

# Breast Cancer Classification using Pretrained Models on Tissue Imaging

Nottingham Trent University  
School of Science and Technology

Project Report submitted in partial fulfilment of the requirements of  
Nottingham Trent University for the degree of MSc Data Science

NATHAN LAWRENCE

## Abstract

Breast cancer is the prevalent cancer in woman, accounting for 15% of the new cancer cases each year. With a continuing rise of diagnosis of breast cancer each year, which is predicted to continue to increase there needs to be a way to be able to diagnose the cancer earlier, while increasing the speed and accuracy of the diagnosis leading to the quicker start of treatment thus potentially dropping the mortality rate. With AI and machine learning being one of the main talking points in the world at the moment, this has prompted ways in which to use CAD (Computer Aided Diagnosis) tools for quicker and more accurate potentially eradicating mis diagnosis of breast cancer. These tools will be able to assist medical professionals into making more accurate and quicker diagnosis which will lead to being able to start treatment timely and ultimately a lower mortality rate. This work presented the use of 5 different pre-trained CNN models, namely VGG-16, InceptionV3, DenseNet121, ResNet50 and EfficientNet B3. These models were all used on the same dataset consisting of 966 H&E stained images of stages 1-3 of IDC breast cancer. The performance differed in all of the models but VGG-16 and EfficientNet B3 preformed the best resulting in 70.82% and 65.44% accuracy respectively. These two models also obtained very good AUG scores as well as performed well according to the confusion matrix. The F1 scores of these models were fair and coincided with the accuracy of the model.

## Table of Contents

<i>Declaration</i> .....	Error! Bookmark not defined.
<i>Abstract</i> .....	1
<i>Acknowledgment</i> .....	Error! Bookmark not defined.
<i>List of Figures</i> .....	3
<i>List of Tables</i> .....	3
<i>Introduction</i> .....	4
<i>Background</i> .....	6
<i>Literature review</i> .....	7
Breast Cancer .....	7
Pre-processing and normalisation .....	9
The Model .....	11
Pre-trained models .....	11
VGG-16 .....	12
Inception V3 .....	13
ResNet50 .....	15
EffcientNet .....	16
DenseNet .....	18
Project planning .....	20
Aims and Objectives .....	20
Project Design .....	21
Project Risks .....	24
PSEL Issues .....	26
Time Plan .....	27
Materials and Methodology .....	29
The Dataset .....	29
Data Pre-Processing .....	29
Convolutional Neural Network (CNN) .....	31
Results .....	35
Discussion .....	41
Conclusion and Future Work .....	43
References .....	44
Appendix .....	48

## List of Figures

FIGURE 1: PIE CHART SHOWING THE ESTIMATED NUMBER OF NEW CASES PER CANCER WORLDWIDE IN 2020.	4
FIGURE 2: IMAGE SHOWING A COMPARISON BETWEEN NORMAL, DCIS AND IDC	5
FIGURE 3: IMAGE SHOWING THE DIFFERENCE BETWEEN IDC AND DCIS	7
FIGURE 4: FIGURE SHOWING A NORMAL BREAST WITH IDC (SEEN IN THE ENLARGED CROSS-SECTION)	8
FIGURE 5: IDC IN BREAST TISSUE WITH 40X MAGNIFICATION STAINED WITH H&E	9
FIGURE 6: GRAPHIC REPRESENTATION SHOWING THE ARCHITECTURE OF A CNN MODEL	11
FIGURE 7: THE FIGURE SHOWS THE LAYERS OF THE VGG-16 MODEL.	12
FIGURE 8: IMAGE SHOWING THE ARCHITECTURE OF THE INCEPTION V3 MODEL	13
FIGURE 9: IMAGE SHOWING THE ARCHITECTURE OF THE RESNET34 MODEL	15
FIGURE 10: IMAGE SHOWING THE WORKINGS OF THE RESIDUAL MAPPING	15
FIGURE 11: TABLE SHOWING THE ARCHITECTURE OF THE RESNET MODEL DEPENDING ON THE SIZE	16
FIGURE 12: IMAGE SHOWING THE BASIC ARCHITECTURE OF EFFICIENTNET B0	16
FIGURE 13: GRAPH COMPARING THE ACCURACY OF THE EFFICIENTNET MODELS TO THE NUMBER OF PARAMETERS IN THE MODEL	17
FIGURE 14: FIGURE SHOWING THE ARCHITECTURE OF THE DENSENET MODEL	18
FIGURE 15: FIGURE SHOWING THE USE OF DENSE BLOCK AND THE LAYERS OF THE MODEL	18
FIGURE 16: THE TIMELINE THAT HAS BEEN CREATED FOR THE PROJECT, INCLUDING THE MILESTONES ABOVE TIMELINE	27
FIGURE 17: HISTOLOGICAL CLASSIFICATION OF BREAST CANCER GRADES 1(A),2(B),3(C) AT MAGNIFICATION OF 40X	29
FIGURE 18: COLLECTION OF IMAGES SHOWING HOW THE SOURCE IMAGE WAS ADJUSTED TO THE SCALES OF THE TARGET IMAGE	30
FIGURE 19: IMAGE SHOWING THE FINAL LAYER OF THE MODEL	32
FIGURE 20: GRAPH REPRESENTING HOW RELU WORKS	32
FIGURE 21: IMAGE SHOWING THE EQUATIONS FOR BATCH NORMALISATION	33
FIGURE 22: SOFTMAX CALCULATION	33
FIGURE 23: EQUATION SHOWING THE LOSS FUNCTION	34
FIGURE 24: EQUATION USED TO DETERMINE THE F1 SCORE OF THE MODEL	34
FIGURE 25: VGG-16 CONFUSION MATRIX AND ROC CURVE	38
FIGURE 26: INCEPTION V3: CONFUSION MATRIX AND ROC CURVE	39
FIGURE 27: DENSENET121: CONFUSION MATRIX AND ROC CURVE	39
FIGURE 28: RESNET50: CONFUSION MATRIX AND ROC CURVE	40
FIGURE 29: EFFECIENTNET B3: CONFUSION MATRIX AND ROC CURVE	40

## List of Tables

TABLE 1: TABLE SHOWING THE RISK THAT COULD BE EXPERIENCED AS WELL AS THE PROBABILITY, SEVERITY AND RISK MANAGEMENT	24
TABLE 2: THE TABLE SHOWS THE START AND END DATE OF THE TASK AS WELL AS THE MAXIMUM DAYS ALLOCATED TO EACH TASK	28
TABLE 3: TABLE REPRESENTING THE MODELS USED, USING THE MODEL API	35
TABLE 4: TABLE REPRESENTING THE MODELS USED, USING THE SEQUENTIAL API	36
TABLE 5: TABLE SHOWING THE BEST PERFORMING MODELS	37
TABLE 6: TABLE SHOWING THE TOTAL NUMBER OF PARAMETERS OF THE MODELS	38

## Introduction

Breast Cancer is one of the most common cancers within women accounting for 15% of all new cancer cases diagnosed each year. Breast Cancer can present at any age but as age increases so does the risk. There are around 56000 new cases of breast cancer diagnosed each year in the UK with around a 76% survival rate. The later that the breast cancer is diagnosed the lower the chance of survival making early diagnosis vital in survival. With breast cancer diagnosis rates continuing to increase, it is predicted between the years 2038 – 2040 that there will be an additional 69900 new cases per year in the UK. Looking at figure 2, it is clear that new cases of breast cancer are coming up at a much higher rate than any other cancer in females thus creating a large concern as to why this is happening and in what ways can diagnosis be improved to catch breast cancer earlier allowing treatment to be successful.

Breast cancer occurs when cells within the breast begin to grow abnormally. This cell will

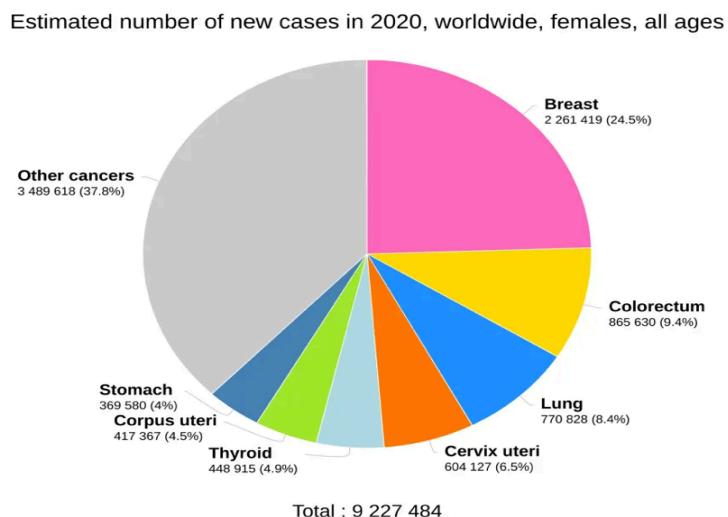
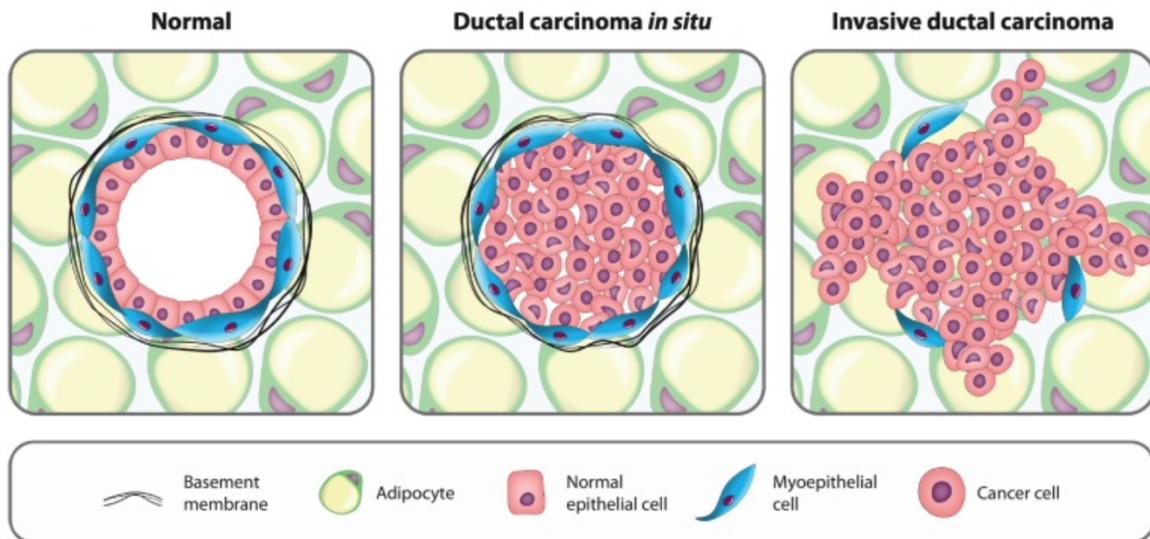


Figure 1: Pie chart showing the estimated number of new cases per cancer worldwide in 2020.

begin to divide quicker than the healthy leading to the creation of a mass, and ultimately the tumour. These cells may metastasise and spread (referred to as invasive ductal carcinoma) to other parts of the body, that is why early diagnosis is of utmost importance allowing for treatment to eradicate the cancer before it begins to spread. In most cases the mass will start in the milk ducts and in rarer cases can begin in the lobules within the glandular tissue. The tumour is classified as either Invasive Ductal Carcinoma (around 80% of all cases) or Ductal Carcinoma in Situ (around 20% of all cases). Invasive Ductal carcinoma (IDC), was mentioned earlier but it is the term used for a tumour that has the potential to be able to spread throughout the body, making treatment harder as well as reducing the chance of survival. Whereas Ductal Carcinoma in Situ (DCIS) is not able to spread throughout the body, making this type of cancer easier to treat than IDC as its location will stay the same, thus giving the person a higher chance of survival. Figure 2 is used to show a comparison between the IDC and DCIS as well as being compared to the structure of a normal breast.



*Figure 2: image showing a comparison between normal, DCIS and IDC*

The main most noticeable symptom seen in breast cancer is a lump or thickened area in the breast tissue. A lump or breast pain does not necessarily mean breast cancer as majority of lumps aren't cancerous but should always be examined by a professional. Many other symptoms can be experienced such as a change in the size or shape of either one or both breasts, a discharge from the nipple that may be streaked with blood, a lump or swelling in either of the armpits, dimpling on the skin on either or both breasts, sometimes a rash can be seen on or around the vicinity of the nipple and lastly a change in the appearance of the nipple.

When looking at the diagnosis of breast cancer there are many ways in which breast cancer is diagnosed. Mammograms, breast ultrasounds, microscopy biopsies and a breast MRI are all gold standard in diagnosis of breast cancer. Majority of these allows the practitioner to determine the location as well as the size of the tumour that is present. During microscopy biopsies, a core of tissue will be extracted from the area of interest. The piece of breast tissue will be stained using haematoxylin and eosin, with haematoxylin staining the cell nuclei blue and eosin staining the cytoplasm a pink or red colour. The stains are examined and will be used to determine the morphological information, this is including the shape, patterns and structure of the cells and tissues. This will allow for a stage from 0-4 to be assigned to the tumour. Stage 0 being non-invasive and contained within the milk ducts while grade 4 being metastatic meaning it has spread to other parts of the body.

With one of the most important factors when dealing with breast cancer being diagnosis speed and accuracy, CAD (Computer Aided Diagnosis) systems have begun to be introduced to overcome issues when dealing with the diagnosis. CAD systems make use of machine learning and artificial intelligence to diagnose breast cancers as well as many other cancers. This works by being able to find many features within the stained sample that could have been missed by humans thus reducing the chance of human error. With a CAD system being able to analyse many pixels/ data points within seconds it will drastically cut the time it takes for a diagnosis to be made with more accuracy than before. This system makes use of deep neural networks that will be touched on further into the paper. The deep neural networks will be used to automate feature extraction and pick up on patterns the human eye could not. Throughout this paper, I will be making use of 5 pretrained CNN (Convolutional Neural Networks) models, mainly, VGG-16, Inception V3, DenseNet121, RestNet50 and EfficientNet B3 to be able to identify extensive patterns and feature extraction within the dataset and assign them to the correct stage of cancer allocated.

## Background

When diagnosing breast cancer through a screening the gold standard is usually a mammogram and microscopy biopsies stained using H&E, these are overall very successful practices but has many limitations such as false positives and false negatives as well as missed tumours within the dense tissue of the breast which can be a common occurrence leaving some results unreliable and with this then comes missed opportunities to start treatment timely. The missed diagnosed results are why there is a need for machine learning and CAD systems in the field. This will give many positive improvements to the practice and can potentially save many lives. A CAD system has the capacity to learn trends and patterns in extremely large datasets of tissue imaging allowing for accurate and quick diagnosis allowing patients to start treatment as early as possible.

Breast cancer is the most common cancer within women and has a high fatality rate when it is not acted on quickly as it allows the cancer to spread throughout the body which can be fatal. There is a need for improvement of accuracy and speed of diagnosis which will in time save many lives. Machine learning has the capacity to be able to identify trends and with this will allow for the accurate diagnosis early which many specialists would miss or would just not possibly be able to see. As it will be looking at many different variations in samples that have been loaded into the dataset it will be able to find many different patterns, structural differences and components that could be missed by the human eye and analyse the inputted tissue image accordingly.

A positive with using machine learning is that it can continuously learn and adapt accordingly allowing its diagnosis and overall performance to improve over time. Allowing the model to become more specific over time eventually being able to identify accordingly to tissue type which will then allow it to look for specific trends according to the pattern of tissue present. This will further then be able to predict specific treatment types based on the type of tissue and further based on the characteristics of tumour that have been found allowing for quick, accurate and specific diagnosis and treatment tailored to the patient.

