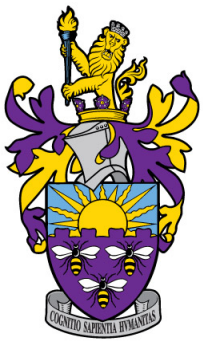

UNIVERSITY OF MANCHESTER

DEPARTMENT OF COMPUTER SCIENCE

Classification of medical images of brains with tumours with respect to the grade of the tumour

Author:
Ammaar SAIYED
Student ID:
10137761

Supervisor:
Dr. Fumie COSTEN

A dissertation submitted to The University of Manchester for the degree of BSc Computer Science and Mathematics with Industrial Placement

in the

Faculty of Science and Engineering

2022

Contents

List of Figures

UNIVERSITY OF MANCHESTER

Abstract

Faculty of Science and Engineering
Department of Computer Science

BSc Computer Science and Mathematics with Industrial Placement

Classification of medical images of brains with tumours with respect to the grade of the tumour

by Ammaar SAIYED

Brain tumours are one of the deadliest types of cancers across the world, despite not being as common as others types of cancer. In the UK alone, over 11,000 people are diagnosed with brain tumours on average per year. The survival rate for these patients is very bad - with more than 85% of them failing to live past 5 years after their initial diagnosis. The range of treatments for these patients vary depending on the severity of the tumour, which is determined based on the World Health Organisation (WHO) grading system. This system classifies brain tumours into one of four grades, with 1 being the least severe, and 4 being the most. Recent advances in medical image analysis and machine learning mean that a lot of research has been dedicated to investigating different machine learning models to be able to classify brain tumours into their grades. One such technique used is called a convolutional neural network. Before the data is passed into these networks, some processing steps are taken initially to improve the performance and efficiency. This project aimed to introduce some novel techniques to perform some of these steps, including region enhancement and dimensionality reduction. These networks require a lot of computational resource to learn features, and the amount of carbon emissions that training networks emit is extremely large. Hence, this project aimed to minimise the computational cost. The model adequately learned generalisable features from the data, and achieved relatively good test accuracy. A graphical user interface was also developed to illustrate the preprocessing pipeline which is useful to help others visualise the pipeline.

Declaration of Authorship and Intellectual Property Statement

Declaration

No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Intellectual Property Statement

i The author of this dissertation (including any appendices and/or schedules to this dissertation) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made **only** in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the thesis, for example graphs and tables ("Reproductions"), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy, in any relevant Dissertation restriction declarations deposited in the University Library, and The University Library's regulations.

Acknowledgements

My sincere thanks to my supervisors Fumie Costen and Uli Sattler for guiding and supporting me throughout the project.

Chapter 1

Introduction

1.1 Motivation

A tumour, also called neoplasm, refers to an abnormal growth of normal tissue, caused by genetic and epigenetic events – with the actual tumour forming when cells divide more than normal or when they don't die when they should (Jaroudi, 2017).

Tumours can be benign – which means that the growth stays in its primary location without invading other sites of the body, or they can be malignant. Benign tumours are not usually problematic but can cause pain and other medical issues, as well as possibly becoming malignant. Malignant tumours are cancerous, and cancerous tumours grow quickly, invade surrounding tissue, and can spread to other parts of the body. Malignant tumours require treatment as soon as possible, which can improve the patients' prognosis (Patel, 2020).

In the UK, around 5,900 people are diagnosed with benign brain tumours, and around 5,500 people are diagnosed with malignant brain tumours each year (Cancer Research UK, 2022c). Brain tumours are one of the deadliest types of cancers, with more than 85% of people failing to survive past 5 years after initially being diagnosed. For those cancers that are more advanced, less than 5 people out of 100 survive for more than 5 years (Office for National Statistics, 2019).

There are many different subcategories of brain tumours, with the most common type being gliomas (Kabir Anaraki, Ayati, and Kazemi, 2019). This is a general term for those tumours that arise from the brain formed from cells that are not nerves or blood vessels – i.e. 'glia' – the supporting cells of the brain (American Association of Neurological Surgeons, 2022).

Severity is usually classified according to the WHO definition, graded by cell activity and aggressiveness on a scale of I to IV (Louis et al., 2016). Depending on the severity of the tumour, physicians and doctors recommend different avenues for treatment and for steps to take going forward. Therefore, an accurate diagnosis, as well as an accurate prognosis is of utmost importance, and these rely on being able to effectively establish the type of tumour that a patient has (Mackillop, 2006).

One paper in particular mentions that 'The James Lind Alliance Priority Setting Partnership in Neuro-Oncology' has identified 'more timely diagnoses among their Top 10 priorities for future research,' and that expedited diagnosis for cancers results in a benefit in morbidity, relief, and mental wellbeing – potentially improving quality of life of such patients (Penfold et al., 2017).

It is then evident that a swift and accurate diagnosis is vital, whether the patient has a more positive or negative prognosis, due to the increased effectiveness of treatments for those with more positive prognoses, and the benefits to the quality of life for all.

Early diagnosis is important for improving possibilities, particularly for patients with gliomas, since they are 'the most infiltrative and life-threatening' tumours (Sajid, Hussain, and Sarwar, 2019). Medical imaging techniques used to make these diagnoses include "[...] Computed Tomography (CT), Single-Photon Emission Computed Tomography (SPECT), Positron Emission Tomography (PET), Magnetic Resonance Spectroscopy (MRS) and Magnetic Resonance Imaging (MRI) [...]," all of which help to paint a better picture of the situation by providing information such as size, location, etc., and assist in making an accurate diagnosis (Işın, Direkoğlu, and Şah, 2016).

In the years between the 1970s to the 1990s, medical image analysis was introduced, and carried out using low-level pixel processing with mathematical modelling – commonly referred to as 'good old-fashioned artificial intelligence.' During the 1990s, convolutional neural networks (CNNs) had begun to be used in medical image analysis but became the method of choice more recently during the 2010s. These days, CNNs are usually the most successful type of machine learning model for image analysis (Litjens et al., 2017).

1.2 Objective and aims

This project aims to produce an effective CNN that can predict the grade of a tumour to a high level in line with other research and models that have been tested in recent years. There have been many 1000s of papers that have created machine learning models, particularly CNNs to do image analysis with brain tumours, made a lot easier by the availability of brain data, and many of these papers have achieved extremely high accuracies.

However, there is still much room for improvement since many of these methods and much of the research focuses mainly on obtaining the highest possible accuracy. Most techniques already achieve accuracy percentages in the high 90s, and so a lot of research is done in trying to make marginal gains in performance. Research from MIT discusses this, mentioning,

"You have to throw a lot more computation at something to get a little improvement in performance. It's unsustainable. We have to find more efficient ways to scale deep learning or develop other technologies. (MIT News, 2020)"

The computational complexity and cost of creating and training machine learning models, particularly CNNs, produces a pertinent challenge for researchers, and there has been an increasing trend in creating AI that is computationally efficient, (Paul and Gvsl, 2018; Ephrath et al., 2020; Sun and Wang, 2020).

Not only are more computationally efficient models more sustainable for the environment, but a shift towards higher efficiency opens up possibilities for things such as carrying out computing on edge devices, making AI more accessible, faster inference, and more (Paul and Gvsl, 2018).

As a result, not only will the project aim to produce a CNN model that is effective at classifying brain tumours, but it will aim to be computationally efficient, reducing computational complexity using a range of techniques from standard to novel methods.

Gliomas were found earlier to be the most life-threatening type of brain tumour, so the project will primarily use scans of gliomas. The project will include completing substantial research into the problem area, investigating prior research on CNNs that have been used for similar work, sourcing an appropriate dataset, preprocessing the dataset for use in the CNN, creating and training a CNN architecture,

tuning the hyperparameters, and finally evaluate the CNN. All of these steps will be taken with sustainability being one of the key focuses in the decision-making.

Chapter 2

Background and theory

2.1 Brain tumours

2.1.1 Brain tumours

Tumours are abnormal growths of tissue cells, also known as neoplasms. The exact causes of tumours are not yet known, but there are many factors, genetic and epigenetic events, that can make one likely to have a tumour (Jaroudi, 2017).

A brain tumour is then an uncontrolled growth of these cells within the brain, and there are other factors that can increase the risk of brain tumours, such as exposure to radiation, or a family with a history of brain tumours (Magadza and Viriri, 2021).

These tumours can either be benign or malignant, i.e., they can either be non-invasive or invasive. Benign tumours are not usually life-threatening, although they do cause other medical issues, and possibly become malignant themselves. Alternatively, malignant tumours are cancerous, and this means they grow and spread rapidly, meaning they need to be treated as soon as possible (Patel, 2020).

Benign brain tumours have a uniformity in their structure, and they are named so as they do not contain active (cancer) cells. Malignant tumours on the other hand have a nonuniform structure and do contain these active cancerous cells (Bahadure, Ray, and Thethi, 2017).

The survival and treatment options that can be offered to brain tumour patients are highly variable depending largely on the grade of the tumours. Tumours can also be differentiated by their histological types – meaning the type and structure of the cells in the tumour. This grading is usually done by pathologists, via a visual inspection of histopathology slides – meaning an analysis, study, and diagnosis of the tissue cells. Beyond histopathological determination, it is also important to determine the grade of the tumour since the treatment that can be offered and that can be effective is totally different depending on the severity of these tumours (Ertosun and Rubin, 2015).

2.1.2 Tumour grading system

A uniformity and standard is important when attempting to grade tumours, and the currently followed standard is that of the World Health Organization (WHO) (Louis et al., 2016). This is the most widely accepted system for classifying tumours from the central nervous system (CNS).

Grading tumours is important as it gives doctors an idea of how the tumour might behave, as well as helping them in prescribing an appropriate course of treatment. As per the WHO system, tumours are graded from grades 1 to 4. This is primarily done by pathologists, by analysing samples of the brain under a microscope,

and assessing how the cells behave. The more normal the cells look, the lower the grade. Grades 1 and 2 are referred to as low grade – slow-growing tumours, and grades 3 and 4 are high grade – tumours that are fast-growing and aggressive (Cancer Research UK, 2022b).

Furthermore, low-grade brain tumours are:

- slow-growing
- relatively contained
- unlikely to spread to other parts of the brain
- less chance of returning if removed,

and high-grade tumours are:

- fast-growing
- can be referred to as 'malignant' or 'cancerous' growths
- more likely to spread to other parts of the brain
- may come back, even if intensively treated.

Some tumours contain a mixture of cells with different grades. The tumour is graded according to the highest grade of cell it contains, even if the majority of it is low grade (The Brain Tumour Charity, 2022).

2.1.3 Gliomas

Within the brain itself, there are different types of cells. The two basic types are neurons and glia, with glia outnumbering neurons but neurons being the key players in the brain (National Institute of Neurological Disorders and Stroke, 2022). Glial cells are quite different to nerve cells, with the major difference being that they do not participate in synaptic interaction and electrical signalling (Purves et al., 2004).

Glial cells instead support nerve cells with energy and nutrients and help maintain the blood-brain barrier, and there are also different types of glial cells too. A glioma is an umbrella term that is used to describe all tumours that arise from glial cells, such as astrocytoma, oligodendroglioma, and glioblastomas. All of these vary in aggressiveness, and malignancy, with different treatment options and prognoses possible (Mayfield Clinic, 2022).

Gliomas are the most prevalent type of brain tumour, making up about 30% of all primary and metastatic tumours, and accounting for more than 75% of malignant brain tumours (American Association of Neurological Surgeons, 2022; Jaroudi, 2017). There are three different types of glial cells, astrocytes, oligodendrocytes, and ependymal – with their corresponding tumours named astrocytoma/glioblastoma, oligodendrogliomas, and ependymomas respectively (Cancer Research UK, 2022a).

Astrocytomas are the most common type of glioma in both adults and children. The survival rates of more than 5 years in England for this type of tumour is more than 90% for grade 1, around 50% for grade 2, more than 20% for grade 3, and around 5% for grade 4 (National Cancer Intelligence Network, 2018).

Symptoms of gliomas include:

- headaches
- seizures

- personality changes
- weakness in the arms, face or legs
- numbness
- problems with speech,

caused by pressing on the brain or spinal cord from the tumour itself. These symptoms are often subtle and slow to appear at first, and some gliomas do not cause any symptoms at all. In these cases, the tumours are only spotted due to tests that doctors were doing for other reasons (John Hopkins Medicine, 2022).

2.2 Diagnosis

Doctors use many techniques to assist them in making a diagnosis of a brain tumour. In general, the process begins with magnetic resonance imaging (MRI). This scan should show if there is a tumour present in the brain, and then the type of brain tumour can be determined by the pathologists by sampling tissue from a biopsy or surgery (Cancer.net, 2021).

2.2.1 Techniques

Computed tomography (CT) scans take pictures inside the body, using x-rays to make detailed cross-sectional images of the brain and spinal cord. CT scans can create detailed images of soft tissue in the body, unlike a regular x-ray. They are not as useful for brain scans as MRIs but help in allowing the doctor to see the effect of the tumour on the skull, or to see further detail of the bone structure near the tumour. It can also help to find bleeding and enlargement in the brain (Bhargava, 2019).

Positron emission tomography (PET) scans are often used to find out more detail about tumours whilst patients are receiving treatment. They involve being injected with a slightly radioactive substance which is mainly found in tumour cells, and then a camera creates a picture of the areas of radioactivity in the body. It is useful when treatment is ongoing since it helps to determine if abnormal areas on MRIs are tumours or scar tissue (Shukla and Kumar, 2006).

There are also further techniques such as lumbar puncture, Single-Photon Emission Computed Tomography (SPECT), Magnetic Resonance Spectroscopy (MRS), Electroencephalography (EEG), and many more that can assist in making an accurate diagnosis (Cancer.net, 2021). In this project, we are primarily concerned with MRIs.

MRIs

MRI scans are very good for looking at the brain and spinal cord and are the preferred technique since they are usually more detailed than computed tomography (CT) scans (Cancer.org, 2022). Another reason they are popular is that they use non-ionising radiation, as well as the ability to obtain many different types of images based on parameters or agents employed (Sultan, Salem, and Al-Atabany, 2019).

MRIs provide substantial detail of the brain, spinal cord, and vascular anatomy, and allow for the visualisation of the axial, sagittal, and coronal planes of the brain

(Case Western Reserve University, 2022). These are the three commonly used anatomical planes, with axial being a horizontal plane dividing the body into upper and lower sections, sagittal being a longitudinal plane dividing the body into right and left, and coronal also being a longitudinal plane, this time dividing into front and back (Park et al., 2010).

MRI scans use the body's natural magnetic properties to produce detailed images, namely the magnetisation properties of atomic nuclei. The scanners produce strong magnetic fields that force protons in the body to align with the magnetic field, as they are normally randomly oriented within the water nuclei of the tissue being examined. A radiofrequency (RF) current is then pulsed through the patient, stimulating the protons, and causing them to spin out of equilibrium (Westbrook and Talbot, 2018; Berger, 2002).

Various processes are then applied to allow the nuclei to return to their resting alignment. These processes are known as relaxation, where spin returns to thermal equilibrium after absorbing RF energy. There are two types of relaxation, longitudinal and transverse, known as T1 and T2 respectively (Grover et al., 2015).

These two relaxation times correspond to the most common MRI sequences, T1-weighted and T2-weighted scans. Additionally, a gadolinium contrast agent can be administered to acquire T1 with contrast agent weighted images (T1-gd); this agent is non-toxic and enhanced the contrast in the image. This is particularly useful when looking at vascular structures and breakdown in the blood-brain barrier (Case Western Reserve University, 2022). Similarly, T2-weighted FLAIR (Fluid Attenuation Inversion Recovery) images provide more contrast than regular T2-weighted images due to the reduction of signal coming from the cerebrospinal fluid (Jaroudi, 2017).

2.3 Deep learning

Machine learning is an umbrella term which includes a broad range of algorithms and models that perform intelligent predictions based on a dataset. Often, these datasets consist of millions of unique data points, and this recent trend of huge datasets is known as big data. A machine learning algorithm, also known as a model, is chosen based on the problem at hand – simpler tasks can be done using techniques such as linear regression, and more complex tasks usually require research, investigation, and testing to determine a suitable method (Nichols, Herbert Chan, and Baker, 2019).

