# Skin Cancer Classification of Different models

Name: Abhishek Panda

CWID : 10478684

## 1. Introduction

The goal of the project is to identify whether a skin cancer is benign or malignant. There is one dataset in this project which is evenly distributed into benign and malignant in the data part and there are train and test data which is also stored in the same folder.

I have implemented 6 models in which the training is done on the train dataset and the validation is done with the test dataset.

The six models which I have implemented are defined below:

- Random Forest Classifier: Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time.
- Logistic regression: Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist.
- Decision tree Classifier: A decision tree is a flowchart-like structure in which each internal node represents a "test" on an attribute (e.g., whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.
- Naive Bayes: Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features.
- Resnet-50: A convolutional neural network with 50 layers is called ResNet-50. The ImageNet database contains a pretrained version of the network that has been trained on more than a million images. The pretrained network can categorize photos into 1000 different object categories, including several animals, a keyboard, a mouse, and a pencil.
- Very deep convolutional network [VGG16]: A convolutional neural network with 16 layers is called VGG-16. The ImageNet database contains a pretrained version of the network that has been trained on more than a million images. The pretrained network can categorize photos into 1000 different object categories, including several animals, a keyboard, a mouse, and a pencil.

Convolutional neural networks and transfer learning are two concepts used to solve the multi-classification challenge in the problem. Convolutional neural network (CNN) is more frequently referred to as "Convolutional Neural Network" and is a subset of deep learning techniques. To assess each pixel, convolution entails applying a kernel or filter of the next dimension to a chosen pixel and its

surrounds, shifting that kernel to the following pixel and its surroundings, and so on. CNN is primarily applied to photos to extract features. Although multilayer sequential neural networks may directly detect features, shapes, and patterns, CNN is more precise. Transfer Learning is an idea in machine learning where we leverage pretrained models to increase the model's accuracy.

## 2. Dataset

A balanced mixture of benign and malignant skin moles are represented in this dataset. The data is divided into two folders, each containing 1800 images (224x244) of the two different varieties of moles. The following link linked here to Kaggle page where the dataset was downloaded:

https://www.kaggle.com/datasets/fanconic/skin-cancer-malignant-vs-benign

I used Image.open() with convert("RGB") for the picture to be loaded in color. Labels were created for storing the tarin and test data as np array. I created the X_train and y_train, X_test and y_test by using the random sample.

Images were displayed with their actual classification. Once the data is distributed and plotted on the histogram, we can see that the train and test dataset is balanced.

## 3. Model Training

- Random Forest Classifier: In this I have kept the number of estimators to be 100. The model is fit on reshaped X_train as the size is larger and y_train. The predicted value is stored in the y_pred1 for using later.
    - The model tends to give an accuracy of 83.33% on the test dataset.
- Logistic Regression Classifier: L2 Regularization was used as penalty in this, the maximum number of iterations was set to be 200. Model was the fit and the prediction was stored in y_pred2.
    - The model tends to give an accuracy of 78.64% on the test dataset.
- Decision Tree Classifier: The classifier used entropy function to measure the quality of a split. The model ran with random state as zero.
    - The model tends to give an accuracy of 76.82% on the test dataset.
- Naive Bayes Classifier: The gaussian naïve bayes model was used with the default parameters. The model was fit on X_train and y_train. And the model prediction was stored in y_pred4.
    - The model tends to give an accuracy of 70.15% on the test dataset.
- Resnet-50: It is a sequential model. We add the ResNer50 layer without including the top and keeping the average max pooling. The weight distribution was taken from the default provided "ImageNet". The model was the added with flatten, dense and dropout layer.
    - Next the model was compiled with the RMSprop optimizers with the learning rate to be 0.0001, the loss was set to be Binary Cross entropy and metrics was set to accuracy.
    - A ReduceLROnPlateau was set on val_accuary to reduce the learning rate if the val_accuary doesn't change for few epochs.
    - The model was run on train_generator f 50 epochs and test_generator data was used as validation_data. Reduce_lr was added as callback.
    - The model tends to give an accuracy of 67.81% on the test dataset.

