Breast Cancer Tumour Tissue Classification: A Deep Learning CNN Approach

Project Final Report

Authors: Chen Huijia (*e0053506@u.nus.edu*), Thangavel Sharan (*e0148276@u.nus.edu*), Khaw Yew Onn (*e0005008@u.nus.edu*), Huang Yida (*e0175270@u.nus.edu*), Jeffrey Tan Jian Sheng (*e0176250@u.nus.edu*), Low Si Jia Jessica (*e0035946@u.nus.edu*)

National University of Singapore

CS3244 Machine Learning

Abstract

Diagnosing breast cancer based on biopsy images can be challenging as malignant (cancerous) or benign (harmless) tumours are visually similar. In this study, we developed an image recognition model capable of classifying breast tumours into two categories: malignant and benign, based on microscopic images of extracted breast tumour tissues. We envision that the findings from this study could enhance medical capabilities in the correct diagnosis of breast cancer.

1 Introduction

1.1 Motivation

Breast cancer is the most common disease among women, with 2 million new cases in 2018 alone [1]. Classification into malignant or benign affects the treatment received by the patient, and a misdiagnosis could lead to harmful diagnostic errors [2]. In a study involving malpractice claims, 59% of diagnosis error is cancer-related, making cancer one of the more commonly misdiagnosed diseases [3]. Thus, by harnessing the potential of available data (>7000 tagged images) and relevant machine learning methods such as CNN, we aim to enhance existing capabilities in breast cancer tumour classification in the areas of efficiency and accuracy.

1.2 Guidelines

This study will examine the effectiveness of various Convolutional Neural Network (CNN) models and supervised learning algorithms for classification of breast tumours, malignant or benign, based on microscopic images of extracted breast tumour tissues.

Through this study, we hope to answer the following guiding questions:

1. Which model class/architecture can best evaluate the malignancy of tumours, based on computational and work requirements, precision and recall?
2. How can we evaluate the importance of hyper-parameters in the respective models?

For this self-initiated study, we will be working on breast cancer histopathological database from the Laboratory of Vision, Robotics and Imaging at the Federal University of Parana [4].

2 Dataset and Related Work

2.1 Dataset

We requested and obtained 7,909 microscopic images of breast tumour tissue that were collected from 82 patients by the Laboratory of Vision, Robotics and Imaging at the Federal University of Parana. This breast cancer histopathological database will be used for training & testing purposes. The data is divided into 4 magnifications.

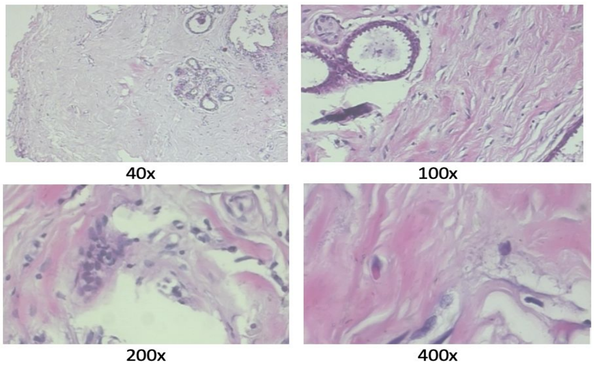
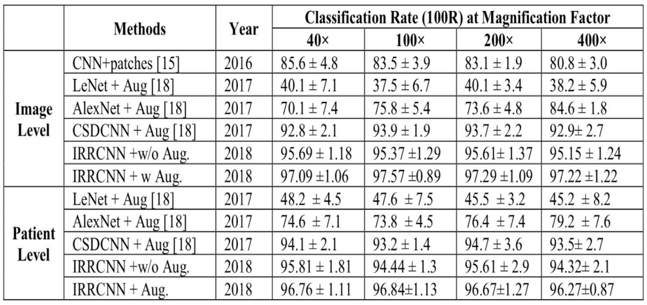


Fig 1: Example of a benign tumour sample with 40x, 100x, 200x & 400x magnification

