ2 ≈ 2, such that for any new ϕ, the total FLOPS will be approximately increased by 2ϕ .

As previously stated, model scaling does not increase the number of predefined convolutional layers in the baseline network. As a result, having an efficient and good baseline network is critical. They have evaluated their scaling strategy using existing ConvNets in this paper (Tan & Le, 2019). However, in order to increase the likelihood that they will be able to demonstrate the viability of their scaling strategy, they have also developed an additional portable size benchmark known as EfficientNet.

We apply our compound scaling method in two steps to scale it up from the base EfficientNet-B0:

STEP 1: Fix ϕ = 1, thinking twice more resources available, and do a small grid search of α, β, γ.

* EfficientNet-B0 are α = 1.2, β = 1.1, γ = 1.15.
* Under the constraint of α · β2 · γ2 ≈ 2.

STEP 2: Then the values of α, β, γ are fixed as constants and scale up baseline network with different ϕ using the above equation to obtain EfficientNet-B1 to B7.

It is possible that better performance might be achieved by searching directly around a large for "α", "β" ,"γ" but the cost of searching becomes much costly on larger models. Hence this issue is solved by starting from scratch with small network. In our paper, we have used the EfficientNet – B7 model to classify the different categories of brain tumours using Transfer learning.

## **Ensemble Model - Bootstrap Aggregation**

For many complex machine learning problems, Ensemble models and methods are considered the state-of-the-art solution. In this method, the performance of the prediction is increased by training multiple models and combining their outcomes(predictions).

In the article by Omer Sagi and Lior Rokach (Sagi & Rokach, 2018), ensemble learning is defined as the umbrella term for methods that combines multiple inducers to decide especially in a supervised machine learning task. Based on a set of labelled data, a base learner (inducer) creates a model that simplifies the given data by using classification or regression. After training a model, it is used to predict future (unlabelled) data.

Any machine learning algorithms such as decision tree, neural network, linear regression model, classifiers etc, can be used as the ensemble base-learners. In our paper, we have used the Bootstrap Aggregation, it is a combination of sampling of data (Bootstrapping) and to form an ensemble model (Aggregation). In this method, we are using the same dataset to train the models instead of using random sub-sets of the original data.

The Ensemble models are most frequently used to improve the accuracy of models and the predictive performance of models due to several factors such as avoiding overfitting, computational advantage, and the ability to combine several models and extend the search space, thus enabling a better fit for the data space to be achieved. In our paper, we have stacked the VGG-16 model, EfficientNet-B7 model and a proposed CNN model to form an Ensemble Model architecture and this is clearly explained in the below mentioned *Figure 15*.

**Prediction 1**

**VGG – 16 (Model 1)**

***Final Predicted Output***

**Prediction 2**

**Proposed CNN (Model 2)**

**Data**

**Prediction 3**

**EfficientNet – B7 (Model 3)**

Figure 15 - Ensemble Model Architecture

## **Extreme Gradient Boosting**

In machine learning, tree boosting is one of the most effective and widely used because of its power of prediction and ease of use. It is a supervised learning algorithm and is used in both classification and regression. Boosting is an ensemble approach, in which merging the predictions from various models into single prediction. In Gradient Boosting, it is accomplished by taking each probability in a sequential manner and designing it based on its previous model’s residuals and then gradually increasing the model design using the sum of the previous models. In this the loss function is minimized using a gradient descent algorithm.

It is end-to-end tree-boosting system used to address many machine-learning challenges. The scalability of XGBoost is attributed to several key systems and algorithmic optimizations developed by Tianqi Chen (Chen, T., & Guestrin, C., (2016, August)). It's a gradient boosting technique with decision trees as "weak" predictors. For a given data set with n examples and m features D = {(xi, yi)} (|D| = n, xi ∈ R m, yi ∈ R), a tree ensemble model (shown in Fig. 1) uses K additive functions to predict the output.

i = φ(xi) = fk(xi),fk ∈ F

In the above equation, where *F = {f(x) = wq(x)}(q : R m → T, w ∈ R T )* is the space of regression trees (also known as CART). Here q represents the structure of each tree and T is the number of leaves in the tree. Each *fk* corresponds to an independent tree structure q and leaf weights *w*. In our paper, we have used the proposed CNN model to extract all the features from the dense layer which is then used as input to the XGBoost classifier.

