

Pilot Study of Breast Cancer Classification Using Deep Learning Approach

BY

CHONG KIM YEW

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology (FICT)
(Kampar Campus)

May 2020

REPORT STATUS DECLARATION FORM

Title: Pilot Study of Breast Cancer Classification Using Deep
Learning Approach


Academic Session: MAY 2020

I CHONG KIM YEW
(CAPITAL LETTER)


declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,



(Author's signature)



(Supervisor's signature)

Address:
12, Jalan Taming Emas 2,
Taman Taming Emas,
43000 Kajang, Selangor

Sayed Ahmad Zikri Bin Sayed Aluwee

Supervisor's name

Date: 10/09/2020

Date: 10 September 2020

Pilot Study of Breast Cancer Classification Using Deep Learning Approach

BY

CHONG KIM YEW

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology (FICT)
(Kampar Campus)

May 2020

DECLARATION OF ORIGINALITY

I declare that this report entitled “**PILOT STUDY OF BREAST CANCER CLASSIFICATION USING DEEP LEARNING APPROACH**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature :  _____

Name : CHONG KIM YEW

Date : 10/09/2020

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Dr Sayed Ahmad Zikri Bin Sayed Aluwee who has given me this bright opportunity to engage in this “Pilot Study of Breast Cancer Classification Using Deep Learning Approach” project. It is an opportunity for me to discover and acquire the latest knowledge of applying deep learning techniques to detect breast cancer. A million thanks and appreciation to you.

Finally, I must say thanks to my parents and family for their love, support and continuous encouragement throughout the course.

ABSTRACT

Breast cancer is harmful diseases and also the most common cancer causes of cancer-related deaths among women. Traditional diagnosis and detection method requires a lot of experience, professional knowledge, and human sometimes might prone to fall diagnostic errors. The effective way to reduce the mortality rate is to diagnose cancer in the earlier stage by screening. The main objective of this project is to develop a preliminary disease diagnosis system for breast cancer detection and segmentation by using a state-of-the-art deep learning technique. Moreover, the methodology applied in this project is using Mask R-CNN pre-trained model. This project was focused on a fundamental study and developed by customizing and training the pre-trained models with the breast ultrasound images. Thus, the model deployed in this project achieved the lowest loss, which is 0.245. By applying this model, three prediction outputs results will show the class label, bounding box coordination and segmentation mask for the detected object. By implementing this project, Malaysia woman might have the benefit and able to prevent cancer at the earlier stage, and it might reduce the rate of mortality caused by breast cancer.

TABLE OF CONTENTS

FRONT COVER	i
REPORT STATUS DECLARATION FORM	ii
TITLE PAGE	iii
DECLARATION OF ORIGINALITY	iv
ACKNOWLEDGEMENTS	v
ABSTRACT	vi
TABLE OF CONTENTS	vii
LIST OF FIGURES	x
LIST OF TABLES	xii
LIST OF ABBREVIATIONS	xiii
Chapter 1 Introduction	1
1.1 Problem statement	1
1.2 Background information	2
1.2.1 Deep learning	2
1.2.2 CNN architectures	2
1.2.3 Breast cancer	3
1.2.4 Mammogram	4
1.2.5 Breast ultrasound	4
1.2.6 Breast biopsy	5
1.2.7 Classification with localization in image and object detection	5
1.2.8 Image segmentation	6
1.3 Project objective	7
1.4 Project scope	7
1.4.1 The medical image classification module	7
1.4.2 Detect breast cancer module	7
1.5 Impact, significance and contribution	8
1.6 Proposed approach/ study	9
1.7 Highlight of what have been achieved	10
1.8 Report Organization	10
	vii

Chapter 2: Literature review	11
2.1 Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening	11
2.2 Multi-Class Breast Cancer Classification using Deep Learning CNN	12
2.3 Breast Cancer Histopathological Image Classification: A Deep Learning Approach	14
2.4 Segmentation of the Prostatic Gland and the Intraprostatic Lesions Using Mask R-CNN	15
2.5 Ensemble Neural Networks for Digital Pathology Images Classification and Segmentation	16
Chapter 3: System Design	18
3.1 Obtaining the image	18
3.2 Pre-processing of training datasets	19
3.2.1 Resize images	19
3.2.2 Mini Mask	20
3.2.3 Transformation of images	22
3.3 Customize the model	22
3.3.1 Modify the configuration	22
3.3.2 Customize and modify the training code	23
3.3.3 Train and visualize the results	25
3.4 Deploy and evaluate the model	26
Chapter 4: System Methodology	28
4.1 Methodologies and tools	28
4.2 System Requirement	31
4.2.1 Hardware	31
4.2.2 Software	31
4.2.3 User Requirements	32
4.3 Specification	32
4.3.1 System Performance Definition	32
4.3.2 Evaluation	32
4.4 Project Timeline	34
Chapter 5: Implementation and result	36

5.1 Dataset preparation	36
5.2 Mask R-CNN installation	36
5.3 Train model	39
5.4 Visualize the result	40
5.5 Deploy model and perform testing	41
Chapter 6: Conclusion	43
6.1 Future Work	43
6.2 Conclusion	43
References	45
APPENDICES	
Appendix A: Project Poster	A-1
Appendix B: Plagiarism Check Summary	A-2
Appendix C: Graph	A-3

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.2.1	Neural network with many hidden layers	2
Figure 1.2.2	CNN architecture	3
Figure 1.2.3	Mammogram	4
Figure 1.2.4	Breast ultrasound image	4
Figure 1.2.5	Normal breast histopathology image	5
Figure 1.2.6	Different between classification and classification with localization	5
Figure 1.2.7	Three different techniques of computer vision	6
Figure 1.2.8	Semantic segmentation and instance segmentation	6
Figure 1.6.1	Flow Chart of proposed approach	9
Figure 2.1.1	Four main tasks in a model	11
Figure 2.2.1	Traditional CNN model	13
Figure 2.2.2	DenseNet with three dense blocks architecture	13
Figure 2.4.1	Mask R-CNN network architecture	15
Figure 2.5.1	T-Net architecture	16
Figure 2.5.2	The Original microscopy images and the segmentation result	17
Figure 3.1.1	System Overview Flow Chart	18
Figure 3.1.2	Sample of breast ultrasound images and mask images	19
Figure 3.2.1	Resizing	20
Figure 3.2.2	Image original size	21
Figure 2.2.2	Resized images and performed augmentation	21
Figure 3.3.1	Modify config.py	22
Figure 3.3.2	Modify utils.py	23
Figure 3.3.3	CustomConfig method	24
Figure 3.3.4	Add two class in training code	24

Figure 3.3.5	TensorBoard show the loss at every epoch	25
Figure 3.4.1	Image information and the actual label	26
Figure 3.4.2	Prediction result	26
Figure 3.4.3	Actual image and the mask image	27
Figure 3.4.4	Prediction result	27
Figure 4.1	Basic structure of the Mask R-CNN architecture	28
Figure 4.1.2	Feature maps for the RPN	29
Figure 4.1.3	Roi Pooling (left) vs Roi Align (right)	30
Figure 4.3.1	IoU formula	32
Figure 4.3.2	Example of computing IoU for various bounding box	33
Figure 4.3.3	Loss calculation	33
Figure 4.4.1	Gantt Chart for the project development timeline (Jan to May semester)	34
Figure 4.4.2	Gantt Chart for the project development timeline (June to October semester)	34
Figure 5.1.1	Online labelling tool	36
Figure 5.2.1	Clone Mask RCNN source code	37
Figure 5.2.2	Run setup.py script	37
Figure 5.2.3	Mask RCNN installed	38
Figure 5.2.4	Install pycocotools	39
Figure 5.3.1	Train model command	39
Figure 5.4.1	Log TensorBoard command	40
Figure 5.4.2	TensorBoard	40
Figure 5.5.1	Import library	41
Figure 5.5.2	Load dataset	41
Figure 5.5.4	Actual images and the label	42
Figure 5.5.5	Predicted Result	42

LIST OF TABLES

Table Number	Title	Page
Table 2.3.1	Results of fine-tuned last layer in different models	11
Table 2.3.2	Results of fine-tuned all layers in different models	11
Table 3.1.1	Hardware used in this project	16

LIST OF ABBREVIATIONS

AUC	Area under the curve
BI-RADS	Breast Imaging Reporting and Data System
CAD	Computer-aided detection and diagnosis
CNN	Convolutional Neural Network
DSC	Dice similarity coefficient
FPN	Feature pyramid network
HD	Hausdorff distance
IoU	Intersect over union
L-CC	Left craniocaudal
MRI	Magnetic Resonance Images
NN	Neural Network
R-CC	Right craniocaudal
R-CNN	Region-based convolutional neural network
R-MLO	Right mediolateral oblique
ROI	Region of interest
RPN	Region proposal network
L-MLO	Left mediolateral oblique

Chapter 1 Introduction

1.1 Problem statement

According to the World Health Organization (WHO), they have shown the most common and top-ranking cancer is the breast cancer which has a negative effect on women health and also the first leading causes of cancer-related deaths among Malaysian women. They also did research in Malaysia, and their analysis shows that the new cases of breast cancer rate in 2018 which is 17.3% and the mortality rate are 11.0% which is the highest rate compared with other cancers such as lung cancer, liver cancer, and etc. Since the 1990s, the traditional computer-aided detection and diagnosis (CAD) system has been developed and used to assist pathologists in analyzing and evaluating medical images. However, the multicentre studies show that the traditional diagnosis method does not significantly increase the predictive accuracy of the diagnostic performance.

Clinical breast cancer screening method consists of mammography, ultrasound, needle biopsy, and other methods. However, this traditional diagnosis method requires their professional knowledge, and this needs an intense workload, especially for the enormous data. Besides that, pathologists are also prone to fall diagnostic errors due to not enough diagnostic experience. Therefore, this shows that the use of the deep neural network to classify histopathological images automatically is not only for the improvement of diagnostic efficiency, and also assist the doctors with more accurate diagnosis results.

1.2 Background information

1.2.1 Deep learning

Deep learning is a part of the machine learning technique and mostly implemented by using neural network techniques. Compared with the classic neural network, deep learning involves artificial neural networks algorithms which are more hidden layers to address complex data with many different structures. Deep learning deploys deep artificial neural network layers with many interconnected nodes that can rearrange and learn by itself when new information comes in without requiring human programming.

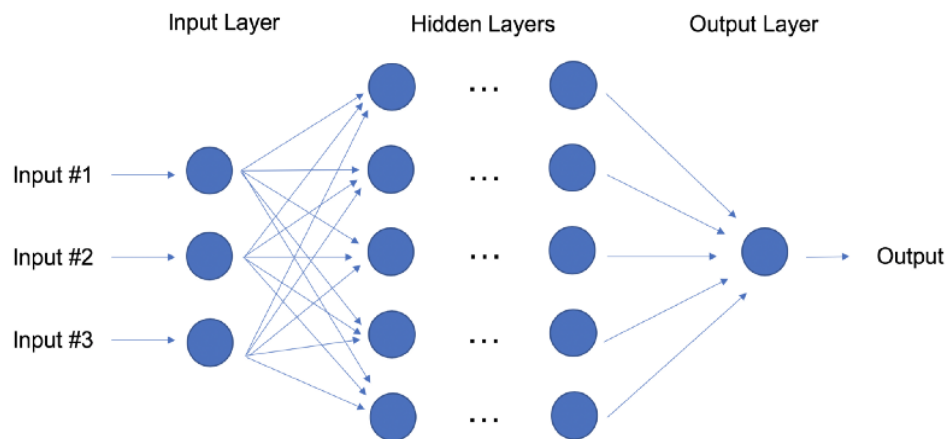


Figure 1.2.1 Neural network with many hidden layers

1.2.2 CNN architectures

Convolutional neural networks (CNN) is a type of deep learning algorithm which is used commonly and famous deep learning architecture. The convolutional neural networks are commonly used in image classification, image recognition, object detection, and etc. Besides that, CNN able to classify the image datasets are high-dimensional, and each image mostly contains thousands of pixels as traits that are high volume and complex (Jiang, et al., 2017). Most of the CNN models have the same architecture, such as ResNet, GoogLeNet and etc. The model consists of a series of convolution layer and pooling layers and followed by a fully connected layer. The convolution layer is a mathematical operation that different filters sizes used to the derivation of the activation map and extract the features of the input images. The pooling layer is a non-linear down-sampling that is used to reduce the dimension of the input data in order to reduce the memory consumption, reduce training time and overcome overfitting. The fully connected layer computes the

probability for each class, and all neurons will connect to this layer. If the data output performs binary classification or multi-label classification, then usually the output is using a sigmoid function. If the data output performs multi-class classification, then most of the time is using softmax function. As the figure is shown below (Dertat, 2017).

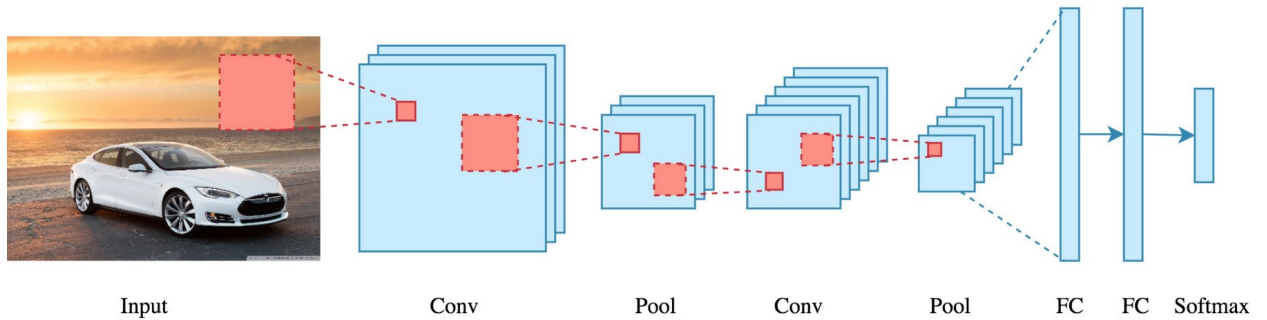


Figure 1.2.2 CNN architecture

1.2.3 Breast cancer

Breast cancer is a disease that the cells develop and grow in breast out of control. Breast cancer consists of a different kind that depends on what cells turn into cancer. Moreover, breast cancer is more common in women than in men. The cancer stage is according to the size of the tumour. Breast cancer has the main five stages, which are stage 0 to stage 4. Stage 0 is the cancer cells are not invaded surrounding tissues. At the first stage, the tumour size is about 2cm or smaller. The survival rate for stage 1 breast cancer approximately is 99% within 5 years. In the second stage, the cancer cell has spread to the lymph nodes. The rate of survival within 5 years is 86%. In stage 3, the cancer cells have spread to nearby tissues and the survival rate is about 57%. In stage 4, the survival rate drops to about 20% due to cancer has spread outside the breast. Furthermore, breast cancer screening is used for checking breasts before the symptoms and signs of the disease. The most common breast cancer screen test is Mammogram, Breast ultrasound. (Anon., 2018)

1.2.4 Mammogram

Mammography is one type of medical imaging that uses a low-dose X-ray system which can be used to detect the cancerous lumps inside the breast, and it is mostly used to screen for breast cancer.

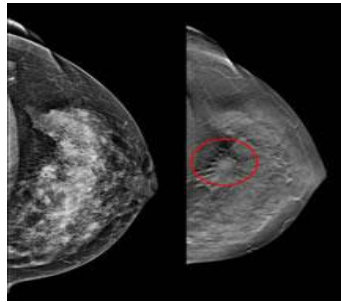


Figure 1.2.3 Mammogram

1.2.5 Breast ultrasound

Breast ultrasound is another type of medical imaging technique used to detect tumours. Ultrasound uses sound waves techniques to produce detailed medical images inside the breast. Moreover, ultrasound is considered suitable and safe for breastfeeding mothers and pregnant women who are not able to use X-rays and CT scans.