One of the main choices to be made based on the structure of the data and the task at hand is between supervised learning and unsupervised learning. Supervised learning involves training a data sample from a data source with the correct classification already known. This means that we know what the label of each data point is and we are trying to teach the model to learn this and to learn the features present in the data that will help to classify this into the correct class (Sathya and Abraham, 2013). The nature of supervised machine learning makes it an ideal choice for tasks such as image classification.

Unsupervised learning refers to, "[...] the ability to learn and organize information without providing an error signal to evaluate the potential solution (Sathya and Abraham, 2013)." The model trains itself in a sense, and this type of learning is usually used in pattern recognition or clustering.

The task of this project is to classify scans of brains into their correct grade based on the MRI scan, which means the task is a classification task, and the amount of data is large. Most models of relevance in medical imaging are classification models

that are trained using supervised learning. This is done by splitting the data into a training set and a testing set. The training data is used to find the parameters that produce model results closest to the true labels, and the test data is then used to assess the performance of the model but does not influence the model parameters (Nichols, Herbert Chan, and Baker, 2019).

Neural networks are a type of machine learning model, designed to recognise patterns and consist of several nodes, ranging from a few to millions that are densely interconnected. These nodes are based on the perceptron, and the wider network is based on the human brain. A perceptron simply takes an input or inputs, has some weights that it multiplies these inputs by, carries out the weighted sum of these inputs and weights, and then applies some function, known as the activation function to this sum (Sharma, 2017).

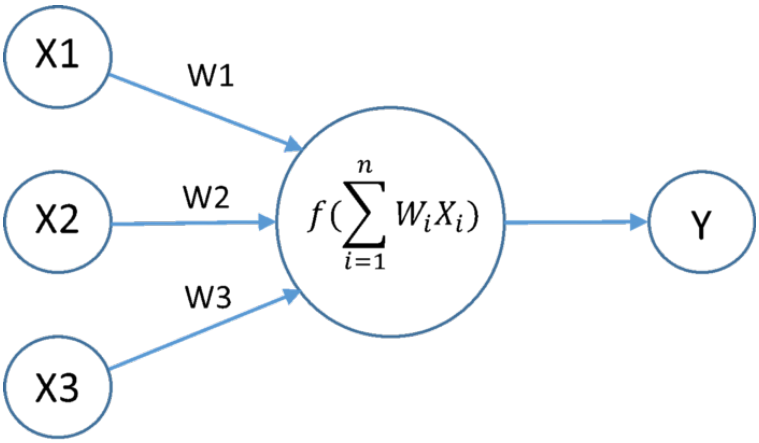

FIGURE 2.1: A simple perceptron.

A neural network consists of multiple layers with these nodes. There is always an input layer, which receives input but doesn't perform computations, and an output layer, which performs the final computation and determines the network's output. A type of machine learning that consists of a neural network with many hidden layers in between the input and output layer is known as deep learning. Deep learning is a subdivision of machine learning, based on learning data representations and hierarchical feature learning. The process by which this is learned is via the arrangement of numerous layers of processing, with each sequential layer providing the input for the next layer. Each layer performs a series of calculations, usually involving multiplication by a set of weights, and the final layer gives the regression or classification being sought. The process of learning in a neural network is when we try to find these optimal weights (Sultan, Salem, and Al-Atabany, 2019).

2.4 Convolutional neural networks

Convolutional neural networks (CNN) are a class of neural network in deep learning, consisting of some layers that perform a convolutional operation. They are commonly used in analysing visual imagery and designed to require lesser preprocessing (Sultan, Salem, and Al-Atabany, 2019). They were created with the assumption that nearby inputs are highly related to one another, and in the case of image classification, we can see why this is effective since nearby pixels are likely to be similar to other nearby pixels. With this assumption, convolutional neural networks

focus on local regions of the images in order to extrapolate local features. This process of extrapolating local features from subregions of the data point is performed in the convolutional layers (Paul et al., 2017). Convolutional neural networks use three basic ideas: local receptive fields, shared weights, and pooling, and all of these will be discussed in more detail (Neural networks and deep learning, 2022).

Convolutional neural networks are also a type of neural network that are called feedforward neural networks. This means that the output of one layer is used as input to the next layer as has been mentioned above. However, there are also some types of neural networks where the feedback loops back around, and such networks are named recurrent neural networks (Lazar, 2009).

In recent times, CNNs have become one of the most popular methods of machine learning to do with brain tumours and also in medical imagery, due to their powerful unbounded performance compared to other models. Their ability to learn spatial hierarchies of features, small patterns and edges, larger patterns etc., make them a good fit for image analysis tasks (Magadza and Viriri, 2021).

2.4.1 Network layers

Input layer

As mentioned above, the input layer is the leftmost layer in the neural network, and the nodes in the layer are referred to as input neurons. These neurons do not perform any calculation or sum, and only provide the input data to the next layer.

Convolutional layer

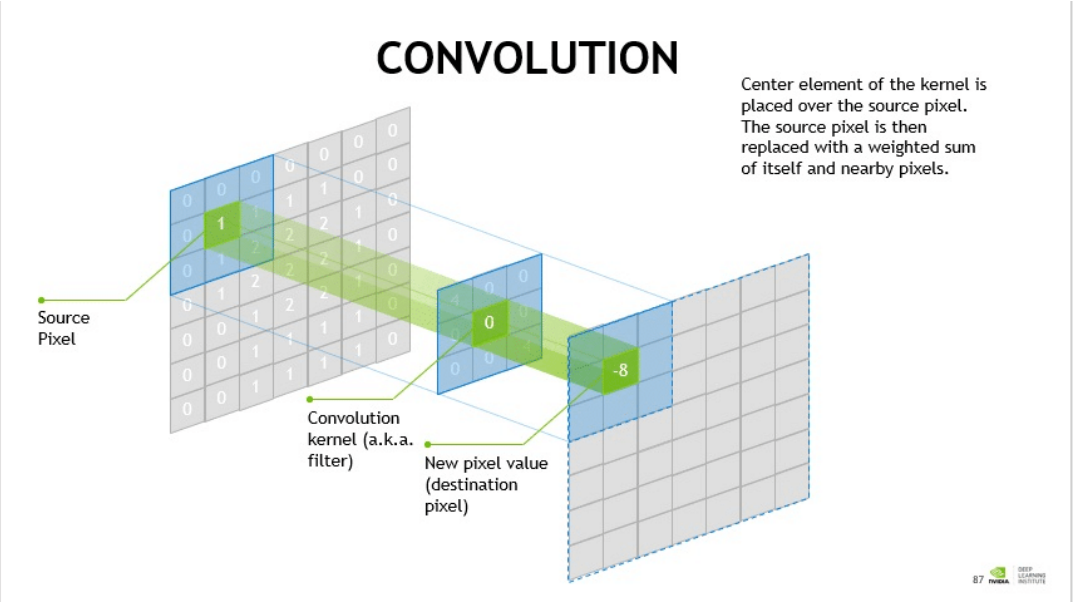

FIGURE 2.2: A convolutional layer.

The convolutional layer in a neural network consists of a set of learnable filters/kernels that slide over the entire input. This is usually the first layer in the CNN. The way in which convolutional layers work is by making use of filters/kernels, and receptive fields (Magadza and Viriri, 2021).

A filter works to localise and generalise the input and to evade connecting every single input neuron to the layer. Instead, a smaller region of the input layer, called the local receptive field, is selected for a neuron in the convolutional layer. Since this neuron requires a weight and a bias, a shared weight and bias is defined for the local receptive field, and this is what is defined as the filter. The filter is an array of numbers that slides across the input or convolves around the input, to generalise a local region of input (Deshpande, 2022).

The filter works by multiplying the values of the shared weights and biases with the values of the input neurons in the local receptive field, carrying out element-wise multiplication – and producing a single number. This number is now the representative of the entire local receptive field it originated from in the next layer. This process is repeated for every location of the input, and this is known as sliding across the input. The resultant values from this convolution are known as an activation map, or a feature map (Deshpande, 2022).

Pooling layer

Pooling layers often follow after convolutional layers, with the goal being to reduce the dimensions of the feature maps that have been produced, whilst still maintaining as much of the important information at the same time. Pooling layers help to decrease the computational power required since they reduce the dimensions of the layers via a process of merging. This is done via one of two methods: max pooling and average pooling (Mandal, 2021).

Max pooling layer

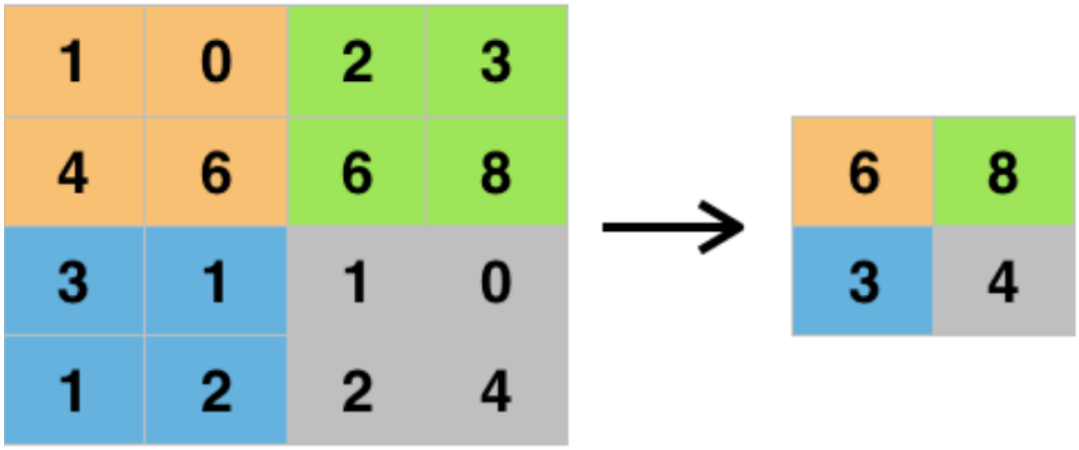

FIGURE 2.3: A max pooling layer.

Max pooling slides a window across the input and only takes the maximum value of the input through. Alternatively, average pooling takes the average of this entire window of values and takes this through instead. In this way, the dimension of the data is reduced. Although this concept may seem like valuable data is being lost, particularly in the max-pooling case, rather the process is obtaining the meaningful part of the data and removing noise. This assist both in reducing overfitting and speeding up computation (Jeong, 2019).

Dropout layer

One of the biggest problems that convolutional neural networks face is overfitting. This is where the model has just learned to recognise the data that it has been trained on and cannot generalise what it has learned to unseen data. Dropout layers assist in combatting this. They work by randomly dropping neurons and their connection from the network during training. Dropout layers also have a positive effect on the performance (Srivastava et al., 2014).

Flatten layer

Flatten layers convert the input into a 1-dimensional array for input to the next layer. For example, if the output was a 32x32 array, we can flatten this to be a single long feature vector of size 1024 instead. This is usually useful before fully connected layers(Jeong, 2019).

Fully connected layer

A fully connected layer, also known as a dense layer, is where every neuron in the previous layer is connected to the current layer. This is opposite to the concept of the convolutional layer, where the purpose is to not have every neuron connected. This is because convolutional layers are used as feature extractors but fully connected layers are used for actual classification (Magadza and Viriri, 2021). The fully connected layer works using matrix-vector multiplication, and since every neuron in the previous layer is connected to every neuron in the fully connected layer, this is a large number of operations to carry out – making it computationally expensive.

Fully connected neurons can be stacked together, meaning more than one is used consecutively, but eventually, the final one must be connected to the output layer, and this returns the output that the model has been trained to find. In these layers, the model is determining which features most correlate to a particular class, using the activation maps and other output produced in the previous layers (Deshpande, 2022).

2.5 Training

2.5.1 Backpropagation

The process of training a convolutional neural network is the process by which the kernels, weights, and biases discussed earlier are found, with the aim being to minimise the difference between the output prediction and the actual output we are expecting. The most commonly used algorithm to do this is backpropagation (Yamashita et al., 2018). An overview of this can be thought of as initially, we randomly assign weights and set biases to 0, and then feed the input forward through the network. This is known as the forward pass. After comparing the output to the ground truth label (what we know the output should be and expect the model to learn to produce), we calculate the error using a loss function. To reduce the difference between the model output and the expected output, we aim to minimise the error, so we update the parameters of the nodes. To do this, we propagate the errors back through the network – also known as backpropagation, or the backward pass– and use an optimisation method to choose new parameters (Weigel, 2022).

Backpropagation can then be split into four distinct sections: the forward pass, the loss function, the backward pass, and the weights update. The forward pass

provides the input to the first layer, which subsequently provides the input for the next layer, and so on until we have an output. If we initially set the weights and biases randomly / set them to 0, then the output layer will likely give equal weighting to each possibility, and the classification has failed. The loss function then uses the ground-truth label, and determines how bad the prediction made was, and depending on this, we aim to find the weights which contributed the most to this loss. This is done via the backward pass, where the gradient/derivative of the loss function with respect to the parameters of the previous layer is taken, allowing for an estimate of how much each layer impacted the result. The result from this can then be used in the weight update, by subtracting the gradient and multiplying by the learning rate. This updates those weights that were found to be problematic, by changing them to be in the opposite direction. This entire process is known as one training iteration, or one epoch – i.e. one forward pass, backward pass, and weight update having been carried out (Deshpande, 2022).

The model is constantly evaluated during the training process, to assist in hyperparameter selection and to assess whether the model is learning effectively. This can allow the model to stop running more epochs, thereby saving on computational resource.

It is this process of training a model that uses the most computational resource, due to the number of iterations/epochs needed, and the number of computations that are carried out in each forward pass.

Many of the concepts discussed here are widely variable, and the choice is down to the person creating the model. The selection of some of these options is known as hyperparameter selection. This is a parameter whose value is used to control the learning process, different to actual parameters that are learned during the process. These can include the loss function discussed above, activation functions, optimisation algorithms, number of layers and types of layers used, number of epochs, weight initialisation function, and more (Yu and Zhu, 2020). Some of these will be discussed here.

2.5.2 Loss function

A loss function, also known as a cost function or sometimes the objective function in the context of optimisation, in machine learning is used to evaluate the current state of a model given the current data. A higher loss value indicates that the output that is produced by the model is far from what the ground truth suggests it should be, and a lower loss value indicates that the output is close to the ground truth. The main goal of training a neural network is to minimise the loss function while trying to ensure that the network is generalisable to unseen data (Magadza and Viriri, 2021).

The loss function is then used during the backwards pass to update the weights and biases. The choice of the loss function is important, and a well-chosen loss function can even assist in the efficiency of the model by speeding up the convergence rate. There are many different types of loss functions, and the choice should be made depending on the problem area. Loss functions for classification include hinge loss, cross-entropy loss, and Kullback Leibler Divergence Loss (KL loss)

Hinge loss

Hinge loss is primarily used for another type of machine learning – known as support vector machines (SVM). It calculates the maximum margin from the hyperplane

(a feature of SVMs) to the class itself. This means that even if an observation is classified correctly, they can incur a penalty from the loss if the margin from the decision boundary to make the classification is not large enough (Sebastian, 2021). Although there has been some research and models developed using hinge loss as the loss function of choice in deep learning, it is not very common (Janocha and Czarnecki, 2017; Ozyildirim and Kiran, 2021).

Cross-entropy loss

Cross entropy loss is a more commonly used loss function in machine learning, particularly in classification using deep learning. Cross entropy loss, also known as log-loss, measures the performance of a classification model and the value increases as the predicted probability diverges from the actual expected label. It is derived from the field of information theory, using the concept of entropy and the difference between two probability distributions (ML Glossary, 2022; Brownlee, 2019).

The way in which cross-entropy loss works – i.e., the higher the difference between the label and the output, the more severe the loss produced, makes it a good choice for classification tasks with neural networks.

KL loss

KL loss works similarly to cross-entropy loss, in that it aims to quantify the difference between probability distributions. The KL loss quantifies how much one probability distribution differs from another, and in many machine learning cases, is identical and interchangeable with cross-entropy (Brownlee, 2019).

