UNIVERSITY of BRADFORD

Faculty of Engineering and Informatics
Department of Computer Science

Detection of Brain Tumours Using Image Processing and Machine Learning

Dissertation Project Final Report

Ali Mirza

20006122

A thesis submitted in partial fulfilment of the degree of BSc (Hons) Computer Science

Supervisor: Dr Savas Konur

04 May 2023

Declaration

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others. The candidate agrees that this report can be electronically checked for plagiarism.

Ali Mirza

Acknowledgements

I would like to thank my project supervisor, Dr Konur for his project insights and advice.

Abstract

This report presents the gaps/limitations of some existing approaches to classify brain tumour MRI images using AI methods and tools. A deep-learning methodology is proposed and then implemented for multiclass classification as well as a prototype implementation for the binary classification of brain tumours using MRI images is compared and discussed in this report. The final deep-learning methodology encompasses deep features from the VGG-16 pre-trained model and GLCM features along with image processing and dimensionality reduction. The deep-learning model achieved an accuracy score of 98.20% and a Matthews correlation coefficient score of 0.9736 with a dataset of 54,000 images. The deep-learning model was deployed in the form of a web application using the Flask microweb framework. The prototype utilised three traditional machine-learning algorithms that are used for supervised learning (SVM, RF and KNN) and a dataset of 30,000 cerebral MRI images. The three classifiers achieved the following accuracy scores (79.68%, 91.26% and 81.81%) respectively with GLCM alone for feature extraction. The three classification algorithms were then implemented alongside three image processing techniques/algorithms (Gaussian Filtering, Image Thresholding and Erosion/Dilation) and this iteration achieved the following accuracy scores (84.14%, 88.55% and 83.12%).

Table of Contents

1 Introduction

Brain tumours occur due to the uncontrolled growth of abnormal cells in the brain. The unchecked growth of cells leads to either cancerous (malignant) or non-cancerous (benign) tissue forming into primary or secondary brain tumours. Tumours which begin from the brain tissue are primary brain tumours, whereas those that occur from cancer spreading to the brain from another area of the body (metastasis) are known as secondary brain tumours (Cancer Research UK 2019). At the core of artificial intelligence, machine learning can solve problems in the domain of medical diagnosis by identifying patterns in data. Due to the high mortality rate of brain tumours, it is imperative that tumours are detected as early as possible to increase treatment options and survival chances. To aid early detection, researchers are employing an array of machine learning and image processing techniques in the domain of brain tumour diagnosis. Despite the extensive research into machine-learning and deep-learning approaches for brain tumour detection, challenges and limitations in the techniques used still remain. Specifically, existing methodlogies for brain tumour classification lack the combination of deep features extracted from a pre-trained model and features from grey-level coocurrence matrices. Another limitation of the current techniques used is the limited size of the datasets being used to train neural network classifiers as most datasets that are used tend to have fewer than 30,000 images. Furthermore, existing state of the art solutions do not implement the global average pooling layer to reduce the thousands of feature maps into one feature map for each of the categories/classes and this in turn, prevents those who implement such solutions to reduce their classifiers complexity and therefore prevent overfitting their small dataset.

1.1 Project Description

This project comprises a machine-learning model combined with image-processing techniques that can detect the presence of brain tumours in MRI scans. The project encompasses a dataset, image-processing techniques, feature extraction methods and machine-learning/deep-learning techniques. Upon completion, the deep learning model will also classify brain tumours into three common types: glioma, pituitary and meningioma (*Primary and secondary brain tumours* 2019) and will also detect healthy MRI images. This project is a multi-class image classification problem as tumours are separated into more than two classes. I will utilise several Python machine learning frameworks like Keras, packages and publicly available brain MRI images.

1.2 Challenges

Many unique challenges still exist when applying deep-learning for brain tumour detection, although extensive research into deep-learning approaches for brain tumour detection already exists. The dataset alone for this project introduces several challenges. For instance, noisy MRI images, data scarcity, data imbalance and duplicates are issues that will doubtlessly arise and require countermeasures such as

data augmentation. Furthermore, data imbalance will cause misclassification of brain tumour types, and if other issues like noisy data also exist, it will exacerbate the misclassification challenge. Image-processing algorithms are likely to introduce further challenges like increased time consumption and could impact detection negatively rather than positively as expected. Therefore, each image-processing technique will be applied, and I will compare the model's performance to that in the absence of image-processing. In my assessment, the lack of MRI brain image datasets is the predominant challenge that I will encounter, and I will overcome this challenge by implementing transfer learning and image augmentation. The consequences surrounding the lack of brain MRI data are alarming. For example, the lack of data could lead to overfitting, which in turn causes false positives or false negatives.

In addition to the aforementioned challenges, the final product which comprises a deep-learning-based methodology as opposed to the traditional machine learning approach implemented previously poses further challenges. For example, the deep-learning approach is substantially more computationally extensive and therefore requires the necessary hardware that is capable of training a complex model with millions of parameters on a large dataset. I aim to negate this particular challenge by employing an MLOps platform to gain access to the latest hardware to reduce computation times for processes such as image processing, deep feature extraction, principal component analysis (PCA) and model training, tuning and evaluation. Another new challenge is that due to the lack of interpretability of neural networks and their “black-box” nature, hyperparameter tuning will prove to be both difficult and time-consuming as hyperparameters and neural networks, in general, are more dependent on experiments as opposed to clear theoretical results. Therefore, hyperparameter tuning will require several iterations/experiments and a trial-and-error approach to produce the best results rather than a fast theoretical approach/solution.

1.3 Aims and Objectives

In this report, I will propose an in-depth deep-learning methodology that addresses some of the limitations and gaps of other existing brain tumour detection solutions. Before presenting my methodology, I will put forward a literature review consisting of background/key concepts and a related work section that addresses and compares existing methods and their strengths or limitations. The background section includes a synopsis of different areas of deep learning, image processing, feature extraction/selection and dimensionality reduction. I will then describe the requirements and my deep-learning methodology as well as a validation methodology along with my final deep-learning model deployment workflow. Next, I present my current deep-learning model development workflow and final/complete implementation in the form of a web application along with the results achieved from each iteration of the deep-learning system. The fifth chapter consists of an extensive discussion of my results/findings as well as a critical comparison of the final web

application with the previous prototype implementation. Finally, I conclude the report and outline limitations and potential improvements regarding the final deep-learning model and the web application.

2 Literature Review

Machine learning and deep-learning are far-reaching disciplines of artificial intelligence which enclose methods that have applications in brain tumour detection. This section is a review of key concepts within machine-learning and deep-learning, and a discussion of related work and the incentive for applying artificial intelligence in the domain of brain tumour detection. This review addresses the novelty of this project by discussing gaps and limitations in related work.

2.1 Background Concepts

2.1.1 Machine Learning

Machine-learning is the discipline of study where computer programs can learn from datasets without the need for any precise programming. Unlike traditional programming approaches, machine-learning can withstand fluctuating environments as it generalises well to unseen data and does not require regular manual tuning to solve complex problems. The domain of machine-learning contains many subcategories: supervised, semi-supervised, unsupervised, reinforcement, online and batch learning. Supervised learning involves the training of machine learning algorithms on data with desired outcomes (labels), whereas semi-supervised learning involves a combination of labelled and unlabelled data and unsupervised learning algorithms require no labelled data (Géron 2019). Reinforcement learning comprises a learning system that can carry out actions based on its perception of an environment and get penalised or rewarded for its actions. The penalties and rewards allow the learning system/agent to form a strategy/action plan that increases its rewards and decreases its penalties. Batch learning is time-consuming as it requires training on all available data and not just a subset of the data. A batch learning system requires retraining on the original dataset and new data if the data related to a problem changes over time. Conversely, online learning-based systems involve training of data incrementally and are therefore quicker as they can learn from new data instances spontaneously without retraining. Each of the forms of learning involved in machine learning employs specific algorithms to automate the process of learning and identifying patterns in data. For instance, some supervised learning algorithms are support vector machine, random forest and k-nearest neighbours.

2.1.2 Deep Learning

Deep learning is a subset of machine learning that contains methods established on artificial neural networks. Deep-learning methods are exemplary for solving machine-learning problems of high complexity and large scale. Artificial neural networks draw inspiration from the neurons of the human brain (Zhao et al. 2022). Artificial neural networks are nodes that form one input layer, one output layer and several hidden layers. Each node in the neural network connects to another node with weights and a threshold value. Data is only passed along nodes if the output of a node is higher than the threshold value. Deep-learning methods outperform traditional machine-learning methods in image segmentation and image classification as they can automatically extract the most salient features from instances of data. There are a variety of deep-learning architectures with specific applications for a range of complex problems. Examples of neural networks include convolutional neural networks (CNN), generative adversarial networks (GAN) and deep belief networks (DBN) etc.

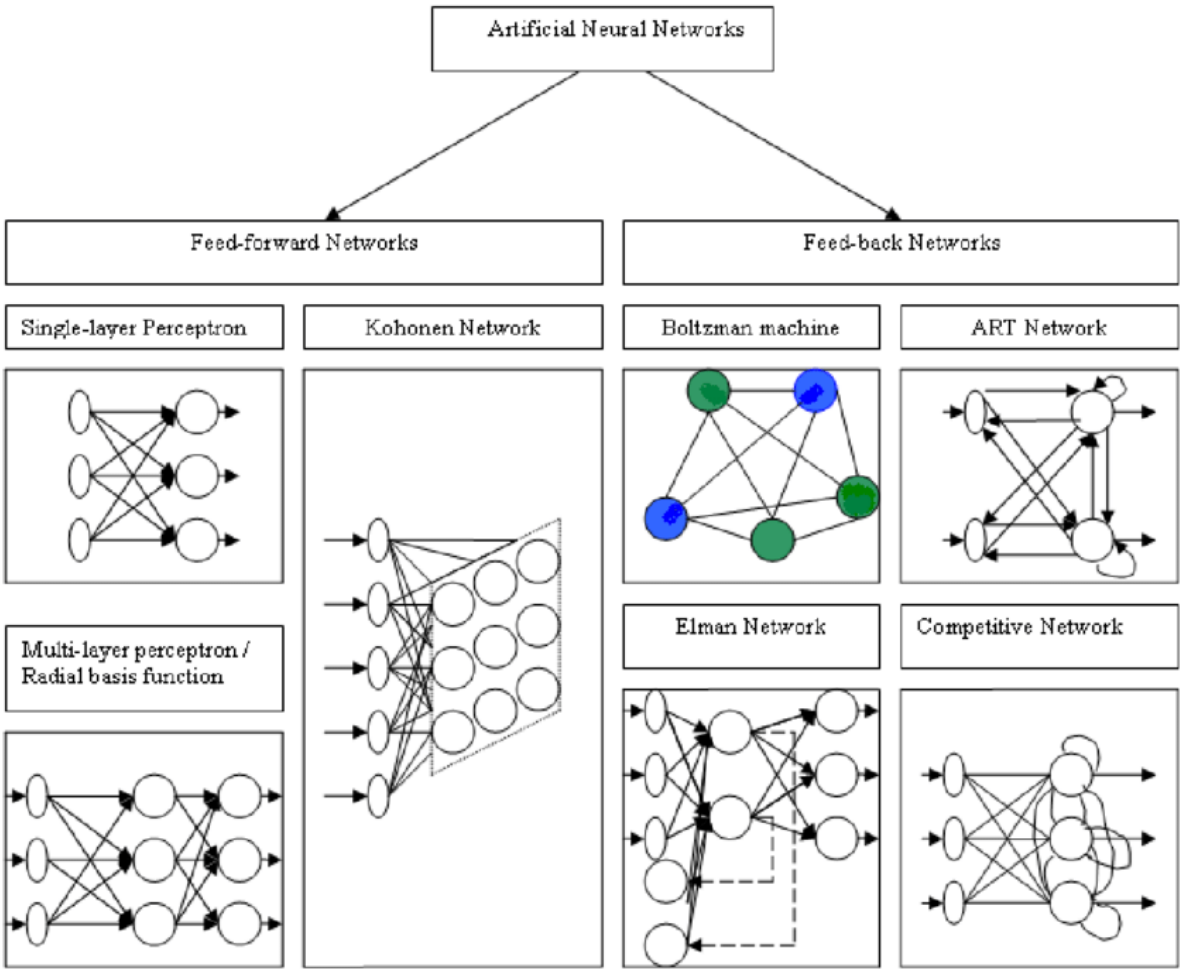


Figure 1: Types of Neural Networks (Rashid 2006)