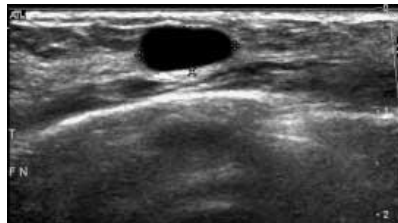


Figure 1.2.4 Breast ultrasound image

1.2.6 Breast biopsy

A breast biopsy is a diagnostic procedure that can certainly detect cancerous cells in a suspicious area. The breast biopsy is to remove the tissues or fluids from the suspicious breast area, the tissues tested and examined under the microscope to detect whether the tissue is cancerous. There are several types of biopsies such as fine-needle aspiration, core-needle biopsy and surgical biopsy. The doctor determines which type of biopsies to use depending on the size, appearance and location of the abnormal area on the breast (Biospy., n.d).

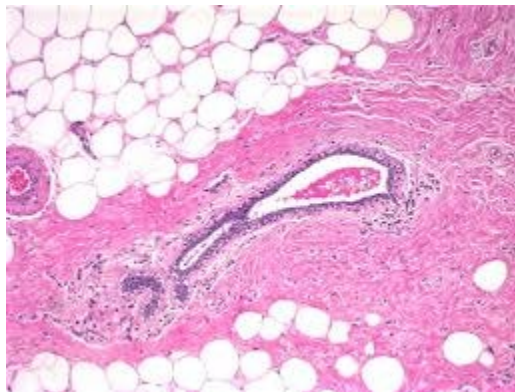


Figure 1.2.5 Normal breast histopathology image

1.2.7 Classification with localization in image and object detection

This classification is more challenging because the output consists of the class label and the location of the object, which is given by the bounding box. The bounding box is a square box or rectangle box around the detected object.

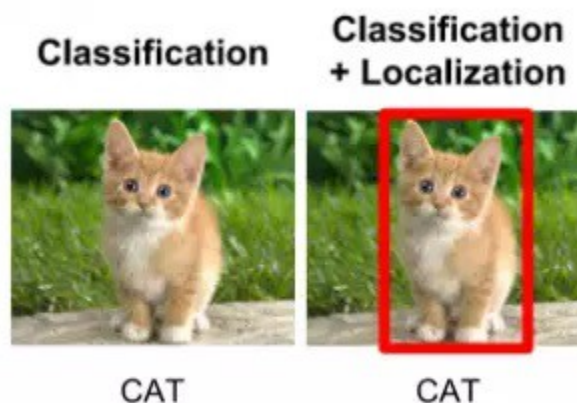


Figure 1.2.6 Different between classification and classification with localization

In object detection, it means the different objects detected with different class labels

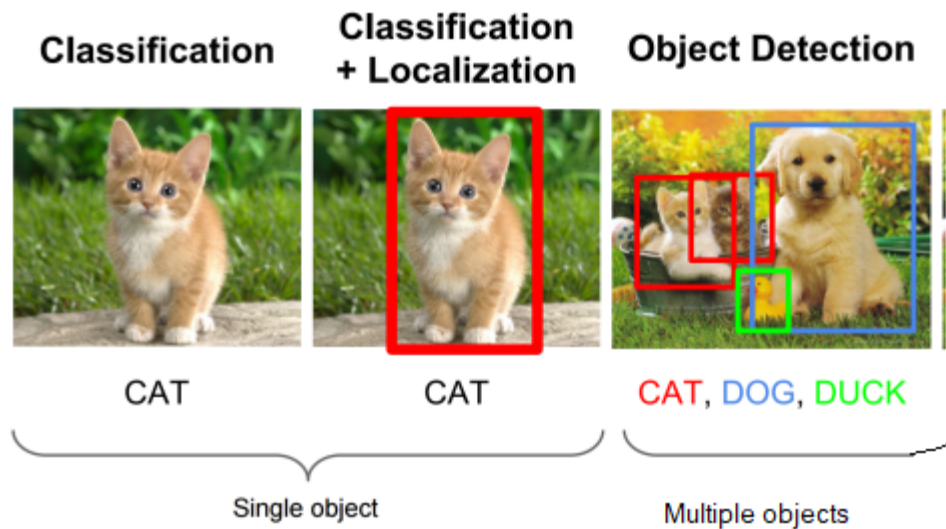


Figure 1.2.7 Three different techniques of computer vision

1.2.8 Image segmentation

Image segmentation consists of two types. The first type is the semantic segmentation. Every pixel in the image belongs to a particular class. For example, the image background is considered in one class, cat in an image as one class and dog in an image as one class. Thus, a total of 3 classes and segment the image in 3 colours mask. The second type is instance segmentation, it segment each detected object in separate colours.

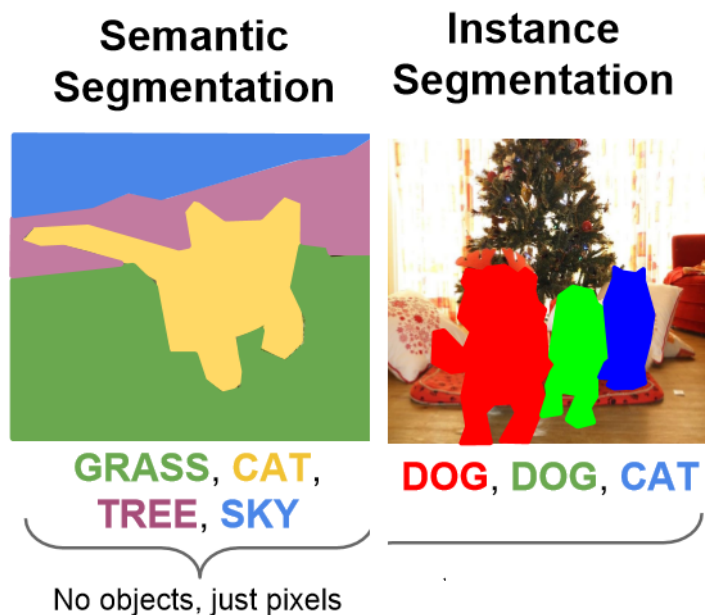


Figure 1.2.8 Semantic segmentation and instance segmentation

1.3 Project objective

The main objective of this project is to study and fundamental develop a preliminary disease diagnosis system for breast cancer detection using the existing techniques of deep learning to improve the performance of cancer detection. This project will customize a pre-trained mask R-CNN model for our project task. This state-of-the-art model has high accuracy performance, and it's able to perform classification and segmentation more efficiently.

The second objective of this project aims to reduce the workload and the time consumption of detecting a large amount of medical image dataset. By applying the pre-trained model into the system, the system able to assist the people in the medical field in order to reduce their hesitation in decision making and also prevent him from making a human mistake.

1.4 Project scope

1.4.1 The medical image classification module

This project will involve a model on medical image classification module in the system to increase the accuracy of the prediction result and further improve the diagnostic efficiency of analyzing medical image datasets. This module will involve the pre-trained deep learning models and customizing this model for this project task to classify cancer in the breast ultrasound images.

1.4.2 Detect breast cancer module

This project will include the breast cancer detection module in the system, and it used to detect and show the cancerous areas in the images within a short time so that this will be able to assist the pathologies in diagnosing the enormous amount of dataset and providing doctors with more accurate and objective diagnosis results. Therefore, this module will show the segmented area and the location of cancer from the medical image.

1.5 Impact, significance and contribution

As mentioned before, the first leading and most popular cause of cancer-related deaths in Malaysia is the breast cancer. In this era of information technology, some various ideas and devices can be used to diagnose breast cancer but using the vision inspection approach. Still, sometimes the doctors will be prone to making a human mistake. Therefore, the advantages of this project can assist the doctors can detect breast cancer in the earlier stage, and reduce the rate of mortality caused by breast cancer.

In addition, the people in medical fields such as doctors or surgeons, need to decide to proceed with the treatment of the patient, if making the wrong decisions it will cause the patient's condition even worse. Other than that, sometimes the symptoms of the disease are often similar, and it can be hard to detect from the medical images completely. Thus, the importance of having this project is to help the people in the medical field can faster to make a confirmation of the final result.

Besides that, the standard procedure used to detect the early stage of breast cancer is using a mammogram. Moreover, the rate of tumour growth in breast cancer is considered high compared to other types of cancer. Therefore, this will need to take some time for the doctors to diagnose cancer and proceed to the treatment, but at the same time, it will affect the other patient if not determined in the earlier stage. Therefore, doctors can reduce their workload by having this breast cancer diagnosis system.

1.6 Proposed approach/ study

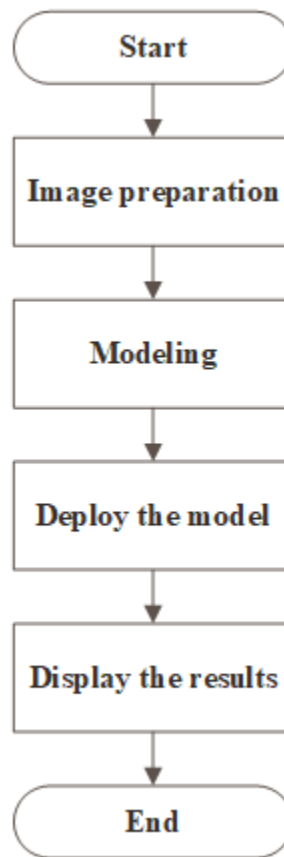


Figure 1.6.1 Flow Chart of proposed approach

In the first stage, the breast cancer images used in this project is the dataset of breast ultrasound images and ensure images used are no noise and suitable for model training. In the second stage, several modelling techniques can be used in this project, and select the most appropriate model and ensure the model can support the input images' format. After that, process the dataset and started training by using the selected model. After training, deploy the mask R-CNN models to perform classification and segmentation to the breast ultrasound images. Lastly, the results displayed by showing the bounding box, labelling, and masking to the area whether it is malignant or benign.

1.7 Highlight of what have been achieved

As mention before, the main objective of this project has been achieved, which is to study and develop a preliminary disease diagnosis system for breast cancer detection using the existing techniques of deep learning. First of all, the model training environment is successfully installed and support to run the training in my own machine. Moreover, the hundred of training and validation images is completely labelled using the online labelling tools. Besides that, the mask R-CNN deep learning model is successfully trained from the own custom datasets by using own machines. During the training process, the results are able to inspect by using the visualization tool and select the best-trained model to deploy. Furthermore, the trained model deployed and achieved an acceptable result which able to classify and detect the location of cancer.

1.8 Report Organization

In chapter 1, the background information, problem statement about the breast cancer and information of deep learning is stated. Besides, the project objective and scope is discussed based on the problem statement. Moreover, the impact, significant and contribution, proposed method and achievement is stated.

In chapter 2 shows the discussion about the review on a few research papers, analysts and study the existing techniques and comparing their performance, strength, weaknesses, accuracy and so on.

In chapter 3 discussed about the system design, such as the type of datasets prepared, how to pre-pre-processing works on the datasets and model. Moreover, it also shows how to customize and modify the model and shows the process of training and model deployment.

In Chapter 4 show the system methodologies such as the flow of the Mask R-CNN network architecture, system requirement and specification.

In chapter 5 shows the system implementation and result. Its shows the steps of installation, training, evaluate and deployment of the Mask R-CNN model.

Chapter 2: Literature review

2.1 Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening

This approach relates to a system for breast cancer screening exam classification from high-resolution images by using a ResNet-based network as a building block of the model. The four different model's architecture used in the system due to the input images having the four standard views used in the mammography. Each of the models classifies whether it is malignant or not malignant and also whether the breast is benign or not benign for each side of the breast. The four models consist of view-wise model, image-wise model, side-wise model and join model.

The four models involve the view-wise model joins the R-CC and L-CC notations and R-MLO and L-MLO notations in order to separate predictions for the left breast and right breast views. The image-wise model predicts the result for each view of the mammography independently. The side-wise models concatenate the R-CC and R-MLO views, and concatenate the L-CC and L-MLO, so that to predict the benign and malignant for each breast separately. The last model is the joint model, which joins all the views and predicts whether it is benign or malignant.

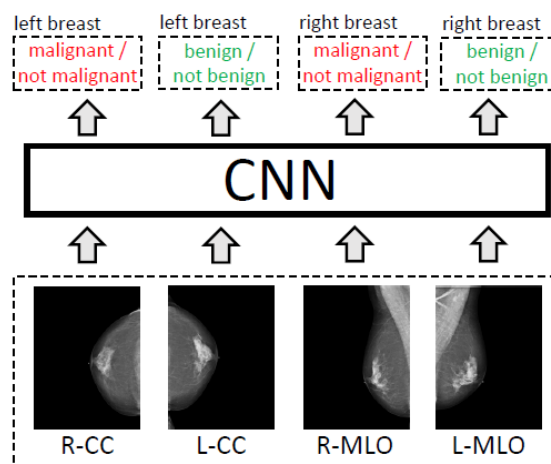


Figure 2.1.1 Four main tasks in a model

All of these four models consist of two main modules that use the same ResNet-based 22-layer networks, and this ResNet22 has a different balance of depth and width compared to the standard ResNet so that this network can compute very high-resolution images. The second module is these models having the same two fully connected layers mapping the computed hidden layer to the output layer. Furthermore, the pre-trained model applied to the Breast Imaging Reporting and Data System (BI-RADS), which is a scoring system for radiologists used to describe mammogram results into six categories. The three classes of BI-RADS are taken into consideration, the score 0 means the examination of mammogram is incomplete, category 1; means the result of mammograms is negative, that is no cancerous sign was found, category 2; benign finding.

The strengths of the system are the models able to compute the high-resolution images and achieve the high prediction result, and attain the high score in the area under the curve AUC of 0.895 in the prediction. The weakness of the system takes a long time in training four different models, and it takes about 12 hours to train to achieve the best performance (Wu, et al., 2019).

2.2 Multi-Class Breast Cancer Classification using Deep Learning CNN

An approach of using DenseNet with the TensorFlow framework to classify multi-class breast cancer. DenseNet is a new CNN architecture which is used for classification with fewer parameters. Besides that, DenseNet replaced the convolution layer and pooling layer into dense blocks from the traditional CNN model, but the transition layer remains the same except for the first convolution layer which concatenates all the features map in the previous layer instead of summing all the features like ResNet. The dense block in the DenseNet consists of convolution layer and non-linear layer. Furthermore, DenseNet uses some optimization techniques in the transition layer, such as batch normalization and dropout in order to reduce the size of the input images.

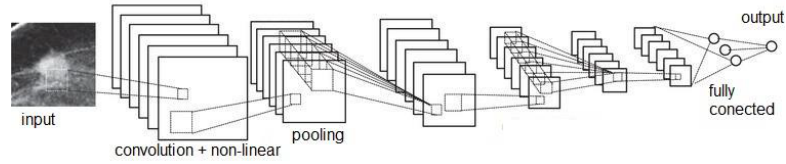


Figure 2.2.1 Traditional CNN model

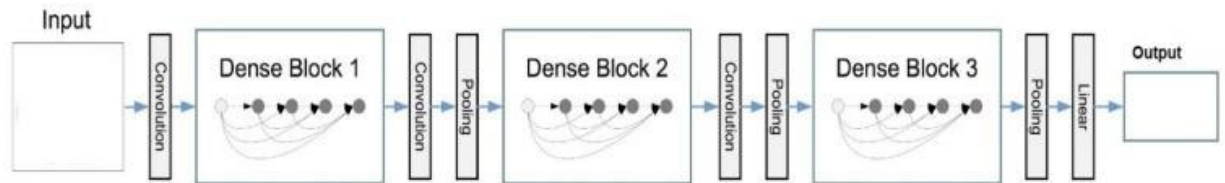


Figure 2.2.2 DenseNet with three dense blocks architecture

The limitation of the DenseNet is the DenseNet has a lower performance to deal with the microscopic images. To overcome this problem, it used the 7x7 size filter for the first convolutional layer to extract and detect a small portion of the important features. Then used a smaller filter size of the convolutional layer in every layer to address the complex images.

