An improved accuracy Melanoma Detection Software using machine learning

Project Report

Table of Contents

Information on the data

The dataset to train and test the model has been taken from the **ISIC** (International Skin Image Collaboration) Archive. It consists of 1800 pictures of benign moles and 1497 pictures of malignant classified moles. The pictures have all been resized to low resolution (224x224x3) RGB.

It has **2 different classes** of skin cancer which are listed below:
1. Benign
2. Malignant

We try to detect 2 different classes of moles using **Convolution Neural Networks** with Keras Library with TensorFlow in the backend and then analyze the result to see how the model can be useful in practical scenarios.

Training - 80% of the data -- 2637 images
Testing - 20% of the data -- 660 images

Statistical measures employed

True positive: tp
False negative: fn
False positive: fp
True negative: tn

Accuracy = (tp + tn)/ (tp +tn + fp + fn)
Precision = tp/ (tp + fp)
Recall or sensitivity = tp/(tp + fn)

F1 score = 2* (precision * recall) / (precision + recall)

Specificity = tn/ (tn + fp)

Confusion matrix:

True Positive	False Positive
False Negative	True Negative

In many tests, including diagnostic medical tests, sensitivity is the extent to which true positives are not overlooked, thus false negatives are few, and specificity is the extent to which true negatives are classified as such, thus false positives are few. A sensitive test rarely overlooks a true positive (for example, showing nothing wrong despite a problem existing); a specific test rarely registers a positive classification for anything that is not the target of testing (for example, finding one bacterial species and mistaking it for another closely related one that is the true target).

F1 Score:

The harmonic mean of precision and recall gives the F1 score which is a measure of performance of the model's classification ability.
F1 score is considered a better indicator of the classifier's performance than the regular accuracy measure.

Deciding Parameter Values

Deciding the number of epochs:
The model is trained as long as the training loss and validation loss decrease with each epoch. If they start to show a gradual yet sustained increase, the model is being overfitted and the number of epochs must be reduced.

Deciding the batch size:
The batch size is preferably a power of the number 2. e.g.: 8, 16, 32, 64
Usually, the higher the batch size, the better the outcome, but the dataset size also has a role to play.

Learning rate:
Learning rate has been kept as 1e-5 for every model tried or trained.

Normalization

All the image values are normalized by dividing the RGB values by 255.
Reason for normalization - Normalizing data by dividing it by 255 improves the performance of activation functions- e.g.: The sigmoid function works best when the data is in the range 0 to 1.0. It is easier for the neural network to find a local or a global minimum on a smaller range of 0-1 than 0-255.

First CNN Architecture

(9 layered sequential model)

The first layer is the convolutional (Conv2D) layer. It is like a set of learnable filters. 64 filters are set for the two conv2D layers. Each filter transforms a part of the image (defined by the kernel size) using the kernel filter. The kernel filter matrix is applied to the whole image. Filters can be said to represent a transformation of the image.

The CNN can isolate features that are useful everywhere from these transformed images.

The second important layer in the CNN is the pooling (MaxPool2D) layer. This layer acts as a down sampling filter. It looks at the two neighboring pixels and picks the maximal value. This is used to reduce the computational cost, and to some extent, also minimize overfitting. The pooling size (i.e., the area size pooled each time) has to be chosen. The more the pooling dimension, the more essential is down sampling.

By combining convolutional and pooling layers, CNN can combine local features and learn more global features of the image.

The next step is Dropout. Dropout is a regularization method, where a proportion of nodes in the layer are randomly ignored (setting their weights to zero) for each training sample. Dropout drops a proportion of the network randomly and forces the network to learn features in a distributed way. This technique also improves generalization and reduces overfitting.

The rectified linear activation function or **ReLU** is used which for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. The rectifier activation function is used to add non-linearity to the network.

The Flatten layer is the seventh layer and is used to convert the final feature maps into one single 1D vector. This flattening step is needed to use completely connected layers after some convolutional/max pool layers. It combines all the found local features of the previous convolutional layers.