2.1.3 Convolutional Neural Networks and VGG-16

Convolutional neural networks (CNN) are the most applicable deep-learning architecture for the image classification problem that I present a solution for. Convolutional neural networks are based on the brain's visual cortex which segments and processes visual data. The structure of a CNN consists of connected layers of nodes that contain activation functions which in turn map bias and weights to an image. The hierarchical structure of a CNN allows early layers to extract low-level features of the image and the subsequent layers can extract higher-level features from an image. Kernels are used alongside an activation function like Relu by convolutional neural networks to generate feature maps that show the detected features from the input image (Pereira et al. 2016). After the convolution layer, the pooling layer decreases computations by decreasing the dimensions of feature maps by applying average pooling or max pooling. Average pooling outputs the average feature values from a feature map, whereas max pooling only outputs the most outstanding features from a feature map. Finally, flattening converts the pooled feature map into a one-dimensional array/vector and the vector is then passed to the fully connected layers of the neural network. The fully connected layer then employs bias and weights to calculate probabilities for each label in the form of one-hot encoding. The sigmoid function is applied in the final layer before the output layer if the classification problem is binary, and the softmax function is applied if the classification problem is multi-class. The output layer simply represents the output classes of the neural network in the form of labels from one-hot encoding.

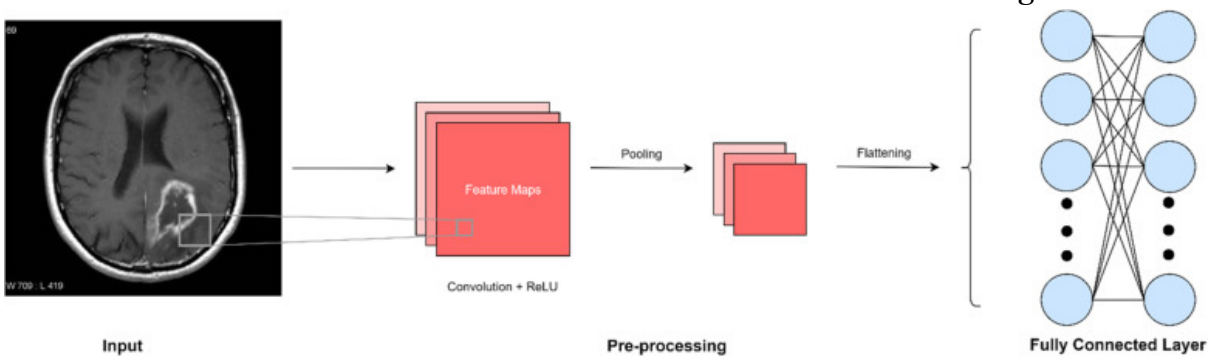


Figure 2: Input into one node within a CNN (Bhandari et al. 2020)

VGG-16 is a pre-trained CNN model which was proposed by (Simonyan and Zisserman 2014) that is made up of 16 layers and has 138 million parameters. VGG-16 is a reliable and firmly established model due to it being trained on 14 million images from the ImageNet dataset. The pre-trained weights from the VGG-16 model can be used to implement VGG-16 as a deep feature extractor by dropping/freezing its fully connected layers and modifying its input layer to receive images related to a specific task.

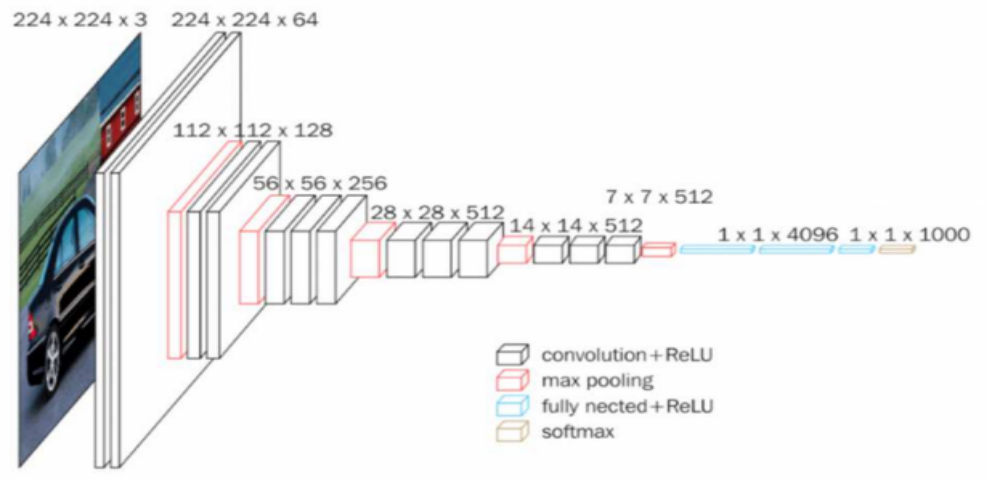


Figure 3: VGG-16 Architecture (Prawira et al.)

2.1.4 Deep Features from Transfer Learning and GLCM Features

Deep features consist of feature maps which are extracted from one of the last few convolutional layers of a convolutional neural network. Feature maps are output channels that result from the array of filters applied to each image in each different convolution layer of a convolutional neural network. Filters allow for low-level spatial feature extraction such as edges and intensity as well as high-level features consisting of representations of different objects in each image that are specific to the class the image belongs to. Feature map visualisations show how abstract the feature maps appear when compared to the original input as more filters and downsampling are applied after each consecutive convolutional layer and pooling layer.

Transfer learning involves repurposing the weights acquired by a pre-trained model and applying that pre-existing knowledge to a similar problem. The negative effects/limitations of limited data availability can be countered by employing transfer learning as pre-trained models are trained on large datasets from different domains and this in turn, guarantees that the existing knowledge can be transferred to another model to tackle a similar problem without obtaining a large dataset. For instance, VGG-16 can be used as a base model to extract features from images using its existing weights and then adjust the outputs from the VGG-16 model to suit a specific task/problem.

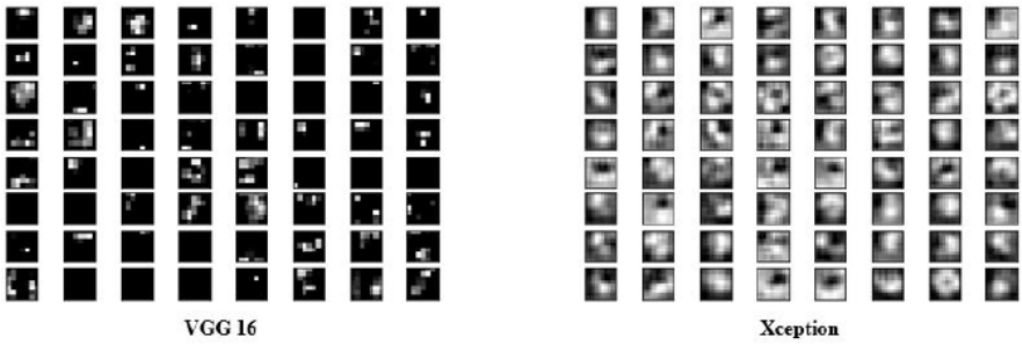


Figure 4: Deep Features extracted from the final convolutional layer of 2 architectures (Bodapati et al. 2020)

Grey-level co-occurrence matrix (GLCM) was the feature extraction method that was implemented to extract second-order texture features from the dataset. GLCM considers the spatial relationships between pixels by calculating how often pixel pairs with certain values and in a specific spatial relationship occur throughout an image. Next, several statistical features are extracted from the matrix such as contrast, correlation, energy and homogeneity. Contrast measures any local variations between pixels and correlation measures the linear grey-level dependence between pixels at specified locations. Energy measures textural uniformity and homogeneity returns the adjacency of the distribution of the elements contained in the matrix to the diagonal of the matrix.

$$Contrast = \sum_{n=1}^{L} n^2 \sum_{|i-j|=n} GLCM(i,j) \quad Energy = \sum_{i=1}^{L}\sum_{j=1}^{L}(GLCM(i,j)^2)$$

$$Homogenity = \sum_{i=1}^{L}\sum_{j=1}^{L}\frac{GLCM(i,j)}{1+(i-j)^2} \quad \text{Correlation} = \frac{\sum_{i=1}^{L}\sum_{j=1}^{L}(i-\mu_i')(j-\mu_j')(GLCM(i,j))}{\sqrt{\sigma_i^2\sigma_j^2}}$$

$$Where: \mu_i' = \sum_{i=1}^{L}\sum_{j=1}^{L}(i)\big(GLCM(i,j)\big),$$

$$\mu_j' = \sum_{i=1}^{L}\sum_{j=1}^{L}(j)\big(GLCM(i,j)\big), \quad \sigma_i^2 = \sum_{i=1}^{L}\sum_{j=1}^{L}((GLCM(i,j))(i-\mu_i')^2),$$

$$\sigma_i^2 = \sum_{i=1}^{L}\sum_{j=1}^{L}((GLCM(i,j))(j-\mu_i')^2)$$

Figure 5: Equations representing the GLCM properties (Widhiarso et al. 2018)

The graycomatrix and graycoprops methods were implemented using the scikit-image library. Energy, correlation, dissimilarity, homogeneity and contrast were extracted for 5 different distances and angles by specifying the distances and angles arguments with the combinations [1,0], [3,0], [5,0], [0,π/4] and [0, π/2]. Despite being computationally expensive, GLCM feature extraction is a cutting-edge tool for feature analysis and differentiating between different MRI images due to its ability to

extract many varying features by using different combinations of parameters and several matrices.

2.1.5 Global Average Pooling

Global average pooling is applied to feature maps to down-sample them. Down-sampling involves the reduction of the resolution of each feature map. The down-sampling of feature maps is a form of dimensionality reduction and therefore aids in the prevention of overfitting the training data. Global pooling layers such as global average pooling differ from non-global pooling layers as they completely down-sample each feature map into one value as opposed to just down-sampling patches from feature maps. Global average pooling computes one value for each feature map into a vector by taking the average of all pixel values from each feature map. Due to global average pooling producing a single feature map, only a single dense layer is often required to classify the feature map produced for each image and this in turn reduces the complexity of a neural network classifier and reduces the chances of overfitting.

2.1.6 Principal Component Analysis (PCA) and Dimensionality-Reduction

Principal component analysis (PCA) is an unsupervised technique used in dimensionality reduction to reduce dimensions of data while retaining the maximum variance through finding directions of max variance to form principal components that are then projected onto a subspace with fewer dimensions. The reduction of data dimensions allows for smaller computation/training times. Furthermore, the reduction of data dimensions also reduces noise in data which in turn leads to a reduction in the generalization error of a neural network and the chances of overfitting.

2.1.7 Image Processing

The image processing stage consists of three techniques gaussian filter, thresholding and erosion/dilation are all from three different areas of image processing which are noise reduction, image segmentation and morphological operations. Noise in images occurs due to variations in colour and brightness information. Noise in images can lead to classification algorithms deducing patterns in the noisy pixels and eventually start making predictions based on the existing patterns in noisy data. Another consequence of the presence of noise is that it can increase computation time while deteriorating the model's performance/accuracy. Due to the negative implications of image noise, it was crucial that a noise reduction technique was implemented in the image processing stage. I chose to apply the Gaussian spatial filter mathematical model to reduce noise in the dataset.

$$G(i,j,k) = \frac{1}{(\sqrt{2\pi})^3 \sigma i \sigma j \sigma k} e^{-(\frac{i^2}{2\sigma_i^2}+\frac{j^2}{2\sigma_j^2}+\frac{k^2}{2\sigma_k^2})}$$

Figure 6: Equation representing the Gaussian Filter mathematical model (Gelvez et al. 2018)

The built-in Gaussian filter method of the ndimage submodule was applied to each image to reduce noise by adding a slight blur to each image. I opted for the Gaussian filter as opposed to other filters like the median filter because it can preserve image structure post-processing as opposed to the median filter which leads to a decrease in image structure and detail. Despite the median filter's lower computation time and distinct advantage in eliminating salt and pepper noise, it was irrelevant to the specific dataset used in the prototype as the images in the dataset are not subjected to any salt and pepper noise.

