ML Final Report

A. Introduction

Melanoma is a deadly type of skin cancer whose primary cause is Ultraviolet light (UV) exposure and it spreads quickly. As a result, it is the worst skin cancer and the leading cause of mortality. When a patient is diagnosed, the classification of cancer stages is a time-consuming and tedious task. Cancer diagnosis at the time of surgical therapy is primarily determined by the cancer's stage or tumour thickness.We broadly classify the melanoma into two categories namely benign melanoma and malignant melanoma. We aim to propose a model which can successfully classify the cancer stage backed by strong accuracy.
Benign is well-differentiated and has abnormally slow growing skin cancer which does not invade surrounding tissue or spread to other parts of the body. cells are not cancerous and it is comparatively easy to remove with surgery. Whereas,Malignant is poorly-differentiated and is abnormally fast growing skin cancer that can invade and destroy nearby tissues and can also spread to other body parts. cells are cancerous and it is difficult to remove them so the patient is treated with chemotherapy and radiation therapy.

B. Related Work

Melanoma can be detected in a variety of ways. Melanoma should be categorised as early as possible, and the patient should be diagnosed as soon as possible. There are many methods and techniques that classify mel- anoma and benign skin lesions (Sangve and Patil, 2014).
Patil and Bellary,(2020) proposed a system to classify melanoma stages bases on the thickness of the tumor. The two stage classification classify tumor thickness of $< 0.76mm$ to be first stage and tumor thickness $\geq 0.76mm$ in second stage. CNN + SMTP gave an accuracy of 92 percent.

C. Dataset and Evaluation

We have taken the Images dataset from ISIC archive. of 2000 images of benign and 2000 images of malignant. We have then split our data into 20 percent test data and 80 percent train data into 5 fold of cross validation i.e 20 percent in each fold. then we are performing preprocessing on the images then extracting features of 4000 images and classify them into two classes (benign and malignant) and storing those features to a CSV file.
We have also used another dataset which had 81 features extracted in it from dataset.

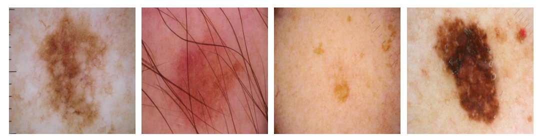

Figure 1. Malignant and Benign lesion types

C.1. Feature Extraction

We have preformed 4 pre-processing layers to our data (dull razor, median filter, Otsu and Chanvese algorithm) which removes hairs, data-segmentation and all the other noises from our images, after which we get our final images that we use for training and testing our model. we are extracting features like Asymmetry index - the average of the difference between lesion image and its horizontal flip and lesion image and its vertical flip and the total lesion area , Eccentricity-The ratio of the distances of a point on the shape from the focus, Border Irregularity - Border irregularity the compact index - $P^2/4\pi*A$, Diameter-Diameter of the Circle with same Area, Corelation- measures the joint probability occurrence of the specified pixel pairs, Homogeniety - measures the closeness of the distribution of elements in the GLCM with respect to the GLCM diagonal, Energy-It is computed as the square root of the sum of squared elements in the GLCM, Contrast- It measures the local variations in the GLCM
These features are then trained on different models.

C.2. Evaluation Metrics

We are using evaluation matrices like Accuracy(to calculate the accuracy of any classification), Precision(tells how precise or accurate our model is out of the predicted positive), Recall(Ratio of correct positive predictions to the total positives) and F1(the harmonic mean of precision and recall).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

$$Precision = \frac{(TP)}{(TP + FP)}$$

$$Recall = \frac{(TP)}{(TP + FN)}$$

$$F1Score = \frac{2 * (Recall * Precision)}{(Recall + Precision)}$$

TP = True Positive
TN = True Negative
FP = False Negative
FN = False Negative

D. Methodology

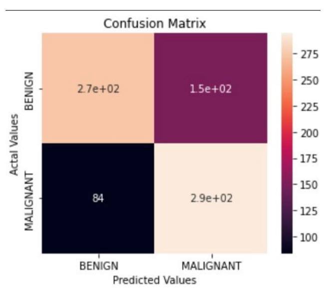

Figure 2. Confusion Matrix for logistic regression.