In the end, the features are used in one fully-connected (Dense) layer, which is just an artificial neural networks (ANN) classifier.
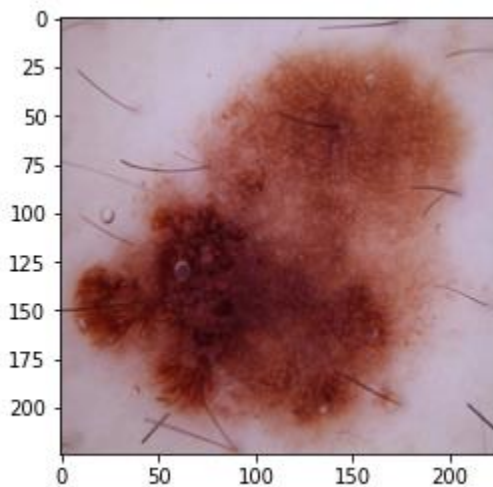
Summary:

No. of layers: 9

List of layers: ['conv2d_2', 'max_pooling2d_2', 'dropout_2', 'conv2d_3', 'max_pooling2d_3', 'dropout_3', 'flatten_1', 'dense_2', 'dense_3']
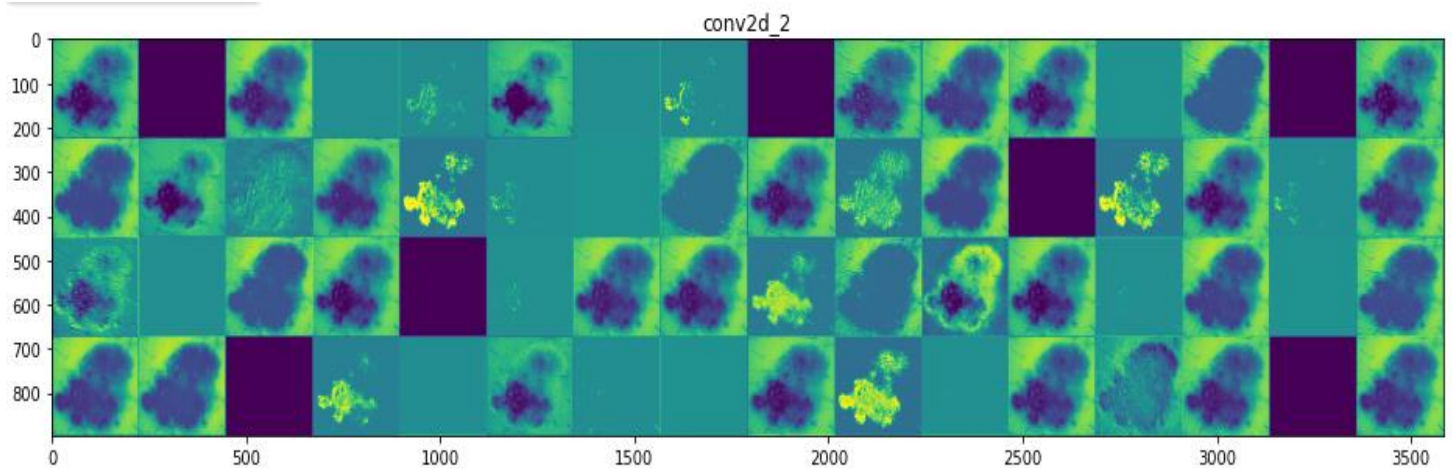
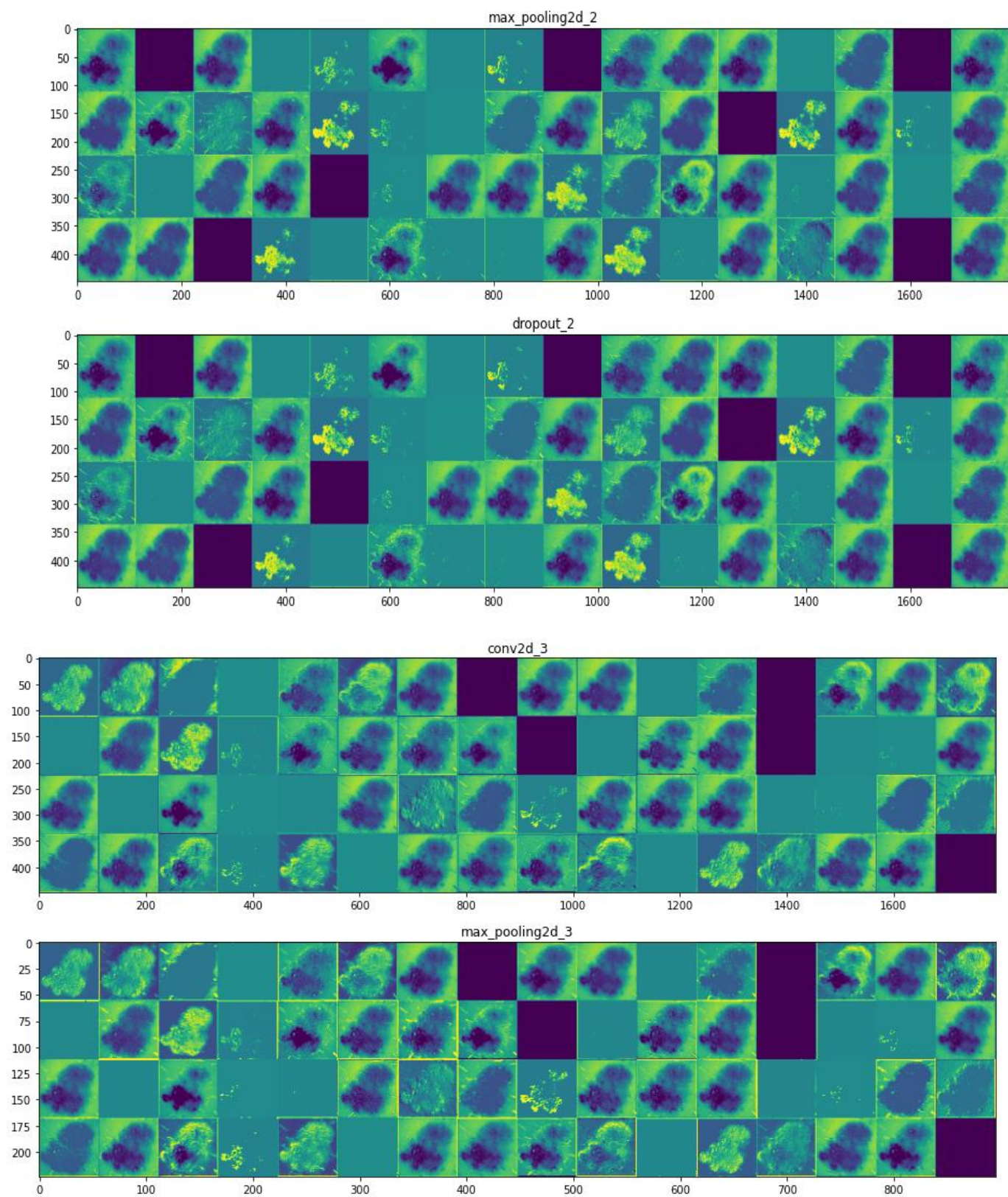**Visual Analysis of Feature Extraction:**
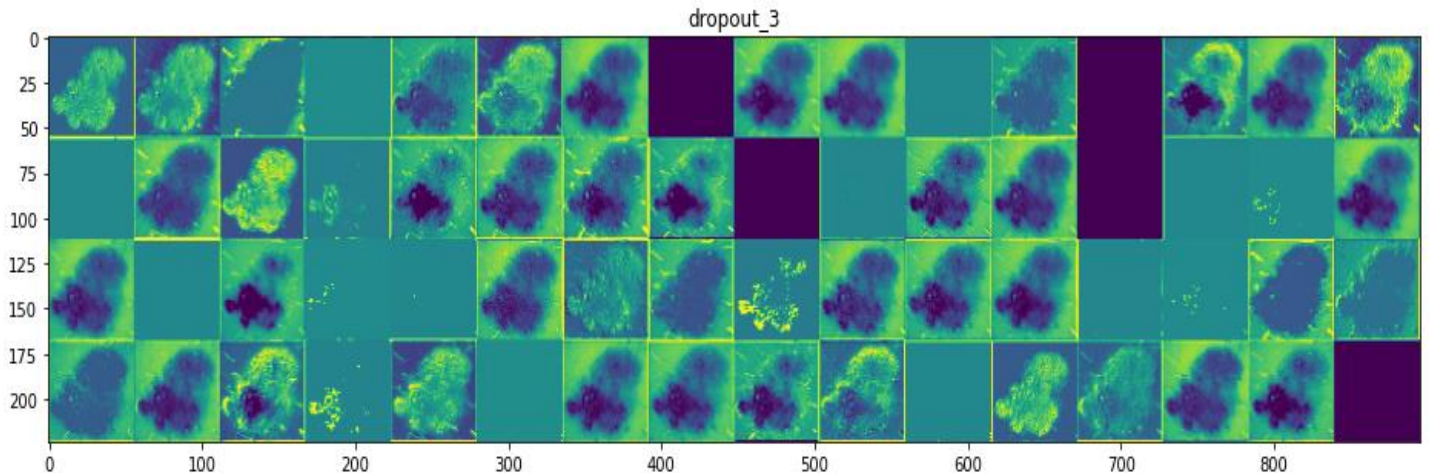
Image to be analyzed:

