

Emerging Trends Project

Histopathologic Cancer Detection

Submitted by

TEAM 3

Chahat - 03904092018

Sreyashi Bhattacharjee - 05104092018

Aarti Uniyal - 05204092018

Deepika - 05504092018

Aakriti Mittal- 05804092018

Under the Supervision of

Dr. Rishabh Kaushal

Assistant Professor

Department of Information Technology



Department of Information Technology

Indira Gandhi Delhi Technical University for Women

Kashmere Gate, Delhi - 110006

December 2020

Contents

1	Introduction	1
2	Literature Survey	2
3	Problem Statement	3
3.1	Research questions	3
4	Dataset	4
4.1	Data Description:	4
4.1.1	Number of data instances and type(s) of instances	4
5	Data Exploration and Visualization	5
5.1	Data Exploration	5
5.2	Visualisation	8
6	Proposed Approach	10
6.1	Cancer Detection	11
6.1.1	ResNet50	12
6.1.2	Vgg16	34
6.1.3	Custom Convolution Neural Network (CNN)	56
7	Evaluation and Comparisons	78
7.0.1	Time v/s Performance	79
7.0.2	Performance v/s Data Size	82
7.0.3	Time v/s Data Size	85
7.0.4	Performance v/s Data Composition — Class Imbalance Effect . . .	88
7.0.5	Model Comparison	91
8	Result	94

Chapter 1

Introduction

Metastatic Cancer Detection

Cancer that spreads from where it started to a distant part of the body is called metastatic cancer. For many types of cancer, it is also called stage IV (4) cancer. The process by which cancer cells spread to other parts of the body is called metastasis.

When observed under a microscope and tested in other ways, metastatic cancer cells have features like that of the primary cancer and not like the cells in the place where the metastatic cancer is found.

Metastatic cancer has the same name as the primary cancer. For example, breast cancer that spreads to the lung is called metastatic breast cancer, not lung cancer. It is treated as stage IV breast cancer, not as lung cancer.

Advancing the fight against cancer requires early detection which can only be possible with an efficient detection system. Techniques have been developed to detect cancer, including medical image processing and digital pathology.

Images are acquired by histopathology, which generally includes biopsy of the affected tissue. Tissues affected by the tumor are extracted by the pathologist and stained by H E, which is the combination of histological stains called hematoxylin and eosin, after which it is examined under a microscope for cancerous cells by finding malignant features in cellular structures such as nuclei. These microscopic images can be collected and used for developing computer-aided detection systems.

This paper looks at classifying the metastatic histopathologic cancer cell images in order to detect the presence of cancer cells.

The paper attempts to study the cell images extracted from the lymph node sections.

Chapter 2

Literature Survey

>> PatchCamelyon (PCam) Dataset : <https://github.com/basveeling/pcam>

The PatchCamelyon benchmark is a new and challenging image classification dataset. It consists of 327.680 color images (96 x 96px) extracted from histopathologic scans of lymph node sections. Each image is annotated with a binary label indicating presence of metastatic tissue. PCam provides a new benchmark for machine learning models: bigger than CIFAR10, smaller than imagenet, trainable on a single GPU.

>> Identify metastatic tissue in histopathologic scans of lymph node sections:

<https://www.kaggle.com/c/histopathologic-cancer-detection>

The data for this project is a slightly modified version of the PatchCamelyon (PCam) benchmark dataset (the original PCam data set contains duplicate images due to its probabilistic sampling, however, the version presented on Kaggle does not contain duplicates).

>> Cancer diagnosis in histopathological image: CNN based approach:

<https://www.sciencedirect.com/science/article/pii/S2352914819301133>

In this paper, the authors have trained a convolutional neural network and obtained a prediction accuracy of up to 99.86%

Chapter 3

Problem Statement

Manual detection of a cancer cell is a tiresome task and involves human error, and hence computer-aided mechanisms are applied to obtain better results as compared with manual pathological detection systems. In deep learning, this is generally done by extracting features through a convolutional neural network (CNN) and then classifying using a fully connected network.

This paper aims to apply various modified versions of pre-trained models as well and attempts to answer various research questions along with cancer detection.

3.1 Research questions

We intend to find solution to the following research questions.

1. What is the class imbalance effect on the predictions made?
2. How does the data size affect the performance of the models ?
3. How is the performance effected with respect to the training time?
4. Does the training time have any relation with the performance of the model?
5. Which model gives a better performance?

Chapter 4

Dataset

4.1 Data Description:

Source : <https://www.kaggle.com/c/histopathologic-cancer-detection/data>

Train Folder : training data images

Test Folder : test data images

Train_labels.csv : CSV file with the labels of training images

4.1.1 Number of data instances and type(s) of instances

Number of training images : 220025

Number of testing images : 57458

Number of distinct labels : 2

Number of images with label "0" (no cancer) : 130908

Number of images with label "1" (with cancer) : 89117

Each image size is : 96x96

Center at : 32x32

Chapter 5

Data Exploration and Visualization

5.1 Data Exploration

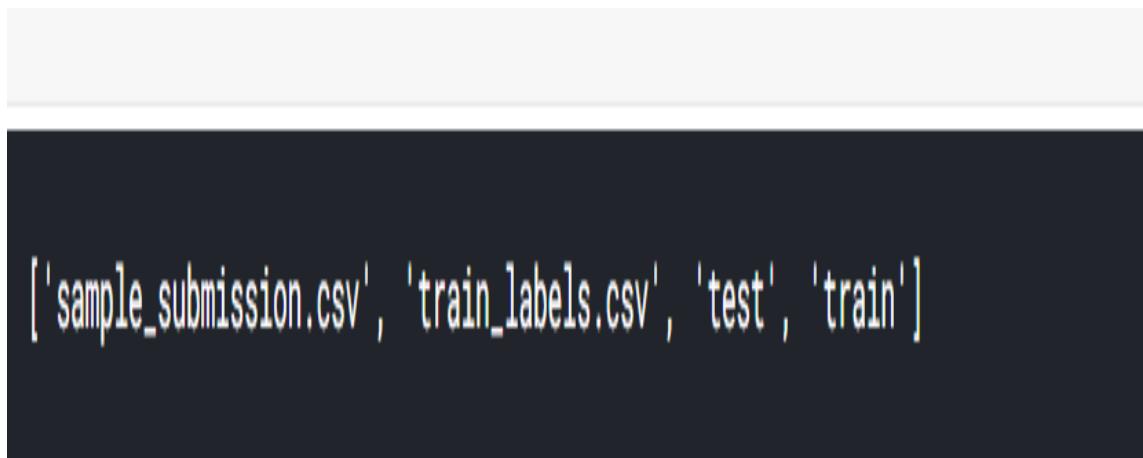


Figure 5.1: Data Sets used



Figure 5.2: Values of training and testing data



Figure 5.3: Number of labels and its type

```
0    130908  
1    89117  
Name: label, dtype: int64
```

Figure 5.4: Number of images containing cancer cells and do not have cancer cells

5.2 Visualisation

The images below has been displayed and classified according to their respective labels.

In the next set of images the center of 32x32 has been highlighted.

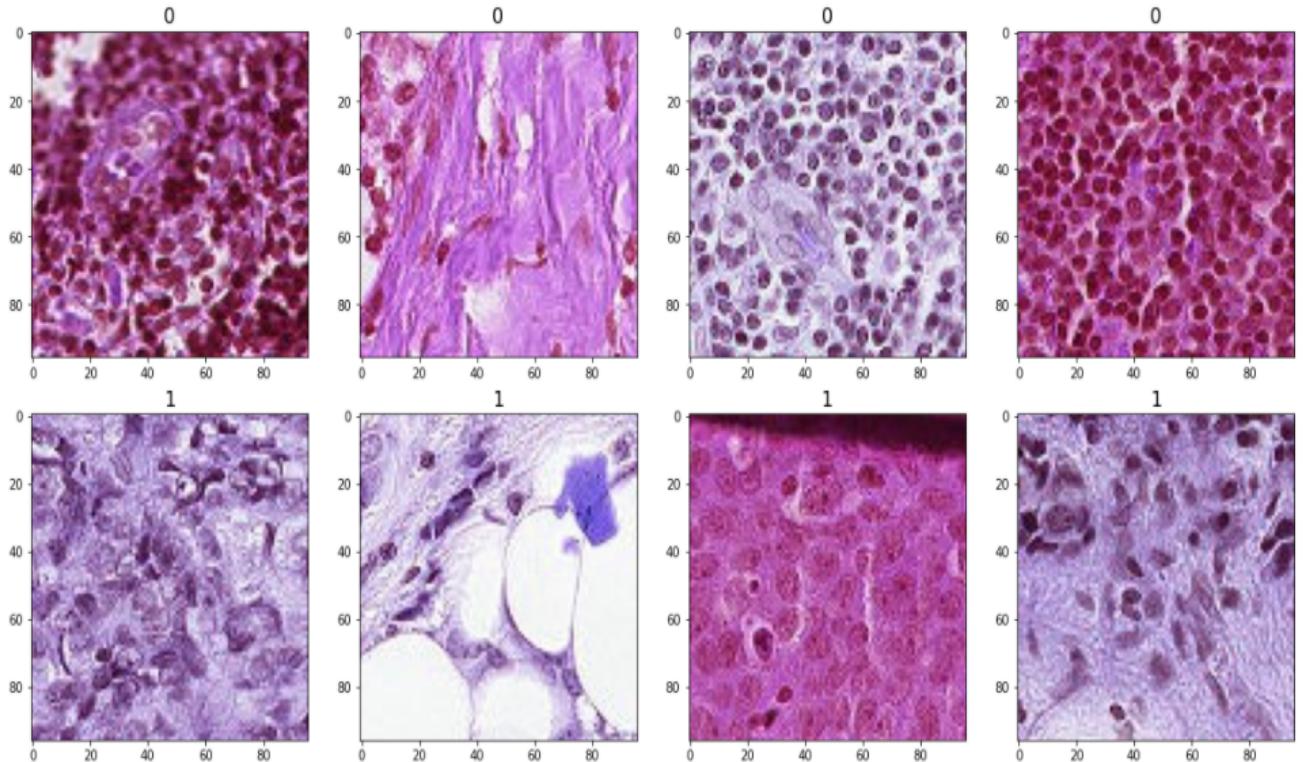


Figure 5.5: Cancer detection

Histopathologic scans of lymph node sections

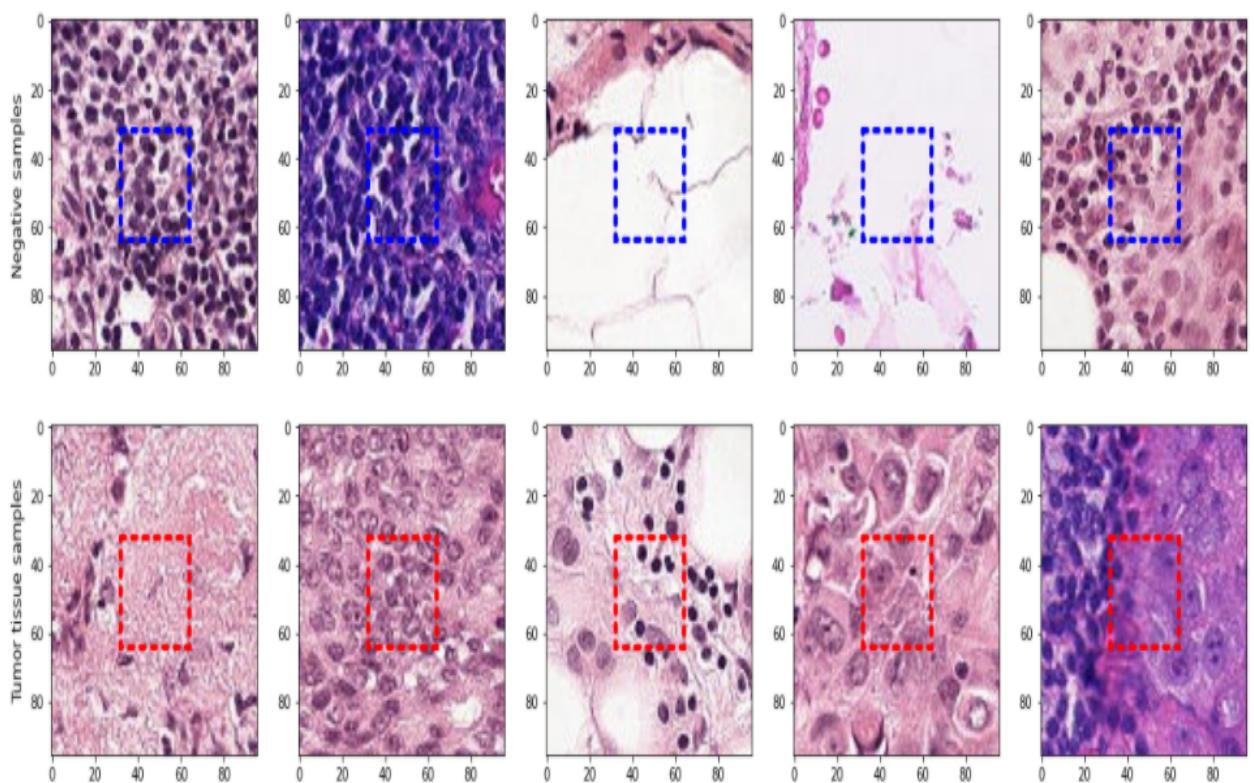


Figure 5.6: Cancer detection with centered marked at $32 * 32$

Chapter 6

Proposed Approach

We have used different deep neural network models. We have analysed the data set and to achieve our goal we have employed the use of three models - custom CNN, Vgg16 and ResNet50.

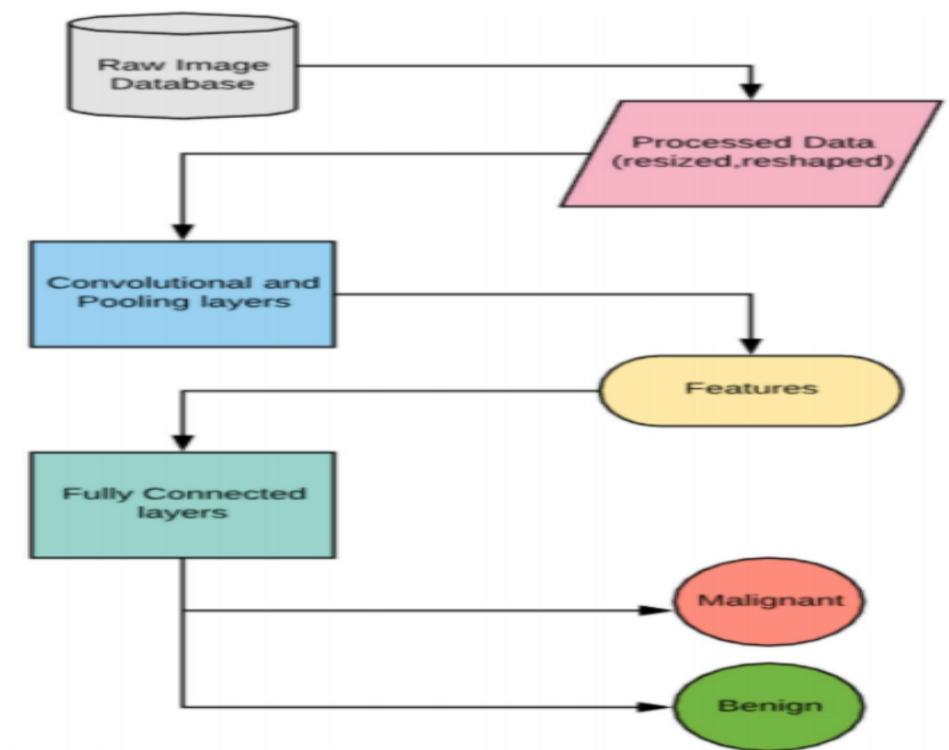


Figure 6.1: Data flow Diagram of proposed methods

6.1 Cancer Detection

This Section works on different combinations and subsets of dataset to seek answers to the research questions.

We first look at the effect of dataset size by working on three subsets of the data, namely:

- 1000 samples of each category (Balanced)
- 500 samples of each category (Balanced)
- 100 samples of each category (Balanced)

We then looked at the effect of data composition in the following subsets:

- 1000 samples of cells without cancer and 500 with cancer cells
- 500 samples of cells with cancer and 1000 without cancer

6.1.1 ResNet50

Description

Deep Residual Network was arguably the most groundbreaking work in the computer vision/deep learning community in the last few years. ResNet makes it possible to train up to hundreds or even thousands of layers and still achieves compelling performance.

ResNet50 is a variant of ResNet model which has 48 Convolution layers along with 1 MaxPool and 1 Average Pool layer. It has 3.8×10^9 floating points operations. It is a widely used ResNet model.

The next page contains the model used in this paper which was customized based on the image data requirement.

ResNet50 Architecture

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112			7×7, 64, stride 2		
				3×3 max pool, stride 2		
conv2_x	56×56	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1			average pool, 1000-d fc, softmax		
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 6.2: ResNet50 Architecture

```
Model: "sequential"
-----
Layer (type)          Output Shape         Param #
-----
resnet50 (Functional) (None, 7, 7, 2048) 23587712
-----
flatten (Flatten)     (None, 100352)       0
-----
dense (Dense)         (None, 256)          25690112
-----
batch_normalization (BatchNo (None, 256) 1024
-----
activation (Activation) (None, 256)        0
-----
dropout (Dropout)     (None, 256)          0
-----
dense_1 (Dense)       (None, 1)            257
-----
Total params: 49,279,105
Trainable params: 49,225,473
Non-trainable params: 53,632
```

Figure 6.3: Custom model to train last few layers

1000 of each class - Balanced Dataset

The ResNet50 model is pre-trained on the ImageNet dataset. Our images are much smaller than the images on ImageNet. Therefore we have trained the last few layers of the model instead of just the last one.

Also, we have resized the images before training to get much higher performance. The training has been done on 30 epochs.

	id	label
0	f51542333ecdc504a775b695c3b15f8175b85640	1.0
1	5b8f5a93ce4f11522c6b6538a542f7c076e752f1	1.0
2	127ef5796b5c44aa30a92c13972ef3372fce47f6	0.0

Figure 6.4: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 1st and 2nd images have cancer cells while the 3rd image has no cancer cells.

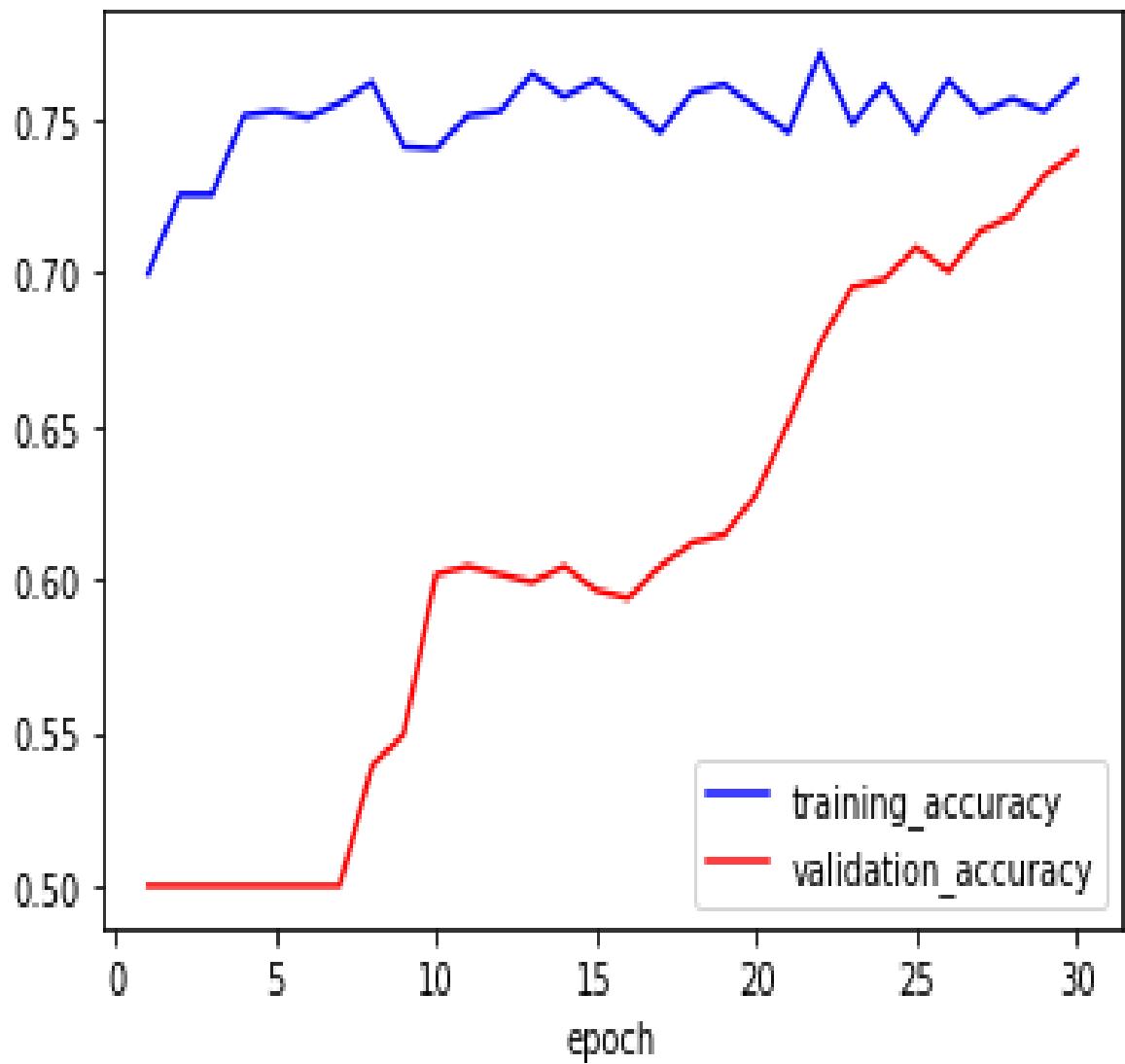


Figure 6.5: Train and Validation Accuracy

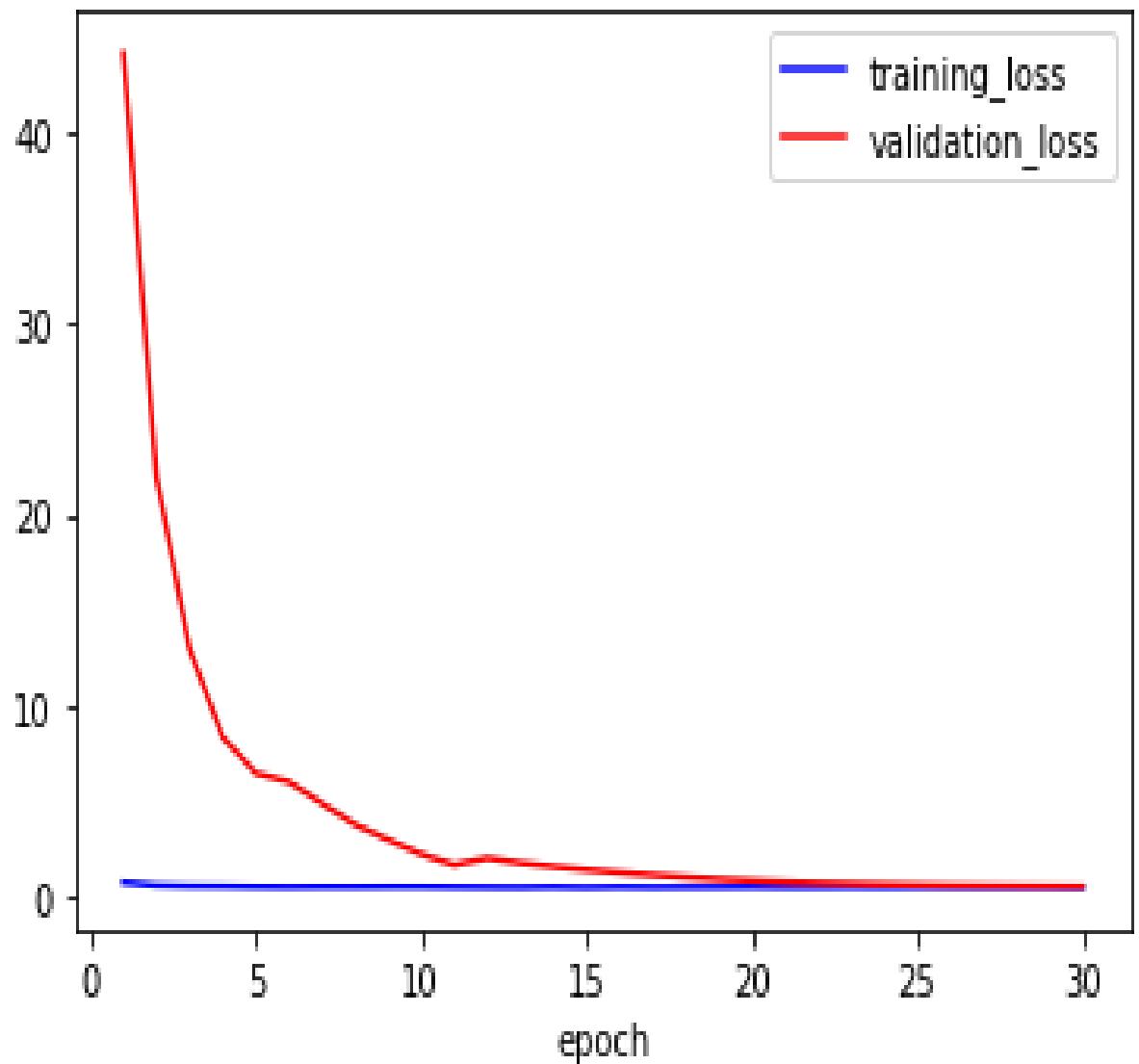


Figure 6.6: Train and Validation Loss

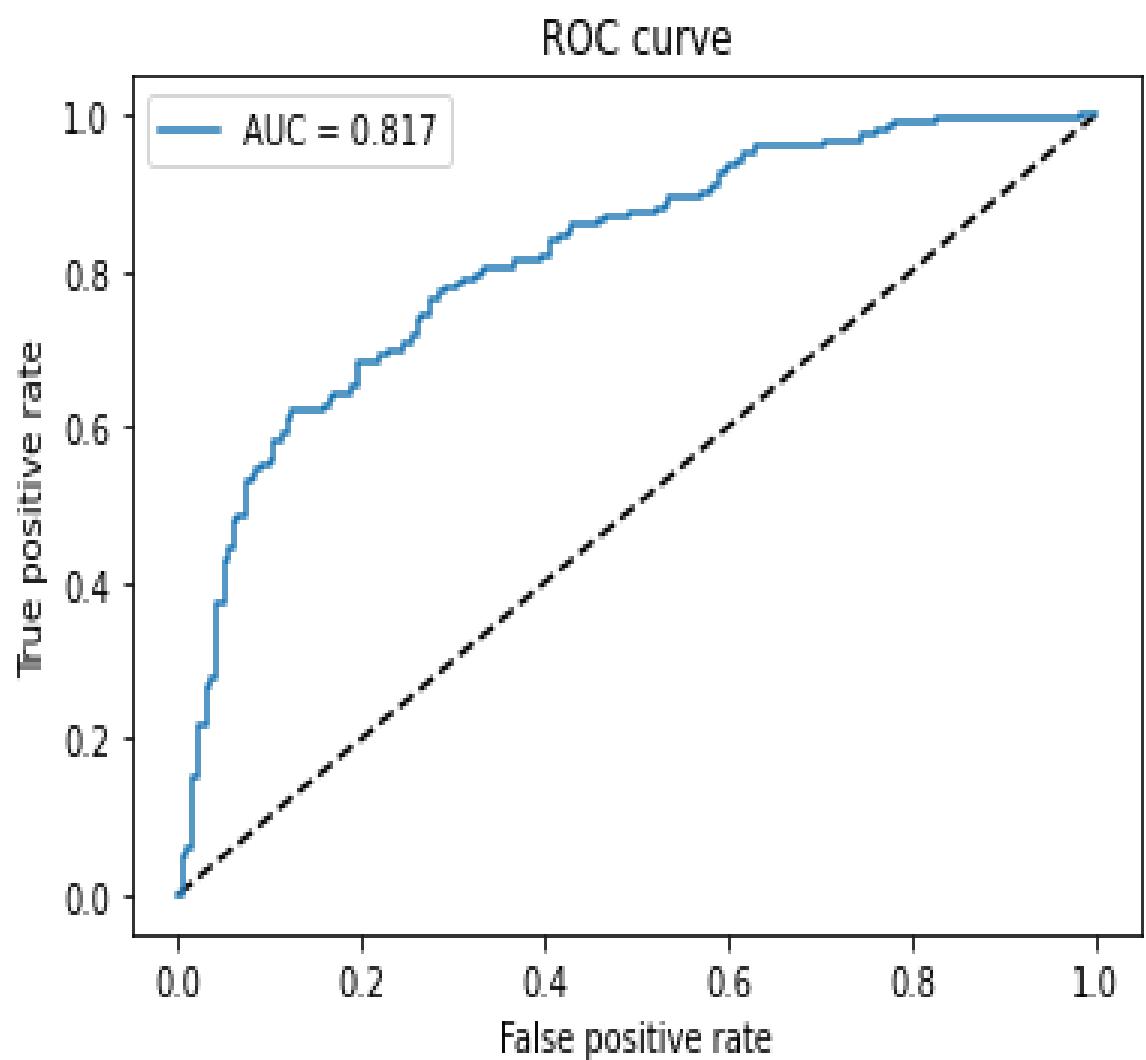


Figure 6.7: ROC-AUC
The AUC score of 0.817 is considered to be excellent.