Figure 16 - XGBoost Classifier (How XGBoost Works - Amazon SageMaker, n.d.)

Diagram

Description automatically generated

# **Implementation**

Models were trained on Google Colab with runtime as GPU. In this paper, we used the "tensorflow" and "keras" libraries to develop our distinct classification (CNN)models to detect several types of brain tumour in the MRI image dataset. The following steps briefly explain the implementation of the code and all the CNN models on our dataset.

**EfficientNet-B7, VGG-16 and proposed CNN Model Implementation:**

* After importing all of the relevant libraries and packages.
* By mounting the Google Drive the dataset is loaded and the image files are read and resized.
* The size of the input images is resized to 128 × 128.
* As a further step, the dataset is divided into Training (70%) and Testing sets (30%). We begin our model building after partitioning the data into training and testing sets.
* Then we have design the EfficientNet-B7 model and VGG-16 model by using the transfer learning with the weights trained on the ImageNet dataset and the input image shape is 128 x 128 x 3.
* For the EfficientNetB7 and VGG-16 models, we have designed the dropout layer rate as 0.3, and added the global average 2D pooling, with 4 classes in the dense layer, with SoftMax activation the model is trained.
* The next is to compile the model using categorical cross-entropy for calculating the loss, for the loss optimization we have chosen Adam optimizer with learning rate of 0.0001 and using accuracy as the metric to evaluate the model’s performance.
* To minimize over-fitting, we constructed the dropout layer at the rate of 0.3, such that few neurons (training parameters) in the final layers are dropped.
* In addition, as a model checkpoint, we are monitoring the validation accuracy and saving the model to utilise in the future.
* Once the model is complete, we can try to fit it to the training data and run it for 10 to 12 epochs.
* Using the graph and epoch output for the EfficientNet-B7 model in Figure 18, we can see that after the first epoch, the training accuracy has increased from 81% to 95.73% and the validation accuracy has increased from 78% to 91.6%. The graph shows that the model has a consistent flow throughout the completion of all epochs.
* In a similar manner, we have designed and implemented the VGG-16 model.
* Similarly, we designed and executed a proposed CNN model with all of the specifications specified in the preceding section 0.
* Figure 17, shows the training, validation accuracy and loss graph for 10 epochs.

Figure 17 - Proposed CNN Model Graph

Chart, line chart

Description automatically generated

Figure 18 - EfficientNet - B7 Model - Training and Validation Accuracy & Loss Graph

Chart, line chart

Description automatically generated

**XGBoost Implementation**:

* The features extracted from the proposed CNN model’s dense layer are used as the input to the XGBoost classifier model.
* Then we train the XGBoost model with these input features and predict the accuracy of the testing data.
* Then we have classified the images using XGBoost algorithm and produced the classification report with accuracy, recall, precision and F1 scores.

**Ensemble Model - Bootstrap Aggregation Implementation:**

* In this type of ensemble model, we are loading the saved proposed CNN model, EfficientNet-B7, and VGG-16 model as model1, model2 and model3.
* All the three models are trained on the same dataset.
* Then have created a list with these 3 models and iterating though the list to predict the unseen data.
* Finally, we can use the argmax for the sum of predicted probabilities from each model for each class in the ensemble model for predicting final class output.

# **Experimental Results and Discussions**

**CNN - Strengths:**

* In machine learning algorithms, features are selected with human supervision. However, in CNN human supervision is not required for selecting the important features.
* Provides high accuracy compared to the ML algorithms in image classification.
* When compared with normal neural networks, CNN requires minimum computation power (Convolutional neural network., n.d.).