## Literature review

### Breast Cancer

According to Smolarz, Nowak and Romanowicz (2022) Breast cancer is the most common cancer found in women in the world, with around 36% of oncological patients being diagnosed with it. They have estimated that around 2.09 million people were diagnosed with breast cancer in 2018 worldwide. It is seen that almost half of the cases are said to be in industrialised countries due to the western lifestyle consisting of poor diet, the use of nicotine and excessive stress with little to no physical activity. The most important risk factors are the gender of the person, the age of the person and how developed the country is in which they are situated. Of course, many other factors also come into play such as hormonal factors, genetic factors, diet/obesity, alcohol consumption and hormonal contraception as well as exposure to radiation at a young age.

There is said to be a link between the lack of screenings and death rate according to Smolarz, Nowak and Romanowicz (2022). Where there are less screenings done, there seems to be a higher rate of deaths compared to countries where regular screenings occur.

The main two types of breast cancer are Invasive Ductal Carcinoma (IDC) and Ductal Carcinoma in Situ (DCIS). Refer to figure 3. Ductal gives the meaning that the cancer has started in the milk ducts within the breast (the tubes that are used to deliver milk from the lobules to the nipple). While carcinoma is used as it is the term used to describe a cancer that begins in the skin or any tissue that covers internal organs.

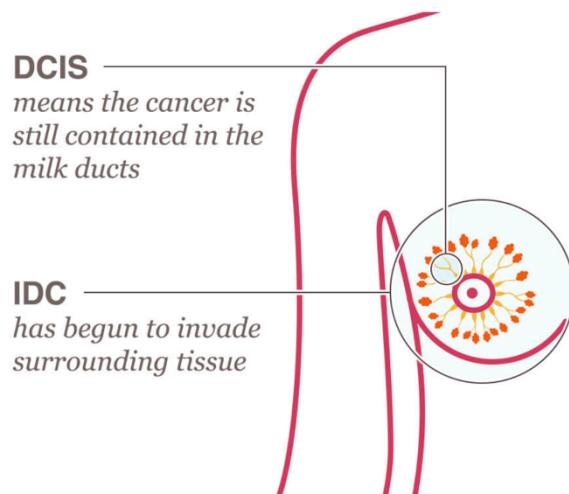


Figure 3: Image showing the difference between IDC and DCIS

It is stated by Cancer Research UK (2023) that DCIS is an early breast cancer and that some of the cells within the ducts of the breast tissue have begun to turn into cancer cells but are still contained within the milk ducts. It is split into 3 grades, grade 1 being low grade, slowly growing, grade 2 being intermediate growth and stage 3 being high grade with a quick growth of the cancer. This is based on the TNM staging system meaning Tumour, Node and Metastasis. The tumour is graded on the size of the tumour, while node is based whether the cancer has begun to spread to the lymph nodes, and metastasis is based on whether the cancer has spread past the lymph nodes and further

into a different part of the body. According to the TNM staging system a cancer classified as DCIS would present as Tis N0 M0.

The dataset that I will be using throughout the project are stages 1 to 3 of IDC breast cancer.

Invasive Ductal carcinoma can also be known as infiltrating ductal carcinoma as they both portray the same meaning that the cancer is no longer portrayed in the milk ducts and have begun to spread to surrounding breast tissue or other places in the body. The symptoms seen in IDC and DCIS present the same and the difference is seen when doing more extensive testing into the already found tumour.

As said by DePolo (2023) the stage of IDC is based on a scale between 1 and 4. Stages 1-3 will describe relatively early stages of the cancer while stage 4 describes a cancer that has spread to other parts of the body. Many considerations go into the grading of the cancer such as the size of the breast cancer, the Nottingham grade of the Cancer, tumour necrosis, tumour margins, lymph node status, lymphovascular invasion, hormone receptor status, HER2 status as well as Ki-67 levels (rate of cell growth).

Touching on a few of these considerations, according to Takahashi et al. (2020) the Nottingham Grade of the cancer has 3 different scores which are based on characteristics of the tumour, these are the tubule formation, nuclear pleomorphism and mitotic activity. Each one of these characteristics will be given a score between 1 and 3 based on the severity (1 being closest to normal, while 3 being the most abnormal). The scores of each characteristic will be added up, producing the Nottingham Grade. Grade 1 will have a total score between 3 and 5 (well differentiated), stage 2 will have a score of between 6 and 7 (moderately differentiated) and lastly, stage 3 will have a score of 8-9 (poorly differentiated). As said by OncoLink Team (2019), grade 1 cancers are seen to be less aggressive and tend to be oestrogen receptor positive (ER+), while grade 3 cancers are seen to be more aggressive and tend to be triple negative meaning that they negative for ER and PR hormone as well as HER2 receptors.

Referring to figure 4, A of the breast shows the ducts, this is where the cancers such as IDC and DCIS begin to form. DCIS will stay within these ducts while IDC will move out of

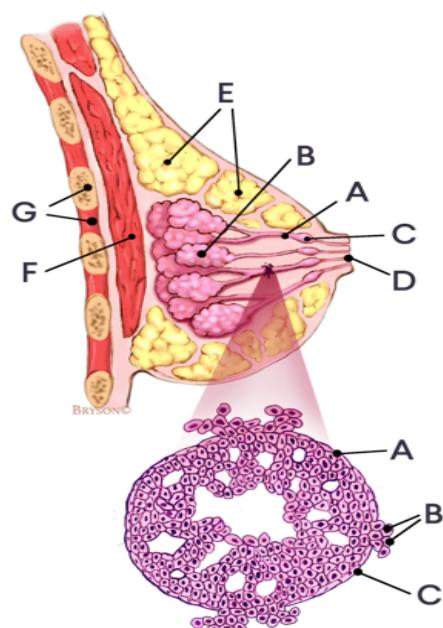


Figure 4: figure showing a normal breast with IDC (seen in the enlarged cross-section)

these ducts into surrounding tissue and then potentially throughout the body. When looking at the enlarged cross-section, A shows a normal duct cell that would be typically seen in the area, C shows the basement membrane and B shows the cancer cells that

have broken through the basement membrane allowing the cells to eventually move to other parts of the breast tissue through a process called metastasis (Figure 2).

Haematoxylin and Eosin (H&E) staining is one of the main diagnostic tools used to classify the class of the tumour that has been found. H&E staining is a gold standard in diagnosis of the cancer stage. Staining done via H&E allows the segment of the tissue to be visually split into its components as well as marking tissue structure making it for easier viewing and diagnosis of the tumour. According to Feldman and Wolfe (2014), the dye haematoxylin has a strong affinity for acidic structures in particular nucleic acids such as DNA and RNA allowing it stain the nuclei by attracting and binding with positively charged haematoxylin molecules it will then form complexes with the negatively charged phosphate groups in the DNA giving it the blue/ purple colour. The eosin dye stains the basic structures in the cell as they have a net basic charge and is often referred to as a counter stain as it is applied after the haematoxylin. According to Sampias and Rolls Haematoxylin shows the ribosomes, chromatin (genetic material) within the nucleus, and other structures as a deep blue/purple colour while Eosin will show the cytoplasm, collagen, connective tissue, and other structures that surround and support the cell as a pink/red colour. This allows for the pathologist to make a comprehensive diagnosis based on the structure and cellular components (refer to figure 5).

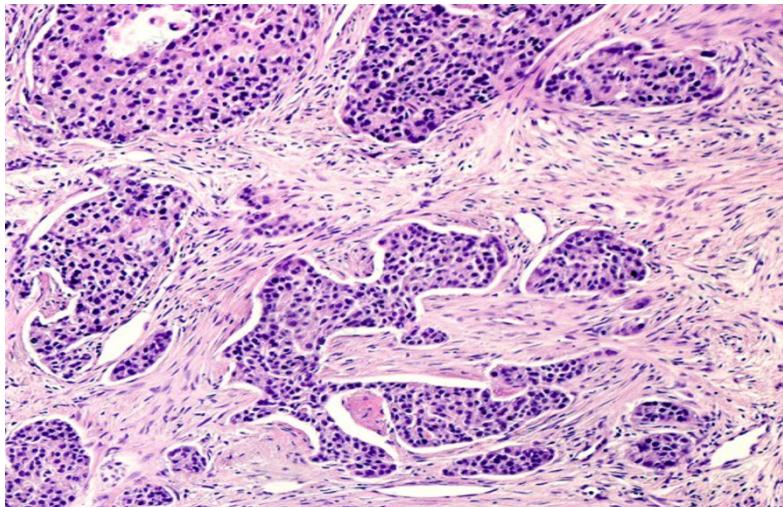


Figure 5: IDC in breast tissue with 40x magnification stained with H&E

When looking at figure 5, it is fairly easy to be able to distinguish between the cellular components as well as being able to view the tissue structure. This allows the pathologist to be able to make a diagnosis accordingly. The genetic material is stained in a purple colour (haematoxylin) while the rest of the support structures of the cell stained in a pink colour in this example.

### Pre-processing and normalisation

Pre-processing and normalisation are of the highest importance throughout this project as without adequate pre-processing and normalisation, the model will not be able to pick up patterns or cellular components leaving many patterns and cell components undetected and potentially causing a missed diagnosis. There are many ways in which to normalise and pre-process the images but ultimately comes down to which is most suited to the images. Upon inspection of the data set it is clear that many of the images differ in noise, as well as in colour. Many of the pinks and blues seen in the images are differing ranging from a very red to a light orange instead of the pink that should be seen such as in figure 5. While the purple of the haematoxylin seen in figure 5 ranges from an almost black colour to a very light blue.

With these differences in values the model will not be able to differentiate between them during processing causing the model to be inaccurate. Upon research that has been conducted there a few attributes that should be added to the pre-processing, according to Mahato (2023), noise reduction/ sharpening of the image is highly important especially in medical images to preserve the components and patterns seen within the image while making the image clearer as a whole. Image resizing is also highly important so that image can be read by the model while being able to use the resources available to the optimum level. Lastly, the most important factor is to be able to colour correct all the images to have them in either the same colour or very similar making it easier for the model to be able to read each images and distinguish patterns and cellular components. The rescaling will be done by rescaling the image between [0,1] making the image all the same throughout the dataset giving the model a consistent input while allowing for greater convergence of the model. This also allows the image to be visually normalised as the pixel values will be the same throughout the dataset. The images will all need to be resized, this will either be done during the pre-processing stage or during data augmentation later on in the process.

According to Paul (2021), the optimal images size for the image to be resized to before adding it the CNN (convolutional neural network) model for processing is a lower dimension of either 224x224 or 299x299, allowing the model to be able to keep up with batch learning as well as to keep up with the computers limitations based on the resources. This will allow the images to be more suited to the pre-trained model instead of the human eye. Many of the pre-trained model require the image to be in the 224x224 format.

Lastly, and probably the biggest challenge in the pre-processing steps is the colour normalisation of the images. As the images are of H&E stains and the RGB colour format the it is vital the colour channels are split properly as these different channels are used to capture different aspects of the image (the H&E channels). As there are many colour imbalances in the image , a constant colour balance needs to be found so that all the images will be in the same regions when it comes to processing of the images. When doing research on how to split the channels I can across the method using SKImage, this allows the image to split into their corresponding channels, in this case the H&E channels. As said by Bankhead (2022) the image is converted from a RGB image to a HED colour space ( haematoxylin, eosin and DAB) as DAB is not present in these stains it will be left out. The images will be stored accordingly to their channel, with the other components of the image turned into grayscale. The image will then be rescaled according to their intensity making all the images even before combining them again to create one image with full colour and grey scale at all the same colour range and intensity. Another option I have encountered is using a target image to be able to perform histogram matching throughout all the images in the dataset. As explained by Rosebrock (2021), this is done to adjust the histogram of an image to the target image images histogram giving the images the same appearance of colours throughout the dataset. Through this process of histogram matching, the code will compute the histogram of the target image used as well as the pixel intensity, it will then do the same for the source image. Once the code has obtained these values it will match them according to the channel they have been split into (H&E). this should give the images very similar colours if not exactly the same, if these do still differ the intensity of each of the images can be readjusted to make sure the pink and purples/blues really stand out in each of the images making the channels and patterns very distinguishable to the model that will be used. This will be done using OpenCV.

## The Model

When researching which kind of model works best with images there is really only one true option and they are Convolutional Neural Network (CNN) models. As explained by Mishra (2019), these models are very effective in being able to reduce the parameters without losing quality of the images. They are highly skilled and are trained to be able to identify edges of the objects in the images, making it very useful when detecting cellular components and patterns, they also treat each pixel as a feature throughout the processing making sure that every pixel is accounted for and considered when making an assumption based on the stage of cancer. With the huge amount of parameters that need to be taken into account the only model capable of processing this is the CNN model.

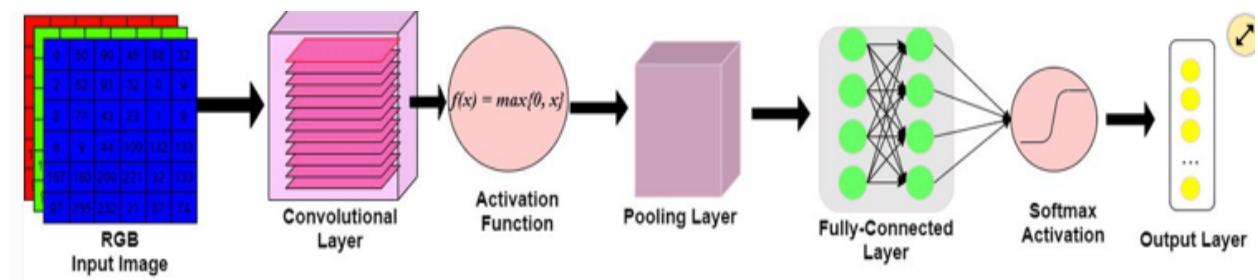


Figure 6: graphic representation showing the architecture of a CNN model

As explained by McDermott (2020), the model (Figure 6) consists of a convolutional layer, an activation function, a pooling layer and a fully connected layer leading to the output layer. The output of nodes related to local areas of the input matrix will be computed by the convolutional layers, these Dot products are computed between values related to a particular region of the input and a set of weights, often known as a filter. For the activation function ReLu will be used for the first and soft max for the second. The output of the convolutional layer will be then moved to the activation function which will determine whether or not the input node has detected a visual feature in the given input data. The activation function will apply a  $\max(0, x)$  function that will threshold at 0 keeping the dimensions of the volume at the original amount. The pooling layer is used as a down sampling technique meaning that it will reduce the width and height of the output volume so that the model wont pick up on the same visual feature multiple times. Lastly, the fully connected layer which consists of nodes will receive the output volume from the previous step. The probabilities of which class the image belongs to are computed which is then outputted in the format of a 3D array with  $[1 \times 1 \times k]$ . The  $1 \times 1$  is the shape of the image that was previously inputted while  $k$  is the predicted class in which the image belongs to.

## Pre-trained models

When creating the model, I will be using a pre-trained model. These models are specific to feature extraction so resources won't be wasted in creating a CNN model from scratch this will also allow for the model to be more specific to the case that it is needed for. These premade models already have many useful features that have been trailed and tested on many different datasets which in essence has led to the model being able to recognise extensive features and patterns of the dataset such as the components and patterns within the image as well as the textures to an extent allowing for more patterns to be recognised and better evaluation to be made.

As these models have been tested on many different datasets, they have learnt many patterns and representations allowing them to adapt and adjust to the dataset being used allowing the model to respond well to limited datasets as well as saving training time and resources as they adapt to previous cases.