Image segmentation involves the dispersing of an image into separate fragments. There are several techniques under the domain of image segmentation such as region-based segmentation, cluster-based segmentation and edge-based segmentation etc. I chose to apply threshold-based image segmentation which comprises the partitioning of pixels according to their intensity when compared to a set threshold value. The OpenCV threshold function was used to apply thresholding to each image. The value for the threshold parameter was set to 70 after reviewing the grayscale histograms of images from the dataset to establish which of the grayscale ranges correspond to the different sections of an image. The specific threshold method used was THRESH_BINARY_INV which converts each image into a binary image by replacing pixel values that are less than the threshold value of 70 to 125 i.e., grey. The largest gain from applying thresholding is that allows for the elimination of unnecessary areas of the cerebral MRI image and regions of interest to remain.

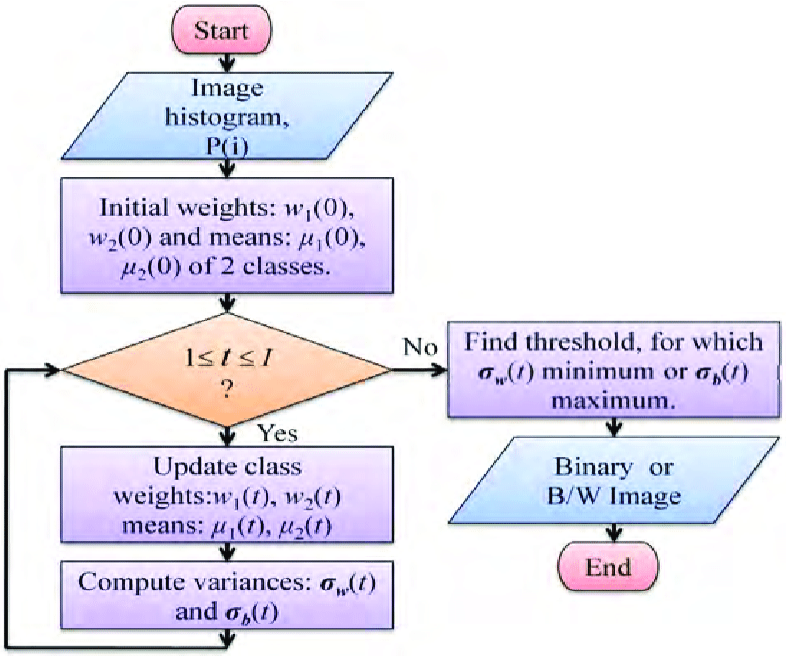


Figure 7: Algorithm for Otsu's Thresholding (Rahman et al. 2015)

Morphological operations were employed to reduce any imperfections that remain following image segmentation. The overall concept of morphological operations is similar to that of many spatial filters as all the pixels from an output image of morphological operations are derived from a comparison with their matching pixels from the input image that have been compared to their respective neighbouring pixels. More specifically in the prototype, the two morphological operations which were applied were erosion followed by dilation. Erosion removes pixels from the boundaries of objects post-image segmentation by iterating over the image and setting each pixel value to the minimum value of its respective neighbours. As a consequence, erosion allowed me to eliminate any unusual areas that were protruding from the tumour area of the MRI image. Dilation sets each pixel of an image to the maximum value of its neighbouring pixels. Using the computer vision library OpenCV, dilation allowed me to remove small gaps which existed in the region of interest (ROI) and also increased the size of the segmented object. Both erosion and dilation further enhanced images after segmentation by removing any imperfections after thresholding.

2.2 Related Work

One trend in the research into deep-learning approaches for brain tumour detection is the use of convolutional neural networks with small/limited datasets. For instance, (Havaei et al. 2017), (Bangalore Yogananda et al. 2020) and (Pereira et al. 2016) all implemented convolutional neural networks with the BraTS dataset, which contains only 3064 T1 images. Due to the low number of MRI images in this dataset, the potential and optimum CNN performance for brain tumour classification cannot be achieved and may lead to overfitting. Using the same BraTS dataset, (Shah et al. 2017) achieved better results demonstrated by higher dice scores through implementing the random forest classifier. Therefore, a deep-learning approach may not be necessary and traditional machine-learning classifiers could be utilised if a small dataset is used to avoid long training times of deep-learning for no additional benefit. However, (Gu et al. 2021) used the larger REMBRANDT dataset (*REMBRANDT* 2022) with CNN along with convolutional dictionary learning and achieved better results that were demonstrated by higher dice/F1 scores that average above 90.0 compared to results achieved by (Havaei et al. 2017), (Bangalore Yogananda et al. 2020) and (Pereira et al. 2016) with the smaller BraTS dataset. (Havaei et al. 2017), (Bangalore Yogananda et al. 2020) and (Pereira et al. 2016) could apply transfer learning to improve the performance of their respective models and negate some of the negative effects of the small BraTS dataset.

A lack of consensus exists on the overall impact of image-processing algorithms on the performance of machine-learning models in the classification of brain tumours. (Khan et al. 2022) argue that “Using mostly traditional medical image-processing

methods, segmenting and classifying brain malignant tumours is a challenging and time-consuming task". While some image-processing algorithms are indeed time-consuming, others can aid classifiers in their objectives while remaining time efficient and therefore all image-processing methods should not be generalised as being time-consuming. For example, (Matuska et al. 2012) present computing times of image-processing algorithms such as the canny edge detector algorithm and propose the argument that some algorithms are perceived as being very time-consuming even though in reality, the same algorithms are much faster when they are implemented with different libraries. For instance, (Matuska et al. 2012) found that the smoothing algorithm used for noise reduction was around 40 times faster when the OpenCV library is applied as opposed to the smoothing algorithm implementation in Matlab.

Research seems to agree that in terms of statistical feature extraction of images, second-order features alone deliver better results than first-order features as well as a combination of first and second-order features. (Priya et al. 2016) present a comprehensive comparison of the effects of first and second-order features on brain tumour classification accuracy of different forms of the SVM classifier. Different methods are used to extract second-order features. For instance, (Abbas et al.) employed a grey level run length matrix (GLRLM) and grey level cooccurrence matrix (GLCM) for second-order texture features such as coarseness, energy and contrast etc. More specifically, (Widhiarso et al. 2018) argue that contrast is the most dominant feature that has the biggest impact on overall accuracy as it may allow easier distinction of the different elements in an MRI image. Despite second-order features alone outperforming a combination of first and second-order features, the combination of both types of statistical features still outperforms wavelet transformation when all performance measures are used for every classifier type according to (Aggarwal and K. Agrawal 2012). On the other hand, (Latif et al. 2019) found that combining first-order, second-order and, wavelet transform feature sets returned better results for each classifier except Naïve Bayes, as opposed to only using second-order texture features. In particular, (Latif et al. 2019) highlight an increase of 6.78% for the multi-layer perceptron classifier when the 38 combined features are used for each MRI modality like flair, T1 and T2. (Latif et al. 2019) did not apply their combined features methodology for a deep-learning approach.

The related work above appertains to the prototype implementation and focuses on GLCM as the feature extraction method however, in the final implementation I have complemented this related work with deep feature extraction by implementing transfer learning using the pre-trained VGG-16 model. (Kang et al. 2021) used a wide range of pre-trained CNNs including VGG-16 for deep feature extraction and then evaluated those deep features using 9 different classifiers including neural networks. While (Kang et al. 2021) did train a neural network classifier on concatenated deep features from pre-trained models such as VGG-16 and Inception V3, they did not concatenate GLCM features to the top deep features. Despite (Kang et al. 2021) not

utilising GLCM features, the results they obtained reinforce my hypothesis/methodology that the fully connected layers of a neural network outperform other traditional classifiers when evaluating deep features from a pre-trained model. In addition, (Srinivasalu and Palaniappan 2019) in their findings also proved that deep neural networks were the top-performing classifier when GLCM features were to be evaluated as opposed to deep features from pre-trained models. Altogether, the findings of (Kang et al. 2021) and (Srinivasalu and Palaniappan 2019) supported my decision to combine deep and GLCM features as well as implement a neural network instead of another classifier for the final classification of the concatenated features.

There seems to be general agreement that in the domain of brain tumour classification, VGG-16 outperforms other pre-trained models that are ResNet 50 and Inception V3. For instance, (Krishnapriya and Karuna 2023) and (Irmak 2021) found that on average VGG-16 outperformed ResNet 50 and Inception V3 in brain tumour classification as supported by the scores they achieved in every performance metric like accuracy, recall and f1-score that they measured, VGG-16 outperformed both ResNet50 and Inception V3 very significantly. Conversely, (Kang et al. 2021) found that in most of their experiments with different datasets in each iteration, deep features obtained from VGG-16 did not outperform features from ResNet 50 and Inception V3 when trained on a neural network. I believe that this could be due to (Kang et al. 2021) only selecting the top 3 features when training the neural network therefore, the decreased observation could have led to the neural network encountering overfitting. Another reason for the discrepancy in results could be due to (Kang et al. 2021) training the fully connected on very small datasets in each of their experiments as the largest dataset they used contained only 3000 MRI images whereas, (Irmak 2021) used a dataset consisting of 4570 images in their final classification task/experiment.

2.3 Gaps in Current Literature

A recurring gap in all the literature discussed above is the lack of a combination of deep and hand-crafted features alongside image-processing steps with a deep-learning model that utilises transfer learning. (Shoaib et al. 2022) also emphasise this specific gap in their future work section.

3 Requirements, Analysis and Methodology

The main overall objective of this project is to detect the presence of a tumour in a brain MRI image and then proceed to classify the tumour into one of the three most common forms such as glioma, pituitary and meningioma or classify the MRI image

as healthy. The project will also involve the application of image-processing algorithms as well as the extraction of deep and hand-crafted features from the dataset containing MRI images. The deep-learning model which is trained using the proposed methodology will be deployed in the form of a web application. This section outlines a list of the requirements/features and the overall steps involved in the proposed methodology as well as the model deployment. An overview of a typical machine-learning/deep-learning development cycle is also included in this section. Each component of the proposed deep-learning methodology cycle will be described in depth later in the section.

3.1 List of Requirements

I. **Gathering Data:** Collection of brain MRI images consisting of 4 groups that are healthy/no tumour, glioma, pituitary and meningioma.
II. **Image Processing:** The MRI images will be augmented, resized, and 3 image processing algorithms will be applied.
III. **Hand-crafted/Deep feature extraction:** Methods to extract first-order, second-order, and deep features are required.
IV. **The Model:** Both a pre-trained model and a neural network model trained on the collected data are required. A pre-trained model is required for deep feature extraction in the form of a base model.
V. **Validation Methods:** The accuracy of the model will be evaluated using a specific methodology. The validation methods allow comparisons to be drawn on the effects of image processing algorithms on the overall classification accuracy as well as comparisons between the prototype implementation and the final deep-learning-based methodology.
VI. **Web Application and User Interface:** A simple web application which will include a function to allow the user to upload an MRI image from a directory and the web application will return the form of the brain tumour detected or if the MRI image shows a healthy cerebral scan. The web application will also display the percentage confidence of the predictions made for each MRI image.

3.2 Proposed Methodology

This section presents an explanation of the complete machine-learning life cycle. The proposed deep-learning methodology is outlined alongside a figure depicting every step of the proposed methodology. The implementation of the requirements of the proposed methodology is thoroughly explained later in the design and implementation section.