The strength of modified DenseNet models is the model achieves high performance with 96% accuracy predicted in the multi-class breast cancer classification (Nawaz, et al., 2018).

2.3 Breast Cancer Histopathological Image Classification: A Deep Learning Approach

This paper used an approach of transfer learning to deal with the problems of high computational cost, long computational time and insufficient number of high-quality labelled datasets. Transfer learning is a machine learning method which reuses pre-trained models on another task. This paper applied the pre-trained Inception models and ResNet models and fine-tuning the last layer to classify the different types of breast cancer histopathological images. Besides that, the research studied fine-tuning the last layer and fine-tuning all layers in the different frameworks of ResNet and Inception on the breast cancer microscopic images and comparing the results.

Model name	Epochs	ACC	TP	TN	FP	FN	P	AUC	S
Inception V1	3,000	0.917	580	252	14	54	0.976	0.917	0.914
Inception V2	3,000	0.848	516	252	14	118	0.973	0.874	0.813
Inception V3	3,000	0.884	568	244	22	66	0.962	0.869	0.895
Inception V4	3,000	0.871	542	258	8	92	0.985	0.905	0.854
ResNet V1 50	3,000	0.993	630	266	0	4	1	0.996	0.993
ResNet V1 101	3,000	0.995	625	272	1	1	0.998	0.994	0.998
ResNet V1 152	3,000	0.992	623	272	1	4	0.998	0.993	0.993

Table 2.3.1 Results of fine-tuned last layer in different models

Model name	Epochs	ACC	TP	TN	FP	FN	P	AUC	S
Inception V1	2,000	0.793	531	254	1	114	0.998	0.91	0.823
Inception V1	3,000	0.971	624	258	8	10	0.987	0.98	0.984
Inception V2	2,000	0.86	621	172	100	7	0.861	0.753	0.988
Inception V2	3,000	0.935	630	232	34	4	0.948	0.972	0.993
Inception V3	2,000	0.65	628	17	255	0	0.7118	0.753	1
Inception V3	3,000	0.764	632	88	178	2	0.78	0.842	0.996
Inception V4	2,000	0.851	618	172	94	16	0.867	0.86	0.974
Inception V4	3,000	0.877	633	159	108	0	0.854	0.855	1
ResNet V1 50	2,000	0.988	627	263	10	0	0.984	0.981	1
ResNet V1 50	3,000	0.998	627	272	1	0	0.998	0.996	1
ResNet V1 101	2,000	0.983	616	273	0	11	1	0.991	0.982
ResNet V1 101	3,000	0.996	626	273	0	1	1	0.999	0.998
ResNet V1 152	2,000	0.992	623	272	1	4	0.998	0.993	0.993
ResNet V1 152	3,000	0.996	626	273	0	1	1	0.999	0.998

Table 2.3.2 Results of fine-tuned all layers in different models

From the tables show that the accuracy results of the four models of ResNet that fine-tuned all layers are increased compared with the inception models. The highest accuracy in this research is the ResNet-50 and fine-tuned all layer's model, which achieved 99.8% accuracy. The results show that the performance of ResNet architecture with the fine-tuned all layers was more effective for the breast cancer classification (Motlagh, et al., 2018).

2.4 Segmentation of the Prostatic Gland and the Intraprostatic Lesions Using Mask R-CNN

This paper implements the Mask R-CNN model to perform segmentation on the prostatic gland and intraprostatic lesions (ILs) for prostate cancer detection. Besides that, this research also compared the performance between the Mask R-CNN, 2D U-net and 3D U-Net. Mask R-CNN is a kind of deep neural network architecture used to address the instance segmentation in computer vision. Instance segmentation performs classification and localization to detect objects in an image or a video by providing a bounding box, labelled classes and mask on the objects.

In this paper, the use of a ResNet 101-layer is used as a feature extractor to learn features activation map. Then, the features activation map proceeds to generate multi-scale activation maps in the Feature pyramid network (FPN). The multi-scale activation maps are then input and filtered by the region proposal network (RPN). RPN used to extract region proposals from the multi-scale activation maps, and it generated two outputs which are objectness scores showing the probability score that an anchor is an object and the bounding box refinement which is a regression for adjusting the anchors. Then the result from the RPN will apply to the ROIAlign algorithm. In the end, the branch of ROI classification involved the take of object detection and instance segmentation. The object detection branch will generate the features then compute the classification score and refine the bounding box again. The segmentation branch will perform convolution on each multi-scale feature map generated from FPN and then use the resultant activation map to generate the masks.

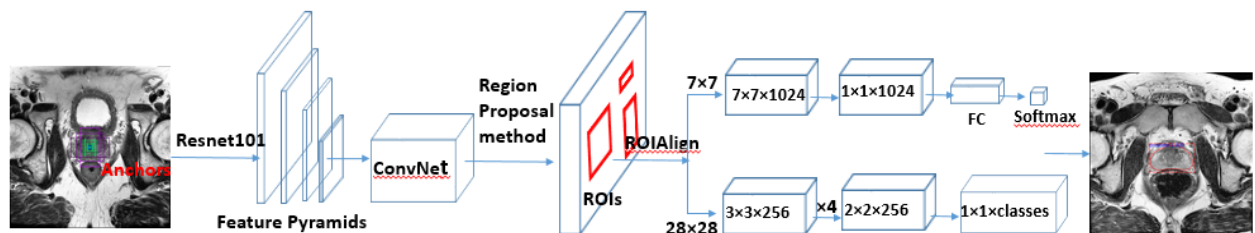


Figure 2.4.1 Mask R-CNN network architecture

In this paper, the evaluation metrics used include the dice similarity coefficient (DSC), Hausdorff distance (HD), sensitivity and specificity. The result of the prostate segmentation by using Mask R-CNN is achieved 0.86 DSC with the testing data which is better than the result using 2D U-Net and 3D U-Net which is 0.85 and 0.83 respectively (Dai, et al., 2019).

2.5 Ensemble Neural Networks for Digital Pathology Images Classification and Segmentation

This paper adopted the convolutional neural networks (CNN) for the 4-class tissues classification and segmentation task by using the Breast Cancer Histology Challenge dataset and achieved 90% accuracy results. Furthermore, the 2 architectures used for classification are ResNet 34 and DenseNet. In this research mentioned the DenseNet has a slight error rate than ResNet on the image dataset. Other than that, this experiment applied Adam optimizer, and this approach causes the performance increase by approximately 15%. In addition, this paper involves a strategy of segmentation. In the experiment, T-Net introduced a new architecture based on U-Net as in figure 2.5. This architecture performs additional convolution between the upsampling and downsampling's connection branches.

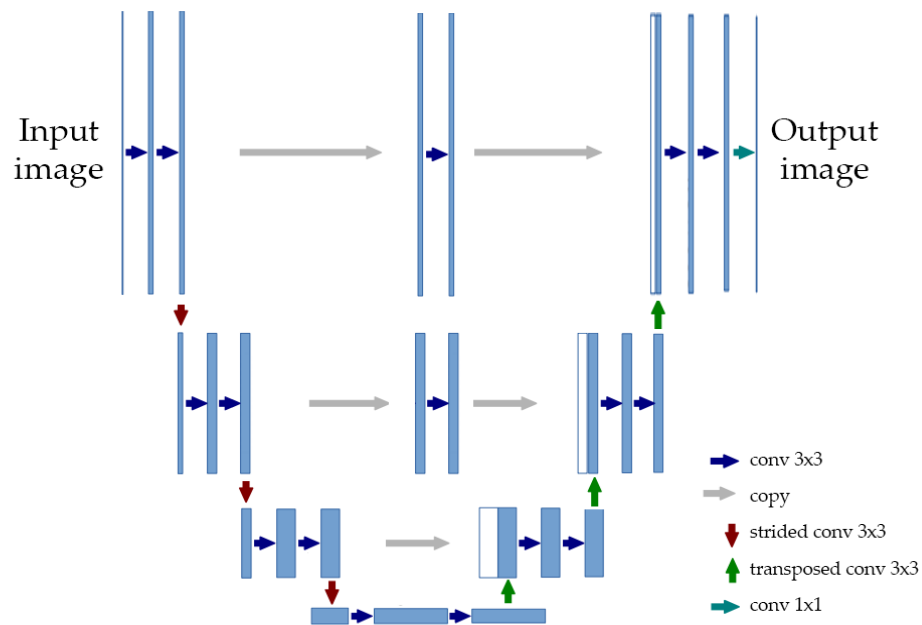


Figure 2.5.1 T-Net architecture

The strength of this method is the result of the ResNet 34 model has the highest accuracy result in 83% compared with the DenseNet 169 and DenseNet 201 achieved 78% and 80% respectively. Besides that, this paper mentioned this experiment task for the segmentation of the whole-slide image obtained controversial results and the results shown in the figure 2.5.2.

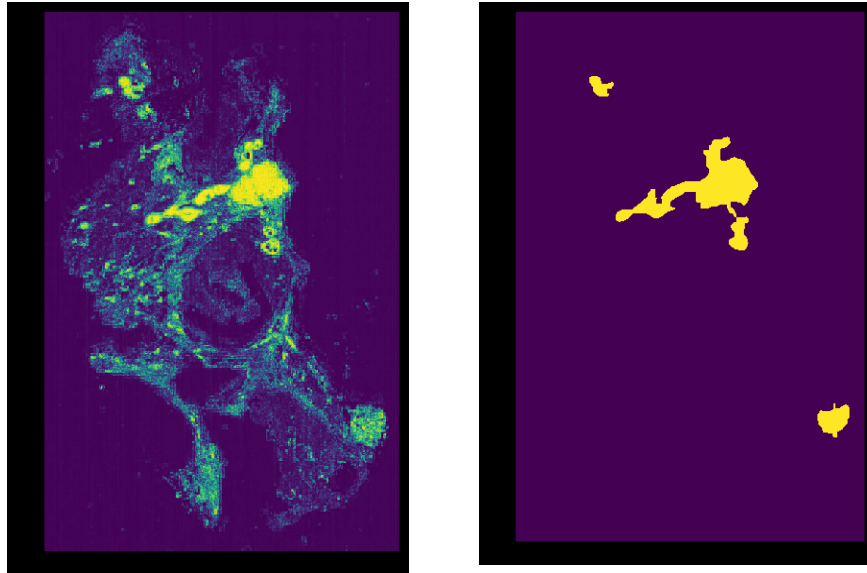


Figure 2.5.2 the Original microscopy images and the segmentation result

Chapter 3: System Design

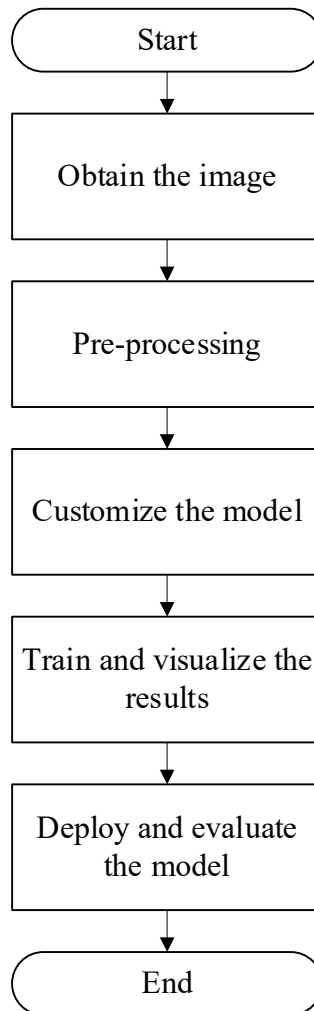


Figure 3.1.1 System Overview Flow Chart

3.1 Obtaining the image

The images collected for this project are the breast ultrasound images among women ages between 25 and 75 years old. The datasets consist of 780 greyscale images in PNG format with images size average in 500 x 500 pixels, and the dataset is categorized into three classes such as normal, benign, and malignant, and each image has its mask image. Besides that, the objective of this project is to classify the benign and malignant, so this project will take 180 benign and malignant images for training and validating. Thus, all of the images will be labelled based on the ground truth images (mask images) by using the online labelling tools, and the images will load into the system for the training process.

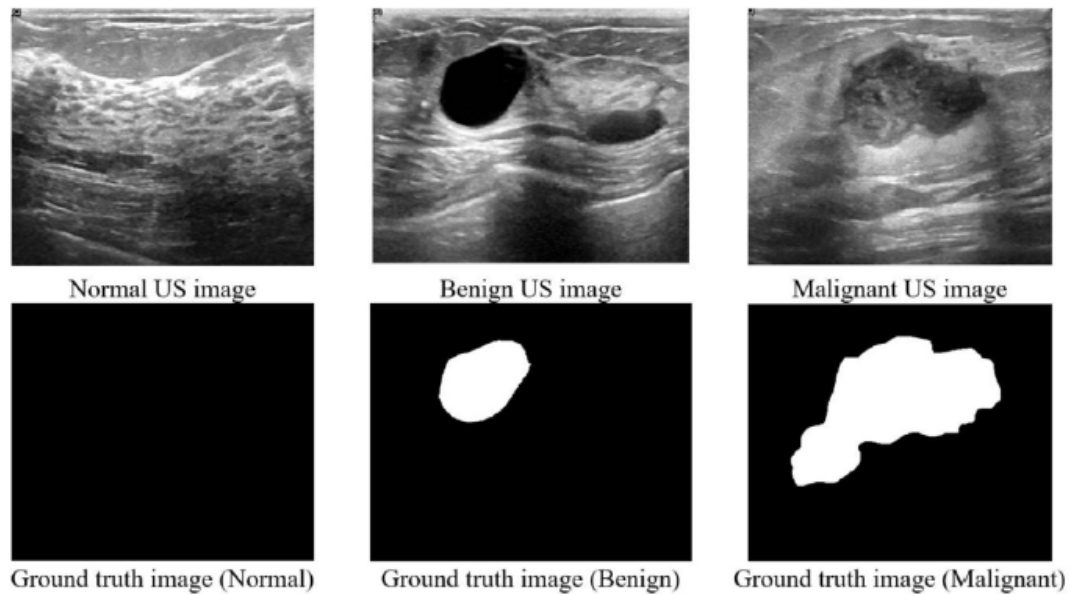


Figure 3.1.2 Sample of breast ultrasound images and mask images

3.2 Pre-processing of training datasets

Before training the model, pre-processing of the training images is a significant process. The different pre-processing method will affect the different results achieved.

