Mask vs. No-Mask classification

Kwanit Gupta , Soumya Vaish
B19EE046 , B19EE082

Abstract – This paper reports our experience with building a Mask vs. No-Mask classifier. We have made our custom dataset consisting of 10,470 images with equal proportions of masked and no-masked people and segregated the same from `self-built-masked-face-recognition-dataset`. Since the masked data was less, image augmentation was incorporated to create an equal proportion of both the classes. We used various classification algorithms and compared their results in this report. We applied the same on the group photos given in the `Real-World-Masked-Face-Dataset-master` and detected the human faces using inbuilt-Cascade classifier (Adaboost based model).

I. Introduction

Face recognition is a fascinating application in the field of image recognition. However, face images with masks create hindrance .The goal of this project is to classify images of people into two classes: Mask and No-Mask based upon whether they wear a mask in the image or not. However, there is a variety in the shape, design, orientation and appearance of masks which makes it challenging. The mask vs. no mask recognition can help in applications like face-recognition based entry systems, detection of suspicious faces through security cameras, etc.
The following images show few examples of masks vs no masks as well as a variety of masks present. (Source : ***self-built-masked-face-recognition-dataset****)*

No- masked data

Masked data

Datasets

self-built-masked-face-recognition-dataset *:* The respective csv file was referred and customly built as the training dataset.
The train dataset contains 10,470 rows where each row represents a train-image with 4097 columns containing :

- 1 Label column : Class label - 0 for no mask and 1 for mask
- 4097 latent vector columns
  The dataset has been split into train and test with test size of 0.5

II. Methodology

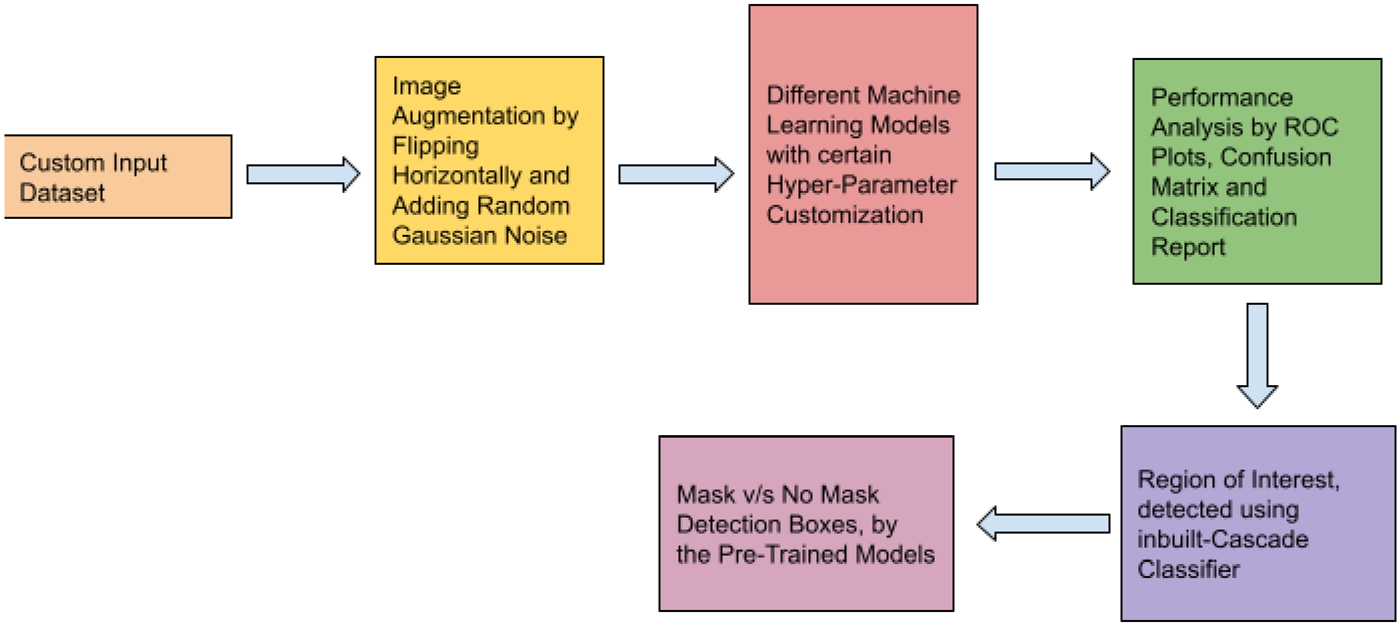
Overview

There are various classification algorithms present out of which we shall implement the following