3.2.1 Machine Learning Life Cycle

The machine-learning life cycle consists of 5 procedures: data collection, data pre-processing, model training, model evaluation and model deployment.

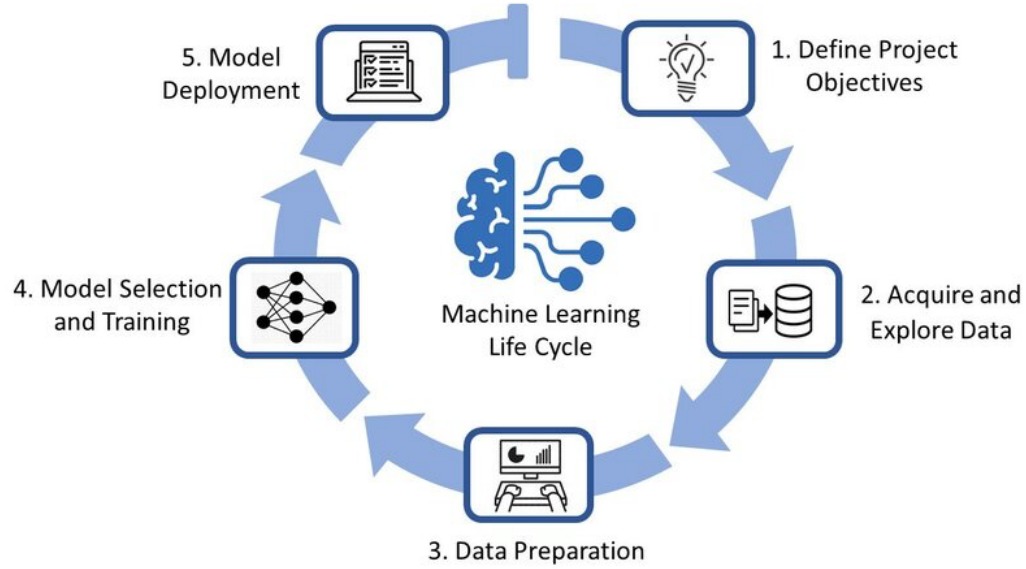


Figure 8: The ML life cycle (Hazratifard et al. 2022)

Data collection involves procuring the specific data type required for training a machine-learning model. During the data collection process, data from different sources and databases may be combined or gathered from a single source. Machine-learning engineers can also generate their own data and this approach is often referred to as private data. Tasks such as augmentation, normalization and the removal of duplicates are all included in data pre-processing. The most crucial step in data pre-processing is the transformation of raw data into a specific format that is compatible with a machine-learning model. Data cleaning allows for noise reduction and filling in any missing values in the dataset. More specifically, noisy data is either resolved through algorithms like a median filter algorithm or a more manual approach that can also be used to remove noise. Model training consists of a machine-learning algorithm learning the patterns in a dataset. This step of the machine-learning cycle also involves the tuning of hyperparameters of an algorithm for a dataset to extract maximum performance. As the name suggests, model evaluation encompasses several performance metrics such as confusion matrices, receiver operating characteristic curve and precision-recall etc. This step allows for confirmation of a model's feasibility as well as comparisons of different models. A recent form of model evaluation is model interpretability which measures how easily the internal mechanism of a machine-learning model can be understood (Gärtler et al. 2021). Finally, model deployment is the release of a machine-learning model to end users. In addition to the challenges of regular software development such as scalability, machine-learning applications introduce further challenges like constant

monitoring and model drift due to sudden changes in the data environment (Spjuth et al. 2021).

3.2.2 Proposed Deep Learning Methodology and Objectives

The proposed brain tumour classification system is constituted of 6 main stages. The 6 main stages are data pre-processing, image processing, hand-crafted feature extraction, transfer-learning-based CNN deep-feature extraction, dimensionality reduction using principal component analysis (PCA), and a neural network multiclass classifier. During the data pre-processing stage, brain MRI images from the dataset will be resized to the dimensions required for the pre-trained model like VGG-16. Image augmentation will also be applied during pre-processing to reduce any imbalance in the dataset. Three algorithms/techniques will be applied during the image processing step and each of the three techniques are from one of the 3 categories: noise removal, segmentation or morphological operations. The image processing techniques/algorithms applied were thresholding, gaussian filter and erosion/dilation. Hand-crafted feature extraction will include first-order features and second-order texture features like energy and correlation. The second-order features will be extracted by using the grey-level co-occurrence matrix (GLCM) method. In the transfer-learning-based stage, the output from the fourth max pooling layer of the VGG-16 CNN architecture will be used as the set of deep features. The combined features will then be used in the classification process by the target CNN model. The CNN model will then classify the brain MRI images into one of 4 classes: no tumour/healthy, glioma, pituitary and meningioma.

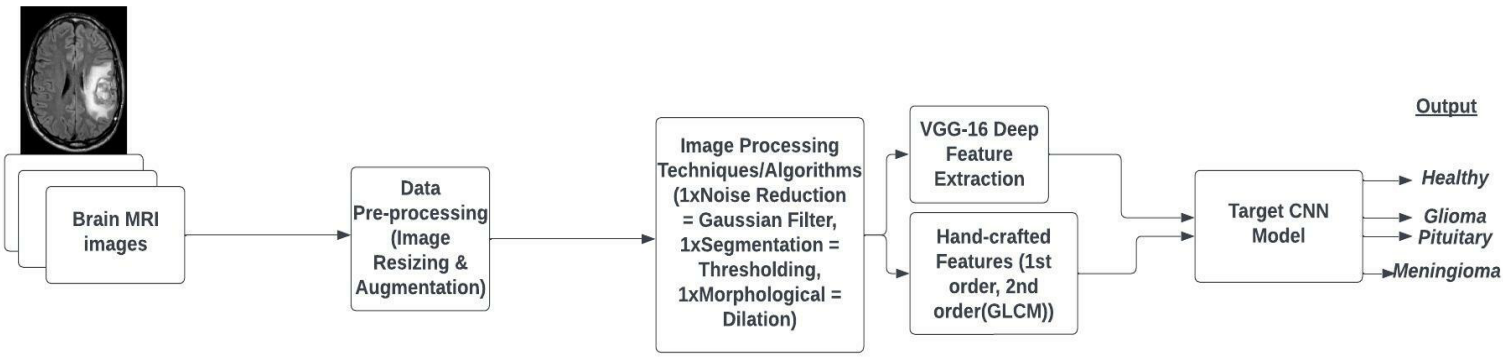


Figure 9: Depicts all Phases of the Proposed Methodology

3.2.3 Validation Requirements

The validation of the trained model will require a set of 13,500 images. It is imperative that the proposed model is validated to ensure its accuracy. The required methods will include k-fold cross-validation and confusion matrix. K-fold cross-validation will require the dataset to be separated into a k number of folds and this is beneficial when there is a limited amount of data available. K-fold cross-validation will deliver the average performance of the proposed model by testing the model on different combinations/splits of the whole dataset rather than a fixed set of validation images. On the other hand, confusion matrices are required as they allow for several performance metrics such as F-1 score, Matthew's correlation coefficient, accuracy, ROC/AUC score and recall metrics to be calculated. Furthermore, the validation of the final web application will involve uploading images outside of the dataset i.e. from an online source and the images uploaded for prediction will already be pre-labelled by medical professionals. Therefore, the pre-labelled images can be considered ground truth and allow me to eliminate the possibility of validation using mislabelled MRI images from an online source. Overall, the required validation methods will allow for the estimation of the effectiveness of processes such as the different image processing methods and the tuning of certain hyperparameters related to neural networks such as optimizers and dropout rates.

3.2.4 Usability

The usability of the deep-learning model has improved significantly with the fulfilment of the user interface requirement with the deployment of the final deep-learning model in the form of a web application. The web application has increased ease of use by avoiding unnecessary elements/buttons and only displaying the necessary buttons that correspond to functions that choose an image, submit an image and start prediction. The usability of the frameworks/libraries which are required in order to implement the training phase of the proposed methodology could present unique challenges. For instance, training the CNN by using the TensorFlow framework with Keras API will be very time-consuming unless on-demand GPU instances are used. TensorFlow is often described as being confusing and inconsistent during use as the methods included with TensorFlow that allow us to create convolution kernels etc have similar names but different implementations or parameters. For example, tf.keras.layers.conv2d and tf.keras.nn.conv2d have very similar implementations but require slightly different parameters. A hardware usability issue is that when GPU use is desired to reduce time consumption when using TensorFlow, only Nvidia GPUs that are CUDA enabled will be compatible. Finally, some frameworks and especially TensorFlow have limited features when used alongside the Microsoft Windows operating system.

3.3 Final Implementation Gantt Chart

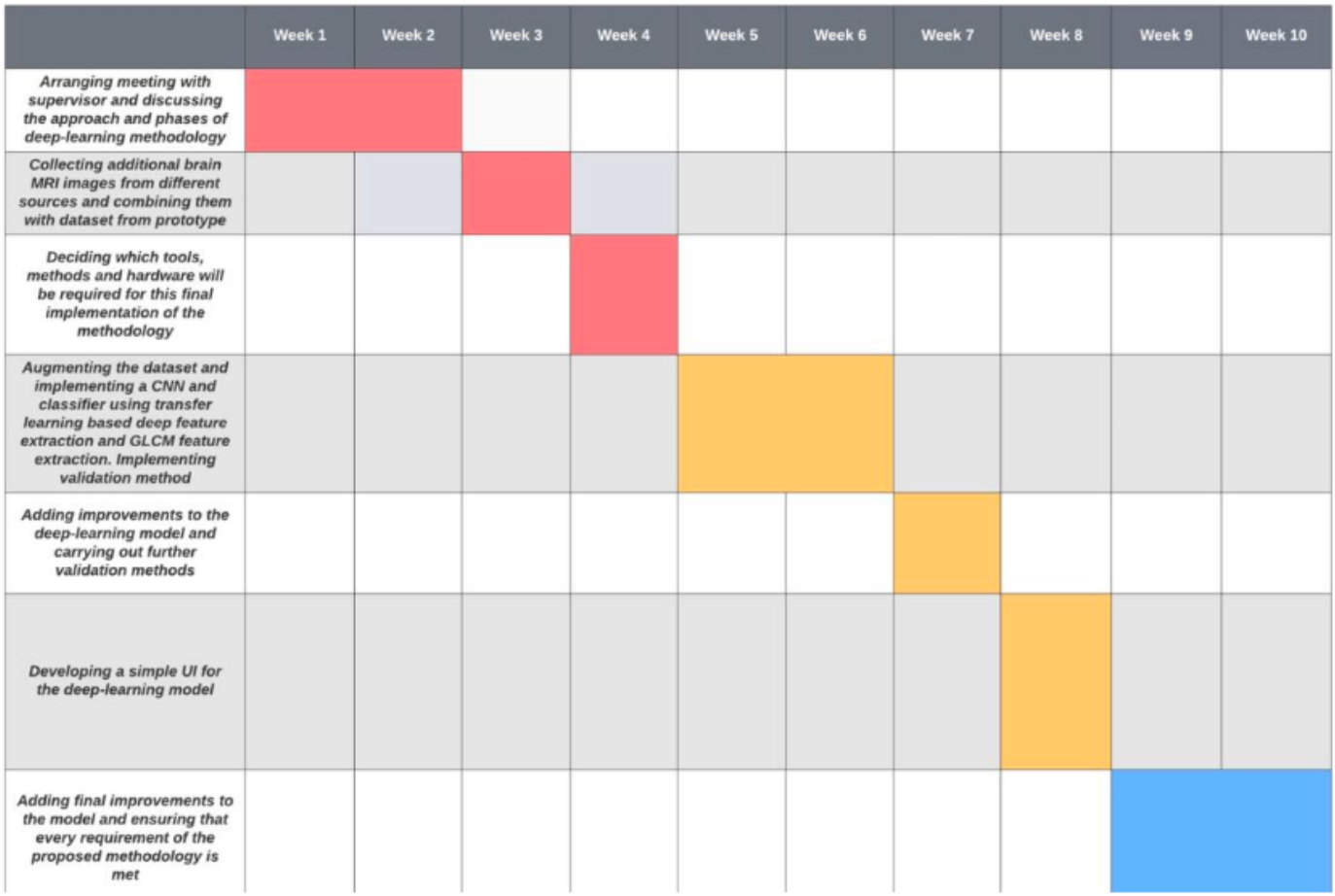


Figure 10: Gantt Chart for the Final Model

4 Design and Implementation

This section includes a comprehensive explanation of the components and the architecture involved in the deep-learning-based implementation. In addition, this section outlines the requirements of the proposed methodology that have been satisfied through the implementation of a deep-learning model and its deployment in the form of a web application. Justification for the implemented solution is presented as well as the overall workflow, frameworks, tools and algorithms utilised in the implemented solution.