500 of each class - Balanced Dataset

	id	label
0	f51542333ecdc504a775b695c3b15f8175b85640	1.0
1	5b8f5a93ce4f11522c6b6538a542f7c076e752f1	1.0
2	127ef5796b5c44aa30a92c13972ef3372fce47f6	1.0

Figure 6.8: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 1st, 2nd and 3rd images have cancer cells.

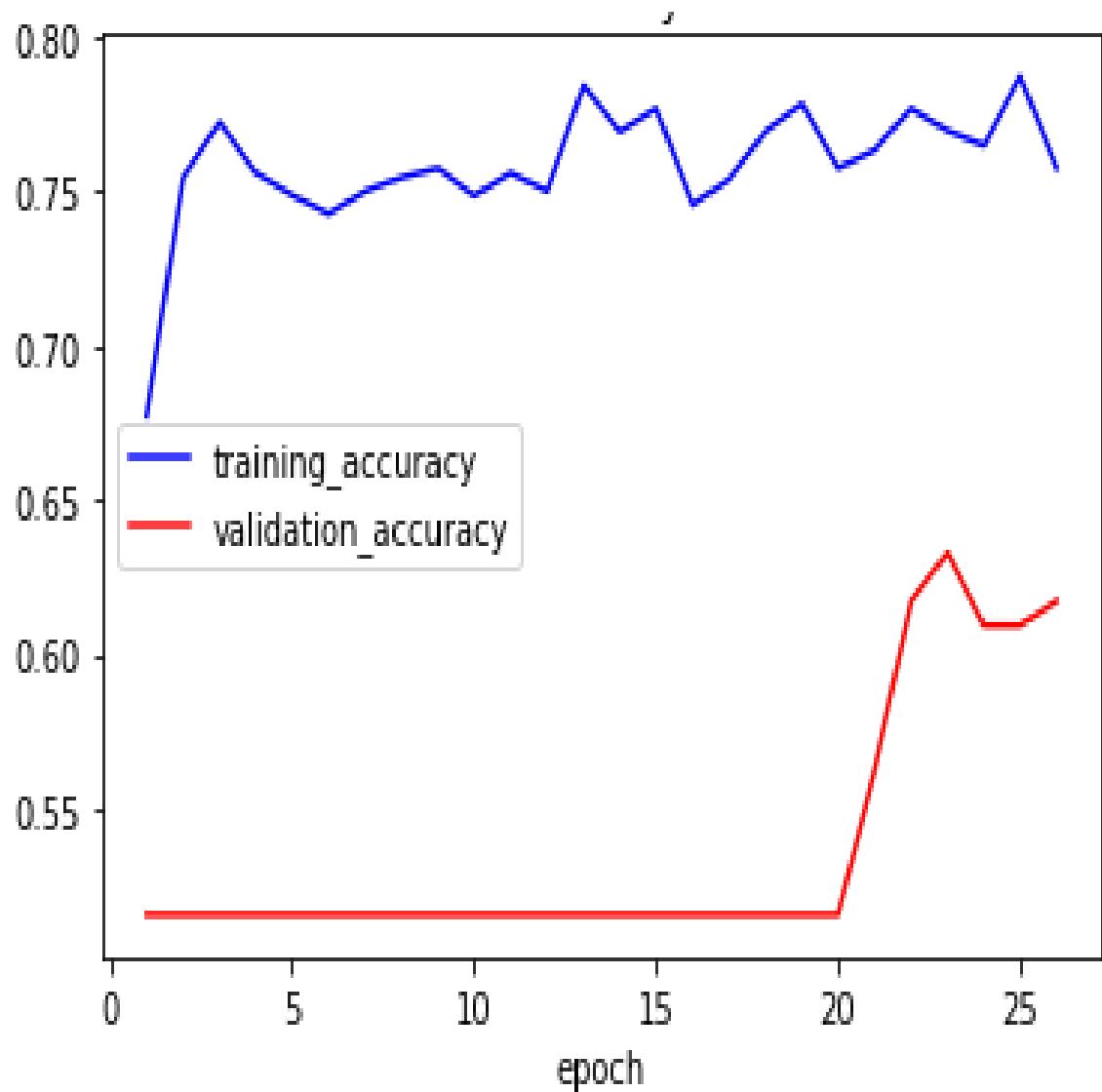


Figure 6.9: Train and Validation Accuracy

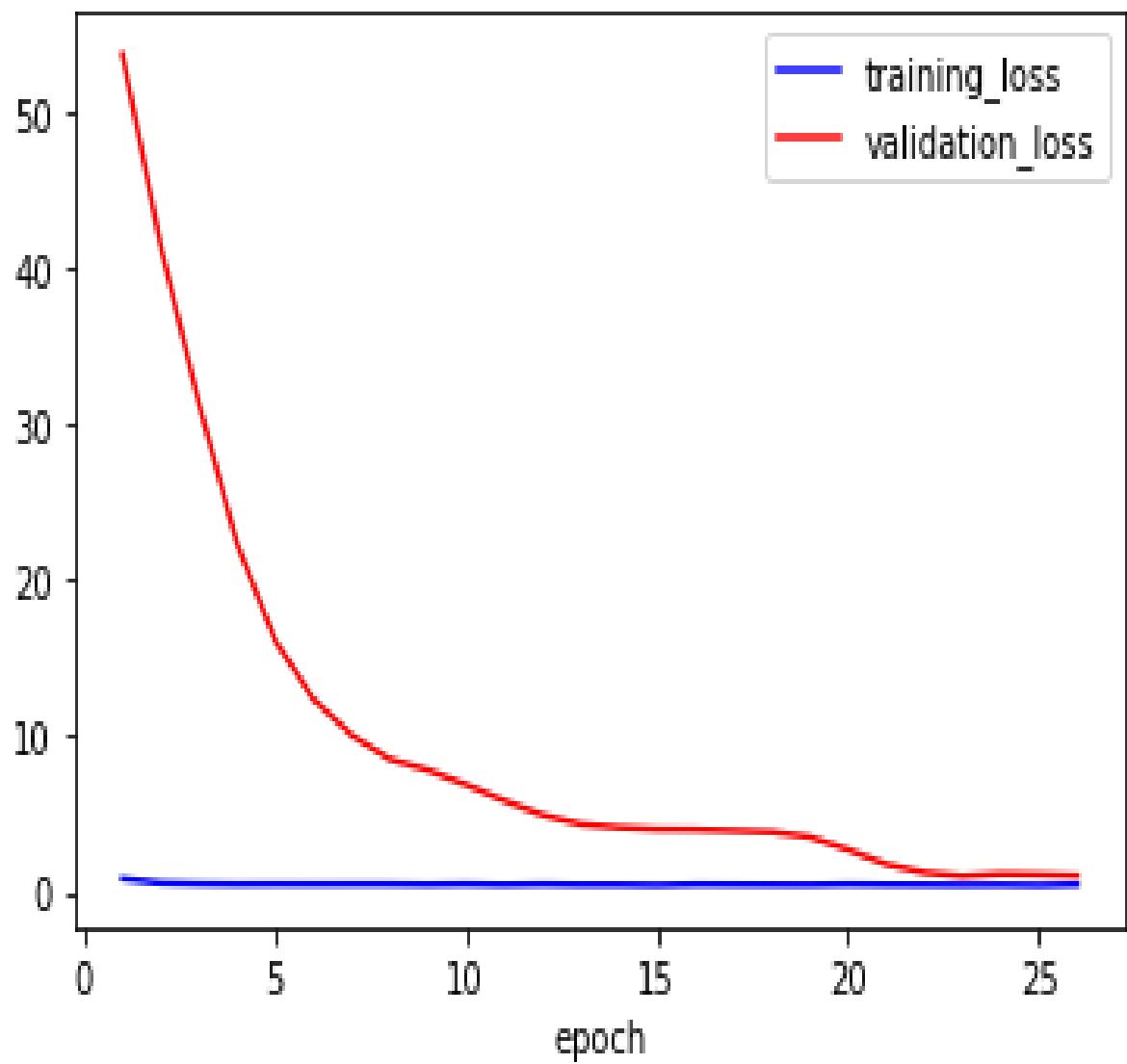


Figure 6.10: Train and Validation Loss

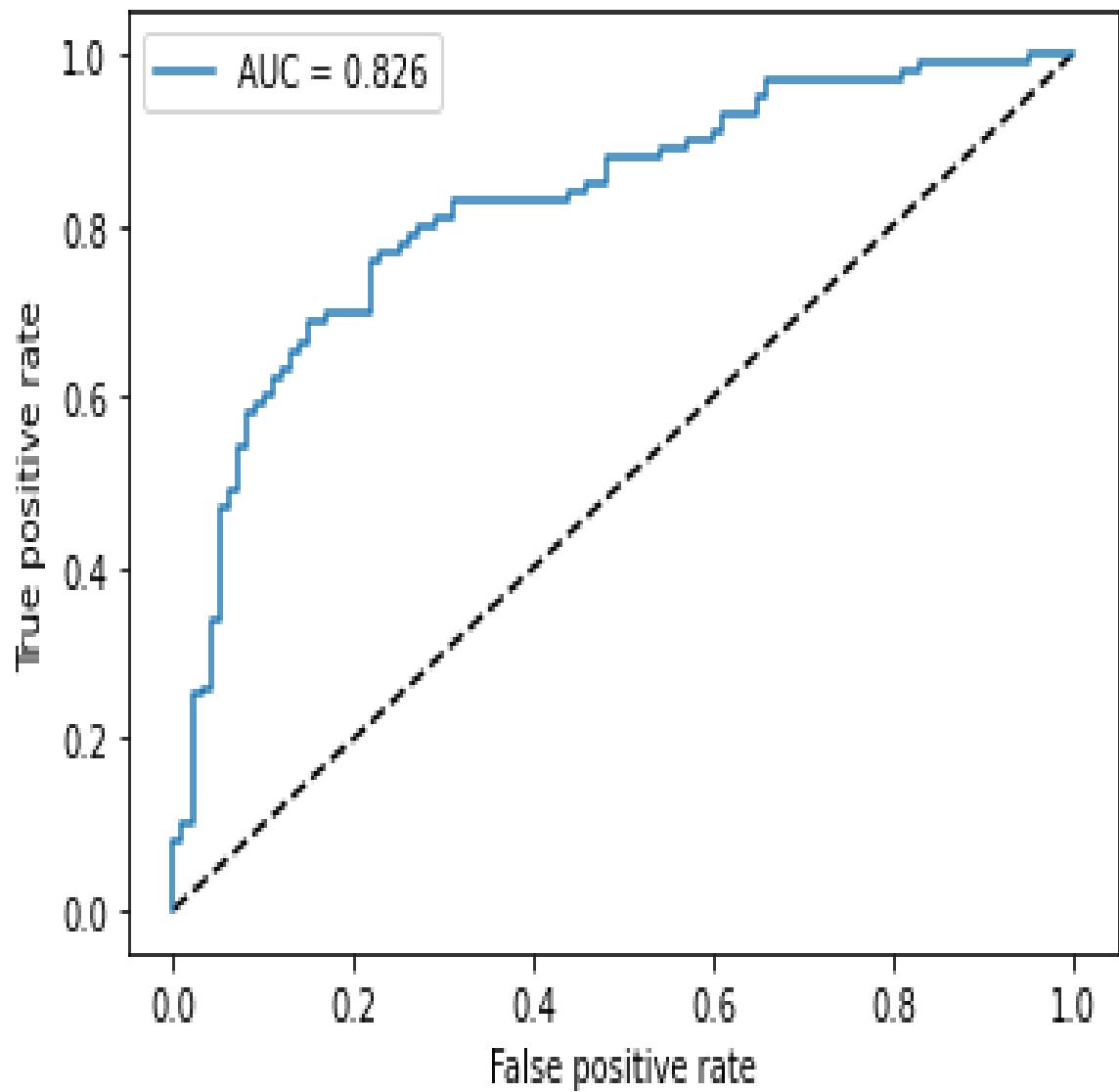


Figure 6.11: ROC-AUC
The AUC score of 0.826 is considered to be excellent.

100 of each class - Balanced Dataset

	id	label
0	f51542333ecdc504a775b695c3b15f8175b85640	1.0
1	5b8f5a93ce4f11522c6b6538a542f7c076e752f1	0.0
2	127ef5796b5c44aa30a92c13972ef3372fce47f6	0.0

Figure 6.12: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 1stimage has cancer cells.2nd and 3rd images do not have cancer cells.

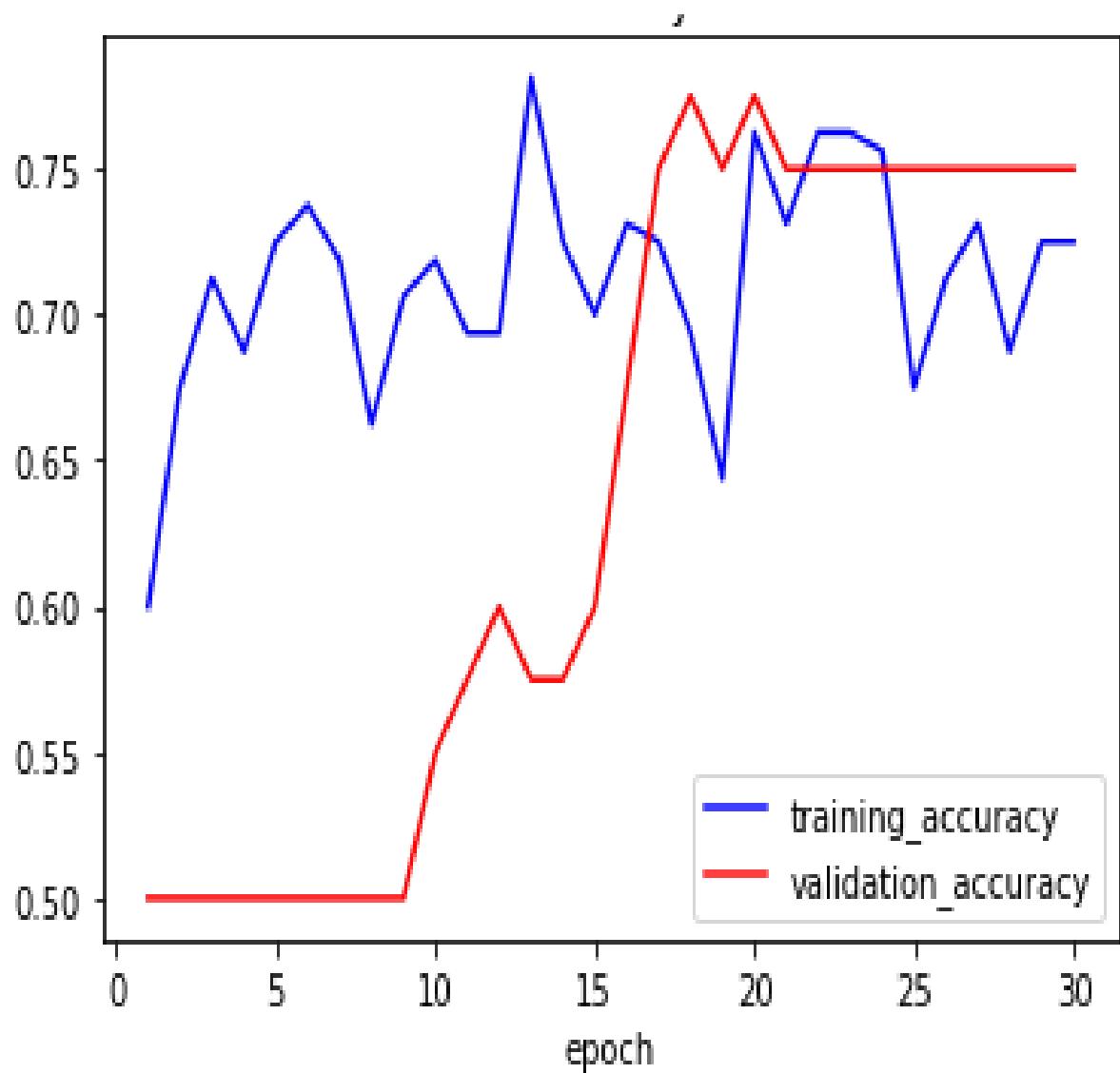


Figure 6.13: Train and Validation Accuracy

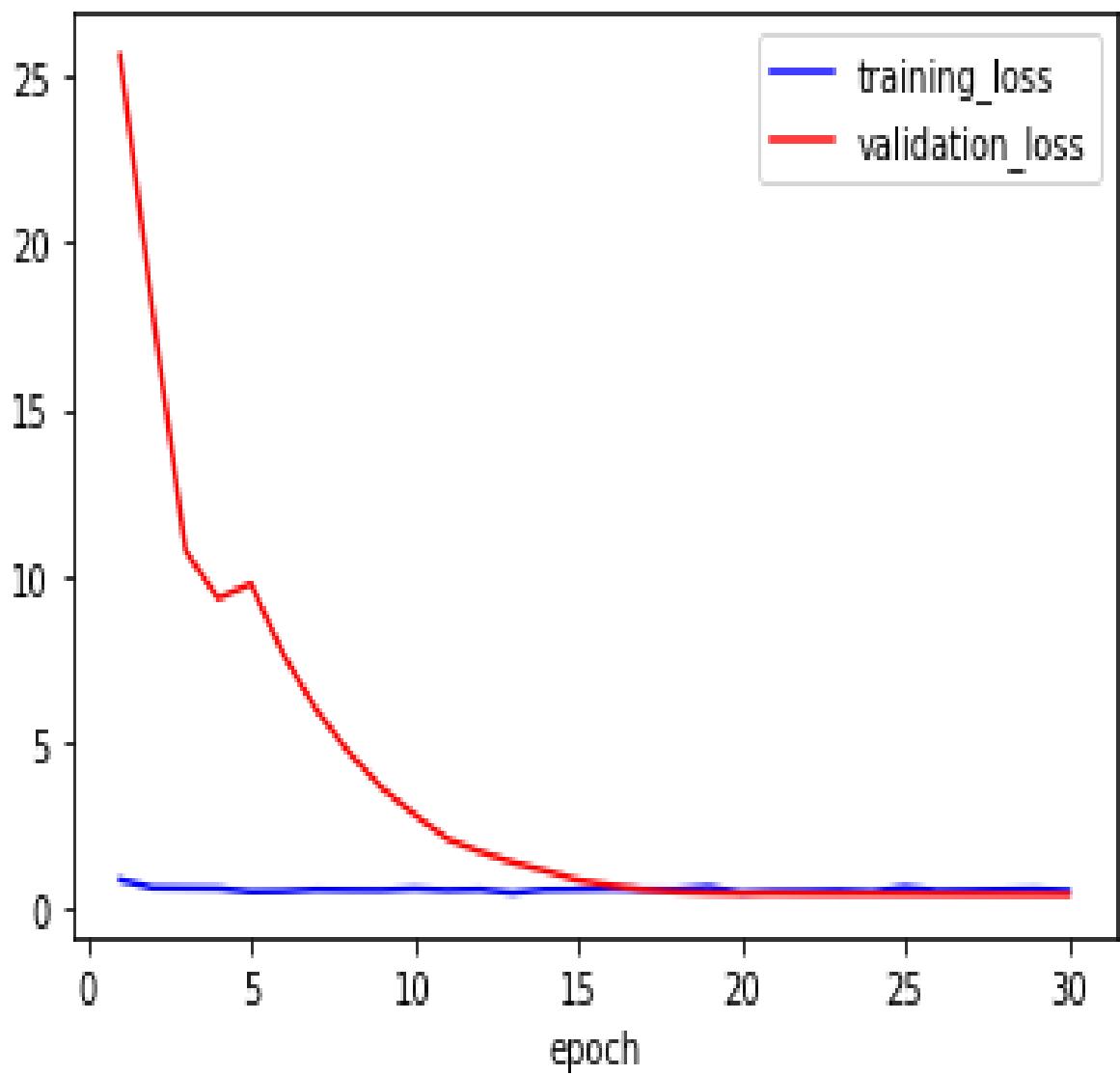


Figure 6.14: Train and Validation Loss

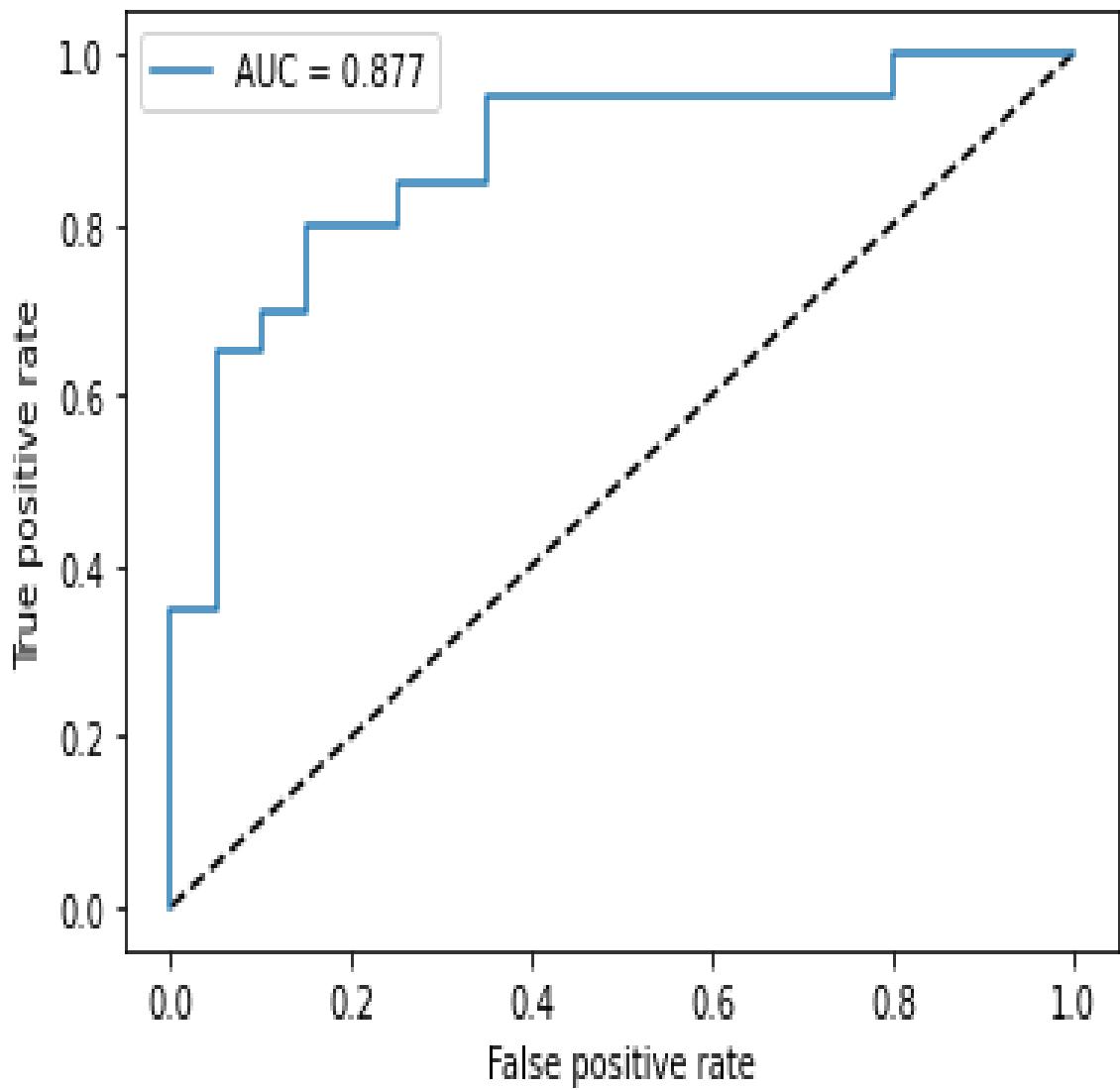


Figure 6.15: ROC-AUC
The AUC score of 0.877 is considered to be excellent.

1000 (no cancer cells) 500 (cancer cells) - Unbalanced Dataset

	id	label
0	f51542333ecdc504a775b695c3b15f8175b85640	1.0
1	5b8f5a93ce4f11522c6b6538a542f7c076e752f1	1.0
2	127ef5796b5c44aa30a92c13972ef3372fce47f6	0.0

Figure 6.16: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 1st and 2nd images have cancer cells while the 3rd image has no cancer cells.