- *MLP*
- *Random Forest Classification*
- *Decision tree classifier*
- *Logistic Regression*
- *Gaussian Naive Bayes*
- *KNN*
- *SVM*

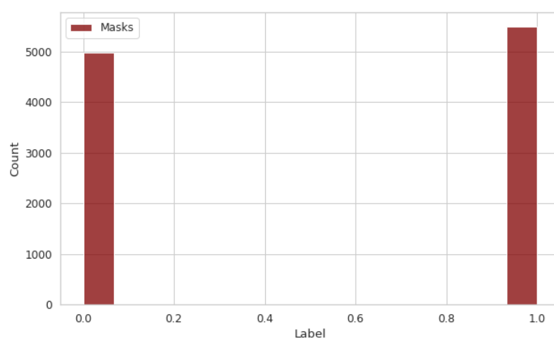
Classification Pipeline

Despite having different Machine Learning Models, the Classification Pipeline remained same and incorporated as following :-



Exploring the dataset and pre-processing

On counting the number of NULL values in the train dataset , it was found that there are no NULL values present.
The column data types for latent vectors are float and for masks field is integer which suits our requirements. Hence , there is no need for general pre-processing and the dataset is well-built.
The given graph was visualised between the classes:



where 0 means no-mask and 1 means masks.

Implementation of classification algorithms

- *Multilayer Perceptron :*MLP is a feedforward Neural Network which uses backpropagation to update weights and improve the results. MLP was applied on the training dataset with custom architecture (first hidden layer consisted of 100 neurons, second hidden layer consisted of 50 neurons and third with 25 neurons, accompanied by ReLU activation function with tolerance value of 10e(-6)).

- *Random Forest Classifier :* Random Forest Classifiers use boosting ensemble methods to train upon various decision trees and produce aggregated results.It is one of the most used machine learning algorithms. It was applied on the training dataset with n_estimators as 100.

- *Decision tree classifier :* A Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter. It was applied on the training dataset with default configuration.

- *Logistic Regression :*Logistic Regression model is widely used for binary classification and hence is well suited for classification into mask vs no mask with parameter n_jobs= -1 so that all the processors in CPU are used to speed up the code.

- *Gaussian Naive Bayes :*GNB is a type of Naive Bayes classifier that assumes that the distribution of data is gaussian and classifies data based on this assumption. It was applied on the training dataset with default configuration.

- *KNN (k - nearest neighbors) :*KNN are supervised algorithms which classify on the basis of distance from similar points.Here k is the number of nearest neighbors to be considered in the majority voting process with parameters n_neighbor=3 and n_jobs= -1 so that all the processors in CPU are used to speed up the code.

- *Support Vector Machine :*In SVM , data points are plotted into n-dimensional graphs which are then classified by drawing hyperplanes. The Gaussian Radial Basis Function was used as the kernel and `c` was taken as 30.

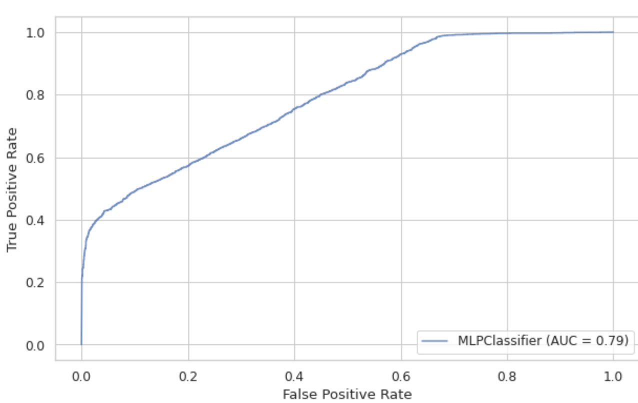
III. Evaluation of models

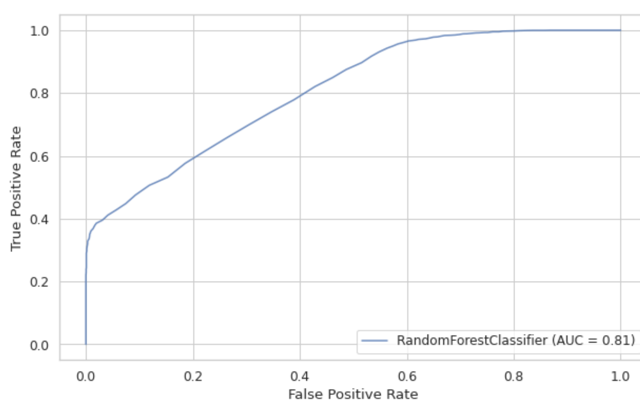
The models implemented were evaluated using techniques like - Classification report : precision , recall , f1 score and support , Confusion matrix , ROC plots , accuracy score and cross validation scores.
Table 1.1 contains the results obtained from the above techniques.