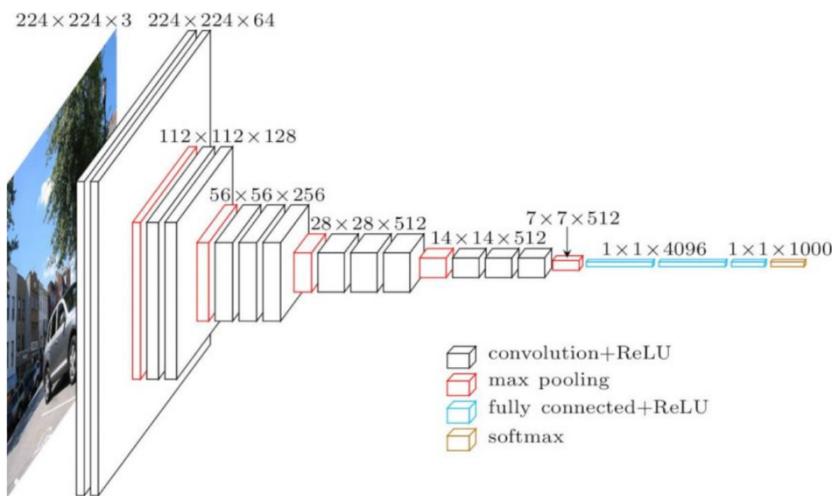
The pretrained models are more prone to overfitting than building one from scratch would be, especially with a smaller dataset such as the one that I have. These models have a strong base as they are specially created for datasets such as the one that I will be using allowing the last few layers to be adapted to the dataset being used.

The use of pretrained models will also allow for the least likeliness that the code will contain bugs or errors in the code as only the last layers will need to be adapted to the used dataset allowing for less chance of error and if errors occur than it will make it easier for the problem to be found and solved.

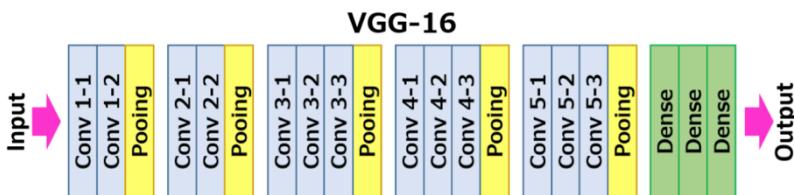
There are many options available when using a pretrained model, the challenge comes in when picking the one most optimal and suitable to the dataset being used. With this comes trial and error on finding the one best suited to the dataset.

## VGG-16

Very deep convolutional networks for large scale image recognition (VGG-16). The writer has claimed that this pretrained model is the most popular when it comes to image classification. This model uses many filters throughout as seen in figure 7 below.



Here is a more intuitive layout of the VGG-16 Model.



*Figure 7: The figure shows the layers of the VGG-16 model.*

As explained by Huilgol (2020) As it is a large model, it will take a lot longer than other models to train the data. The model is sequential (following in a logical order) and makes use of many filters. Through each stage the image will go through small  $3 \times 3$  filters

reducing the number of parameters each time. All these layers will use the ReLU activation function which will be touched on further later in this section. The model contains 13 convolutional layers, 5 pooling layers as well as 3 dense layers. As explained by Boesch (2021), the input need for the VGG model is 224x224. VGG's convolutional layers make use of a small receptive field, namely 3x3, the smallest size that still captures up/down and left/right. There are also 11 convolution filters that act as linear transforms of the input, this is followed by a ReLU unit. It will then return an input if positive otherwise it will return 0. The convolution stride is set at 1 pixel to maintain spatial resolution after convolution. The VGG network's hidden layers all use ReLU. Local Response Normalisation (LRN) is rarely used in VGG since it increases memory usage and training time. Furthermore, it has no effect on total accuracy. Lastly, the VGGNet is made up of three fully connected layers. The first two layers each have 4096 channels, while the third has 1000 channels, one for each class.

The number 16 in the term VGG-16 refers to the fact that it is a neural network with 16 layers. This suggests that this is an extremely large network with approximately 138 million parameters. With such a huge model running through so many parameters a down side to using this is that the training time is very extensive but with this comes a high level of performance and has said to outperformed smaller networks with around a 7% test error surpassing networks such as Inception by around 1%.

## Inception V3

This model has been created by google and is a lot smaller than the VGG model with around 24 million parameters compared to the 138 million that is present in the VGG model. Inception is also said to have a lower error rate than the previous model making it one of a kind and very suitable choice when working this sort of dataset.

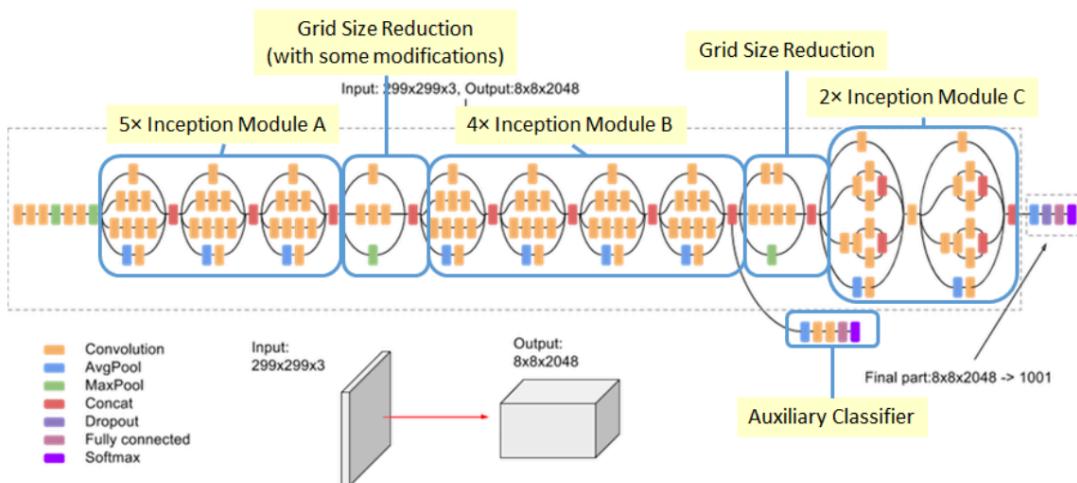


Figure 8: image showing the architecture of the Inception V3 model

As explained by the Google Cloud Team (2019) while looking at the figure 8 above, it can be seen that there are 42 convolutional layers compared to the 13 present in the VGG-16 model. This will allow for the model to be able to identify more complex patterns throughout the image as it will go through more layers of processing potentially being able to pick up patterns that might have not necessarily been picked up in other models with smaller convolutional layers. With this large increase in convolutional layers there is potential for overfitting so the dropout layers will need working at an optimal to prevent this from happening. This model also has a reduced error rate of 4.2% which is of extremely high value when working with datasets in the medical field where errors are extremely costly and could be the difference between life and death when referring to the diagnosis of breast cancer.

As said by Kurama (2020) while looking at figure 8, inception module A consists of 5 branches, the first branch is a 1x1 convolutional layer used for reducing dimensions and capturing features. The second branch is again a 1x1 convolution that is followed up by a 3x3 convolutional layer which is used to combine the local and global information. Third layer consists of a 1x1 convolutional layer followed by 2 3x3 convolutional layers which allows the model to gather more in-depth information. The fourth layer is a begins with an average pooling layer (seen as the blue colour) which will then go through a 1x1 convolution layer allowing the model to gather data from the entire input. Lastly a 1x1 convolution layer and 5x5 convolution layer is added to allow the model to obtain information from a broader receptive field. A grid size reduction is introduced which is a technique used to reduce the dimensions of the feature maps of the image and reduces the computational power needed cutting down on the model training time and capturing more abstract features that could have been missed prior. The inception module B has four branches and follows the same pattern as module A does but instead introduces 7x1 convolution layer to a 1x7 convolution layer and then after that uses a 1x7 to a 7x1 convolutional layer this allows the model to capture elongated features allowing the model to find features at different scales. Lastly it will contain the same layout of the prior max pooling structure. A second grid size reduction is used which will have the same effects as the previous one. Module C consists of 2 branches, firstly a 1x1 convolutional layer followed by two parallel 3x3 convolutional layers, this will allow the model to capture multiple different features across various scales. It makes use of auxiliary classifiers which include their own loss functions during the training of the data. The goal of these classifiers is to address the vanishing gradient problem by adding additional gradients to the network's previous layers and once used the auxiliary classifiers are discarded.

As explained by the Google Cloud Team (2019) the inception model can make use of 3 different optimizers in the processing of the image, these are SGD, momentum and RMSProp.

RMSProp being the most accurate and for this reason being the most popular with many users. This optimizer is also a good alternate to Adam. RMSProp was first proposed by Geoffrey Hinton, it was developed as a technique for mini batch learning. "RMSprop deals with the above issue by using a moving average of squared gradients to normalize the gradient. This normalization balances the step size (momentum), decreasing the step for large gradients to avoid exploding and increasing the step for small gradients to avoid vanishing." ([medium.com](https://medium.com)). To put this in simple terms, it explains that it will use an adaptive learning rate throughout, allowing it to change over time instead of being a static number throughout.

The inception model makes use of new techniques not used in models such as VGG-16. This model uses a series of convolution layers containing different kernel sizes, this allows the model to work parallel and capture unseen information at different scales allowing the model to learn a wider range of features. It makes use of 1x1 convolution layers, allowing these smaller layers to reduce the computing resources while reducing the dimensions of the image meaning that it will reduce the input channels before the application of the larger convolutional layers.

Multiple branches of inception modules apply different-sized convolutions to the input. These branches' outputs are then combined along the channel dimension, allowing the network to collect a wide range of features in the image. Instead of standard fully connected layers at the network's end (as seen in many other models), Inception v3 employs global average pooling. This technique decreases the spatial dimensions of the feature maps and aids in the prevention of overfitting. The inception model makes use of regularisation techniques throughout the model, it makes use of dropout and L2 weight

decay to prevent overfitting. These strategies improve the model's generalisation performance on previously unknown data picking up on components or patterns that could have been previously missed in other models.

## ResNet50

This model is not the first from the RestNet family this model is the successor to the original model created which struggled with accuracy as it simply was not deep enough.

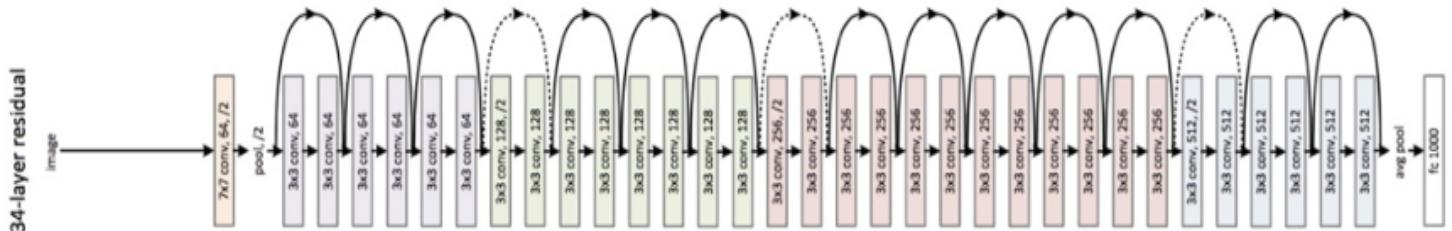


Figure 9: Image showing the architecture of the ResNet34 model

Explained by Figure 9 shows the ResNet34 model, the architecture stays the same in the ResNet50 model just more layers are added. As explained by Huilgol (2020) The Res in ResNet stands for residual and that is the main focus of this model presented, the model is fairly standard throughout but adds a new technique based on skipped connections which are referred to as identity shortcut connections, this technique makes use of residual blocks. The creators of ResNet had come to the conclusion that the fitting of a residual map is easier than the fitting of actual mapping.

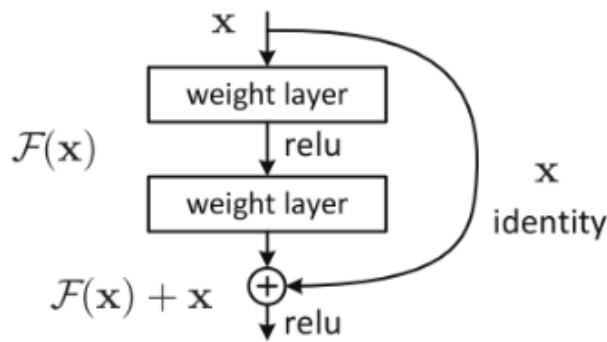


Figure 10: image showing the workings of the residual mapping

Looking at figure 10, after every 2 layers the model will bypass the layer in between. The identity path is used to copy the input directly to the output while creating the identity mapping while the residual path learns the residual (the difference between the output and input). The skipped connections directly add the input to the output of the residual blocks, even if the model learns identity mapping in the residual block the skip connection will allow the model to pass through the input remaining unmodified. This in a whole will allow the model to train very deep networks by eradicating the gradient

mitigation letting the gradient flow through the skip connection. Optimization is also improved as the model focuses on residual mappings instead of the whole function. The model is extremely intuitive as the more layers that are stacked on it, the model performance should not worsen.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112			$7 \times 7, 64, \text{stride } 2$		
				$3 \times 3 \text{ max pool, stride } 2$		
conv2_x	56×56	$\left[ \begin{array}{l} 3 \times 3, 64 \\ 3 \times 3, 64 \end{array} \right] \times 2$	$\left[ \begin{array}{l} 3 \times 3, 64 \\ 3 \times 3, 64 \end{array} \right] \times 3$	$\left[ \begin{array}{l} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{array} \right] \times 3$	$\left[ \begin{array}{l} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{array} \right] \times 3$	$\left[ \begin{array}{l} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{array} \right] \times 3$
conv3_x	28×28	$\left[ \begin{array}{l} 3 \times 3, 128 \\ 3 \times 3, 128 \end{array} \right] \times 2$	$\left[ \begin{array}{l} 3 \times 3, 128 \\ 3 \times 3, 128 \end{array} \right] \times 4$	$\left[ \begin{array}{l} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{array} \right] \times 4$	$\left[ \begin{array}{l} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{array} \right] \times 4$	$\left[ \begin{array}{l} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{array} \right] \times 8$
conv4_x	14×14	$\left[ \begin{array}{l} 3 \times 3, 256 \\ 3 \times 3, 256 \end{array} \right] \times 2$	$\left[ \begin{array}{l} 3 \times 3, 256 \\ 3 \times 3, 256 \end{array} \right] \times 6$	$\left[ \begin{array}{l} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{array} \right] \times 6$	$\left[ \begin{array}{l} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{array} \right] \times 23$	$\left[ \begin{array}{l} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{array} \right] \times 36$
conv5_x	7×7	$\left[ \begin{array}{l} 3 \times 3, 512 \\ 3 \times 3, 512 \end{array} \right] \times 2$	$\left[ \begin{array}{l} 3 \times 3, 512 \\ 3 \times 3, 512 \end{array} \right] \times 3$	$\left[ \begin{array}{l} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{array} \right] \times 3$	$\left[ \begin{array}{l} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{array} \right] \times 3$	$\left[ \begin{array}{l} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{array} \right] \times 3$
	1×1			average pool, 1000-d fc, softmax		
FLOPs		$1.8 \times 10^9$	$3.6 \times 10^9$	$3.8 \times 10^9$	$7.6 \times 10^9$	$11.3 \times 10^9$

Figure 11: table showing the architecture of the ResNet model depending on the size

## EffcientNet

As explained by Huilgol (2020) google are the authors of EfficientNet and have introduced a new scaling method by the name of compound scaling. This means that the dimensions are scaled by a fixed amount and fixed time allowing the performance to improve. These dimensions are the width, depth and resolution of the model allowing for better resource allocation as well.