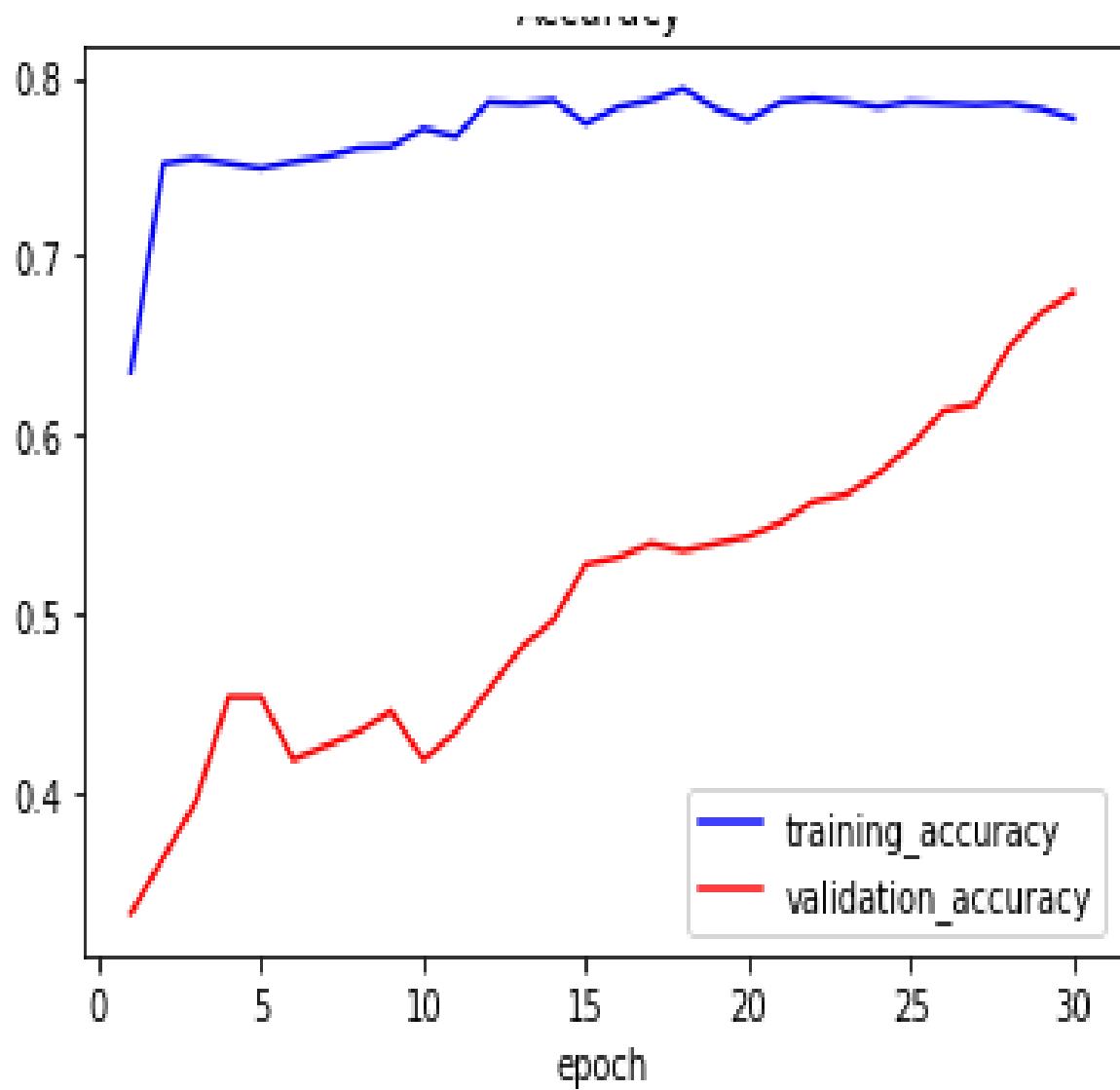


Figure 6.17: Train and Validation Accuracy

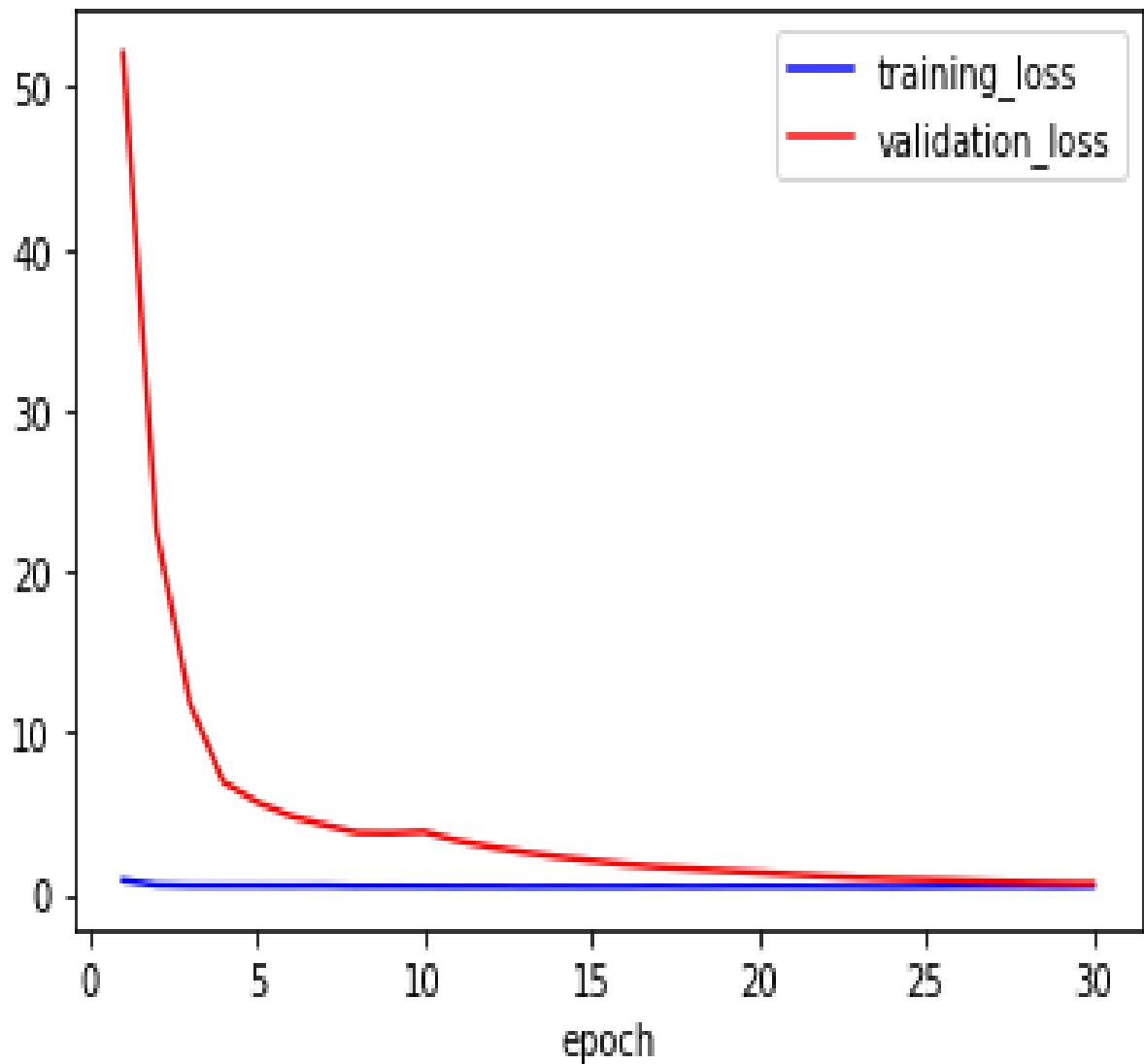


Figure 6.18: Train and Validation Loss

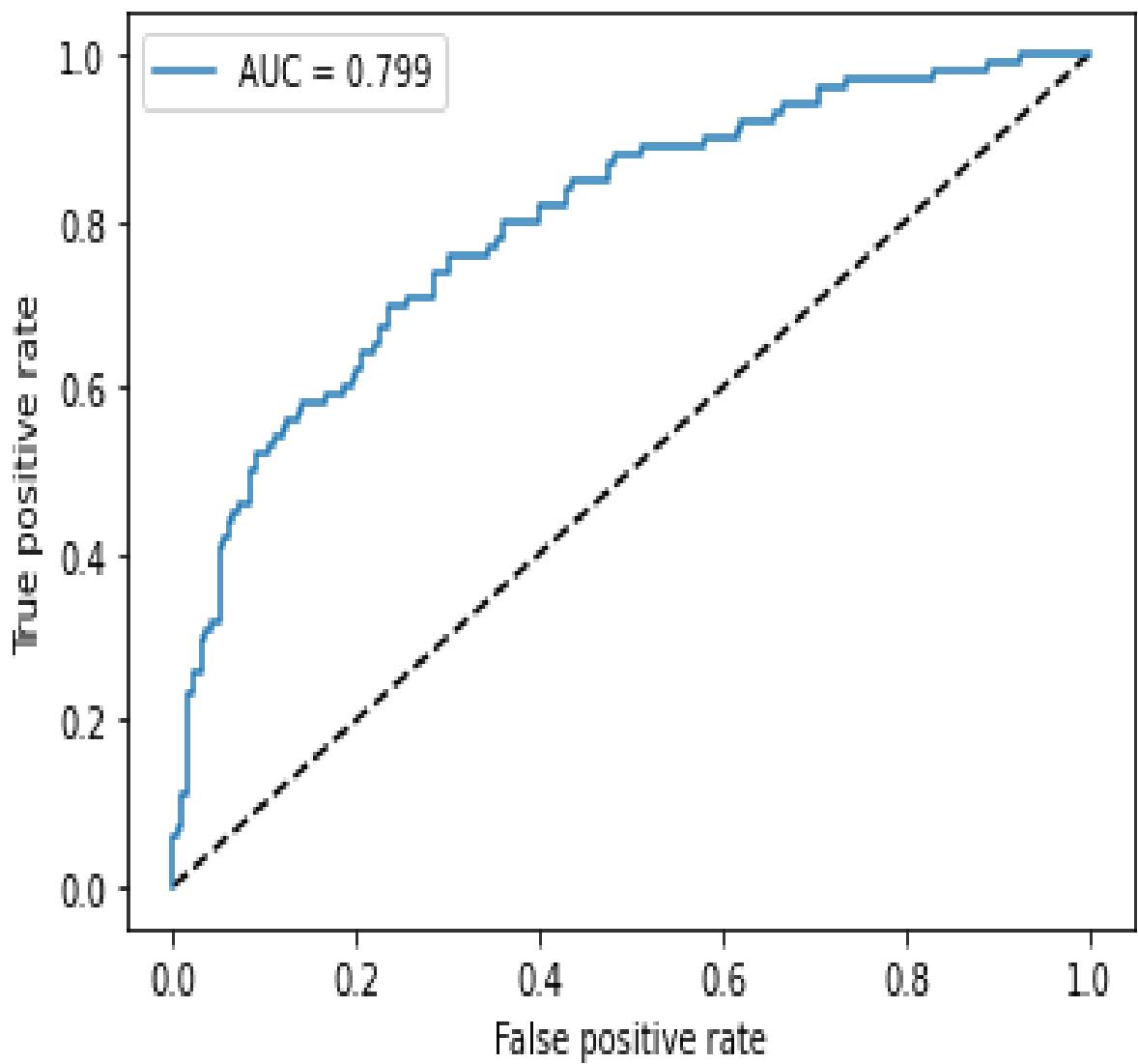


Figure 6.19: ROC-AUC
The AUC score of 0.799 is considered to be acceptable.

500 (no cancer cells) 1000 (cancer cells) - Unbalanced Dataset

	id	label
0	f51542333ecdc504a775b695c3b15f8175b85640	1.0
1	5b8f5a93ce4f11522c6b6538a542f7c076e752f1	1.0
2	127ef5796b5c44aa30a92c13972ef3372fce47f8	1.0

Figure 6.20: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 1st, 2nd and 3rd images have cancer cells.

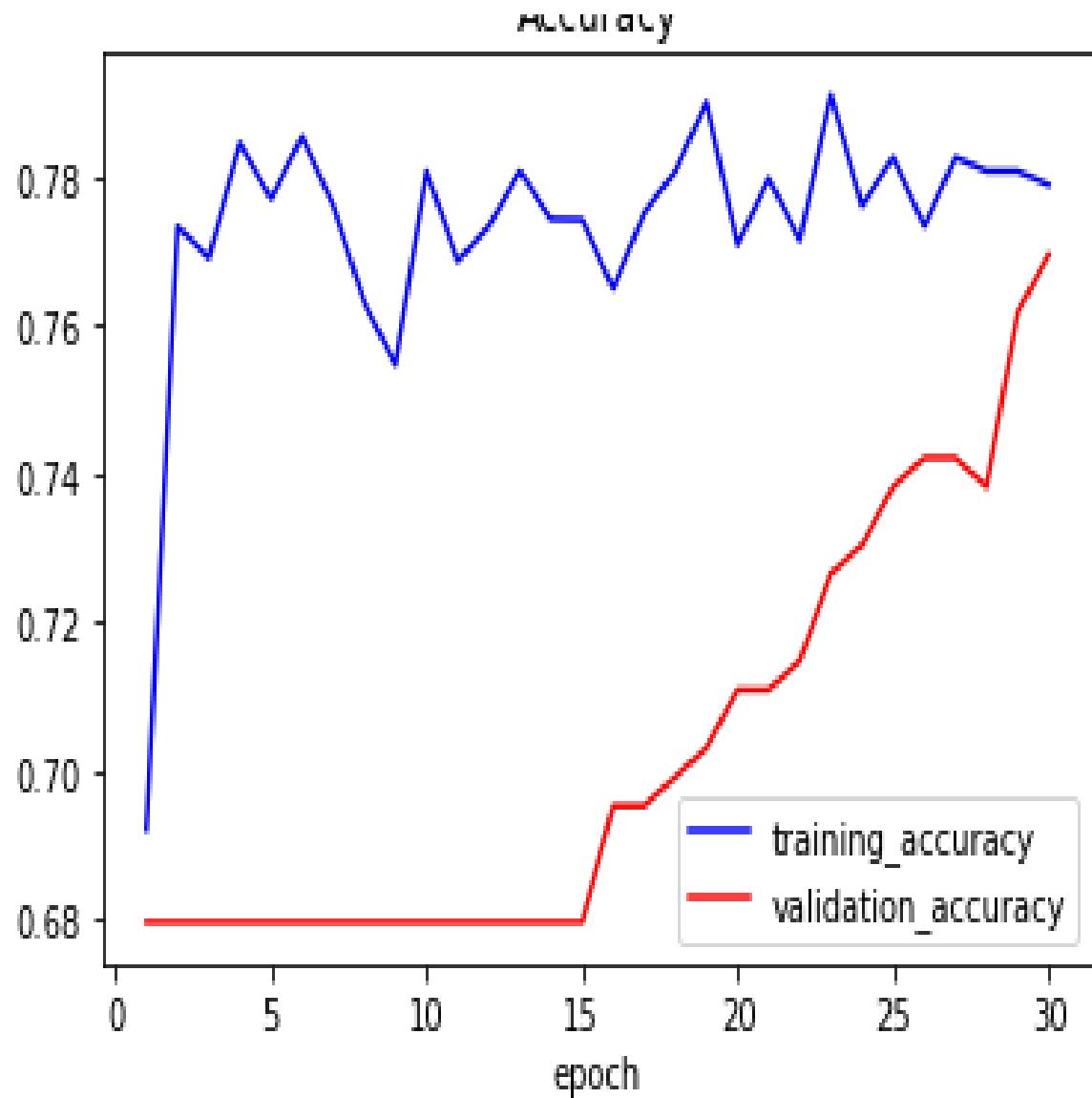


Figure 6.21: Train and Validation Accuracy

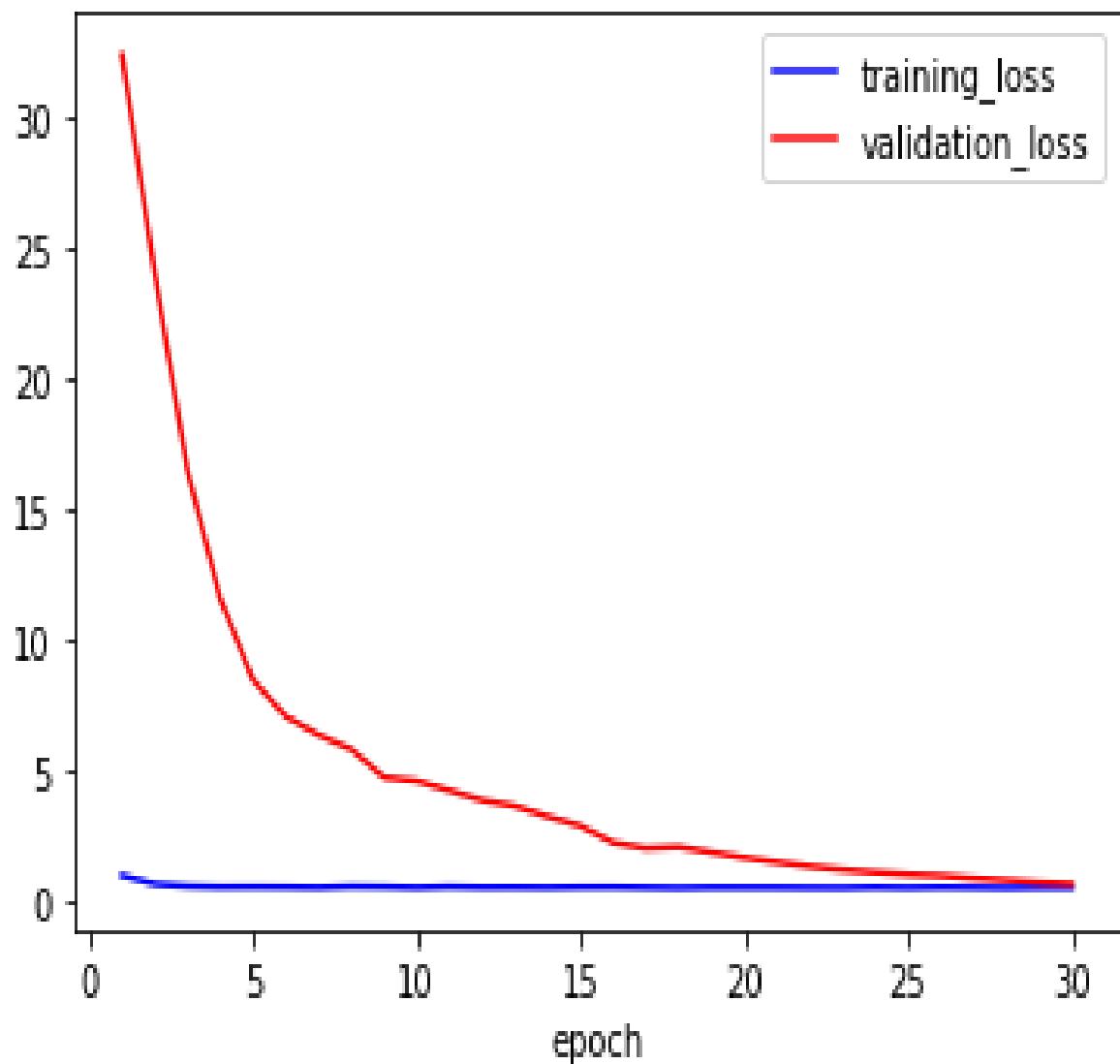


Figure 6.22: Train and Validation Loss

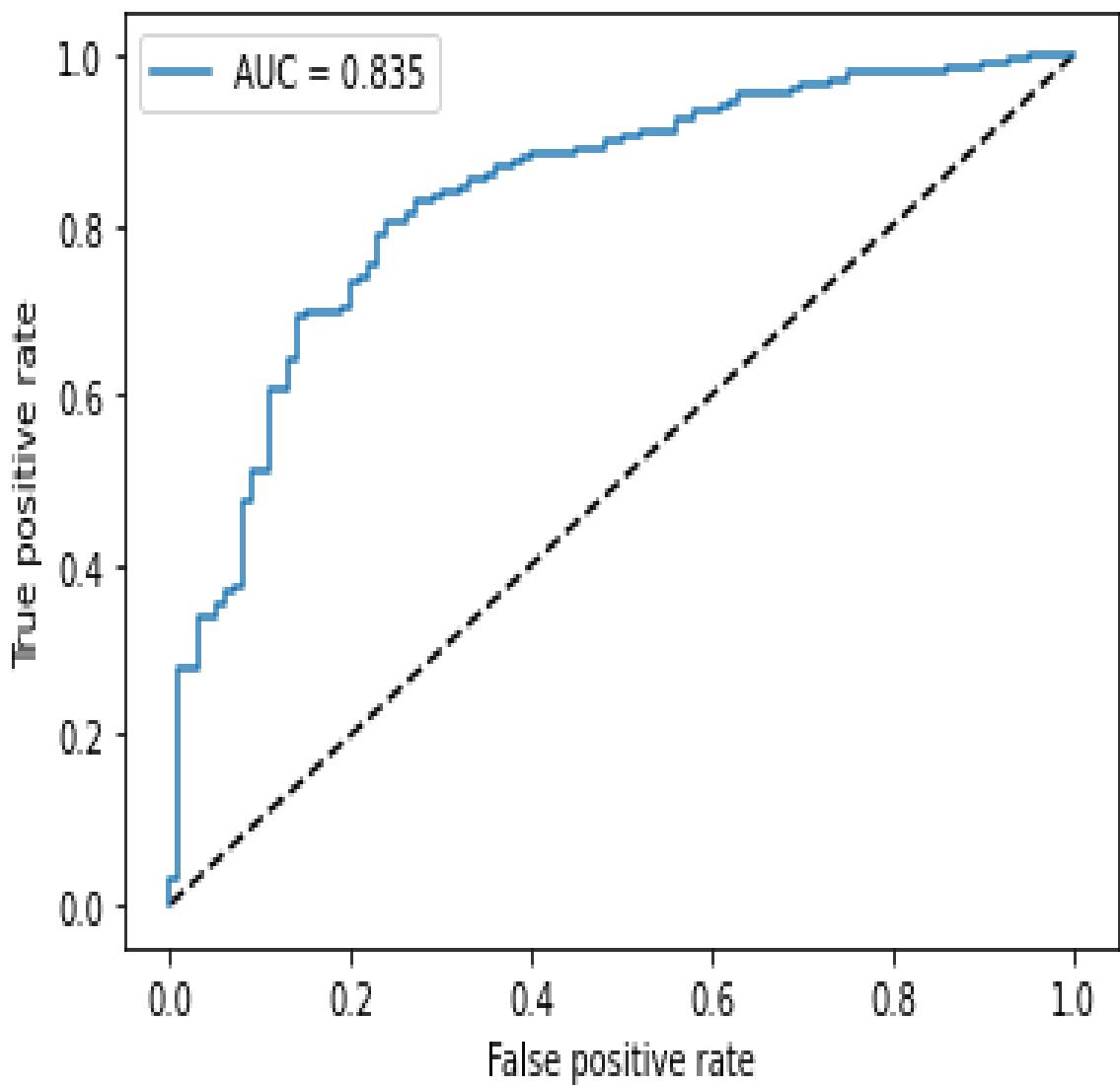


Figure 6.23: ROC-AUC
The AUC score of 0.835 is considered to be excellent.

6.1.2 Vgg16

Description

VGG16 is a convolution neural net architecture and is considered to be one of the excellent vision model architecture.VGG16 has a very uniform architecture.It is pre-trained version of the network trained on more than a million images from the ImageNet.

The VGG16 has large number of hyper-parameters instead it is very unique because it focused on having convolution layers of 3×3 filter with a stride 1 along with same padding and maxpool layer of 2×2 filter of stride 2.In the end it has 2 fully connected layers followed by a softmax for output.

The 16 in VGG16 refers it has 16 layers that have weights.This is pretty large network and it has about 138 million(approx.) parameters.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096	FC-4096	FC-4096	FC-1000	FC-1000	FC-1000
soft-max					

Figure 6.24: VGG 16 Architecture

The custom model which was used to train last few layers of our VGG16 is in the next page.

```
Model: "sequential"
-----
Layer (type)          Output Shape         Param #
-----
vgg16 (Functional)    (None, 3, 3, 512)   14714688
-----
flatten (Flatten)     (None, 4608)        0
-----
dense (Dense)         (None, 1024)        4719616
-----
dropout (Dropout)     (None, 1024)        0
-----
dense_1 (Dense)       (None, 512)         524800
-----
dropout_1 (Dropout)   (None, 512)         0
-----
dense_2 (Dense)       (None, 2)           1026
-----
Total params: 19,960,130
Trainable params: 19,960,130
Non-trainable params: 0
```

Figure 6.25: Custom model to train last few layers

1000 of each class - Balanced Dataset

The VGG16 model is pre-trained on ImageNet dataset. Our images are much smaller than the images on ImageNet. Therefore we have trained the last few layers of model instead of just the last one.

Also, we have resized the images before training to get much higher performance. The training has to be done on 30 epochs.

	<i>id</i>		<i>label</i>
0	f51542333ecdc504a775b695c3b15f8175b85640		0.0
1	5b8f5a93ce4f11522c6b6538a542f7c076e752f1		0.0
2	127ef5796b5c44aa30a92c13972ef3372fce47f6		0.0

Figure 6.26: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 1st and 2nd and the 3rd image has no cancer cells.