2.5.3 Activation function

As discussed earlier, an activation function is part of a perceptron, and by extension, one of the defining features of a node in a neural network. It is the function that is applied to the weighted sum of the inputs that the node has received. It is important as it helps to preserve the important information while suppressing the irrelevant data from the input. They introduce non-linearity to the network, to improve efficiency and accuracy (Hong, 2020).

Without activation functions, a neural network would just be a simple linear function, which would be simple to implement, but its complexity and ability to learn would be extremely limited. It is also for this reason that non-linear functions are chosen for activation functions, meaning those functions that have curvature when plotted. They give the model the ability to be dynamic and extract complex and complicated information from data, allowing a non-linear mapping from inputs to outputs. Another key consideration of these functions is that they must be differentiable so that they can be used during backpropagation to minimise the loss and optimise the weights (Sharma, Sharma, and Athaiya, 2020).

There are many choices for activation functions, including binary step function, sigmoid function, tanh function, rectified linear units (ReLU) function, softmax function, and many more. Some of these will be discussed here.

Sigmoid function

The sigmoid function ranges from 0 to 1 and can be applied in the output layer of classification. It is one of the most commonly used activation functions. This is

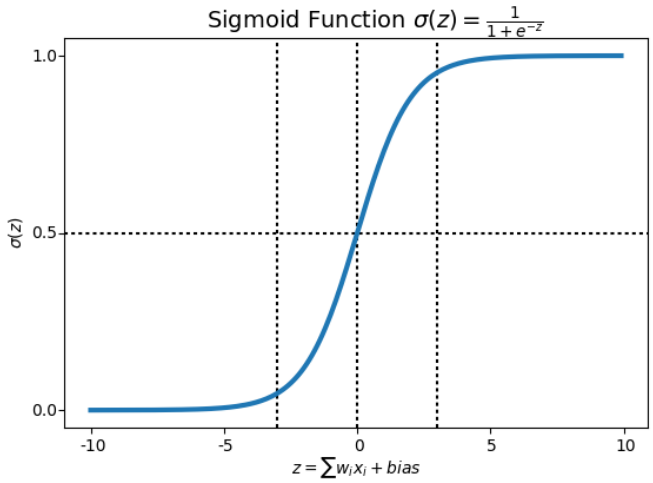

FIGURE 2.4: Sigmoid function.

because it presents a softer version of the signum function, and therefore is differentiable more easily as mentioned is needed for backpropagation. Also, the gradient is steeper around 0 and smooths out as it moves further away from either side, allowing the optimiser to update the weights in a way such that the output is pushed either towards 0 or 1, leading to a faster convergence rate (Santosh, Das, and Ghosh, 2022). Some drawbacks of the sigmoid function are that it can be slow due to the exponentiation in its function, it is not zero-centric, and so the mean activation is not 0 – which has been found to enjoy faster convergence, and it suffers from a common problem in activation functions, the vanishing gradient problem. This is a problem where during backpropagation, the derivative is very small, preventing the weight from changing value (Basodi et al., 2020).

Tanh function

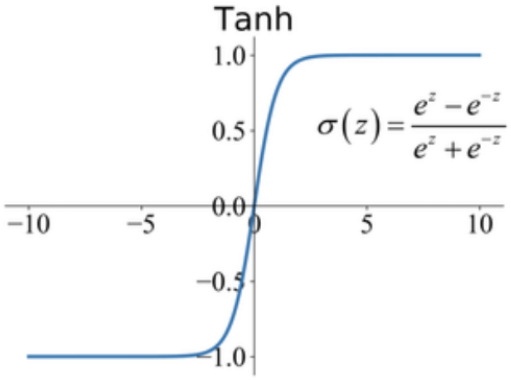

FIGURE 2.5: tanh function.

The tanh function comes from trigonometry and is generally a good function when the output of the neuron is desired to be between -1 and 1. It shares most of

its' properties with the properties described in the sigmoid function section, with the key exception being the range of values that it can take. It does have some advantages over the sigmoid function, namely that the derivatives are steeper and the output is 0-centric. However, just like sigmoid, tanh suffers from the same vanishing gradient problem. This makes both of these functions inefficient when stacking together layers of neutrons, and instead, the ReLU function was proposed (Santosh, Das, and Ghosh, 2022; Nair and Hinton, 2010).

ReLU function

Since its inception, ReLU has been the activation function of choice for most deep learning applications (**nair**; Nwankpa et al., 2018). It is a faster learning function and offers better performance and generalisation in deep learning in comparison to both sigmoid and tanh functions (Dahl, Sainath, and Hinton, 2013).

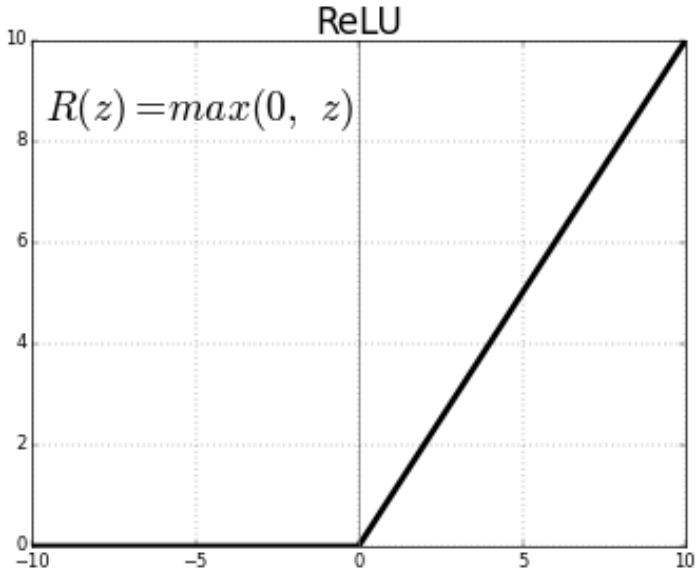

FIGURE 2.6: ReLU function.

It is based on incorporating non-linearity in the standard linear function, which has the beneficial property of having a gradient of 1. This means the chain of partial derivatives computed in the backpropagation step is not affected by a vanishing gradient that the prior two functions were affected by. This is beneficial from a computational cost perspective, as well as being further beneficial in this same aspect, because the function commonly outputs 0, creating sparsity in the system, which results in less computationally expensive models (Glorot, Bordes, and Bengio, 2011).

The reason that many neurons are dropped from the network is that if the activation of the node falls below 0, then the neuron is disconnected. The drawbacks of ReLU are that it suffers from 'dying ReLU' problem, which halts the updating of weights and stops the flow of information to the proceeding layers. This can be rectified using leaky ReLU, which allows a small constant gradient in the negative zone to recover the weight if required (He et al., 2015). Another option is ELU (exponential LU), which introduces a parameter slope for the negative values of the function, however, this introduces more computational cost due to the exponentiation added (Clevert, Unterthiner, and Hochreiter, 2015).

Softmax function

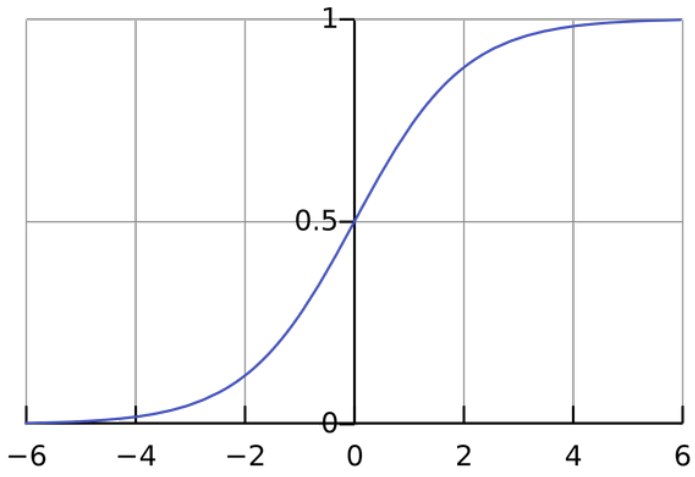

FIGURE 2.7: Softmax function.

The softmax function is another function that is used often in deep learning. It is used to compute a probability distribution from a vector of real numbers, and produces an output between 0 and 1, with the additional property that the sum of all the probabilities it produces is 1. This makes it particularly useful for the final layer of the neural network, where we classify the input into the appropriate output class. There are not really any major drawbacks of the function, but it may not have a better performance compared to other functions such as ReLU (Sharma, Sharma, and Athaiya, 2020).

2.5.4 Optimisation

Both the loss function and activation functions discussed so far have been part of the wider aspect of training the neural network, and both have been mentioned to be important. As part of this discussion of their importance, it was discussed that they are both used as part of the wider optimisation function of the network. This is what will be discussed in this section.

Optimisation as a standalone topic requires an objective function, and in the case of deep learning, the objective function is the loss function, and we are aiming to minimise this. Although there is a lot that can be borrowed and utilised from the branch of optimisation for deep learning, ultimately, they are too different. Whereas optimisation simply aims to minimise an objective, deep learning is focused on trying to find a suitable approximation based on finite data. Not only this, but the optimisation we carry out in deep learning aims to minimise the training error, but in reality, we wish to reduce the generalisation error, so attention must also be given to overfitting when carrying out optimisation (Zhang et al., 2021).

The performance of a convolutional neural network is improved through the process of training, utilising the activation and loss functions above. The method by which we use these functions to update the weights is known as the optimisation algorithm or function. This optimisation can be done per single sample, subset, or the full set of training samples (Magadza and Viriri, 2021).

Gradient descent techniques

The principles of gradient descent are what the most popular types of optimisation used in deep learning are built on. Gradient descent given a cost function and initial set of parameters, runs iteratively to find the optimal values to obtain the minimum value of the function. This is done by taking the derivative at each iteration and updating the weight to go in the opposite of the direction found via the derivative.

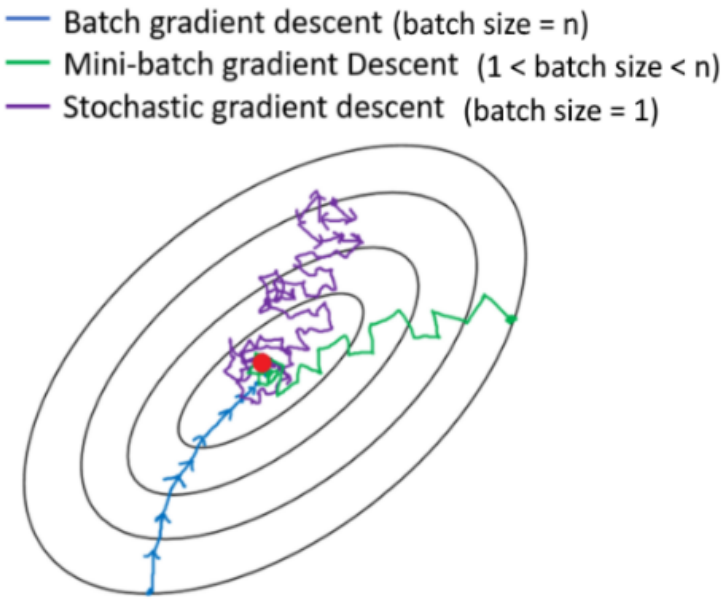

FIGURE 2.8: The differences in paths of the different gradient descent techniques.

There are then different types of gradient descent, such as batch, stochastic, and mini-batch.

Batch gradient descent

Batch gradient descent calculates the gradient of the entire dataset on each iteration before the weight is updated. This involves taking the sum of all individual training samples, and is then evidently computationally expensive – making it unfeasible for large datasets (Wilson and Martinez, 2003).

Stochastic gradient descent

Stochastic gradient descent (SGD) aims to address this problem by opting to update the weights for each training example rather than calculating the gradient over all training examples. It does this by estimating the gradient-based on a single randomly picked example (Bottou, 2012). Because SGD calculates the gradient of the loss function for a single example at each iteration, the process to reach the minima is usually noisier, but this is mitigated since the minima is still reached and in a significantly shorter time. This also means that SGD requires more iterations, but it is still computationally less expensive than batch gradient descent. Another advantage is that the calculation time does not depend on the total number of training samples,

further saving computational resource. However, a drawback of stochastic gradient descent is that it can be difficult to choose an appropriate learning rate, and also that the solution can become trapped in a saddle point in some cases (Sun et al., 2019). SGD is a common type of optimisation used in deep learning.

Mini batch gradient descent

Mini-batch gradient descent combines concepts from both of these methods, splitting the training dataset into small batches, and performing updates on these batches instead. This attempts to balance computational resource required and speed. By sampling a subset of the data, the process requires fewer iterations as it learns quicker, and at the same time, it requires less computation as the whole dataset is not processed at once. For this reason, mini-batch gradient descent is the most common optimisation algorithm used (IBM Cloud Education, 2020).

Adaptive gradient

Similar to the other algorithms, adaptive gradient (adagrad) is a gradient-based technique, with the ability to adapt the learning rate based on the data at each iteration. It is particularly useful in sparse datasets as it has a higher capability to learn rare and infrequent features (Tahmassebi et al., 2018).

Adaptive moment estimation

Adaptive moment estimation (ADAM) optimisation is one of the most popular algorithms in recent times. It builds on the adagrad algorithm mentioned above, and also adds another idea – momentum. The notion of adding momentum is to force the gradient descent to keep moving in the same direction as the previous iterations. This is done by keeping track of the running mean of the gradients up until the current time – or the velocity – and a value referred to as friction, which is a constant that aims to decay. At each step, the velocity is updated by decaying the previous velocity by the friction constant, and the derivative of the current weights is then added to the current time. The weights are then updated in the direction of the velocity calculated. This momentum allows for the escape of local minimums as mentioned above, and also reduced the noise of the gradients. There is also a secondary momentum value that is tracked, called the squared gradients average (Kingma and Ba, 2014).

This method has gained a lot of popularity in modern times, and this is because it is relatively stable, and utilises the best features of many other techniques. It is also highly suitable for problems with large datasets and in higher dimensional spaces. A drawback is that it may not converge in some cases (Sun et al., 2019).

2.5.5 Initialiser

Deep learning training is an iterative process by nature, and the first forward pass requires the creator of the model to set initial values for the weights. This choice of initialisation can greatly influence the speed with which the model will converge, or if it converges at all. The effectiveness of the initialisation then also affects the computational efficiency of the model training (ML Glossary, 2022).

Initial values that are set too large cause issues during the backwards pass, as the gradients can then begin to 'explode', the opposite of the vanishing gradient

problem discussed earlier. Alternatively, initial values that are too small can also cause this vanishing gradient problem.

Random

As is evident from the name, this type of initialisation sets the parameters to be randomly sampled for the first pass. It serves the purpose of breaking symmetry, whilst also giving much better accuracy. The weights are initialised to values very close to 0 randomly, so each neuron is not performing the same calculation (Barrera, 2021).

Xavier/Glorot

This is a more advanced random initialisation technique and can be used for the sigmoid activation function or the tanh activation function. It works by setting an equal variance of the inputs and outputs of each layer to avoid vanishing gradients. This is done by initialising the weights from a normal distribution with a mean of 0 and variance. This technique was developed when sigmoid activation was the default choice of activation. Once ReLU became popular, then this initialisation technique was not as effective (Murat, 2019).

He Normal

Once ReLU began gaining popularity, a new technique was developed to initialise the weights, using the same principles as the Xavier initialiser – i.e., balancing the variance of the activation. The weights are drawn from a normal distribution with zero mean, but this time, the variance is multiplied by a factor of 2. The size of the previous layers is considered with this technique, assisting in attaining global minimum faster and more efficiently (Kakaraparthi, 2018).

2.6 Other considerations

2.6.1 Overfitting and underfitting

The goal of machine learning is to be able to learn to make inferences from data that it has not seen. Although the training process of the model aims to optimise the loss, in reality, the goal is to be able to use the model on unseen data. In other words, it aims to be able to generalise its weights and biases to be able to make inferences on the population, having learned from a sample (Peng and Nagata, 2020).