2.2 Related Work

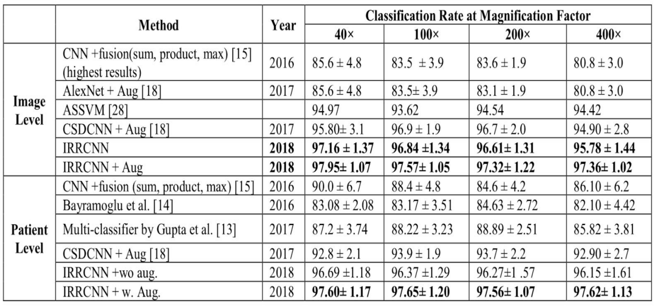
Recently, a study by Alom and his team proposed a method for breast cancer classification - Inception Recurrent Residual Convolutional Neural Network (IRRCNN), which was claimed to be a powerful model combining Inception Network (Inception-v4), Residual Network (ResNet) and Recurrent Convolutional Neural Network (RCNN) [5]. Briefly, the team implemented data augmentation to introduce variation in the dataset via rotation, shifting and flipping. Thereafter, the IRRCNN model was utilised with stochastic gradient descent (SGD) as the optimisation function [5].

Using the same dataset that our team would be using, the research found that magnification affects the accuracy in training and testing and training with 100x magnification results in the best performance (Table 1) [5]. Moreover, comparison of binary classification results (malignant & benign) against other models showed that IRRCNN results in better performance (Table 2) [5].



	Methods	Year	Classification Rate (100R) at Magnification Factor			
			40×	100×	200×	400×
Image Level	CNN+patches [15]	2016	85.6 ± 4.8	83.5 ± 3.9	83.1 ± 1.9	80.8 ± 3.0
	LeNet + Aug [18]	2017	40.1 ± 7.1	37.5 ± 6.7	40.1 ± 3.4	38.2 ± 5.9
	AlexNet + Aug [18]	2017	70.1 ± 7.4	75.8 ± 5.4	73.6 ± 4.8	84.6 ± 1.8
	CSDCNN + Aug [18]	2017	92.8 ± 2.1	93.9 ± 1.9	93.7 ± 2.2	92.9± 2.7
	IRRCNN +w/o Aug.	2018	95.69 ± 1.18	95.37 ±1.29	95.61± 1.37	95.15 ± 1.24
	IRRCNN + w Aug.	2018	97.09 ±1.06	97.57 ±0.89	97.29 ±1.09	97.22 ±1.22
Patient Level	LeNet + Aug [18]	2017	48.2 ± 4.5	47.6 ± 7.5	45.5 ± 3.2	45.2 ± 8.2
	AlexNet + Aug [18]	2017	74.6 ± 7.1	73.8 ± 4.5	76.4 ± 7.4	79.2 ± 7.6
	CSDCNN + Aug [18]	2017	94.1 ± 2.1	93.2 ± 1.4	94.7 ± 3.6	93.5± 2.7
	IRRCNN +w/o Aug.	2018	95.81 ± 1.81	94.44 ± 1.3	95.61 ± 2.9	94.32± 2.1
	IRRCNN + Aug.	2018	96.76 ± 1.11	96.84±1.13	96.67±1.27	96.27±0.87

Table 1: Comparison of neural network methods against different magnifications



	Method	Year	Classification Rate at Magnification Factor			
			40×	100×	200×	400×
Image Level	CNN +fusion(sum, product, max) [15] (highest results)	2016	85.6 ± 4.8	83.5 ± 3.9	83.6 ± 1.9	80.8 ± 3.0
	AlexNet + Aug [18]	2017	85.6 ± 4.8	83.5± 3.9	83.1 ± 1.9	80.8 ± 3.0
	ASSVM [28]		94.97	93.62	94.54	94.42
	CSDCNN + Aug [18]	2017	95.80± 3.1	96.9 ± 1.9	96.7 ± 2.0	94.90 ± 2.8
	IRRCNN	**2018**	**97.16 ± 1.37**	**96.84 ±1.34**	**96.61± 1.31**	**95.78 ± 1.44**
	IRRCNN + Aug	**2018**	**97.95± 1.07**	**97.57± 1.05**	**97.32± 1.22**	**97.36± 1.02**
Patient Level	CNN +fusion (sum, product, max) [15]	2016	90.0 ± 6.7	88.4 ± 4.8	84.6 ± 4.2	86.10 ± 6.2
	Bayramoglu et al. [14]	2016	83.08 ± 2.08	83.17 ± 3.51	84.63 ± 2.72	82.10 ± 4.42
	Multi-classifier by Gupta et al. [13]	2017	87.2 ± 3.74	88.22 ± 3.23	88.89 ± 2.51	85.82 ± 3.81
	CSDCNN + Aug [18]	2017	92.8 ± 2.1	93.9 ± 1.9	93.7 ± 2.2	92.90 ± 2.7
	IRRCNN +wo aug.	2018	96.69 ±1.18	96.37 ±1.29	96.27±1 .57	96.15 ±1.61
	IRRCNN + w. Aug.	2018	**97.60± 1.17**	**97.65± 1.20**	**97.56± 1.07**	**97.62± 1.13**