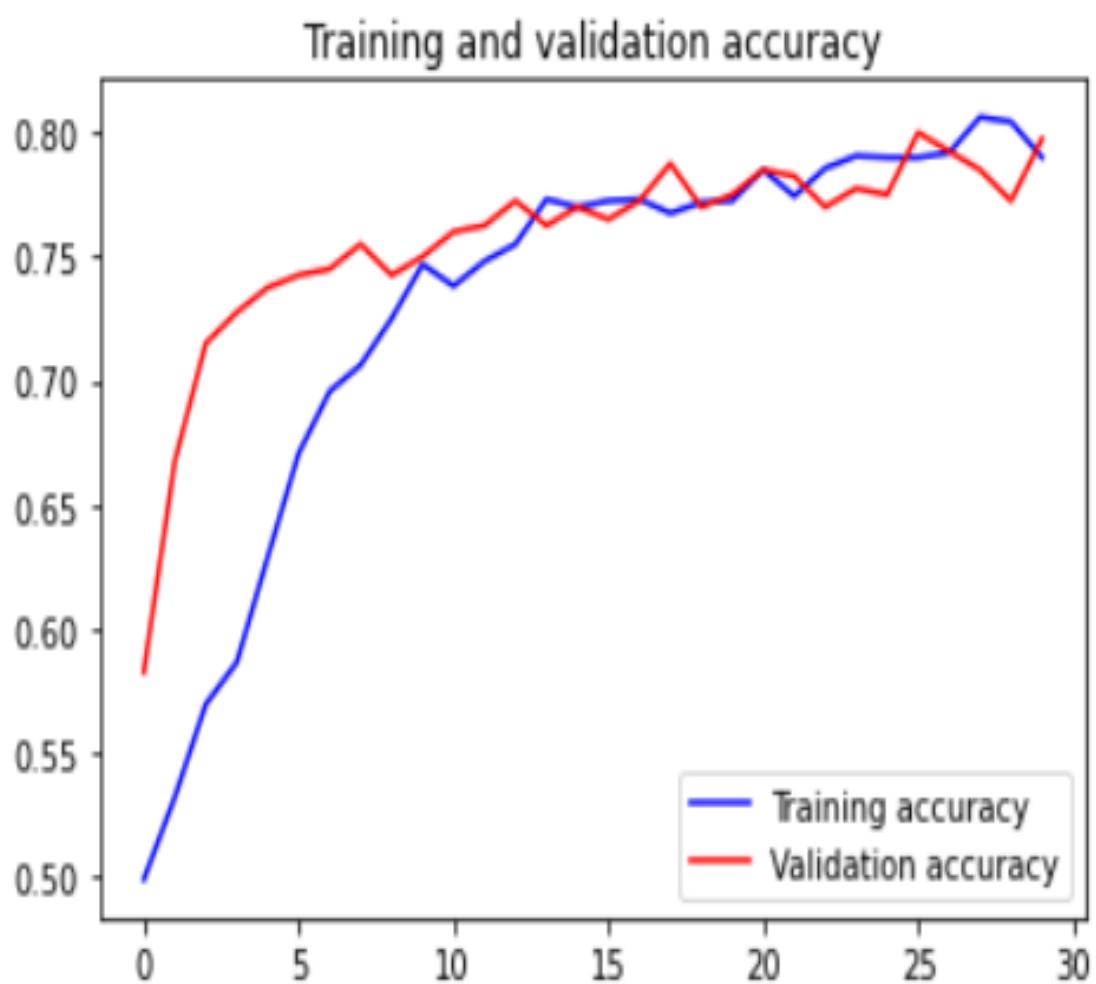


Figure 6.27: Train and Validation Accuracy

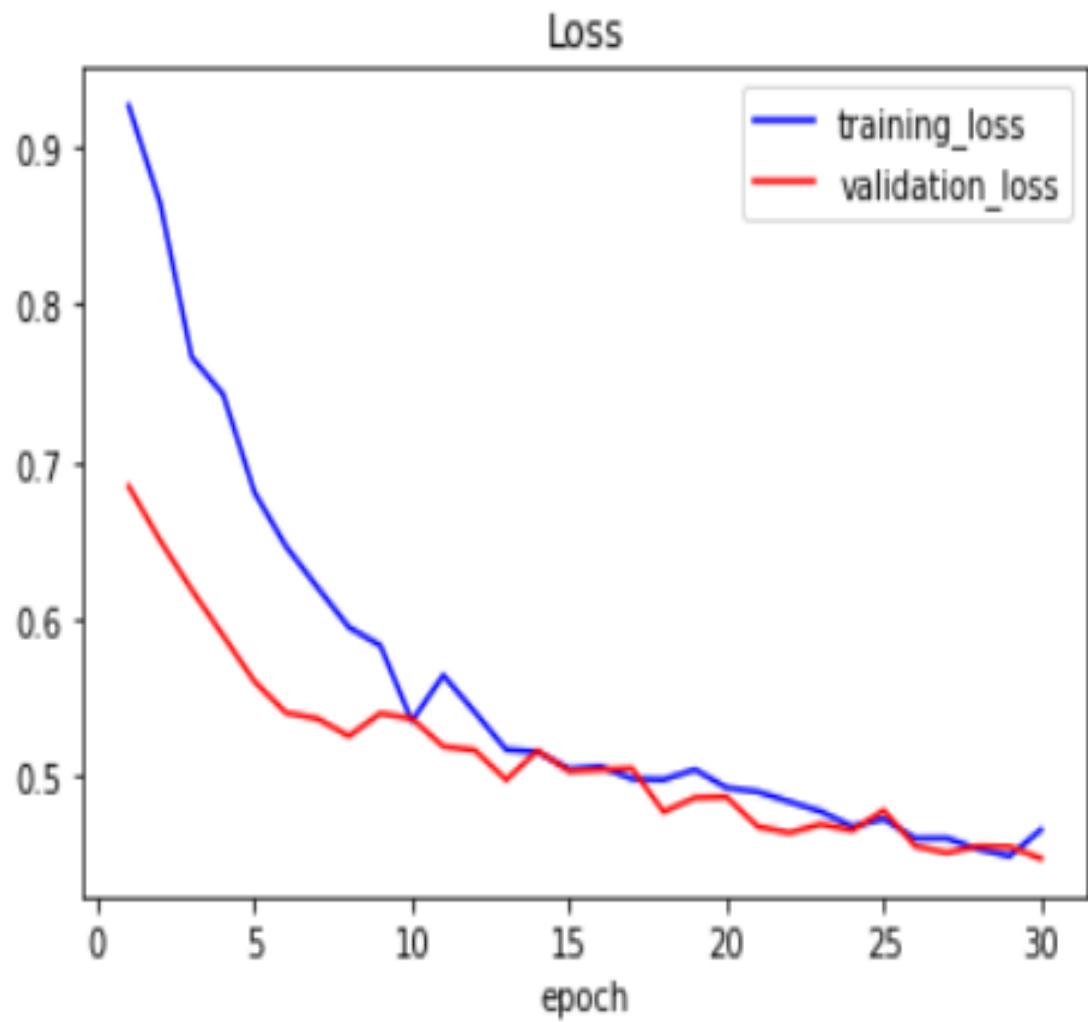


Figure 6.28: Train and Validation Loss

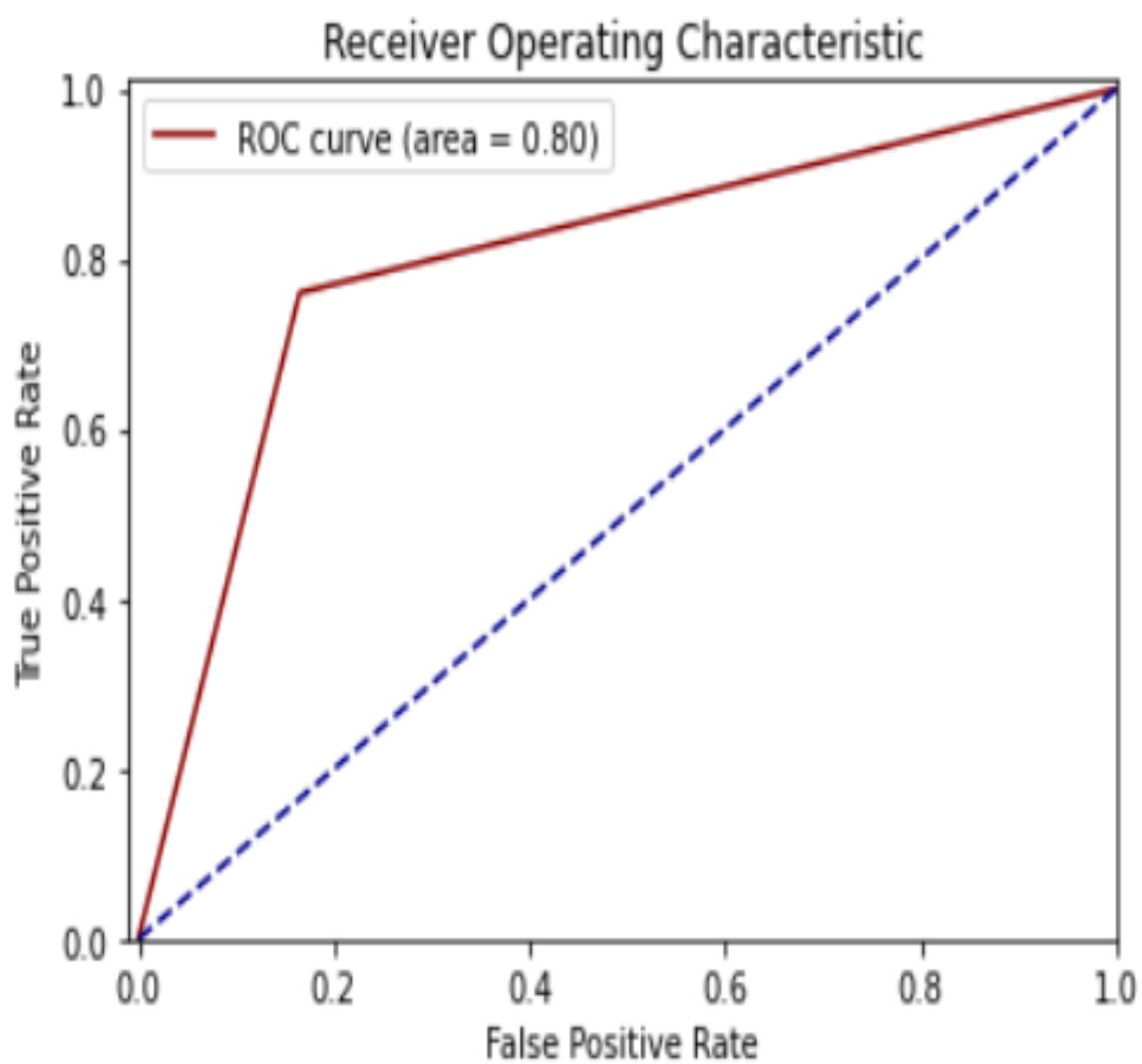


Figure 6.29: ROC-AUC
The AUC score of 0.80 is considered to be excellent.

500 of each class - Balanced Dataset

	id	label
0	f51542333ecdc504a775b695c3b15f8175b85640	0.0
1	5b8f5a93ce4f11522c6b6538a542f7c076e752f1	0.0
2	127ef5796b5c44aa30a92c13972ef3372fce47f6	0.0
3	1ed6bfe0719062b3e3ebbb5d742592587a6b2885	0.0
4	f5193eb47bf1b036ff63282c74c6cf2a5a00919	0.0
5	562d8190e3877da04576fa5be1012ef3aa252e04	0.0
6	52c233896fbfb722f2918cf208d8b789f491160e	0.0
7	50e20354f48f57f010000bbd070dd8b4e2aa56ed	0.0
8	53de489a281d84dc1dd478e4fbf541b5fc4e04ba	0.0
9	635a5eb4c271f7bbdb6d0e798d533c4e38bf1a84	0.0

Figure 6.30: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, all the images shown have no cancer cells.

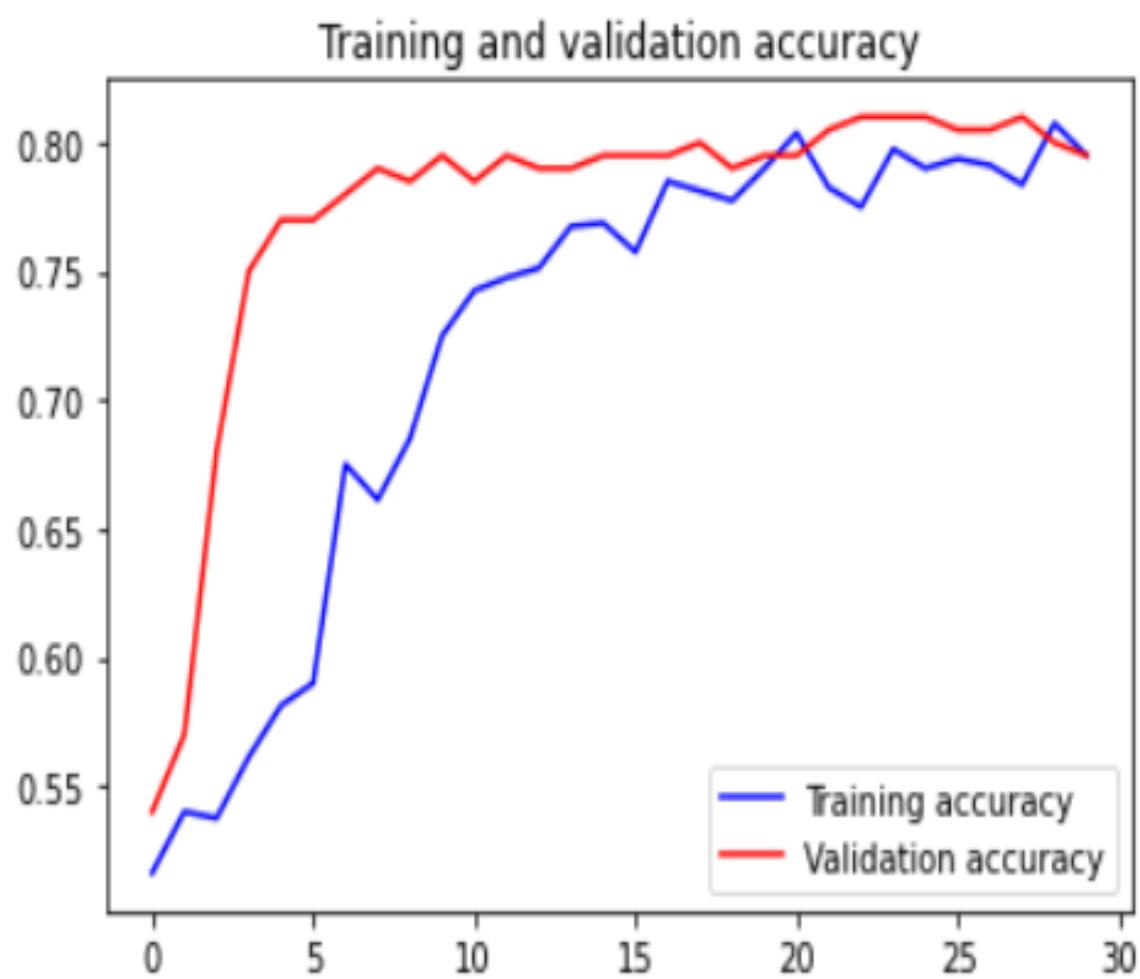


Figure 6.31: Train and Validation Accuracy

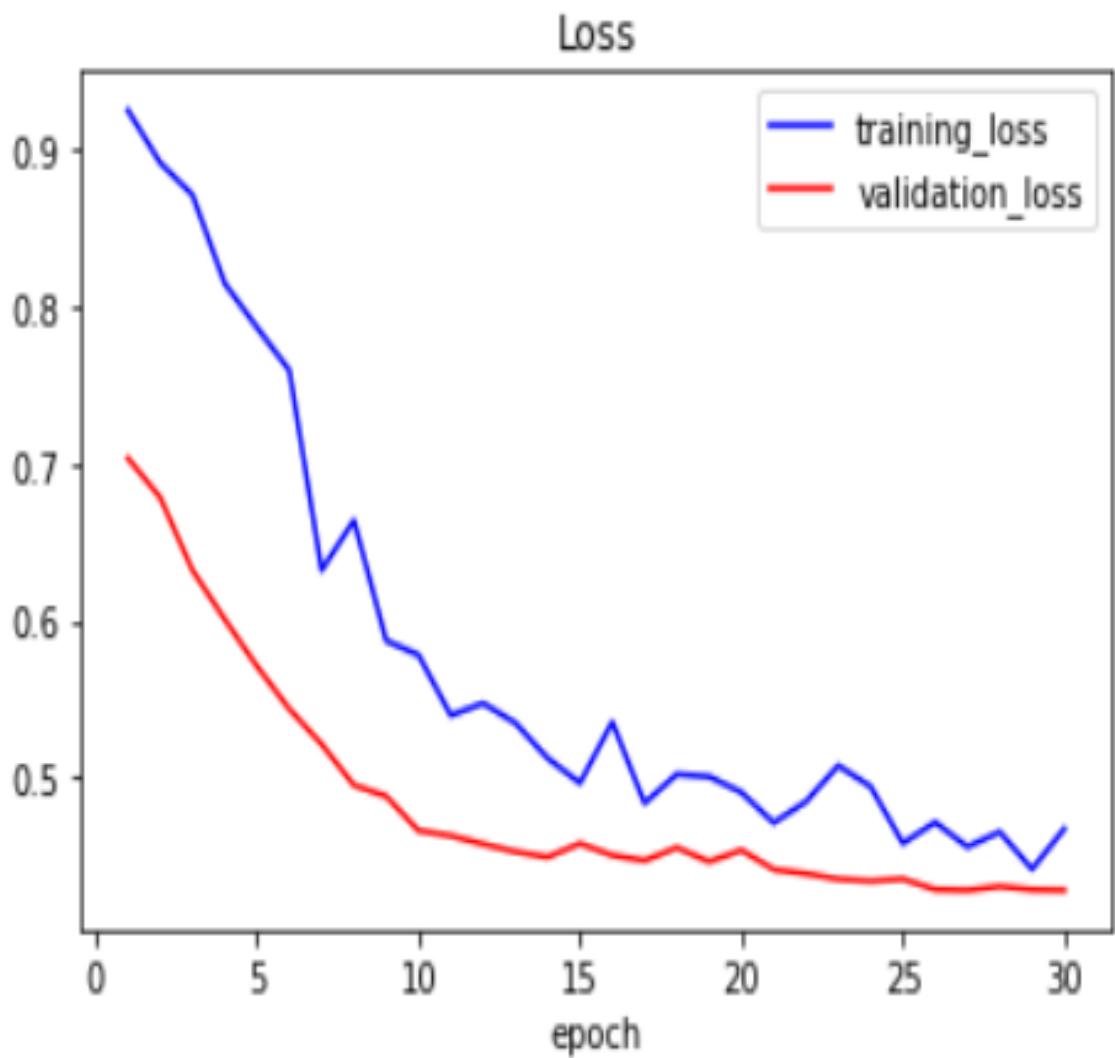


Figure 6.32: Train and Validation Loss

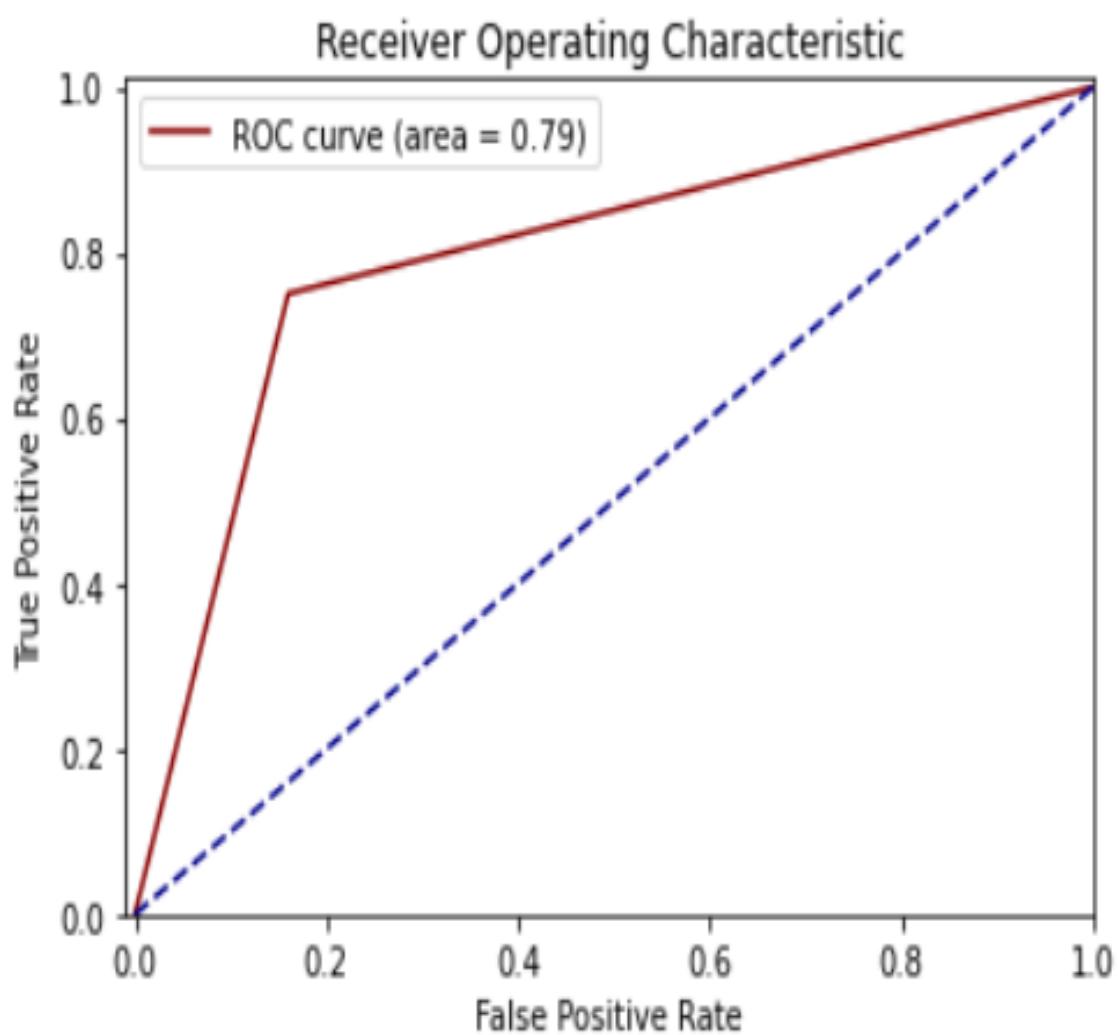


Figure 6.33: ROC-AUC
The AUC score of 0.79 is considered to be acceptable.

100 of each class - Balanced Dataset

	id	label
0	f51542333ecdc504a775b695c3b15f8175b85640	0.0
1	5b8f5a93ce4f11522c6b6538a542f7c076e752f1	0.0
2	127ef5796b5c44aa30a92c13972ef3372fce47f6	0.0

Figure 6.34: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 1st, 2nd and 3rd images do not have cancer cells.

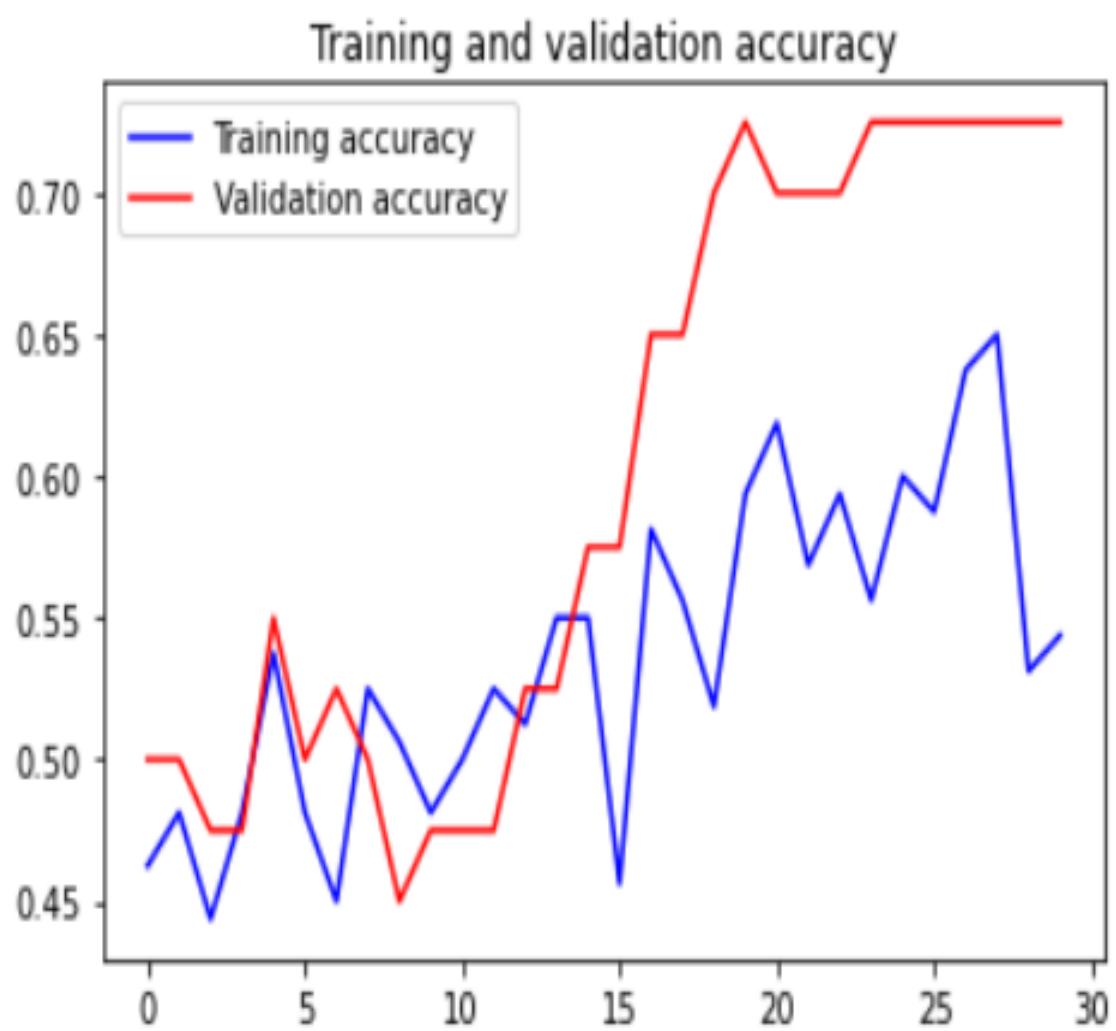


Figure 6.35: Train and Validation Accuracy

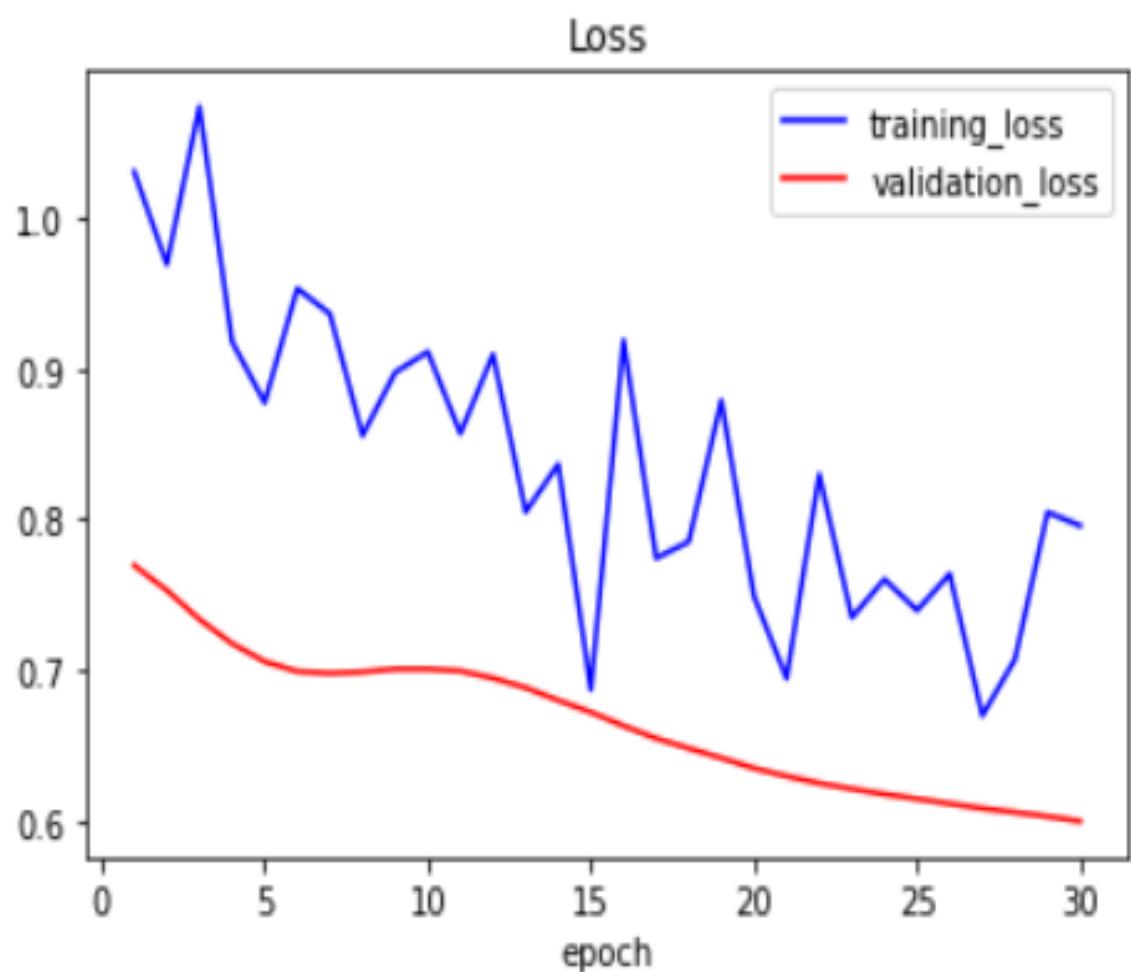


Figure 6.36: Train and Validation Loss

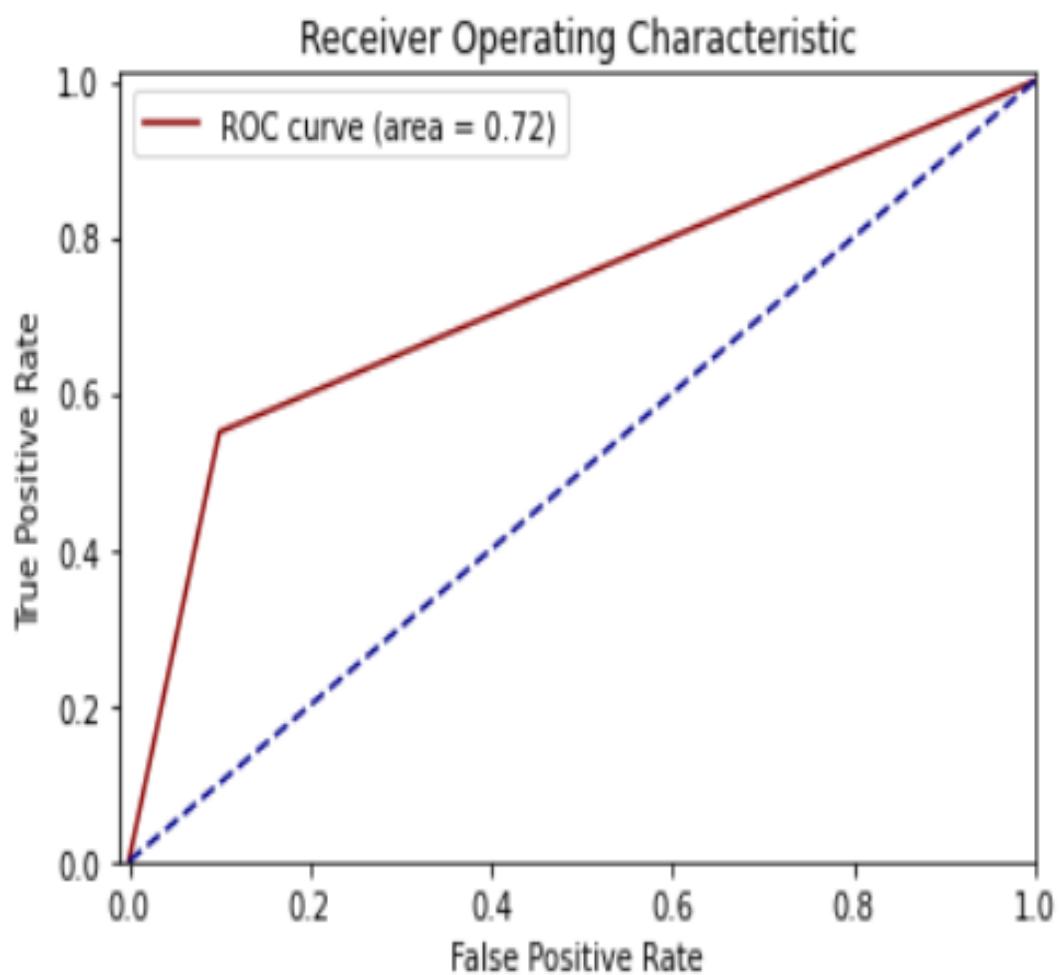


Figure 6.37: ROC-AUC
The AUC score of 0.72 is considered to be acceptable.