3.2.1 Resize images

Resize all the training images into one size, so that the training process can support multiple images per batch. As figure 3.2.1 shown below, the shape of the image converted from the size of 584 x 655 into 1024x1024.


```

In [6]: # Load random image and mask.
image_id = np.random.choice(dataset.image_ids, 1)[0]
image = dataset.load_image(image_id)
mask, class_ids = dataset.load_mask(image_id)
original_shape = image.shape
# Resize
image, window, scale, padding, _ = utils.resize_image(
    image,
    min_dim=config.IMAGE_MIN_DIM,
    max_dim=config.IMAGE_MAX_DIM,
    mode=config.IMAGE_RESIZE_MODE)
mask = utils.resize_mask(mask, scale, padding)
# Compute Bounding box
bbox = utils.extract_bboxes(mask)

# Display image and additional stats
print("image_id: ", image_id, dataset.image_reference(image_id))
print("Original shape: ", original_shape)
log("image", image)
log("mask", mask)
log("class_ids", class_ids)
log("bbox", bbox)
# Display image and instances
visualize.display_instances(image, bbox, mask, class_ids, dataset.class_names)

image_id: 145 C:\Users\KimYew\Desktop\Mask_RCNN\datasets/train\malignant (66).png
Original shape: (584, 655, 3)
image          shape: (1024, 1024, 3)      min: 0.00000 max: 247.00000 uint8
mask           shape: (1024, 1024, 1)     min: 0.00000 max: 1.00000 uint8
class_ids      shape: (1,)               min: 2.00000 max: 2.00000 int32
bbox           shape: (1, 4)              min: 222.00000 max: 571.00000 int32

```

3.2.1 Resizing

3.2.2 Mini Mask

The training speed will be affected due to the instance binary mask become larger when training with high-resolution images. For instance, when training an image sized in 1024x1024, then the single instance mask requires 1MB of memory, if an image has 100 instances, then 100MB of memory is required for the masks alone. To shorter the training speed, the masks will be optimized by resizing the mask into a smaller size (56x56) and store the mask pixels that are inside the object bounding box, rather than a mask of the full image. Although resizing the mask will lose a bit of accuracy, but most of the object annotations are not very accurate in the beginning, so this loss is negligible for this practical purposes.

```

image_id = np.random.choice(dataset.image_ids, 1)[0]
image, image_meta, class_ids, bbox, mask = modellib.load_image_gt(
    dataset, config, image_id, use_mini_mask=False)

log("image", image)
log("image_meta", image_meta)
log("class_ids", class_ids)
log("bbox", bbox)
log("mask", mask)

display_images([image]+[mask[:, :, i] for i in range(min(mask.shape[-1], 7))])

```

image	shape: (1024, 1024, 3)	min: 0.00000	max: 255.00000	uint8
image_meta	shape: (15,)	min: 1.00000	max: 1024.00000	float64
class_ids	shape: (1,)	min: 2.00000	max: 2.00000	int32
bbox	shape: (1, 4)	min: 139.00000	max: 830.00000	int32
mask	shape: (1024, 1024, 1)	min: 0.00000	max: 1.00000	uint8




3.2.2 Image original size

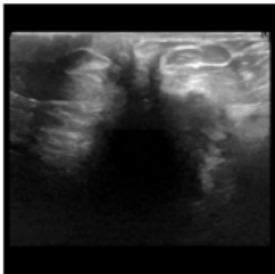

```

# Add augmentation and mask resizing.
image, image_meta, class_ids, bbox, mask = modellib.load_image_gt(
    dataset, config, image_id, augment=True, use_mini_mask=True)
log("mask", mask)
display_images([image]+[mask[:, :, i] for i in range(min(mask.shape[-1], 7))])

```

WARNING:root:'augment' is deprecated. Use 'augmentation' instead.

mask	shape: (56, 56, 1)	min: 0.00000	max: 1.00000	bool
------	--------------------	--------------	--------------	------

3.2.3 Resized images and performed augmentation

3.2.3 Transformation of images

Transformation the training images is an important stage when using a small number of datasets to train the model. The techniques used to transform the images is using data augmentation. Data augmentation is a technique of artificially generates more training samples to overcome the problem of small amounts of data available for training. There are a few methods to alter the training data such as horizontal flipping, cropping, rotation, colour shifting, etc. The augmentation applied in this project is transforming the input images by horizontally flip the training images randomly. Therefore, transform the images, it creates more new and different training samples to improves the training results.

3.3 Customize the model

In this process, the model implemented for this project is the mask R-CNN pre-trained model on Python, Keras, and TensorFlow for object detection and segmentation. This network is based on feature pyramid network and resnet101 backbone as its feature extractor. Besides that, the model was built for the instance segmentation task based on the cocodataset2017. The network previously was used for classifying for 80 classes. Therefore, the pre-train model will be customized to train our dataset and modify it for classifying the two classes in our datasets in order to achieve our objective.

3.3.1 Modify the configuration

The model is designed to work with RGB images. In this study, the datasets used for this project is training the grayscale images, which consist of one colour channel instead of 3 colour channels. Then, there are a few places of the code need to customize to accommodate our input. In the subclass of config, set the IMAGE_CHANNEL_COUNT to 1.

```
# Number of color channels per image. RGB = 3, grayscale = 1, RGB-D = 4
# Changing this requires other changes in the code. See the WIKI for more
# details: https://github.com/matterport/Mask\_RCNN/wiki
IMAGE_CHANNEL_COUNT = 1

# Image mean (RGB)
MEAN_PIXEL = np.array([123.7])
```

Figure 3.3.1 Modify Config.py

Besides that, the `load_image()` method in the Dataset class (`utils.py`) is designed for RGB datasets. When loading the images, it will convert the grayscale images into RGB images and also removes the fourth alpha channel if present for the consistency. To handle our datasets, override this method to convert the RGB images into grayscale images.

```
def load_image(self, image_id):
    """Load the specified image and return a [H,W,3] Numpy array.
    """
    # Load image
    image = skimage.io.imread(self.image_info[image_id]['path'])
    # If grayscale. Convert to RGB for consistency.
    #if image.ndim != 3:
    #    image = skimage.color.gray2rgb(image)
    if image.ndim != 1:
        image = skimage.color.rgb2gray(image)
    # If has an alpha channel, remove it for consistency
    if image.shape[-1] == 4:
        image = image[..., np.newaxis]
    return image
```

Figure 3.3.2 Modify Utils.py

3.3.2 Customize and modify the training code

In `CustomConfig()` method, it's a configuration for training on the breast cancer dataset. Based on figure 3.3.3, the `images_per_gpu` set as one since the smaller GPU is used in training. Besides that, the `num_classes` are set as three, which include the background, benign and malignant.

```

class CustomConfig(Config):
    """Configuration for training on the dataset.
    Derives from the base Config class and overrides some values.
    """
    # Give the configuration a recognizable name
    NAME = "breastcancer"

    # Adjust down if you use a smaller GPU.
    IMAGES_PER_GPU = 1

    # Number of classes (including background)
    NUM_CLASSES = 1 + 2 # Background + breastcancer

    # Number of training steps per epoch
    STEPS_PER_EPOCH = 100

    # Skip detections with < 90% confidence
    DETECTION_MIN_CONFIDENCE = 0.9

```

Figure 3.3.3 CustumConfig method

In the `load_custom()` method, under the dataset class. There are a few codes that need to modify in order to achieve the objective. In this study, the model should be able to classify breast cancer, whether it is malignant or benign. In the `load_custum()` method should add the class name and label. Based on the figure 3.3.4 shown, the benign class is labelled as 1, and the malignant class is labelled as 2.

```

# Add classes. We have two classes to add.
self.add_class("breastcancer", 1, "benign")
self.add_class("breastcancer", 2, "malignant")

```

Figure 3.3.4 Add two class in training code

3.3.3 Train and visualize the results

In this stage, the dataset is split into two parts, 90% of the dataset for training and the remaining 10% of the dataset used for validation. After that, train a new model starting from the pre-trained Imagenet weights. The reason for not using the pre-trained Coco weights due to the Cocodatasets classes is not including any classes for medical images datasets. During the training process, visualize the training result by using the Tensorboard. Tensorboard is an excellent debugging and visualization tool. The model is configured to log losses and save weights at the end of every epoch. By using the Tensorboard, every training result will show at every iteration of the training process. At the end of the training, choose the weight with the lowest loss so that it can achieve an accurate result. Besides that, Mask R-CNN there are 5 losses which are bounding box loss (mrcnn_bbox_loss), classification loss (mrcnn_class_loss), mask loss (mrcnn_mask_loss), RPN bounding box loss (rpn_bbox_loss), and RPN classifier loss (rpn_class_loss). In addition, the loss of the Mask R-CNN means the summation of the loss of classification, localization (bounding box loss) and segmentation mask. The loss of RPN is the addition of RPN classification loss and RPN bounding box loss ($rpn_class_loss + rpn_bbox_loss$). Lastly, the total loss is the total of RPN loss and Mask R-CNN loss.

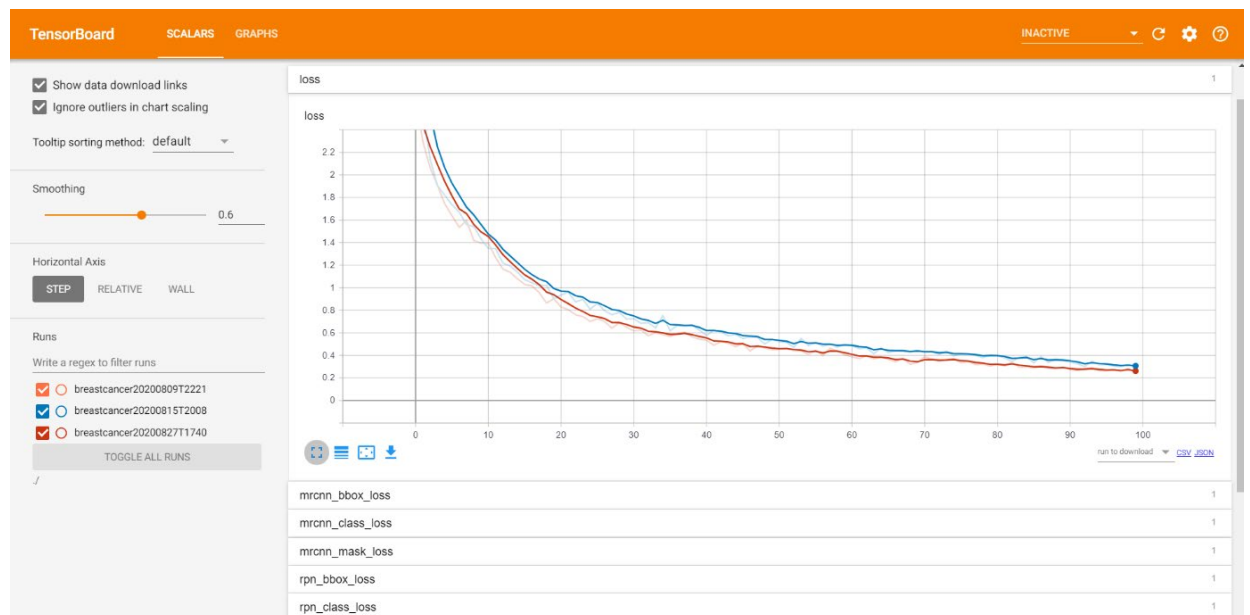


Figure 3.3.5 TensorBoard show the loss at every epoch

3.4 Deploy and evaluate the model

In this stage are focusing on testing, debugging and evaluating the trained model previously by using the Jupyter notebook. After that, load the trained model and the validation dataset, we can run the detection. During the process of detecting the validation image, the input image will randomly choose from the 20 validation images to perform testing. After detecting the input image, the image information will be shown before the detection result, such as the image information and the actual label. The detected results will show the predicted label, segmentation mask, and bounding box in the image. The label shows the image either is benign or malignant. The masks will colour the recognized cancerous region, and the bounding box will show the location of cancer.

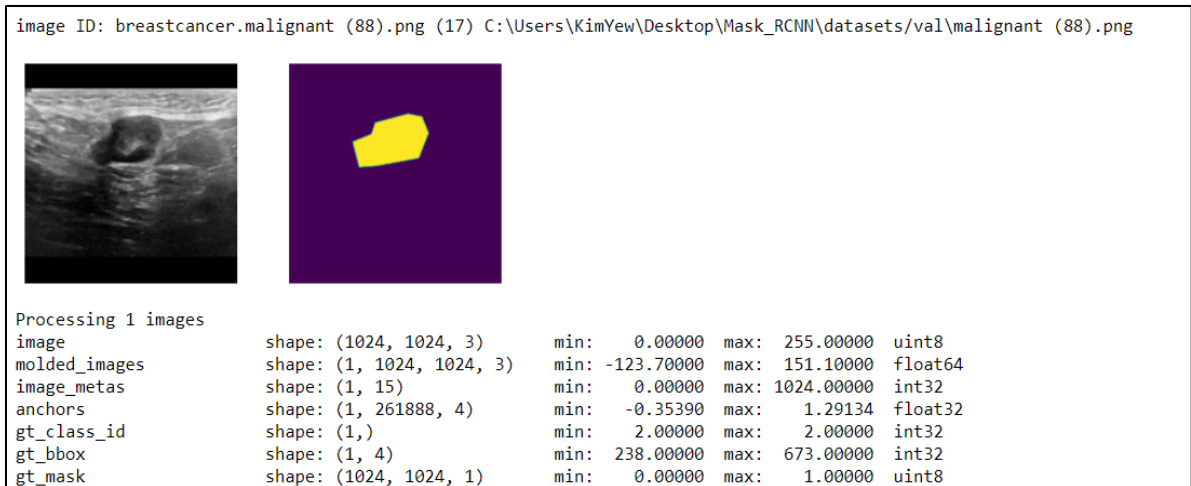


Figure 3.4.1 Image information and the actual label

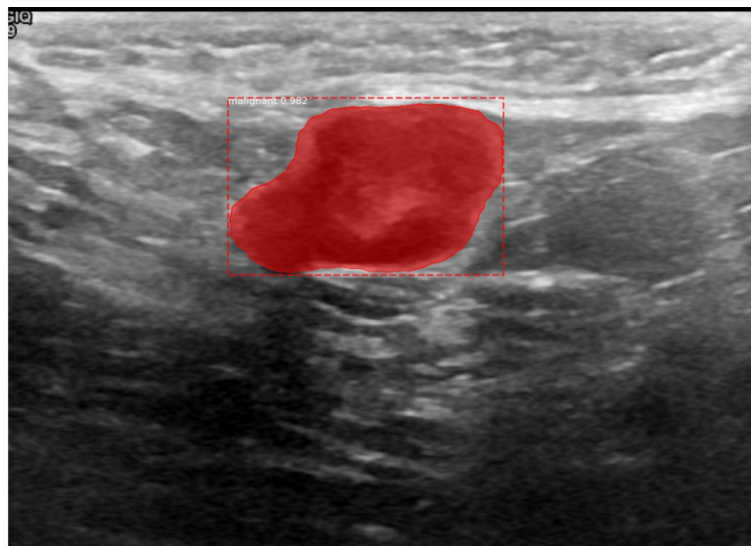


Figure 3.4.2 Prediction result

To testing the model, not only test the image from the 20 validation images. This project also used other breast ultrasound images for testing the model as well. However, the other images don't have the label, so the actual result is loaded from the mask images, and it used to compare the prediction results.