4.1 Model, View and Controller

The model from the MVC architecture in the context of my implementation of the proposed deep-learning methodology is the backend of the web application which includes the trained model, the image processing steps and feature extraction steps which are applied to each image uploaded by the user for a prediction. The model

section of the MVC then makes a prediction on the images it received from the user and returns it to the view. The view is represented by the frontend section of the web application which the user will have access to and will interact with. The view takes data provided to it by the model and renders it to a webpage using HTML, CSS and JavaScript. Finally, the controller component of the MVC architecture is represented by the API and endpoints that allow the controller to act as a channel of communication between the model (backend) and the view (frontend) so that it can receive requests from the user through the view and then use the model of the MVC to retrieve the data which in this case are predictions. The controller then returns the response/prediction to the user through the view which is the frontend of the web application.

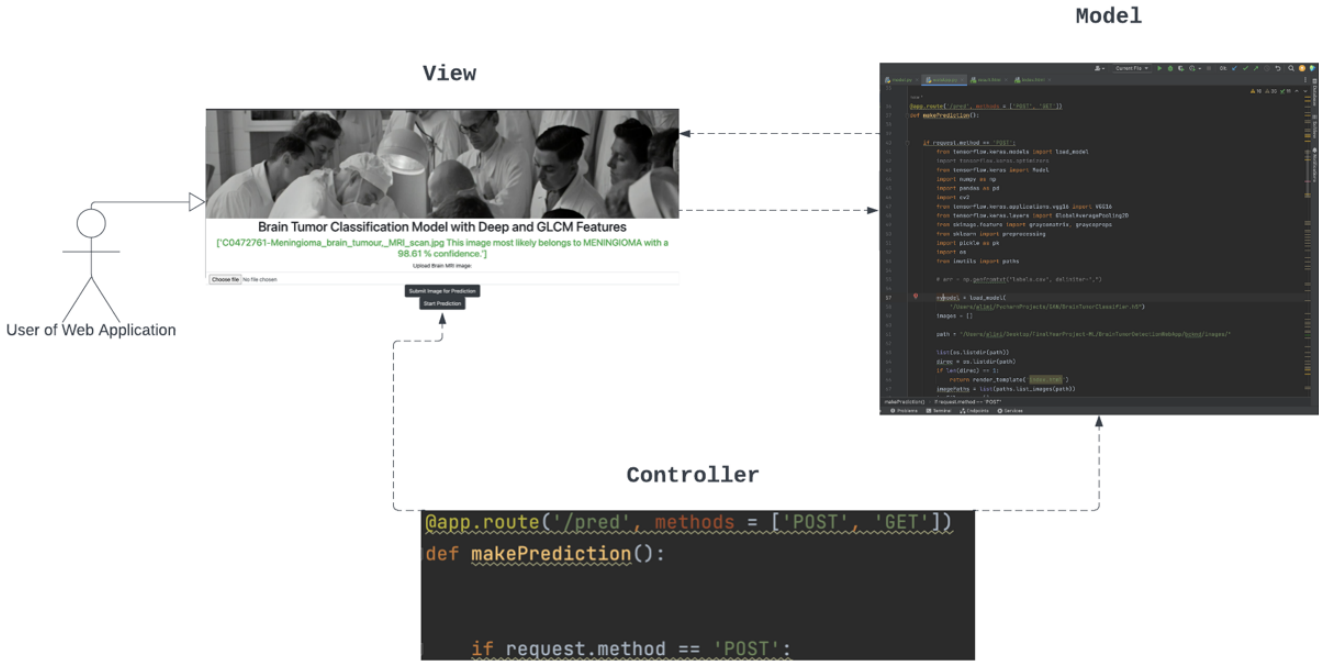


Figure 11: Overview of the MVC architecture of the web application

4.2 Overview and Workflow

The web application meets all of the mandatory requirements, including data collection, image augmentation, three image processing techniques, hand-crafted features (GLCM), deep features from VGG-16, validation methods and a user interface. The final deep-learning model and therefore the web application can classify MRI images into four classes which are no tumour/healthy, pituitary, meningioma and glioma. Overall, the final deep-learning model has two iterations: without the three image processing techniques and with the three image processing techniques. Each step and the architecture of the deep-learning methodology implementation are depicted in the workflow diagram below.

4.2.1 Workflow Diagram

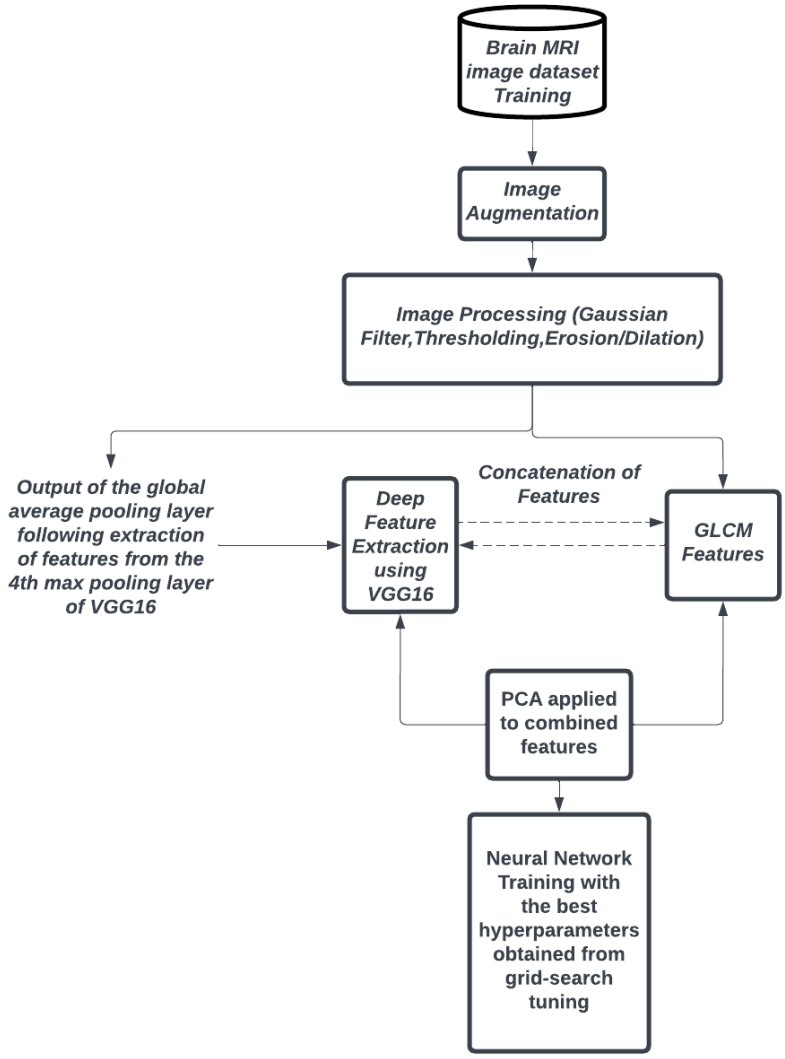


Figure 12: Workflow for the proposed methodology

In the subsequent sections, each component of the methodology is discussed comprehensively.

4.3 Tools and Packages

An array of tools and packages were deployed for the implementation of the proposed methodology. The Python programming language was used alongside several of its libraries and packages. All aspects related to the model were implemented on the computing platform/IDE known as PyCharm however, the kernel used for model development was a remote kernel provided through a private cluster by Paperspace. The private cluster which was used for development consisted of an Nvidia RTX A4000 GPU, 90GB of ram and a 16-core CPU. Every brain MRI

image in the dataset was stored in JPG format. Python's flexibility, simplicity and various machine-learning libraries made it the logical choice for the implementation of the proposed methodology.

4.4 Dataset

The brain MRI images dataset was used to train and test the proposed model. The images were collected from multiple sources on the Kaggle website and the REMBRANDT dataset (Scarpace et al. 2019), (Nickparvar 2021) and (Bhuvaji et al. 2020). Due to the images included in the dataset being collected from multiple sources, it was imperative that any duplicates were detected and removed by assigning images their unique hash codes and if images were found to have identical hash codes, the duplicate(s) were removed. All images were saved in the jpg format. Before image augmentation, the dataset consisted of 11,250 images from healthy, 13,212 from pituitary, 11,600 from glioma, 11,825 from meningioma. To increase the total size of the dataset to 54,000 and eliminate any data imbalance image augmentation was applied. Every image in the dataset was resized to 225x225 and converted to greyscale if they were not already in greyscale. The images were augmented using Keras's image data generator to carry out transformations that include vertical flips and a zoom range of 0.5. The folder structure of the dataset consists of a folder that includes four subfolders for each class. The dataset was split into 75% for training and 25% for testing. The decrease in data imbalance was vital as most classification algorithms will assume that the data, they are trained on is balanced. Therefore, the weights assigned to each of the samples and classes are equal but if the data is imbalanced, the classifier will be more biased towards the majority class which in the prototype would have been the tumour class before augmentation. Furthermore, the issues surrounding data imbalance could be propagated in the implementation as it involves multiclass classification and could lead to a tumour type belonging to a minority class not being detected or even classified as one of the other three classes.

4.5 The Classification Model

This section discusses the image processing process, combined feature extraction methods and the neural network classifier.

4.5.1 Image Processing Methods

The image processing stage consists of three techniques gaussian filter, thresholding and erosion/dilation are all from three different areas of image processing which are noise reduction, image segmentation and morphological operations. Noise in images occurs due to variations in colour and brightness information. Noise in images can

lead to classification algorithms deducing patterns in the noisy pixels and eventually start making predictions based on the existing patterns in noisy data. Another consequence of the presence of noise is that it can increase computation time while deteriorating the model's performance/accuracy. Due to the negative implications of image noise, it was crucial that a noise reduction technique was implemented in the image processing stage. I made the choice to apply the Gaussian spatial filter mathematical model to reduce noise in the dataset.

$$G(i,j,k) = \frac{1}{(\sqrt{2\pi})^3 \sigma i \sigma j \sigma k} e^{-(\frac{i^2}{2\sigma_i^2}+\frac{j^2}{2\sigma_j^2}+\frac{k^2}{2\sigma_k^2})}$$

Figure 13: Equation representing the Gaussian Filter mathematical model (Gelvez et al. 2018)

The built-in Gaussian filter method of the ndimage submodule was applied to each image to reduce noise by adding a slight blur to each image. I opted for the Gaussian filter as opposed to other filters like the median filter because it can preserve image structure post-processing as opposed to the median filter which leads to a decrease in image structure and detail. Despite the median filter's lower computation time and distinct advantage in eliminating salt and pepper noise, it was irrelevant to the specific dataset used in the prototype as the images in the dataset are not subjected to any salt and pepper noise.