**CNN – Weakness:**

* To get a higher accuracy and increase model’s performance, CNN needs large training datasets.
* If proper resources, such as a GPU, are not available, the time required to train the model increases.
* Max pooling procedures can m\slow down the process.
* CNNs struggle to recognise and classify images with varying positions because they struggle to encode the alignment and positioning of the objects (Convolutional neural network., n.d.).

**EfficientNet-B7 - Strengths:**

* In EfficientNet model, increasing all three dimensions (width, height, and resolution) contributes to improved accuracy in larger networks.
* The constraint of single dimension scaling is that it reaches 80% accuracy and then quickly saturates.
* EfficientNet models outperform other ConvNets greatly. EfficientNet-B7, in instance, delivers new state-of-the-art 84.4% top-1 accuracy as described in **Error! Reference source not found.**.

Figure 19 - Model Size vs ImageNet Accuracy (Tan & Le, 2019)

A picture containing graphical user interface

Description automatically generated

**EfficientNet-B7 - Weakness:**

* Computational constraints include drive I/O, processing capacity, storage capacity, and so on.
* Support in low-level libraries such as cuDNN and OneDNN is a significant determinant of computational efficiency and should be considered to increase the computational when creating a effective model.
* Though grouped convolutions (including depth wise convolutions) have lower FLOPS are not always faster to train than normal convolutions.
* EfficientNet are based on MBConv blocks which consists of two regular and one grouped convolutional layer. Depth wise is a special case of grouped convolutional layer and that it works on a single input channel poses a problem for low-level libraries, which may employ the channels as one of the elements to parallelize (Leeuwen, 2021).

**VGG-16 - Strengths:**

* Conceptually simple - typical architecture for CNNs.
* Most popular machine learning libraries have it and easy to use.
* Image classification performance is excellent, and it generalises effectively to related problems (Mike, 2022).
* Has been trained with millions of images.

**VGG-16 - Weakness:**

* Because of its large network size, computation time is longer to train its parameters.
* The model size is large (~500MB on disk), as it has more than 100 million trained weights.
* Cannot add layers like VGG-20 or VGG-50 because the backpropagation algorithm is used to update the weights of a neural network, which makes minor changes to each weight to reduce the model's loss. As a result, the training duration is greatly increased (Mike, 2022).

**XGBoost - Strengths:**

* Effective tree pruning
* Relative shrinking of leaf nodes
* Added randomization factors
* Handling missing values
* Selection of features automatically
* Speed and maximum performance
* Supports regularization

**XGBoost - Weakness:**

* Performs better for structed data and tabular data
* Overfitting is possible, especially if the trees are very deep and the data is noisy.
* Sensitive to outliers, as each classifier is compelled to correct the errors of the previous learners.
* Does not perform well on larger dataset with too many features.
* Not scalable since the predictors rely on previous predictors for correctness, making the approach difficult to simplify.

**Ensemble Model – Bootstrap Aggregation– Strengths:**

* When compared to other individual models, in ensemble method it has higher accuracy rate as it is a combination of various models.
* As different models are combined in this method, it is very effective as it can handle both linear and non-linear datatypes.
* Once again, since various models are combined together in this approach, over-fitting and underfitting problems are solved, as each different models have unique strengths and weaknesses (Makhijani, 2020).
* More stable and less noisy.
* By combining multiple models, the errors and inaccuracies of a single inducer can be compensated by other inducers, which is the principle that underlies Ensemble learning. As a result, this helps to improve the predictive performance of the Ensemble Model a better model than a single inducer model.

**Ensemble Model – Bootstrap Aggregation – Weakness:**

* The results of the model are hard to forecast and explain. As a result, the ensemble concept is difficult to sell and obtain useful market findings.
* As it is hard to learn, if we choose any wrong model selection it may lead to less performance than compared with single model.
* In terms of both memory space and computational timings, it is costly when compared to single models (Makhijani, 2020).