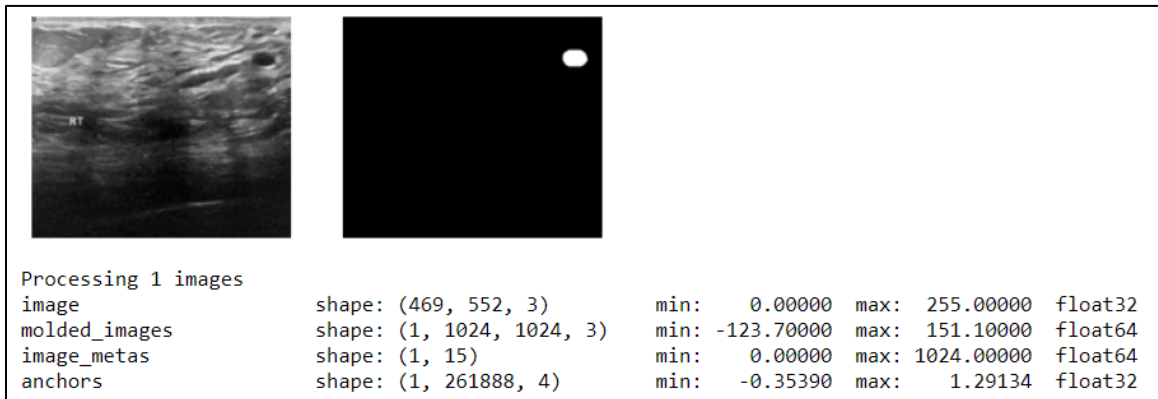


Figure 3.4.3 Actual image and the mask image

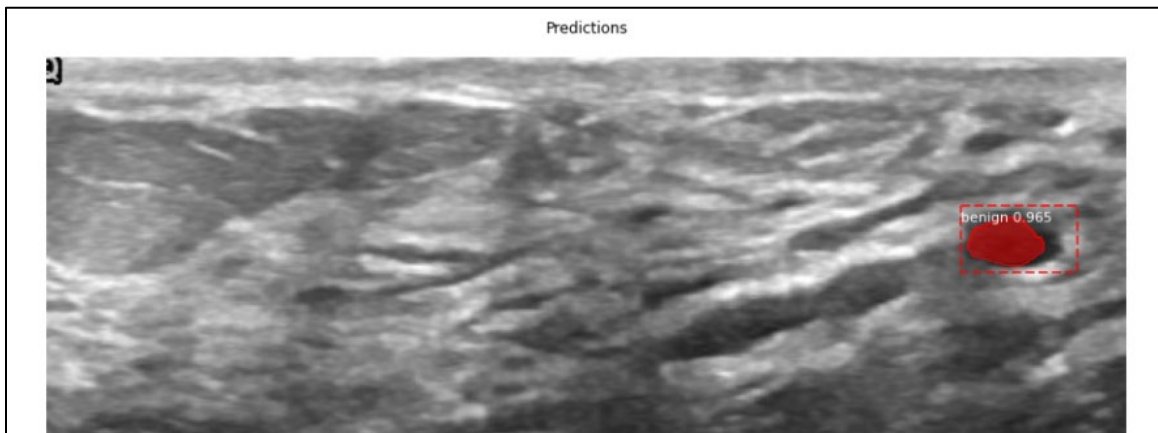


Figure 3.4.4 Prediction result

Chapter 4: System Methodology

4.1 Methodologies and tools

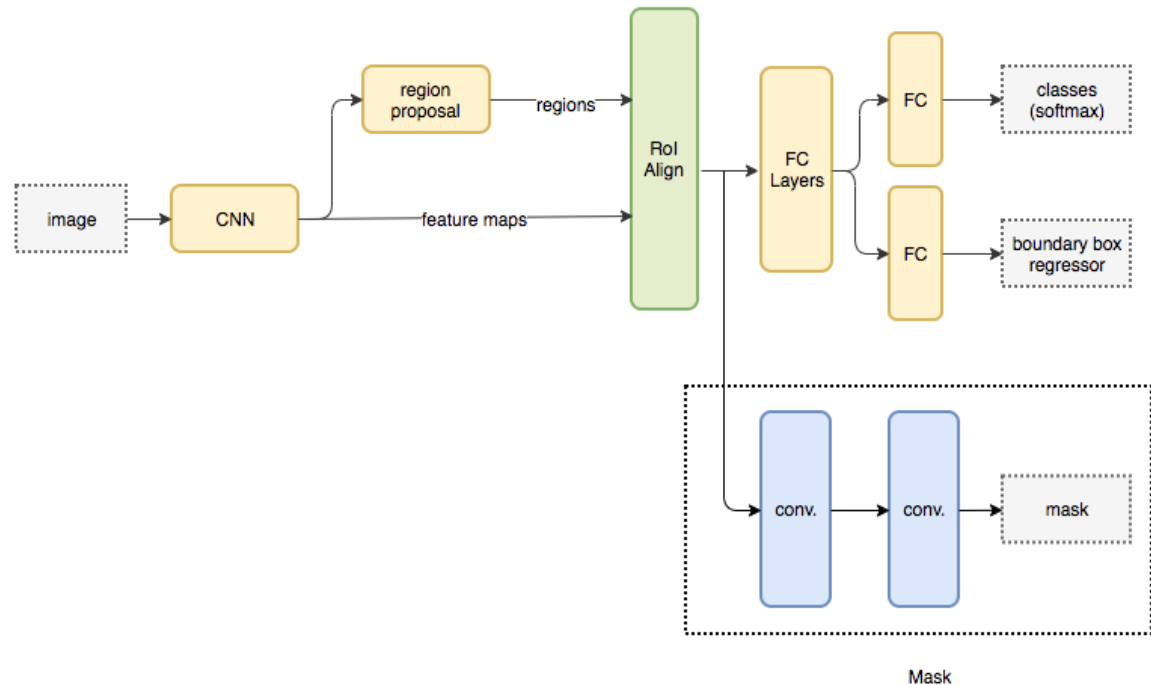


Figure 4.1 Basic structure of the Mask R-CNN architecture

The Mask R-CNN is built on top of the Faster R-CNN, and the network architecture is basically the same. The difference is the ROI Align layer and the additional mask layer for object instance segmentation. Recently, the Mask R-CNN consider is a popular and one of the best models for object instance segmentation and its launch for the COCO dataset.

In the first stage, the model uses a convolution neural network CNN as a feature extractor to extract features for the whole image. CNN works like our eyes to detect colours, edges, and shapes of an object. There are a few steps of CNN for feature extraction. Firstly, take an image as an input. After that, divide the image into several regions using the sliding window technique, such as divide a 100 x 100 pixel image into 10 x 10 regions, and each region considers separated images. Then, process these regions into different colours.

Lastly, combine each region into original image called feature maps with the detected edges.

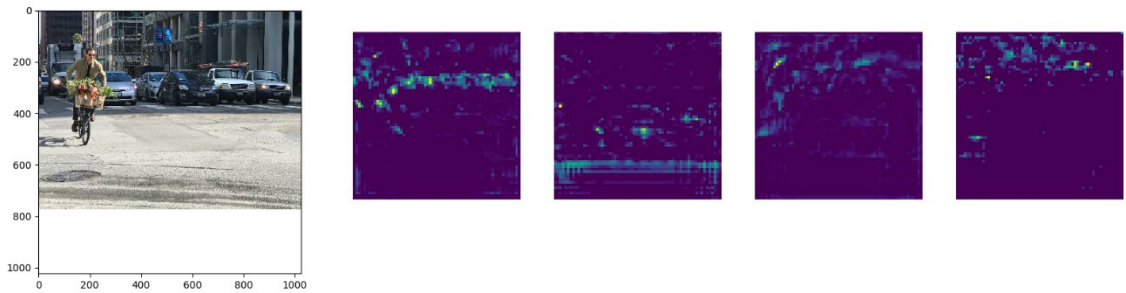


Figure 4.1.2 Feature maps for the RPN

In the second layer, the model applies region proposal network RPN to create object proposals such as the Region of Interests ROIs along with their abjectness score. ROIs is the extracted boxes from an image which contain any object.

After that, ROI Align is used in Mask R-CNN instead of using ROI Pooling in Faster R-CNN. Then the ROIs and corresponding feature maps are the input of the ROI Align layer. When using ROI Pooling has a problem of location misalignment caused by quantization. Therefore, using ROI Align the ROI is mapped accurately from the image on to the feature map instead of rounding up to integers by using ROI pooling. ROI Align makes a significant improvement in the accuracy compare to using ROI pooling.

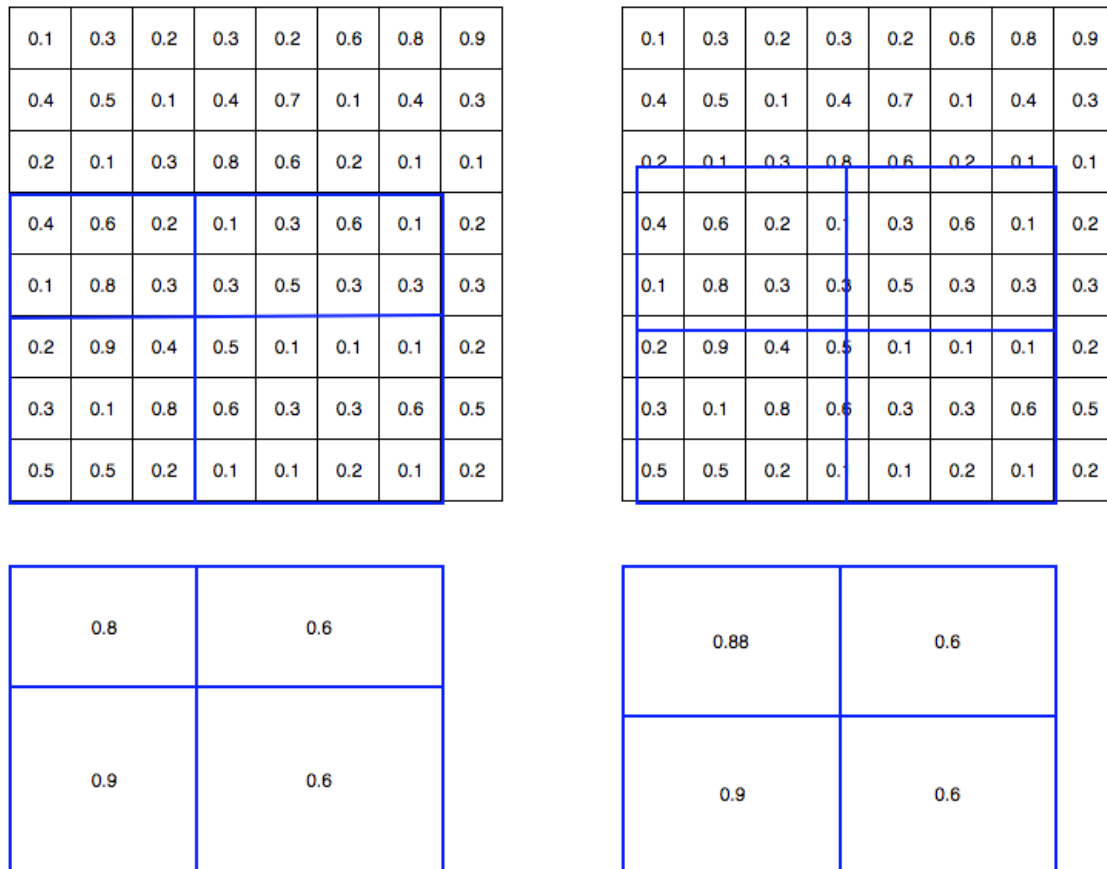


Figure 4.1.3 Roi Pooling (left) vs Roi Align (right)

In the fully connected network, the softmax layer is used on the top of the network to predict the classes. Besides that, a linear regression layer is also used parallelly to output the bounding box coordination for the detected object.

From the output of the ROI Align layer also go separately to a few convolutional layers to perform mask prediction.

4.2 System Requirement

4.2.1 Hardware

Operating System	Windows 10 64-bit
Processor	AMD Ryzen 5 3600 @ 3.60GHz
RAM	16.00GB
Graphics	Nvidia GTX 1660ti 6GB
Storage	1TB

Table 3.1.1 Hardware used in this project

4.2.2 Software

The software that has been used for developing this system are:

1. Anaconda

In this project, Anaconda is used for training model and it a free open-source distribution of the R programming language and Python languages for data science, machine learning applications, etc. For instance, it provides the Jupyter Notebook, which is the web-based interactive computing notebook environment for the Python code.

2. TensorFlow

TensorFlow is a machine learning framework developed by Google which has facilitated application development and machine learning research. TensorFlow mostly used to design and train deep learning models. Furthermore, it is an open-source software library, and it also provides a collection of workflows to build machine learning model using python.

3. Keras

Keras library is used in this project, and it is an open-source neural network library written in Python language. Besides that, its able the run on top on the TensorFlow. Its mainly designed for fast experimentation with deep neural networks.

4.2.3 User Requirements

- User shall have a basic understanding of the type of breast cancer image.
- User shall identify the images that have been successfully diagnosed by the system
- User shall have a basic knowledge of identifying the breast cancer

4.3 Specification

4.3.1 System Performance Definition

The system performance of this project is targeted to reach a high accuracy result in the process of classification and segmentation of a grayscale image. The mask R-CNN pre-trained model requires an optimal modification to fit the dataset for training and also to enhance accurate performance. Besides the model, the pre-process of datasets also needs to be more optimized so that it will increase the model's performance further.

4.3.2 Evaluation

Object detection evaluation

In this work, the system should be able to classify cancerous areas in breast ultrasound images with high accuracy and consistency. However, the accuracy result may be affected by false detection and other different circumstances during the classification process. For example, the classification model overfitting the datasets, the number of input images for training the model is insufficient and etc. Hence, there is an evaluation metric used to measure the quality and the performance of the models, which is the intersect over union (IoU) method to evaluate the accuracy of the object detection. The IoU value is computing the area of the overlap over the area of the union as in figures 4.3.1.

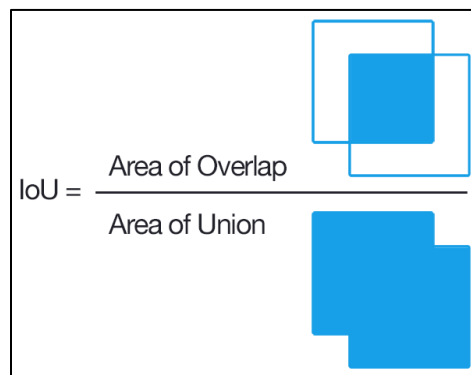


Figure 4.3.1 IoU formula

In figure 4.3.2, the higher value of the IoU means that the predicted bounding boxes are heavily overlapping with the ground-truth bounding boxes. In simple words, the high IoU (more than 0.5) values mean that the object detection as true positive. If IoU less than 0.5, then detected as false negative.



Figure 4.3.2 Example of computing IoU for various bounding box

Loss function in Mask R-CNN

As mentioned before, the simple way to measure the performance of the Mask R-CNN is using the loss function shows the accuracy of the model. The lower lost having a better result. The loss in Mask R-CNN consists of two, the first loss is due to the RPN, and the another is due to classification, segmentation mask and localization as shown in figure 4.3.3. Therefore, the optimization problem is to minimize the total loss in order to achieve the objective in this project.

$$1. \text{Loss(RPN)} = \text{RPN_Class Loss} + \text{RPN_BBox Loss}$$

$$2. \text{Loss(Mask R-CNN)} = \text{Loss(class labels prediction)} + \text{Loss(Bounding Box prediction)} + \text{Loss (Mask Prediction)}$$

$$\text{Total Loss} = \text{Loss(RPN)} + \text{Loss(Mask R-CNN)}$$

Figure 4.3.3 Loss calculation

4.4 Project Timeline

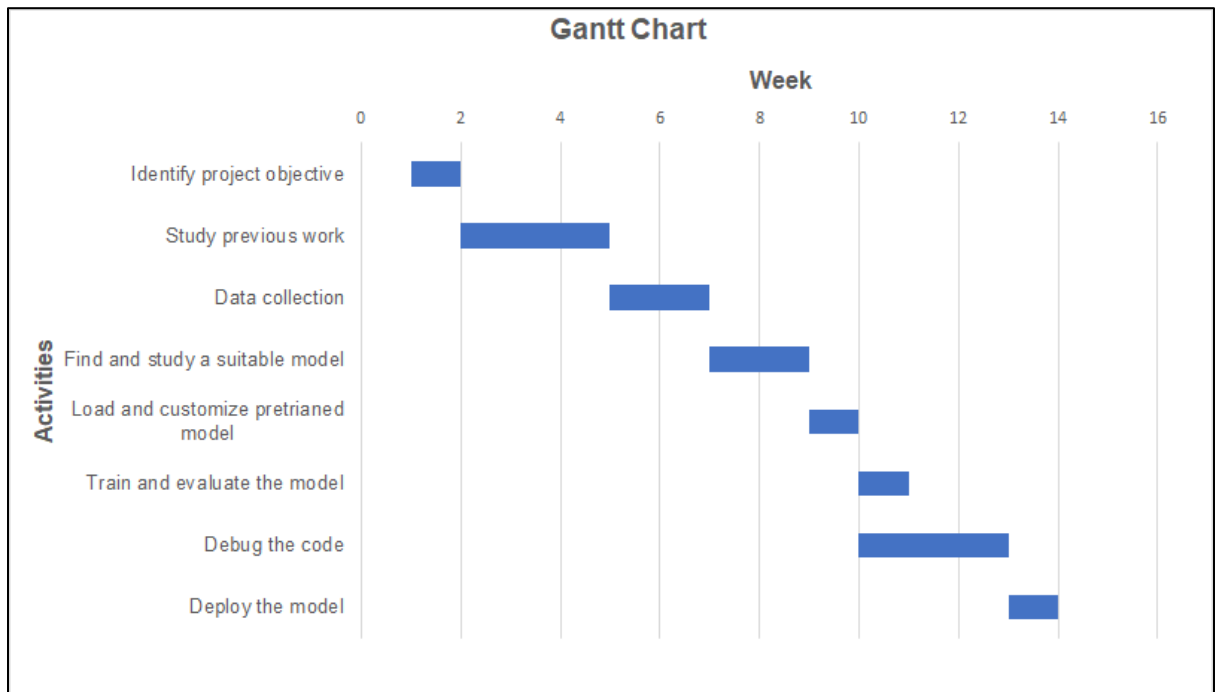


Figure 4.4.1 Gantt Chart for the project development timeline (Jan to May semester)

