Course Project - Masked Image Classification

Pattern Recognition and Machine Learning

CSL2050

***Abstract* - This document is an analytic report for the course project for the Pattern Recognition and Machine Learning course.**

I. INTRODUCTION

For the project, we use the masked and unmasked images data set. This data set contains around 90,000 images. it contains both masked and unmasked images.

We have 2 classes 0 and 1. 0 represents an unmasked image and 1 represents masked image.

Due to lack of time, we have used just 15,890 images from the data set. First 15,890 images were selected from the data and put into a folder. The link to the zip file of the folder containing these images is https://drive.google.com/file/d/1PXvUqWKJzNlBUaiQ3o3NX-9FUI03eO94/view.

The complete data set can be downloaded from the link provided. The algorithm can be extended for a complete dataset by putting the images in a single folder and renaming them accordingly. The link is https://drive.google.com/file/d/1EG_A3kRwaPn15AFUGmEaQXatKZNXrWxH/view.

The csv file for the modified dataset containing label values of classes, i.e. 0 and 1 for unmasked and masked respectively can be downloaded from the link mentioned here. The link to the csv file is https://drive.google.com/file/d/12Fm_bF4TfDMBmxiv_9CqaW7rbnW6AEoP/view.

The updated data set contains 340 masked images and rest unmasked images.

II. DATA UPLOADING

First, the data mentioned above was uploaded to google drive to make it easily accessible.

Then, google drive was mounted in the google colab notebook file. Required dependencies were imported.

The data was then uploaded in the colab file. The data contained jpg images, so PIL.Image library was used to convert the images into numpy arrays of pixels.

III. Preprocessing

For preprocessing, first the matrix was converted to an array, and then the extra data points from each array was removed from the beginning as we only need the lower part, where the mask is worn.

Then, the data was divided into training, testing and validation data sets. This was done to avoid overfitting.

The data points in the test batch were allotted to the testing data set.

The data was then scaled using the StandardScaler model from sklearn.preprocessing library.

IV. Dimensionality Reduction

For dimensional reduction, Linear Discriminant Analysis was performed.

Linear Discriminants are used in classification problems where the inter class distance between different classes is increased, whereas the intraclass distance is decreased.

We fit the Linear Discriminant model from sklearn.linear_discriminant_analysis library over the training data. Then, the feature variables for training data, testing data and validation data were transformed using the model.

V. Feature Selection

For feature selection, SelectKBest model from the sklearn.feature_selection library.

SelectKBest method selects the k best features from the entire feature variable set.

Thus, SelectKBest removes all but k highest scoring features. Thus, the best 100 features among all other features were selected for creating a new data set.

The SelectKBest model was fitted over the training data. Then, the training, testing and validation feature variables were transformed using the trained model.

VI. Classifier Models

Decision tree Classifier, Adaboost Classifier, Naive Bayes Classifier and MultiLayer Perceptron Classifier were used for training over the training data and testing over the validation and testing data.

Performance of these models are compared below.

A. Decision Tree Classifier

Decision Trees are a non-parametric supervised learning method used for classification and regression.

The DecisionTreeClassifier model from sklearn.tree library was used for training, validation and testing. The max_depth was kept 7 to avoid overfitting.

Three kinds of models were trained, i.e. scaled data (transformed using StandardScalar), data with reduced dimensions (transformed using LinearDiscriminantAnalysis), and data with reduced features (transformed using SelectKBest).

The obtained results are:

	Training Accuracy	Validation Accuracy	Testing Accuracy
Scaled Data	0.988	0.974	0.971
LDA Transformed data	1.000	0.962	0.962
Data obtained from feature selection	0.984	0.977	0.975

* Accuracies are rounded off upto 3 decimal digits.

5 fold Cross validation was applied for each model over the training data.

The obtained results are:

Scaled Data	0.972	0.971	0.969	0.971	0.972
LDA Transformed data	0.999	0.999	1.000	1.000	1.000
Data obtained from feature selection	0.972	0.977	0.975	0.974	0.974

* Accuracies are rounded off upto 3 decimal digits.

B. Adaboost Classifier

Adaboost Classifier is an ensemble method that uses multiple weak learners to create a strong classifier. This method helps decrease overfitting.

The AdaBoostClassifier model from sklearn.ensemble library was used for training, validation and testing. The base estimator was kept as a DecisionTreeClassifier with max_depth as 3. Number of n_estimators was kept equal to 10.

Three kinds of models were trained, i.e. scaled data (transformed using StandardScalar), data with reduced dimensions (transformed using LinearDiscriminantAnalysis), and data with reduced features (transformed using SelectKBest).

The obtained results are:

	Training Accuracy	Validation Accuracy	Testing Accuracy
Scaled Data	0.986	0.978	0.979
LDA Transformed data	1.000	0.962	0.962
Data obtained from feature selection	0.983	0.979	0.977

* Accuracies are rounded off upto 3 decimal digits.

5 fold Cross validation was applied for each model over the training data.

The obtained results are:

Scaled Data	0.975	0.972	0.974	0.975	0.973
LDA Transformed data	0.999	0.999	1.000	1.000	1.000
Data obtained from feature selection	0.979	0.976	0.974	0.976	0.974

* Accuracies are rounded off upto 3 decimal digits.

C. Naive Bayes Classifier

Naive Bayes Classifier uses probability to find the posterior probability for a data point being in a particular class. The posterior probability is calculated assuming that all features are independent.

The GaussianNB model from sklearn.naive_bayes library was used for training, validation and testing.

Three kinds of models were trained, i.e. scaled data (transformed using StandardScalar), data with reduced dimensions (transformed using LinearDiscriminantAnalysis), and data with reduced features (transformed using SelectKBest).

The obtained results are:

	Training Accuracy	Validation Accuracy	Testing Accuracy
Scaled Data	0.724	0.740	0.714
LDA Transformed data	0.999	0.970	0.966
Data obtained from feature selection	0.746	0.749	0.731

* Accuracies are rounded off upto 3 decimal digits.

5 fold Cross validation was applied for each model over the training data.

The obtained results are:

Scaled Data	0.754	0.730	0.720	0.698	0.735
LDA Transformed data	0.999	0.999	0.998	1.000	0.999
Data obtained from feature selection	0.735	0.758	0.740	0.740	0.757

* Accuracies are rounded off upto 3 decimal digits.

VII. Conclusion

The data was analysed, uploaded, and preprocessed. Models were trained for different classifiers.

Maximum validation accuracy observed when AdaBoostClassifier was trained for data obtained from feature selection. The observed validation accuracy was 0.979.

Maximum training accuracy observed when AdaBoostClassifier was trained for scaled data. The observed validation accuracy was 0.979.

Maximum individual accuracy for cross validation was observed when DecisionTreeClassifier and AdaBoostClasisfier were trained for LDA transformed data. The observed accuracy was 1.000.

Maximum average accuracy for cross validation was observed when DecisionTreeClassifier and AdaBoostClasisfier were trained for LDA transformed data. The observed accuracy was 1.000.

All models were compared and data was visualized. This helped to understand the working of image related data in the real world and also helped to get a better understanding about how Machine Learning is useful for real world applications.