Table 2: Comparison of binary classification among different models

Another research by Xie and his team used transfer learning on Inception_ResNet_V2, which they claimed to be the best performing deep learning architecture for breast cancer diagnosis, to extract features for unsupervised learning [6]. Transfer learning was based on the same model trained on the ImageNet dataset, which was considerably large. Thereafter, an autoencoder network was built to transform the features to a lower dimensional space, where clustering analysis of the image is performed [6].

Similar to the previous research, the accuracy for both binary and multi-class classification is very high with at least 90% accuracy, with augmented dataset doing much better than raw dataset [6]. However, the paper found that clustering accuracies are not as good as classification accuracies and that better clustering accuracies could be achieved via further research [6].

It is important to note that how their dataset is split into training and validation is unknown and there might be potential contamination of test data. With reference to a similar work from a medium article written by Abhinav Sagar, the author did not split the data by patients, hence the same person's cell image data (but of differing locations) appeared in both training and testing [7]. Below are the results after we ran his code from the medium article:

	Precision	Recall	F1-score
Benign	0.93	0.94	0.94
Malignant	0.95	0.94	0.95
Accuracy	0.94		

Table 3: Results before splitting dataset by patients

	Precision	Recall	F1-score
Benign	0.97	0.51	0.67
Malignant	0.77	0.99	0.87
Accuracy	0.81		

Table 4: Results after splitting dataset by patients

It is evident after comparing Table 3 and Table 4 that eliminating contamination of the test set significantly reduced the accuracy of the trained model. Therefore, while we try to replicate a highly accurate model that takes less time to train, we also aim to build our model in a less biased way during data sampling.

3 Methodology

3.1 Aims

Our aim is to build a model that is able to distinguish between malignant and benign tumours with high classification accuracy. We also aim for a high recall rate of malignant tumours because we want to reduce the number of false positive outputs (i.e. classifying a malignant tumour as benign). Misclassification of malignant tumours as benign are serious mistakes and can potentially cause death.

In addition, we would like to build a model that trains fast and still produces desirable results.

3.2 Metric Definition

3.2.1 Classification accuracy

Classification accuracy measures the ratio of the number of correct predictions to the total number of predictions made, i.e the amount of test data. In this case, we are looking at correctly classifying breast cancer as malignant or benign. If n=100 and there are 96 correctly classified test data points, then the classification accuracy would be 96/100 =0.96.

3.2.2 Precision

Precision is a measure of the number of correct positive predictions, i.e the ratio of true positives to the total number of positive predictions (true positives & false positives).

3.2.3 Recall

Recall is a measure of the proportion of actual positives that was predicted as true positives, i.e. the ratio of true positives to the total number of true positives & false negatives.