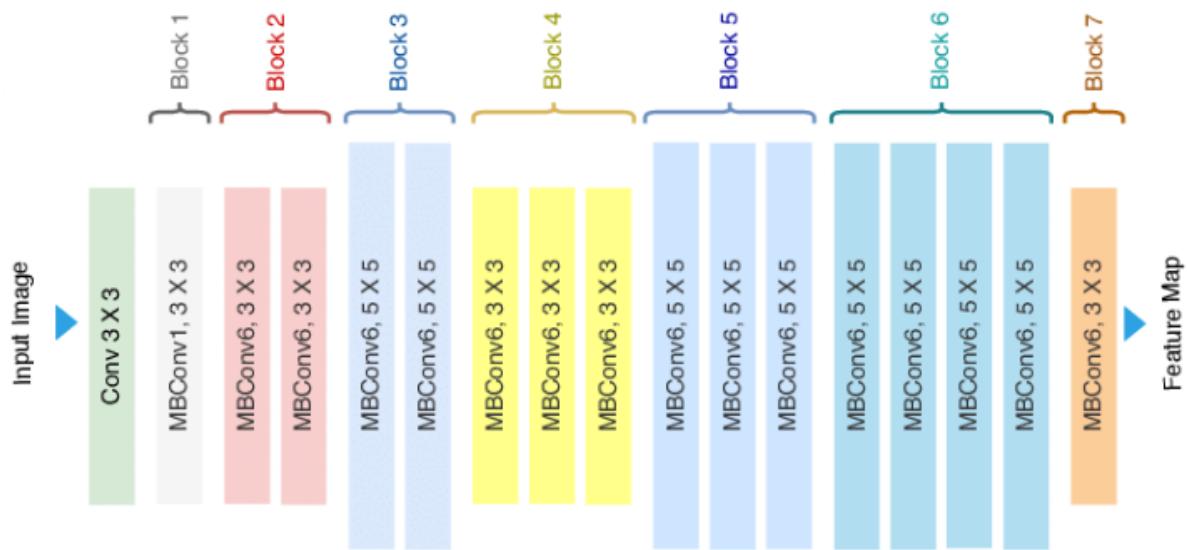


Figure 12: Image showing the basic architecture of EfficientNet B0

Efficient net has various versions ranging from B0 to B7, the differences between all these versions is the compound scaling factors, these scaling factors used determine the number of channels as well as the number of layers.

Figure 12 shows the basic architecture of the EfficientNet B0 model, the label MBConv stands for mobile inverted bottleneck convolution. As the model is based on the scaled features B0 has the following scaling coefficients, a depth of 1.20, width of 1.10 and a resolution of 1.15, as the model number increases the scaling coefficients increase as well leading to a greater number of parameters, for instance the B0 model has around 5.3 million parameters while B7 will have a round 66 million so by making small changes they will expand the model exponentially.

All the versions of EfficientNet make use of inverted residual blocks that contain separable convolutional layers, the increase in parameters is there because as the model allows for larger input as you go through the models the parameters need to be increased to able to cope with the larger amount of pixels. With all of this comes a trade-off, the lower models have a very quick convergence speed and training time but potentially a lower accuracy. As the model increases, it becomes bigger and more parameters are added which results in a much longer training time but with that comes a much higher accuracy. This can be seen in figure 13.

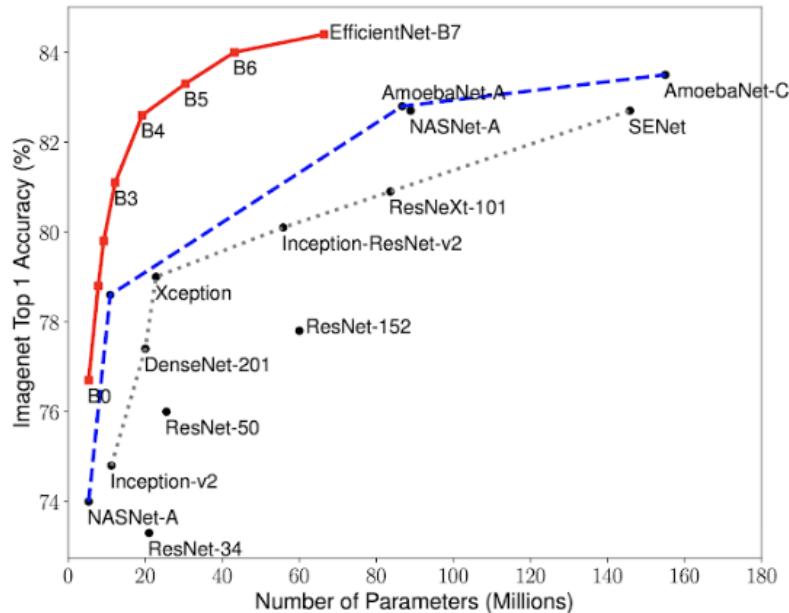
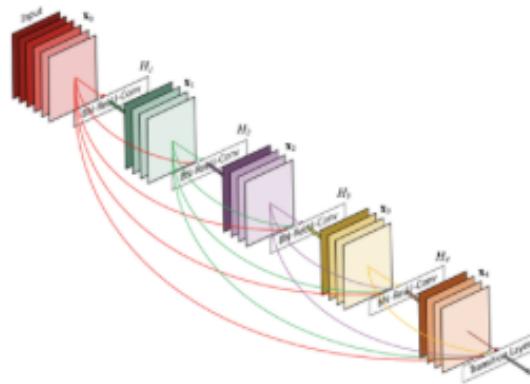


Figure 13: Graph comparing the accuracy of the EfficientNet models to the number of parameters in the model

DenseNet

Explained by Baldha (2022) , DenseNet stands for densely connected convolutional network. This model is extremely similar to the previous model ResNet with some very minor differences. The main technique used in this model is that it will use all previous output as an input for a future layer (refer to figure 14).

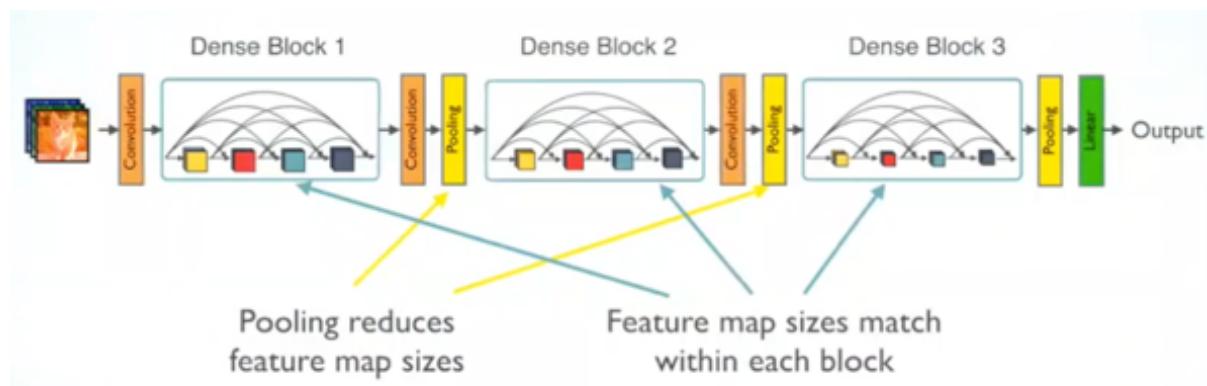


## DenseNet Structure

$$a^{[l]} = g([a^{[0]}, a^{[1]}, a^{[2]}, \dots, \dots, a^{[l-1]}])$$

*Figure 14: Figure showing the architecture of the DenseNet Model*

This will include the use of dense blocks which consists of multiple layers densely connected to one another. This will allow the feature map to be used from all proceeding layers within the same block. DenseNet leverages dense connection by concatenating all preceding layers' feature mappings in a dense block. This means that each layer not only receives its own feature maps as input, but also the feature maps of all previous layers in the block. Because of this deep interconnectedness, each layer has access to a diverse set of features from various stages of processing (refer to figure 15)



*Figure 15: Figure showing the use of Dense block and the layers of the model*

Another technique used in this are bottleneck layers which is introduced before every convolutional layer, this is in place to reduce the number of input feature maps reducing computing power needed. To regulate the spatial dimensions of the feature maps, transition layers are provided between dense blocks. A convolution layer is often followed by a pooling layer in these transition layers. Before transmitting the information to the next dense block, transition layers help to reduce the number of feature maps and spatial dimensions. The DenseNet architecture commonly concludes with a global average pooling layer, followed by a fully connected layer with SoftMax activation for classification tasks such as with the dataset that I am using. This model is summarised by its key advantage which is that you are in the third layer, the third layer will take input from not just the second layer but also as the first layer, and by doing so, the model is trained better, picking up on more patterns and in this case components of the image.

## Project planning

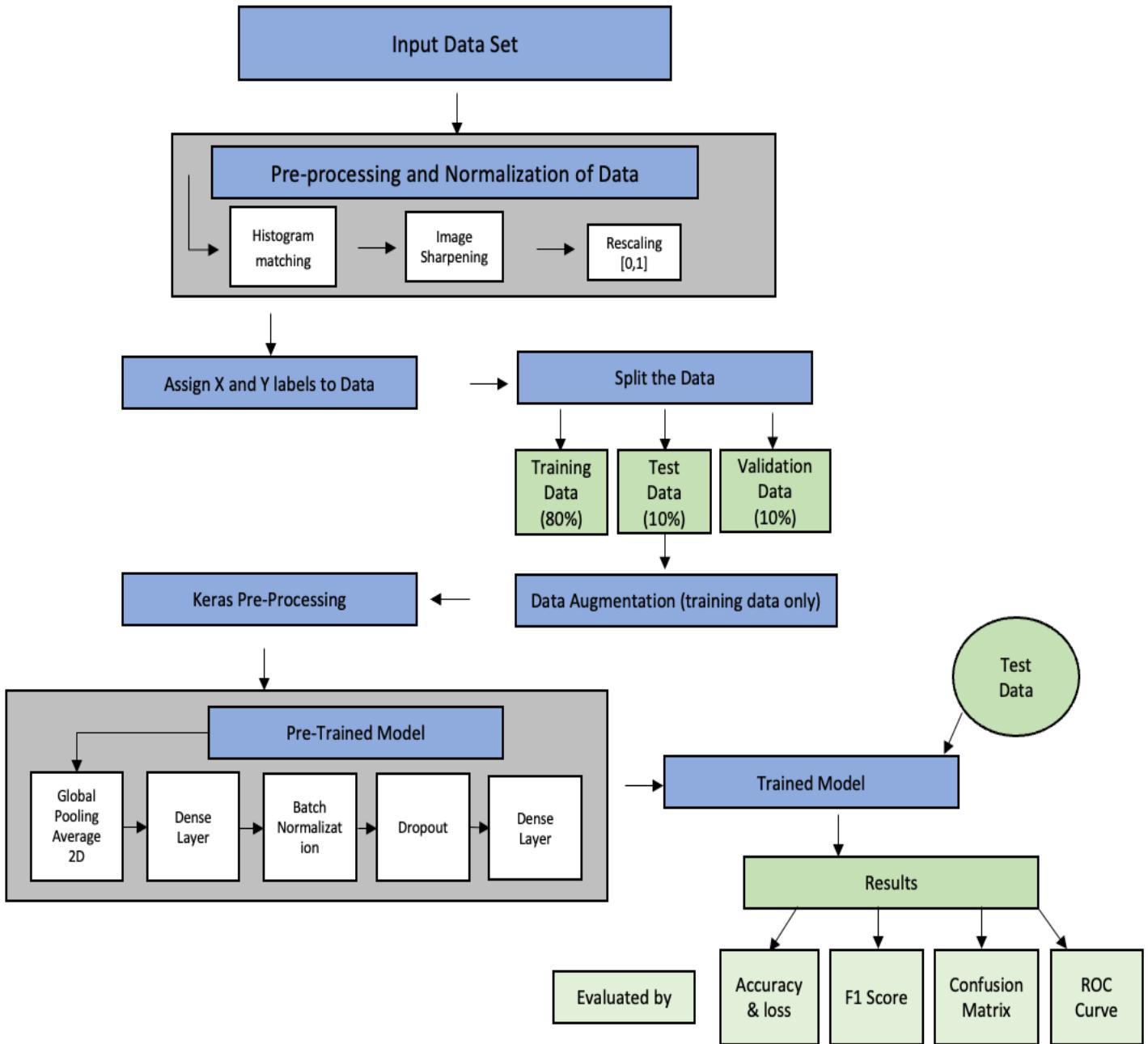
### Aims and Objectives

The aim of the project is to be able to train the machine successfully to be able correctly identify the stages 1,2 or 3 of IDC breast cancer within the breast tissue sample at a high speed with a high accuracy. This will allow for the overall increase of survival rate of the patient removing human error and slow turn over times which delay the start of treatment.

- Allow the pre-trained used to be able to accurately determine the stage of the breast cancer according to the allocated class (0,1,2).
- Successfully normalising and pre-processing the images in dataset making it easier for the model to classify these images into allocated classes.
- Allow for the model to be able to look for and understand patterns/ trends in the images of the dataset and classify them accordingly.
- Allowing the machine to have a high accuracy in determining the class of the sample at a high-speed allowing for quick and accurate diagnosis for the medical professional.
- Create an algorithm that doesn't have weak points and will be able to work on all images inputted.
- Evaluate and score the models used thoroughly to determine which one of the pre-trained models is best suited to this case.

The aim of this project is to be able to use machine learning with the use of pre-trained models to be able to predict the stage of IDC breast cancer from H&E stained images

## Project Design



The design of the project seen above will be further expanded on in the forthcoming sections of the paper.

The tasks that will be set within this section will all be in line with the SMART (Specific, Measurable, Achievable, Relevant and Time bound) goals technique.

#### **Stage 1- Conduct Research**

This will be consisting of conducting all necessary research that will give me the knowledge to be able to complete this project. This will allow me to build a base as well as review where other people struggled to allow me to get ahead of the issues that will be experienced. It will also help in finding ways in which to improve previous research that has been done. This will give me a path in which to go that will allow me to successfully complete this project.

#### **Stage 2 - Familiarise and gather data that will be used in the project.**

After the base of research has been conducted the data needed will need to be collected and made sense of so that I will be able to identify whether the model that has been trained is making the correct accurate observations. This will be looking at the data and making sense of it before inputting them into the model, allowing myself to understand what needs to be done in pre-processing and normalisation, giving myself the optimal results.

#### **Stage 3 – Creating the algorithm**

Begin to start building the algorithm that will allow the model to be able to identify the differences within stages of breast cancer according to the data. This allow me to test different algorithms and select the most effective one, this will be done based on prior research that I would have done. I will begin imputing images to test the model and adjust accordingly, this will be based on any sort of errors or inaccuracies seen in the model.

#### **Stage 4 – Model training and testing**

Testing will commence once I am satisfied with the model and how it is performing on the training data. Once the code is working and up to the level that is wanted the fine tuning of the model will proceed. Once completely satisfied with the results, stage 5 will commence.

#### **Stage 5 – Review of the algorithm**

During this stage I will look at the results I have received from the previous steps and decide whether the results received are accurate and up to a high standard. If any issues present or if I do not believe are up to standard then I will return to stage 3 and 4 to make improvements to the code, then work my way back up to stage 5 and review again until I am completely satisfied with the code I have produced. This stage will involve the use of F1 score, confusion matrix and ROC curves to have a true understanding of the model and received results.

#### **Stage 6 – Report**

This stage will begin when I am completely satisfied with the results received. During this stage I will begin to write my thesis on the results I have throughout stages 1 to 5. This will include all the background information I have researched through the beginning stages to give the reader the necessary information to understand the results in full. This stage will take me a couple weeks to complete, and a lot of hours will need to be put into this section, so it is vital that time management plays an important role. This will allow me to have the thesis completed by draft deadlines and will allow for me to review it many times before submission.

#### **Stage 7 – Feedback from Draft**

This stage will come after the draft submission of the thesis. Each comment made by my supervisor will be taken in and improved on so that the best possible report will be submitted. These changes will be made in the period before the final submission and

extensive time management will be needed during this short period to make sure all sections that need improvements can be acted on before the final submission.