Class: Malignant

Analyzing feature maps for each layer:

max_pooling2d_2

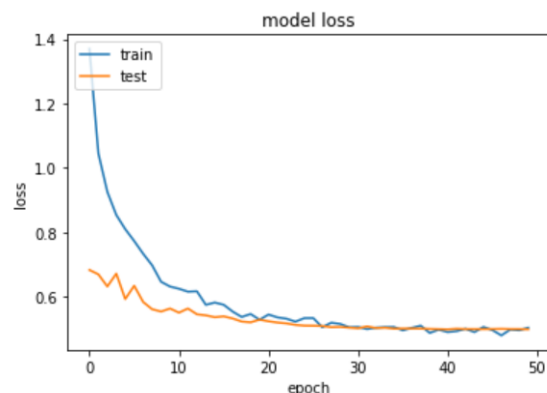dropout_2
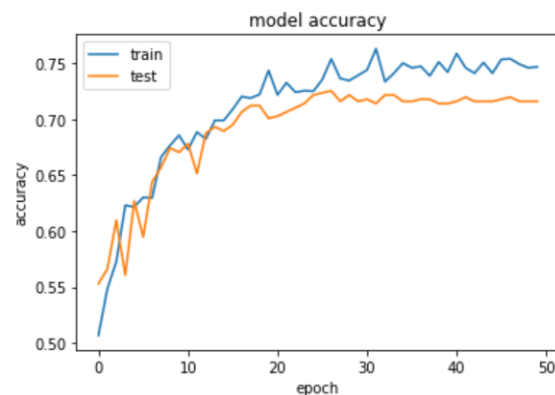

conv2d_3

max_pooling2d_3

8

dropout_3

Interpretation:

- The first layer retains the complete shape of the lesion although there are several filters that are not activated and are left blank. At this stage, the activations retain nearly all the information possessed by the initial picture.
- As the layers get deeper, the activations become increasingly abstract and less visually interpretable. The deeper layers provide more information related to the class of the image by encoding higher-level concepts such as single borders, corners and angles.

Accuracy and loss during training:

Cross-Validation:

9

```
Model: "sequential"
_____
Layer (type)                Output Shape              Param #
=================================================================
conv2d (Conv2D)             (None, 224, 224, 64)      1792
_____
max_pooling2d (MaxPooling2D) (None, 112, 112, 64)     0
_____
dropout (Dropout)           (None, 112, 112, 64)      0
_____
conv2d_1 (Conv2D)           (None, 112, 112, 64)      36928
_____
max_pooling2d_1 (MaxPooling2 (None, 56, 56, 64)       0
_____
dropout_1 (Dropout)         (None, 56, 56, 64)        0
_____
flatten (Flatten)           (None, 200704)            0
_____
dense (Dense)               (None, 128)               25690240
_____
dense_1 (Dense)             (None, 2)                 258
=================================================================
Total params: 25,729,218
Trainable params: 25,729,218
Non-trainable params: 0
_____
accuracy: 75.77%
75.77% (+/- 0.00%)
```

Testing accuracy: 78.485%

precision score: 0.9428571428571428

recall score: 0.22

f1 score: 0.3567567567567568

confusion matrix: [[356   4]

                                      [234 66]]

(with 660 test pictures)

===
The 9 layered sequential CNN is not a very sophisticated model; hence the most modern architectures are implemented.