3.2.4 Confusion matrix

The confusion matrix gives an overview of 4 important terms - true positives, false positives, true negatives and false negatives. The accuracy of the confusion matrix can be calculated by adding the true positives and negatives together and dividing by the total predictions. For instance, the accuracy of the confusion matrix below would be (60+120)/200=0.9 (Table 5).

n = 200	Predicted: No	Predicted: Yes
Actual: No	60	10
Actual: Yes	10	120

Table 5: Example of a Confusion Matrix

3.3 Pre-processing

3.3.1 Google Colab, ImageNet and Keras

Our group decided to use Google Colab as our development platform because of its many benefits:

1) Free GPUs
2) Easy to collaborate
3) Support for Jupyter Notebook
4) One person can create multiple Google Colabs, effectively giving the productivity of multiple computers

Our group decided to use pre-trained network due to the limited size of our dataset. We decided to use pre-networks that are trained from ImageNet because of the following reasons:

1) Keras readily supports ImageNet, saving us time and allowing us to learn about deep learning concepts earlier.
2) ImageNet is a huge database with millions of images that are general enough for pre-trained models to be repurposed to most image classification tasks [8].

Our model has the following parameters for the subsequent experiments:

1) Image size of 200 by 200
2) Trained on 30 epochs then select the weights from the epoch with the best validation accuracy.
3) VGG16 model that is pre-trained by ImageNet. Only the convolutional layers are imported, while the dense layers are not.
4) Two dense layers of size 512 and 2 were added to the imported convolution base. 2 is selected as the final layer because this project is a binary classification.
5) Training, validation and test data is split by the ratio 50 : 25 : 25.

These numbers are arbitrarily chosen to begin our trial and error approach to find the best model. These values will be kept constant until we build the desired model.

3.3.2 Data Partitioning

As explained from our experiment shown in Table 3 and Table 4, we split the data according to patients, such that if a patient's cell image appears in the training set, it will not be used for both validation and testing. We used 50% of our images for training, 25% for validation and 25% for testing.

From the dataset, it is also important to note that the number of malignant tumour images (5429) far exceeds the benign tumour images (2480). As an experiment, we decided to test if this affects our final results using VGG16.

	Precision	Recall	F1-score
Benign	0.82	0.60	0.73
Malignant	0.85	0.93	0.89
Accuracy	0.84		
Run time	41 seconds per epoch		

Table 6: Results with original dataset size of 5429 malignant tumour images and 2480 benign tumour images

It is evident that our model did poorly when classifying benign tumours. We observed that the model is favoured to classifying any image as malignant and still have a high accuracy rate due to this data discrepancy. This imbalance of our training dataset resulted in an "Accuracy Paradox".

To counter this, we multiplied the number of benign tumour images to match the number of malignant tumour images (5429 benign, 5429 malignant) so that the model would not receive a higher score by being more likely to classify any image as malignant. Later we did the reverse, by reducing the malignant tumour dataset to match the number of benign tumour data (2480 benign, 2480 malignant). To avoid any bias, we randomly choose the malignant images to remove.

We obtained the following results:

	Recall	Precision	F1-score
Benign	0.88	0.84	0.86
Malignant	0.84	0.88	0.86
Accuracy	0.86		
Run time	58 seconds per epoch		

Table 7: Results after expansion of benign dataset to match the size of malignant dataset, both at 5429 images

	Precision	Recall	F1-score
Benign	0.87	0.78	0.83
Malignant	0.81	0.89	0.85
Accuracy	0.84		
Run time	30 seconds per epoch		

Table 8: Results after reduction of malignant dataset to match the size of benign dataset, both at 2480 images

Comparing Table 7 (expanding benign dataset) and Table 8 (reducing malignant dataset), we observed that the accuracy for both options are comparable yet the run time for Table 8 is significantly faster. The model in Table 7 runs slower because it has a larger dataset to train on. Subsequently, we chose to reduce the size of malignant dataset to match the size of benign dataset for its shorter training time.

3.3.3 Data Augmentation

We applied data augmentation while loading the image data into the model for training using Keras' ImageDataGenerator. This augmentation makes the model takes in a slightly different image in each epoch, hence reducing the chance of overfitting. We applied rotation, flipping and shifting to the images. The following table shows the result of the same model but with data augmentation.

	Precision	Recall	F1-score
Benign	0.92	0.78	0.85
Malignant	0.82	0.94	0.87
Accuracy	0.86		
Run time	30 seconds per epoch		

Table 9: Results from using data augmentation

Comparing Table 8 (without data augmentation) and 9 (with data augmentation), it is evident that data augmentation of our training set improved the accuracy of our test results, without compromising on training time.