Table 1.1

	Accuracy	CV-Score
MLP	0.6914	0.586
Random Forest	0.7060	0.706
Decision Tree Classifier	0.6676	0.664
Logistic Regression	0.63	0.627
Gaussian Naive Bayes	0.61	0.597
KNN	0.68	0.685
SVM	0.71	0.719

ROC - Plots
2 fold classification was performed and cv score was obtained from the same.





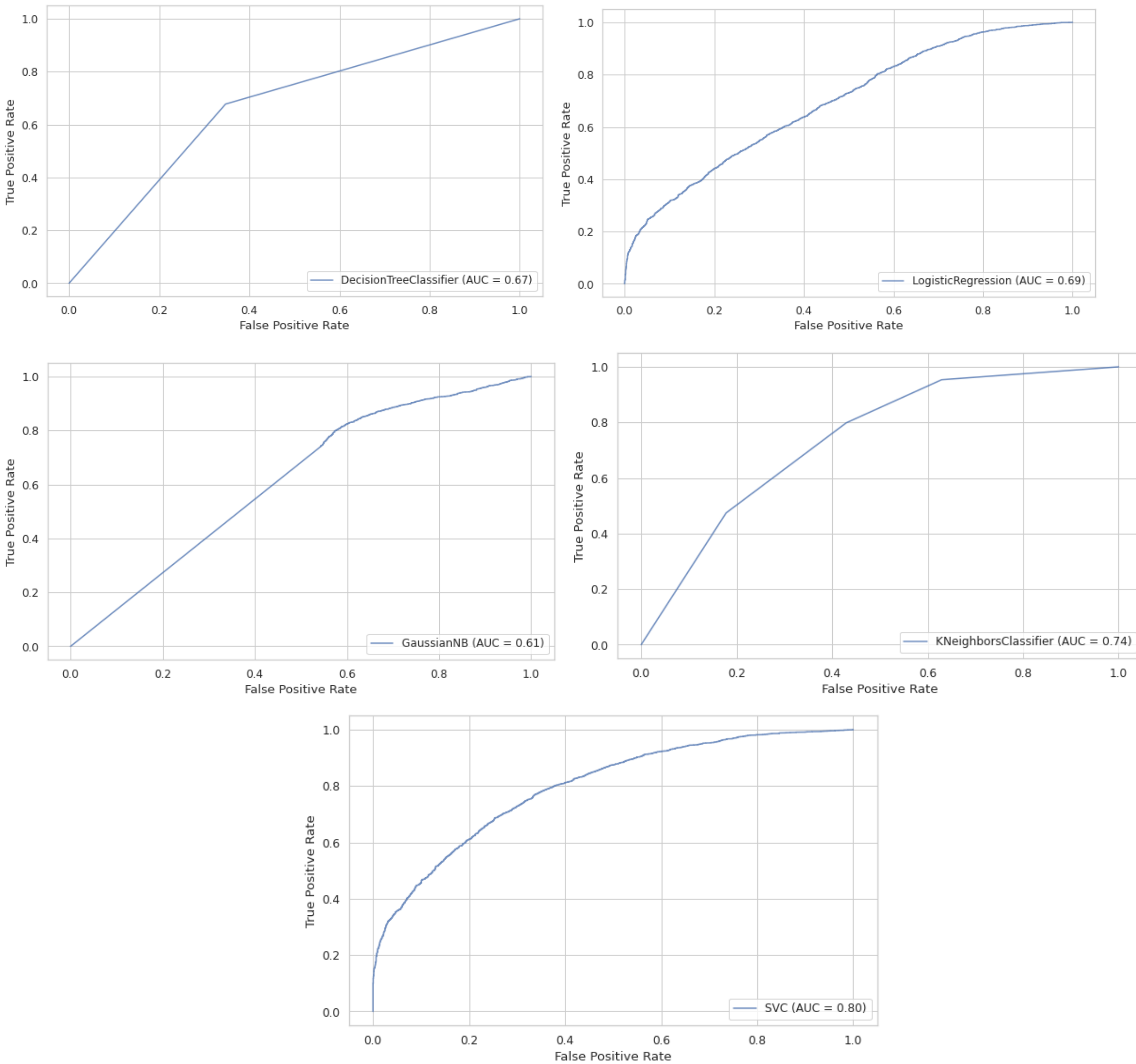
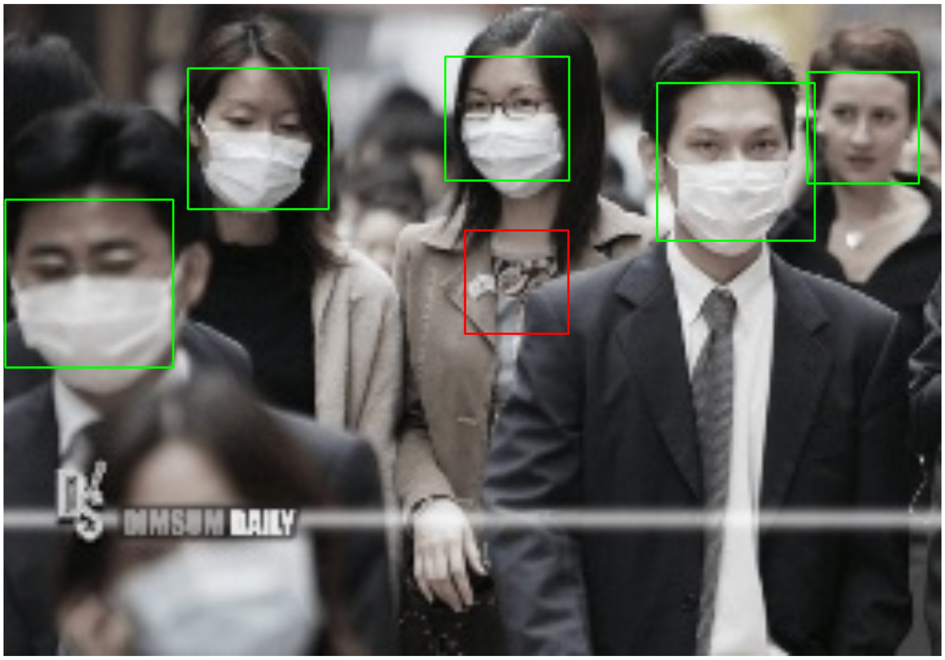