These Include:
   1) ResNet 50
   2) Reset V2 101
   3) ResNet V2 152
   4) Xception

===

ResNet50

ResNet 50 is a well-known convolutional network that is used in classifying images. As its name suggests, it is a 50 deep layered network. It has 48 convolutional layers accompanied by 1 MaxPool and 1 Average pool layer. It is known to have the ability to classify images into 1000 different categories. This has further allowed it to learn detailed feature representations of images and work with efficiently classifying them.

Besides learning features of the images, ResNet 50 learns the residual aspects of images (the aspects other than features present in an image) through the process of layer subtraction. In addition to the uses of image classification, the ResNet 50 network can be used for object localization and object detection.
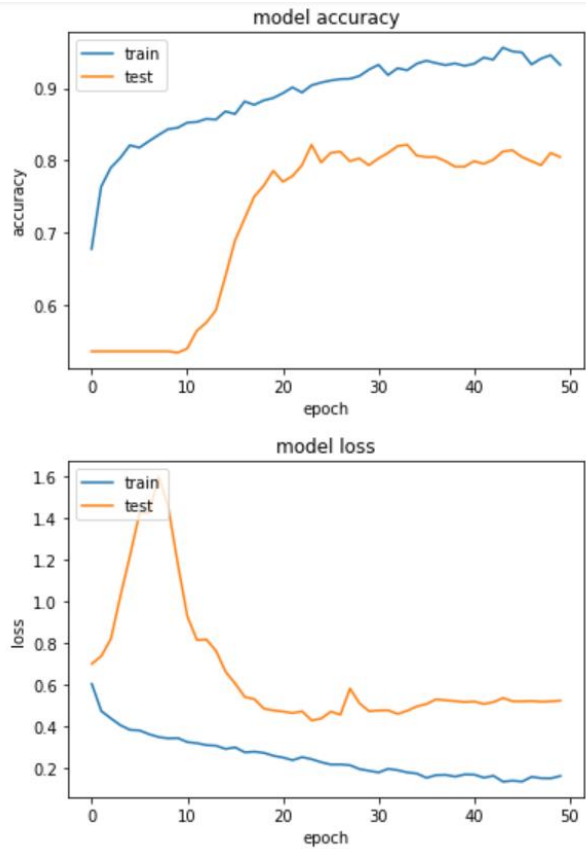
Epochs = 50
Batch size = 64
Learning rate = 1e-5

Accuracy and loss during training:

model accuracy

model loss

Testing Accuracy: 77.42 %
precision score:
0.8212765957446808
recall score: 0.6433333333333333
f1 score: 0.7214953271028037
confusion matrix: [[318 42]
                              [107 193]]

ResNet V2 101

ResNet V2 is a family of networks that develop based on the ResNet network architecture. The size of the input images for this network is required to be 224 x 224 pixels by default, while other inputs are possible with defined limits. The module uses the TF-Slim implementation of ResNet V2 101 with 101 deep network layers. The ResNet V2 101 convolutional network contains a trained instance of the ImageNet network.