3.3.4 Data Loading

As our team uses the Google Colab server, our data images require a long time to be loaded from our Google Drive. Initially, it took more than an hour to load our images using a single thread of control in our Python code. However, having only one thread is inefficient because the CPU is left idling while waiting for the image to fetch from Google Drive to Google Colab. By creating multiple threads to collect the images, we maximise the usage of the CPU and effectively cut the time to load the images by 80%. The time required to load the images went from more than an hour to just 10 minutes. This achieves our goal of having a shorter run time.

3.3.5 Choosing the image size

We have only used image size of 200 * 200 so far. However, we could run our model with different image sizes. A higher resolution image (i.e. image with larger size) might mean that the model could receive a higher level of details from its input, potentially leading to higher accuracy score. However, higher accuracy could also come at the cost of longer training time as the model needs to process more information. Below, we experimented on using different image sizes and recorded the time taken and the results' accuracy score.

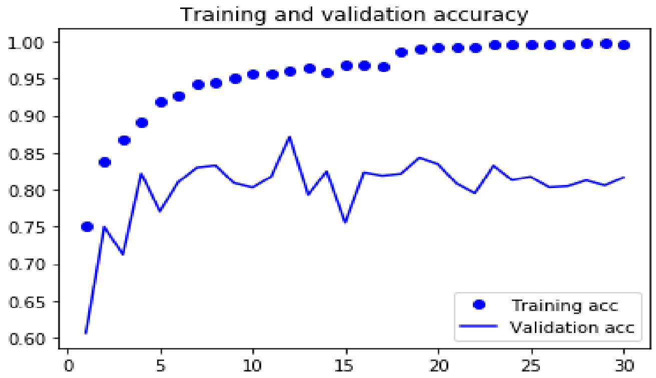
Image size (pixel)	Time per epoch	Accuracy Score (%)
32 * 32	7 seconds	0.77
100 * 100	11 seconds	0.80
200 * 200	29 seconds	0.86
250 * 250	38 seconds	0.82
300 * 300	42 seconds	0.85
350 * 350	55 seconds	0.80
400 * 400	Google Colab ran out of memory	

Table 10: Results for using different image sizes

From Table 10, it could be observed that larger image size generally leads to higher accuracy, but with longer run time. We decided to use 300 * 300 because it's accuracy score is comparable with 350 * 350 but with shorter run time.

3.3.6 Epoch number

Our default approach is to train the model with 30 epochs but have modelled it to pick the weights from the epoch that has the best validation accuracy. Below is a graph of validation accuracy and training accuracy against epoch number for the model mentioned in 3.3.3.



Graph 1: Accuracy score against epoch number

Graph 1 shows that the validation accuracy plateaus from around epoch 15 onwards. In the above case, our model picks the weight at epoch 13 because it has the best validation accuracy score. As the model plateaus way before the 30th epoch, we will continue using 30 epochs as our default.

3.4 Building the Model

3.4.1 CNN Architectures

During exploratory data analysis on our dataset in section 3.3, we have only used VGG16. To improve our results, we compared several other models and their performance on classifying our breast cancer biopsy image dataset, namely Xception, InceptionRes, ResNet and DenseNet.

3.4.2 Hyperparameter Tuning

All the above models in 3.4.1 ran with weights from ImageNet unfrozen. Results could be improved if we freeze the weights from a certain layer onwards. We first ran an experiment to determine the correct layer to unfreeze from. Next, we will run the model with weights frozen first before unfreezing some layers to determine the best combination. This latter method is also known as fine-tuning.

4 Results Analysis

4.1 Results

We ran the different CNN models, as explained in section 3.4.1. Note that image size is 300 * 300. Our results are summarised in the following table:

Model	Time per epoch (seconds)	Accuracy score (%)
VGG16	56	75.27 %
Xception	288	84.99%
InceptionRes	118	81.94 %
ResNet	120	82.76 %
DenseNet201	111	80.70 %

Table 11: Results using different CNN models.