**Results:**

* In our paper, we have conducted two experiments, the main difference in each experiment is that the number of training data is different.
* In Experiment 1, we have used less training data when compared with Experiment 2. In the below mentioned table, we can see the number of training and test images used in both these experiments.

|  |  |  |
| --- | --- | --- |
| **Total Number of Images – 7023** | | |
|  | Training Image Set | Test Image Set |
| Experiment 1 | 4916 images (20% is used for validation set) | 2107 images |
| Experiment 2 | 5712 images | 1311 images |

* As a result of these experiments, we can see that the accuracy score for all the models is increased when the training image set size is increased.
* Table 3 and Table 4 shows the accuracy results of training and testing set obtained in experiment 1 and 2.
* Other than increasing the image size in the training set, there are no changes done in the model’s design.
* Thus, we have chosen to use the Experiment 2 with more training images to be the final experiment for our paper to classify brain tumour images.
* All the models have improved both the training and testing accuracy scores, and this shows that the size of the dataset has a vital role in getting higher and efficient performance of each model.
* The proposed CNN model’s test images accuracy is increased drastically from 88% to 93%.
* Once the proposed CNN model’s accuracy score is increased, the XGBoost model’s performance is also automatically increased, as they use the dense layer from the CNN model as their input.

|  |  |  |
| --- | --- | --- |
| **Experiment 2 - Results** | | |
| **Model Name** | **Training Accuracy** | **Test Accuracy** |
| Proposed CNN Model | 99.49% | 93.00% |
| EfficientNet-B7 | 99.93% | 98.40% |
| VGG-16 | 97.95% | 95.00% |
| CNN + XGBoost | 99.88% | 93.21% |
| Ensemble Model | 99.16% | |

* XGBoost has increased the test set accuracy score from 89% to 93%.

Table 3 - Experiment 1 Results  Table 4 - Experiment 2 Results

|  |  |  |
| --- | --- | --- |
| **Experiment 1 - Results** | | |
| **Model Name** | **Training Accuracy** | **Test Accuracy** |
| Proposed CNN Model | 96.40% | 88.80% |
| EfficientNet-B7 | 98.39% | 96.96% |
| VGG-16 | 97.40% | 93.93% |
| CNN + XGBoost | 98.08% | 89.36% |
| Ensemble Model | 96.68% | |

* The VGG-16 model’s score is increased from 93% to 95% in the test set.
* The test set accuracy for EfficientNet-B7 has increased from 96% to 98% after increasing the training image size.
* Also, this in turn leads to the increase score of the ensemble model, as it uses the combined scores of all the individual models. In our case, the score is increased from 96.68% to 99.16%.
* As a result, from our experiments, we can conclude that the Ensemble model is the best performing model with higher accuracy scores.

Table 5 - Test Set - Accuracy Score for Experiment 1 and 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Image set**  **Accuracy Score** | **EfficientNet B7** | **VGG-16** | **Proposed  CNN** | **Ensemble  Model** | **CNN +  XGBoost** |
|  |
| Experiment 1 | **97.63%** | 93.93% | 88.80% | 96.68% | 89.36% |  |
| Experiment 2 | 98.86% | 97.18% | 93.14% | **99.16%** | 93.21% |  |

* In the Experiment 1, when compared with all the model’s results EfficientNet-B7 was performing high.
* However, after changing the training imge set size in Experiment 2, Ensemble model has achieved an highest accuracy score compared to all other models.
* From the Table 5, we can see that after increasing the accuracy socre for all models in experiment 2, the score of Ensemble model is increased from 97.82% to 99.16%.
* As a result, for our brain tumour MRI image dataset, Ensemble Model performs well while compared to all other models in our paper.
* Both the Experiment 1 and Experiment 2 code files and PDF documents have been uploaded in my GitHub page and can be accessed using the following link <https://github.com/GoldenHari/Brain-Tumour-classification-using-Deep-learning>

# **9. Project Management**

For any project, planning them is really important because without a proper plan we cannot be able to complete the process within the timeline. The best practice is to always start the project with a scheduled table of tasks to be completed or any chart to track the progress.

## **9.1. Project Schedule**

For the successful completion of the project, I divided the tasks into 8 PI's (Project Increments) with corresponding features and tasks in each PI and attempted to complete them all within the time frame. In our project plan, each PI is divided into features, and features are further divided into smaller tasks. With the help of MS Excel, I have created a project schedule plan and attached in the and a chart as described in Figure.