1000 (no cancer cells) 500 (cancer cells) - Unbalanced Dataset

	id	label
0	f51542333ecdc504a775b695c3b15f8175b85640	1.0
1	5b8f5a93ce4f11522c6b6538a542f7c076e752f1	0.0
2	127ef5796b5c44aa30a92c13972ef3372fce47f6	0.0
3	1ed6bfe0719062b3e3ebbb5d742592587a6b2885	0.0
4	f5193eb47bf1b036ff63282c74c6cf2a5a00919	0.0
5	562d8190e3877da04576fa5be1012ef3aa252e04	0.0
6	52c233896fbfb722f2918cf208d8b789f491160e	0.0
7	50e20354f48f57f010000bbd070dd8b4e2aa56ed	0.0
8	53de489a281d84dc1dd478e4fbf541b5fc4e04ba	0.0
9	635a5eb4c271f7bbdb6d0e798d533c4e38bf1a84	0.0

Figure 6.38: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 1st image have cancer cells while the other images shown has no cancer cells.

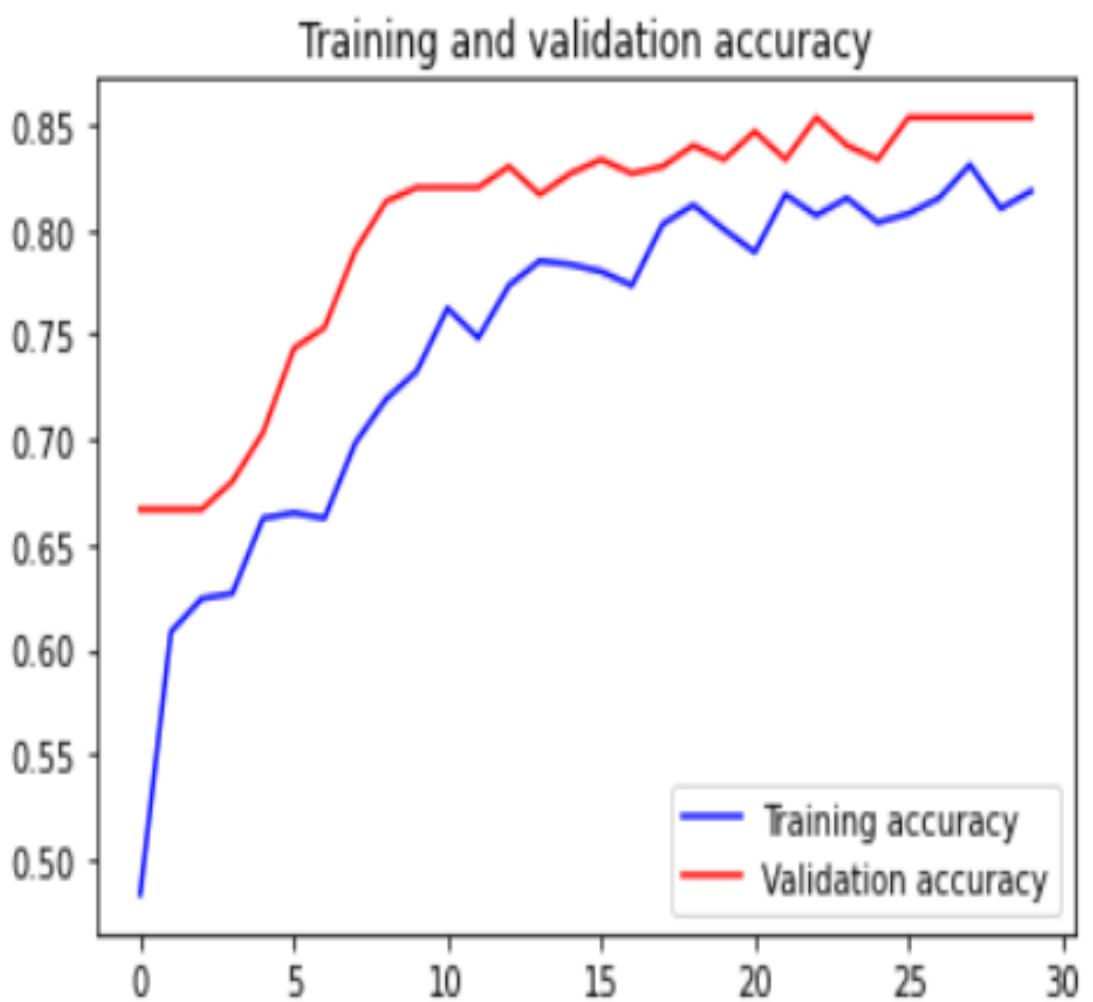


Figure 6.39: Train and Validation Accuracy

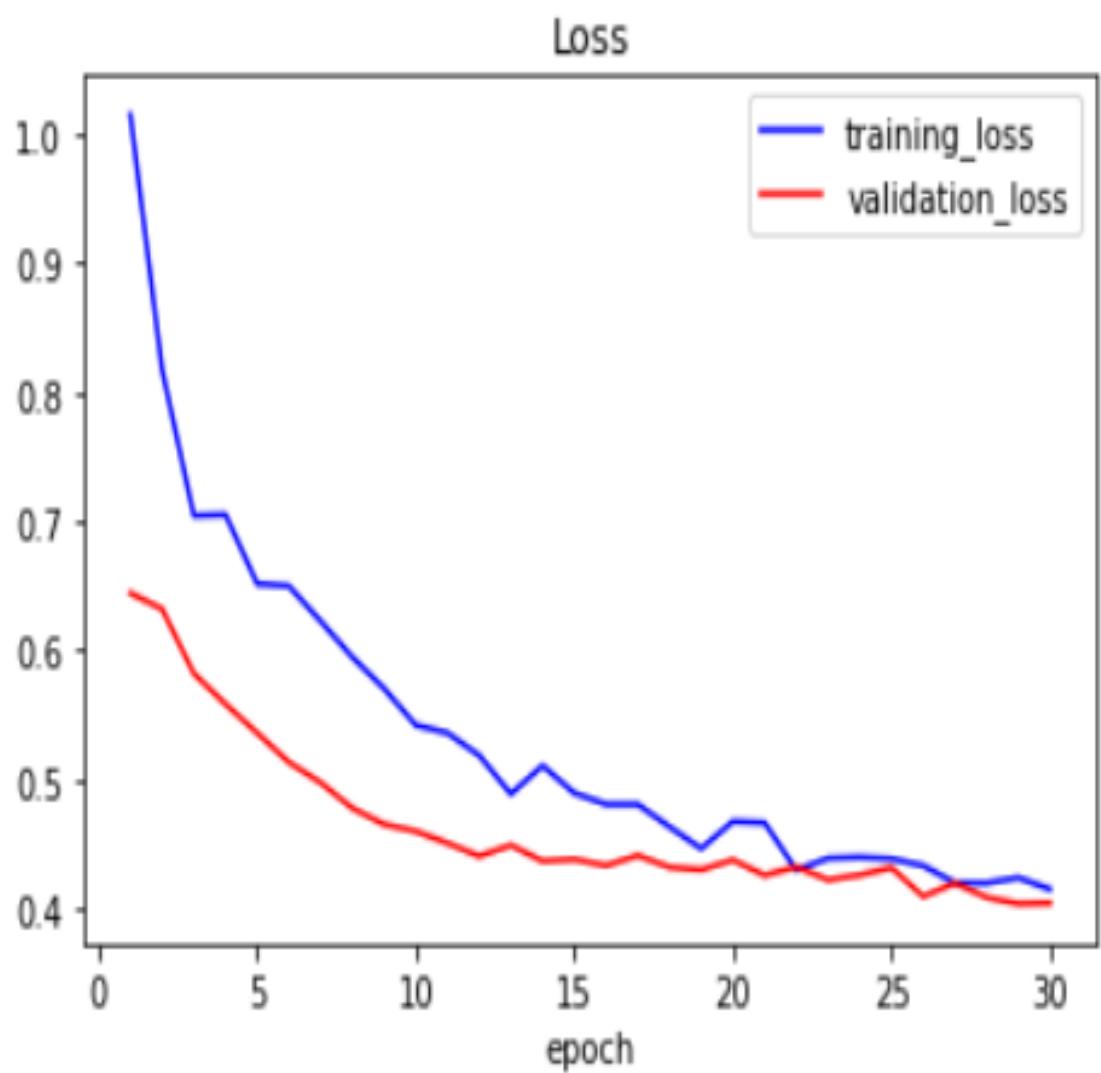


Figure 6.40: Train and Validation Loss

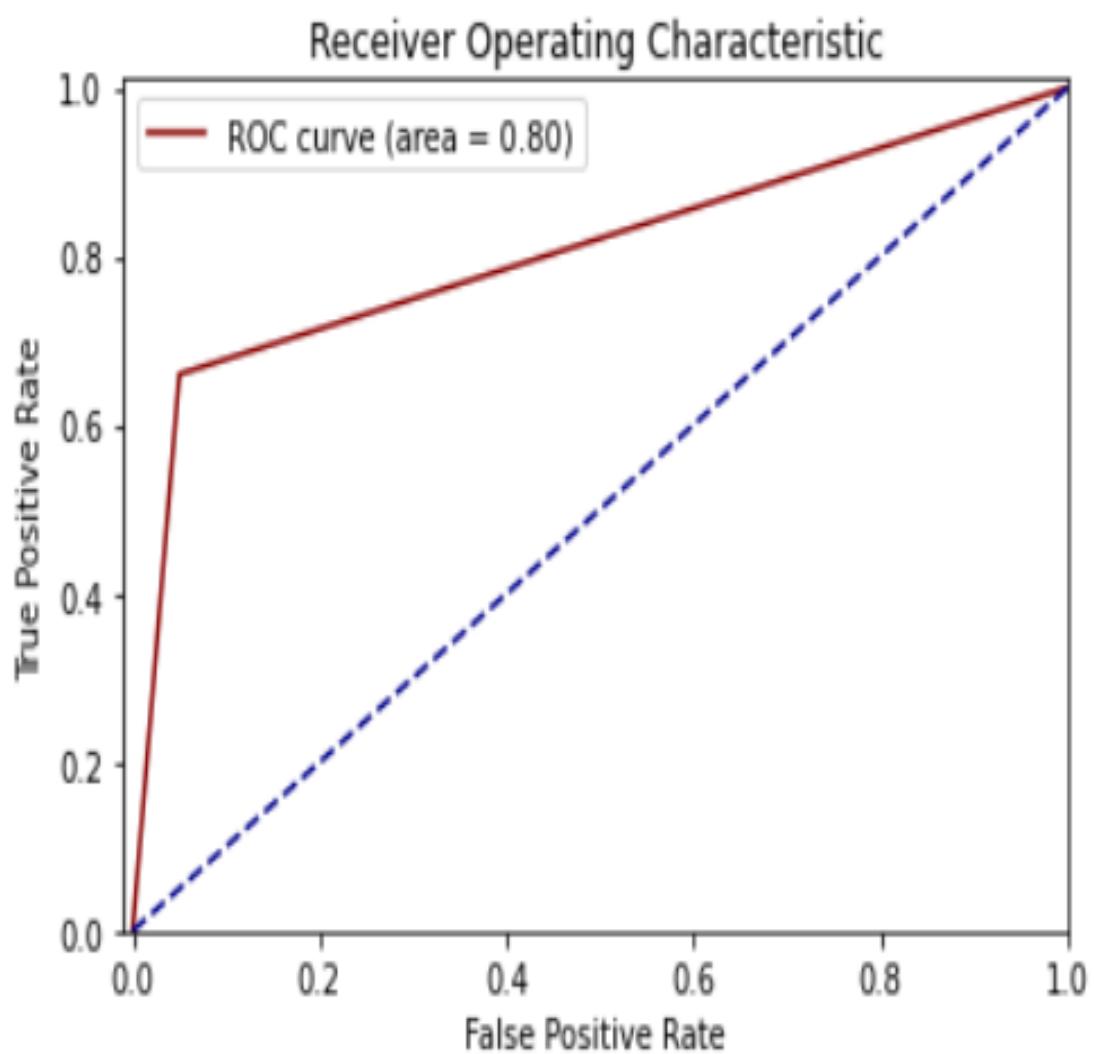


Figure 6.41: ROC-AUC
The AUC score of 0.80 is considered to be excellent.

500 (no cancer cells) 1000 (cancer cells) - Unbalanced Dataset

	id	label
0	f51542333ecdc504a775b695c3b15f8175b85640	0.0
1	5b8f5a93ce4f11522c6b6538a542f7c076e752f1	0.0
2	127ef5796b5c44aa30a92c13972ef3372fce47f6	0.0
3	1ed6bfe0719062b3e3ebbb5d742592587a6b2885	0.0
4	f5193eb47bf1b036ff63282c74c6cf2a5a00919	0.0
5	562d8190e3877da04576fa5be1012ef3aa252e04	0.0
6	52c233896fbfb722f2918cf208d8b789f491160e	0.0
7	50e20354f48f57f010000bbd070dd8b4e2aa56ed	0.0
8	53de489a281d84dc1dd478e4fbf541b5fc4e04ba	0.0
9	635a5eb4c271f7bbdb6d0e798d533c4e38bf1a84	0.0

Figure 6.42: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, shown images do not have cancer cells.

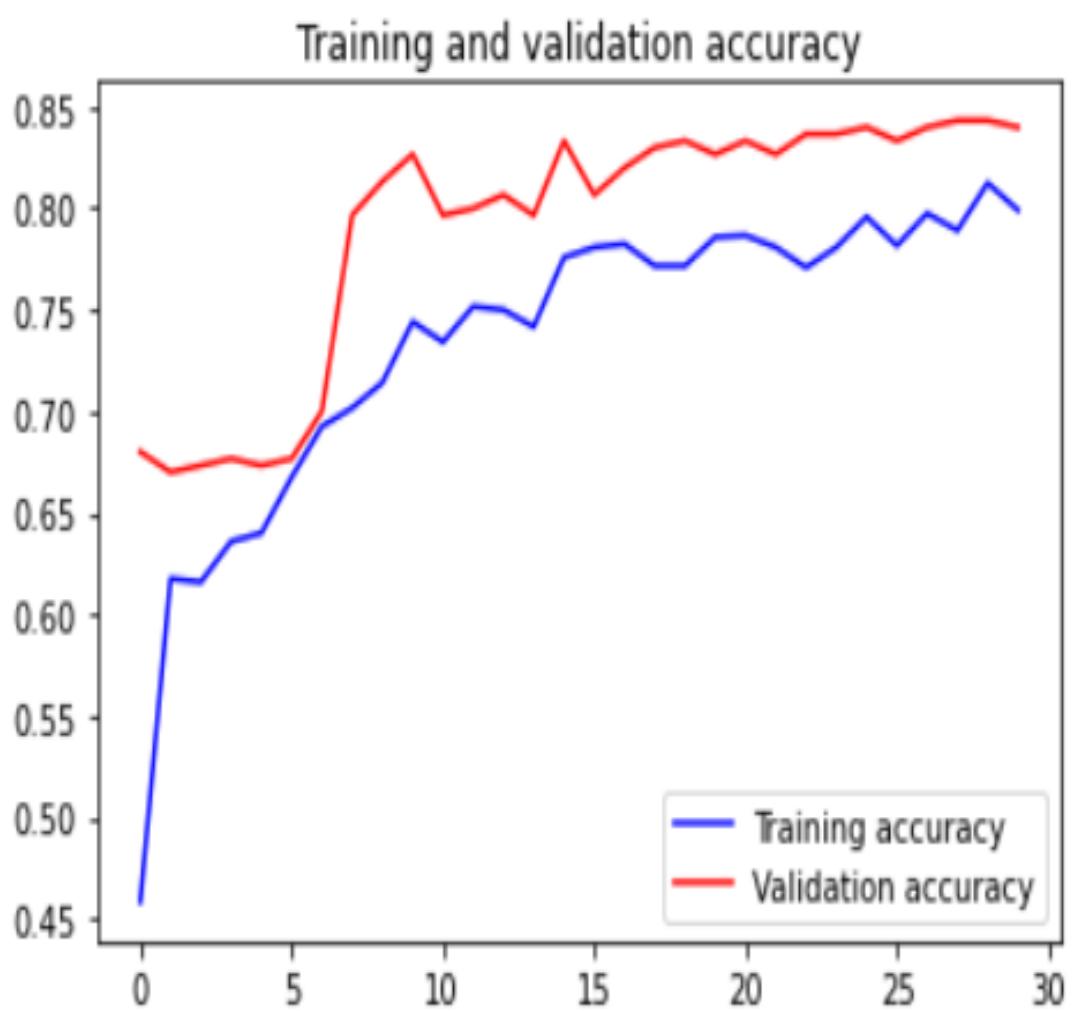


Figure 6.43: Train and Validation Accuracy

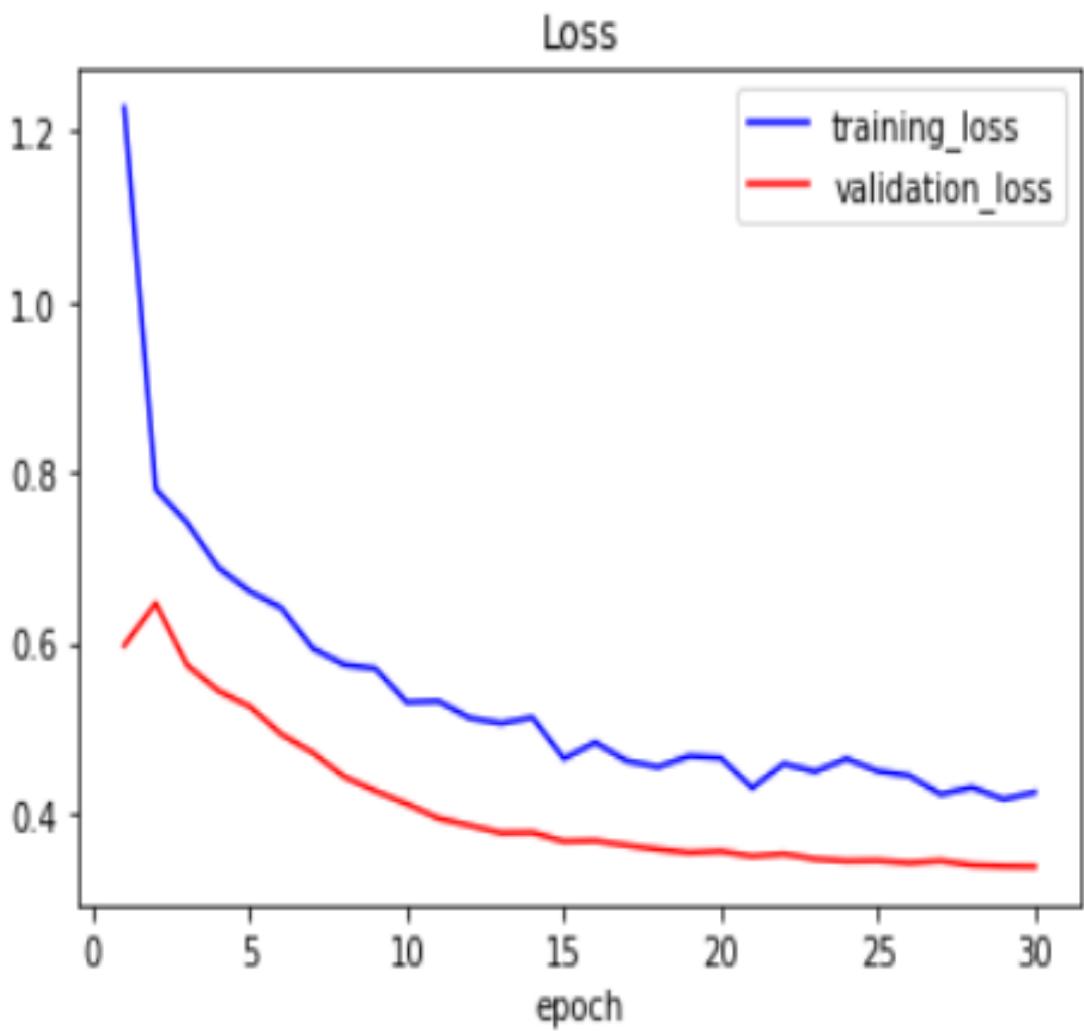


Figure 6.44: Train and Validation Loss

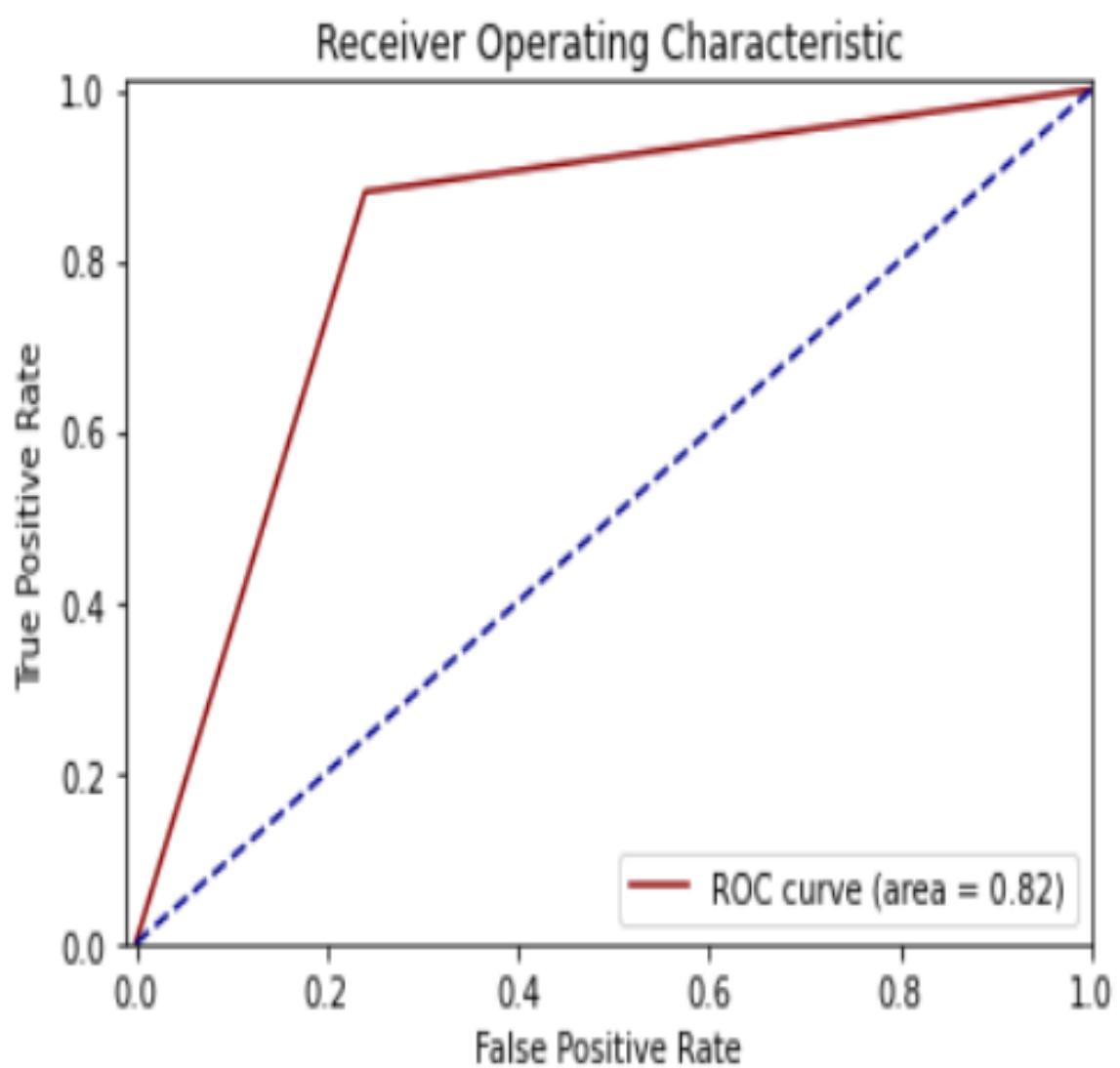


Figure 6.45: ROC-AUC
The AUC score of 0.82 is considered to be excellent.

6.1.3 Custom Convolution Neural Network (CNN)

Description

A convolution neural network is a deep learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.

The preprocessing required in a convNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

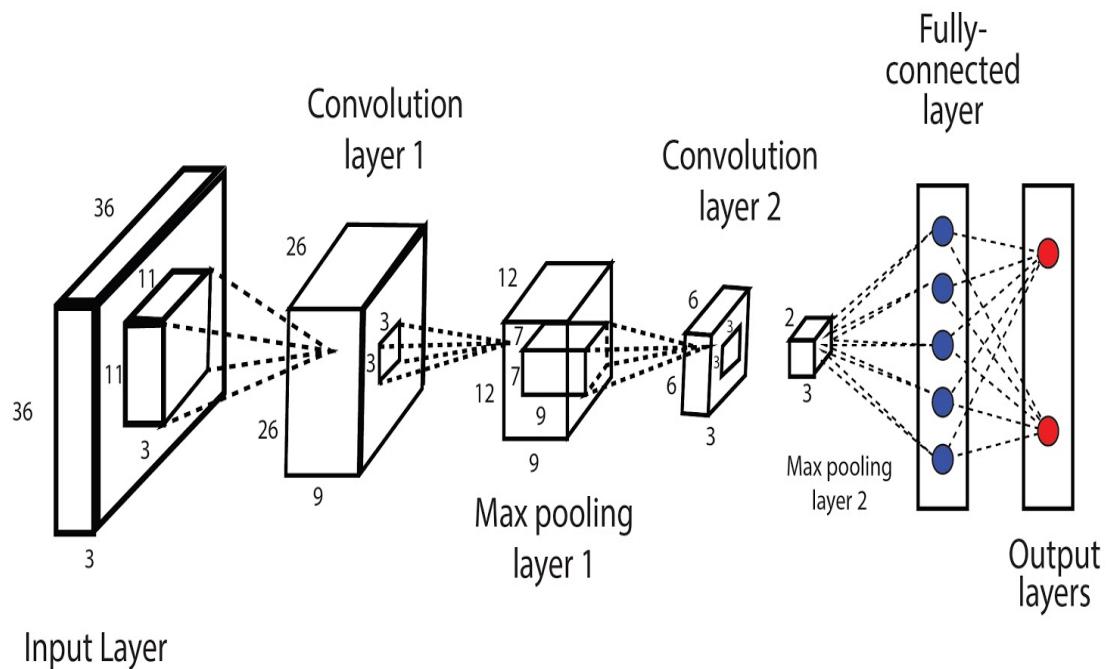


Figure 6.46: CNN Architecture

The custom model which was used to train last few layers of our CNN is in the next page.

```

Model: "sequential"

Layer (type)          Output Shape         Param #
=====
conv2d (Conv2D)        (None, 194, 194, 32)    896
conv2d_1 (Conv2D)      (None, 192, 192, 32)    9248
conv2d_2 (Conv2D)      (None, 190, 190, 32)    9248
max_pooling2d (MaxPooling2D) (None, 95, 95, 32)    0
dropout (Dropout)      (None, 95, 95, 32)    0
conv2d_3 (Conv2D)      (None, 93, 93, 64)     18496
conv2d_4 (Conv2D)      (None, 91, 91, 64)     36928
conv2d_5 (Conv2D)      (None, 89, 89, 64)     36928
max_pooling2d_1 (MaxPooling2D) (None, 44, 44, 64)    0
dropout_1 (Dropout)    (None, 44, 44, 64)    0
conv2d_6 (Conv2D)      (None, 42, 42, 128)    73856
conv2d_7 (Conv2D)      (None, 40, 40, 128)    147584
conv2d_8 (Conv2D)      (None, 38, 38, 128)    147584
max_pooling2d_2 (MaxPooling2D) (None, 19, 19, 128)    0
dropout_2 (Dropout)    (None, 19, 19, 128)    0
flatten (Flatten)     (None, 46208)       0
dense (Dense)          (None, 256)        11829504
dropout_3 (Dropout)    (None, 256)        0
dense_1 (Dense)        (None, 2)          514
=====

Total params: 12,310,786
Trainable params: 12,310,786
Non-trainable params: 0

```

Figure 6.47: Custom Model to train last few layers

1000 of each class - Balanced Dataset

The CNN model is pre-trained on ImageNet dataset. Our images are much smaller than the images on ImageNet. Therefore we have trained the last few layers of model instead of just the last one.