Planning is crucial in the timely completion of the project. This is one of the most important aspects of any sort of research, it is crucial in a successful project which I am aiming at completing.

There are a few elements that are crucial when planning a project, these need to put into practice to be able to complete a successful paper.

Project goals – the goals of the project need to be clearly defined so that you know what needs to be achieved, allowing yourself to stay on track and focused throughout.

Time management/ timeline – I will be creating a timeline that will allow me to stay on time throughout the project preventing myself from falling behind or missing important steps or deadlines. One of the most important part of aspects of creating a timeline is that it needs to be realistic, when creating my timeline, I will also be giving myself additional time which deviate for any delays or obstacles I will potentially come across keeping myself on time throughout.

Resources - identifying the resources prior to the beginning of assignment will allow me to be completely prepared to begin the project while not having to worry about obtaining anything during the project which will cause setbacks. The resources used in this project will be the information needed which will touched on in literature review.

Risks – identifying risks prior to the start of the project will allow me to get a head start on any potential risks I will run into during the duration of the project. This will allow me to be prepared and quickly combat these risks if any come forward. Keeping the project on track preventing falling behind and missing goals on my timeline.

## Project Risks

During the duration of my project, I am, as anyone is bound to encounter some sort of error or issues with the work. Looking at the potential risks I may encounter will allow me to get a head start on these errors and allowing myself to stay on track preventing falling behind in any sections. During this section I will be looking at any potential risks I may encounter as well as the severity and how to deal with them by looking at potential solutions.

*Table 1: table showing the risk that could be experienced as well as the probability, severity and risk management*

Risk	Risk Description	Probability	Severity	Risk management
Data privacy and security. Ethical considerations	Working with sensitive patient data can have an impact of the ethics of this project.	low	High	Appropriate measures will need to be taken to data used. Make sure that all the data is free to use and not copyrighted.
Bias and fairness of model	If the model is not correctly trained it can create a bias towards a certain result	Medium	High	Making sure the data set is very diverse allowing for all types of results. Focusing on the pre-processing and normalisation.
Interpretability	Difficulty interpreting the results created by the model	Medium	Medium	The model needs to be created so that it is easy to interpret results and communicate these easily. Make use of other tests to interpret results.
Time related risk	Falling off track due to other university commitments or procrastination. Leading to missing a deadline or late starts on the next task	Low	High	Make sure reminders are set for each task based on the timeline. Limit procrastination by removing distractions. If timeline isn't working out,

	prompting changes in the timeline created.			then quickly adapt a new timeline without wasting needed time debating how to change it.
Communication errors	Lack of communication with my supervisor can lead to a lack of progress or movement of the project in the wrong direction.	Low	High	Make sure to have regular check ins with supervisor to stay on track allowing for thorough progression of the project.
Technical issues	Experiencing technical issues with any sort of software or hardware that leads to delay of project. Including not saving work leading to the loss of the work.	Low	Jeopardy	Make sure all software and hardware is up to date and working correctly. Making sure to save updated project/progression to device as well as to the cloud giving multiple copies that can be accessed if any sort of error had to occur

Many of these potential risks can be easily prevented if the correct steps and measures are put into place. As easily as they can be prevented, if the correct steps are not put into place, they will have a severe impact on the project and results that the model will produce. It is with the utmost importance that these risks are avoided thus leading to the successful and timely completion of the project.

## PSEL Issues

When completing a project of this magnitude, every aspect needs to be considered to avoid any risks, errors, or PSEL issues. PSEL issues consist of ethical, legal, professional, and social issues that can arise within this project. These are crucial for the conductor of the project/ professional to be able to understand work through their roles by maintaining integrity and staying professional throughout. This will also allow the conductor of the project to maintain their highest possible standards.

Ethical issues – when working with medical data there is always going to be considerations to adhere to. The data used within this study needs to stay adhere to the data protection act as sensitive patient data will be used also making sure it is free to use. I will need to have consent to use the data set and data privacy. I will be using an already generated sample dataset so patient sensitivity will not apply in this case. The technologies need to be fair and not adhere to any sort of bias based on the patient such as the gender, ethnicity, sexuality or disability (Takahashi et al., 2020).

Legal issues – this project makes use of many types of software as well as a data set with sensitive information, with the use of this software and data, there is a need to make sure that all copyright and intellectual property rights are adhered to. It is crucial for the project that these laws are stuck to, and no overstepping occurs. This would be detrimental for the project. Throughout this project anything that has been used from any source or that software/ data that needs acknowledgment to use will be acknowledged and referenced with the source allowing the original author or creator to obtain full credit for their work/ findings. Thus, not breaking any laws regarding intellectual property rights inevitably voiding the project.

Professional issues – throughout the project, the project must be conducted professionally keeping integrity within all the research and results. Meaning that no results shall be falsified or fabricated all the results need to be 100% accurate and honest. These results that will be obtained will not be misrepresented or misinterpreted leading to the wrong conclusion. This will allow for a valid project that will adhere to all guidelines with true results so that the results/ conclusion will be reliable and completely accurate.

Social Issues – the social impact will need to be taken into account when conducting the project. These are concerns or problems that can affect communities, professions and the individuals of this profession. This issue could potentially be with how the model is communicated to the healthcare sector. The model and its results will be needed to be able to be explained and interpreted by medical professionals to give the patient an understanding on how they have reached decisions based on the results (Takahashi et al., 2020).

## Time Plan

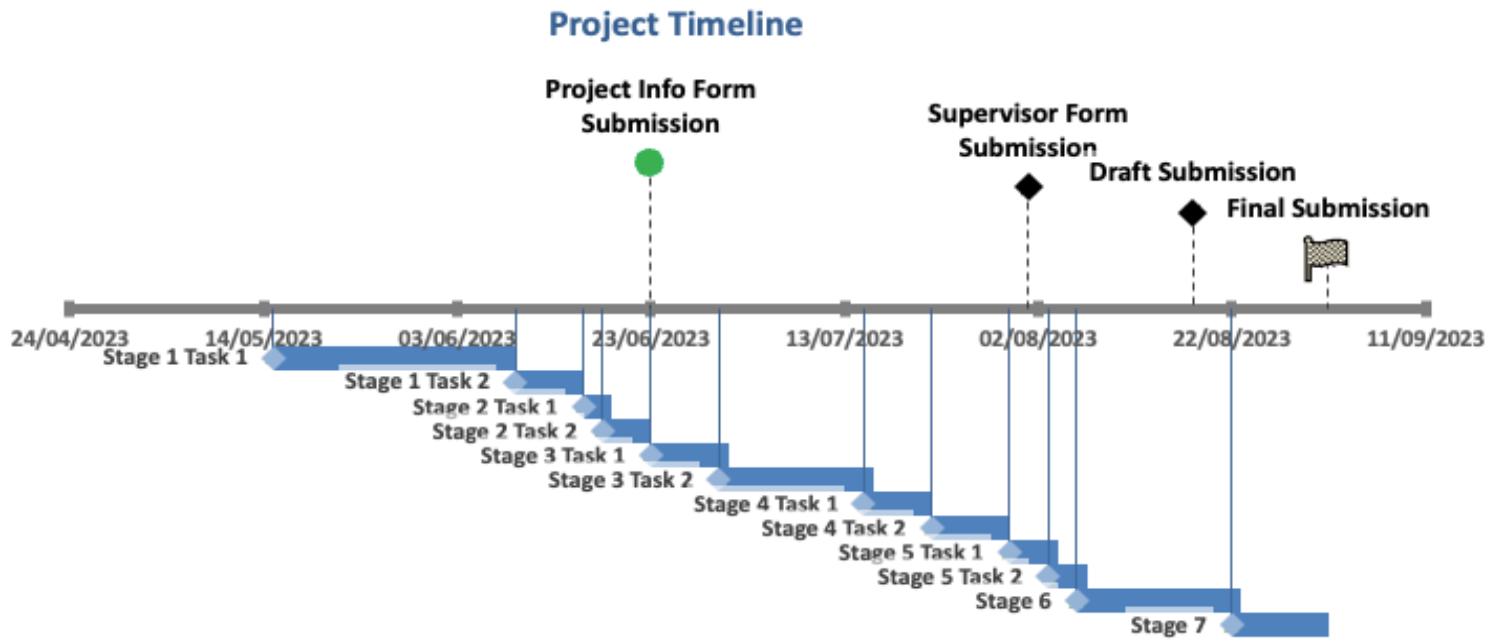


Figure 16: The timeline that has been created for the project, including the milestones above timeline

The following time line is broken down into stages and tasks in the following table (table 2).

Each step and the time they have been allocated will be located underneath in a condensed format. Deadlines are subject to change and once confirmed will change dates accordingly. The number of days is the maximum allocated and will potentially finish tasks before these dates.

*Table 2: The table shows the start and end date of the task as well as the maximum days allocated to each task*

Task	Start date	End date	Maximum days
<b>Stage 1</b>			
Additional research on topic	15/05/2023	08/06/2023	25
Finding potential data set	09/06/2023	15/06/2023	7
<b>Stage 2</b>			
Confirm dataset	16/06/2023	18/06/2023	3
Inspect the dataset	18/06/2023	22/06/2023	5
<b>Project information form submission</b>		23/06/2023	
<b>Stage 3</b>			
Research algorithms	23/06/2023	30/06/2023	8
Code the model/ create algorithm	30/06/2023	14/07/2023	16
<b>Stage 4</b>			
Training the model	15/07/2023	21/07/2023	7
Test the model	22/07/2023	29/07/2023	8
<b>Stage 5</b>			
Evaluate and score the model	30/07/2023	03/08/2023	5
Analyse the results	03/08/2023	06/08/2023	4
<b>Supervisor form submission</b>		01/08/2023	
<b>Stage 6</b>			
Write thesis/ report	06/08/2023	22/08/2023	17
<b>Draft submission</b>		18/08/2023	
<b>Stage 7</b>			
Review report based on feedback	22/08/2023	31/08/2023	10
<b>Final submission</b>		01/09/2023	

## Materials and Methodology

### The Dataset

The dataset has been created by Poursina Hakim research centre of Isfahan University of Medical Sciences in Iran. This dataset consists of histopathological microscopy images of patients diagnosed with breast cancer ranging from stages 1 to 3. This dataset will be used for grade classification obtained by using a model to be able to predict the label assigned to each of the images. This dataset consists of 922 images, all the images are in a RGB format as a JPG type with resolution of 2100 x 1574 x 1276 x 956 pixels. All the images are stained using H&E and have been taken from 124 patients with their diagnosis between 2014 and 2019.

Within the dataset there are four levels of magnification, namely 4x, 10x, 20x and 40x. the grading of each of the images are based on Bloom Richardson histological grading which involves the evaluation of the amount of tubule formation, the extent of nuclear pleomorphism as well as the mitotic count. Based on each one of these listed variables they are given a score between 1 and 3 and then will receive an overall grade based on those scores. Scores between 3-5 will receive a grade of 1, scores between 6 and 7 will receive a grade of 2 while scores between 8 and 9 will receive a grade of 3. The grade is related to how differentiated the cells are with grade 1 being well differentiated, stage 2 being moderately differentiated and stage 3 being poorly differentiated.

Grade 1 has a total of 259 images, Grade 2 has a total of 366 while Grade 3 has a total of 297 images. This will serve as a challenge as the data will be uneven making training harder as the model will have an uneven number of examples per grade.

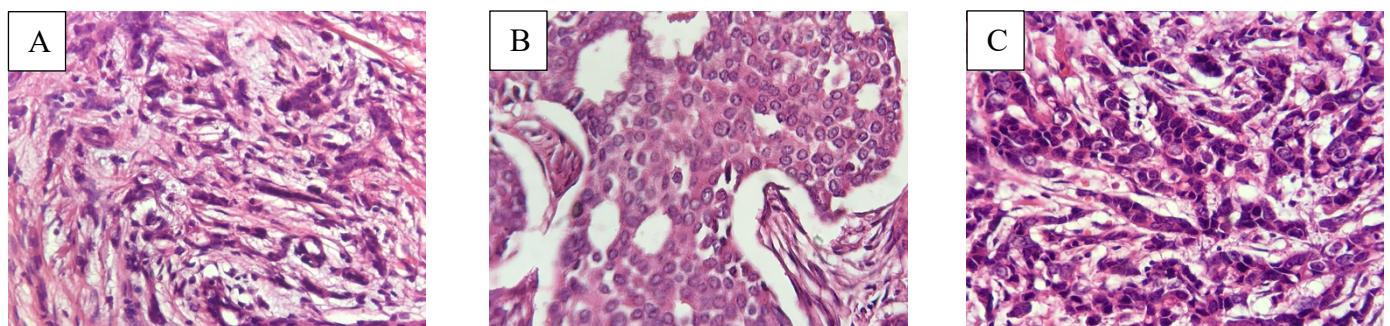


Figure 17: Histological classification of Breast Cancer grades 1(A),2(B),3(C) at magnification of 40x

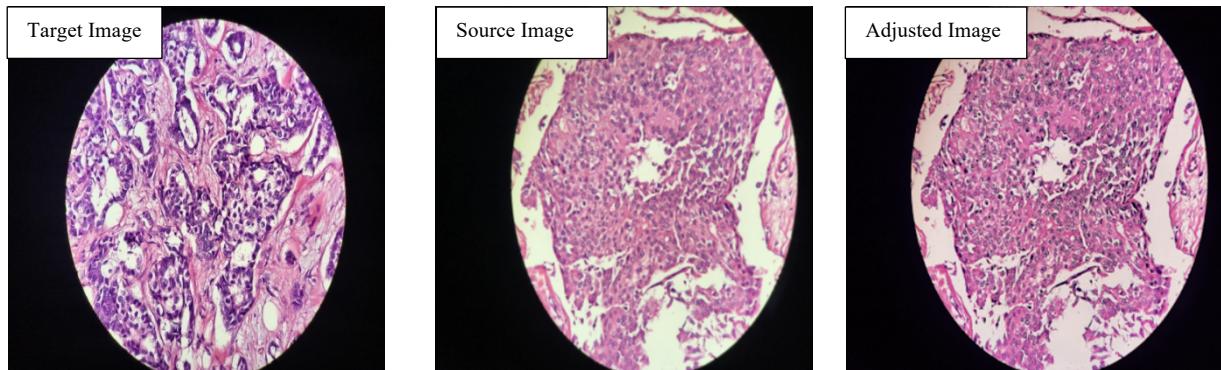
### Data Pre-Processing

During the pre-processing and normalisation of the data, I have rescaled all the images to [0,1]. This was done to normalise all the pixels of the images getting them on the same scale and in a consistent range which will help in channel extraction and modifications to these channels. This will also help when adding the images into the model as there won't be any sort of gradient descent allow the model to converge quicker as well as allowing stabilisation of the training. The consistency in the pixels will also allow for the colours to be adapted within the dataset again by keeping all the pixels within the same range allowing all of them to be adapted equally which will be touched on in the next step of pre-processing. This was done by dividing the pixel values of the image by 255 this will get all the images in the same range.

Next, after many different attempts at trying to correct the differences in colour throughout the dataset, I had come across a technique called histogram matching. This

has been expanded on in the literature review but to summarise, it will take a target image which is set, split this target image into HED channels and get the assigned values. It will then take the source image (the image needing to be changed) and split this into HED channels, and get the assigned values of each channel. Lastly it will take the values of the target image and apply these to the source image matching the images in colour. This will allow for all the images to be on the same colour scale with the pinks and blue/ purple of the haematoxylin and eosin to match throughout.

When adding the target image to the source image, it had come apparent that it began to put a pink filter over the white of the image in the regions that have not been stained, to counter act this I had implemented a mask over the white part of the images so they would not be impacted by this. This was done by creating a grey scale image and then setting the threshold of the mask to these regions so that they would not be affected by the histogram matching. Lastly the intensities of each channels were intensified to make sure that the pink and purple colours were clearly distinguishable in the image making it easier for the model to be able to pick up on the clear differences between colours and intensify the patterns seen. The eosin channel was adjusted to 1.5x the current intensity while the haematoxylin channel was adjusted to 1.1x the current intensity giving the adjusted image a greater difference in the colours creating a better contrast between the cellular components.