Figure 20 - Project Schedule Chart

Figure 21 - Project Schedule

## **9.2. Risk Management**

There are many unanticipated events or risks in any project. Before beginning a project, it is advisable to categorize them according to their level of severity, and then implement the required mitigation measures to overcome the risks. The table below highlights some of the potential dangers associated with our research.

|  |  |  |  |
| --- | --- | --- | --- |
| Identified Risk  (Potential risk which can  occur during the project) | Impact  (What is the impact due to risk) | Mitigation Plan  (What is action plan if the risk occurs) | Risk Severity  (based on  impact) |
| Data Loss due to system crash | If the data is lost, it is extremely risky as we have to redo all of the work once again. | Regular backup of data in an external device or in the Google drive. | High |
| System Compatibility | Having a GPU allows us to implement CNN models considerably faster, efficient, and reliable. | In case of no GPU in our system, we can use Google Colab for free access to GPU for CNN model execution. | Medium |
| Delay in Data acquisition  from organization | Delays in the initial stage may result in delays in the subsequent processes. This may cause a delay in the completion of the entire project. | In this situation, we can acquire data from multiple publicly accessible  free dataset repositories. | Medium |
| Complexities faced in image  processing | Because image processing requires more time and computational resources, it may take to finish than intended. This could cause the project to be pushed back. | In this instance, the ideal strategy is to always try to implement each model with a small number of images,  which helps us to finish within the time frame. | Medium |

## **9.3. Quality Management**

Another key part of project management activities that must be achieved for any successful project is quality. In our project, we have implemented the project according to the process workflow mentioned in section 5. In addition to regular meetings with the supervisor, updates and reviews from the supervisor allowed me to keep the project on track and avoid delays and achieve great quality. Also, the project schedule was utilised as a guide for work activities and timetables, and the plan was updated weekly to ensure that all tasks were completed on time.

## **Social, Legal, Ethical and Professional Considerations**

In today's technology, data is the most sensitive and personal information. For any data science-related projects, social, legal, ethical, and professional considerations may arise as it is solely dependent on data. The data used in our paper is obtained from a freely accessible public dataset repository website (Kaggle). At the same time, the data does not contain any personally identifiable information such as name, age, or any other medical information, and the legal ownership of the data has been acknowledged. The university has carried out and authorised this project's ethical approval procedures.

# **Critical Appraisal**

When it comes to deep learning, one of the most well-known projects is brain tumour categorization. However, if we were able to improve on the prior study, it is always fantastic because it saves many people's lives by predicting brain tumours in a faster and more effective manner. This section provides an objective analysis of the project's strengths and weaknesses, as well as the learning outcomes obtained through this initiative.

The size of the dataset is the project's strength, the bigger the dataset, the better the model. In our dataset we have more number images to train the model and test them to estimate the model’s performance which was a strength to the research. By using an incremental and iterative process to handle my analysis, research, and documentation, I was able to feel that the workload was not too high and was able to complete the assignment on time. To the best of my knowledge, no previous analogous work has reached the accuracy scores and results that we did in our research.

We were able to run the model on a free GPU with the support of Google Colab. However, due to significant RAM utilisation in the Colab session, the session occasionally crashes. In this instance, we have to re-run the model again from the beginning when the runtime or kernel restarts, which was considered a slight inconvenience.

Throughout the course of this research, I have gained a tremendous amount of knowledge. For this project, I aimed to understand what a CNN is and how it works, as well as all of the terminologies and components included in the CNN. Similarly, I was able to comprehend the architecture, design, and operation of the five models used in our research.

# **Conclusions**

## 

## **11.1. Achievements**

As a result of this project, we were able to achieve our primary goal of establishing the optimal CNN model for brain tumour classification using MRI images. According to the experimental results, the Ensemble model outperformed the other CNN models with an accuracy score of 99.16%.All the objectives of our project have been successfully achieved by increasing the size of the training set images, the performance of the all the models is increased. As a result, it helped us to further enhance the accuracy of the Ensemble model as well as all other models.