From Table 11, we observed that Xception has the highest accuracy score. However, we decided to use VGG16 for our subsequent experiment because it has the shortest run time and it has a relatively simple architecture. The shorter run time allows us to run multiple iterations of our model quickly while we trial and error to reach the best parameter. The relatively simple architecture also means that we could understand the model better and allow us to tweak the various hyper-parameters to reach our model that could give us the best result. Note that we do consider recall, but due to their relatively similar percentages, we have used accuracy as the baseline comparison. A more detailed result, Table 14, will be included in the appendix for reference.

To further enhance pre-trained VGG16's accuracy on our dataset, we decided to freeze some layers during the training. There are two approaches:

1) Add two dense layers at the end of the model. Freeze the model up to the desired layer then train the model. Only one iteration of training here.
2) Add two dense layers at the end of the model. Freeze all the weights from the pre-trained VGG16 model and train only the weights of the two dense layers. Once the weights from the two dense layers are initialised, unfreeze the model to a desired layer, then perform another round of training. Note that there are two iterations of training in this approach. This approach is also known as fine-tuning.

Begin unfreezing from:	Without fine-tuning	With fine-tuning
Unfreeze everything	0.78	0.84
block1_conv1	0.86	0.82
block2_conv1	0.80	0.72
block3_conv1	0.80	0.88
block4_conv1	0.86	0.83
block5_conv1	0.82	0.89
Freeze everything	0.79	NA

Table 12: Results from freezing and unfreezing different layers of VGG16

From table 12, we observed that unfreezing from block5_conv1 layer of the VGG16 with fine-tuning gives the best result.

Finally, we arrive at our final model:

1) Input image size = 300 * 300
2) CNN model = VGG16
3) Fine-tuning by freezing all weights in the first round of training then unfreezing from block5_conv1 onwards for the second round of training

	Recall	Precision	F1-score
Benign	0.92	0.85	0.88
Malignant	0.86	0.93	0.89
Accuracy	0.89		
Run time	60 seconds per epoch		

Table 13: Results of our final model

4.2 Discussion

During exploratory data analysis, we found out about the importance of avoiding data contamination. There must be sufficient knowledge of the data before any separation of data can be done. Having an unequal dataset can also result in "Accuracy Paradoxes", stressing the importance of having a balanced dataset.

Next, data augmentation helps in reducing variance and prevents overtraining of data.

Additionally, even though having larger image sizes can sometimes produce better results, the time and space complexities can scale upwards and present a problem. Also, there is an upper limit of image size that depends on the memory size of the given CPUs and GPUs.

After exploratory data analysis, we concluded that having equal representation of each data in benign and malignant tumour classes by removing malignant images then randomly augmenting training data gives the best results.

During the choosing of models, Xception worked particularly well in identifying tumour tissues. However, its run time is too long, making it unsuitable for us to run multiple iterations of the model given a limited time to complete this project. Hence, in the interest of time, we continued with our default model, VGG16.

From Table 12, we observed that unfreezing from block5_conv1 onwards with fine-tuning returns the best accuracy result. It is also noteworthy that fine tuning at layer block5_conv1 performs significantly better than freezing every layer from the convolutional base. This observation suggests that ImageNet may have insufficient cell images leading to the failure to generalise to our dataset after training. However, unfreezing beyond block5_conv1 caused the result to be worse. This finding suggests that layers before block5_conv1 encode more-generic feature that is reusable to the cell image classification, while the weights at block5 is too specific to be reused for cell image classification.

Finally, our team arrived with a tuned model shown in Table 13, with a result that is significantly better from what we started with in Table 4.

5 Limitations and Future Work

5.1 Limitations

In terms of generalisation, it is unknown if our model could extend beyond breast cancers to differentiate between benign and malignant tumours of other cancers. Most of the time, different cancer imaging holds different features for the identification of benign or malignant tumour. For instance, malignancy in lung cancer was more prevalent in nodules >20mm in size or in nodules with irregular borders [9]. On the other hand, malignancy in breast cancer was more prevalent in lesions with dimensions 1.1cm to 2cm [10]. Given the specificity of features to different cancers, it is thus likely that the model may not be generalisable across different cancers.