- Very deep convolutional network [VGG16]: There are 16 layers in this model by which all the images are passes and prediction are done based on that.
    - The trainable layers are set to false. Multiple layers are added to the Sequential layer after adding the vgg layer.
    - The model is complied with binary cross entropy loss and Adam optimizer with learning rate to be 1e-6.
    - The model was saved after each epoch, to use later.
    - The model was fit on train_generator with 50 epochs. And the accuracy was found to be 82.34%.

## 4. RESULTS

I will have shown few predictions of each classifier.

I. Random Forest classifier

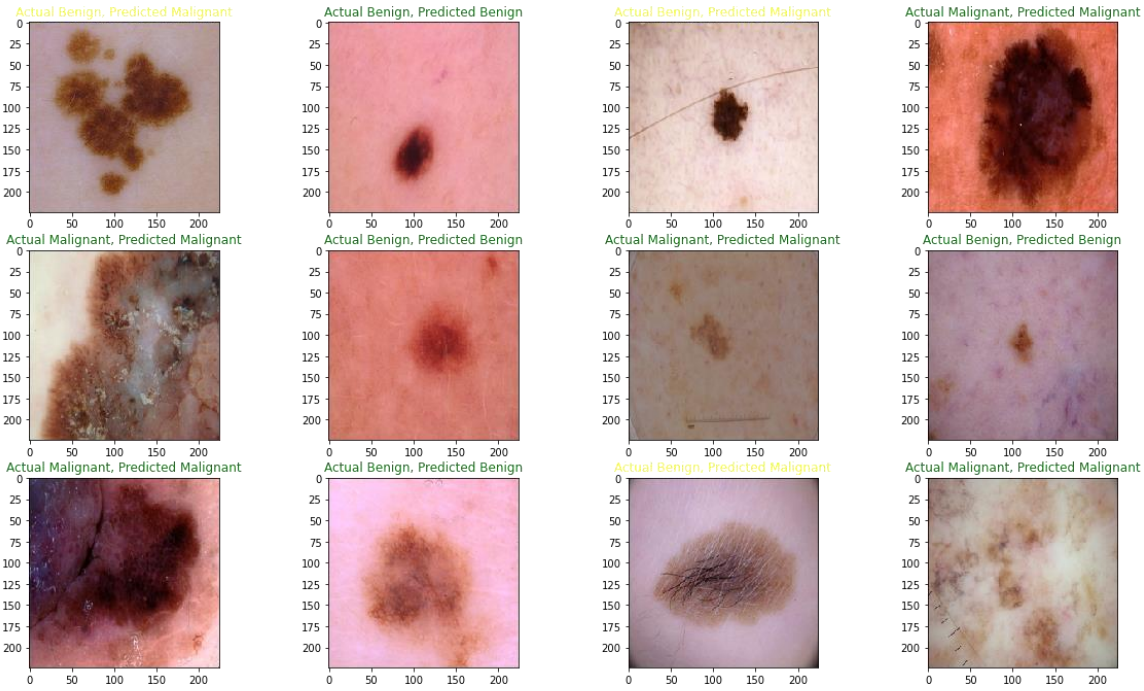


Fig 1. Random forest prediction

In Fig 1., the prediction done by random forest was mostly correct, with few wrong predictions.