Also, we have resized the images before training to get much higher performance. The training has to be done on 30 epochs.

	label
id	
0037f55a37a6549f1b552e20206d380c24217ec4	0.0
00e83ffb49800dc3a24eb04f8d512b712b526223	1.0
034180e47a2d092c2c6f015fdd917351e34d3904	0.0
045c402c5c75088329c166370daf3eb825be91ae	0.0
056a21ad079dd54c709b1c994474a8aa42ef51b4	1.0

Figure 6.48: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 2nd and 5th images have cancer cells while the other images shown have no cancer cells.

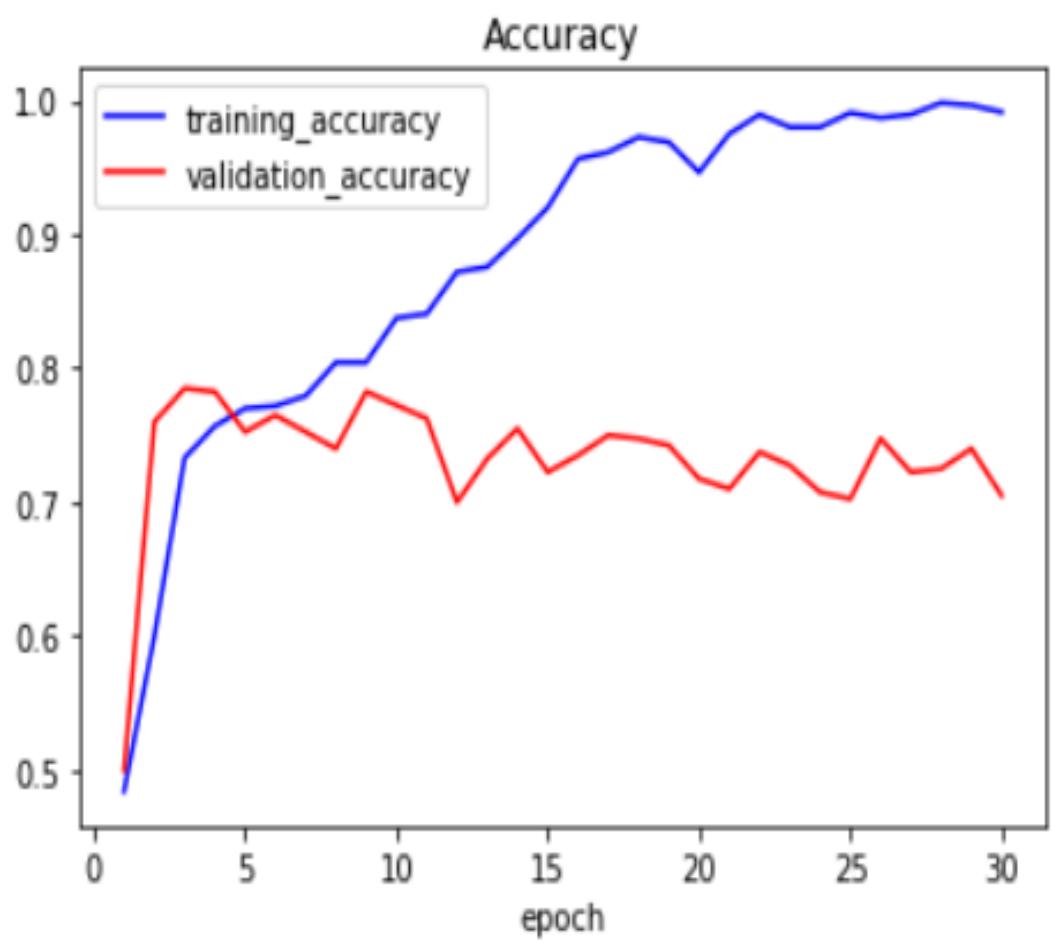


Figure 6.49: Train and Validation Accuracy

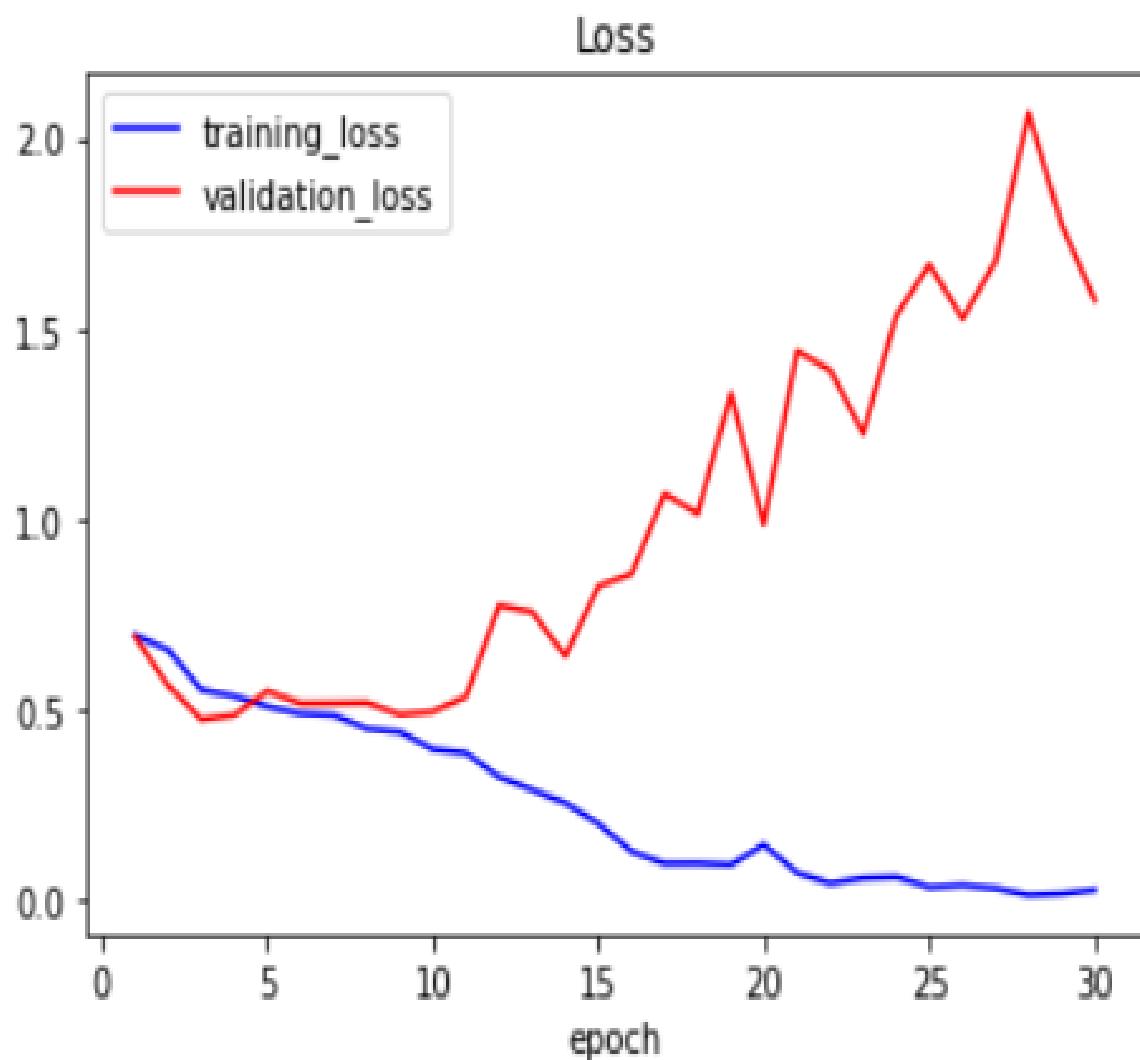


Figure 6.50: Train and Validation Loss



Figure 6.51: ROC-AUC SCORE
The AUC score of 0.762 is considered to be acceptable.

500 of each class - Balanced Dataset

	label
id	
01cb05abca279f6092c1c0990520a562e0456126	1.0
037536173143d2a61a8ba0d6b949064d75024671	0.0
07af628e68bf1d2afd4cabfbeae344290ea6f2eb	0.0
0a9bdf1aa4d9fc3a10f2f5728812273b79c9843e	0.0
0cfb46304aa4536db8e450773ce3a56110e22257	0.0

Figure 6.52: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 1st image have cancer cells and other images shown do not have cancer cells.



Figure 6.53: Train and Validation Accuracy



Figure 6.54: Train and Validation Loss



Figure 6.55: ROC-AUC SCORE
The AUC score of 0.748 is considered to be acceptable.

100 of each class - Balanced Dataset

	label
id	
0f7883a06cd64e36501edfb93a63b914284e1d74	1.0
10b0dc0bb7fbdd0e17f4b35d819c0d6e7be4421d	0.0
135f91d4601bc5291571f1d52bc4773d6859ef05	0.0
158d72ef53152016970b56c72d29f0896bd95c2f	0.0
18abf4933040b57c2097b4c5b81f5157c73d2f32	0.0

Figure 6.56: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 1stimage has cancer cells. While other images shown do not have cancer cells.



Figure 6.57: Train and Validation Accuracy

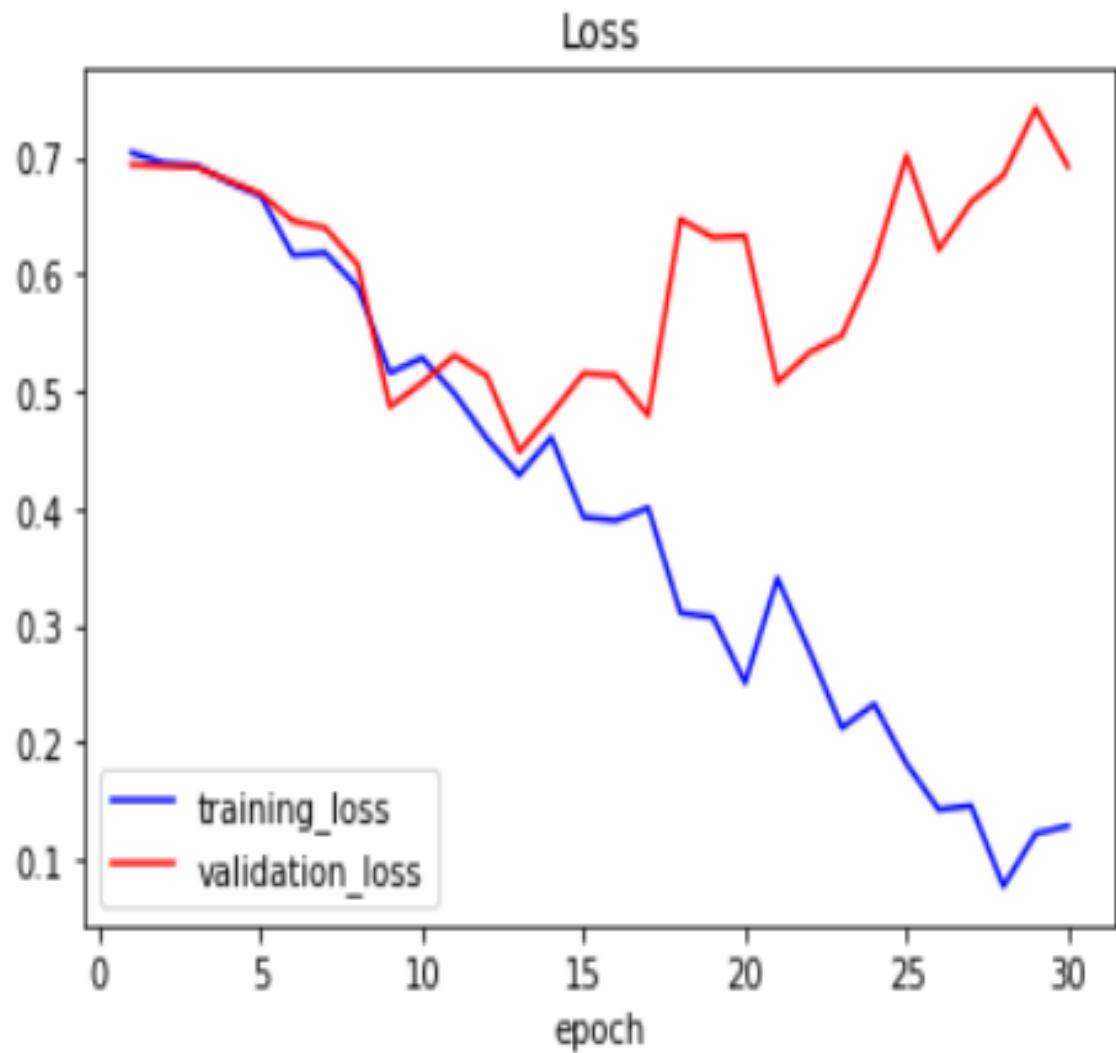


Figure 6.58: Train and Validation Loss



Figure 6.59: ROC-AUC SCORE
The AUC score of 0.86 is considered to be excellent.

1000 (no cancer cells) 500 (cancer cells) - Unbalanced Dataset

	label
id	
034180e47a2d092c2c6f015fdd917351e34d3904	0.0
03c117c4dc060862e43847e23d2c0cdeaca4bdac	0.0
056a21ad079dd54c709b1c994474a8aa42ef51b4	1.0
06fee08f7dc39a9b4637749d974235717d59e7f6	0.0
0934c40cbcebe2d2568e663ec714c47369689435	0.0

Figure 6.60: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 3rd image have cancer cells while the other images shown has no cancer cells.

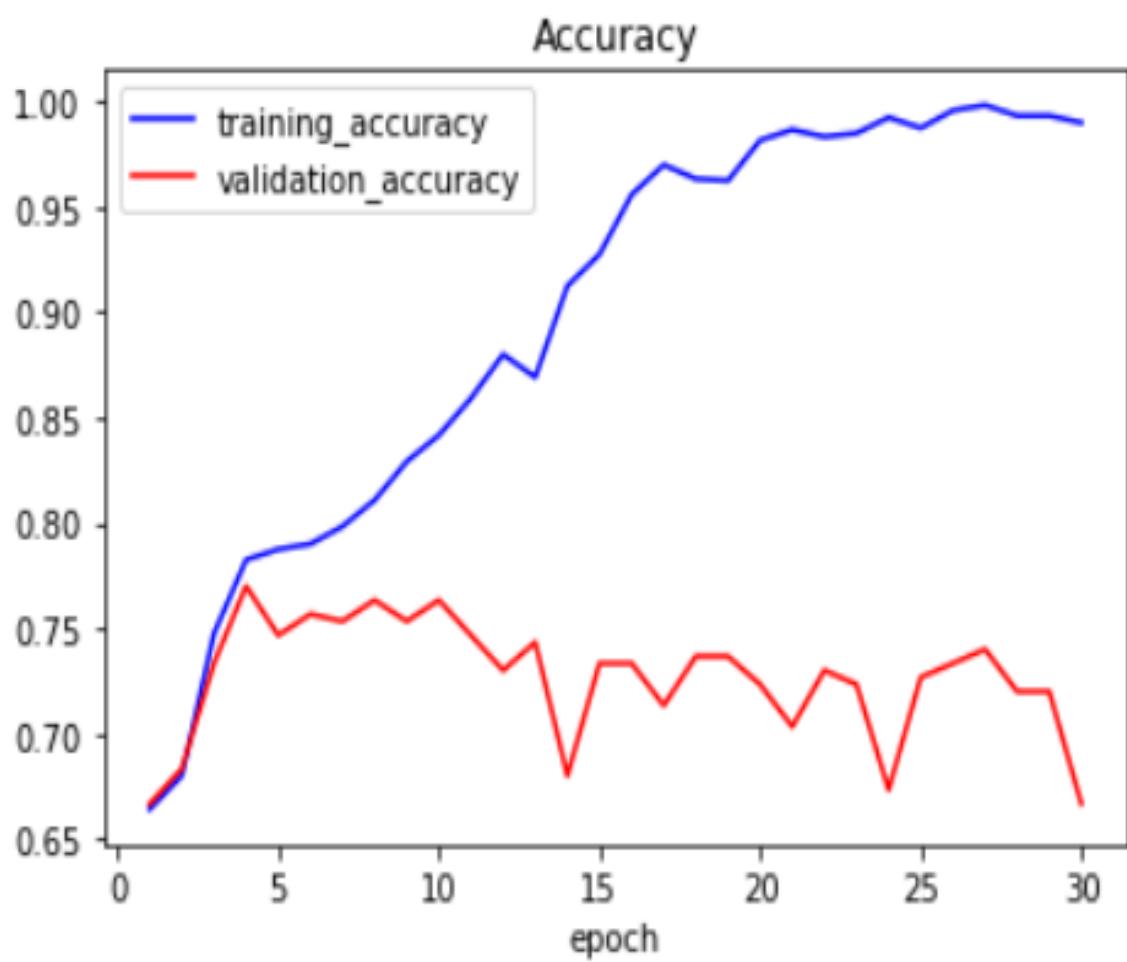


Figure 6.61: Train and Validation Accuracy



Figure 6.62: Train and Validation Loss



Figure 6.63: ROC-AUC SCORE
The AUC score of 0.723 is considered to be acceptable.

500 (no cancer cells) 1000 (cancer cells) - Unbalanced Dataset

	label
id	
0150591da689866d1c35e99cc2af4f3afaafdd38	1.0
0423186255e941acc74d840ed7bf908cc332d958	0.0
0680a534938ebc52cd573cd30dbb8aaf8cb929d8	1.0
06fee08f7dc39a9b4637749d974235717d59e7f6	0.0
0934c40cbcebe2d2568e663ec714c47369689435	0.0

Figure 6.64: Predictions

The "id" corresponds to the ID assigned to the images. The label depicts whether there is a presence or absence of cancer cells.

In the above predictions, the 1st and 3rd images have cancer cells while other images shown do not have cancer cells.

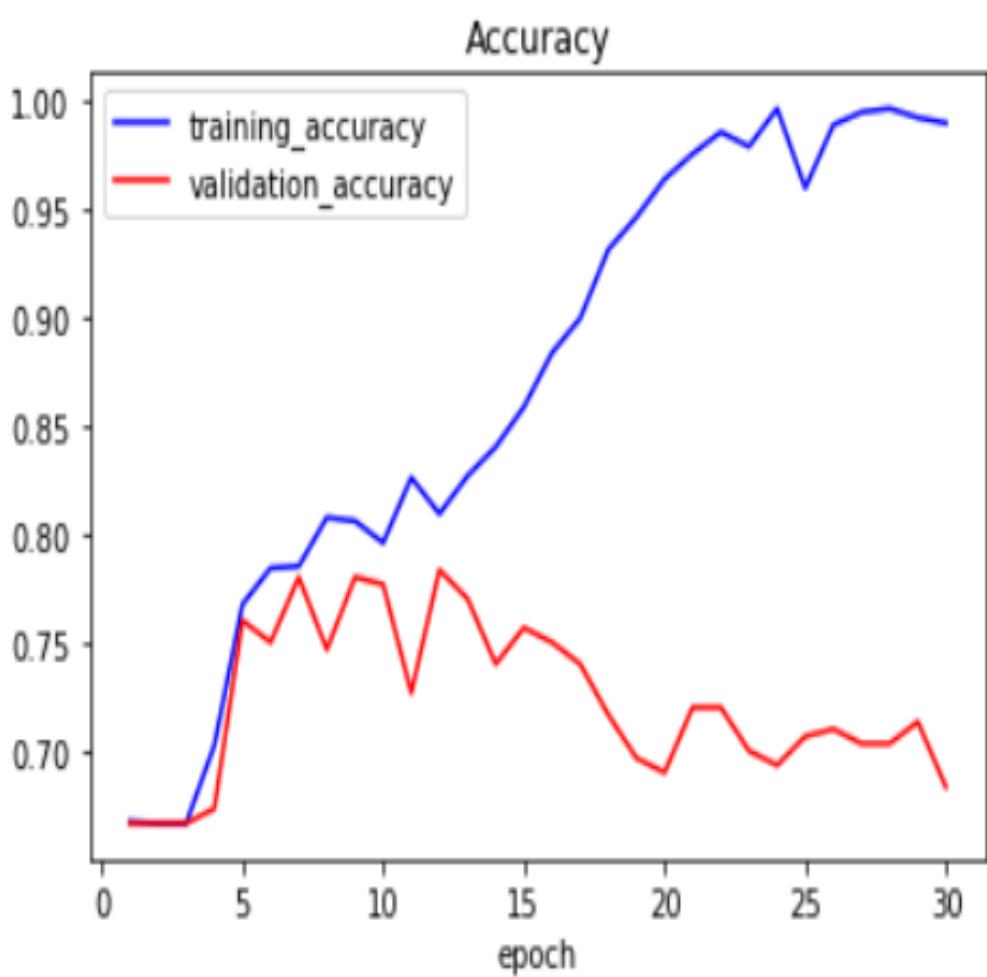


Figure 6.65: Train and Validation Accuracy

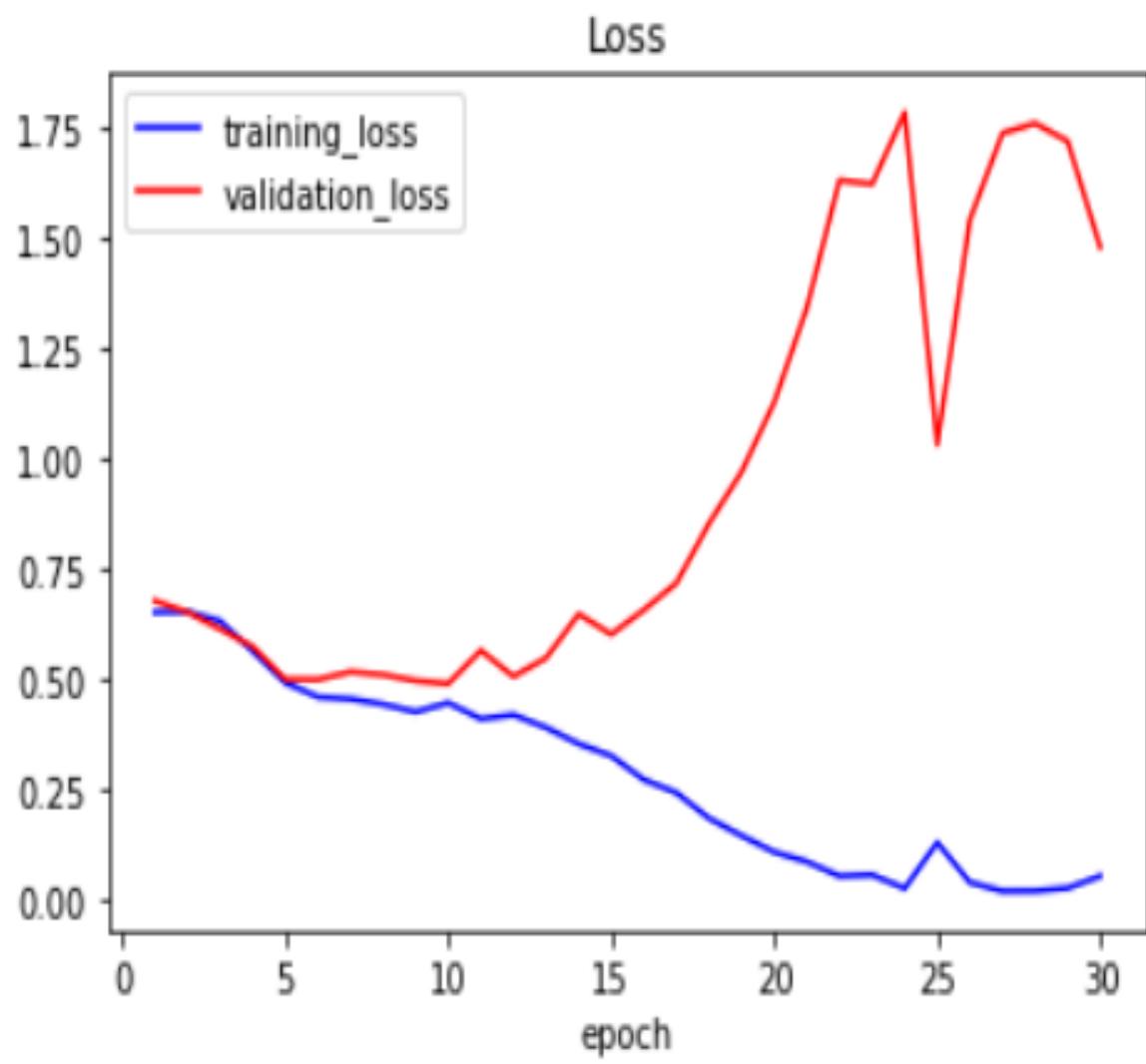


Figure 6.66: Train and Validation Loss



Figure 6.67: ROC-AUC
The AUC score of 0.695 is considered to be poor.