## **11.2. Future Work**

Though we achieved our main objective, there are always possibility of improvement and for future work. First, even though we have trained our model with a big dataset, in the future we can still use a bigger dataset and train the models to get a much higher accuracy score. Due to the limited time, few other methods were not able to be tested and implemented. For future work, we can try using Batch Normalization layer and further improve the accuracy of the model. Also in future, methods can be extended to localize and mark the tumour region with bounding boxes and categorize them.

# **Student Reflections**

This assignment assisted me in learning more about data science and deep learning models, project planning and management, and time management. Simultaneously, with my supervisor's prompt direction and comments, I was able to do my best work. While conducting research for the paper, I had the opportunity to read and review a wide range of literature material and journals, which helped me understand deep learning techniques and various practical use cases that would not have been possible otherwise.

Furthermore, the knowledge gathered during the course of the project is unquestionably transferrable and applicable to my future career. The most difficult aspect of the project for me was the coding part, where the session kept crashing due to the long run time to receive the results for the models and the limitations of hardware sources such as online GPU resources. Fortunately, by adhering to the project plan, I was able to complete the project successfully and on time, while gaining a variety of soft and technical skills all along.

# **Bibliography and References:**

(2022). *Mdpi.com*. Retrieved from https://www.mdpi.com/applsci/applsci-12-07282/article\_deploy/html/images/applsci-12-07282-g002-550.jpg

Abiwinanda, N., Hanif, M., Hesaputra, S., Handayani, A., & Mengko, T. (2018). Brain tumor classification using convolutional neural network. *In World congress on medical physics and biomedical engineering*(doi: https://doi.org/10.1007/978-981-10-9035-6\_33), 183-189. Springer, Singapore.

Agrawal, S. K. (2021, July 20). Evaluation Metrics For Classification Model | Classification Model Metrics. *Analytics Vidhya.* Retrieved from https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/

AI vs. Deep Learning vs. Machine Learning: Beginner’s Guide. (n.d.). (n.d.). *AI vs. Deep Learning vs. Machine Learning: Beginner’s Guide. (n.d.)*. Retrieved from https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide

Aljaafari, Nura. (2018). Ichthyoplankton Classification Tool using Generative Adversarial Networks and Transfer Learning 10.25781/KAUST-K902H.

Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network. (2022, July 28). *UpGrad Blog. https://www.upgrad.com/blog/basic-cnn-architecture/#:~:text=other%20advanced%20tasks.-*.

Basics of CNN in Deep Learning. (2022). *Analytics Vidhya*. Retrieved from https://www.analyticsvidhya.com/blog/2022/03/basics-of-cnn-in-deep-learning/

Brain Tumors and Brain Cancer. (n.d.). (n.d.). Retrieved from https://www.hopkinsmedicine.org/health/conditions-and-diseases/brain-tumor#:~:text=Diagnosing%20a%20brain%20tumor%20usually

Chen, T., & Guestrin, C. ((2016, August)). Xgboost: A scalable tree boosting system. *In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785-794.

contributors, W. (2022, November 3). Softmax function. *In Wikipedia, The Free Encyclopedia*.

*Convolutional neural network.* (n.d.). Retrieved from Engati.: https://www.engati.com/glossary/convolutional-neural-network

*Deep Learning Workflow*. (n.d.). Retrieved from Codecademy.: https://www.codecademy.com/article/deep-learning-workflow

Ganesh, S. (2020, July 30). What’s The Role Of Weights And Bias In a Neural Network? Retrieved from https://towardsdatascience.com/whats-the-role-of-weights-and-bias-in-a-neural-network-4cf7e9888a0f

Gao, P., Shan, W., Guo, Y., Wang, Y., Sun, R., Cai, J., . . . Wu, Z. (2022). Development and Validation of a Deep Learning Model for Brain Tumor Diagnosis and Classification Using Magnetic Resonance Imaging. *JAMA Network Open, 5(8), e2225608–e2225608.* doi:https://doi.org/10.1001/jamanetworkopen.2022.25608

Habiba, S., Islam, M., Nahar, L., Tasnim, F., Hossain, M., & Andersson, K. (n.d.). Brain-DeepNet: A Deep Learning Based Classifier for Brain Tumor Detection and Classification. *International Conference on Intelligent Computing & Optimization. ICO 2022, 569. Springer, Cham*.