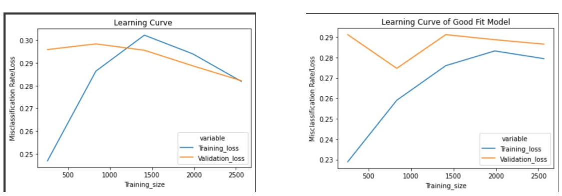

Figure 3. Loss vs Experience for logistic regression.

we have implemented two baseline models logistic regression and Random forest. Challenges that we are facing is computational power i.e. basic models could be easily trained for our data set but it under fits the data set and complex models would require a high computational power. In this we would be using supervised classification models. We have used the gridsearchCV for hyperparameter tuning Firstly in logistic regression we trained the model for 240 images and got an accuracy of 53 percent which indicated that our dataset was too small for applying models then we increased our dataset to 4000 images with just basic logistic regression which gave an accuracy of 64 percent. for further improving the model we did the hyperparameter tuning using gridsearch which improved its accuracy to 73 percent but further improvement was not possible since dataset is not linearly seprable. In confusion matrix we can clearly see that our model (Logistic Regression) has a good true positive rate for malignant cancer while for benign cancer the false positive is high .These graphs are the loss against the experience graph for 1000 and 2000 iteration for logistic regression since the training loss increases as the experience increases and the validation loss remains more or less constant that means that our model is under fitted.
Increasing number of iterations had no significant change as the logistic regression is a linear classifier which is not able to linearly separate our data set and hence our data set is not linearly separable.

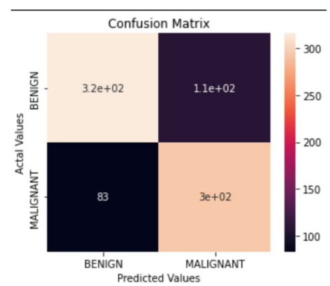

Figure 4. Confusion Matrix for Random Forest.

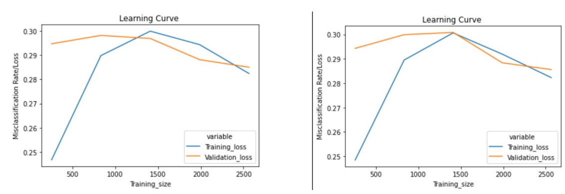

Figure 5. Loss vs Experience for Random Forest.

In this confusion matrix random forest performs better than logistic regression as the True positive of benign is significantly higher than that of the logistic regression.These are graphs of loss vs experience for different number of decision trees pasted the graph for 1000 and 2000 decision tree.

In the next step we trained KNN classifier - It decides the k nearest neighbours and works on the principal every data point near to same class belongs to that particular class. The training and validation loss of knn also decreases with experience this emplies it also has a good fit on the data with accuracy of 76.43 percent
The confusion matrix shows that both the classes are equally classified and misclassified

In the next step we trained Mlp classifier - Multilayer Perceptron has input and output layer with many hidden layers in between. At each layer the summation of weight

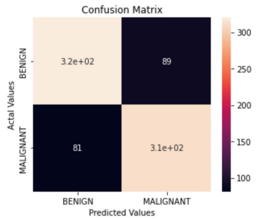

Figure 6. Confusion Matrix KNN

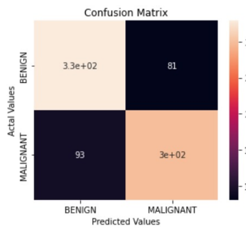

Figure 7. Confusion Matrix MLP

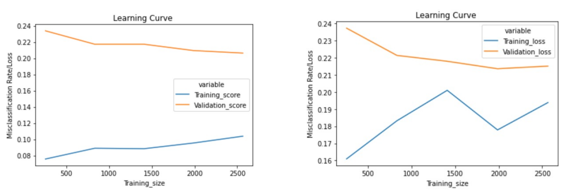

Figure 8. Loss vs epoch (i)KNN on left (ii)MLP on right

and features is fead to activation function and in backward propogation weights are updated.

Validation loss decreases as the experience increases but the train loss shows a different behaviour it increases at start then decreases later .So according to validation loss our model is good fit.

Confusion matrix has almost equally classified and misclassified both the classes.

In the next step we combined KNN , Random forest , Mlp classsifier and Svm using stacking classifier . Stacking classifier has two layers of models first layer consists of many models which give the outputs then the second layer model is trained on the output of firstlayer here first layer was - KNN , Random forest , Mlp classsifier and Svm and second layer was -logistic regression. The graph shows that it was a good fit model as training and validation loss decreases as the experience increases hence we got an accuracy of 77.9 percent.

Confusion matrix depicts that malignant is more misclassified as compared to benign

In all the above classifier we have used **gridsearch cv** to tune all the parameters on the basis of accuracy.