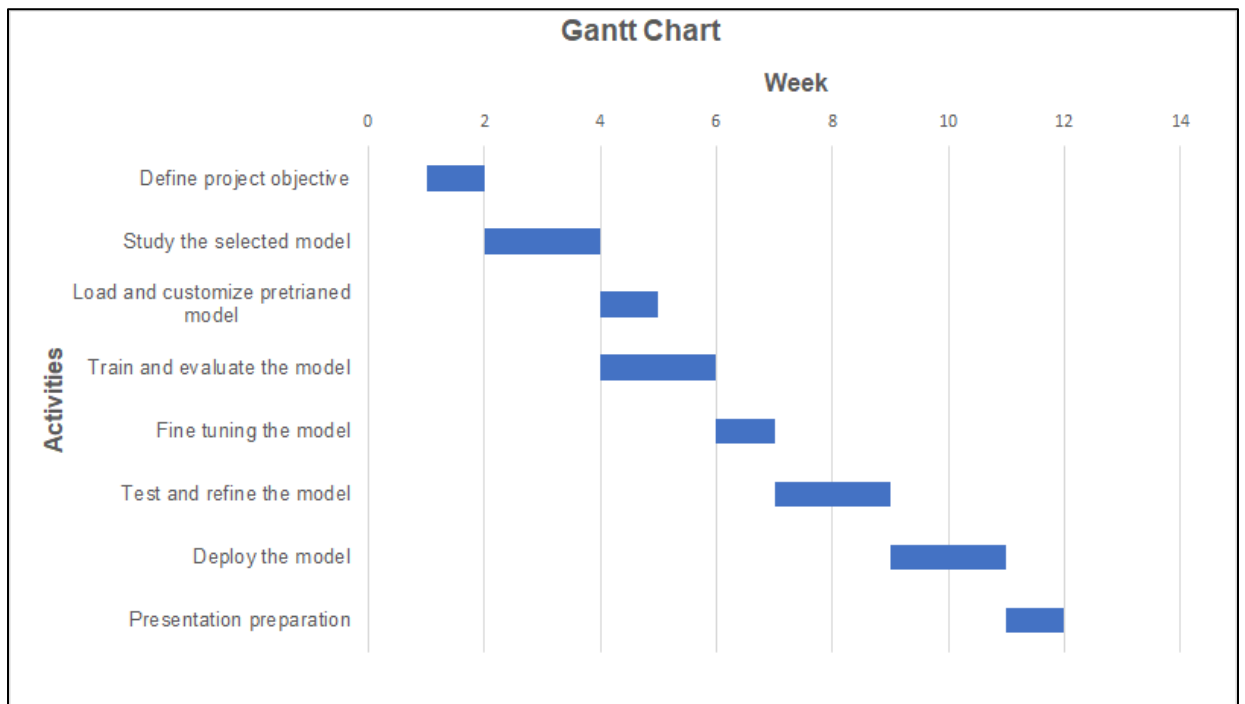


Figure 4.4.2: Gantt Chart for the project development timeline (June to October semester)

Figure 4.4.1 shows the Gantt Chart for the development timeline for the previous semester, and the timeline shows this project is more focused and take a longer time on studying the previous work and deep learning model. Furthermore, debugging and modify the code needs to take longer because customizing the pre-trained model is easily prone to error and needs time to debug and repeat the process of training and testing the model.

Figure 4.4.2 shows the timelines for the project development timeline in the current semester. It should take longer time and focus on understanding the selected pre-trained model in order to achieve better performance and achieve the project objective. Furthermore, the testing and deployment phase needs to take approximately two weeks to complete to make sure that the prediction result can be more accurate and achieve the project objectives.

Chapter 5: Implementation and result

5.1 Dataset preparation

First of all, collect datasets and divide into two sets consisting of 160 train and 20 validation images. Besides that, the data consists of two classes which are benign and malignant. After that, follow the ground truth images to label the images by using online labelling tool. Online labelling tool <http://www.robots.ox.ac.uk/~vgg/software/via/via-1.0.6.html>

In figure 5.1.1, label 1 stands for benign class and label 2 stands for malignant. Then save the JSON file into the train folder and repeat the step for the validation images.

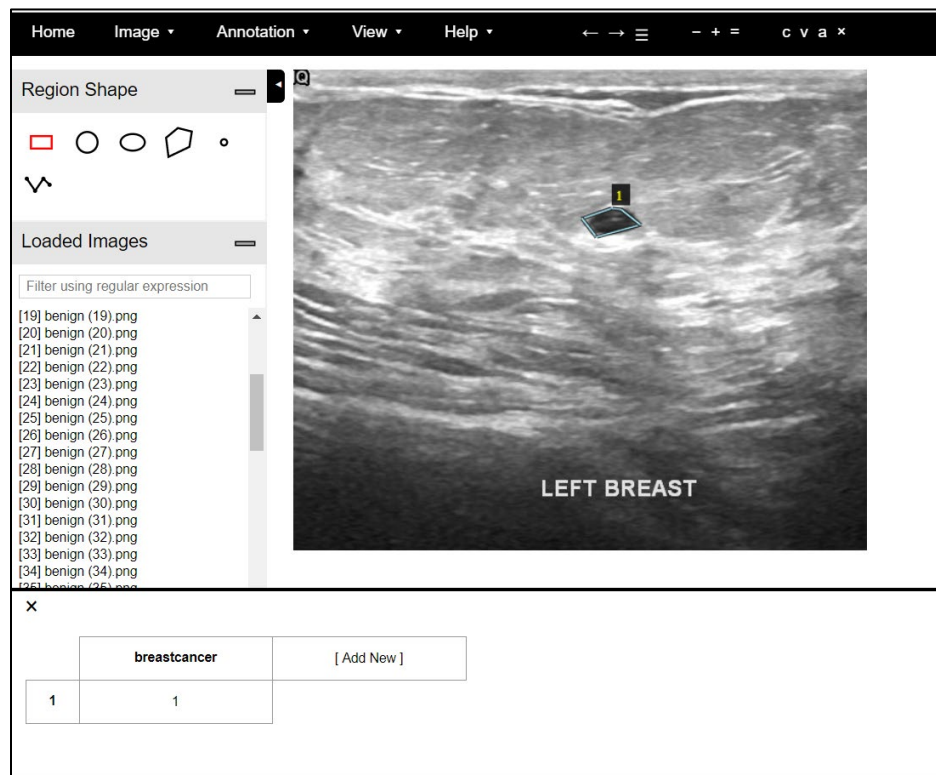


Figure 5.1.1 Online labelling tool

5.2 Mask R-CNN installation

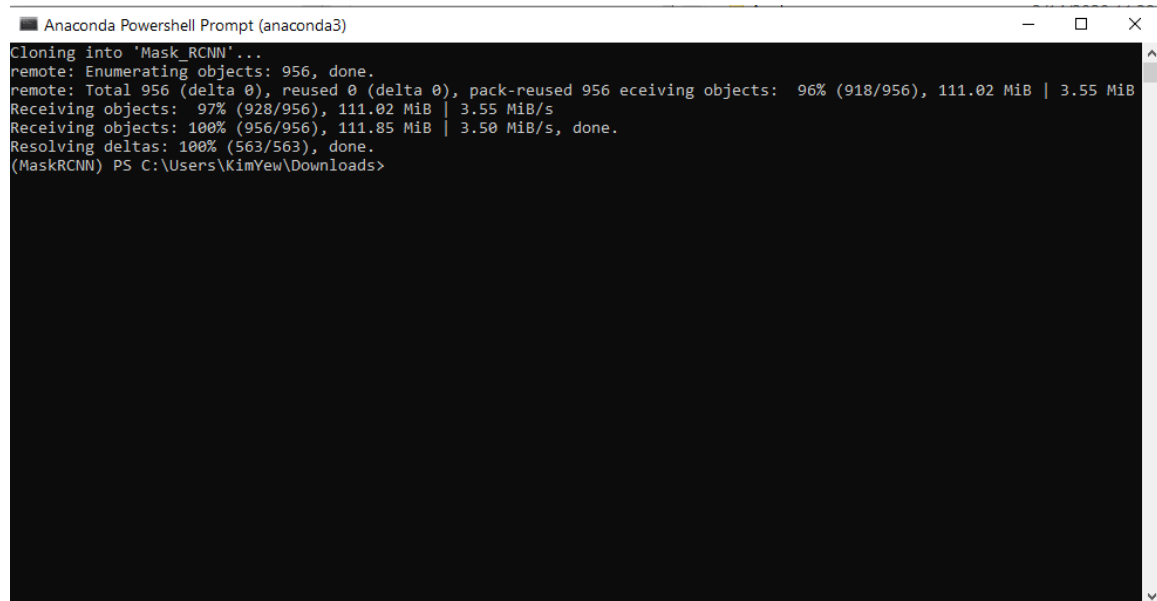
There are a few steps to install the Mask R-CNN. The first step is creating a virtual environment to install the Mask R-CNN so that this won't affect another environment.

Run command > `conda create -n MaskRCNN python=3.6 pip.`

To activate the virtual environment. Run command > `conda activate MaskRCNN.`

Secondly, clone or download the repository from Github by using command and the folder saved in named Mask_RCNN.

Run command > `git clone https://github.com/matterport/Mask_RCNN.git`

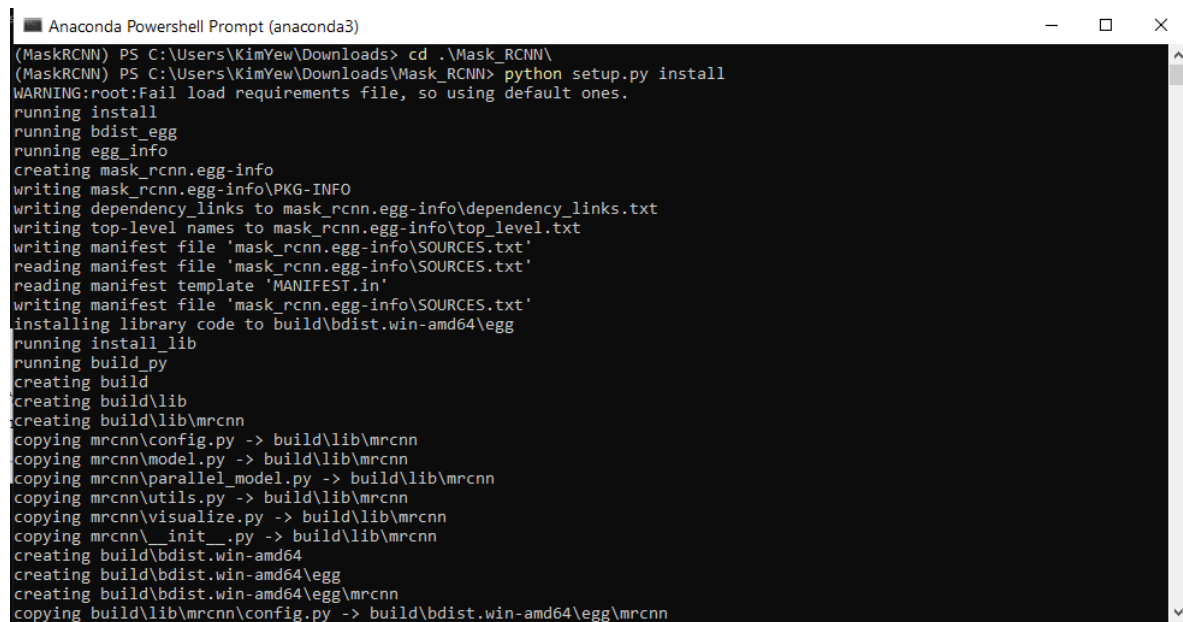


```
Anaconda Powershell Prompt (anaconda3)
Cloning into 'Mask_RCNN'...
remote: Enumerating objects: 956, done.
remote: Total 956 (delta 0), reused 0 (delta 0), pack-reused 956
Receiving objects: 96% (918/956), 111.02 MiB | 3.55 MiB/s
Receiving objects: 100% (956/956), 111.85 MiB | 3.50 MiB/s, done.
Resolving deltas: 100% (563/563), done.
(MaskRCNN) PS C:\Users\KimYew\Downloads>
```

Figure 5.2.1 Clone Mask RCNN source code

In cmd, change to Mask_RCNN directory and run install script command

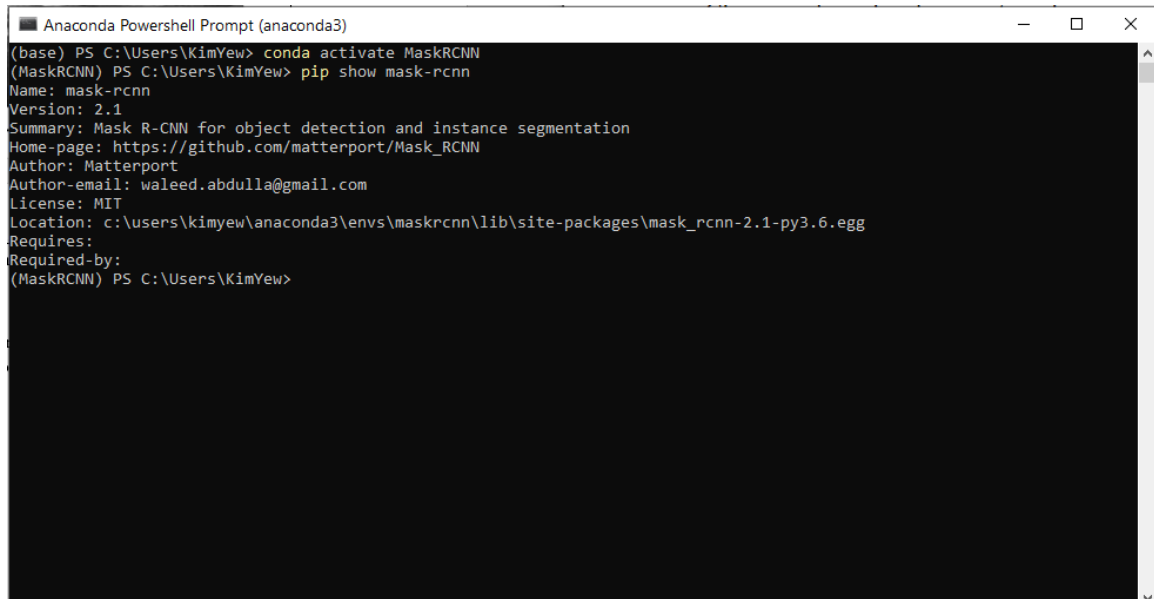
> `python setup.py install.`



```
Anaconda Powershell Prompt (anaconda3)
(MaskRCNN) PS C:\Users\KimYew\Downloads> cd .\Mask_RCNN\
(MaskRCNN) PS C:\Users\KimYew\Downloads\Mask_RCNN> python setup.py install
WARNING:root:Fail load requirements file, so using default ones.
running install
running bdist_egg
running egg_info
creating mask_rcnn.egg-info
writing mask_rcnn.egg-info\PKG-INFO
writing dependency_links to mask_rcnn.egg-info\dependency_links.txt
writing top-level names to mask_rcnn.egg-info\top_level.txt
writing manifest file 'mask_rcnn.egg-info\SOURCES.txt'
reading manifest file 'mask_rcnn.egg-info\SOURCES.txt'
reading manifest template 'MANIFEST.in'
writing manifest file 'mask_rcnn.egg-info\SOURCES.txt'
installing library code to build\bdist.win-amd64\egg
running install_lib
running build_py
creating build
creating build\lib
creating build\lib\mrcnn
copying mrcnn\config.py -> build\lib\mrcnn
copying mrcnn\model.py -> build\lib\mrcnn
copying mrcnn\parallel_model.py -> build\lib\mrcnn
copying mrcnn\utils.py -> build\lib\mrcnn
copying mrcnn\visualize.py -> build\lib\mrcnn
copying mrcnn\__init__.py -> build\lib\mrcnn
creating build\bdist.win-amd64\egg
creating build\bdist.win-amd64\egg\mrcnn
copying build\lib\mrcnn\config.py -> build\bdist.win-amd64\egg\mrcnn
```