*Figure 18: collection of images showing how the source image was adjusted to the scales of the target image*

When looking at figure 18, it is clear that the colours have been adjusted to match the colours of the target image with a greater intensity seen in the pink and purple colours of the H&E images creating a larger contrast between these colours making it easier for the model to be able to pick up on differences in components as well as the patterns. Lastly, when inspecting the dataset the a lot of noise was seen in many of the images. The image was then sharpened to remove the noise and make the cellular components easy to see as many of them were missed due to the blurring around the outer edges of many of the images. This was done by using the Gaussian Blur feature to create a blur on the image removing noise and then removing this from the original image creating the sharpened image. The place holder for this is alpha which is set to 3, increasing the sharpness created by a value of 3, allowing the components of the image to stand out significantly while also removing the blur and noise seen in many of the images (refer to appendix for code).

Data augmentation was used to improve the images of the dataset and make the model adapt to different versions of the image within the model. This will allow the model to become more robust to different conditions as well as reducing bias in the model. This will also allow the image to pick up on more features and adapt to them, thus allowing the model to pick up on more patterns as more features will become distinguishable. It was created using image data generator which is part of the TensorFlow Keras

processing module and will be stored as datagen. Datagen was created (refer to appendix for code) the following parameters, a rotation range of 40, meaning it will rotate the image over a range of -40 to 40 degrees, allowing the model to learn the images from a different angle which is necessary in this case as they are a few duplicate images in the dataset so it will be able see each duplicate from a different angle not recognising it as the same image. Wide shift range as well as the height shift range was set to 0.2, allowing the image to move to left and right and up and down up to a total of 20%, giving the model a different perspective of the image allowing it to become more robust when training the data as it will pick up on different variations of the image. The shear range was set to 0.2 giving the object a slant of up to 0.2 radians. The zoom range was set between 1 and 1.2 allowing the image to be zoomed in a maximum of 20% and lastly, horizontal flip was set to true this will allow the image to be flipped horizontally giving the model new perspectives on duplicate images and not overfitting because of the duplicates. I had also added rescaling to the images to make sure that they were all in the same range before inputting the testing data into the model.

Before this, the data was split into 80% for training and 20% for testing, the testing was further split by 50% allowing for 10% to be used as validation data and 10% to be used for testing data. The random state was set to 36 instead of the standard 42 in most papers as this seemed to work better when testing the model (refer to appendix for code). Datagen that was created will only be used on the training data as this will make the model more robust to real world scenarios when variability of the images will change and make the model work better on new never before seen data because of the variability created by the data augmentation. By also creating this variation in the data the model should become more prone to overfitting as it will not be able to memorise the images as there are multiple different variations of the image.

## Convolutional Neural Network (CNN)

When looking into CNN models, I learnt that using a pre trained model would be best for this case as building one from scratch is very time consuming and computing resources extensive meaning it will take a lot more resources as well as skill for building architecture to be able to process and train this large amount of images with a high level of complexity associated to each one of the images. Another positive of using a pre-trained that when they were created they were made to recognise shapes, textures as well as more complex features that you would expect from a model built from scratch. The pre trained models are built of many layers but the last layer is left for fine tuning of your specific dataset, with this in mind it leads way for the model to be able to converge quicker than it would if a model built from scratch was used as it is adapted to the specific task at hand.

I have made use of 5 different pre-trained models to be able to test the difference between many variables such as the training time, the accuracy and loss of both the test and training data as well as the f1 score of each model. Through research done I had seen that there is a difference in using model and sequential API, the model API allows for more complex architectures while sequential is used for more linear architectures, model API also is more flexible in using custom architectures while Sequential is more straight forward and uses a linear stack of layers (Cloud, 2023). This again is down to trial and error as to which API will work better, both have been tested and results are available in the results section of the report. I have made use of VGG-16, InceptionV3, ResNet50, DenseNet121 and EfficientNet B3. I have made use of the same final layer of the model to test the performance of the models without changing the variables to be able to get a fair and reliable reading of the pre-trained models (Refer to Figure 19).

```

for layer in base_model.layers:
    layer.trainable = False

VGGmodel = Sequential()
VGGmodel.add(base_model)
VGGmodel.add(GlobalAveragePooling2D())
VGGmodel.add(Dense(512, activation='relu'))
VGGmodel.add(BatchNormalization())
VGGmodel.add(Dropout(0.5))
VGGmodel.add(Dense(3, activation='softmax'))

VGGmodel.summary()

```

Figure 19: Image showing the final layer of the model

Referring to figure 19, this is the final layer of the model for all the models used in the project. Global Average Pooling 2D is used as it reduces the spatial dimensions of the feature maps created to a single value per channel. This acts as a regulariser, it is used as it prevents overfitting in the model and according to research is a good alternate to fully connected layers. In the dense layer, 512 which is number of neurons used in each of the layers, each one of these neurons will take input from a previous layer and will produce an output accordingly. This number of neurons seemed to do well when trialled and tested in the model, other options were 1024 and 256 but it seemed to overfit and underfit respectively.

The activation of ReLU is used during this step.

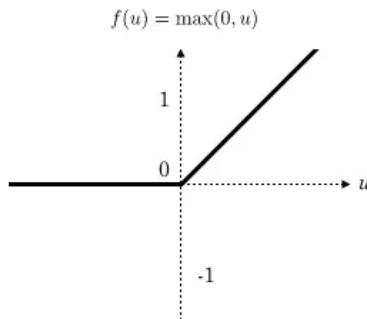


Figure 20: Graph representing how ReLU works

Referring to figure 20, it is used to introduce non-linearity to the model, this gives it the advantage to learn complex patterns in the images. To explain the graph ReLU will return the input value as long as it is greater than or equal to 0, if it is less than 0 it will return 0 thus taking away the linearity which would make it harder to learn more complex patterns and potentially allow the model to overfit (Deelaka, 2021). This can be further explained by a simple equation as seen at the top of figure 20,  $f(x) = \max(0, x)$ ,  $x$  is the input to the neuron, while  $f(x)$  is the output of the neuron when ReLU is used. Meaning that whatever is greater between 0 and  $x$  is what the output of the neuron will be. Batch normalisation was then used to normalise the feature map that will be received before moving it to the next layer of the model which should work in increasing the convergence rate as well as reducing the chances of overfitting. This works by effectively changes the mean and standard deviation of the pixels in all feature maps. It begins by z-score normalising all pixels, then multiplies the normalised values by an arbitrary parameter alpha (scale), followed by another arbitrary parameter beta (offset) (Olu-Ipinlaye, 2022). This can be shown by the following set of equations.

1	$\mu = \frac{1}{m} \sum h_i$	2	$\sigma = \left[ \frac{1}{m} \sum (h_i - \mu)^2 \right]^{1/2}$	3	$h_{i(\text{norm})} = \frac{(h_i - \mu)}{\sigma + \epsilon}$	4	$h_i = \gamma h_{i(\text{norm})} + \beta$
---	------------------------------	---	--	---	--	---	---

Figure 21: Image showing the equations for batch normalisation

Referring to figure 21, step 1 refers to the calculation needed to calculate the mean of the hidden activation, in this case m is the number of neurons at level h. step 2 is used to calculate the standard deviation of the hidden activations. In step 3, it is used to normalise the hidden activations,  $\epsilon$  is added as a soothing term which assures numerical stability within the operation which does not allow for division by a 0 value. The final step, step 4 is used to rescale and offset the input. Gamma and beta are used to rescale (gamma) and shift (beta) the vector which contain previous operations values. These are both learnable parameters and during training optimal values will be used to ensure the accuracy of normalisation of each batch (Saxena, 2021).

A dropout rate is used in the next layer which is set to 0.5 which will then drop out 50% of the neurons this is step included to prevent overfitting of the model.

In the last layer a dense layer is created, it is set at 3 as this is the number of classes in the dataset (stage 1, stage 2 or stage 3), this will drop the number of neurons down to 3. The SoftMax activation is used to transform logits into probabilities that of which class it will belong to. This can be shown by the following calculation.

$$S(y)_i = \frac{\exp(y_i)}{\sum_{j=1}^n \exp(y_j)}$$

Figure 22: SoftMax calculation

In figure 22, y is an input vector to the SoftMax function (S), this will consist of n elements for n classes which are the possible outcomes.  $y_i$  is the number of the input vector,  $\exp(y_i)$  is the standard exponential function applied on  $y_i$ . The whole factor under the division line is a normalisation term which will ensure that values of the output vector (SoftMax function) equals to 1 for the number of class of each of them. Every one of them will be in the range from 0-1 which allows the probability distribution to be valid (Koech, 2021).

As previously stated, this model will be used throughout all the pre-trained models as the final layer for fine tuning. In the final two lines of the code, compile and fit, in the compile of the model the Adam optimiser was used at a learning rate of 0.0001, this typically at the range of 0 to 1 with 0.0001 being 0.1% at the gradient of the regular learning rate. This is done to prevent over fitting, this was found using trial and error and drastically slowed down the convergence time but with that came better accuracy and that implements that trade off that was spoken about previously. The loss function was set to sparse categorical cross-entropy which is used when the classes of the data is mutually exclusive and over 2 classes. SoftMax will measure the difference between the predicted probability distribution and the actual probability distribution and how well they match each other. The loss function can be explained using the following calculation.

Referring to figure 23, n is the number of classes in this case is 3. The  $t_i$  is the truth label  $p_i$  is the SoftMax probability for  $i^{\text{th}}$  class (Rahman, 2023).

$$L_{\text{CE}} = - \sum_{i=1}^n t_i \log(p_i)$$

*Figure 23: Equation showing the loss function*

As images are needed to be in different sizes according to the pre-trained model, the image will imported using a function I have created a function called “load images from folder” which will load the images from the target folder at the desired size such as 224x224 for VGG-16 and 299x299 for InceptionV3 before assigning labels to each class and loading the data into the training, test and validation split.

The F1 score was added to the end of the model, this is implemented to be able to assess the accuracy of the model, this works by combining the precision and recall of the model. This in simple terms will calculate how many times the model made a correct prediction across all the classes of the dataset. This can be explained by the following set of calculations.

$$\text{F1 Score} = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

*Figure 24: Equation used to determine the F1 score of the model*

In this equation, TP stands for the true positive determined by the model, FP stands for the false positive and FN stands for the false negative of the model (Kundu, 2022).

## Results

The results differed through all the models with many of them converging quick and performing better over 10 epochs instead of a larger amount of epochs, this could be due to overfitting but will be touched on further in this section. As previously mentioned both model and sequential API were used to test the difference as to which API worked better.

Firstly, I will touch on the use of the model API then proceeding with the Sequential API.

*Table 3: Table representing the models used, using the model API*

Model	Epochs	Training time (In min)	F1 Score	Training Loss	Test Loss	Training Accuracy	Test Accuracy
<b>VGG16</b>	10	21:41	0.6910	0.6418	0.89	72.2%	70.82%
	15	32:24	0.6472	0.58	0.93	75.85%	66.52%
	20	32:48	0.5738	0.52	1.1141	79.42%	61.73%
<b>Inception V3</b>	10	4.51	0.5011	0.97	1.11	59.16%	52.31%
	15	7:11	0.5552	0.92	1.08	58.21%	55.54%
	20	10:24	0.6032	1.09	0.818	76.47%	61.02%
<b>ResNet50</b>	10	7:27	0.5815	0.730	0.879	69.20%	57.69%
	15	12:31	0.6077	0.6459	1.05	73.68%	60.99%
	20	15:57	0.6363	0.5274	0.995	78.55%	64.29%
<b>DenseNet121</b>	10	16:26	0.5790	0.8286	0.84	65.26%	58.91%
	15	21:55	0.6444	0.7754	0.89	66.08%	65.37%
	20	22:40	0.5790	0.6489	0.957	71.64%	57.84%
<b>EffecientNetB3</b>	10	8:47	0.6524	0.895	0.807	60.11%	65.44%
	15	14:10	0.5709	0.7963	0.9558	65.81%	57.76%
	20	17:48	0.5492	0.7169	1.0181	69.61%	54.61%

When looking at table 3, models such as VGG-16, DenseNet121 and EfficientNet B3, these models unlike the others saw a drop in test accuracy as the epochs were increased but saw a rise in training accuracy as the epochs were increased which is usually seen when working with CNN models. This shows that the model began to overfit as the epochs were increased causing the training data to be well recognised but unable to transfer that understanding onto the test data causing the training accuracy to continue to rise while the test data either decreases or remains the same.

Inception V3 and ResNet50 both increased in training and testing accuracy as the epochs increased which is usually seen in other models that have been researched.

The loss seen throughout all the models used has been fairly poor. Many added techniques have been tried and used to try and combat this high loss such as decreasing the learning rate, simplifying the model as well as using batch normalisation. Data augmentation and adding pre-processing units which are specific to each model were used to add to the data before inputting it into the model to try and bring down the loss of each model which were successful but not successful enough to get the loss down to a rate that is satisfactory. The F1 score was implemented to be able to assess the accuracy of the model, this works by combining the precision and recall of the model. This in simple terms will calculate how many times the model made a correct prediction across all the classes of the dataset. This is not an ideal metric as the classes are unbalanced with some classes having more data than others but still gives a good idea into the accuracy of the model and should co-inhere to the accuracy of the model using the test data. All the results seen in table 4 and 5 were created using the Adam optimiser at a learning rate of 0.0001 and sparse categorical cross-entropy as the loss function.

*Table 4: Table representing the models used, using the Sequential API*

Model	Epochs	Training time (In min)	F1 Score	Training Loss	Test Loss	Training Accuracy	Test Accuracy
<b>VGG16</b>	10	18:36	0.6394	0.6077	1.04	74.5%	64.22%
	15	27:54	0.6135	0.5240	0.9397	78.02%	62.82%
	20	31:21	0.5922	0.5631	0.981	79.61%	59.37%
<b>InceptionV3</b>	10	3:41	0.5497	0.7129	0.9336	70.83%	56.69%
	15	5:43	0.5736	0.5437	1.093	76.26%	57.76%
	20	7:47	0.5880	0.430	1.075	82.9%	58.84%
<b>ResNet50</b>	10	7:54	0.5525	0.7471	0.9126	66.49%	55.69%
	15	13:35	0.5908	0.6098	1.0042	74.63%	59.91%
	20	16:42	0.5351	0.4945	1.004	79.92%	64.22%

<b>DenseNet121</b>	10	8:14	0.647 6	0.8286	1.18	71.78%	67.52%
	15	11:08	0.589 0	0.552	1.08	76.53%	59.99%
	20	16:38	0.612 8	0.9115	1.20	80.46%	61.80%
<b>EffecientNetB3</b>	10	8:50	0.615 5	0.9465	0.925 7	58.89%	62.06%
	15	12:38	0.517 5	0.7913	1.053	65.4%	52.39%
	20	17:52	0.581 0	0.7618	0.939 4	68.25%	57.76%

Referring to table 4, the main purpose of inputting a sequential API was to try increase the simplicity of the model (refer to literature review) which in essence could potentially combat the loss of the model. Looking at the results it was not successful in this but was successful in reducing the training time across all of the epochs while keeping a similar accuracy to what was seen in the Model API, it was even more successful in doing so in the DenseNet121 model which is experienced a higher testing accuracy but with this came a greater difference between the training and testing accuracy as well as a larger loss seen than with using the Model API which again hints at overfitting being the main issue this model is not preforming to the level it should be. This will be further discussed in the discussion.