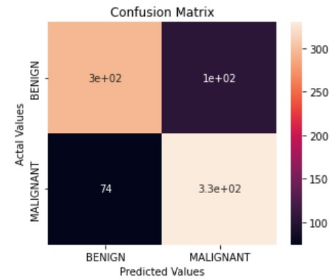

Figure 9. Confusion Matrix Stacking

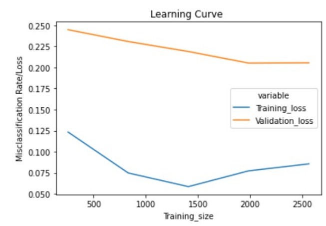

Figure 10. Loss vs epoch Stacking

Convolutional Neural Network: For an input sequence with n entries: x1,x2,..., xn , where n be total features in dataset x be the feature.The next step is applying convolution operation. The output of the convolutional process is then sent via non linear activation function. The most widely recognized nonlinear activation function utilizes ReLU that performs following function:

$$f(x) = max(0, x)$$

Rectified linear function is applied to the output of the hidden layers. Batch Normalisation is done between the layers of the neural network.It serves to speed up training and use higher learning rates, making learning easier. The pooling step entails sliding a 2D filter across each channel of the feature map and summing the features that exist in the area guarded by a channel.

The *binary cross entropy* is used as the loss function, it calculates the loss on the basis of how close or far we are from the actual value. The Optimizer is used as Adam with learning rate to be 0.01.The *Adam* optimizer has the best accuracy in enhancing the CNN ability in classification and segmentation. CNN was applied on two different feature set. The set with 7 features gave test accuracy of 77.56 while running the network on 80 extracted features gave an accuracy of 97.13

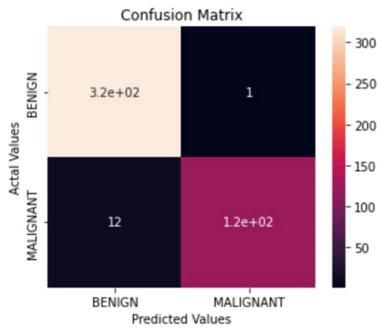

Figure 11. Confusion Matrix CNN

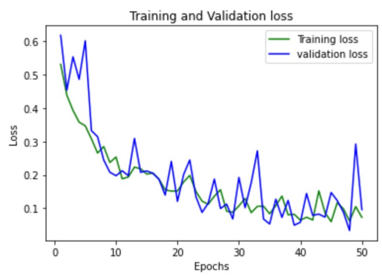

Figure 12. Loss vs epoch curve

E. Results

So far the **best model** was the **CNN** with **81 features** it had an accuracy of **97.13 percent** this was **better then the state of art** which was **96 percent**. We used CNN with cross entropy as the loss function on 81 features while the state of art had CNN with SMTP as loss function.

All the other have less accuracy because the no of features we extracted were 7 due to which all the other models were limited to accuracy of less than 80 percent on it. We solved this by using the dataset with 81 extracted features.

F. Contribution

F.1. Delivered

Arham and Vikhyat Collected Data and pre-processed it using multiple techniques to remive hair and other noise from the image. Vikhyat and Abhyudit visualised the data and Feature Extraction selection. Abhyudit and Arham performed EDA Training model and different Ensemble techniques for better accuracy.

F.2. Not Delivered

CNN with SMTP Loss function is not delivered, instead we did it using binary cross entropy.

F.3. Individual Contribution

Arham : Data collection , Pre-processing & Applying Models likes Logistic Regression, Random Forest, KNN.

Abhyudit : Pre-processing, Feature Extraction, Stacking Classifier , MLP Classifier.

Vikhyat : Pre-processing, CNN with binary cross Entropy, K-fold cross validation.

Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.7319	0.7341	0.7319	0.7321
Random Forest	0.7618	0.7629	0.7618	0.7616
MLP Classifier	0.7830	0.7836	0.7830	0.7831
KNN Classifier	0.7643	0.7643	0.7643	0.7643
Stacking Classifiers	0.7793	0.7820	0.7793	0.7796
CNN 7 features	0.7755	0.7804	0.7755	0.7760
CNN 80 features	0.9713	0.9734	0.9713	0.9717

References

[1] Patil & Bellary (2020) . *Machine learning approach in melanoma cancer stage detection*, Journal of King Saud University

[2] Pascal Getreuer, Image Processing On Line (2012) *Chan-Vese Segmentation*, scikit-image: Image processing in Python (2014)

[3] Murzova & Seth (2020) *Otsu's Thresholding with OpenCV*

[4] Jason Brownlee (2019) *A Gentle Introduction to Pooling Layers for Convolutional Neural Networks*