On the other hand, CNN is also computationally expensive and requires a large amount of training data. Although data augmentation could help supplement data scarcity, especially the data imbalances, it is still insufficient given the complexity of breast cancer identification [11]. Therefore, more imaging data could be given to further improve our current model, especially for the training of multi-classification of tumours.

5.2 Future Work

In the future, more could be done by tuning the hyperparameters on different kind of network apart from VGG16. Due to the limited time of this project, other pre-trained models and their layers are not researched upon extensively. Therefore, future extension include running trial and error on different models, such as the freezing and unfreezing of different layers, despite their longer run time. Other hyperparameters of the CNN model, such as the filter size and stride steps, could also be further evaluated to determine the best parameters for the model and dataset.

As explained earlier, data scarcity is a problem for the accuracy of our model, given the complexity of breast cancer identification. Hence, more datasets pertaining to breast cancer imaging, such as those comparable to the scale of ImageNet, is preferable.

Moreover, the results of generalisation of our model towards other types of cancers is currently unknown. There could be a possibility of classifying tumours of cancers with a similar morphology or taxonomy on breast tumours despite the unlikelihood, as mentioned earlier. Future work could thus focus on improving the generalisation of the model to other cancer types that are morphologically similar.

Another area to explore is to create a custom cost function that makes the misclassifying of malignant tumor as benign tumor more expensive. This makes the model better at capturing malignant tumor which is important given that late treatment of malignant breast tumor usually leads to death.

Also, breast cancer consists of different significant risk factors, such as gender, race, and genetics [12]. To further improve our model, we could utilise transfer learning to train on these risk factors, which may improve the performance of our model.

Lastly, apart from the binary classification of benign and malignancy of tumours, future work could attempt the multi-classification of tumours. Under malignant tumours, they can be further classified as ductal, lobular, mucinous or papillary carcinomas. Under benign tumours, they can be further classified as adenosis, fibroadenomas, phyllodes tumours or tubular adenomas. Our current dataset splits the tumours into these 8 categories, but each category contains insufficient data for the training of an accurate model. Therefore, with sufficient data in the future, perhaps a multi-class classification model can be built for better understanding and identification of the sub-class of tumours.

6 Conclusion

By following the important concepts of machine learning, such as implementing data augmentation and avoiding data contamination, we are able to come up with a sufficient model for diagnosing benign and malignant breast cancers. Although limited by its generalisation power and data scarcity, the model still works particularly well with an overall accuracy of 0.89. Moreover, future works to further improve the ability to generalise and multi-classification is required for better understanding of the sub-class of tumours.

7 Acknowledgements

First, we would like to thank our professor, Dr Kan Min-Yen, for organising this project, as well as teaching this module. We would also like to thank our teaching assistant, Mr Lee Yi Quan, for providing us with guidance and resources for this project.

We would also like to thank Mr Abhinav Sagar, the author who wrote a medium article which inspired us to take on this project. Lastly, we would like to thank Google for providing us with a server and Keras library for making this project possible.

8 References

[1]. Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., & Jemal, A. (2018). Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: *a cancer journal for clinicians, 68*(6), 394-424.

[2]. Singh, H., Schiff, G. D., Graber, M. L., Onakpoya, I., & Thompson, M. J. (2017). The global burden of diagnostic errors in primary care. *BMJ Qual Saf, 26*(6), 484-494.

[3]. Gandhi, T. K., Kachalia, A., Thomas, E. J., Puopolo, A. L., Yoon, C., Brennan, T. A., & Studdert, D. M. (2006). Missed and delayed diagnoses in the ambulatory setting: a study of closed malpractice claims. *Annals of internal medicine, 145*(7), 488-496.

[4]. Spanhol, F., Oliveira, L. S., Petitjean, C., Heutte, L. (2016). A Dataset for Breast Cancer Histopathological Image Classification, *IEEE Transactions on Biomedical Engineering (TBME), 63*(7):1455-1462.

[5]. Alom, M. Z., Yakopcic, C., Nasrin, M. S., Taha, T. M., & Asari, V. K. (2019). Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network. *Journal of digital imaging*, 1-13.

[6]. Xie, J., Liu, R., Luttrell, I. V., & Zhang, C. (2019). Deep Learning Based Analysis of Histopathological Images of Breast Cancer. *Frontiers in genetics, 10*, 80.