Often, this is difficult to achieve, as the model has learned from the training data too well, and cannot generalise to unseen data. This phenomenon is known as overfitting. This is one of the key issues that supervised learning has. When learning algorithms fit the training data so well, it learns the noise, patterns, and specificities of the training data itself, rather than learning the features of the data. In predictive modelling, this is known as learning noise rather than signal, where the signal is the true pattern that is sought to be learned, and noise is the randomness in a specific sample (Silver, 2012). A common cause of this is a small dataset, as the model does not have enough data to effectively learn from (Jabbar and Khan, 2014). Overfit models tend to have less bias, but more variance – meaning the model has not oversimplified the solution but has overestimated the importance of the training data.

The opposite of overfitting is underfitting, where the model struggles to even learn to recognise any patterns from the training data. The model is too simple and makes it inflexible to learn from the dataset. These models tend to have less variance in their predictions, but more bias toward the wrong outcome – alternatively, this means that the models have oversimplified to make the function easier to estimate, but hasn't underestimated the amount the target function will change (Elite Data Science, 2022).

Models with too much bias and low variance are not complex enough and are unable to learn the signal from the data. Alternatively, models with too much variance and low bias are too complex, and simply memorise the noise in the data rather than the signal. This leads to the bias-variance tradeoff, where we aim to train a model enough so that it is not biased, and not so much that it has too much variance. This is the ideal point where the model is neither overfitting nor underfitting (Doroudi, 2020).

Some techniques that can be used to combat overfitting are to reduce the complexity of the model, use regularisation – such as early stopping and dropout which has already been discussed, use validation, and use more data.

2.6.2 Validation

Rather than split the dataset into only a train and test set, the concept of validation is to add a third partition which is used to validate the training. During training, this validation set is used to assess how well the model is learning, and this can help prevent overfitting by validating on data that the model has not seen (Solawetz, 2020). This can also cause further overfitting towards the validation set however, and an alternative technique is to use cross-validation.

Cross-validation is a more powerful technique where the data is split into k folds, and the algorithm is trained using k-1 folds, with the final fold used as the test set. This allows the model to learn and be tested on completely unseen data and prevents overfitting of the validation set (Baheti, 2022).

2.6.3 Augmentation

One of the key causes of overfitting is a lack of data, and sometimes, it is not possible to just obtain more data, possibly due to a lack of high-quality data. In these cases, augmentation can be applied. Augmentation is the process of extending a dataset by applying different augmentation techniques. These are rotation, skewing, flipping, shearing, Gaussian blurring, sharpening, and embossing (Sajjad et al., 2019).

By applying these transformations, new data points can be created and assist the model in learning without overfitting.

2.6.4 Dimensionality

The nature of MRI scan data means that the number of features that are present in each training point is very large. In machine learning, as the number of features increases, and the dimensionality of the space increases, models often find it harder to learn, models are harder to design, and importantly for the context of this project, they have higher running times – by virtue of the fact they are more computationally expensive. Although storing data in higher dimensions allows for more information to be stored, and perhaps spatial features to be conserved, it does not always due to the amount of noise and redundancy in data (Choudhury, 2019).

Techniques can be employed to reduce the dimensionality of data, known as dimensionality reduction techniques. There are two ways in which this can be done: "[...] by only keeping the most relevant variables from the original dataset (this technique is called feature selection) or by exploiting the redundancy of the input data and by finding a smaller set of new variables, each being a combination of the input variables, containing basically the same information as the input variables (this technique is called dimensionality reduction). (Sorzano, Vargas, and Montano, 2014)"

The most common technique for dimensionality reduction is principal component analysis (PCA), which converts the original input to a new set of data, which is a linear combination of the original. These are unrelated and are hence known as principal components (Bhattacharjee, 2017).

One of the drawbacks of PCA is that it is computationally expensive, due to the amount of matrix multiplication it carries out, as well as needing to calculate covariance matrices, and eigenvalue decompositions. Although alternative methods using singular value decomposition are available, this is still expensive (Banerjee, 2020).

2.7 Sustainability

There has been a lot of discussion over the last few years pertaining to the sustainability of AI and machine learning, and movements such as 'green AI', and 'sustainable AI' are being started.

"[...] [it is time to] address the sustainability of developing and using AI systems. In this paper I propose a definition of Sustainable AI; Sustainable AI is a movement to foster change in the entire lifecycle of AI products (i.e. idea generation, training, re-tuning, implementation, governance) towards greater ecological integrity and social justice (Wynsberghe, 2021)."

"The emergence of Artificial Intelligence (AI) and its progressively wider impact on many sectors across the society requires an assessment of its effect on sustainable development. (Yousra, Abdelhakim, and Mohamed, 2021)"

Developing models that minimise the amount of time and power is one of the biggest challenges in modern AI since the amount of time and computing power needed to train machine learning models is quite considerable (Ning, Guan, and Shen, 2019).

This is particularly true for CNNs, as although they provide extremely powerful features, they also come with considerable storage, computational, and energy requirements. An example of this is the VGG-16 CNN, which has over 135 million parameters, requiring more than 15 billion floating-point operations per second (FLOPs), to simply classify a single image of size 224x224 (Qin et al., 2021).

Other research reiterates this point, noting that CNNS require much more computational resource compared to some other systems. Although computational capacity continuing to increase allows for the extension of the limits, the resource can never be limitless, and so, "[...] continuously improving the computation efficiency, i.e., performing a given function with less computation, is a critical issue in designing CNN systems. (Sun and Wang, 2020)"

Hence, one of the primary concerns alongside producing an effective model capable of accurately classifying brain tumours into their correct grades in this project was to balance the computational cost and efficiency of the model and the process of creating the model.

Since 2012, the amount of computational resource being used for the training of machine learning models has been increasing exponentially, with a growth rate far greater than the similarly defined Moore's law – with Moore's law having a 2-year doubling period and the amount of compute used for training doubling approximately every 3.5 months. Although improvements in hardware have made this growth possible, it is worth preparing for systems that may not be able to deal with increasing demand (OpenAI, 2019).

Besides the consideration of the potential physical limitations of hardware, there are more serious reasons to need to consider the sustainability. The electrical energy that is required and consumed when training AI, as well as the building, collecting and storing of the physical resources that are used to process them for many providers such as Google Colab, contribute to increased carbon emissions. By using an estimation of the amount of CO2 emitted on average per kilowatt consumed, an estimate for the energy used by Amazon Web Services is comparable to that of the entire United States of America (Mahipal, 2021).

GPT-3, an AI released by OpenAI was trained on the entirety of English Wikipedia, while only making up only 0.6% of its training data – illustrating the vast amount of computation and energy that is being consumed by modern AI research (TheNextWeb, 2022). The University of Massachusetts estimated that training a large deep-learning model produces 626,000 pounds of CO2, equal to the carbon emissions of 5 cars across their entire lifetime (Strubell, Ganesh, and McCallum, 2019). Similarly, another study found that the process of 'building and testing a final paper-worthy model' when converted to an approximation for carbon emissions is equivalent to 78,000 pounds of CO2. One of the key recommendations made by the researchers from the university was a "concerted effort by industry and academia to promote research of more computationally efficient algorithms [...] (Strubell, Ganesh, and McCallum, 2019)."

The energy required to train models is then directly linked to the computational cost of training and producing the model, although it is not limited to this. Many factors can contribute to this computational cost, such as the number of operations being carried out, the load on the GPU due to the size of the dataset and files, or even the types of operations being carried out, such as exponentiation.

With this background and theory in mind, the goal of the project was to create a convolutional neural network that is capable of accurately classifying brain tumours – namely gliomas – from MRI scans. The model will need to be sufficiently trained so that it is not overfitting, and can generalise well to the test data. There have been many papers and research that have already done this to a higher level, however, as a result of new demands in the industry, considerations will be made where possible to select the most computationally efficient methods and techniques, as well as to attempt novel techniques to carry out tasks if this assists in reducing computational resource needed. This is in comparison with the majority of the research carried out to classify brain tumours which only focuses on improving accuracy.

Chapter 3

Methodologies

3.1 Software and hardware used

To create and train the model, many different pieces of software were used. The key components were:

Google Colab This is a "[...] product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing access free of charge to computing resources including GPUs."

The main advantages of using Colab are access to more powerful resources, such as powerful GPUs (graphical processing units) and CPUs (central processing units) with higher RAM (random access memory) capacities. These are all vital to being able to train machine learning models effectively (Google, 2022).

Colab uses the Ubuntu version of Linux, so the entire project can be considered to have been trained and run on Ubuntu.

Additionally, Colab comes packaged with many libraries and modules that are frequently used, removing the need to download and install them yourself.

Python Python version 3.7 was used via Colab. Although this is not the latest version, Colab aims to maximise compatibility worldwide.

Keras and TensorFlow To create, train, and test the model, TensorFlow was used in conjunction with Keras. TensorFlow allows for the efficient execution of low-level tensor operations on CPUs and GPUs and provides much of the functionality needed. Keras is a high-level API which provides an interface to TensorFlow and makes creating CNNs much easier.

KerasTuner KerasTuner is a hyperparameter optimisation framework that runs the training of the model repeatedly to assist in finding the best hyperparameters.

Hardware specifications The Google Colab hardware that was assigned at runtime was a Tesla T4 GPU with 16GB RAM, and 8 Intel Xeon CPUs @ 2.00GHz, with 25GB VRAM.

3.2 Dataset

Much investigation was carried out to determine a suitable dataset for the task at hand. Commonly used datasets in existing research include Multimodal Brain

Tumor Segmentation Challenge (BRATs) datasets, The Cancer Genome Atlas Low-Grade Glioma (TCGA-LGG) dataset, and the Repository of Molecular Brain Neoplasia Data (REMBRANDT) dataset (Menze et al., 2015; Pedano et al., 2016; Scarpace et al., 2019). However, none of these were found to be suitable for the goals and objectives of this project.

Some of these datasets required extensive data preprocessing before they could be used. Others did not have the necessary information required, such as the tumour WHO grade, or the correct MRI modalities to be able to carry out the project.

Eventually, the Erasmus Glioma Database (EGD) was selected as the dataset of choice for the project. This dataset came with many benefits that assisted in the training and creation of the network. Firstly, for all patients, four different modalities were provided, including the one chosen for this project – T2-FLAIR (Voort et al., 2021).

In addition to this, a lot of preprocessing was already done or provided for ease of use. The dataset consisted of scans of 774 patients with gliomas, with 502 patients having a grade 4 tumour, 77 patients having a grade 3 glioma, and 135 patients having a grade 2 glioma, with the remaining patients having no information for the grade. One of the limitations of the dataset is that it has no patients with grade 1 tumours – but this is an extremely rare feature to find in any public dataset. Most research that has carried out four-class classification has obtained private data.

The scans were created via four different machine vendors but were all provided in the same dimensions – 197x233x189 voxels (a pixel with volume). All the scans were converted to the NIFTI file format, and registered to the MNI152 atlas, as well as having been affinely registered to their respective atlases using Elastix version 5.0.0. After registration, the scans were defaced to include as much of the skull as possible while simultaneously removing facial features.

Segmentation masks were also provided for each patient, which provides a delineation of the area where the tumour is present on the MRI. Furthermore, a skull-stripping mask was provided as part of the dataset.

All of these things reduced the need to carry out these steps as part of the project and reduce the amount of computation required to carry out these steps. This makes this dataset useful as if these steps are carried out once initially, it removes the need for every subsequent person to use the dataset to carry out the steps – greatly reducing the computational resource used.

3.3 Dataset preparation

To prepare the dataset for use in the neural network, many steps had to be carried out to simplify the structure of the data. The files once downloaded came in a convoluted folder structure, with each scan containing many subfolders, most of which were unnecessary. To simplify the structure of folders, some BASH scripts were developed to unpack them. As mentioned earlier, the dataset was largely imbalanced, with there being abundantly more grade 4 data points, as well as more grade 2 data points than grade 3. To avoid the issues of imbalanced datasets, the number of samples used for each grade was limited to 77 – which was the maximum number of samples available for grade 3 gliomas (Lemaitre, Nogueira, and Aridas, 2016). The 77 samples for the other two grades were randomly selected.

Once the dataset consisting of 231 samples was selected, the next step of preprocessing could begin.

3.4 Preprocessing

The preprocessing of the dataset is an extremely important part of the development of a machine learning model. This preprocessing is where the raw dataset is transformed into a clean dataset. Inconsistencies are normalised, formatting is made appropriate, and transformational and feature extraction steps are carried out to expand the performance of the model (Singh et al., 2021).

The steps carried out in the project include skull stripping, region of interest enhancement, resizing, augmentation, dimensionality reduction, normalisation, and change of file type.

3.4.1 Loading the data

Before the preprocessing steps can be carried out, the MRI data must first be loaded into the environment. The MRI data is provided in the NIFTI (Neuroimaging Informatics Technology Initiative) format – a file format for neuroimaging (Neuroimaging Informatics Technology Initiative, 2013). These files are registered in a local coordinate system and contain metadata as well as the data itself. The data is organised into a 3D array of voxels (Benson, 2022).

To process the data within the notebook, the NIFTI file is loaded using a library called NiBabel, which provides tools to work with NIFTI files in Python (NiBabel, 2022). Once this is done, the data can be stored as a regular array, using the Python NumPy library (NumPy, 2022). This allows for easier access and processing of the data and reduces the need to keep using the NIFTI file, which contains a lot of irrelevant data.

3.4.2 Skull stripping

Deskulling a brain scan, also known as skull stripping and brain extraction, is the process where non-brain tissue signal is removed from MRI data. The process is useful to anonymise data, but also has major benefits for the neural network training itself. It removes irrelevant and distracting tissue, and this usually leads to better performance. This skull stripping should be performed before other steps that may be applied to the brain, and so this was the first step in the preprocessing pipeline (Kalavathi and Prasath, 2016; Hoopes et al., 2022).

The skull was stripped using a mask provided with the dataset. The mask identifies the voxels in the scan where skull tissue is present, and this was used in conjunction with the Python NumPy library. The brain mask has the same dimensions as the brain scan itself, and the voxels where the mask has identified there to be a skull have values of 0, with the rest of the voxels being 1. This means that a simple element-wise multiplication of the two 3D arrays consisting of the brain data and the mask data would set all the voxels where there is non-brain tissue to 0, thereby achieving the intended purpose of deskulling the brain.

3.4.3 Resizing

The scan data was then resized from the dimensions 197x233x189 voxels to 128x128x64 voxels. The purpose of this is to reduce the size of the image, and this is to improve the efficiency and computational cost of the model (Rukundo, 2022). Due to the original dimensions being non-uniform, the result is a slightly warped image, but the key features of the image are preserved, and the warping is not egregious. The

result is a square shape with a factor of two can contribute to model performance, since pooling and other layers can process the input more easily. Some detail of the original image is lost, but the tumour and brain tissue can still be easily identified. This step balances preserving the information and improving performance.

3.4.4 Region of interest enhancement

Since the dataset provided a delineated mask with the tumour region, this was used as part of the preprocessing steps in a slightly novel way. Research has been carried out that shows that the performance of brain tumour classification can be enhanced via tumour augmentation and partition (Cheng et al., 2015). In this project, the tumour mask was used to enhance the tumour itself in the brain scan. By enhancing the intensity of the voxels present in the tumour mask in the brain scan, the visibility of the tumour is increased in the brain scan, and this is expected to improve the performance of training as well as the model. This intensity enhancement was done by increasing the voxel value to a normalised max relative to the values in the brain scan. This enhancement was intended to improve performance and reduce computational costs.

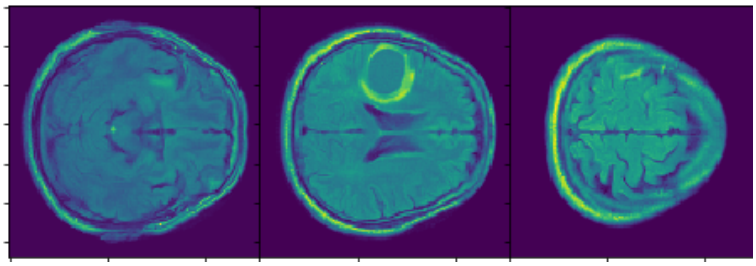

FIGURE 3.1: Slices of a sample point before the skull has been stripped, and ROI enhanced.