Image segmentation involves the dispersing of an image into separate fragments. There are several techniques under the domain of image segmentation such as region-based segmentation, cluster-based segmentation and edge-based segmentation etc. I chose to apply threshold-based image segmentation which comprises the partitioning of pixels according to their intensity when compared to a set threshold value. The OpenCV threshold function was used to apply thresholding to each image. The value for the threshold parameter was set to 70 after reviewing the grayscale histograms of images from the dataset to establish which of the grayscale ranges correspond to the different sections of an image. The specific threshold method used was THRESH_BINARY_INV which converts each image into a binary image by replacing pixel values that are less than the threshold value of 70 to 125 i.e., grey. The largest gain from applying thresholding is that allows for the elimination of unnecessary areas of the cerebral MRI image and regions of interest to remain.

Morphological operations were employed to reduce any imperfections that remain following image segmentation. The overall concept of morphological operations is similar to that of many spatial filters as all the pixels from an output image of morphological operations are derived from a comparison with their matching pixels from the input image that have been compared to their respective neighbouring pixels. More specifically in the prototype, the two morphological operations which were applied were erosion followed by dilation. Erosion removes pixels from the

boundaries of objects post-image segmentation by iterating over the image and setting each pixel value to the minimum value of its respective neighbours. As a consequence, erosion allowed me to eliminate any unusual areas that were protruding from the tumour area of the MRI image. Dilation sets each pixel of an image to the maximum value of its neighbouring pixels. Using the computer vision library OpenCV, dilation allowed me to remove small gaps which existed in the region of interest (ROI) and also increased the size of the segmented object. Both erosion and dilation further enhanced images after segmentation by removing any imperfections after thresholding.

4.5.2 Feature Extraction Methods and PCA

Deep features were extracted using the output of the fourth max pooling layer of the pre-trained VGG-16 model. A global average pooling layer was appended to the end of the VGG-16 model to flatten the output of the fourth max pooling layer and to generate one feature map for each filter/type of feature. The decision to append a global average pooling layer to the VGG-16 model was due to its ability to categorise the hundreds of features by finding associations within the thousands of feature maps that are generated from the max pooling layer. Furthermore, the global average pooling layer allowed for the recognition of the classes even prior to any training of the neural network classifier by identifying specific components within the feature maps. Due to this attribute of global average pooling layers, they are often used interchangeably with fully connected layers however, this was not the case for the implementation of my proposed methodology as both a global average pooling layer and a neural network classifier were utilised. Finally, the deep features generated using the global average pooling layer succeeding the max pooling layer allowed me to decrease the complexity of the neural network classifier and as a consequence, avoid overfitting due the global average pooling layers ability to categorise feature maps prior to any training of the fully connected layers of a neural network.

Layer (type)	Output Shape	Param #
input_5 (InputLayer)	[(None, 225, 225, 1)]	0
block1_conv1 (Conv2D)	(None, 225, 225, 64)	640
block1_conv2 (Conv2D)	(None, 225, 225, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
global_average_pooling2d_4 (	(None, 512)	0

Total params: 7,634,112
Trainable params: 0
Non-trainable params: 7,634,112

Figure 14: Summary of the deep feature extractor with global average pooling

The graycomatrix and graycoprops methods were implemented using the scikit-image library. Energy, correlation, dissimilarity, homogeneity and contrast were extracted for 5 different distances and angles by specifying the distances and angles arguments with the combinations [1,0], [3,0], [5,0], [0,π/4] and [0, π/2]. Despite being computationally expensive, GLCM feature extraction is a cutting-edge tool for feature analysis and differentiating between different MRI images due to its ability to extract many varying features by using different combinations of parameters and several matrices.

The dimensions of the concatenated features were reduced through implementing PCA. Prior to dimensionality reduction, the combined features were scaled to a range individually using a min-max scaler. All components of the scaled features were plotted against the percentage variance to determine and select the optimum number of components without a significant loss of variance. The application of PCA to reduce the dimensions of the combined feature set reduced training time and increased the ability of the model to generalize to unseen instances due to a reduction in noise/outliers.

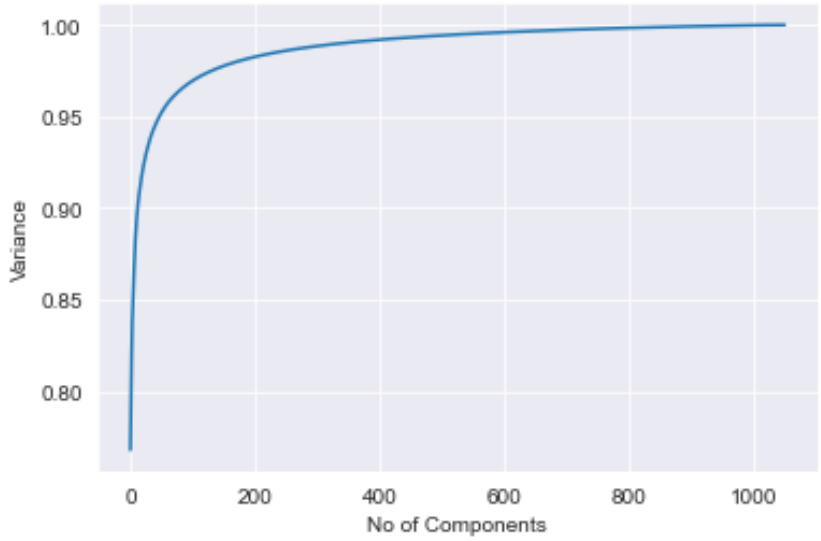


Figure 15: The relationship between the no of components selected and the percentage maximum variance in the feature set

700 components were selected with a maximum variance loss of only 0.27% and resulted in faster training of the neural network classifier.

4.5.3 Neural Network Classifier

The neural network classifier encompasses only two layers that are fully connected and a dropout layer between the two dense layers. The first of the two dense layers has only 60 neurons and implements the ReLU activation function to remove all negative values from the concatenated feature set post-dimensionality reduction with PCA. The reasoning behind the decision to only add 60 neurons to the first dense layer was due to a potential risk of the neural network overfitting the training feature set. Furthermore, 60 neurons were selected due to 60 being the optimum level of high training and high validation accuracies, suggesting no occurrence of overfitting. The optimum number of neurons was determined through hyperparameter tuning/optimization using a grid-search of parameters including optimizers, neurons and batch sizes. Neurons lower than 60 were also experimented with however, this diminished the learning capabilities of the neural network and decreased performance. The ReLU activation function was implemented due to its fairly simple computation of a gradient when compared to more complex activation functions such as the hyperbolic tangent function (tanh). The lower complexity of ReLU allowed for faster learning speeds and therefore, a reduction of computation time and improved performance over other activation functions. The experiments of (Alhassan and Zainon 2021) supported the findings from my experiments involving hyperparameter tuning.

The second and final dense layer of the neural network classifier included 4 output neurons i.e., one neuron for each class that was predicted. The activation function known as Softmax was utilised as opposed to sigmoid due to the form of image

classification problem being multiclass in the final implementation rather than binary classification like the prototype implementation. The softmax function outputs the probability of each predicted image being in each of the 4 output classes. The class with the maximum probability from the output of the softmax function is the prediction which is presented to the user of the web application. Finally, the dropout layer was inserted into the neural network architecture to reduce overfitting by reducing bias through setting some inputs to 0 at random with a frequency rate specified as 0.4 i.e., 40% of the inputs to the layer are set to 0 at random. The neural network was trained on a batch size of 40 and only 50 epochs. Only 50 epochs were used and when epochs were increased, training accuracy only increased by negligible amounts while decreasing the validation accuracy by inducing overfitting and increasing training time exponentially.

4.6 Web Application and Deployment

The Python micro web framework known as Flask was used to deploy the brain tumour classification model. Flask allows for quick model deployment with the use of a simple API and endpoints to allow the model/backend to be exposed to the user via the front end of the web application. Flask was selected due to its built-in web server and its ability to not rely on many dependencies. The model was saved using the save function built into Keras as an h5 file. The Flask API has three endpoints that are the home page endpoint, the upload image endpoint and the prediction endpoint. The front end of the web application was developed using HTML to implement forms that receive images which are uploaded by the user, JavaScript/REACT for components like button functionality and CSS was used for styling the buttons etc along with the bootstrap CSS framework. The flask backend python script also applies all the steps required for the implementation of the deep-learning methodology such as image processing and deep feature extraction etc prior to passing the processed features to the saved model for a prediction. The prediction from the model is then posted to the HTML frontend webpage for the user to view.

Figure 16:Web application with prediction

4.7 Validation

The validation of the effects of certain parameter choices, image-processing methods and feature extraction methods is carried out in the methodology using the performance measures such as F-1 score, Matthew's correlation coefficient, ROC/AUC score, recall, accuracy, confusion matrix and 25% of the dataset called a test set. A confusion matrix is made up of a table of 4 sections and each section includes how many instances and their predicted labels from the test data belong to either true positive, false positive, false negative or true negative. Several measures of performance are derived from confusion matrices like recall, accuracy, precision and F-1 score. Confusion matrices provided me with useful insights regarding the impact of certain image processing techniques, feature extraction methods, etc., during the whole deep-learning web application development cycle and how well the model was classifying cerebral MRI images into the 4 classes (Healthy, Pituitary, Glioma and Meningioma). The validation accuracy for each epoch during training of the neural network classifier was also measured to track if the validation accuracy continues to improve as epochs are increased and that the neural network is learning sufficiently i.e. not underfitting or overfitting.

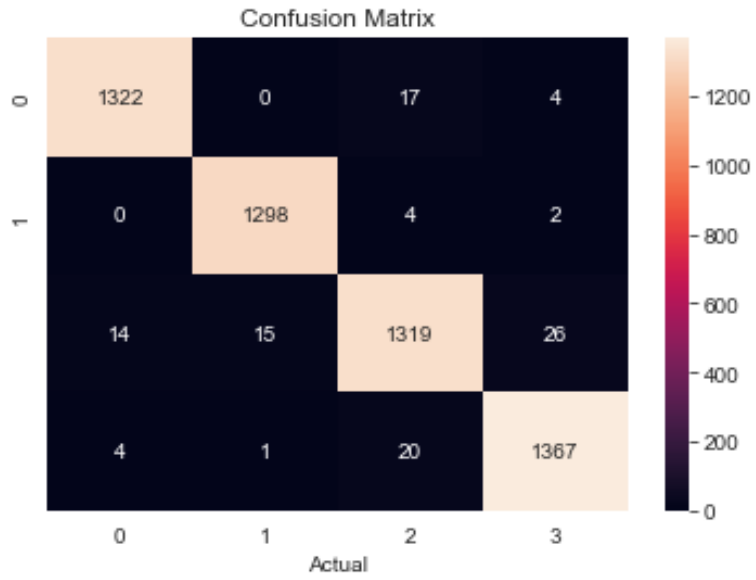


Figure 17: Confusion Matrix for the Brain Tumour Classifier with GLCM+Deep Features

5 Results and Discussion

This section presents the results obtained from the validation of the prototype as well as a comparison of the results between the prototype implementation and the final deep-learning web application model. Furthermore, the two implementation iterations are compared with and without image processing. Limitations and possible improvements to the deep-learning model/methodology and web application are also

discussed along with legal, social, ethical and professional issues surrounding the proposed methodology and web application.