[7]. Sagar, A. (2019, September 29). Convolutional Neural Network for Breast Cancer Classification. Retrieved from https://towardsdatascience.com/convolutional-neural-network-for-breast-cancer-classification-52f1213dcc9.

[8]. Fei-Fei, L. (2010, March). ImageNet: crowdsourcing, benchmarking & other cool things. In *CMU VASC Seminar*(Vol. 16, pp. 18-25).

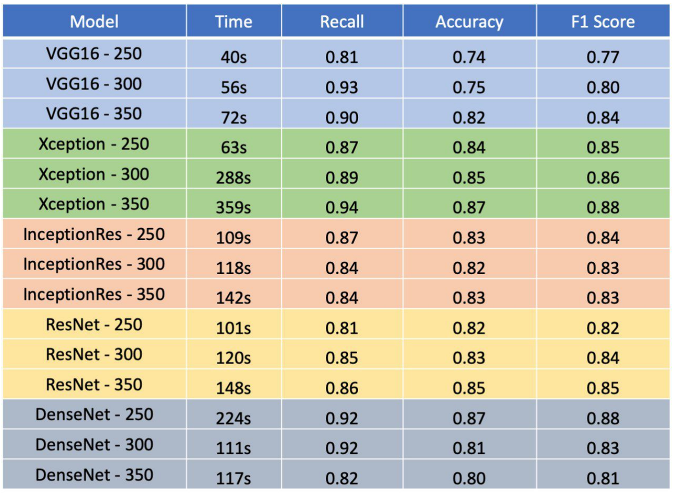
[9]. Wahidi, M. M., Govert, J. A., Goudar, R. K., Gould, M. K., & McCrory, D. C. (2007). Evidence for the treatment of patients with pulmonary nodules: when is it lung cancer?: ACCP evidence-based clinical practice guidelines. *Chest, 132*(3), 94S-107S.

[10]. Luo, W. Q., Huang, Q. X., Huang, X. W., Hu, H. T., Zeng, F. Q., & Wang, W. (2019). Predicting Breast Cancer in Breast Imaging Reporting and Data System (BI-RADS) Ultrasound Category 4 or 5 Lesions: A Nomogram Combining Radiomics and BI-RADS. *Scientific reports, 9*(1), 1-11.

[11]. Du, S. S., Wang, Y., Zhai, X., Balakrishnan, S., Salakhutdinov, R. R., & Singh, A. (2018). How many samples are needed to estimate a convolutional neural network?. In *Advances in Neural Information Processing Systems* (pp. 373-383).

[12]. Collins, A., & Politopoulos, I. (2011). The genetics of breast cancer: risk factors for disease. *The application of clinical genetics, 4*, 11.

9 Appendix



Model	Time	Recall	Accuracy	F1 Score
VGG16 - 250	40s	0.81	0.74	0.77
VGG16 - 300	56s	0.93	0.75	0.80
VGG16 - 350	72s	0.90	0.82	0.84
Xception - 250	63s	0.87	0.84	0.85
Xception - 300	288s	0.89	0.85	0.86
Xception - 350	359s	0.94	0.87	0.88
InceptionRes - 250	109s	0.87	0.83	0.84
InceptionRes - 300	118s	0.84	0.82	0.83
InceptionRes - 350	142s	0.84	0.83	0.83
ResNet - 250	101s	0.81	0.82	0.82
ResNet - 300	120s	0.85	0.83	0.84
ResNet - 350	148s	0.86	0.85	0.85
DenseNet - 250	224s	0.92	0.87	0.88
DenseNet - 300	111s	0.92	0.81	0.83
DenseNet - 350	117s	0.82	0.80	0.81

Table 14: Additional results for each model explored in 3.4.1, showing Recall and F1 scores of varying sizes of images 250 * 250, 300 * 300 (shown in Table 11), and 350 * 350.

Codes used for the final model are available in a Jupyter Notebook via the following link:

https://colab.research.google.com/drive/1WbYpvzyohbnu8Ddz-kDvlLupDji-YAcC