HealthITSecurity. (2022, January 25). *How Can Artificial Intelligence Change Medical Imaging?* Retrieved from https://healthitanalytics.com/features/how-can-artificial-intelligence-change-medical-imaging

How XGBoost Works - Amazon SageMaker, n.d. (n.d.). *Docs.aws.amazon.com*. Retrieved from https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost-HowItWorks.html

Hubel, D., & Wiesel, T. (1959). Receptive fields of single neurones in the cat’s striate cortex. *Journal of Physiology, 148(3)*, 574-591.

Hubel, D., & Wiesel, T. (1962). Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *Journal of Physiology, 160(1)*, 106-154.

Install TensorFlow with pip. (n.d.). (2022, November 20). Retrieved from https://www.tensorflow.org/install/pip#cpu

Iterative Development: A Starter’s Guide. (2021, October 21). *DistantJob - Remote Recruitment Agency.* Retrieved from https://distantjob.com/blog/iterative-development/

Kennedy, S. (2022, August 17). Deep Learning May Improve Identification of Brain Tumors. *HealthITAnalytics.* Retrieved from https://healthitanalytics.com/news/deep-learning-may-improve-identification-of-brain-tumors

Khan, H. A., Jue, W., Mushtaq, M., & Mushtaq, M. U. (2020). Brain tumor classification in MRI image using convolutional neural network. *Mathematical Biosciences and Engineering. DOI: http://dx.doi.org/10.3934/mbe.2020328, 17*(5: 6203-6216).

Khandelwal, V. (2020). The Architecture and Implementation of VGG-16. *Medium*. Retrieved from https://pub.towardsai.net/the-architecture-and-implementation-of-vgg-16-b050e5a5920b

Kim, D. (2020). Brain tumor detection: 2 novel approaches. doi:doi:10.20944/preprints202008.0641.v1

Kim, D. (2021). Brain Tumor Detection based on Ensemble Learning. doi:https://doi.org/10.20944/preprints202008.0641.v2

Leeuwen, C. v. (2021, July 9). *Why EfficientNet is not very computationally efficient*. Retrieved from Medium: https://medium.com/@casparvanleeuwen/why-efficientnet-is-not-very-computationally-efficient-505fc08087b9

Makhijani, C. (2020, October 6). Advanced Ensemble Learning Techniques. *Medium*. Retrieved from https://towardsdatascience.com/advanced-ensemble-learning-techniques-bf755e38cbfb

Mdpi.com. (2022). *Mdpi.com*. Retrieved from https://www.mdpi.com/applsci/applsci-12-05645/article\_deploy/html/images/applsci-12-05645-g001-550.jpg

Mike. (2022, June 28). Why is the VGG Network Commonly Used? *Nnart*. Retrieved from https://nnart.org/why-is-vgg-commonly-used/#Benefits\_and\_Drawbacks\_of\_VGG

Nayak, D. R., Padhy, N., Mallick, P. K., Zymbler, M., & Kumar, S. (2022). Brain Tumor Classification Using Dense Efficient-Net. *Axioms, 11(1)*(34). doi:https://doi.org/10.3390/axioms11010034

Neural Networks Part 6: Cross Entropy. (n.d.). (2022, April 29). Retrieved from https://www.youtube.com/watch?v=6ArSys5qHAU

Nickparvar, M. (2021). Brain Tumor MRI Dataset. doi:https://doi.org/10.34740/KAGGLE/DSV/2645886

Padding and Stride. (2022). *Dive into Deep Learning 1.0.0-alpha1.post0 documentation. (n.d.). D2l.ai.* Retrieved from https://d2l.ai/chapter\_convolutional-neural-networks/padding-and-strides.html#img-conv-reuse

Pan, S., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering 22(10), 1345–1359*. doi:https://doi.org/10.1109/tkde.2009.191

Pan, Y., Huang, W., Lin, Z., Zhu, W., Zhou, J., Wong, J., & & Ding, Z. (2015, August). Brain tumor grading based on neural networks and convolutional neural networks. *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE,DOI: 10.1109/EMBC.2015.7318458*, 699-702.