This network additionally focuses on modifying the non-linearity as an identity mapping. This involves an additional operation between the identity mapping and the residual mapping. The difference between the ResNet V1 and ResNet V2 functionality is that ResNet V2 applies Batch Normalization and ReLU activation before the convolution process, while ResNet V1 does the ReLU activation and Batch Normalization after the convolution process.
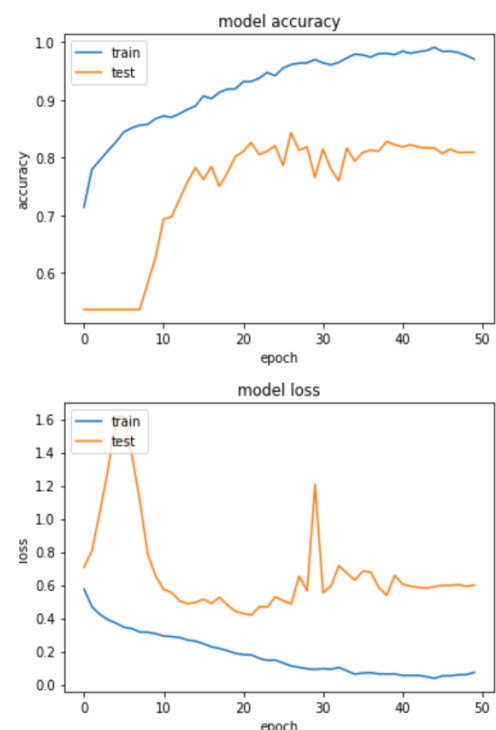
Learning rate = 1e-5

epochs = 50

batch size = 64

Accuracy and loss during training:

Testing Accuracy: 81.818%
precision score: 0.8061224489795918
recall score: 0.79
f1 score: 0.7979797979797979
confusion matrix: [[303 57]
                       [63 237]]

13

ResNet V2 152

ResNet V2 152 belongs to the ResNet V2 family of networks with 152 deep layers. It is a convolutional network that is used to improve inception and image classification. It is said to give higher accuracy rates in the learning of models for image classification due to its number of layers. Similar to the ResNet V2 101 convolutional neural network, this contains a trained instance of the ImageNet network.

As this network belongs to the ResNet V2 family of networks, it applies the Batch Normalization and ReLu activation before the convolution process, unlike the ResNet V1 family of networks. The ResNet V2 152 achieves a new state of the art in terms of accuracy on the ILSVRC image classification benchmark, further allowing significant simplification of the inception of blocks.

Learning rate = 1e-5
epochs = 50
batch size = 64

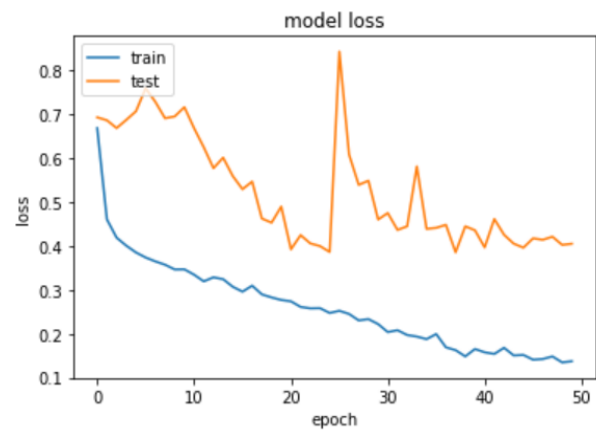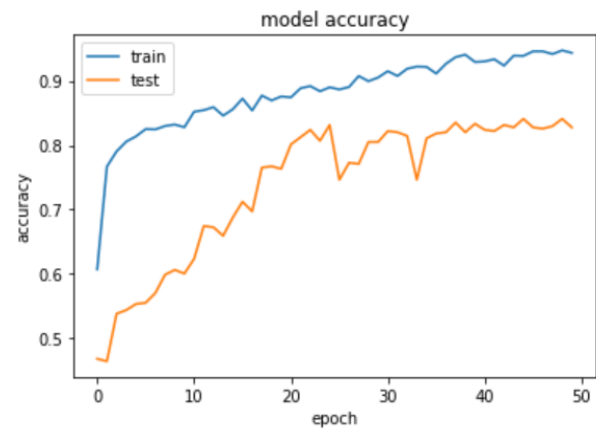**Accuracy and Loss during training:**

Testing Accuracy: 83%
precision score: 0.8241379310344827
recall score: 0.7966666666666666
f1 score: 0.8101694915254237
confusion matrix: [[309 51]
[61 239]]

Xception

Xception is a convolutional neural network consisting of 71 deep layers. More than one million images from the ImageNet Database can be trained through the pre-trained version. This network takes an image input size of 299 x 299. Xception is a network that is based on the Inception V3 network. It outperforms the Inception V3 network on the ImageNet Database and a larger image classification dataset comprising 350 million images.