Table 5: Table showing the best performing models

Model	Epochs	Training time (In min)	F1 Score	Training Loss	Test Loss	Training Accuracy	Test Accuracy
<b>VGG-16 (M)</b>	10	21:41	0.6910	0.6418	0.89	72.2%	70.82%
<b>InceptionV3 (M)</b>	20	10:24	0.6032	1.09	0.818	76.47%	61.02%
<b>ResNet50 (M)</b>	20	15:57	0.6363	0.5274	0.995	78.55%	64.29%
<b>DenseNet121 (S)</b>	10	8:14	0.6476	0.8286	1.18	71.78%	67.52%
<b>EffecientNetB3 (M)</b>	10	8:47	0.6524	0.895	0.807	60.11%	65.44%

Table 5 shows the best performance of each of the models, this is done to summarise all the data in tables 3 and 4 and give a clear indication as to which model performed best. The model name will either be followed by an "M" or "S" indicating that the model either used model or sequential API. The best model was the VGG-16 model using the model API (M), this model received the highest test accuracy with 70.82% as well as a training accuracy of a similar value indicating that overfitting wasn't as much as a factor as it was in other models. The second best model performance was EffecientNet B3, this received the lowest loss of all the models at 0.807 with the third highest accuracy of 65.44% the significance here is that the training accuracy was lower than the testing accuracy which could potentially be signs of underfitting as well as having the second highest f1 score showing that the model worked well overall in comparison to others. The training time is also the second lowest of all the models showing that it picks up components and patterns within the image fairly accurately in a short amount of time

where as models such as VGG-16 took 21 minutes and 41 seconds to achieve slightly better results.

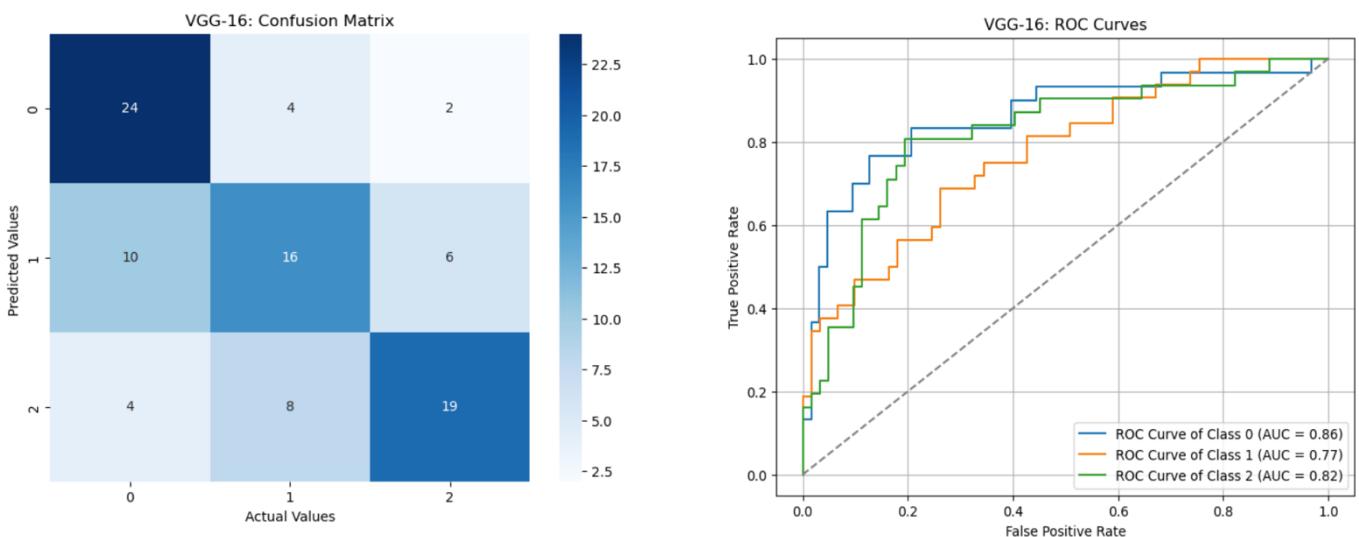
*Table 6: Table showing the total number of parameters of the models*

Model	Number of parameters
VGG-16	14,980,931
InceptionV3	22,855,459
ResNet50	24,640,387
DenseNet121	7,565,891
EffecientNetB3	11,574,066

Table 6 shows the number of parameters that each model will go through when completing training of the images. The parameters should increase with the training time with logical thinking but this is not the case when referring back to table 3, looking at the 10 epochs of each model, logically ResNet50 should have taken the longest but took just over 7 minutes to run through 24.64 million parameters while VGG-16 took over 21 minutes to run through almost 15 million parameters. This is all down to the layering as well as the techniques specific to each model that have implemented(further expanded on in the literature review).

The following figures containing the confusion matrix and ROC curves will be touched on throughout the discussion of the report. The confusion matrix of the models are used as a measurement of performance of the models. The confusion matrix will compare actual values obtained to the predicted values. As seen in figure 25, for stage 1 (0), the model predicted 24 of the test images correctly while 10 of the images were predicted as stage 2 (1) and another 4 were predicted as stage 3 (2).

The ROC curve is used again to measure performance of the model. The ROC is a probability curve where AUC (area under curve) is the measurement of separability. The closer to 1 the curve is the better the model is predicting the class as the class it should be , referring to figure 25, the curve of class 0 (stage 1) has a AUC of 0.86 which is good level of separability. An AUC reading of 0.5 will mean that it does not have class separation capability (Narkhede, 2018).



*Figure 25: VGG-16 Confusion Matrix and ROC curve*

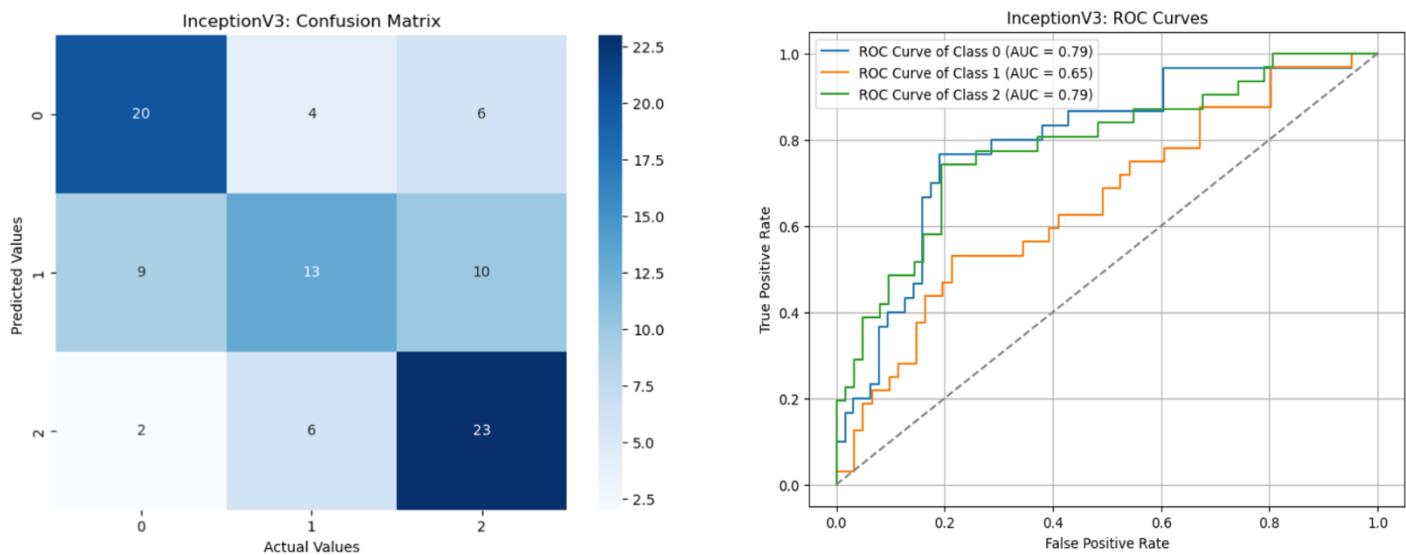


Figure 26: Inception V3: Confusion Matrix and ROC Curve

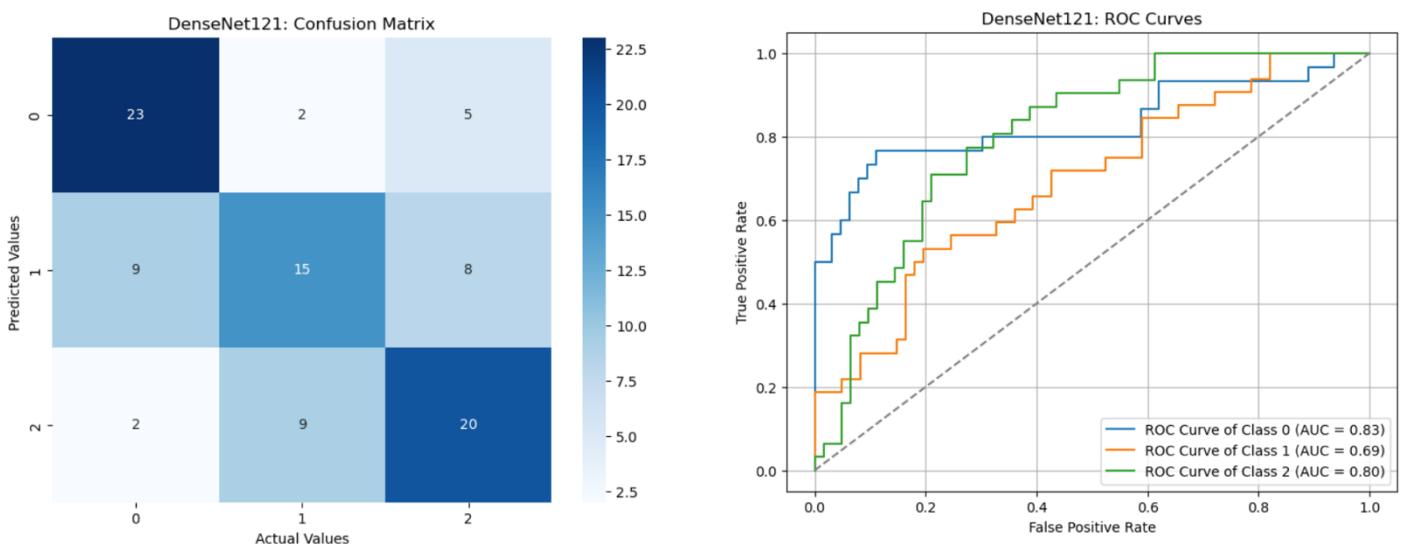


Figure 27: DenseNet121: Confusion Matrix and ROC Curve

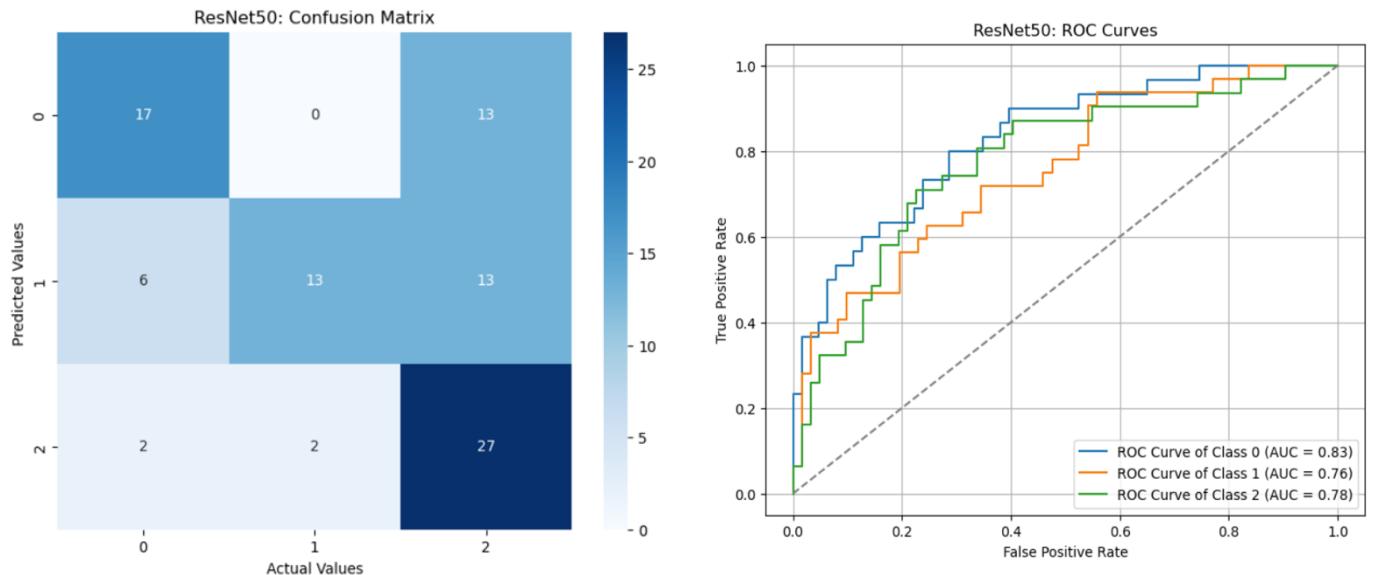


Figure 29: ResNet50: Confusion Matrix and ROC Curve

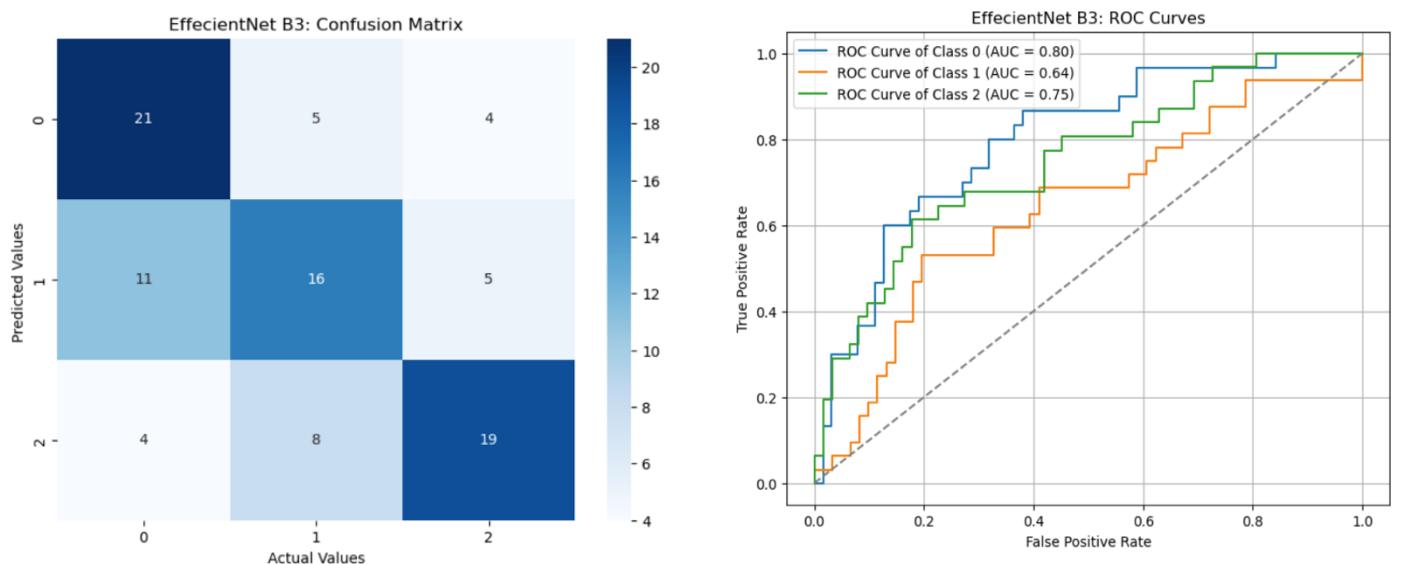


Figure 28: EffecientNet B3: Confusion Matrix and ROC Curve

## Discussion

The aim of the project was to be able to use machine learning to be able to predict the stage of breast cancer from H&E stained images. As this has worked to a degree, there is room for improvement to be able to get a higher test accuracy as well as in lowering the loss of the model.