## II. Logistic Regression

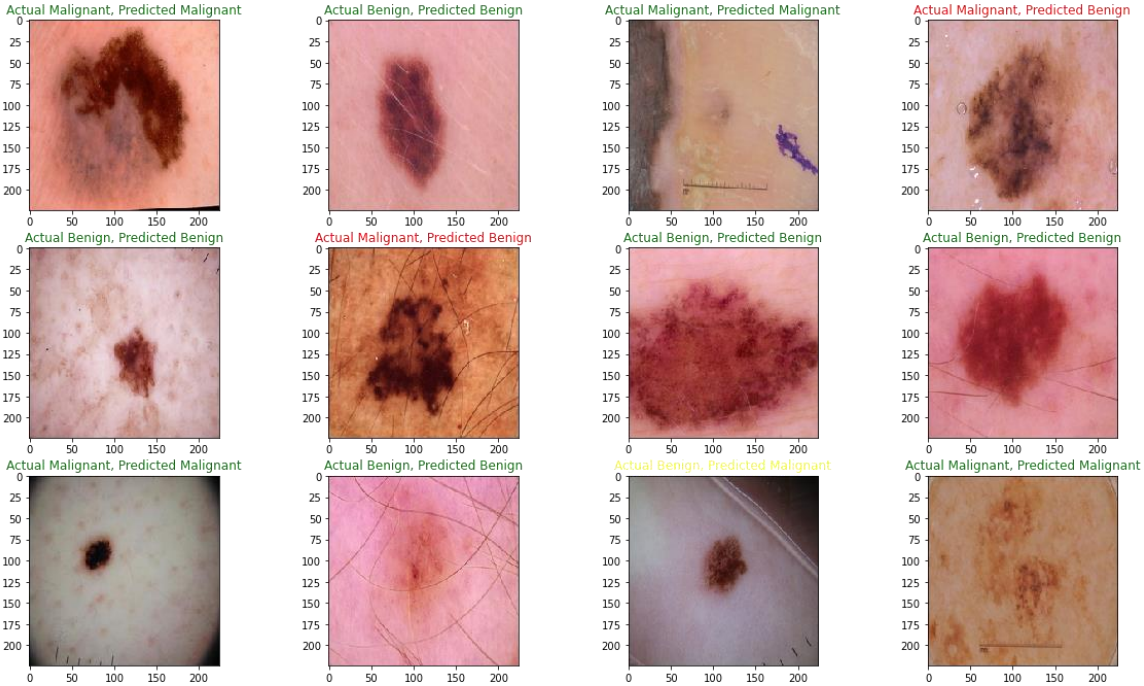


Fig 2. Logistic Regression prediction

In Fig 2., we can see few of them are wrong, but as per the accuracy the predictions which are done by the model are good.