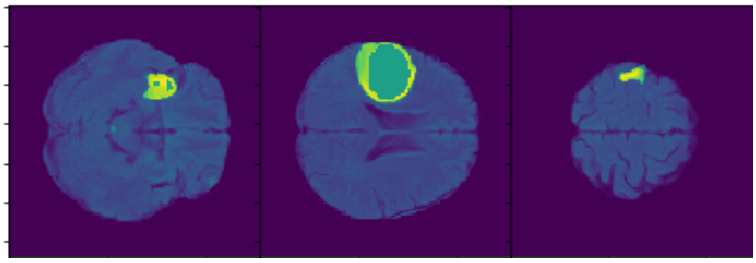

FIGURE 3.2: Slices of a sample point after the skull has been stripped, and ROI enhanced.

3.4.5 Augmentation

Dataset augmentation is the process of extending a dataset by applying different augmentation techniques. The result is a more diverse set of data, which can allow the model to train better, generalise better, and also reduce the possibility of overfitting. The accuracy of deep learning models depends largely on the amount of data available for training. As mentioned earlier in the dataset section, the number of

samples available per class is extremely low. To combat this, data augmentation, the process of artificially increasing the amount of data was applied.

Studies have shown that rotational augmentation is particularly effective in extending datasets, and so as part of the augmentation, each sample was rotated at several angles: 15, -15, 30, -30, 45, -45, and 180 degrees (Safdar, Kobaisi, and Zahra, 2020). This increased the number of samples by a factor of 7. A secondary augmentation technique was also applied, called gaussian blur. This adds Gaussian noise and also assists in preventing overfitting of the model. The final result of the augmentation meant that for each sample, there were now 14 more data samples available, transforming from 1 to 15 samples. For the wider dataset, this meant that the number of samples in each class went from 77 to 1155, and the full dataset went from a size of 231 to 3465 – largely improving the likelihood of training the model successfully, as well as improving the chances of creating a generalisable model.

The augmentations were all carried out using a Python package from the library SciPy, called ndimage, which assists in the processing of multidimensional image processing (SciPy, 2022). The package provides functions that rotate 3-dimensional arrays by the angle provided, as well as functions that can add gaussian noise. One of the drawbacks of this preprocessing that was carried out was that the 3-dimensional convolution can be computationally expensive.

Although the rotation could be done after the other steps, the blur was more effective at this point, and the decision was made to keep the two steps together rather than separate them.

3.4.6 Dimensionality reduction

An important consideration to make when creating the model was whether to choose a 3D architecture or a 2D architecture. The input data is in 3 dimensions, so a logical choice may be to opt for 3D architecture. 3-dimensional CNNs allow for the preservation of interslice connections, and spatial features that may be shared across the third dimension. However, they also largely increase computational costs (Burton, 2019).

Research does show that 3D CNN models can achieve higher accuracy than 2D, but their computational cost is increased a lot due to the extra dimension in the input. It also shows that a 2D model performed the best, achieving the highest accuracy, and avoiding overfitting. Given the focus on computational cost, the choice was made to opt for a 2D architecture.

This meant that the data needed to be dimensionally reduced. This reduction can be done in many ways. Feature selection is the process by which only the more relevant pieces of information are kept, exploiting the redundancy of irrelevant data. Other techniques may create new variables entirely, based on some combination of the input variables (Sorzano, Vargas, and Montano, 2014).

The most common type of reduction – principal component analysis (PCA) uses the latter, with the key idea being to find a new coordinate system where the input data can be expressed with fewer variables. The process uses a lot of linear algebra; computing covariance matrices and eigenvectors as well as needing to carry out lots of matrix multiplication (Jaadi, 2021). As a result, the choice was made to investigate other methods.

One journal article suggests that the choice of technique should be made based on the input data, as well as your intuition and domain knowledge (Nguyen and Holmes, 2019). As such, an attempt was made to use a novel technique to carry out the reduction. The understanding of the structure of the 3D input was important

here, as it was known that each slice in the third dimension corresponds to an actual 2D slice of the brain. The assumption was then made that the central slices are likely to hold the most important data, and as the slices move further away from the centre, the less likely they are to be of importance. This was inferred from viewing many of the samples in slicing technologies.

Some research was then carried out into techniques such as feature quantization, mean average slicing estimation, as well as random forest for variable importance. The latter technique was interesting and some principles of this were used. Rather than fitting a linear regression model on the slices to determine the best weighting, the inference made in the previous paragraph was used to test different weights, such as quadratic, linear, and exponential.

Quadratic weighting was found to be the best choice via visualisation of the output. To expand on this, for each slice of the data, a range was created from 0 to 1 to 0, with 1 corresponding to the central slice, and 0 for the outer slices. The slices in between were spaced evenly between 0 and 1 depending on the number of slices present. This is just a linear weighting. These weights were then squared to further emphasise the importance of the central slices over the outer slices, and this worked well and is much less computationally expensive than PCA.

3.4.7 Normalisation

The image array data from a NIFTI MRI scan file comes with metadata that defines the maximum and minimum display intensities. Values close to and above the maximum threshold appear white, and those close to and below the minimum threshold are black. Those in-between are a lighter or darker shade of grey depending on the intensity (Cox, 2019).

These values are not in any specific unit, but rather are on a relative scale. This can affect data processing. To avoid dependence on the choice of this scale or the relativity of it, data values should be normalised, which involves transforming the data to fall within a smaller common range, such as between -1 and 1, or 0.0 and 1.0. This can also help speed up the training phase, thereby reducing computational cost (Han and Kamber, 2012).

By normalising the data, the model is given the best chance to learn the solution. A common technique is called the min-max normalisation method, and this was the technique employed in this project.

$$x_{normal} = \frac{x - min(x)}{max(x) - min(x)}$$

Applying the min-max normalisation transforms the values in the image into the range of 0 and 1. This is particularly useful when considering activation functions. For example, ReLU outputs a value from 0 to infinity, and the output increases linearly relative to the size of the input. If the values in the original data are large, the output of ReLU will also be large, and this can cause exploding/vanishing gradient issues during backpropagation. Hence, values between 0 and 1 are better suited to avoid these issues (Gavali and Banu, 2019).

The processed output was then exported as a NumPy array file (.npy), which further improves the computational cost, as these files are larger and require less processing when being read into the program.

The data was then split into train, test and validation sets, with a split of 80-10-10.

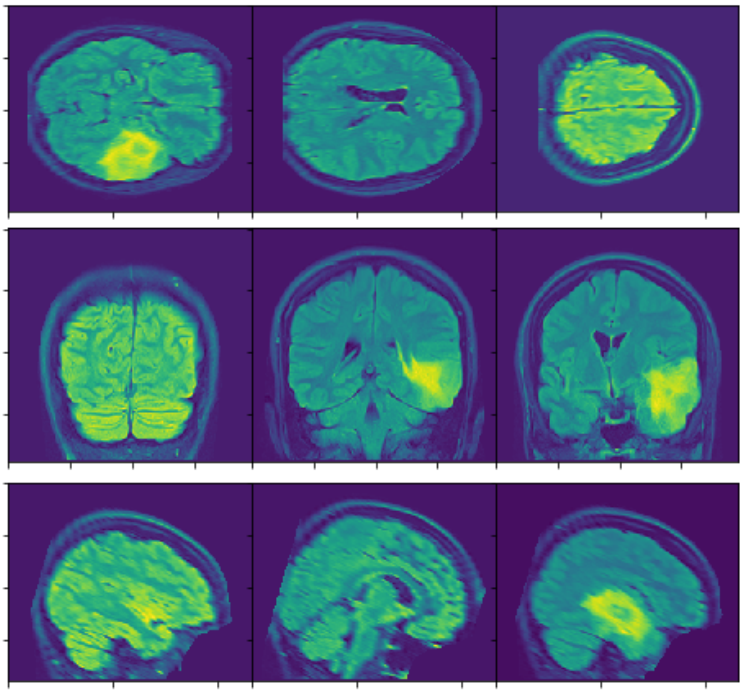

FIGURE 3.3: Sliced view of one of the samples without any processing

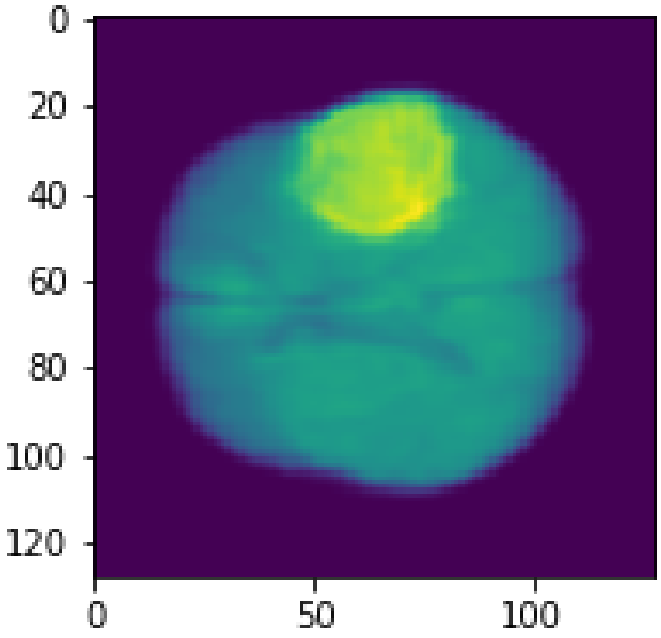

FIGURE 3.4: Result after my dimensionality reduction technique

3.5 Training and hyperparameter tuning

Once the data was ready, the model architecture and hyperparameter selection could now be determined. To select the architecture, prior research was consulted, and a common type of architecture was selected as the base.

As part of the hyperparameter selection, a Python package called OptKeras was used to assist in the finding of these hyperparameters, including the number of fully connected layers, kernel size, number of output filters, activation function, weight initialisation function, dropout rate, number of epochs, batch size, and optimisation algorithm.

The parameters tested include:

- Number of fully connected layer: 2,3,4,5,6
- Kernel size: 3,5,7
- Number of output filters: 16, 30, 32, 64, 128
- Activation function: ReLU, ELU, sigmoid, softmax
- Weight initialisation: HeNormal, RandomNormal
- Dropout rate: 0.1, 0.2, 0.3, 0.4, 0.5
- Number of epochs: 25, 30, 35, 50, 60, 80, 100, 200, 250
- Batch size: 1, 4, 8, 16, 32, 64, 128
- Optimisation algorithm: SGD, ADAM

Chapter 4

Results and conclusions

The result of the hyperparameter search was 4 fully connected layers, kernel size of 3, 30 and 32 output filters, ReLU activation function, HeNormal initialisation, 0.4 dropout rate, 30 epochs, 128 mini-batch size and ADAM optimisation. The loss function used was categorical cross-entropy.

The test data was used to evaluate the effectiveness of the hyperparameters, with resampling done when necessary. For each set of hyperparameters, the training and evaluation were done multiple times to gain a consistent score.

The final result for the selected hyperparameters was 100% training accuracy, 94% validation accuracy, and 93% test accuracy. Although there is some overfitting, the outcome is good considering accuracy wasn't the only focus of the research. The test accuracy is comparable with many other research papers in the industry, although it is not of the level of the highest quality papers, which achieve accuracies close to 100%. These are often obtained via more complex methods and techniques as well as huge abundances of data. The confusion matrix for both raw data and percentage data are shown below.

The project aimed to create a convolutional neural network that is capable of accurately classifying brain tumours – namely gliomas – from an MRI scan, such that it is generalisable to unseen data. A focus on sustainability and reducing the computational complexity and cost was intended, via both established and novel techniques.

The novel techniques that were attempted: region of interest enhancement and dimensionality reduction seem to have worked effectively enough to obtain a relatively successful outcome. In addition to this, the augmentation techniques seem to have worked effectively enough to allow the model to be complex enough to learn some patterns, whilst also largely avoiding the problem of overfitting.

The project as a whole, split into the research, preprocessing, and training stages went fairly well. During the research stage, a good understanding of the literature was established, which allowed for the effective planning of the proceeding steps. In the preprocessing, the input data was adjusted, with unnecessary input removed, and additional samples created. Finally, the training and hyperparameter tuning was also successful, obtaining parameters that resulted in a relatively successful model.

Another successful part of the project was the GUI that was created to demonstrate and illustrate the pipeline of the model. The GUI was created using Python and allows a user to step through each part of the preprocessing pipeline, to see the effect of the steps being taken.

There were also many limitations to the project, however. The use of only T2-FLAIR images and a lack of comparison of the possible differences and benefits in using other modalities or combinations of modalities were not explored. Furthermore, a lack of a substantial amount of data largely limited the scope of the project.

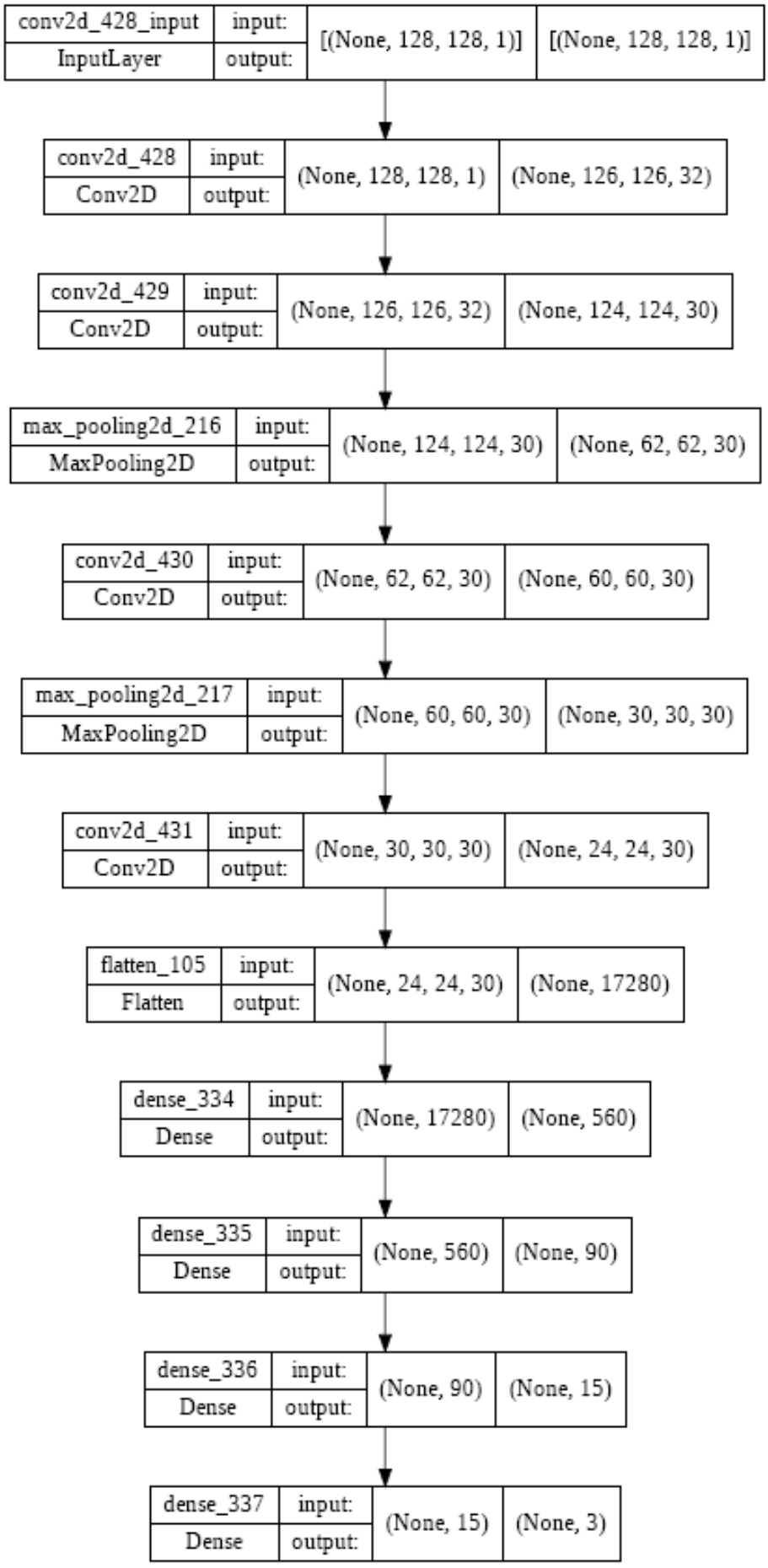

FIGURE 4.1: Final model architecture

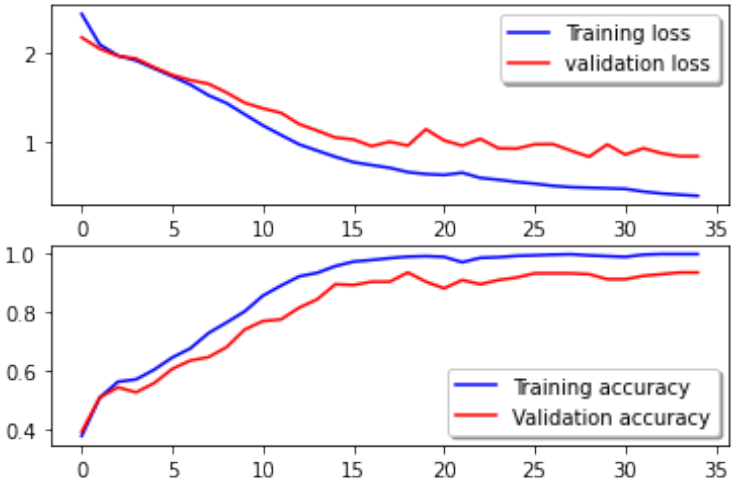

FIGURE 4.2: The training and validation accuracies.