Figure 5.2.2 Run setup.py script

After installed, run the command as shown in the figure 5.2.3 to check whether Mask RCNN is installing successfully.



```

Anaconda Powershell Prompt (anaconda3)
(base) PS C:\Users\KimYew> conda activate MaskRCNN
(MaskRCNN) PS C:\Users\KimYew> pip show mask-rcnn
Name: mask-rcnn
Version: 2.1
Summary: Mask R-CNN for object detection and instance segmentation
Home-page: https://github.com/matterport/Mask_RCNN
Author: Matterport
Author-email: waleed.abdulla@gmail.com
License: MIT
Location: c:\users\kimyew\anaconda3\envs\maskrcnn\lib\site-packages\mask_rcnn-2.1-py3.6.egg
Requires:
Required-by:
(MaskRCNN) PS C:\Users\KimYew>
```

Figure 5.2.3: Mask RCNN installed

Besides that, install the dependencies listed in the requirements text file such as the TensorFlow, Keras, cython, etc.

Run command > `pip install -r requirements.txt`

After that, install the pycocotools, and this tool requires to install Visual C++ 2015 Build Tools.

Run command > `git clone https://github.com/philferriere/cocoapi.git`

> `pip to install pycocotools`

> `pip install`

`git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI`

```
Anaconda Powershell Prompt (anaconda3)
(MaskRCNN) PS C:\Users\KimYew\Downloads\Mask_RCNN> git clone https://github.com/philferriere/cocoapi.git
Cloning into 'cocoapi'...
remote: Enumerating objects: 994, done.
remote: Total 994 (delta 0), reused 0 (delta 0), pack-reused 994 eceiving objects: 96% (955/994), 10.63 MiB | 2.31 MiB/s
Receiving objects: 100% (994/994), 12.26 MiB | 2.41 MiB/s, done.
Resolving deltas: 100% (589/589), done.
(MaskRCNN) PS C:\Users\KimYew\Downloads\Mask_RCNN> pip install pycocotools
Requirement already satisfied: pycocotools in c:\users\kimyew\anaconda3\envs\maskrcnn\lib\site-packages (2.0)
(MaskRCNN) PS C:\Users\KimYew\Downloads\Mask_RCNN> pip install git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI
Collecting git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI
  Cloning https://github.com/philferriere/cocoapi.git to c:\users\kimyew\appdata\local\temp\pip-req-build-lhyn8rtt
  Running command git clone -q https://github.com/philferriere/cocoapi.git 'C:\Users\KimYew\AppData\Local\Temp\pip-req-build-lhyn8rtt'
Requirement already satisfied (use --upgrade to upgrade): pycocotools==2.0 from git+https://github.com/philferriere/cocoapi.git#subdirectory=PythonAPI in c:\users\kimyew\anaconda3\envs\maskrcnn\lib\site-packages
Building wheels for collected packages: pycocotools
  Building wheel for pycocotools (setup.py) ... |
```

Figure 5.2.4 Install pycocotools

5.3 Train model

After preparing the dataset and installed the environment, we can start training the model. Run the command followed by your directory path for the datasets. The time for the training takes 3 to 4 hours approximately.

Run command > `python breastcancer.py train --dataset=C:\Users\KimYew\Desktop\Mask_RCNN\datasets --weights=imagenet`

```
Anaconda Powershell Prompt (anaconda3)
(MaskRCNN) PS C:\Users\KimYew\Desktop\Mask_RCNN\samples\breastcancer> python breastcancer.py train --dataset=C:\Users\KimYew\Desktop\Mask_RCNN\datasets --weights=imagenet
2020-09-09 20:09:57.586855: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library cudart64_100.dll
Using TensorFlow backend.
Weights: imagenet
Dataset: C:\Users\KimYew\Desktop\Mask_RCNN\datasets
Logs: C:\Users\KimYew\Desktop\Mask_RCNN\logs

Configurations:
BACKBONE                resnet101
BACKBONE_STRIDES         [4, 8, 16, 32, 64]
BATCH_SIZE               1
BBOX_STD_DEV             [0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE  None
DETECTION_MAX_INSTANCES 100
DETECTION_MIN_CONFIDENCE 0.9
DETECTION_NMS_THRESHOLD 0.3
FPN_CLASSIF_FC_LAYERS_SIZE 1024
GPU_COUNT                1
GRADIENT_CLIP_NORM       5.0
IMAGES_PER_GPU           1
IMAGE_CHANNEL_COUNT       3
IMAGE_MAX_DIM            1024
IMAGE_META_SIZE          15
IMAGE_MIN_DIM            800
IMAGE_MIN_SCALE          0
IMAGE_RESIZE_MODE         square
IMAGE_SHAPE              [1024 1024 3]
LEARNING_MOMENTUM         0.9
```

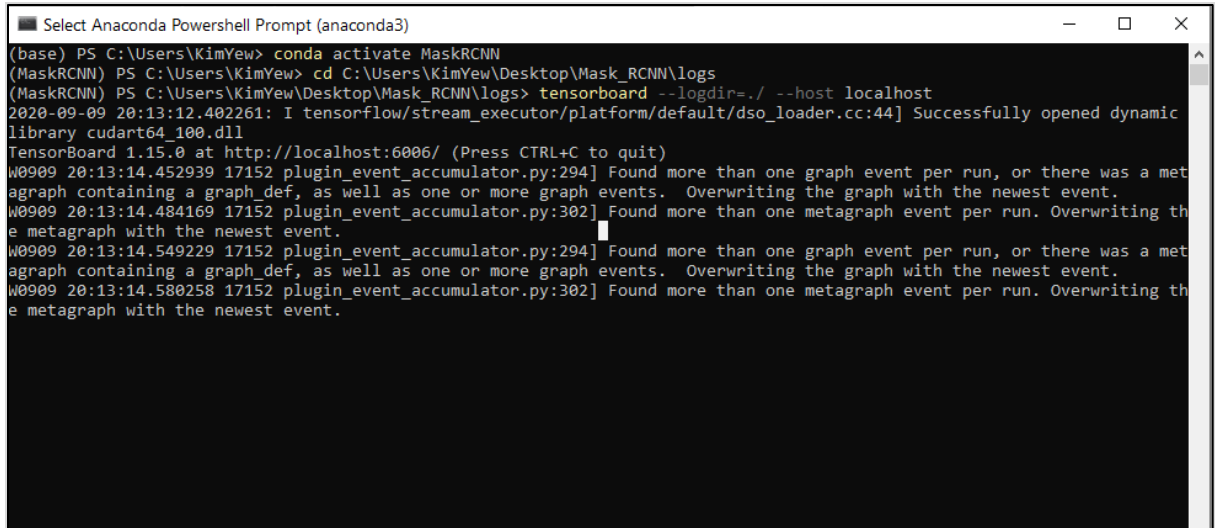
Figure 5.3.1 Train model command

5.4 Visualize the result

During the training, we can visualize the training result by using TensorBoard visualization tool. Firstly, change the directory path to the log folder. Then run the following command.

Then, copy the link to access the TensorBoard by using the browser.

Run command > `tensorboard --logdir=./ --host localhost`



```
Select Anaconda Powershell Prompt (anaconda3)
(base) PS C:\Users\KimYew> conda activate MaskRCNN
(MaskRCNN) PS C:\Users\KimYew> cd C:\Users\KimYew\Desktop\Mask_RCNN\logs
(MaskRCNN) PS C:\Users\KimYew\Desktop\Mask_RCNN\logs> tensorboard --logdir=./ --host localhost
2020-09-09 20:13:12.402261: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic
library cudart64_100.dll
TensorBoard 1.15.0 at http://localhost:6006/ (Press CTRL+C to quit)
W0909 20:13:14.452939 17152 plugin_event_accumulator.py:294] Found more than one graph event per run, or there was a met
agraph containing a graph_def, as well as one or more graph events. Overwriting the graph with the newest event.
W0909 20:13:14.484169 17152 plugin_event_accumulator.py:302] Found more than one metagraph event per run. Overwriting th
e metagraph with the newest event.
W0909 20:13:14.549229 17152 plugin_event_accumulator.py:294] Found more than one graph event per run, or there was a met
agraph containing a graph_def, as well as one or more graph events. Overwriting the graph with the newest event.
W0909 20:13:14.580258 17152 plugin_event_accumulator.py:302] Found more than one metagraph event per run. Overwriting th
e metagraph with the newest event.
```

Figure 5.4.1 Log TensorBoard command

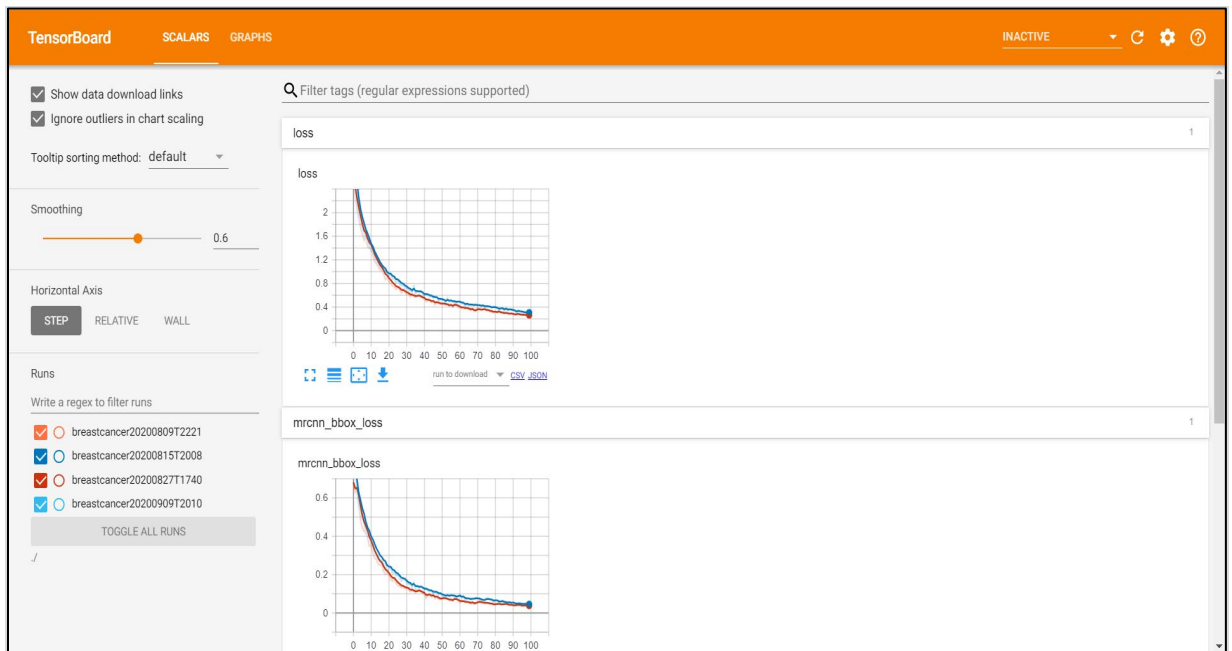


Figure 5.4.2 TensorBoard

Based on figure 5.4.2 shows the result obtain during the training, the different colour of the line in the graph means the different training model. In order to achieve better performance, deploy the lowest loss model, and the lowest total loss achieve 0.245.

5.5 Deploy model and perform testing

After done the training, the model deployed using Jupyter notebook. To deploy the model, there are few steps to follow.

1. Firstly, import libraries and Mask RCNN library and fill in the directory path for the weight trained previously.

```
import os
import sys
import random
import math
import re
import time
import numpy as np
import tensorflow as tf
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.patches as patches

# Root directory of the project
ROOT_DIR = os.path.abspath("../..")

# Import Mask RCNN
sys.path.append(ROOT_DIR) # To find Local version of the library
from mrcnn import utils
from mrcnn import visualize
from mrcnn.visualize import display_images
import mrcnn.model as modellib
from mrcnn.model import log

from samples.breastcancer import breastcancer

%matplotlib inline

# Directory to save Logs and trained model
MODEL_DIR = os.path.join(ROOT_DIR, "logs")

# Path to breastcancer trained weights
# You can download this file from the Releases page
# https://github.com/matterport/Mask_RCNN/releases
BREASTCANCER_WEIGHTS_PATH = "../breastcancer20200827T1740/mask_rcnn_breastcancer_0100.h5" # TODO: update this path
```

Figure 5.5.1 Import library

2. Load Validation Dataset

```
# Load validation dataset
dataset = breastcancer.CustomDataset()
dataset.load_custom(CUSTOM_DIR, "val")

# Must call before using the dataset
dataset.prepare()

print("Images: {}\nClasses: {}".format(len(dataset.image_ids), dataset.class_names))

Images: 20
Classes: ['BG', 'benign', 'malignant']
```

Figure 5.5.2 Load dataset

3. Load the model

```
# Create model in inference mode
with tf.device(DEVICE):
    model = modellib.MaskRCNN(mode="inference", model_dir=MODEL_DIR,
                              config=config)
```

Figure 5.5.3 Load model

4. Run Detection on the validation images. The result will show the three output such as the segmentation mask, bounding box and class of cancer.