Fig1.1 : ROC plots for a) MLP , b) Random Forest , c) Decision Tree ,d) Logistic regression , e) Gaussian Naive Bayes , f) KNN g) SVM

VI. Results and Analysis

The table shows that all classifiers had accuracies in the range of 60-70%. Out of all the models, we saw that SVM and MLP outperformed others with the accuracy of 71% because in MLP, every weight associated with perceptrons captures every feature of an image and in SVM, we saw that gaussian performed better than others (linear, polynomial and sigmoid) because it normalises the data distribution which enabled us to deal with outliers. The Gaussian Naive Bayes did not perform well when compared to the others since in naive bayes, it is assumed that all the features are independent which was not the case.

Face recognition using Cascade-classifier:



Although the no-masked face was classified as masked, we can see the faces were correctly detected and classified. In other models the no-masked faces were outlined in red colour.

Contributions

The learning and planning was done as a team.The individual contributions are as given :-

- Soumya Vaish (B19EE082) : Data preprocessing and exploratory analysis. Implemented KNN , SVM, Logistic Regression and Gaussian Naive Bayes. Did the report.
- Kwanit Gupta (B19EE046) : Converted the image dataset into dataframe with necessary image augmentations. Implemented MLP, Random Forest and decision trees. Also made a face-detection system using a cascade classifier.

References

[1] Multi-Layer Perceptron | Sklearn Documentation
[2] Random Forest | Sklearn Documentation
[3] Decision Tree Classifier | Sklearn Documentation
[4] Logistic Regression | Sklearn Documentation
[5] Gaussian Naive Bayes | Sklearn Documentation
[6] Support Vector Classifier | Sklearn Documentation
[7] K-Neighbor Classifier | Sklearn Documentation
[8] Cascade Classifier | OpenCV Documentation