The lack of grade 1 tumours as well as the lack of other types of brain tumours highly limited the scope of the research carried out.

More research could also have been done to investigate the effects of different types of augmentation, as well as different types of dimensionality reduction, 3D architecture, and other hyperparameters rather than just discussing them theoretically. By actually carrying out PCA, or by actually implementing a 3D architecture, a tangible result could have been achieved and presented as to the difference in the result of the implementation.

This can also then be extended to the investigation of sustainability. Much of the research was discussed theoretically, but not actually measured due to the time constraints. By implementing some techniques to measure the probable cost of computation or another similar metric, the choices made could have been discussed in more detail or more concretely and justified.

At present, the discussions such as the cost-benefit of the novel method of dimensionality reduction over PCA, or the choice of optimisation algorithm are bound to the theory, but if some implementation of measure of the computation carried out were implemented a more fruitful discussion could have followed.

Also, despite being focussed on sustainability, the number of hyperparameters being tested meant that an extremely large number of training runs were being carried out. Although the research project was on finding ways to make networks more efficient, and not ensuring the entire process used is efficient itself, this could have been lessened. However, this will always be an issue in the early stages of moving towards more sustainable AI, since extensive testing is required to obtain techniques that are more efficient before these efficient techniques can actually be applied. Nevertheless, some of this impact was aimed to be reduced via the use of Bayesian Optimisation for the hyperparameter selection process through KerasTuner.

4.1 Future work

Following from these limitations, many things can be considered for inclusion in future work. A more detailed consideration of different MRI sequences and modalities could be employed to investigate the effectiveness of each one or the combinations

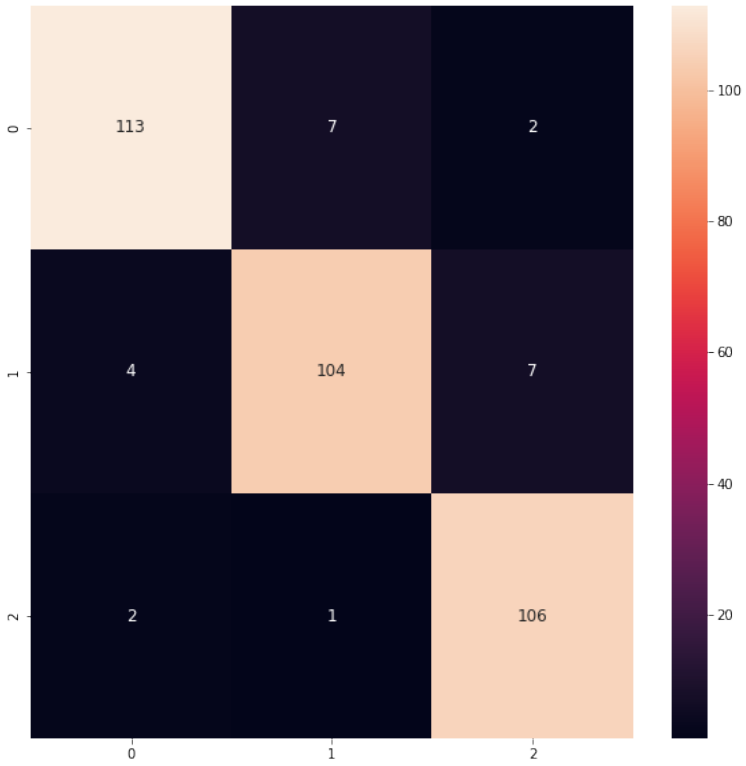

FIGURE 4.3: Raw values confusion matrix.

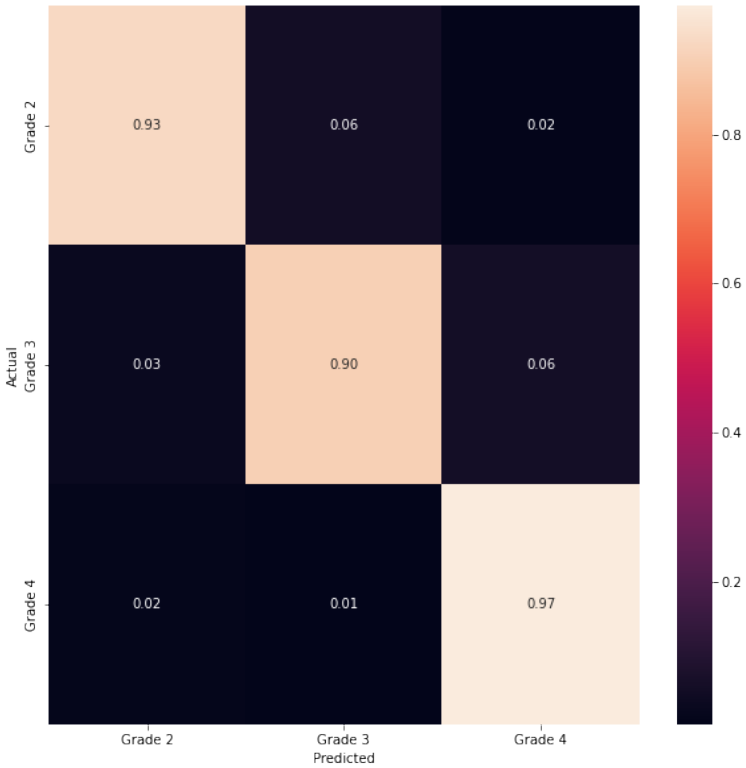

FIGURE 4.4: Confusion matrix as percentages.

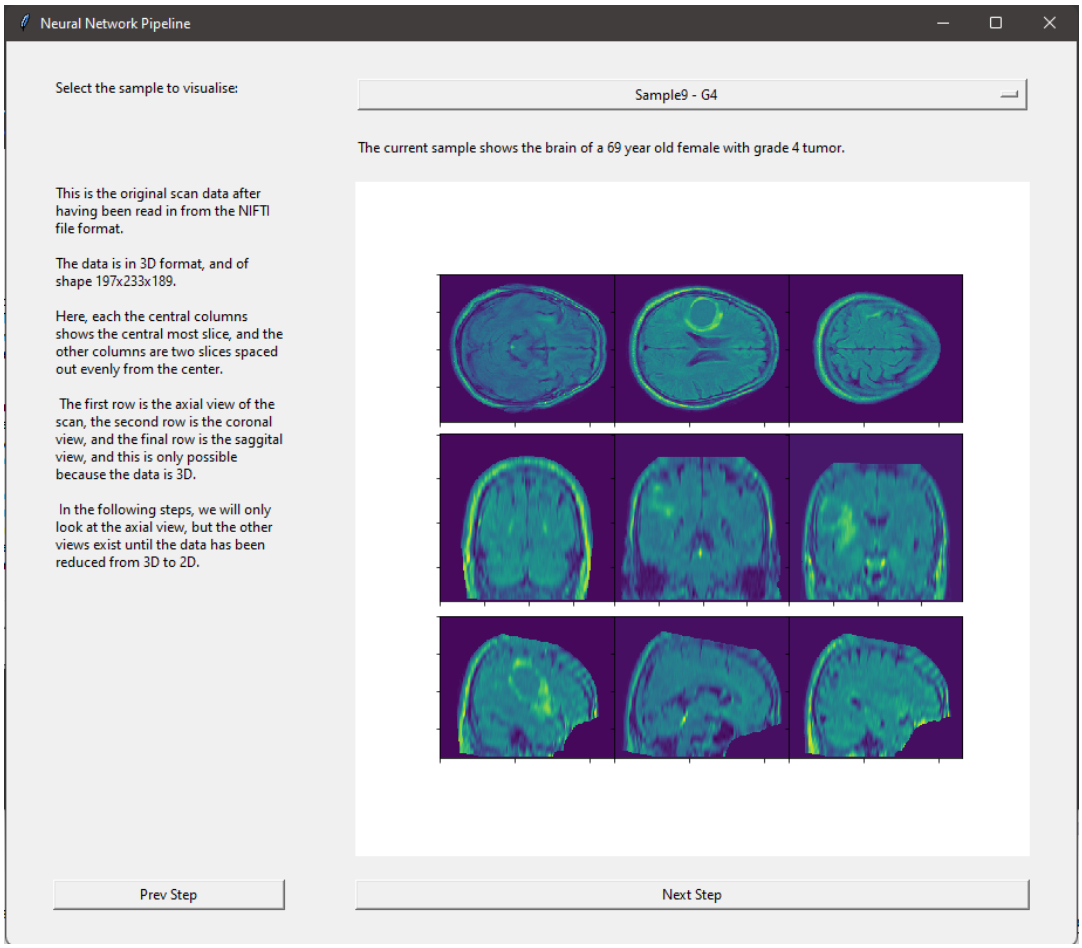

FIGURE 4.5: Screenshot of one of the slides of the GUI.

of these modalities. Different types of brain tumours could also be investigated, rather than just limiting the research to glial tumours, or perhaps even a comparison of how effectively the model learns for different types of tumours.

More work could be planned to actually investigate the choices made, such as the choice of enhancement of the region of interest, or a comparison of how effectively and efficiently the model learns with and without the enhancement. Currently, the novel techniques attempted were not tested rigorously and were only used based on understanding.

On another note, simpler things such as cross-validation could have been implemented, to investigate the effect of possible reduction of overfitting, and the possible effects on efficiency. 3-dimensional neural networks should also be investigated closely in comparison to 2D networks to understand better the tradeoff between accuracy and efficiency.

Similarly, more tangible methods should be employed to study the sustainability of the techniques being used. A lot of research has already begun on this. 'Active Deep Reuse' is a tool developed by researchers at North Carolina State University, which claims to be able to reduce training times for larger networks (Ning, Guan, and Shen, 2019).

An open-source tool developed at Stanford University has been developed named the 'experiment-impact-tracker', aimed to track energy usage, carbon emissions, and compute utilisation of the system (Henderson, 2022; Henderson et al., 2020). The

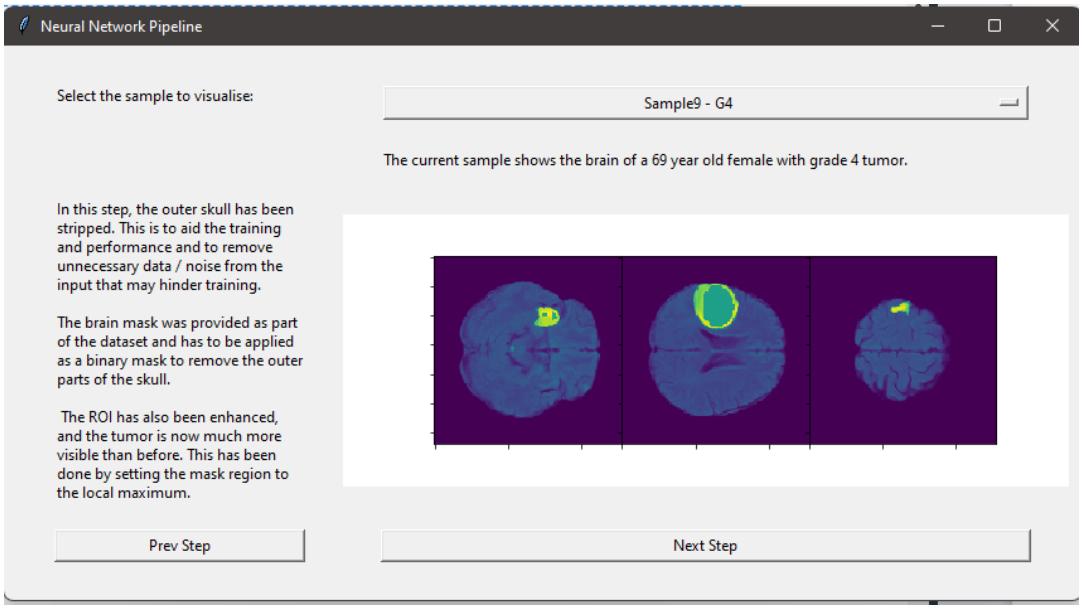

FIGURE 4.6: Screenshot of one of the slides of the GUI).

utilisation of this tool in future work would assist in accurately determining how efficient the choices being made are.

Other techniques have also been developed to further improve efficiency, such as an activation function named 'Full-ReLU'. It claims to effectively minimise the number of kernels required in a convolutional layer without degrading processing quality, another example of techniques that could be investigated (Sun and Wang, 2020). Investigation could have been done into modifying the input size of the image and trying to determine if it could be minimised any further whilst still keeping performance benefits.

The effects of pruning a neural network can also be investigated, which aims to identify nodes and weights that don't contribute a significant amount to the network output. The aim is for the model to produce the same output in a pruned network given a particular input as a non-pruned network. Some more novel theories have emerged in recent years too. The 'Neural Network Lottery Hypothesis' builds on the concept of pruning, claiming that it is possible to prune a network randomly before training, obtaining a 'lottery' subnetwork that could theoretically achieve an accuracy equal to or better than a non-pruned network (Frankle and Carbin, 2018).

Although this theory seems as though it is unlikely to be useful in practical applications, the concept and research that is being carried out suggest that more innovation is being done in the field, and further, more extensive and detailed research is possible in future works.

Bibliography

American Association of Neurological Surgeons (2022). *Brain Tumors*. URL: https://www.aans.org/en/Patients/Neurosurgical-Conditions-and-Treatments/Brain-Tumors (visited on 06/10/2022).

Bahadure, Nilesh Bhaskarrao, Arun Kumar Ray, and Har Pal Thethi (2017). "Image Analysis for MRI Based Brain Tumor Detection and Feature Extraction Using Biologically Inspired BWT and SVM". en. In: *International Journal of Biomedical Imaging* 2017, pp. 1–12. ISSN: 1687-4188, 1687-4196. DOI: 10.1155/2017/9749108. URL: https://www.hindawi.com/journals/ijbi/2017/9749108/ (visited on 06/10/2022).

Baheti, Pragati (May 2022). *What is Overfitting in Deep Learning and How to Avoid It*. URL: https://www.v7labs.com/blog/overfitting (visited on 06/10/2022).

Banerjee, Alekhyo (Aug. 2020). *Computational Complexity of PCA*. URL: https://alekhyo.medium.com/computational-complexity-of-pca-4cb61143b7e5 (visited on 06/10/2022).

Barrera, Francisco (Feb. 2021). *Random Initialization of Weights in a Neural Network*. URL: https://www.baeldung.com/cs/ml-neural-network-weights (visited on 06/10/2022).