Post-NIPS\*95 Workshop on Transfer in Inductive Systems, n. (2022). *Socrates.acadiau.ca.* Retrieved from http://socrates.acadiau.ca/courses/comp/dsilver/NIPS95\_LTL/transfer.workshop.1995.html

python - How to plot an animated matrix in matplotlib. (2022). *Stack Overflow* . Retrieved from https://stackoverflow.com/questions/52067833/how-to-plot-an-animated-matrix-in-matplotlib

Rawat, W., & Wang, Z. . (n.d.). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation, 29(9), 2352-2449*.

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8*(4). doi:https://doi.org/10.1002/widm.1249

Sethy, P.K., Behera, S.K. (n.d.). A data constrained approach for brain tumour detection using fused deep features and SVM. *Multimedia Tools and Applications*(80(19), 28745-28760, DOI: https://doi.org/10.1007/s11042-021-11098-2).

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.* doi:https://doi.org/10.48550/arXiv.1409.1556

System Requirements: Deep Learning. (n.d.). (2022, November 20). *Www.l3harrisgeospatial.com*. Retrieved from https://www.l3harrisgeospatial.com/docs/deep\_learning\_SystemRequirements.html

Tammina, S. (2019). Transfer learning using vgg-16 with deep convolutional neural network for classifying images. *International Journal of Scientific and Research Publications (IJSRP), 9*(10), 143-150. doi:http://dx.doi.org/10.29322/IJSRP.9.10.2019.p9420

Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. *In International conference on machine learning*, 6105-6114. Retrieved from https://arxiv.org/pdf/1905.11946v3.pdf

Team, G. L. (2021). Introduction to VGG16 | What is VGG16? *Great Learning Blog: Free Resources What Matters to Shape Your Career!* Retrieved from https://www.mygreatlearning.com/blog/introduction-to-vgg16/#VGG%20Configurations

Tkachenko, N. (2021, October 6). Machine Learning in Healthcare: 12 Real-World Use Cases – NIX United. *Custom Software Development Company in US*. Retrieved from https://nix-united.com/blog/machine-learning-in-healthcare-12-real-world-use-cases-to-know/

Ullah, N., Khan, J., Khan, M., Khan, W., Hassan, I., Obayya, M., . . . Salama, A. (2022). An Effective Approach to Detect and Identify Brain Tumors Using Transfer Learning. *Applied Sciences, 12*(11), 5645. doi:https://doi.org/10.3390/app12115645

VGG-16 | CNN model. (2020). *GeeksforGeeks*. Retrieved from https://www.geeksforgeeks.org/vgg-16-cnn-model/

What is Iterative and Incremental Development? (n.d.). (n.d.). *Wrike*. Retrieved from https://www.wrike.com/project-management-guide/faq/what-is-iterative-incremental-development/

*Working of Convolutional Neural Network - Javatpoint*. (n.d.). Retrieved from Www.javatpoint.com.: https://www.javatpoint.com/working-of-convolutional-neural-network-tensorflow

Younis, A., Qiang, L., Nyatega, C. O., Adamu, M. J., & Kawuwa, H. B. (2022). Brain Tumor Analysis Using Deep Learning and VGG-16 Ensembling Learning Approaches. *Applied Sciences, 12*(14), 7282. doi:https://doi.org/10.3390/app12147282

Yu, D., Wang, H., Chen, P., & Wei, Z. (2014, October). Mixed pooling for convolutional neural networks. *In Proceedings of the 9th International Conference on Rough Sets and*, 364-375. doi:https://doi.org/10.1007/978-3-319-11740-9\_34