Xception is a neural network architecture that relies solely on depth wise separable convolutional layers. Additionally, the Xception architecture consists of residual connections. In terms of the classification performance, Xception architecture shows a more significant improvement on the JFT Dataset compared to the ImageNet Dataset. Xception offers the highest top-1 and top-5 accuracy, having the values of 0.790 and 0.945, respectively.
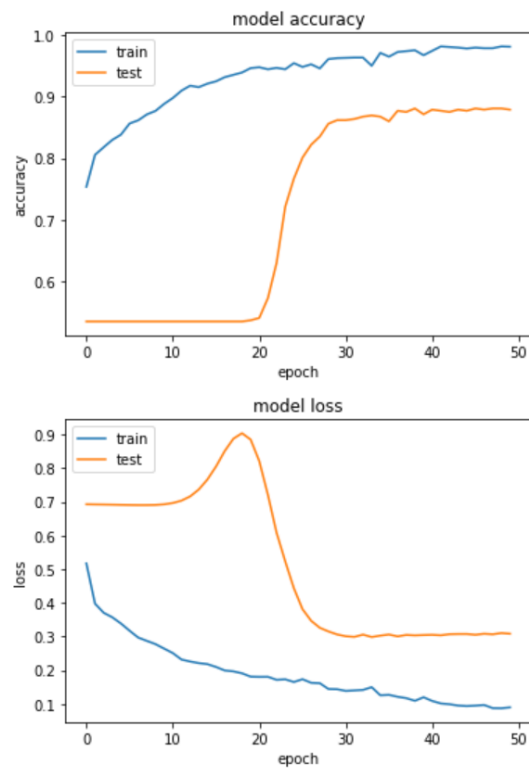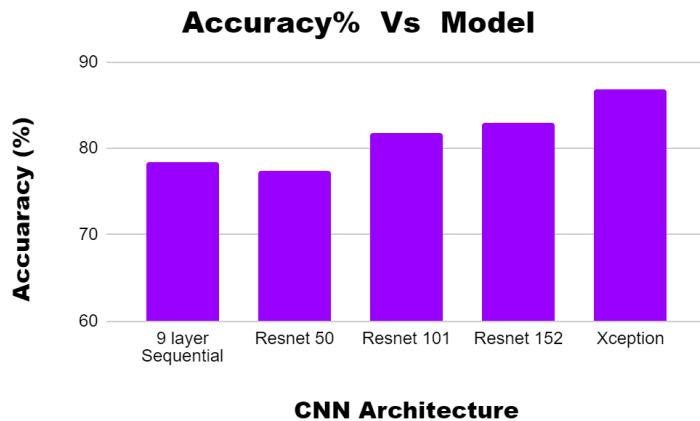
Learning rate = 1e-5
epochs = 50
batch size = 64

Accuracy and Loss During Training:

accuracy score: 87.1%
precision score:
0.8402555910543131
recall score:
0.8766666666666667
f1 score: 0.8580750407830342
confusion matrix: [[310 50]
   [37 263]]

Summary of results

Accuracy% Vs Model

The Xception architecture performs better than the ResNet models and the nine layered sequential. Although accuracy is the most widely used choice of measure, the F1 score serves as a better measure for medical image classification.