Basodi, Sunitha et al. (Sept. 2020). "Gradient amplification: An efficient way to train deep neural networks". In: *Big Data Mining and Analytics* 3.3, pp. 196–207. ISSN: 2096-0654. DOI: 10.26599/BDMA.2020.9020004. URL: https://ieeexplore.ieee.org/document/9142152/ (visited on 06/10/2022).

Benson, Noah (2022). *MRI Data Representation and Geometry*. URL: https://nben.net/MRI-Geometry/ (visited on 06/10/2022).

Berger, A. (Jan. 2002). "How does it work?: Magnetic resonance imaging". en. In: *BMJ* 324.7328, pp. 35–35. ISSN: 0959-8138, 1468-5833. DOI: 10.1136/bmj.324.7328.35. URL: https://www.bmj.com/lookup/doi/10.1136/bmj.324.7328.35 (visited on 06/10/2022).

Bhargava, Rajat (June 2019). "CT Imaging in Neurocritical Care". en. In: *Indian Journal of Critical Care Medicine* 23.S2, pp. 98–103. ISSN: 0972-5229, 1998-359X. DOI: 10.5005/jp-journals-10071-23185. URL: https://www.ijccm.org/doi/10.5005/jp-journals-10071-23185 (visited on 06/10/2022).

Bhattacharjee, Joydeep (Aug. 2017). *Dimensional Reduction and Principal Component Analysis — I*. URL: https://medium.com/technology-nineleaps/dimensional-reduction-and-principal-component-analysis-i-8ce60a5ed2c2 (visited on 06/10/2022).

Bottou, Léon (2012). "Stochastic Gradient Descent Tricks". en. In: *Neural Networks: Tricks of the Trade*. Ed. by Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller. Vol. 7700. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 421–436. ISBN: 978-3-642-35288-1 978-3-642-35289-8. DOI: 10.1007/978-3-642-35289-8_25. URL: http://link.springer.com/10.1007/978-3-642-35289-8_25 (visited on 06/10/2022).

Brownlee, Jason (Oct. 2019). *A Gentle Introduction to Cross-Entropy for Machine Learning*. URL: https://machinelearningmastery.com/cross-entropy-for-machine-learning/ (visited on 06/10/2022).

Burton, Will (Apr. 2019). *2D or 3D? A Simple Comparison of Convolutional Neural Networks for Automatic Segmentation of Cardiac Imaging*. URL: https://towardsdatascience.com/2d-or-3d-a-simple-comparison-of-convolutional-neural-networks-for-automatic-segmentation-of-625308f52aa7 (visited on 06/10/2022).

Cancer Research UK (2022a). *Brain tumour: Gliomas*. URL: https://www.cancerresearchuk.org/about-cancer/brain-tumours/types/glioma-adults (visited on 06/10/2022).

— (2022b). *Brain tumour: Grades*. URL: https://www.cancerresearchuk.org/about-cancer/brain-tumours/grades (visited on 06/10/2022).

— (2022c). *Survival*. URL: https://www.cancerresearchuk.org/about-cancer/brain-tumours/survival (visited on 06/10/2022).

Cancer.net (Sept. 2021). *Brain tumor: diagnosis*. URL: https://www.cancer.net/cancer-types/brain-tumor/diagnosis (visited on 06/10/2022).

Cancer.org (2022). *How diagnosed*. URL: https://www.cancer.org/cancer/brain-spinal-cord-tumors-adults/detection-diagnosis-staging/how-diagnosed.html (visited on 06/10/2022).

Case Western Reserve University (2022). *MRI Basics*. URL: https://case.edu/med/neurology/NR/MRIBasics.htm (visited on 06/10/2022).

Cheng, Jun et al. (Dec. 2015). "Correction: Enhanced Performance of Brain Tumor Classification via Tumor Region Augmentation and Partition". en. In: *PLOS ONE* 10.12, e0144479. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0144479. URL: https://dx.plos.org/10.1371/journal.pone.0144479 (visited on 06/10/2022).

Choudhury, Ambika (May 2019). *Curse Of Dimensionality And What Beginners Should Do To Overcome It*. URL: https://analyticsindiamag.com/curse-of-dimensionality-and-what-beginners-should-do-to-overcome-it/ (visited on 06/10/2022).

Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter (2015). "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)". In: Publisher: arXiv Version Number: 5. DOI: 10.48550/ARXIV.1511.07289. URL: https://arxiv.org/abs/1511.07289 (visited on 06/10/2022).

Cox, Bob (Mar. 2019). *NiFTI file format*. URL: https://nifti.nimh.nih.gov/pub/dist/src/niftilib/nifti1.h (visited on 06/10/2022).

Dahl, George E., Tara N. Sainath, and Geoffrey E. Hinton (May 2013). "Improving deep neural networks for LVCSR using rectified linear units and dropout". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. Vancouver, BC, Canada: IEEE, pp. 8609–8613. ISBN: 978-1-4799-0356-6. DOI: 10.1109/ICASSP.2013.6639346. URL: http://ieeexplore.ieee.org/document/6639346/ (visited on 06/10/2022).

Deshpande, Adit (2022). *A Beginner's Guide To Understanding Convolutional Neural Networks*. URL: https://adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/ (visited on 06/10/2022).

Doroudi, Shayan (July 2020). "The Bias-Variance Tradeoff: How Data Science Can Inform Educational Debates". en. In: *AERA Open* 6.4, p. 233285842097720. ISSN: 2332-8584, 2332-8584. DOI: 10.1177/2332858420977208. URL: http://journals.sagepub.com/doi/10.1177/2332858420977208 (visited on 06/10/2022).

Elite Data Science (2022). *Overfitting in Machine Learning: What It Is and How to Prevent It*. URL: https://elitedatascience.com/overfitting-in-machine-learning (visited on 06/10/2022).

Ephrath, Jonathan et al. (May 2020). "LeanConvNets: Low-Cost Yet Effective Convolutional Neural Networks". In: *IEEE Journal of Selected Topics in Signal Processing* 14.4, pp. 894–904. ISSN: 1932-4553, 1941-0484. DOI: 10.1109/JSTSP.2020.2972775. URL: https://ieeexplore.ieee.org/document/8989808/ (visited on 06/10/2022).

Ertosun, M. Gunhan and D. Rubin (2015). "Automated Grading of Gliomas using Deep Learning in Digital Pathology Images: A modular approach with ensemble of convolutional neural networks". In: *AMIA ... Annual Symposium proceedings. AMIA Symposium* 2015, pp. 1899–908.

Frankle, Jonathan and Michael Carbin (2018). "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks". In: Publisher: arXiv Version Number: 5. DOI: 10.48550/ARXIV.1803.03635. URL: https://arxiv.org/abs/1803.03635 (visited on 06/10/2022).

Gavali, Pralhad and J. Saira Banu (2019). "Deep Convolutional Neural Network for Image Classification on CUDA Platform". en. In: *Deep Learning and Parallel Computing Environment for Bioengineering Systems*. Elsevier, pp. 99–122. ISBN: 978-0-12-816718-2. DOI: 10.1016/B978-0-12-816718-2.00013-0. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780128167182000130 (visited on 06/10/2022).

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). "Deep Sparse Rectifier Neural Networks." In: *AISTATS*. Ed. by Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík. Vol. 15. JMLR Proceedings. JMLR.org, pp. 315–323. URL: http://dblp.uni-trier.de/db/journals/jmlr/jmlrp15.html#GlorotBB11.

Google (2022). *Google Colab: FAQ*. URL: https://research.google.com/colaboratory/faq.html (visited on 06/10/2022).

Grover, Vijay P.B. et al. (Sept. 2015). "Magnetic Resonance Imaging: Principles and Techniques: Lessons for Clinicians". en. In: *Journal of Clinical and Experimental Hepatology* 5.3, pp. 246–255. ISSN: 09736883. DOI: 10.1016/j.jceh.2015.08.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S0973688315004156 (visited on 06/10/2022).

Han, Jiawei and Micheline Kamber (2012). *Data Mining*. en. Elsevier. ISBN: 978-0-12-381479-1. DOI: 10.1016/C2009-0-61819-5. URL: https://linkinghub.elsevier.com/retrieve/pii/C20090618195 (visited on 06/10/2022).

He, Kaiming et al. (Feb. 2015). *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*. Number: arXiv:1502.01852 arXiv:1502.01852 [cs]. URL: http://arxiv.org/abs/1502.01852 (visited on 06/10/2022).

Henderson, Peter (2022). *experiment-impact-tracker*. URL: https://github.com/Breakend/experiment-impact-tracker (visited on 06/10/2022).

Henderson, Peter et al. (2020). "Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning". In: Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2002.05651. URL: https://arxiv.org/abs/2002.05651 (visited on 06/10/2022).

Hong, Won-Kee (2020). "Artificial-intelligence-based design of the ductile precast concrete beams". en. In: *Hybrid Composite Precast Systems*. Elsevier, pp. 427–478. ISBN: 978-0-08-102721-9. DOI: 10.1016/B978-0-08-102721-9.00010-8. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780081027219000108 (visited on 06/10/2022).

Hoopes, Andrew et al. (2022). "SynthStrip: Skull-Stripping for Any Brain Image". In: Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2203.09974. URL: https://arxiv.org/abs/2203.09974 (visited on 06/10/2022).

IBM Cloud Education (Oct. 2020). URL: https://www.ibm.com/cloud/learn/gradient-descent (visited on 06/10/2022).

Işın, Ali, Cem Direkoğlu, and Melike Şah (2016). "Review of MRI-based Brain Tumor Image Segmentation Using Deep Learning Methods". en. In: *Procedia Computer Science* 102, pp. 317–324. ISSN: 18770509. DOI: 10.1016/j.procs.2016.09.407. URL: https://linkinghub.elsevier.com/retrieve/pii/S187705091632587X (visited on 06/10/2022).

Jaadi, Zakaria (Apr. 2021). *A Step-by-Step Explanation of Principal Component Analysis (PCA)*. URL: https://builtin.com/data-science/step-step-explanation-principal-component-analysis (visited on 06/10/2022).

Jabbar, Haider Khalaf and Rafiqul Zaman Khan (2014). "Methods to Avoid Over-Fitting and Under-Fitting in Supervised Machine Learning (Comparative Study)". en. In: *Computer Science, Communication and Instrumentation Devices*. Research Publishing Services, pp. 163–172. ISBN: 978-981-09-5247-1. DOI: 10.3850/978-981-09-5247-1_017. URL: http://rpsonline.com.sg/proceedings/9789810952471/html/017.xml (visited on 06/10/2022).

Janocha, Katarzyna and Wojciech Marian Czarnecki (2017). "On Loss Functions for Deep Neural Networks in Classification". In: Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1702.05659. URL: https://arxiv.org/abs/1702.05659 (visited on 06/10/2022).

Jaroudi, Rym (Oct. 2017). "Inverse Mathematical Models for Brain Tumour Growth". ISBN: 9789176854402. Ph.D. Linköping, Sweden: Linköping University. DOI: 10.3384/lic.diva-141982. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-141982 (visited on 06/10/2022).

Jeong, Jiwon (Jan. 2019). *The Most Intuitive and Easiest Guide for Convolutional Neural Network*. URL: https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480 (visited on 06/10/2022).

John Hopkins Medicine (2022). *Gliomas*. URL: https://www.hopkinsmedicine.org/health/conditions-and-diseases/gliomas (visited on 06/10/2022).

Kabir Anaraki, Amin, Moosa Ayati, and Foad Kazemi (Jan. 2019). "Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms". en. In: *Biocybernetics and Biomedical Engineering* 39.1, pp. 63–74. ISSN: 02085216. DOI: 10.1016/j.bbe.2018.10.004. URL: https://linkinghub.elsevier.com/retrieve/pii/S0208521618300676 (visited on 06/10/2022).

Kakaraparthi, Vishnu (Sept. 2018). *Xavier and He Normal (He-et-al) Initialization*. URL: https://prateekvishnu.medium.com/xavier-and-he-normal-he-et-al-initialization-8e3d7a087528 (visited on 06/10/2022).

Kalavathi, P. and V. B. Surya Prasath (June 2016). "Methods on Skull Stripping of MRI Head Scan Images—a Review". en. In: *Journal of Digital Imaging* 29.3, pp. 365–379. ISSN: 0897-1889, 1618-727X. DOI: 10.1007/s10278-015-9847-8. URL: http://link.springer.com/10.1007/s10278-015-9847-8 (visited on 06/10/2022).

Kingma, Diederik P. and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization". In: Publisher: arXiv Version Number: 9. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980 (visited on 06/10/2022).

Lazar, Andreea (2009). "SORN: a Self-organizing Recurrent Neural Network". In: *Frontiers in Computational Neuroscience* 3. ISSN: 16625188. DOI: 10.3389/neuro.10.023.2009. URL: http://journal.frontiersin.org/article/10.3389/neuro.10.023.2009/abstract (visited on 06/10/2022).

Lemaitre, Guillaume, Fernando Nogueira, and Christos K. Aridas (2016). "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning". In: Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1609.06570. URL: https://arxiv.org/abs/1609.06570 (visited on 06/10/2022).

Litjens, Geert et al. (Dec. 2017). "A survey on deep learning in medical image analysis". en. In: *Medical Image Analysis* 42, pp. 60–88. ISSN: 13618415. DOI: 10.1016/j.media.2017.07.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S1361841517301135 (visited on 06/10/2022).

Louis, David N. et al. (June 2016). "The 2016 World Health Organization Classification of Tumors of the Central Nervous System: a summary". en. In: *Acta Neuropathologica* 131.6, pp. 803–820. ISSN: 0001-6322, 1432-0533. DOI: 10.1007/s00401-016-1545-1. URL: http://link.springer.com/10.1007/s00401-016-1545-1 (visited on 06/10/2022).

Mackillop, William J. (July 2006). "The Importance of Prognosis in Cancer Medicine". en. In: *TNM Online*. Ed. by Leslie H. Sobin. 1st ed. Wiley. ISBN: 978-0-471-42019-4. DOI: 10.1002/0471463736.tnmp01.pub2. URL: https://onlinelibrary.wiley.com/doi/10.1002/0471463736.tnmp01.pub2 (visited on 06/10/2022).

Magadza, Tirivangani and Serestina Viriri (Jan. 2021). "Deep Learning for Brain Tumor Segmentation: A Survey of State-of-the-Art". en. In: *Journal of Imaging* 7.2, p. 19. ISSN: 2313-433X. DOI: 10.3390/jimaging7020019. URL: https://www.mdpi.com/2313-433X/7/2/19 (visited on 06/10/2022).

Mahipal, Sukriti (Dec. 2021). *AI's Carbon Footprint*. URL: https://storymaps.arcgis.com/stories/55219556f9f545dd9b06333350f0b339 (visited on 06/10/2022).

Mandal, Manav (May 2021). *Introduction to Convolutional Neural Networks (CNN)*. URL: https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/ (visited on 06/10/2022).

Mayfield Clinic (2022). *Glioma brain tumors*. URL: https://www.mayfieldclinic.com/pe-glioma.htm (visited on 06/10/2022).

Menze, Bjoern H. et al. (Oct. 2015). "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)". In: *IEEE Transactions on Medical Imaging* 34.10, pp. 1993–2024. ISSN: 0278-0062, 1558-254X. DOI: 10.1109/TMI.2014.2377694. URL: http://ieeexplore.ieee.org/document/6975210/ (visited on 06/10/2022).

MIT News (Aug. 2020). *Shrinking deep learning's carbon footprint*. URL: https://news.mit.edu/2020/shrinking-deep-learning-carbon-footprint-0807 (visited on 06/10/2022).

ML Glossary (2022). *Loss function*. URL: https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html#cross-entropy (visited on 06/10/2022).

Murat, Mustafa (Feb. 2019). *Weight Initialization Schemes - Xavier (Glorot) and He*. URL: https://mmuratarat.github.io/2019-02-25/xavier-glorot-he-weight-init (visited on 06/10/2022).

Nair, Vinod and Geoffrey E. Hinton (2010). "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. Ed. by Johannes Fürnkranz and Thorsten Joachims, pp. 807–814.