## III. Decision Tree

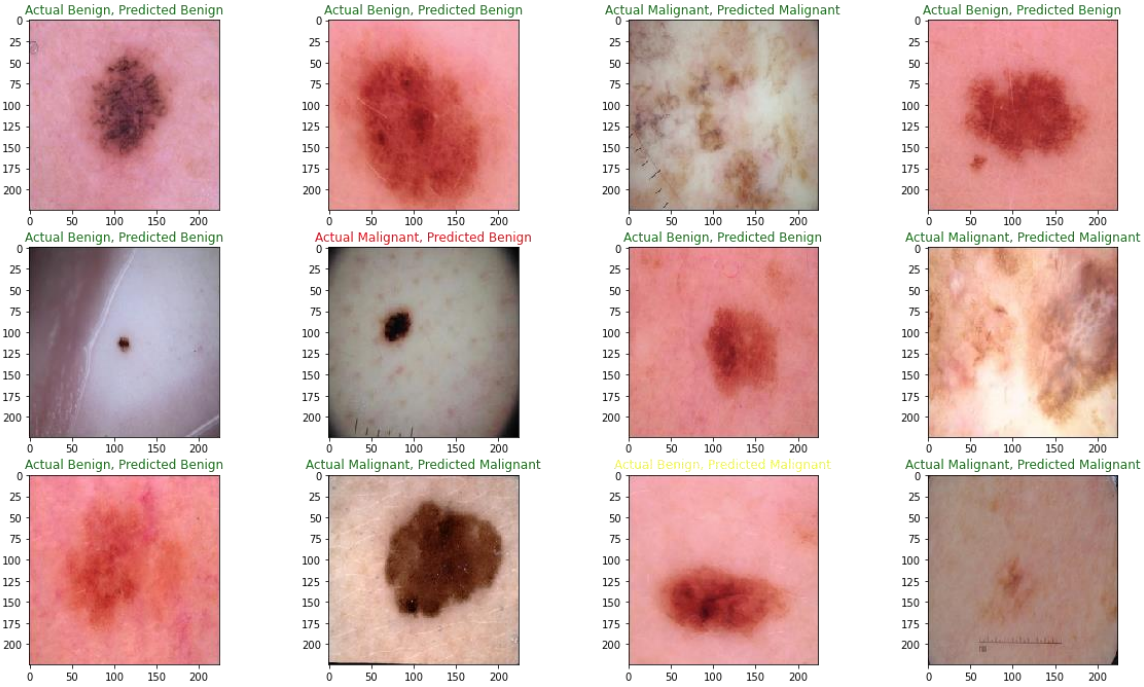


Fig 3. Decision Tree prediction

In Fig 3., the model was able to catch most of the sin cancer correctly but few of them were wrongly classified.

## IV. Naive Bayes

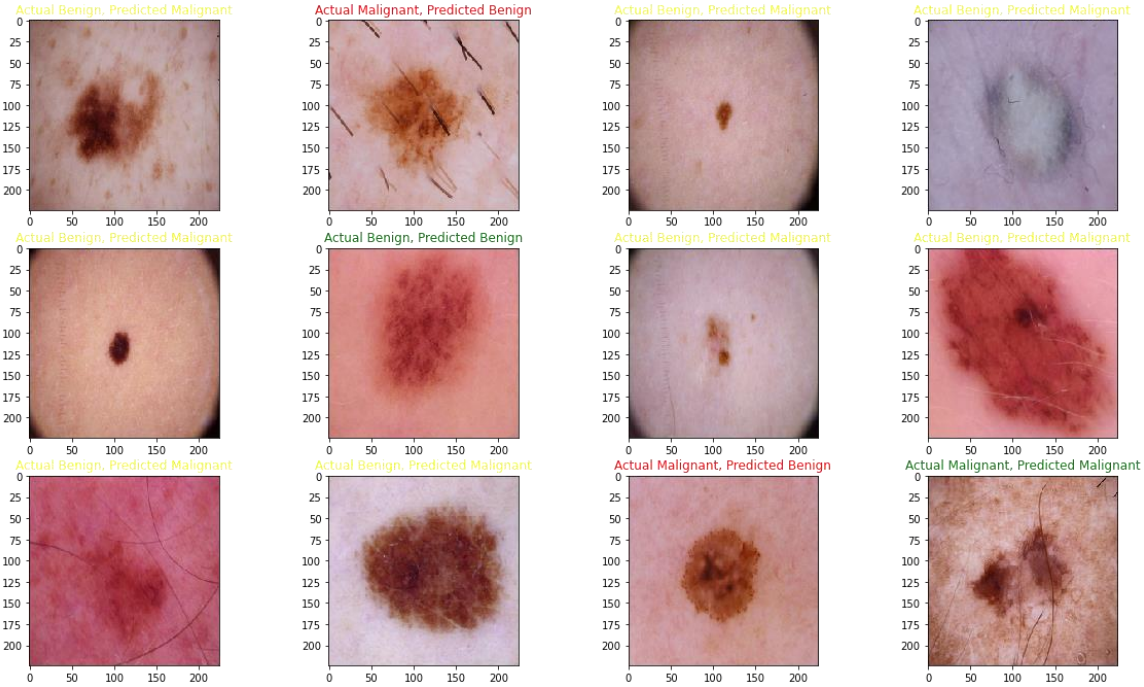


Fig 4. Naive Bayes prediction

As the accuracy score of the model was low, in fig 4., the model predicted most of them wrong. We can see in fig4., that only 2 images were classified correctly.

