Face Mask Detection

Abstract

After the breakout of the worldwide pandemic COVID-19, there arises a severe need of protection mechanisms, face mask being the primary one. The basic aim of the project is to detect the presence of a face mask on human faces using images.

I. Introduction

Since November 2019, the COVID-19 epidemic had been a major social and healthcare issue.Wearing a face mask has become a necessity in order to stop the spreading of this deadly virus that has taken lives of many.It is necessary to wear masks in public places . Even with the successful development of many vaccines, wearing a mask is still one of the most effectiveand affordable ways to block 80 percentage of all respiratory infections and cut off the route of transmission.Even after this knowledge there are still a considerable number of people who forget or refuse to wear masks, or wear masks improperly.Therefore, for this face mask monitoring systems have been developed in order to supervise.

II. Goal

The Goal of this project is to find out the best and accuracte fitting model for face mask detection after fitting the data to various models and comparing these models.

III. Implementation

A. Preprocessing

Pre-processing is the process of transforming raw data into a comprehensible format such that a machine learning model can understand. For finishing the pre-processing of the addressed data set, we use an end-to-end pipeline.
Pre-processing entails the following steps:

1)The extraction of the zip files using python
2)Two lists have been created from the dataset of images which are data and label.
3)Iteration to each of the folders has been done. and i is labelled as path of each folder and j is labelled as the path for each image in i.
4) If there is no mask then 0 has been appended for the label.If there is mask then 1 has been appended for the label.
5) As all the images are not of the same size they were resized using
data = np.array(data).reshape(-1,100*100*3)
label = np.array(label).reshape(-1,1) 6) The shape of the features, data and label are observed.
7) The numpy arrays of features are converted into pandas dataframe.
8) Splitting of dataset equally into training and testing dataset.
9)Shape of the training and testing dataset of the feature data is observed.
10)Standardization of data is done using standard scaler build in functionality of sklearn.

B. Classification

1) Decision Trees: A Decision Tree is a basic diagram for categorizing examples. It is a type of Supervised Machine Learning in which data is continually separated according to a parameter.
The major benefit of utilizing a decision tree is that it separates data, and since we're dealing with binary classification, we'll just need to split it into two leaves per feature.
In this case, the scaled data is inserted into the decision tree and accuracy of the model has been calculated.

2) Naive Bayes: It's a probabilistic classifier that can figure out the pattern of evaluating a group of categorized documents. It compares the contents of the documents to a list of terms to categorize them into the appropriate category.
The Python NLTK package may be used to train and classify using the Nave Bayes Machine Learning approach.
The Gaussian Naive Bayes Classifier was utilized in this example,and the model's accuracy has then improved.

3) Perceptron: For binary classification problems, the Perceptron is a linear machine learning technique. It's a classification algorithm that uses a linear predictor function to combine a set of weights with the feature vector to create predictions.
The classifier is fitted with the scaled data, and accuracy is determined.

4) Support Vector Machine: The data is analyzed using a support vector machine (SVM), decision boundaries are defined, and computations are conducted in input space using kernels. For face mask detection, a Gaussian kernel is utilized.
We selected the one versus rest classification because it considers the masked label to be one and the unmasked label to be rest. The model's accuracy was then calculated.

C. Data

Our model was trained on two datasets, one of which contained images of people with face masks and the other one without face masks.

D. Evaluation of model

We evaluate our model with the metrics, the binary accuracy,log loss and f1 score.
We chose these specific metrics to compare our models as they are the most common and widely used metrics in analysing the face mask detection predictions.

E. Experimental details and results

The machine learning models used are as discussed above Gaussian Naive Bayes Classifier, Decision Tree classifier,Perceptron and Support Vector Machine classifier.

Model and Specification	Accuracy
Decision tree classifier	89.78
Gaussian Naive Bayes Classifier	85.46
Perceptron	95.32
SVM classifier with rbf kernel	97.324

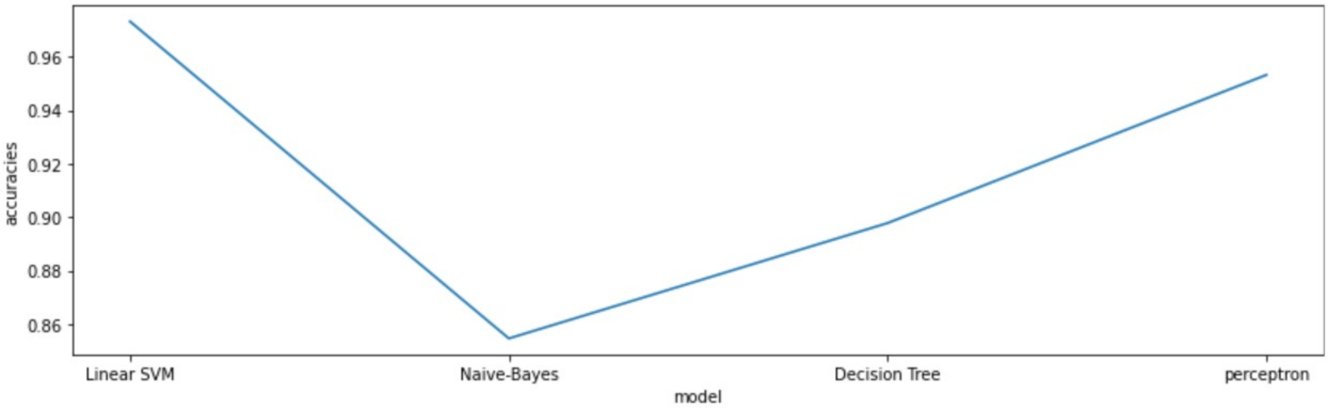
Below is the table of log loss of different models.
log loss measures the performance of a classification model whose output is a probability value between 0 and 1. log loss increases as the predicted probability diverges from the actual label

Model and Specification	log loss
Decision tree classifier	3.58
Gaussian Naive Bayes Classifier	5.02
Perceptron	1.61
SVM classifier with rbf kernel	0.92

Below is the table of f1score of different models.
The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall.

Model and Specification	f1 score
Decision tree classifier	0.85
Gaussian Naive Bayes Classifier	0.79
Perceptron	0.93
SVM classifier with rbf kernel	0.96

Here is a graph of all the accuracies obtained in a visual manner for a better comparative analysis. As mentioned in the table, we observe a sharp decline in the accuracies for Gaussian Naive Bayes Classifier.



IV. Conclusion

From the results obtained it can be concluded that the best fitting model for face mask detection is Support Vector Machine Classifier with rbf kernel as it has the highest accuracy and f1 score and also the least log loss.

V. Contributions

The face mask detection project was completed by Nandani(B19EE054) and Gudur Venkata Rajeshwari(B19EE032).
The implementation of classifiers Decision tree and Perceptron were done by Nandani.
The implementation of classifiers Gaussian Naive Bayes and SVM with rbf kernel were done by Gudur Venkata Rajeshwari.
The report was made by both Nandani and Gudur Venkata Rajeshwari