F1 Score Vs Architecture
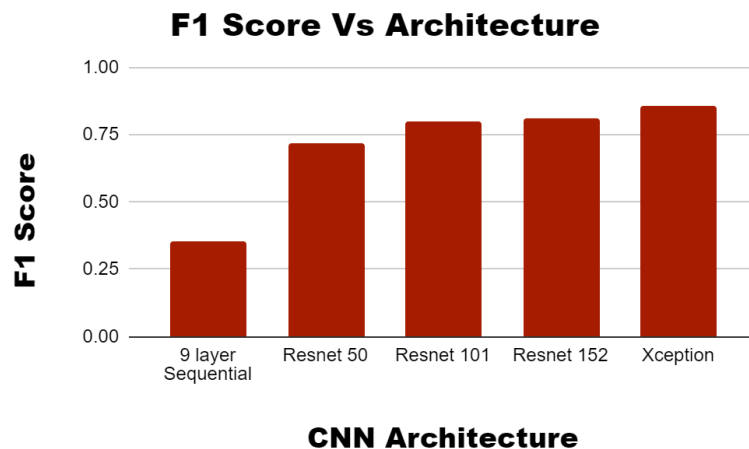
The F1 score shows that the ResNet models are comparable to Xception architecture. The ResNet architectures with more than 100 layers are better than the ResNet 50. However, the ResNet 152 shows no apparent increase from the ResNet 101 model. This depicts an increase in overfitting with CNN depth and a decrease in the generalization of the features.

Recall Vs Architecture
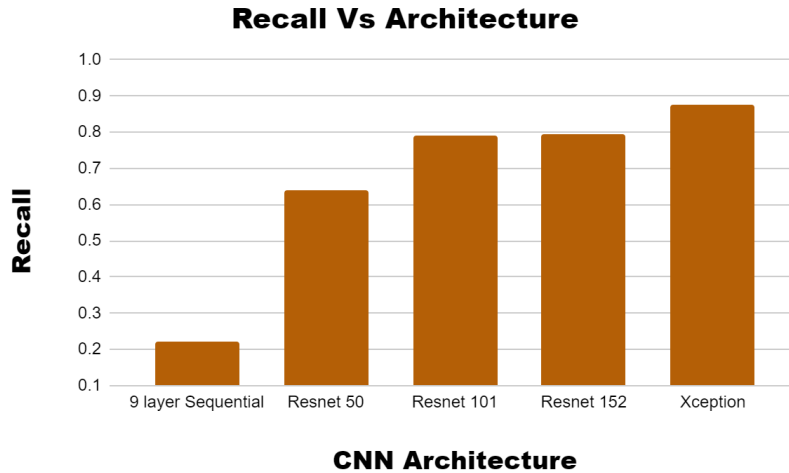
The recall measure is crucial when the correct classification of the positive class (here, malignant) is a priority. It gives the number of positive cases correctly identified to the total number of positive cases provided. The Xception model again outperforms the other architectures. The ResNet 152 does not perform better than ResNet 101 on account of overfitting.

Precision Vs Architecture
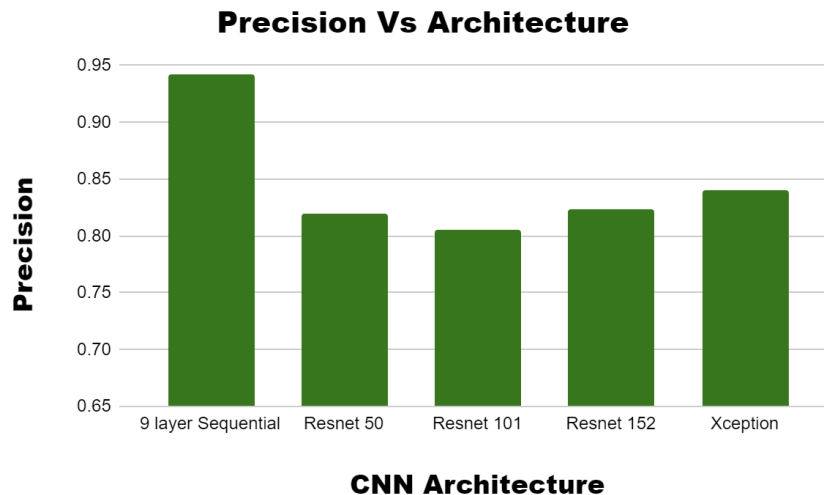
The nine layered sequential model offers an exceedingly high precision value. However, the precision is of lower importance than the recall because it is essential where the classification of the negative case is a priority. Xception provides a better precision score than other architectures.