## V. ResNet50

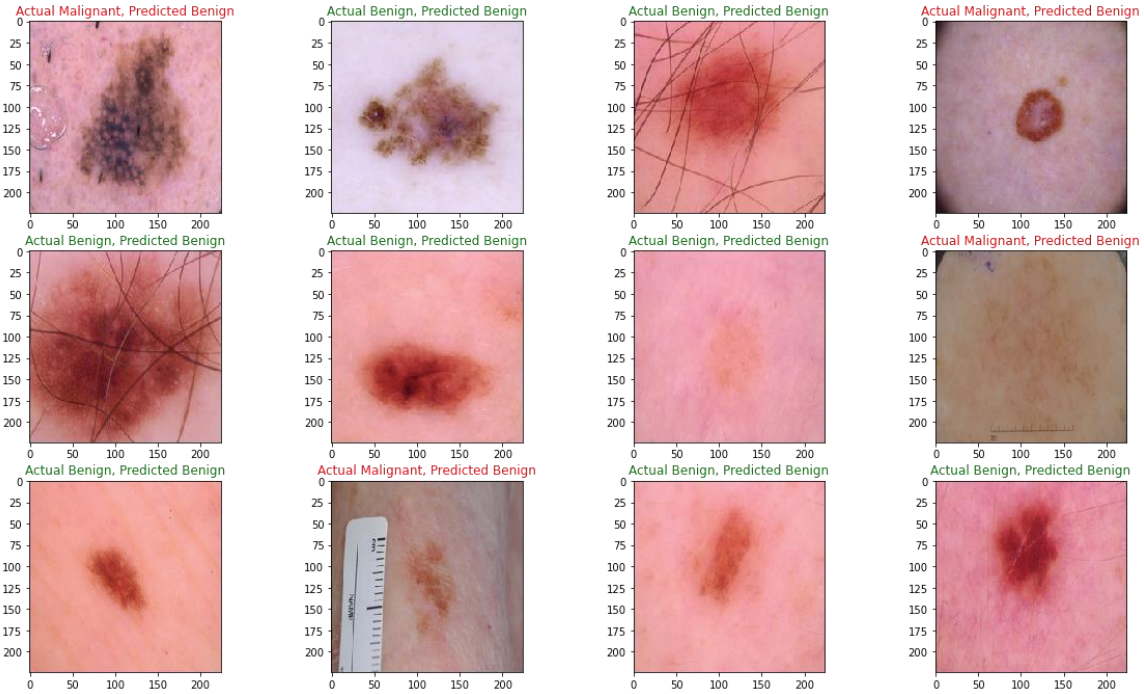


Fig 5. ResNet50 prediction

ResNet50 in general is a good model if the model had run for more epochs, then the predictions would have been better. Still in fig 5., we can see the model was able to predict most of the skin cancers correctly. I could also observe that the model was predicting once class output most of the time. But it changed on every run.

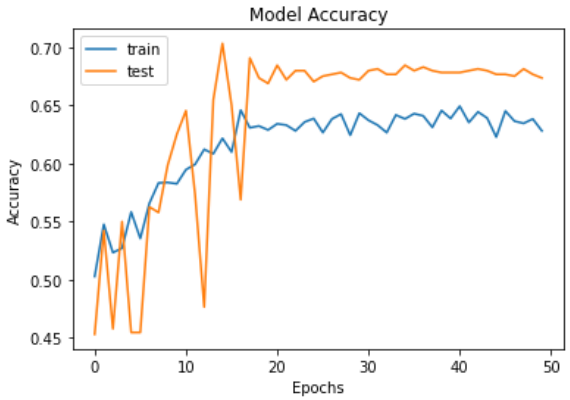


Fig 6. Train accuracy vs test accuracy of ResNet50

Fig 6., chart show how ResNet50 was able to get an increasing trend on accuracy with the train and test data.