Figure 5.5.4 Actual Images and the label

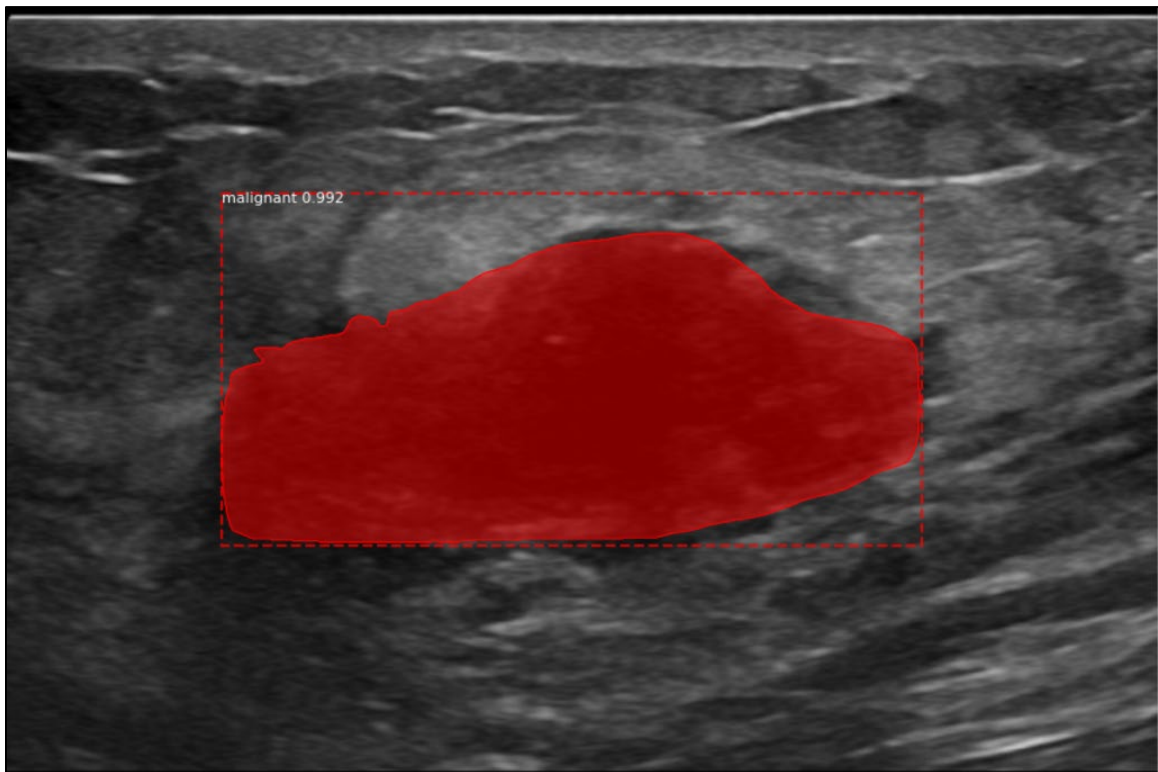


Figure 5.5.5 Predicted Result

Chapter 6: Conclusion

6.1 Future Work

This project is able to classify breast cancer, which is benign or malignant and detecting the location of cancer in the breast ultrasound images by using the Mask R-CNN model. However, this project is a preliminary study and fundamental of using the deep learning approaches to detect breast cancer. There are a few aspects that need to be improved in order to have a complete system. For example, user-friendly, accuracy and the performance of the model. Furthermore, the model deployment in this project is not user friendly because of not every user familiar with using the Jupyter notebook. Besides, it is considered challenging to perform prediction on the greyscale medical images. There is challenging to improve the accuracy in predicting the medical images that are limited features. For instance, the breast cancer there no specific shape, or any colour in the breast ultrasound images. Therefore, the future work of this project is the accuracy, and the performance of the model is expected to increase. Furthermore, the model will be deployed in the application to enhance the user experience, such as deploy into a web-based application, mobile application, etc.

6.2 Conclusion

In conclusion, breast cancer is the first leading cause of cancer-related death among Malaysian women and has a negative effect on woman health. Besides that, the traditional diagnosis systems do not significantly increase the prediction accuracy of the diagnostic performance. By using the traditional method, this will need an intensive workload and easily prone to fall diagnostic errors.

In order to solve those problems, the motivation in this project is to create a system by using deep learning techniques which can classify and detect breast cancer in the breast ultrasound images. By using this system, it is able to improve the diagnostic performance and assist doctors in reducing their workload and the time consumption in the diagnostic process.

To achieve the goal, the technique proposed for classification and segmentation of breast cancer ultrasound images is applying and fundamental study the pre-trained mask R-CNN model. By using this neural network model, its ability to perform object detection and instance segmentation and the results will show the class label, bounding box coordination and segmentation mask for the detected object.

By implementing this project, Malaysia woman might have the benefit and able to prevent cancer at the earlier stage. Furthermore, it might reduce the rate of mortality caused by breast cancer.

References

- Anon., n.d. *Biopsy*. [Online]
Available at: <https://www.nationalbreastcancer.org/breast-cancer-biopsy>
[Accessed 13 February 2020].
- Dai, Z. et al., 2019. *Segmentation of the Prostatic Gland and the Intraprostatic Lesions on Multiparametric MRI Using Mask-RCNN*. [Online]
Available at: <https://arxiv.org/abs/1904.02575>
[Accessed 27 February 2020].
- Dertat, A., 2017. *Applied Deep Learning - Part 4: Convolutional Neural Networks*. [Online]
Available at: <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>
[Accessed 12 February 2020].
- Early breast cancer detection is in your hands, 2018*. [Online]
Available at:
<https://www.freemalaysiatoday.com/category/leisure/2018/03/27/early-breast-cancer-detection-is-in-your-hands/>
[Accessed 16 February 2020].
- Javaid, N., 2018. *Machine Learning —Fundamentals*. [Online]
Available at: <https://towardsdatascience.com/machine-learning-basics-part-1-a36d38c7916>
[Accessed 10 February 2020].
- Jiang, F. et al., 2017. *Artificial intelligence in healthcare: past,present and future*. [Online]
Available at: <https://svn.bmj.com/content/svnbmj/2/4/230.full.pdf>
[Accessed 11 February 2020].
- Motlagh, M. et al., 2018. *Breast Cancer Histopathological Image Classification: A Deep Learning Approach*. [Online]

Available at: <https://www.biorxiv.org/content/10.1101/242818v1>

[Accessed 23 February 2020].

Nawaz, M., Adel, A. & Taysir, H., 2018. *Multi-Class Breast Cancer Classification using Deep Learning Convolutional Neural Network*. [Online]

Available at: <http://dx.doi.org/10.14569/IJACSA.2018.090645>

[Accessed 20 February 2020].

Wu, N. et al., 2019. *Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening*. [Online]

Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8861376>

[Accessed 20 February 2020].

Appendixes

APPENDIX A: Project Poster

Pilot Study of Breast Cancer Classification Using Deep Learning Approach



Chong Kim Yew

Faculty of Information and Communication Technology (FICT),
Universiti Tunku Abdul Rahman, Kampar

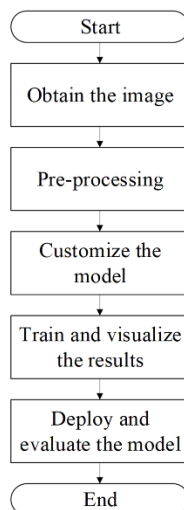
Problem statement

- Breast cancer is the most common and top-ranking cancer is the breast cancer which has a negative effect on women health and also the first leading causes of cancer-related deaths among Malaysian women.
- Traditional diagnose method do not significantly increase the predictive accuracy of the diagnostic performance.
- Pathologists will prone to fall diagnostic errors due to not enough diagnostic experience

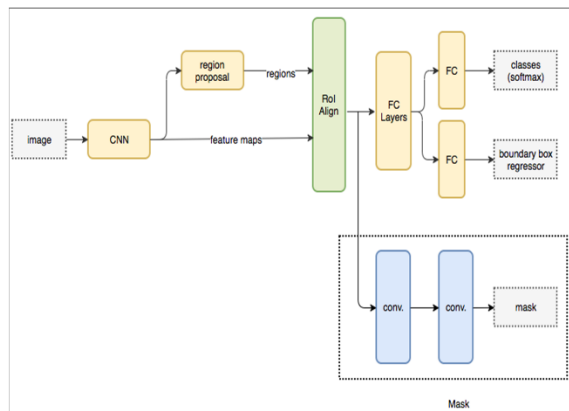
Project Objective

- Develop a preliminary disease diagnosis system for breast cancer detection using the existing techniques of deep learning to improve the performance of cancer detection
- To reduce the workload and the time consumption of detecting a large amount of medical image dataset

System Design

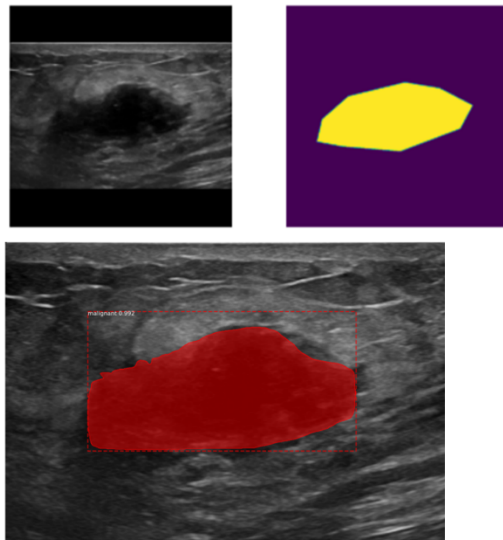


Methodology



Mask RCNN network Architecture

Results



After deploy the model, the result show segmentation mask, bounding box and class label of the detected cancer.

Appendix B: Plagiarism Check Summary

Feedback Studio - Google Chrome

ev.turnitin.com/app/carta/en_us/?u=1064427501&lang=en_us&o=1299057450&s=&student_user=1

feedback studio

Chong Kim Yew | FYP2 report

-- /0 ?

15

15

15

15

15

15

15

15

15

15

15

15

15

15

15

15

Pilot Study of Breast Cancer Classification Using Deep Learning Approach

BY

CHONG KIM YEW

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology (FICT)

(Kampar Campus)

May 2020

Match Overview

15%

1

hdl.handle.net

Internet Source

1%

>

2

Submitted to Liverpool ...

Student Paper

1%

>

3

cnqzu.com

Internet Source

1%

>

4

Nan Wu, Jason Phang, ...

Publication

1%

>

5

Zhenzhen Dai, Eric Car...

Publication

1%

>

6

*Advanced Computing ...

Publication

1%

>

7

*Proceedings of the Int...

Publication

<1%

>

8

Walid Al-Dhabyani, Mo...

Publication

<1%

>

9

gitlab-vs.informatik.uni...

Internet Source

<1%

>

10

*Artificial Neural Netwo...

Publication

<1%

>

11

Submitted to Royal Holl...

Student Paper

<1%

>

12

Jiazheng Shi, Berkman ...

Publication

<1%

>

13

Submitted to University...

Student Paper

<1%

>

14

fraser.stlouisfed.org

Internet Source

<1%

>

15

*Machine Learning in ...

Publication

<1%

>

16

Submitted to University...

Student Paper

<1%

>

Page: 1 of 60

Word Count: 8274

Text-only Report

High Resolution

On

Q

Q

FYP2 report

ORIGINALITY REPORT

15%

SIMILARITY INDEX

6%

INTERNET SOURCES

10%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1

hdl.handle.net

Internet Source

1%

2

Submitted to Liverpool John Moores University

Student Paper

1%

3

cnqzu.com

Internet Source

1%

4

Nan Wu, Jason Phang, Jungkyu Park, Yiqiu Shen et al. "Deep Neural Networks Improve Radiologists' Performance in Breast Cancer Screening", IEEE Transactions on Medical Imaging, 2020

Publication

1%

5

Zhenzhen Dai, Eric Carver, Chang Liu, Joon Lee, Aharon Feldman, Weiwei Zong, Milan Pantelic, Mohamed Elshaikh, Ning Wen. "Segmentation of the Prostatic Gland and the Intraprostatic Lesions on Multiparametric MRI Using Mask R-CNN", Advances in Radiation Oncology, 2020

Publication

1%

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1 of 1



**FACULTY OF INFORMATION AND COMMUNICATION
TECHNOLOGY**

Full Name(s) of Candidate(s)	Chong Kim Yew
ID Number(s)	17ACB00134
Programme / Course	BACHELOR OF COMPUTER SCIENCE (HONS)
Title of Final Year Project	Pilot Study of Breast Cancer Classification Using Deep Learning Approach

Similarity	Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)
Overall similarity index: <u>15</u> % Similarity by source Internet Sources: <u>6</u> % Publications: <u>10</u> % Student Papers: <u>5</u> %	
Number of individual sources listed of more than 3% similarity: <u>-</u>	
Parameters of originality required and limits approved by UTAR are as Follows: (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Sayed Ahmad Zikri
Signature of Supervisor

Signature of Co-Supervisor

Name: Sayed Ahmad Zikri Bin Sayed Aluwee

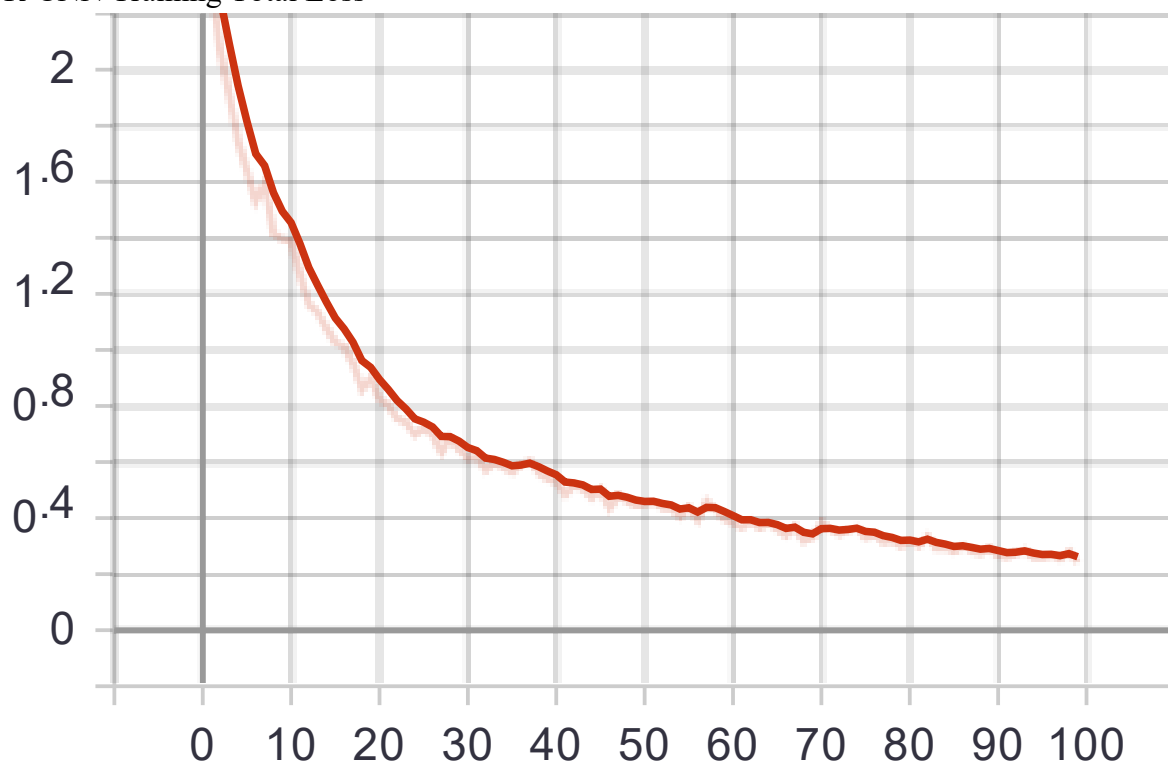
Name: _____

Date: 10 September 2020

Date: _____

Appendix C: Chart

Mask R-CNN Training Total Loss






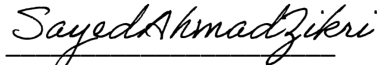
UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION & COMMUNICATION
TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	17ACB00134
Student Name	Chong Kim Yew
Supervisor Name	Dr Sayed Ahmad Zikri Bin Sayed Aluwee

TICK (✓)	DOCUMENT ITEMS
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
✓	Front Cover
✓	Signed Report Status Declaration Form
✓	Title Page
✓	Signed form of the Declaration of Originality
✓	Acknowledgement
✓	Abstract
✓	Table of Contents
✓	List of Figures (if applicable)
✓	List of Tables (if applicable)
	List of Symbols (if applicable)
✓	List of Abbreviations (if applicable)
✓	Chapters / Content
✓	Bibliography (or References)
✓	All references in bibliography are cited in the thesis, especially in the chapter of literature review
✓	Appendices (if applicable)
✓	Poster
✓	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

<p>I, the author, have checked and confirmed all the items listed in the table are included in my report.</p> <p></p> <p>(Signature of Student) Date: 10/09/2020</p>	<p>Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.</p> <p></p> <p>(Signature of Supervisor) Date: 10 September 2020</p>
---	---