	DL Model without image processing
DL Model with image processing	
Accuracy 89.34%	98.20%
F-1 score 0.8901	0.9804
Matthews Correlation Coefficient 0.8846	0.9736
Recall 0.88436581 0.87471781 0.82471929 0.84269118	0.98436337 0.99539877 0.95997089 0.98204023
ROC/AUC score 0.8532	0.9986

Figure 18: Results for each performance measure of final DL model

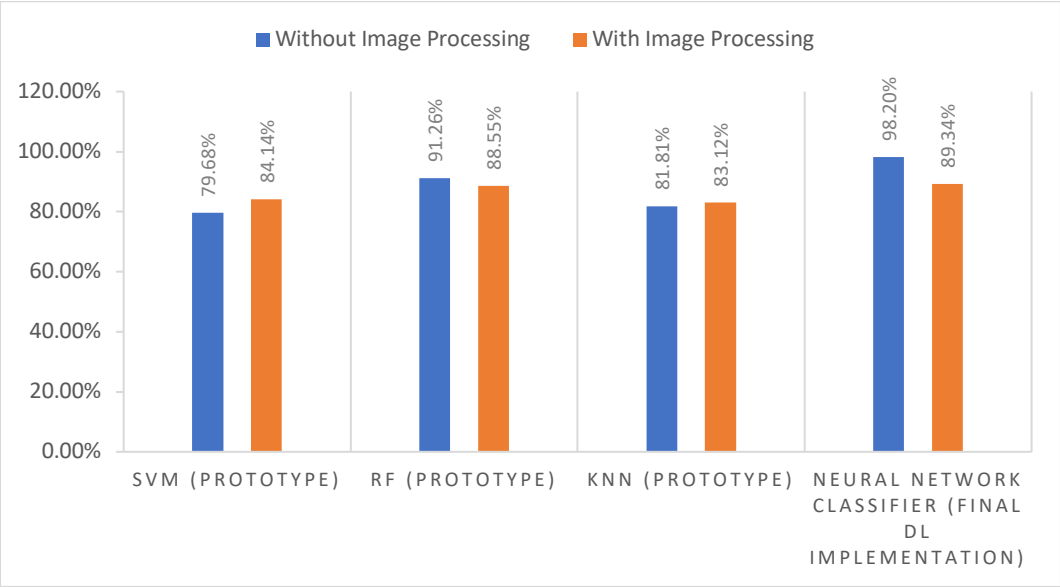
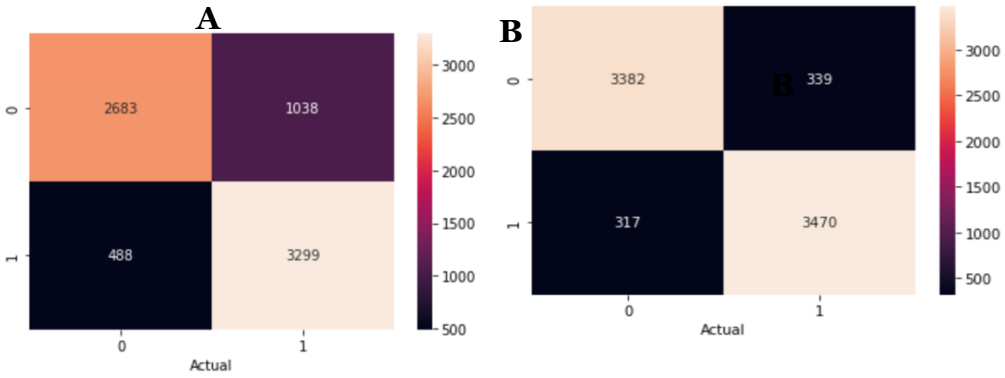


Figure 19: Accuracy scores of all 3 classifiers from the prototype and final deep-learning classifier



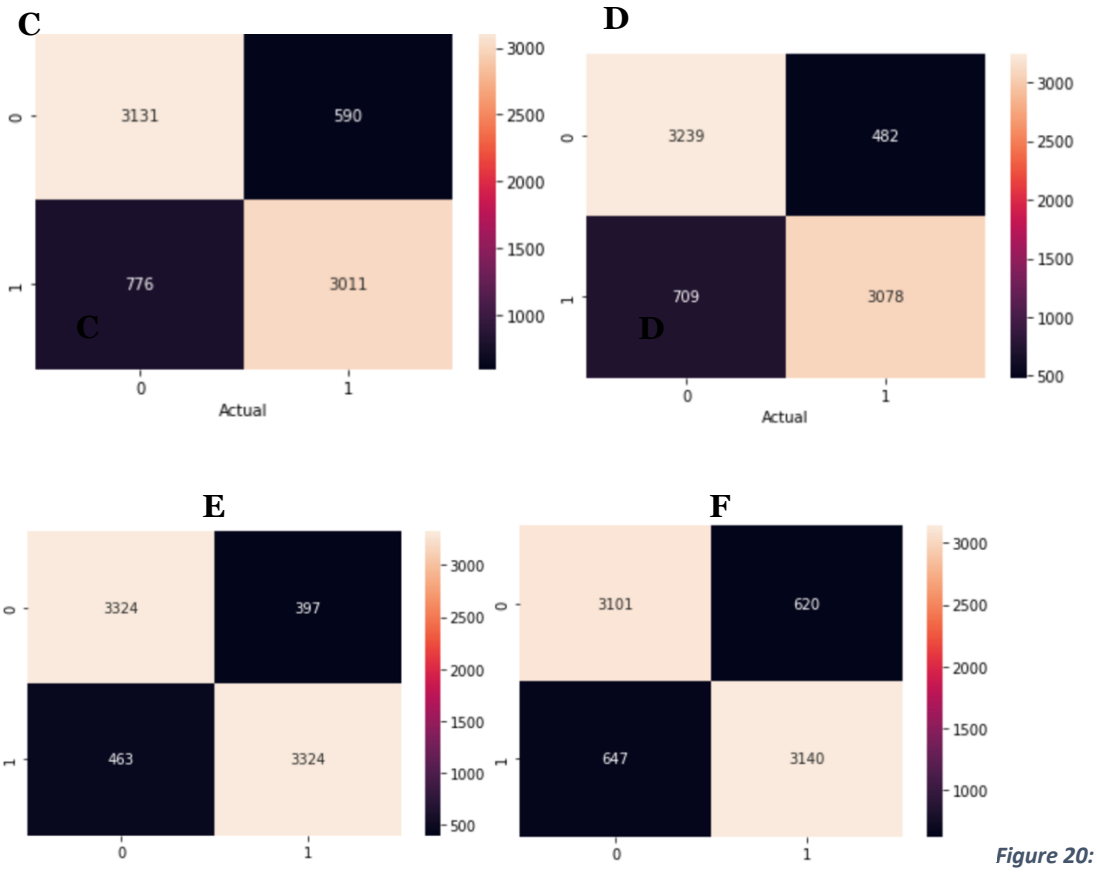


Figure 20: Confusion Matrices of SVM without image processing (A), RF without image processing (B), KNN without image processing (C), SVM with image processing (D), RF with image processing (E), KNN with image processing (F)

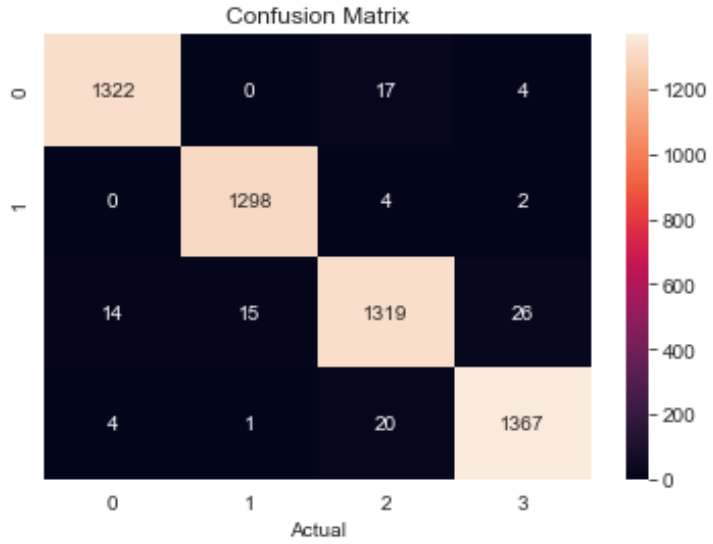


Figure 21: Confusion matrix for the final DL model implementation without image processing

The above results for the prototype implementation were all obtained using the same dataset and keeping all other variables like the feature extraction method the same/constant. The three image processing steps (gaussian filter, thresholding and erosion/dilation) improved SVM and KNN accuracy scores by 4.28% and 1.40%

respectively as expected. However, the decrease in performance for RF was surprising as it was the leading classifier in terms of performance out of the three prior to the three image processing steps being applied. The decrease of 2.71% could be due to the Gaussian filter as even though it reduces noise efficiently, it could be reducing fine details in the images due to the blurring effect. On the other hand, another possible reason for the decrease in classification accuracy of RF could be due to the fact that some noise may still remain in the data after the Gaussian filter was applied as the kernel size for the filter may have been too small. Therefore, the remaining noise after filtering could have led to overfitting as the decision trees in the RF could have captured noise in the training data. This could be countered by implementing more decision trees/estimators however this would have a negative effect on computational efficiency. This balance between computational efficiency and avoiding overfitting by increasing trees presents a dilemma as a small increase in accuracy could be justified for slower training or, a slight reduction in accuracy but less training time can also be justified. The increase in classification accuracy for SVM and KNN was expected as both SVM and KNN are sensitive to noise in data and therefore the Gaussian filter image processing technique was beneficial for their respective classification accuracies. SVM and KNN are sensitive to noisy data for similar reasons. SVM is affected by noise due to target classes overlapping and the fact that noise can cause the SVM's separating margin to be a stochastic variable (Li et al. 2013). KNN is affected by noise as some of the data may overlap into a region of the other class. Therefore, mislabelled instances would dramatically change the decision boundary and could lead to new data points being misclassified. The increase in KNN and SVM performance could also be due to the image thresholding step decreasing data complexity slightly as the binary image obtained after thresholding leads to a reduction in data complexity. Therefore, the decrease in data complexity in turn decreases data dimensionality which improves KNN's performance as the distance of data points is decreased in all dimensions. Despite the increase in classification accuracy for SVM and KNN after image processing, they both required more time for training after image processing was utilised and could still not outperform the random forest classifier after its accuracy decreased due to the three image processing steps.

The prototype met some but not all of the requirements. The requirements it met were data gathering/image augmentation, 3 image processing techniques, extraction of hand-crafted features, traditional machine learning algorithms and a validation method. The code from the prototype has been uploaded to a GitHub repository and can be accessed through this link.

All three classifiers used in the prototype have many limitations in the classification of brain MRI images. For instance, the three classifiers used in the prototype all deliver moderate results when compared to the results obtained by (Akter et al. 2022) with their InceptionResNetV2 deep-learning model despite them using a smaller dataset of only 3264 images compared to the 30,000 images used in the

prototype. This discrepancy in performance despite a larger dataset indicates a specific limitation that the three traditional classifiers used in the prototype cannot learn high-level features from the larger dataset provided, unlike many deep-learning algorithms and deep features. An obvious limitation of the prototype is that it can only classify the brain MRI images into two classes at this stage rather than detecting the form of tumour present. Another limitation of the prototype is that it does tend to overfit the training data to some extent as I had to depend on data augmentation quite heavily due to a lack of varying cerebral MRI images available publicly due to some legal, social, ethical and professional issues surrounding the distribution of medical images.

The final deep-learning model/web application, which meets all the requirements of the proposed methodology outperforms all three classifiers from the prototype in every performance metric which was measured as expected due to several reasons. For instance, the second iteration/final implementation DL model was trained on a larger dataset than the prototype consisting of 54,000 cerebral MRI images. The larger dataset allowed for the neural network to not overfit the training combined feature set. Furthermore, the larger dataset also reduces variance in the bias-variance trade-off as the larger dataset did not affect/increase bias while reducing variance and chances of overfitting due to more samples being available to the neural network. Another potential reason for the discrepancy in performance between the final implementation and the prototype is that with the use of transfer learning in the final implementation i.e. deep feature extraction allowed for more abstract and complex features/objects like the specific tumour type to be detected and segmented in feature maps. Conversely, in the prototype implementation, only GLCM features were used and the lack of complex features from the pre-trained VGG16 model with existing weights meant that the segmented feature maps could have allowed for the neural network to learn more class-specific details in each image as segmentation simplifies complex MRI images into simple yet significant features that are easier for the classifier to analyse. The pre-existing weights of VGG-16 may have also positively impacted performance due to VGG-16 being trained on the large ImageNet dataset for image classification including 4,000,000 images whereas, the prototype classifiers did not have any pre-trained weights from a large dataset of images for deep feature extraction.