During pre-processing and normalisation of the images I encountered many challenges which did take me additional time to get right and overcome these issues. Many of techniques suggested and researched just did not seem to work such as histogram equalisation as well as balancing the HSV (Hue, Saturation and Value) of all the images using a technique found in a research paper. After many attempts of finding a technique to work, I had eventually come across a paper suggesting the use of a target image to normalise all the images into the same colour ranges. A technique of splitting the channels specifically for H&E stains was used to firstly split the colours of the image into their respected colour channels once this was obtained, the target image was split into the same H&E channels using the same technique. This then allowed for the target images channels to be adjusted to the source image producing an adjusted image with the same or very similar colour ranges to the target image. These images were then enhanced and rescaled to make sure that the channels (H&E components) were highlighted thus driving a large contrast between the colour channels allowing the model to easily distinguish between these channels, picking up patterns and cellular components. These images were then sharpened to remove noise and inconstancies in the images and really allowing the images to portray their patterns and cellular components. This technique when inspecting the images seemed to have worked well for majority of the images, I have seen that many of the images in stage 2 did not respond as well as the images in stages 3 and 4 did which can be seen in the results. This lead to the overall decrease in accuracy and increase of loss of the model. I believe that not getting this section of the project perfect had a negative effect of the overall outcome of the model and the project. With this came many positives as I have learnt many new ways to normalise images that will help me when I am able to implement these techniques in the future. This step was challenging, more I thought it would have been, it was under estimated and more time should have been allocated to this when working out the time plan for the project. Even though it was not as good as it could have been, I believe that it still produced decent results and showed out of the box thinking to be able to implement and adapt a technique that is not used as much as other techniques are.

After a lot of trial and error while experimenting with many pre-trained models, I believe I eventually came to a well-made model that is able to process and categorise the images in the dataset in a satisfactory manner. With the use of data augmentation and pre-processing units, I believe I have created enough of a real-world scenario allowing the images to be trained at different variations as it would be in the real world outside of a controlled dataset and then applying what the model has learnt to the validation and testing data. The results were obtained were fair and mainly down to the pre-processing and normalisation not being as good as it could have been. With the VGG-16 model performing the best out of all the pre-trained models with an accuracy of over 70% on the test data and a f1 score of over 0.69. This is a fair result but there was definitely room for improvement. EfficientNet B3 performed second best with an accuracy of over 65% on the test data and an F1 score of over 0.65, while training the data in just under a third of the time that it took the VGG-16 model, which is extremely efficient when working with large datasets saving time and resources. When referring to figure 25, majority of the images had been classified correctly, with stage 1 (0) and stage 3 (2) being the most accurately assigned within the dataset, this is apparent through all the models confusion matrixes (Figure 25 - 29), there is a clear pattern in these figures, all showing that stage 2 (1) was assigned the least accurately through all the classes in the dataset. As stage 2 of breast cancer is the mid-point between stage 1 and stage 3, the

images in this class will share similar features with the others which may have caused the model to get confused and interpret this accordingly. When looking at the ROC curves of the pre-trained models (Figure 25 - 29), there is another pattern in the AUC of the classes that is the same throughout all of the models. This pattern is seen in that stage 2 performed the worst and had the lowest AUC of all of the stages. This again can be down to normalisation of the image as I had mentioned earlier that the stage 2 images did not respond as well to the adjustments in colour as the other stages did and also that stage 2 shares many similarities of stage 1 and stage 3 as it is the midpoint between the stages. Stage 1 had the highest AUC which was 0.86 in the VGG-16 model which is very good, while its lowest AUC was seen at 0.79 in the Inception V3 model (Figure 26). This is still very good and shows that all the models were able to identify stage 1 accurately. Stage 2 received its highest AUC of 0.77 in the VGG-16 model (Figure 25), while it received the lowest AUC of all the models in EfficientNet B3 model of 0.64 (Figure 34). Stage 3 classification was accurate receiving an AUC of 0.82 using the VGG-16 model (Figure 25) and its lowest of 0.75 using the EfficientNet B3 model (Figure 29).

The average accuracy and high loss of the model are bought down to two reasons, these being overfitting of the model and not enough pre-processing and normalisation of the images before inputting it into the model. The learning rate was set to 0.0001 to try and combat this, which it did as the loss was much higher before doing so, RMSProp was also trialled as the optimiser which did not have a positive impact on the loss or accuracy of the model. L1 and L2 optimisers were also fit into the model but again did not have a positive effect on the final accuracy or loss of the model. The complexity of the model was also increased adding more dense and dropout layers but no positive effect was seen.

Overall the model performed well but there is room for improvement, which is down to the pre-processing and normalisation of the images. These improvements that could be made to this step will allow for a decrease in loss as well as an increase in accuracy in all of the models creating greater more sought after results.

## Conclusion and Future Work

When taking the model as a whole, I believe that It performed well but there is room for improvement. These improvements will need to be made in the pre-processing and normalisation of the images before inputting the data into the model as well as in combatting the overfitting of the model as epochs increased. There was a definite best performer out of all the models and this was the VGG-16 model, this clearly seen when looking at the overall accuracy, f1 score, confusion matrix and the ROC curves which outperformed all of the other models used. The only downside when using the VGG-16 model was the time it took to train the data with it being far greater than all the other training times but as mentioned before this is one of the trade-offs that is expected, also with keeping in mind that it is running through just under 15 million parameters meaning that training time is expected to be high. With improvements to the pre-processing and normalisation of the data, I believe that the VGG-16 model will be improved drastically and would achieve the sought after results that is needed to use this in the medical field. This work in the future will definitely aid the medical sector by providing quick and accurate results but more work is needed before it is ready.

## References

- Alanazi, S.A., Kamruzzaman, M.M., Islam Sarker, M.N., Alruwaili, M., Alhwaiti, Y., Alshammari, N. and Siddiqi, M.H. (2021). Boosting Breast Cancer Detection Using Convolutional Neural Network. *Journal of Healthcare Engineering*, 2021, pp.1–11. doi:<https://doi.org/10.1155/2021/5528622>.
- Baldha, S. (2022). *Introduction to DenseNets (Dense CNN)*. [online] Analytics Vidhya. Available at: [https://www.analyticsvidhya.com/blog/2022/03/introduction-to-densenets-dense-cnn/?utm\\_source=reading\\_list&utm\\_medium=https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/](https://www.analyticsvidhya.com/blog/2022/03/introduction-to-densenets-dense-cnn/?utm_source=reading_list&utm_medium=https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/) [Accessed 28 Aug. 2023].
- Bankhead, P. (2022). *Python: Channels & colors — Introduction to Bioimage Analysis*. [online] bioimagebook.github.io. Available at: <https://bioimagebook.github.io/chapters/1-concepts/4-colors/python.html> [Accessed 27 Aug. 2023].
- Boesch, G. (2021). *VGG Very Deep Convolutional Networks (VGGNet) - What you need to know*. [online] viso.ai. Available at: <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/>.
- Brownlee, J. (2019). *How to Configure the Learning Rate Hyperparameter When Training Deep Learning Neural Networks*. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/>.
- Cancer Research UK (2023). *Ductal carcinoma in situ (DCIS)*. [online] www.cancerresearchuk.org. Available at: <https://www.cancerresearchuk.org/about-cancer/breast-cancer/types/ductal-carcinoma-in-situ-dcis>.
- Cloud, S. (2023). *Keras: Why Do Sequential and Model Give Different Outputs? | Saturn Cloud Blog*. [online] saturncloud.io. Available at: <https://saturncloud.io/blog/keras-why-do-sequential-and-model-give-different-outputs/> [Accessed 29 Aug. 2023].
- deelaka, nipun (2021). *How ReLU works?* [online] Analytics Vidhya. Available at: <https://medium.com/analytics-vidhya/how-relu-works-f317a947bdc6>.
- DePolo, J. (2023). *Invasive Ductal Carcinoma (IDC)*. [online] www.breastcancer.org. Available at: <https://www.breastcancer.org/types/invasive-ductal-carcinoma> [Accessed 20 Aug. 2023].

Feldman, A.T. and Wolfe, D. (2014). Tissue processing and hematoxylin and eosin staining. *Methods in molecular biology* (Clifton, N.J.), [online] 1180, pp.31–43. doi:[https://doi.org/10.1007/978-1-4939-1050-2\\_3](https://doi.org/10.1007/978-1-4939-1050-2_3).

Google Cloud Team (2019). *Advanced Guide to Inception v3 on Cloud TPU | Cloud TPU | Google Cloud*. [online] Google Cloud. Available at: <https://cloud.google.com/tpu/docs/inception-v3-advanced>.

Huilgol, P. (2020). *Top 4 Pre-Trained Models for Image Classification with Python Code*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/#:~:text=The%20VGG%2D16%20is%20one> [Accessed 27 Aug. 2023].

isitapol2002 (2022). *Histogram matching with OpenCV, scikit-image, and Python*. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/histogram-matching-with-opencv-scikit-image-and-python/>.

Koech, K.E. (2021). *Softmax Activation Function — How It Actually Works*. [online] Medium. Available at: <https://towardsdatascience.com/softmax-activation-function-how-it-actually-works-d292d335bd78#:~:text=Softmax%20is%20an%20activation%20function> [Accessed 29 Aug. 2023].

Kumaraswamy, E., Kumar, S. and Sharma, M. (2023). An Invasive Ductal Carcinomas Breast Cancer Grade Classification Using an Ensemble of Convolutional Neural Networks. *Diagnostics*, [online] 13(11), p.1977. doi:<https://doi.org/10.3390/diagnostics13111977>.

Kundu, R. (2022). *F1 Score in Machine Learning: Intro & Calculation*. [online] www.v7labs.com. Available at: <https://www.v7labs.com/blog/f1-score-guide>.

Kurama, V. (2020). *A Guide to ResNet, Inception v3, and SqueezeNet*. [online] Paperspace Blog. Available at: <https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezezenet/>.

Mahato, A. (2023). *Getting started with Image Processing Using OpenCV*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2023/03/getting-started-with-image-processing-using-opencv#:~:text=Some%20powerful%20image%20preprocessing%20techniques> [Accessed 27 Aug. 2023].

McDermott, J. (2020). *Hands-on Transfer Learning with Keras and the VGG16 Model*. [online] www.learndatasci.com. Available at: <https://www.learndatasci.com/tutorials/hands-on-transfer-learning-keras/>.

Mishra, P. (2019). *Why are Convolutional Neural Networks good for image classification?* [online] Medium. Available at: <https://medium.datadriveninvestor.com/why-are-convolutional-neural-networks-good-for-image-classification-146ec6e865e8>.

MLK (2020). *Tutorial - numpy.zeros() , numpy.ones() and numpy.eye() in Python*. [online] MLK - Machine Learning Knowledge. Available at: [https://machinelearningknowledge.ai/numpy-zeros-numpy-ones-and-numpy-eye-in-python/?utm\\_content=cmp-true](https://machinelearningknowledge.ai/numpy-zeros-numpy-ones-and-numpy-eye-in-python/?utm_content=cmp-true) [Accessed 31 Aug. 2023].

Narkhede, S. (2018a). *Understanding AUC - ROC Curve*. [online] Medium. Available at: <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>.

Narkhede, S. (2018b). *Understanding Confusion Matrix*. [online] Medium. Available at: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>.

Olu-Ipinlaye, O. (2022). *Batch Normalization in Convolutional Neural Networks*. [online] Paperspace Blog. Available at: <https://blog.paperspace.com/batch-normalization-in-convolutional-neural-networks/> [Accessed 29 Aug. 2023].

OncoLink Team (2019). *Nottingham Score for Breast Cancer | OncoLink*. [online] Oncolink.org. Available at: <https://www.oncolink.org/frequently-asked-questions/cancers/nottingham-score-for-breast-cancer> [Accessed 26 Aug. 2023].

Paul, S. (2021). *Keras documentation: Learning to Resize in Computer Vision*. [online] keras.io. Available at: [https://keras.io/examples/vision/learnable\\_resizer/](https://keras.io/examples/vision/learnable_resizer/).

Rahman, M. (2023). *What You Need to Know about Sparse Categorical Cross Entropy*. [online] Medium. Available at: <https://rmoklesur.medium.com/what-you-need-to-know-about-sparse-categorical-cross-entropy-9f07497e3a6f#:~:text=Sparse%20categorical%20cross%20entropy%20is%20an%20extension%20of%20the%20categorical> [Accessed 29 Aug. 2023].

Rosebrock, A. (2021). *Histogram matching with OpenCV, scikit-image, and Python*. [online] PyImageSearch. Available at: <https://pyimagesearch.com/2021/02/08/histogram-matching-with-opencv-scikit-image-and-python/> [Accessed 27 Aug. 2023].

Sampias, C. and Rolls, G. (n.d.). *An Intro to H&E Staining: Protocol, Best Practices, Steps & More*. [online] www.leicabiosystems.com. Available at:

<https://www.leicabiosystems.com/en-gb/knowledge-pathway/he-staining-overview-a-guide-to-best-practices/#:~:text=The%20H%26E%20stain%20provides%20a> [Accessed 27 Aug. 2023].

Saxena, S. (2021). *Batch Normalization | What is Batch Normalization in Deep Learning*. [online] Analytics Vidhya. Available at:

<https://www.analyticsvidhya.com/blog/2021/03/introduction-to-batch-normalization/>.

scikit-learn (2019a). *sklearn.metrics.confusion\_matrix — scikit-learn 0.21.3 documentation*. [online] Scikit-learn.org. Available at: [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion\\_matrix.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html).

scikit-learn (2019b). *sklearn.metrics.f1\_score — scikit-learn 0.21.2 documentation*. [online] Scikit-learn.org. Available at: [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html).

scikit-learn (2019c). *sklearn.metrics.roc\_auc\_score — scikit-learn 0.23.2 documentation*. [online] scikit-learn.org. Available at: [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc\\_auc\\_score.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html).

scikit-learn (2019d). *sklearn.metrics.roc\_curve — scikit-learn 0.23.0 documentation*. [online] scikit-learn.org. Available at: [https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc\\_curve.html](https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html).

Smolarz, B., Nowak, A.Z. and Romanowicz, H. (2022). Breast Cancer—Epidemiology, Classification, Pathogenesis and Treatment (Review of Literature). *Cancers*, 14(10), p.2569. doi:<https://doi.org/10.3390/cancers14102569>.

Takahashi, H., Oshi, M., Asaoka, M., Yan, L., Endo, I. and Takabe, K. (2020). Molecular Biological Features of Nottingham Histological Grade 3 Breast Cancers. *Annals of Surgical Oncology*. doi:<https://doi.org/10.1245/s10434-020-08608-1>.

Talpur, S., Rashid, M., Khan, S.J. and Syed, S.A. (2022). Automatic Detection System to Identify Invasive Ductal Carcinoma by Predicting Bloom Richardson Grading from Histopathological Images. *Journal of Independent Studies and Research Computing*, 20(1). doi:<https://doi.org/10.31645/jisrc.22.20.1.6>.

Vellido, A. (2018). Societal Issues Concerning the Application of Artificial Intelligence in Medicine. *Kidney Diseases*, [online] 5(1), pp.11–17. doi:<https://doi.org/10.1159/000492428>.

Wolfram Neural Net Repository (2021). *DenseNet-121 - Wolfram Neural Net Repository*. [online] resources.wolframcloud.com. Available at:

<https://resources.wolframcloud.com/NeuralNetRepository/resources/DenseNet-121-Trained-on-ImageNet-Competition-Data/> [Accessed 28 Aug. 2023].