Chapter 7

Evaluation and Comparisons

Dataset Size	Model	ipynb name	Train Time	Epoch	Accuracy		Loss		AUC
					Train	Validation	Train	Validation	
1000 each (balanced)	CNN	CNNB1000	3.20minutes	30	99.12	70	2.76	157.38	0.762
	ResNet50	res1000	9.16 minutes	30	76.29	73.96	50.51	55.25	0.817
	Vgg16	vgg16_1000	3.09minutes	30	79	79.75	46.62	44.79	0.797
500 each (balanced)	CNN	CNNB500	1.51 minutes	30	99.83	69.5	0.7	213	0.748
	ResNet50	res500	3.75 minutes	30	75.74	61.72	51.88	108.24	0.826
	Vgg16	vgg16_500	1.48minutes	30	79.5	79.5	46.71	42.78	0.79
100 each (balanced)	CNN	CNNB100	0.51minutes	30	93.75	75	12.85	69.17	0.86
	ResNet50	res100	1.11 minute	30	72.5	75	57.76	45.48	0.877
	Vgg16	vgg_100	0.38 minutes	30	54.37	72.5	79.58	60.02	0.72
1000 no cancer 500 cancer (unbalanced)	CNN	CNNU1000	2.31minutes	30	99	66.7	2.47	206.69	0.723
	ResNet50	resu1000	6.46 minutes	30	77.71	67.97	48.25	65.43	0.799
	Vgg16	vgg16_u1000	2.55minutes	30	80.58	85.33	44.06	31.82	0.822
1000 cancer 500 no cancer (unbalanced)	CNN	CNNU500	2.06minutes	30	98.92	68.33	5.35	147.91	0.695
	ResNet50	resu500	6.68 minutes	30	77.89	76.95	47.5	61.83	0.835
	Vgg16	vgg16_u500	2.47minutes	30	79.91	83.99	42.39	33.67	0.82