Interestingly, the discrepancy between the final deep-learning model with and without image processing was significant. The image processing steps deteriorated performance in every metric while increasing computation time significantly. The deterioration in performance could be due to the thresholding value causing the area of the tumour in each image to be inaccurately segmented. Additionally, the incorrect threshold value in inverse binary thresholding could be causing normal brain tissue or skull sections to be segmented as opposed to the specific tumour area and therefore, valuable information in each MRI image could have been lost. In addition, the Gaussian filter may have also reduced complex details as well as noise. The

Gaussian filter could have also negatively affected the classification performance of the neural network as the Gaussian filter has a blurring effect which may have caused edges of tumours in the MRI image for instance to become unclear and weak.

After rigorous testing of the web application with unseen images, a decrease in the accuracy of predictions was observed with certain images. I recognised the fact that cerebral MRI images of certain modalities were predicted with less accuracy than others. This difference in prediction accuracy between modalities could be due to modalities like T2 having fewer instances/representations in the dataset due to a lack of T2 MRI images available online for use over other modalities. Therefore, the model has seen fewer instances of this particular modality during training. The code from the web application and final deep-learning model has been uploaded to a GitHub repository and can be accessed through this link.

5.1 LSEPI

Several legal, social, ethical and professional issues (LSEPI) surround some aspects of the project, and the issues often overlap. The dataset alone presents several LSEP issues. For instance, a legal challenge surrounding the collection of patient cerebral MRI image data is that the data will be subject to the UK's data protection act of 2018. Therefore, it is paramount that all seven of the data protection principles are strictly followed. For example, the first data protection principle refers to "sensitive processing" and explicitly states that "processing of data concerning health" falls under sensitive processing and therefore one condition of Schedule 10 must be met (*Data protection act 2018* 2018). More specifically, the condition met for the dataset used in the prototype is that the "processing is necessary for medical purposes" and medical purposes include "medical research" in the act (*Data protection act 2018* 2018) and this project falls under the umbrella term of "medical research". A social issue relating to the implementation and the project as a whole is related to interpretability as many members of society struggle to understand deep-learning methods as they are often labelled as "black boxes". Therefore, due to the black box label and difficulty in understanding the inner workings of deep-learning models, members of society distrust deep-learning approaches in every domain and this would prevent vital implementations such as the web application I deployed from being ignored and receive scepticism. Another issue is a societal issue as well as an ethical issue which is the preservation of privacy. For instance, the prototype faces the issue of privacy as the MRI images could be used to reconstruct a patient's face which would lead to the patient's data privacy and anonymity being violated. Moreover, to avoid this issue I was asked to complete a licence agreement form for some of the cerebral MRI images I gathered from The Cancer Imaging Archive (TCIA) (*REMBRANDT* 2022). Due to the nature of this prototype and this project, there are no specific professional issues about this project, however, the code of conduct and four principles from the British computer society (BCS) were followed.

6 Conclusion

To conclude, the final deep-learning methodology implementation and web application detect the presence and form of a brain tumour in images accurately when a user uploads an image for prediction via the web application in a matter of seconds. The web application also displays the percentage confidence of the prediction using the output from the softmax layer of the final layer of the neural network classifier. However, the web application/model does not return the correct number of predictions between different modalities proportionately. This difference in performance between modalities shows that the lack of MRI images is still a very significant factor in the generalization abilities of brain tumour classification models.

6.1 Future Work

Future work will involve further enhancements to the web application. The web application could be developed to scan MRI images for brain tumours through a webcam/digital camera rather than only predicting from images which are downloaded and then uploaded by the user. Future work could also involve training the same model implemented using the proposed methodology on an even larger dataset which includes images from a range of different modalities to further increase the generalization ability of the web application. Finally, the deep-learning model could also be deployed in the form of a smartphone application in the near future to increase portability and access to the model to more potential users.

7. References

Abbas, H. K., Fatah, N. A., Mohamad, H. J. and Alzuky, A. A. Brain Tumor Classification Using Texture Feature Extraction. Bristol, 2021. Vol. 1892. IOP Publishing. https://go.exlibris.link/8CFVfcjQ.

Aggarwal, N. and K. Agrawal, R. (2012) First and Second Order Statistics Features for Classification of Magnetic Resonance Brain Images. *Journal of signal and information processing* 3 (2), 146-153.

Akter, N., Junjun, J. A., Nahar, N., Hossain, M. S., Andersson, K. and Hoassain, M. S. (2022) Brain Tumor Classification using Transfer Learning from MRI Images. *Proceedings of International Conference on Fourth Industrial Revolution and Beyond 2021.* Singapore, 2022//. Springer Nature Singapore.

Alhassan, A. M. and Zainon, W. M. N. W. (2021) Brain tumor classification in magnetic resonance image using hard swish-based RELU activation function-convolutional neural network. *Neural computing & applications* 33 (15), 9075-9087.

Bangalore Yogananda, C. G., Shah, B. R., Vejdani-Jahromi, M., Nalawade, S. S., Murugesan, G. K., Yu, F. F., Pinho, M. C., Wagner, B. C., Emblem, K. E., Bjørnerud, A., Fei, B., Madhuranthakam, A. J. and Maldjian, J. A. (2020) A Fully Automated Deep Learning Network for Brain Tumor Segmentation. *Tomography (Ann Arbor)* 6 (2), 186-193.

Bhandari, A., Koppen, J. and Agzarian, M. (2020) Convolutional neural networks for brain tumour segmentation. *Insights into imaging* 11 (1), 77-77.

Bhuvaji, S., Kadam, A., Bhumkar, P., Dedge, S. and Kanchan, S. (2020) Brain Tumor Classification (MRI).

Bodapati, J., Veeranjaneyulu, N., Shareef, S., Hakak, S., Bilal, M., Reddy, P. and Jo, O. (2020) *Blended Multi-Modal Deep ConvNet Features for Diabetic Retinopathy Severity Prediction.*

Data protection act 2018. (2018) Legislation.gov.uk: Queen's Printer of Acts of Parliament. https://www.legislation.gov.uk/ukpga/2018/12/part/4/chapter/2/crossheading/the-data-protection-principles/enacted Accessed 13/01/23.

Gärtler, M., Khaydarov, V., Klöpper, B. and Urbas, L. (2021) The Machine Learning Life Cycle in Chemical Operations – Status and Open Challenges. *Chemie Ingenieur Technik* 93 (12), 2063-2080.

Gelvez, E., Vera, M., Huérfano, Y., Valbuena, O., Salazar, W., Vera, M. I., Borrero, M., Barrera, D., Hernández, C., Molina, Á. V., Martínez, L. J., Sáenz, F., Vivas, M., Vanegas, J., Contreras, J., Restrepo, J., Salazar, J. and Contreras, Y. (2018) Smoothing filters in synthetic cerebral magnetic resonance images: A comparative study. *Revista latinoamericana de hipertensión* 13 (4), 335-338.

Géron, A. l. (2019) *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems.* Second edition. Beijing [China];Sebastopol, CA;: O'Reilly Media, Inc.

Gu, X., Shen, Z., Xue, J., Fan, Y. and Ni, T. (2021) Brain Tumor MR Image Classification Using Convolutional Dictionary Learning With Local Constraint. *Frontiers in neuroscience* 15, 679847-679847.

Havaei, M., Davy, A., Warde-Farley, D., Biard, A., Courville, A., Bengio, Y., Pal, C., Jodoin, P.-M. and Larochelle, H. (2017) Brain tumor segmentation with Deep Neural Networks. *Medical image analysis* 35, 18-31.

Hazratifard, M., Gebali, F. and Mamun, M. (2022) Using Machine Learning for Dynamic Authentication in Telehealth: A Tutorial. *Sensors (Basel, Switzerland)* 22 (19), 7655.

Irmak, E. (2021) Multi-Classification of Brain Tumor MRI Images Using Deep Convolutional Neural Network with Fully Optimized Framework. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* 45 (3), 1015-1036.

Kang, J., Ullah, Z. and Gwak, J. (2021) Mri-based brain tumor classification using ensemble of deep features and machine learning classifiers. *Sensors (Basel, Switzerland)* 21 (6), 1-21.

Khan, M. M., Omee, A. S., Tazin, T., Almalki, F. A., Aljohani, M. and Algethami, H. (2022) A Novel Approach to Predict Brain Cancerous Tumor Using Transfer Learning. *Computational and mathematical methods in medicine* 2022, 1-9.

Krishnapriya, S. and Karuna, Y. (2023) Pre-trained deep learning models for brain MRI image classification. *Frontiers in Human Neuroscience* 17.

Latif, G., Iskandar, D. N. F. A., Alghazo, J. M. and Mohammad, N. (2019) Enhanced MR Image Classification Using Hybrid Statistical and Wavelets Features. *IEEE access* 7, 9634-9644.

Li, H.-X., Yang, J.-L., Zhang, G. and Fan, B. (2013) Probabilistic support vector machines for classification of noise affected data. *Information Sciences* 221, 60-71.

Matuska, S., Hudec, R. and Benco, M. (2012) The comparison of CPU time consumption for image processing algorithm in Matlab and OpenCV. *2012 ELEKTRO.* 21-22 May 2012.

Nickparvar, M. (2021) Brain Tumor MRI Dataset.

Pereira, S., Pinto, A., Alves, V. and Silva, C. A. (2016) Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images. *IEEE Transactions on Medical Imaging* 35 (5), 1240-1251.

Primary and secondary brain tumours. (2019) Cancer Research UK. *https://www.cancerresearchuk.org/about-cancer/brain-tumours/types/primary-secondary-tumours#:~:text=Tumours%20that%20start%20in%20the,brain%20cancer%20or%20brain%20metastases.* Accessed 13/01/23.

Priya, K. M., Kavitha, S. and Bharathi, B. (2016) Brain tumor types and grades classification based on statistical feature set using support vector machine. *2016 10th International Conference on Intelligent Systems and Control (ISCO).* 7-8 Jan. 2016.

Rahman, M. M., Khatun, F., Islam, S., Al-Amin, M. and Bhuiyan, M. (2015) Binary Features of Speech Signal for Recognition. *International Journal of Applied Research on Information Technology and Computing (IJARITAC)* 6.

Rashid, T. (2006) *A Novel Recurrent Neural Network Model: A Case Study in Energy Load Forecasting*.

REMBRANDT. (2022) The Cancer Imaging Archive.

Scarpace, L., Flanders, A. E., Jain, R., Mikkelsen, T. and Andrews, D. W. (2019) *Data From REMBRANDT*. The Cancer Imaging Archive.

Shah, N., Ziauddin, S. and Shahid, A. R. (2017) Brain tumor segmentation and classification using cascaded random decision forests. *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON).* 27-30 June 2017.

Shoaib, M. R., Elshamy, M. R., Taha, T. E., El-Fishawy, A. S. and Abd El-Samie, F. E. (2022) Efficient deep learning models for brain tumor detection with segmentation and data augmentation techniques. *Concurrency and computation* 34 (21), n/a-n/a.

Simonyan, K. and Zisserman, A. (2014) Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Spjuth, O., Frid, J. and Hellander, A. (2021) The machine learning life cycle and the cloud: implications for drug discovery. *Expert opinion on drug discovery* 16 (9), 1071-1079.

Srinivasalu, P. and Palaniappan, A. (2019) Combining Wavelet Texture Features and Deep Neural Network for Tumor Detection and Segmentation Over MRI. *Journal of intelligent systems* 28 (4), 571-588.

Widhiarso, W., Yohannes, Y. and Prakarsah, C. (2018) Brain Tumor Classification Using Gray Level Co-occurrence Matrix and Convolutional Neural Network. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems) (Online)* 8 (2), 179-190.

Zhao, F., Zeng, Y. and Bai, J. (2022) Toward a Brain-Inspired Developmental Neural Network Based on Dendritic Spine Dynamics. *Neural computation* 34 (1), 172-189.