National Cancer Intelligence Network (Feb. 2018). *Chemotherapy, radiotherapy and tumour resection in England, 2013-14*. URL: http://www.ncin.org.uk/view?rid=3459 (visited on 06/10/2022).

National Institute of Neurological Disorders and Stroke (2022). *Brain Basics: The Life and Death of a Neuron*. URL: https://www.ninds.nih.gov/health-information/patient-caregiver-education/brain-basics-life-and-death-neuron (visited on 06/10/2022).

Neural networks and deep learning (2022). *Chapter 6: Introducing convolutional network*. URL: http://neuralnetworksanddeeplearning.com/chap6.html#introducing_convolutional_networks (visited on 06/10/2022).

Neuroimaging Informatics Technology Initiative (Dec. 2013). *Neuroimaging Informatics Technology Initiative*. URL: https://nifti.nimh.nih.gov/ (visited on 06/10/2022).

Nguyen, Lan Huong and Susan Holmes (June 2019). "Ten quick tips for effective dimensionality reduction". en. In: *PLOS Computational Biology* 15.6. Ed. by Francis Ouellette, e1006907. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1006907. URL: https://dx.plos.org/10.1371/journal.pcbi.1006907 (visited on 06/10/2022).

NiBabel (2022). *NiBabel*. URL: https://nipy.org/nibabel/ (visited on 06/10/2022).

Nichols, James A., Hsien W. Herbert Chan, and Matthew A. B. Baker (Feb. 2019). "Machine learning: applications of artificial intelligence to imaging and diagnosis". en. In: *Biophysical Reviews* 11.1, pp. 111–118. ISSN: 1867-2450, 1867-2469. DOI: 10.1007/s12551-018-0449-9. URL: http://link.springer.com/10.1007/s12551-018-0449-9 (visited on 06/10/2022).

Ning, Lin, Hui Guan, and Xipeng Shen (Apr. 2019). "Adaptive Deep Reuse: Accelerating CNN Training on the Fly". In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. Macao, Macao: IEEE, pp. 1538–1549. ISBN: 978-1-5386-7474-1. DOI: 10.1109/ICDE.2019.00138. URL: https://ieeexplore.ieee.org/document/8731452/ (visited on 06/10/2022).

NumPy (2022). *NumPy*. URL: https://numpy.org/ (visited on 06/10/2022).

Nwankpa, Chigozie et al. (Nov. 2018). *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*. Number: arXiv:1811.03378 arXiv:1811.03378 [cs]. URL: http://arxiv.org/abs/1811.03378 (visited on 06/10/2022).

Office for National Statistics (Jan. 2019). *Cancer survival in England: national estimates for patients followed up to 2017*. URL: https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/conditionsanddiseases/bulletins/cancersurvivalinengland/nationalestimatesforpatientsfollowedupto2017 (visited on 06/10/2022).

OpenAI (Nov. 2019). *AI and Compute*. URL: https://openai.com/blog/ai-and-compute/ (visited on 06/10/2022).

Ozyildirim, Buse Melis and Mariam Kiran (Nov. 2021). "Levenberg–Marquardt multi-classification using hinge loss function". en. In: *Neural Networks* 143, pp. 564–571. ISSN: 08936080. DOI: 10.1016/j.neunet.2021.07.010. URL: https://linkinghub.elsevier.com/retrieve/pii/S0893608021002732 (visited on 06/10/2022).

Park, Jin Seo et al. (2010). "A Proposal of New Reference System for the Standard Axial, Sagittal, Coronal Planes of Brain Based on the Serially-Sectioned Images". en. In: *Journal of Korean Medical Science* 25.1, p. 135. ISSN: 1011-8934, 1598-6357. DOI: 10.3346/jkms.2010.25.1.135. URL: https://jkms.org/DOIx.php?id=10.3346/jkms.2010.25.1.135 (visited on 06/10/2022).

Patel, Aisha (Sept. 2020). "Benign vs Malignant Tumors". en. In: *JAMA Oncology* 6.9, p. 1488. ISSN: 2374-2437. DOI: 10.1001/jamaoncol.2020.2592. URL: https://jamanetwork.com/journals/jamaoncology/fullarticle/2768634 (visited on 06/10/2022).

Paul, Ananya and Tejpratap Gvsl (Nov. 2018). "A Computationally Efficient Neural Network For Faster Image Classification". In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. Bangalore, India: IEEE, pp. 154–159. ISBN: 978-1-5386-9276-9. DOI: 10.1109/SSCI.2018.8628751. URL: https://ieeexplore.ieee.org/document/8628751/ (visited on 06/10/2022).

Paul, Justin S. et al. (Mar. 2017). "Deep learning for brain tumor classification". In: ed. by Andrzej Krol and Barjor Gimi. Orlando, Florida, United States, p. 1013710. DOI: 10.1117/12.2254195. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2254195 (visited on 06/10/2022).

Pedano, Nancy et al. (2016). *Radiology Data from The Cancer Genome Atlas Low Grade Glioma [TCGA-LGG] collection*. Version Number: 3 Type: dataset. DOI: 10.7937/K9/TCIA.2016.L4LTD3TK. URL: https://wiki.cancerimagingarchive.net/x/BANR (visited on 06/10/2022).

Penfold, Clarissa et al. (June 2017). "Diagnosing adult primary brain tumours: can we do better?" en. In: *British Journal of General Practice* 67.659, pp. 278–279. ISSN: 0960-1643, 1478-5242. DOI: 10.3399/bjgp17X691277. URL: https://bjgp.org/lookup/doi/10.3399/bjgp17X691277 (visited on 06/10/2022).

Peng, Yaohao and Mateus Hiro Nagata (Oct. 2020). "An empirical overview of nonlinearity and overfitting in machine learning using COVID-19 data". en. In: *Chaos, Solitons & Fractals* 139, p. 110055. ISSN: 09600779. DOI: 10.1016/j.chaos.2020.110055. URL: https://linkinghub.elsevier.com/retrieve/pii/S0960077920304525 (visited on 06/10/2022).

Purves, Dale et al., eds. (2004). *Neuroscience*. 3rd ed. Sunderland, Mass: Sinauer Associates, Publishers. ISBN: 978-0-87893-725-7.

Qin, Na et al. (May 2021). "LeanNet: An Efficient Convolutional Neural Network for Digital Number Recognition in Industrial Products". en. In: *Sensors* 21.11, p. 3620. ISSN: 1424-8220. DOI: 10.3390/s21113620. URL: https://www.mdpi.com/1424-8220/21/11/3620 (visited on 06/10/2022).

Rukundo, Olivier (May 2022). *Effects of Image Size on Deep Learning*. Number: arXiv:2101.11508 arXiv:2101.11508 [cs, eess]. URL: http://arxiv.org/abs/2101.11508 (visited on 06/10/2022).

Safdar, Muhammad, Shayma Kobaisi, and Fatima Zahra (2020). "A Comparative Analysis of Data Augmentation Approaches for Magnetic Resonance Imaging (MRI) Scan Images of Brain Tumor". In: *Acta Informatica Medica* 28.1, p. 29. ISSN: 0353-8109. DOI: 10.5455/aim.2020.28.29-36. URL: https://www.ejmanager.com/fulltextpdf.php?mno=88216 (visited on 06/10/2022).

Sajid, Sidra, Saddam Hussain, and Amna Sarwar (Nov. 2019). "Brain Tumor Detection and Segmentation in MR Images Using Deep Learning". en. In: *Arabian Journal for Science and Engineering* 44.11, pp. 9249–9261. ISSN: 2193-567X, 2191-4281. DOI: 10.1007/s13369-019-03967-8. URL: http://link.springer.com/10.1007/s13369-019-03967-8 (visited on 06/10/2022).

Sajjad, Muhammad et al. (Jan. 2019). "Multi-grade brain tumor classification using deep CNN with extensive data augmentation". en. In: *Journal of Computational Science* 30, pp. 174–182. ISSN: 18777503. DOI: 10.1016/j.jocs.2018.12.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877750318307385 (visited on 06/10/2022).

Santosh, KC, Nibaran Das, and Swarnendu Ghosh (2022). *Deep Learning Models for Medical Imaging*. en. Elsevier. ISBN: 978-0-12-823504-1. DOI: 10.1016/C2020-0-00344-0. URL: https://linkinghub.elsevier.com/retrieve/pii/C20200003440 (visited on 06/10/2022).

Sathya, R. and Annamma Abraham (2013). "Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification". en. In: *International Journal of Advanced Research in Artificial Intelligence* 2.2. ISSN: 21654069, 21654050. DOI: 10.14569/IJARAI.2013.020206. URL: http://thesai.org/Publications/ViewPaper?Volume=2&Issue=2&Code=IJARAI&SerialNo=6 (visited on 06/10/2022).

Scarpace, Lisa et al. (2019). *Data From REMBRANDT*. Version Number: 2 Type: dataset. DOI: 10.7937/K9/TCIA.2015.588OZUZB. URL: https://wiki.cancerimagingarchive.net/display/Public/REMBRANDT (visited on 06/10/2022).

SciPy (2022). *ndimage*. URL: https://docs.scipy.org/doc/scipy/reference/ndimage.html (visited on 06/10/2022).

Sebastian (Aug. 2021). *Understanding Hinge Loss and the SVM Cost Function*. URL: https://programmathically.com/understanding-hinge-loss-and-the-svm-cost-function/ (visited on 06/10/2022).

Sharma, Sagar (Sept. 2017). *What the Hell is Perceptron?* URL: https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53 (visited on 06/10/2022).

Sharma, Siddharth, Simone Sharma, and Anidhya Athaiya (May 2020). "ACTIVATION FUNCTIONS IN NEURAL NETWORKS". In: *International Journal of Engineering Applied Sciences and Technology* 04.12, pp. 310–316. ISSN: 24552143. DOI: 10.33564/IJEAST.2020.v04i12.054. URL: https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf (visited on 06/10/2022).

Shukla, Ak and Utham Kumar (2006). "Positron emission tomography: An overview". en. In: *Journal of Medical Physics* 31.1, p. 13. ISSN: 0971-6203. DOI: 10.4103/0971-6203.25665. URL: http://www.jmp.org.in/text.asp?2006/31/1/13/25665 (visited on 06/10/2022).

Silver, Nate (2012). *The signal and the noise: why so many predictions fail–but some don't*. New York: Penguin Press. ISBN: 978-1-59420-411-1.

Singh, Pushpa et al. (2021). "Diagnosing of disease using machine learning". en. In: *Machine Learning and the Internet of Medical Things in Healthcare*. Elsevier, pp. 89–111. ISBN: 978-0-12-821229-5. DOI: 10.1016/B978-0-12-821229-5.00003-3. URL: https://linkinghub.elsevier.com/retrieve/pii/B9780128212295000033 (visited on 06/10/2022).

Solawetz, Jacob (Sept. 2020). *Train, Validation, Test Split for Machine Learning*. URL: https://blog.roboflow.com/train-test-split/ (visited on 06/10/2022).

Sorzano, C. O. S., J. Vargas, and A. Pascual Montano (2014). "A survey of dimensionality reduction techniques". In: Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1403.2877. URL: https://arxiv.org/abs/1403.2877 (visited on 06/10/2022).

Srivastava, Nitish et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Strubell, Emma, Ananya Ganesh, and Andrew McCallum (2019). "Energy and Policy Considerations for Deep Learning in NLP". In: Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.1906.02243. URL: https://arxiv.org/abs/1906.02243 (visited on 06/10/2022).

Sultan, Hossam H., Nancy M. Salem, and Walid Al-Atabany (2019). "Multi-Classification of Brain Tumor Images Using Deep Neural Network". In: *IEEE Access* 7, pp. 69215–69225. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2919122. URL: https://ieeexplore.ieee.org/document/8723045/ (visited on 06/10/2022).

Sun, Shiliang et al. (Oct. 2019). *A Survey of Optimization Methods from a Machine Learning Perspective*. Number: arXiv:1906.06821 arXiv:1906.06821 [cs, math, stat]. URL: http://arxiv.org/abs/1906.06821 (visited on 06/10/2022).

Sun, Yanming and Chunyan Wang (2020). "A Computation-Efficient CNN System for High-Quality Brain Tumor Segmentation". In: Publisher: arXiv Version Number: 3. DOI: 10.48550/ARXIV.2007.12066. URL: https://arxiv.org/abs/2007.12066 (visited on 06/10/2022).

Tahmassebi, Amirhessam et al. (Sept. 2018). "Multi-stage optimization of a deep model: A case study on ground motion modeling". en. In: *PLOS ONE* 13.9. Ed. by Ivan Olier, e0203829. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0203829. URL: https://dx.plos.org/10.1371/journal.pone.0203829 (visited on 06/10/2022).

The Brain Tumour Charity (2022). *How brain tumours are graded*. URL: https://www.thebraintumourcharity.org/brain-tumour-diagnosis-treatment/how-brain-tumours-are-diagnosed/how-brain-tumours-are-graded/ (visited on 06/10/2022).

TheNextWeb (2022). *Is there a more environmentally friendly way to train AI?* URL: https://thenextweb.com/news/more-environmentally-friendly-way-to-train-ai/amp (visited on 06/10/2022).

Voort, Sebastian R. van der et al. (Aug. 2021). "The Erasmus Glioma Database (EGD): Structural MRI scans, WHO 2016 subtypes, and segmentations of 774 patients with glioma". en. In: *Data in Brief* 37, p. 107191. ISSN: 23523409. DOI: 10.1016/j.dib.2021.107191. URL: https://linkinghub.elsevier.com/retrieve/pii/S2352340921004753 (visited on 06/10/2022).

Weigel, Anna (2022). *Artificial Neural Networks: Deep Dive*. URL: https://www.modulos.ai/blog/artificial-neural-networks-deep-dive/ (visited on 06/10/2022).

Westbrook, Catherine and John Talbot (2018). *MRI in practice*. Fifth edition. Hoboken, NJ: Wiley. ISBN: 978-1-119-39196-8.

Wilson, D.Randall and Tony R. Martinez (Dec. 2003). "The general inefficiency of batch training for gradient descent learning". en. In: *Neural Networks* 16.10, pp. 1429–1451. ISSN: 08936080. DOI: 10.1016/S0893-6080(03)00138-2. URL: https://linkinghub.elsevier.com/retrieve/pii/S0893608003001382 (visited on 06/10/2022).

Wynsberghe, Aimee van (Aug. 2021). "Sustainable AI: AI for sustainability and the sustainability of AI". en. In: *AI and Ethics* 1.3, pp. 213–218. ISSN: 2730-5953, 2730-5961. DOI: 10.1007/s43681-021-00043-6. URL: https://link.springer.com/10.1007/s43681-021-00043-6 (visited on 06/10/2022).

Yamashita, Rikiya et al. (Aug. 2018). "Convolutional neural networks: an overview and application in radiology". en. In: *Insights into Imaging* 9.4, pp. 611–629. ISSN: 1869-4101. DOI: 10.1007/s13244-018-0639-9. URL: https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9 (visited on 06/10/2022).

Yousra, Dahdouh, Anouar Boudhir Abdelhakim, and Ben Ahmed Mohamed (2021). "Sustainability of Artificial Intelligence and Deep Learning Algorithms for Medical Image Classification: Case of Cancer Pathology". en. In: *Emerging Trends in ICT for Sustainable Development*. Ed. by Mohamed Ben Ahmed et al. Series Title: Advances in Science, Technology & Innovation. Cham: Springer International Publishing, pp. 19–28. ISBN: 978-3-030-53439-4 978-3-030-53440-0. DOI: 10.1007/978-3-030-53440-0_3. URL: http://link.springer.com/10.1007/978-3-030-53440-0_3 (visited on 06/10/2022).

Yu, Tong and Hong Zhu (2020). "Hyper-Parameter Optimization: A Review of Algorithms and Applications". In: Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2003.05689. URL: https://arxiv.org/abs/2003.05689 (visited on 06/10/2022).

Zhang, Aston et al. (2021). "Dive into Deep Learning". In: Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.2106.11342. URL: https://arxiv.org/abs/2106.11342 (visited on 06/10/2022).