Figure 7.1: Observations of different models

7.0.1 Time v/s Performance

ResNet50

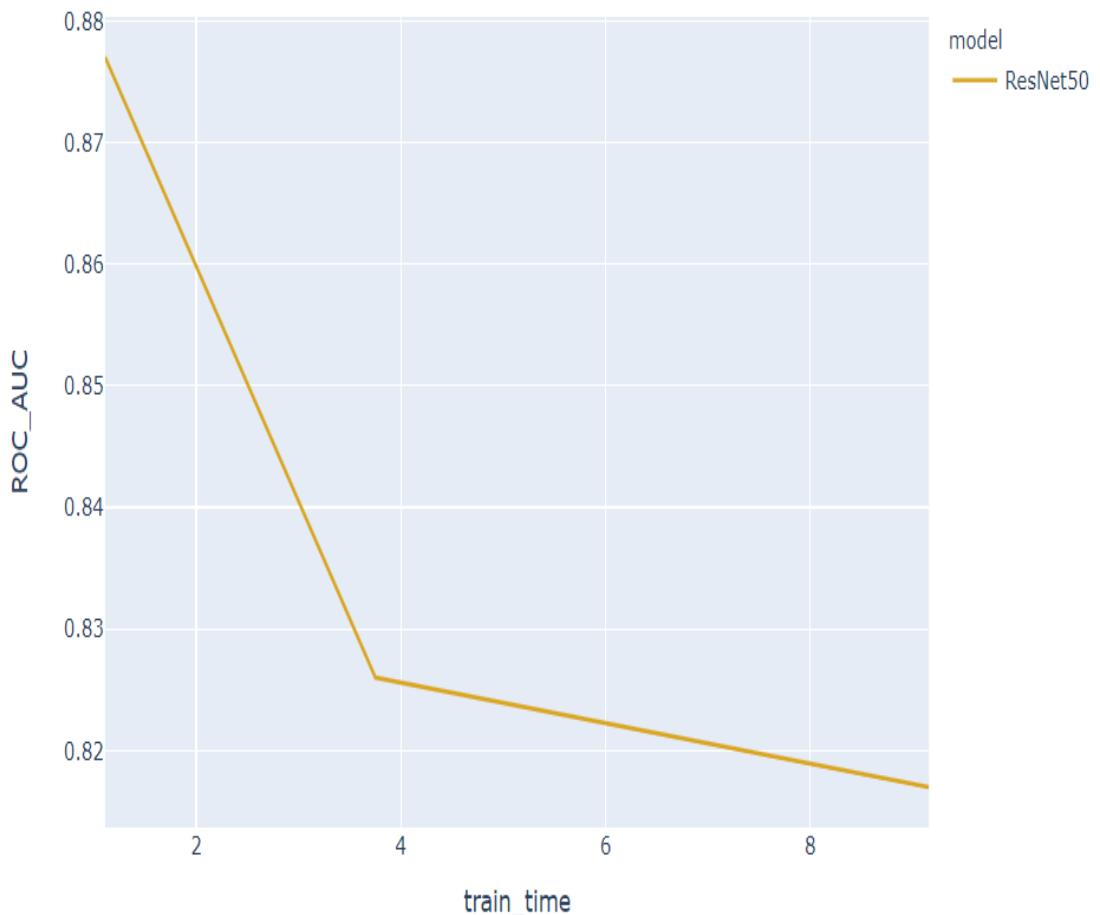


Figure 7.2: Resnet50 time v/s performance

VGG 16

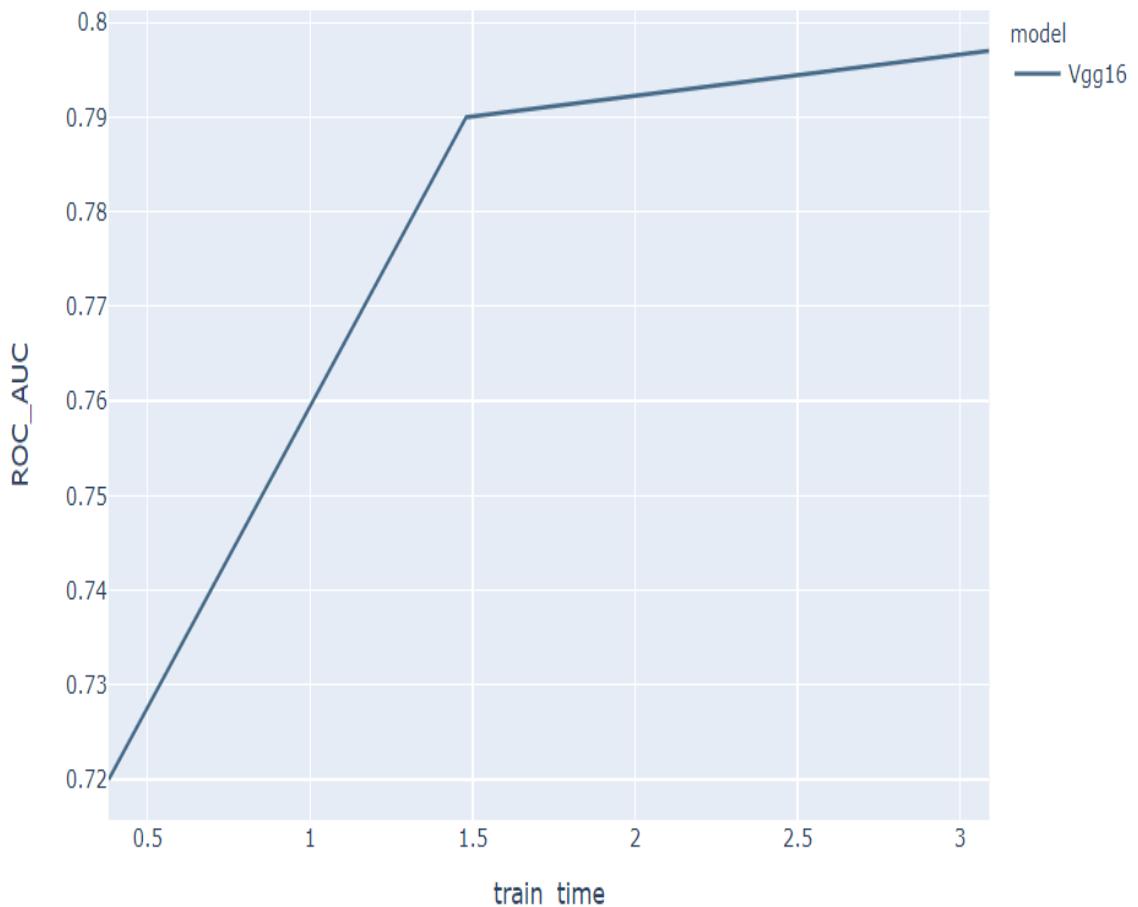


Figure 7.3: VGG16 time v/s performance

CNN

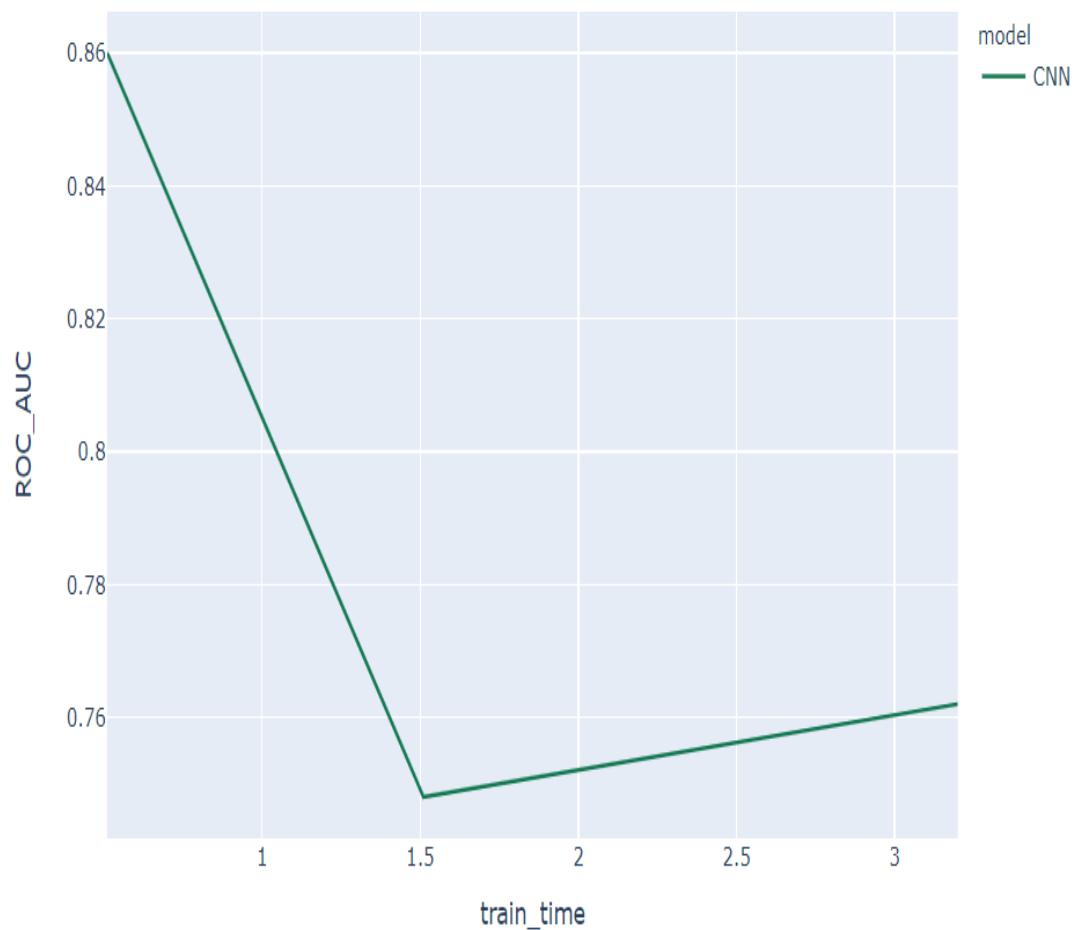


Figure 7.4: CNN time v/s performance

7.0.2 Performance v/s Data Size

ResNet 50

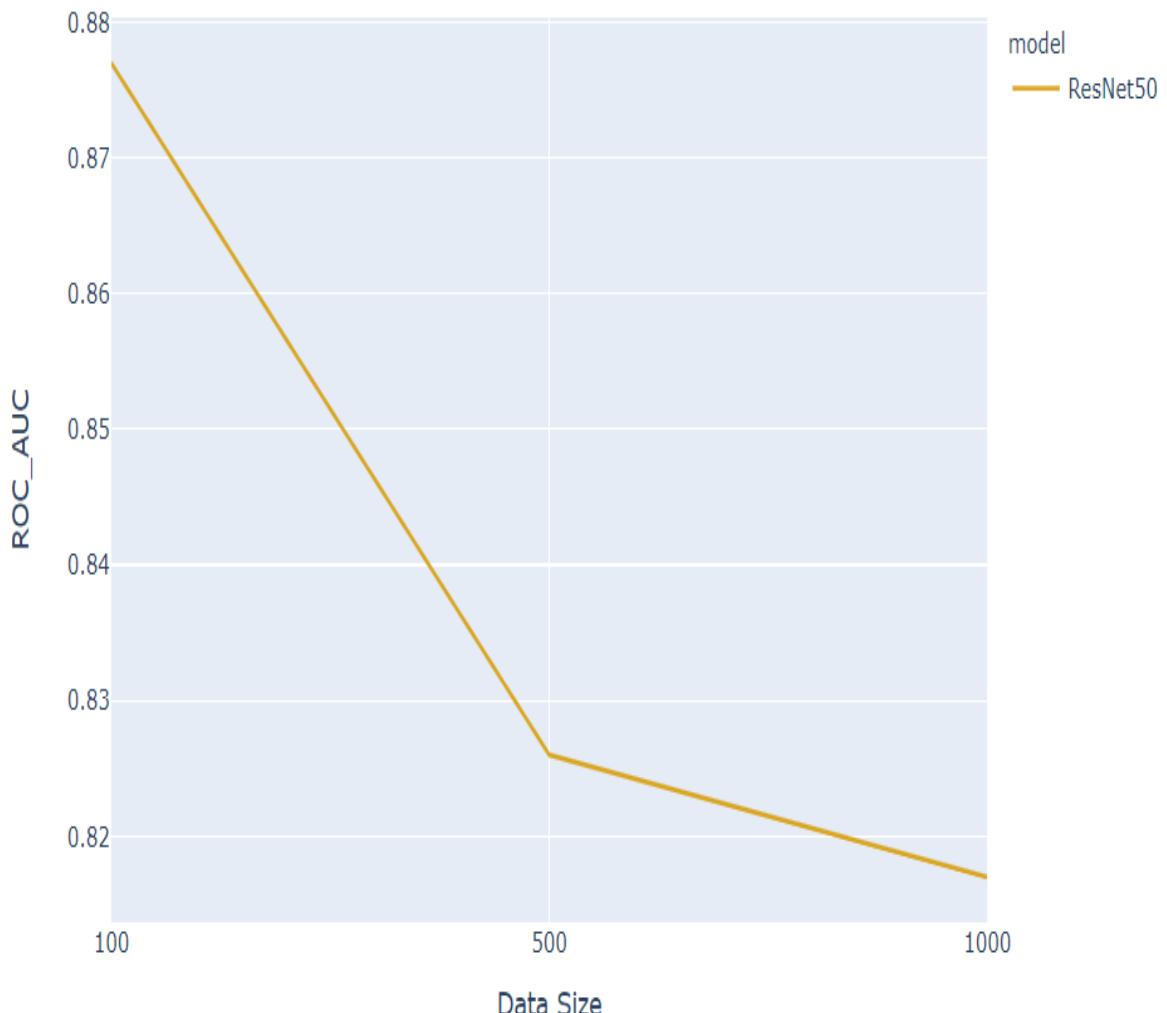


Figure 7.5: ResNet50 performance v/s Data size

VGG 16

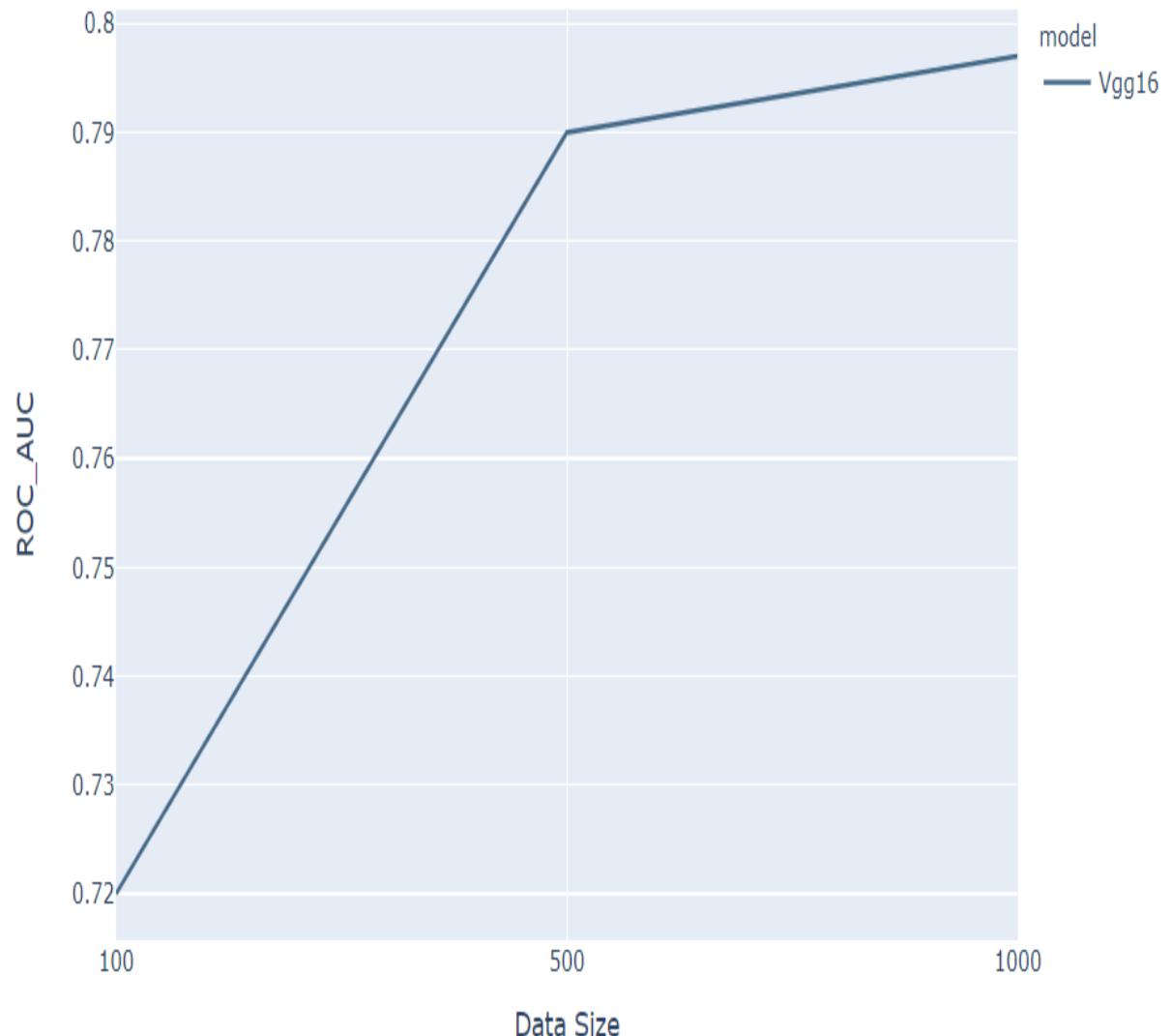


Figure 7.6: Vgg16 performance v/s Data size

CNN

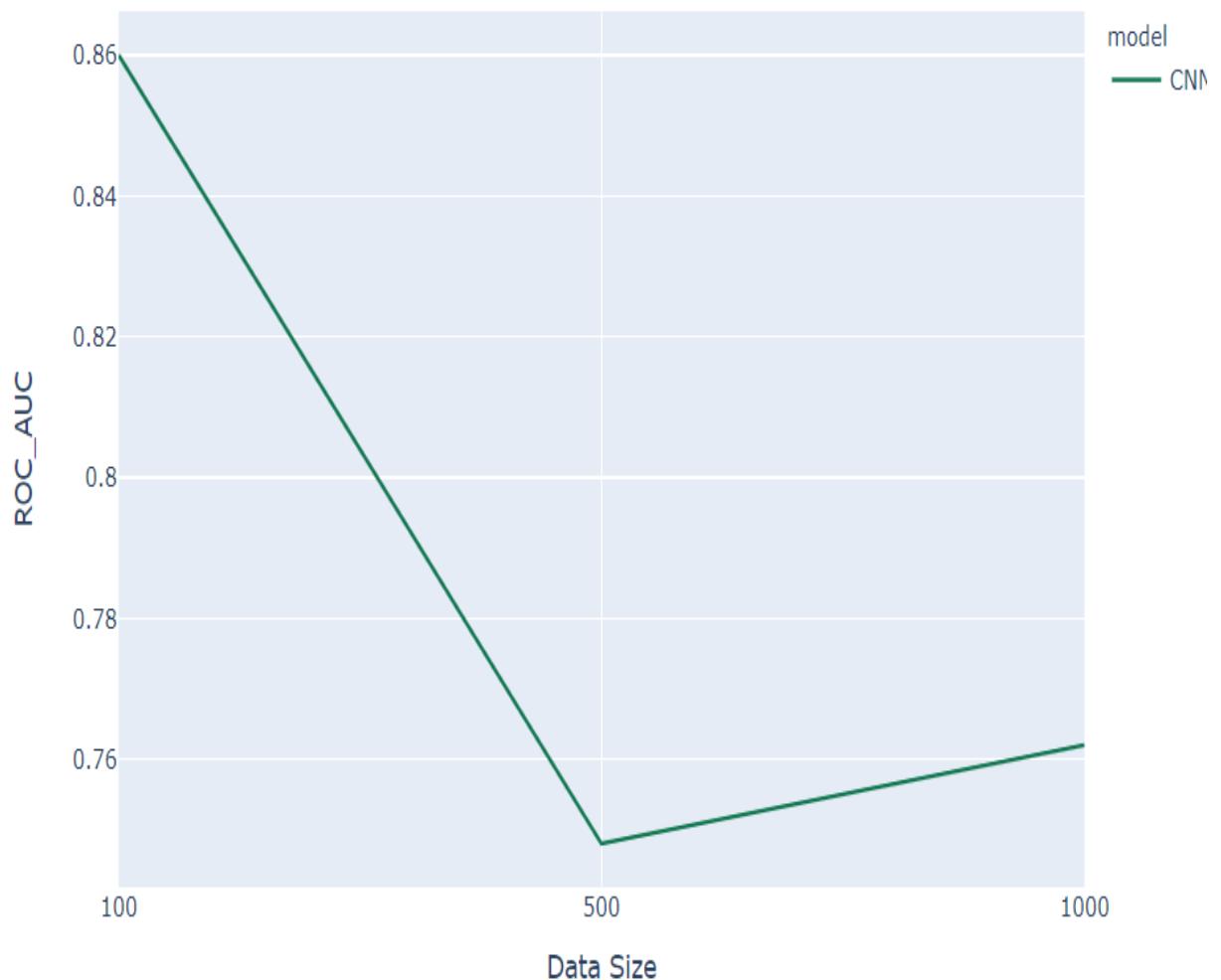


Figure 7.7: CNN performance v/s Data size

7.0.3 Time v/s Data Size

ResNet50

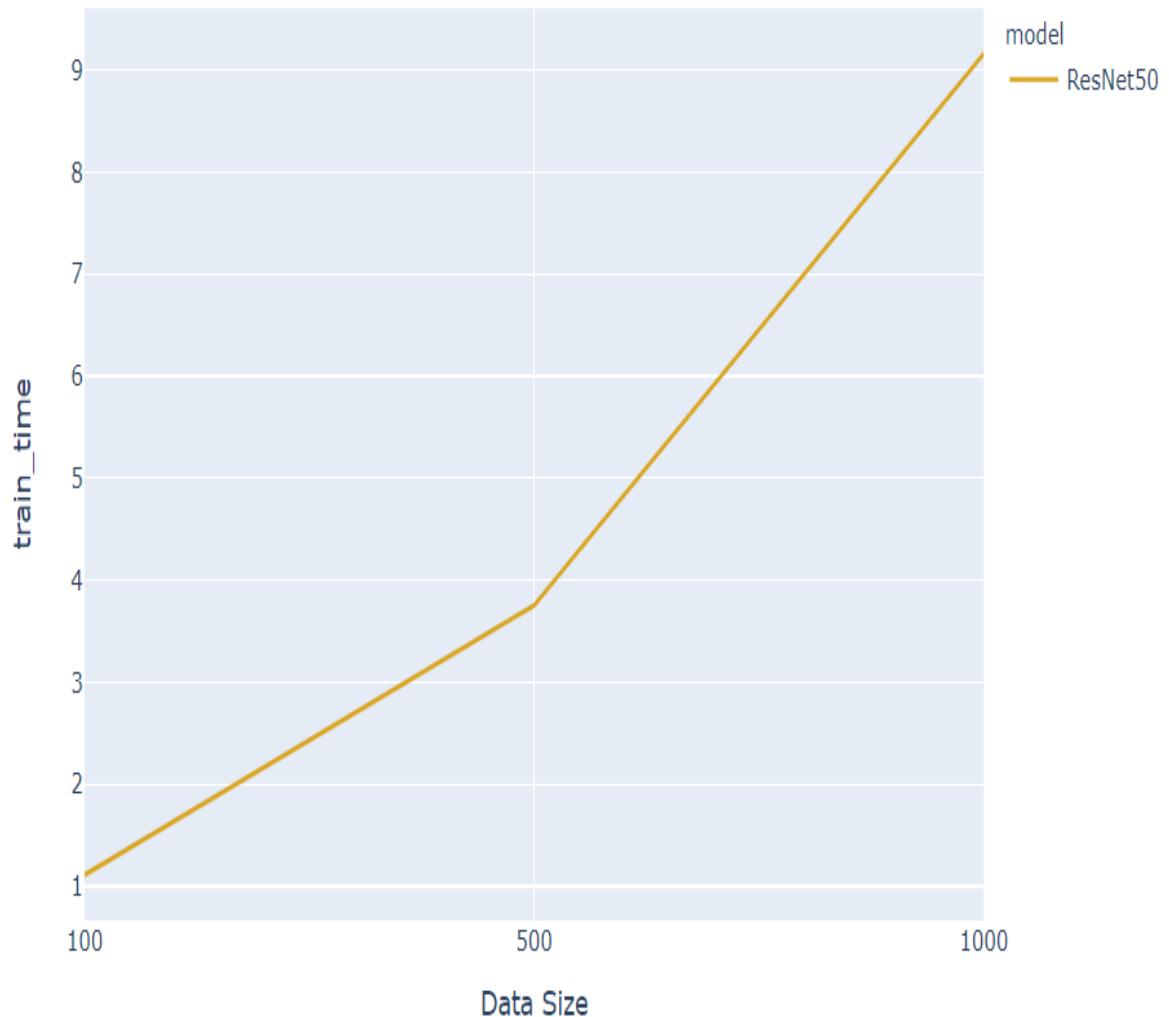


Figure 7.8: ResNet50 Time v/s Datasize

VGG 16

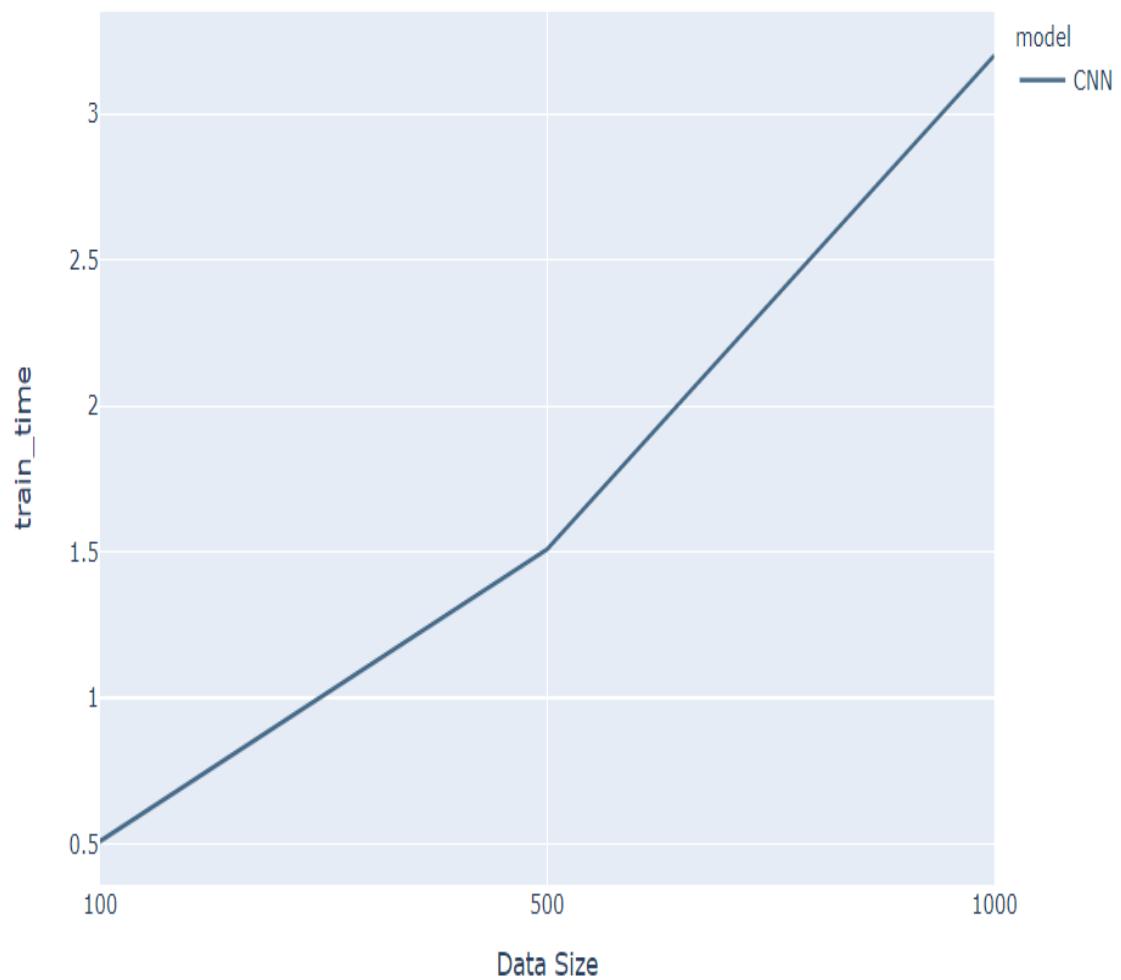


Figure 7.9: VGG16 Time v/s Datasize

CNN

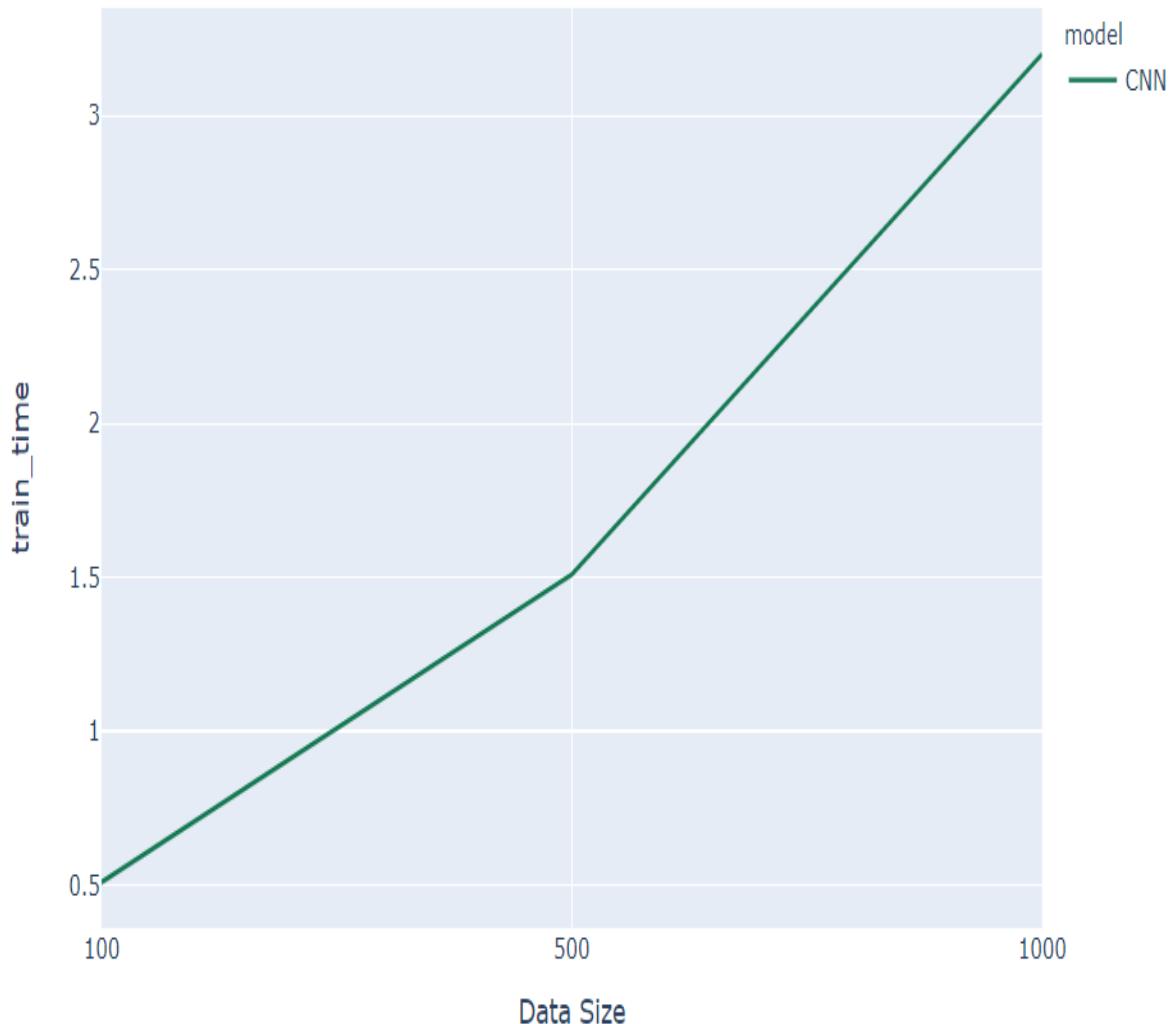


Figure 7.10: CNN Time v/s Datasize

7.0.4 Performance v/s Data Composition — Class Imbalance Effect

ResNet 50

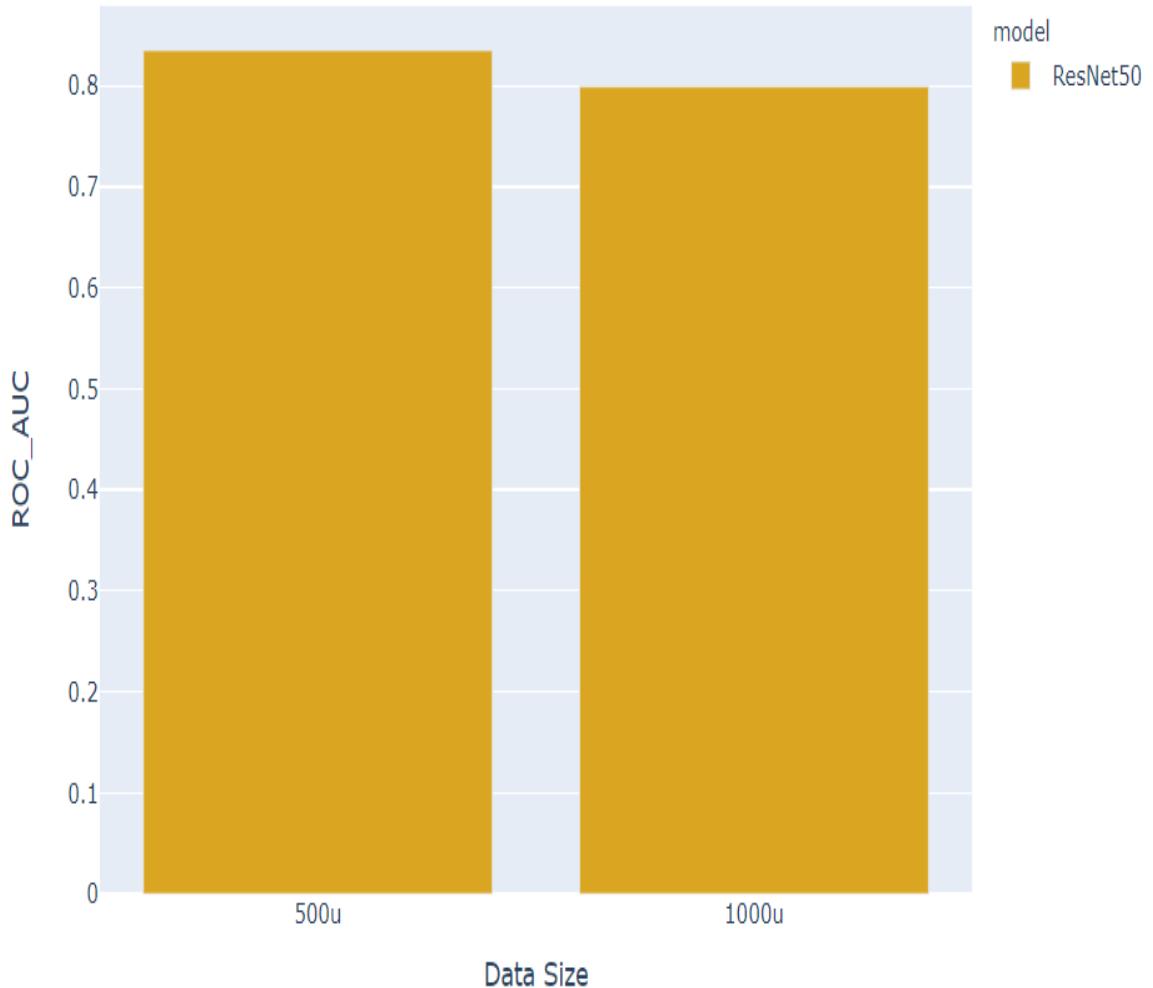


Figure 7.11: Resnet50 performance v/s Data composition

VGG 16

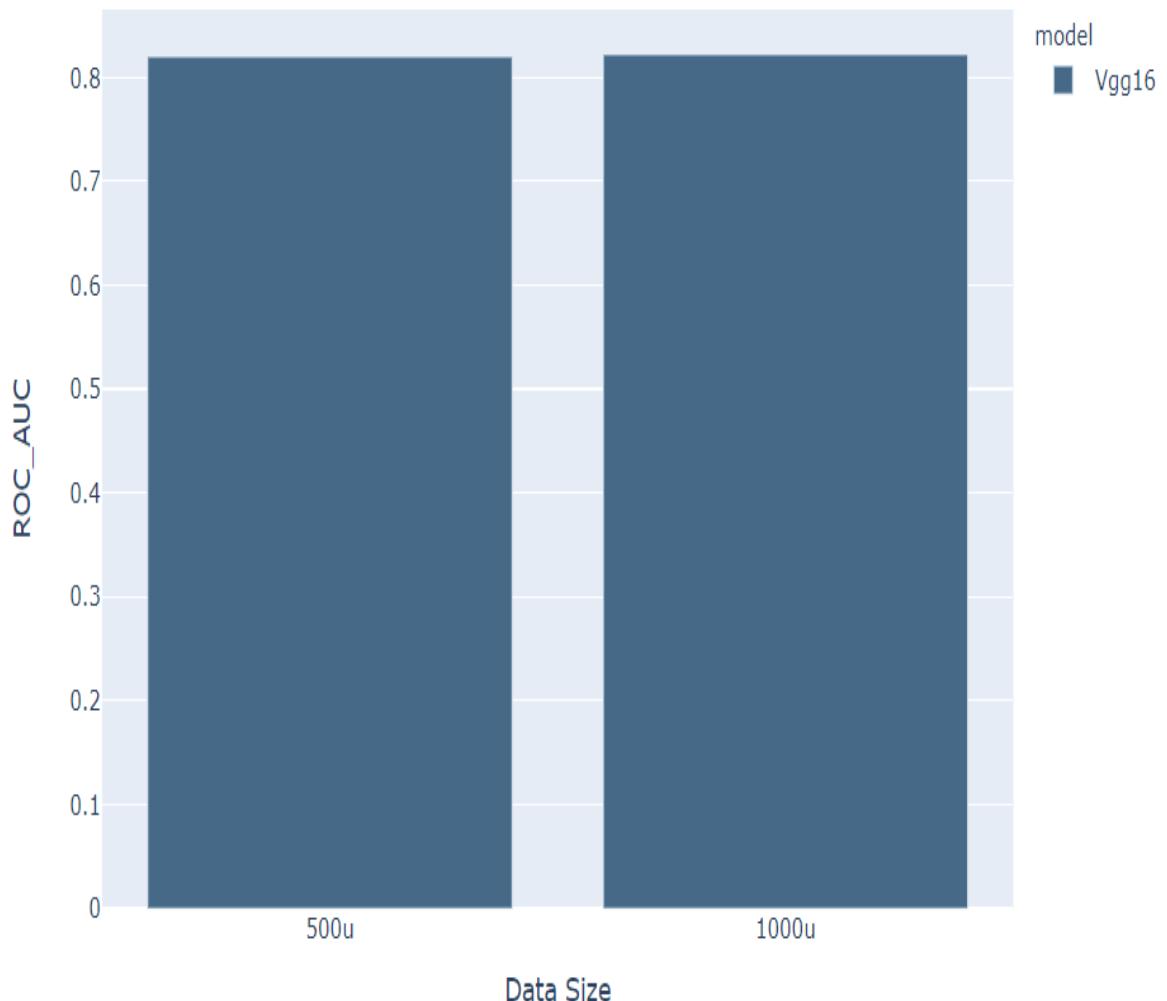


Figure 7.12: VGG16 performance v/s Data composition

CNN

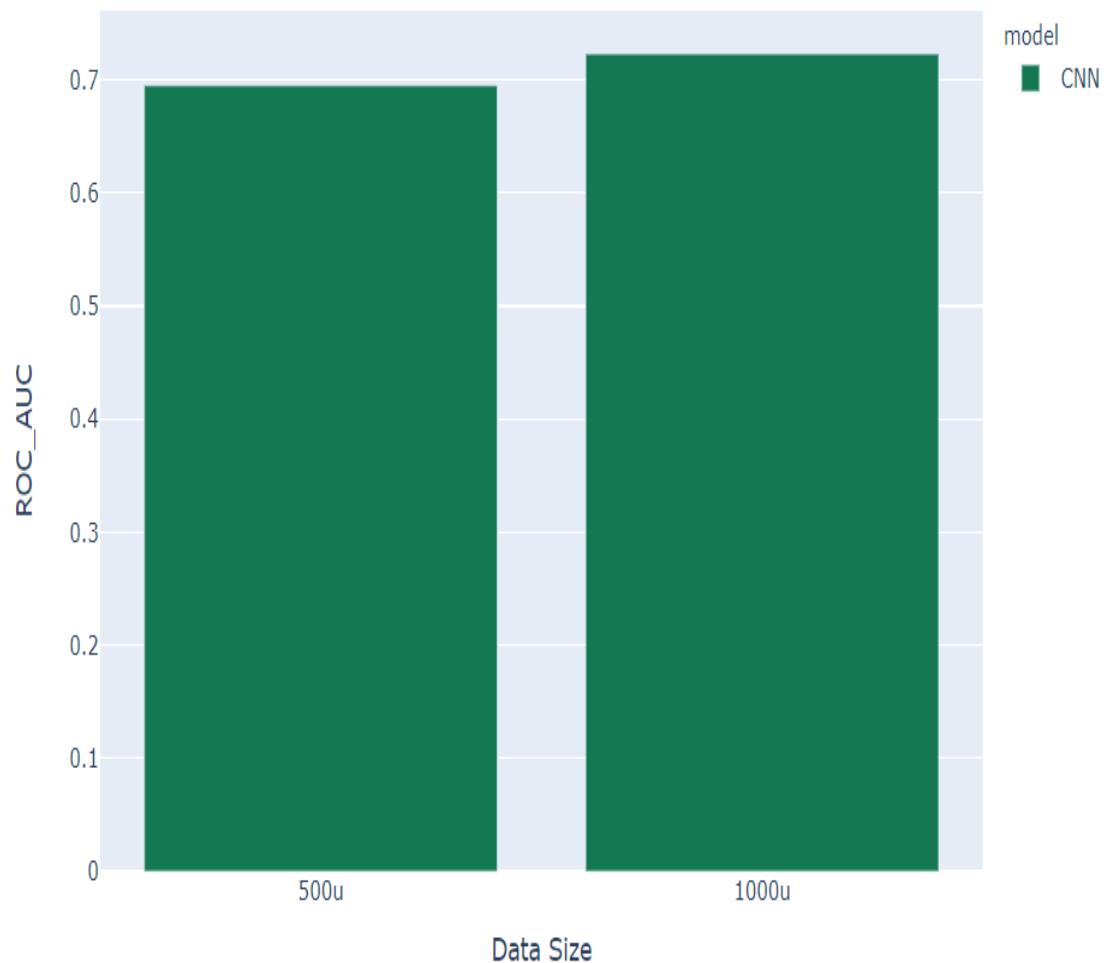


Figure 7.13: CNN performance v/s Data composition

7.0.5 Model Comparison

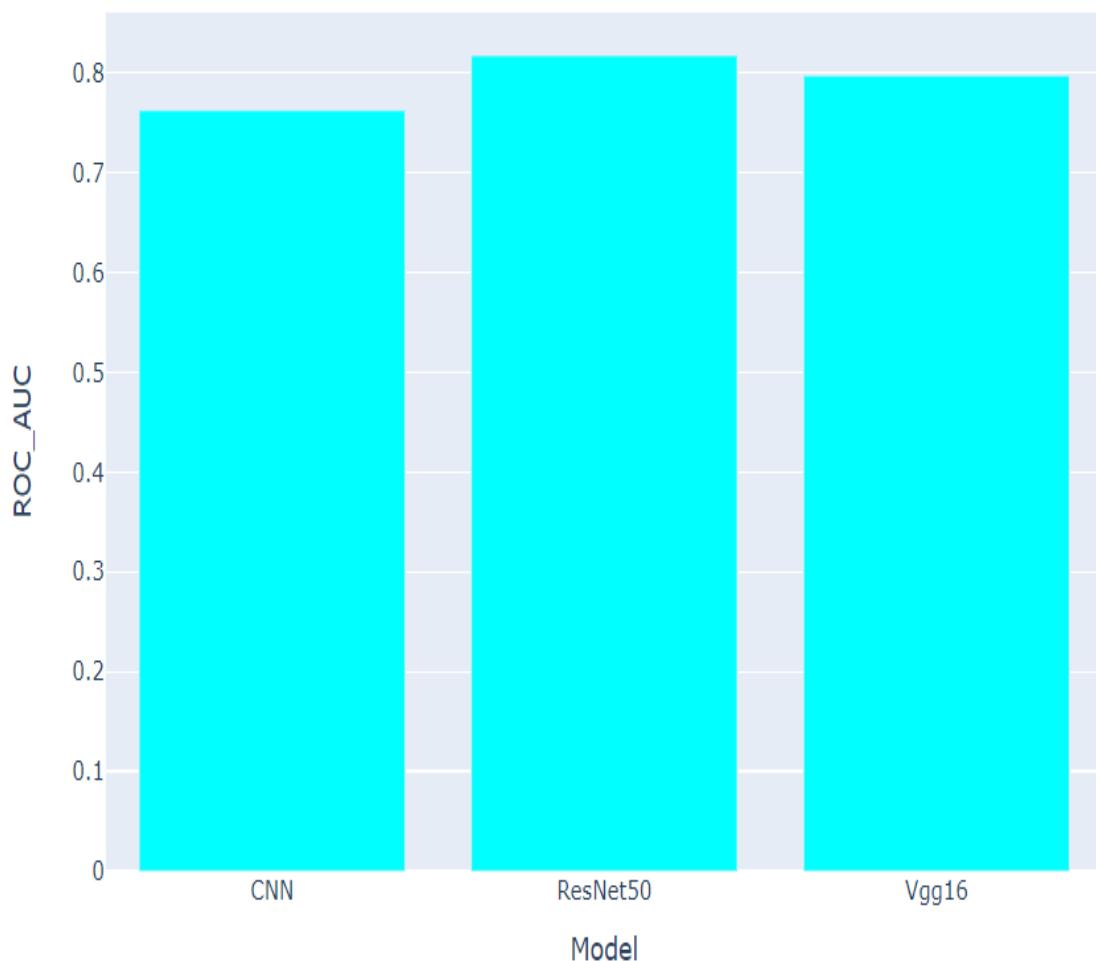


Figure 7.14: Comparison between different models for 1000 each dataset.

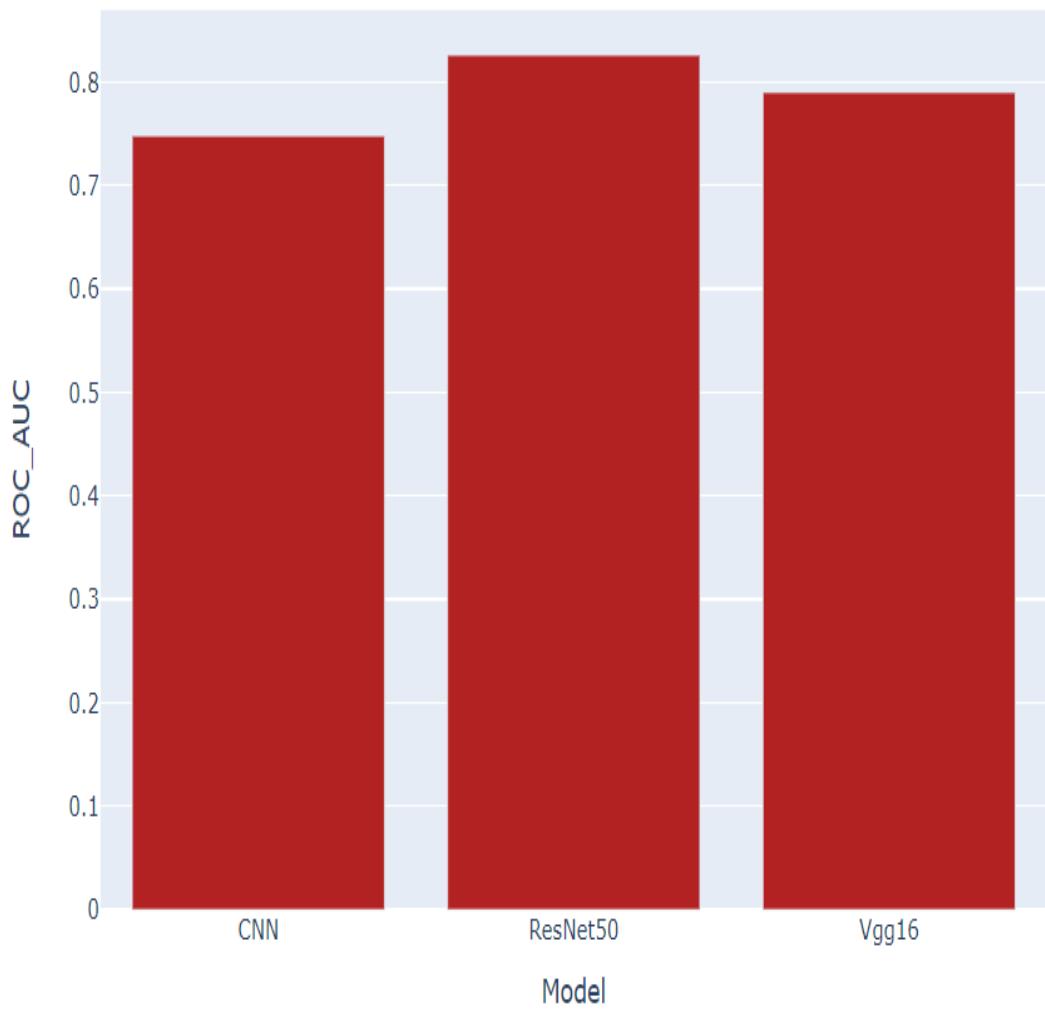


Figure 7.15: Comparison between different models for 500 each dataset.

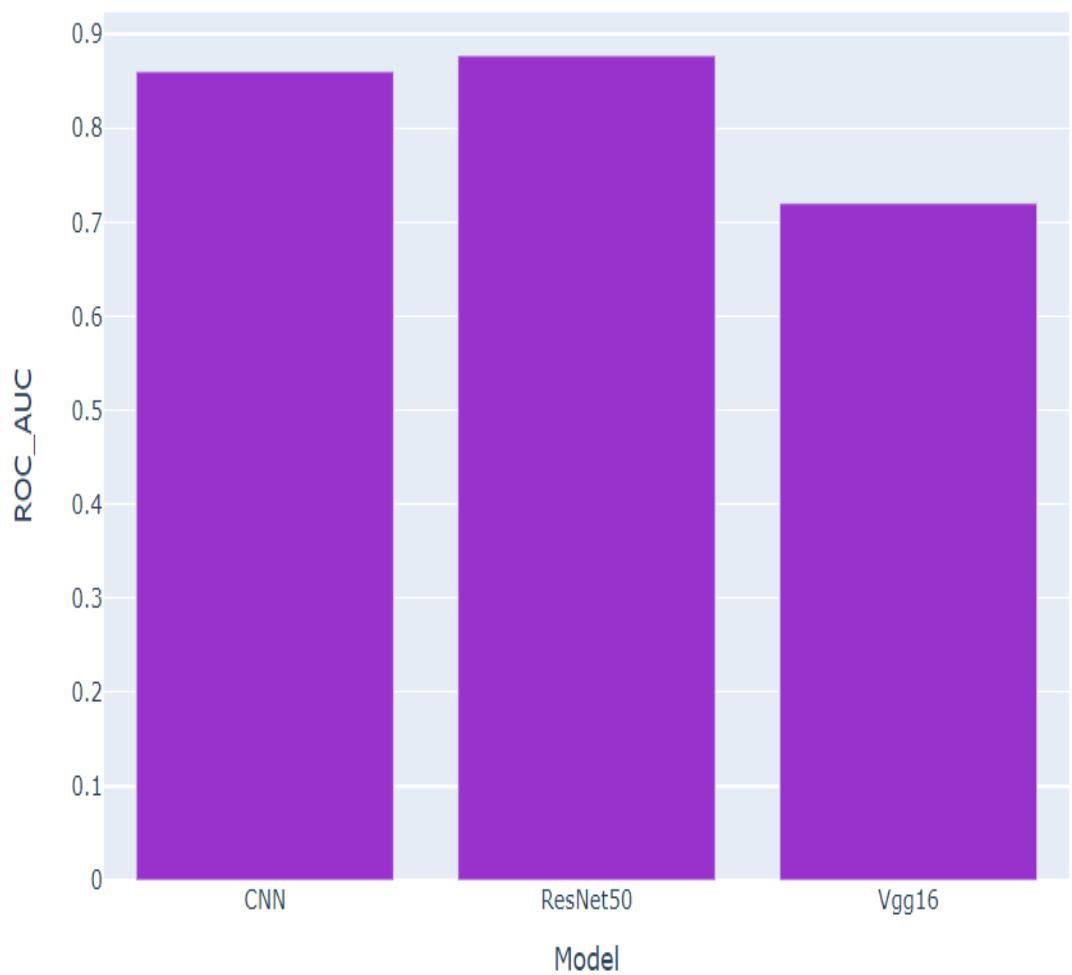


Figure 7.16: Comparison between different models for 100 each dataset.

Chapter 8

Result

This paper looks at the performance of models - CNN, Vgg16 and ResNet50 - used for classification of images. It also attempts to predict the presence of Cancer cells using the help of histopathologic images.

To measure the performance we used the metric of AUC-ROC. AUC (Area Under Curve) ROC (Receiver Operating Characteristics) curve is used to check or visualize the performance of the multi - class classification problem. It tells how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between patients with disease and no disease.

This paper answered the various research questions by observing various subsets of the original data set. To understand the effect of training time on performance we looked at various balanced subsets - 1000 , 500 and 100. It was concluded that for ResNet50 model the performance decreased as the training time increased, for Vgg16 model the performance increased as training time increase and for CNN model the maximum dip in performance occurred at 1.51 minutes of training time.

Next, we looked at the effect of Data size on the performance of the models. For ResNet50, as the size increased the performance decreased, for Vgg16 as the data size increased the performance increased and for CNN performance dipped maximum at data size 500.

The class imbalance effect was very much evident. It affected the ResNet50 and custom CNN model. The performance was much better when the data set had 500 images of class 0 (no cancer cells) and 1000 images of class 1 (yes cancer cells) for ResNet50. For CNN, it was other way round.

In conclusion, the model that performed the best in this classification task was ResNet50. Its ROC-AUC score was highest in all the three sub sets of the data set.