## VI. VGG16

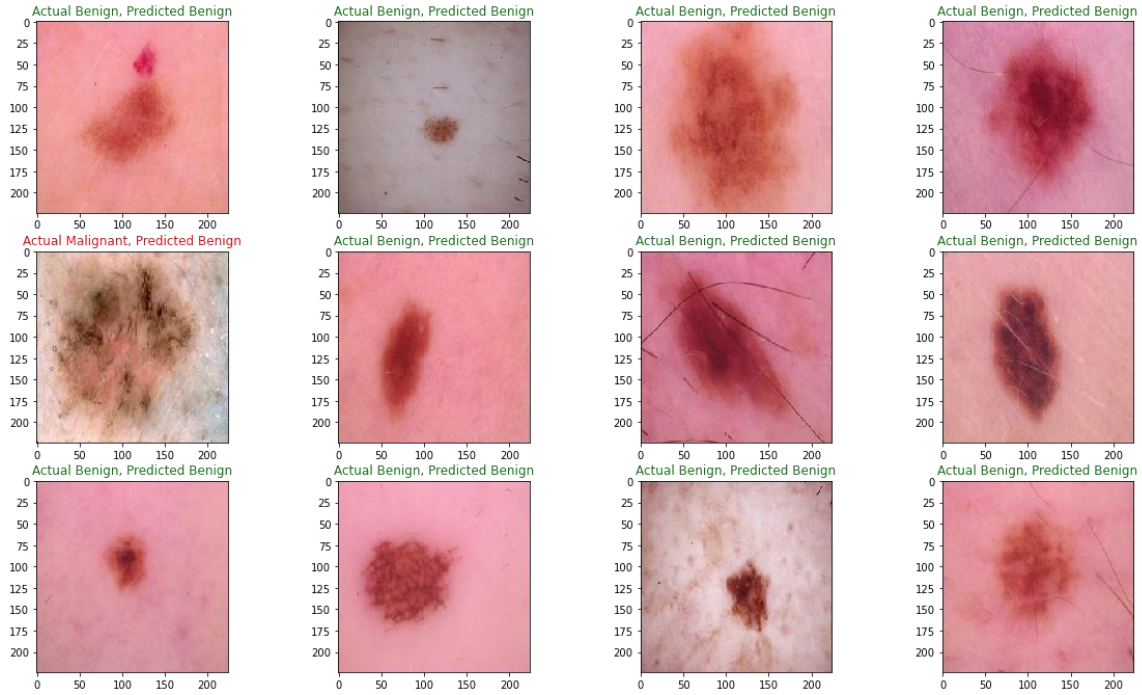


Fig 7. VGG16 Predictions

Even though the model got an accuracy of 82.34% the model was able to predict almost all of them correctly except few of them. If the model was trained on few more epochs, then we could have gotten a better prediction.

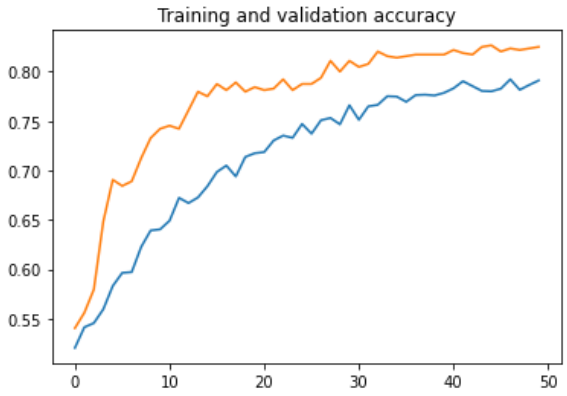


Fig 8. Train vs Test Accuracy

There was a smooth graph between both the training and testing accuracy.


Fig 9. Train vs Test Loss

Even the loss graph of VGG16 has decreasing trend on test and train data.

## 5. Conclusion

In this project I have used the Skin Caner dataset to determine whether a skin cancer is malignant or benign using different models. The data were called using the Image(). The dataset was balanced in regard to labels.

Out of all the six models' random forest classifier was able to get the best accuracy but the VGG16 model would have gotten a better accuracy if it would have gotten a little more epochs to learn. Even ResNet50 could have achieved better accuracy if it would have been trained on more epochs. Other models could have gotten better accuracy on different hyper parameters.

Below is the accuracy of all the models on the dataset.

| | Algorithm | Accuracy |
|---|---|---|
| 0 | Random Forest classifier | 83.333333 |
| 1 | Logistic Regression | 78.636364 |
| 2 | Decision Tree | 76.818182 |
| 3 | Naive Bayes | 70.151515 |
| 4 | ResNet50 | 67.812502 |
| 5 | VGG16 | 82.343751 |