Pattern Recognition and Machine Learning Course Minor Project

Detecting whether the person is wearing masks or not

Team :
Aniket Wani (B19EE009)
Harpal Sahil Santosh (B19CSE107)

ABSTRACT

In today's era , the images dataset are widely used for various purposes like face recognition,vehicle and number plate identification. For processing and understanding these and to extract useful information from images ,we use Natural Images Processing.Image processing is manipulation of an image that has been digitised and uploaded into a computer. Software programs modify the image to make it more useful, and can for example be used to enable image recognition.

Introduction

Our approach in image processing is that we have use various machine learning algorithms and their performance on the with mask and without mask face images dataset with and without feature selection. The dataset includes 4406 sample images of objects belonging to 2 different classes : class 0 = without mask and class 1 = with mask . We used 3 different classifiers as : Support Vector Machines, Random forest,k nearest neighbor.

Dataset

To train and compare machine learning models to classify whether a person is wearing a mask or not, we need to find a good dataset with a fair amount of images for both classes:

- wearing a mask
- not wearing a mask

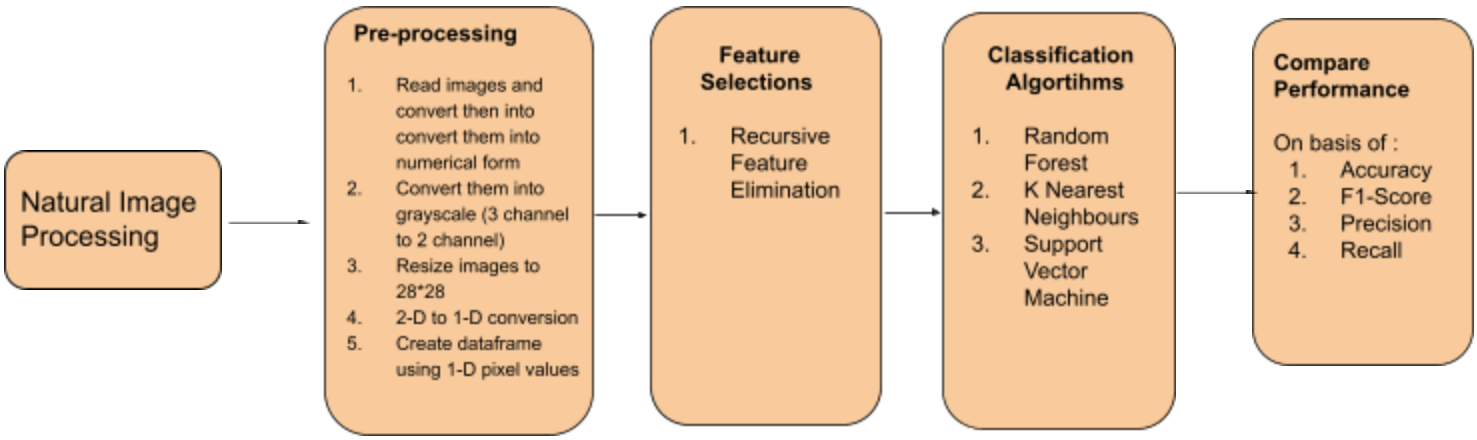
self-built-masked-face-recognition-dataset.zip provides just what we need! In this data set there are a total 2203 with mask images and 90468 without mask images.

In our project we have taken all masked images and randomly selected the same number of without mask images from available dataset, so that our model won't give biased results just because the number of without mask images are more. In the given dataset only the face part of the person is visible and that is nothing but our region of interest. So, we don't need to explicitly select ROI.

Then we performed some pre processing, Such as,

1) Read images and convert then into convert them into numerical form
2) Convert them into grayscale (3 channel to 2 channel)
3) Resize images to 28*28
4) 2-D to 1-D conversion
5) Create dataframe using 1-D pixel values
6) Standardize dataset
7) Split data into train and test (50% for training and 50% for testing)

Machine learning Pipeline



Algorithms :

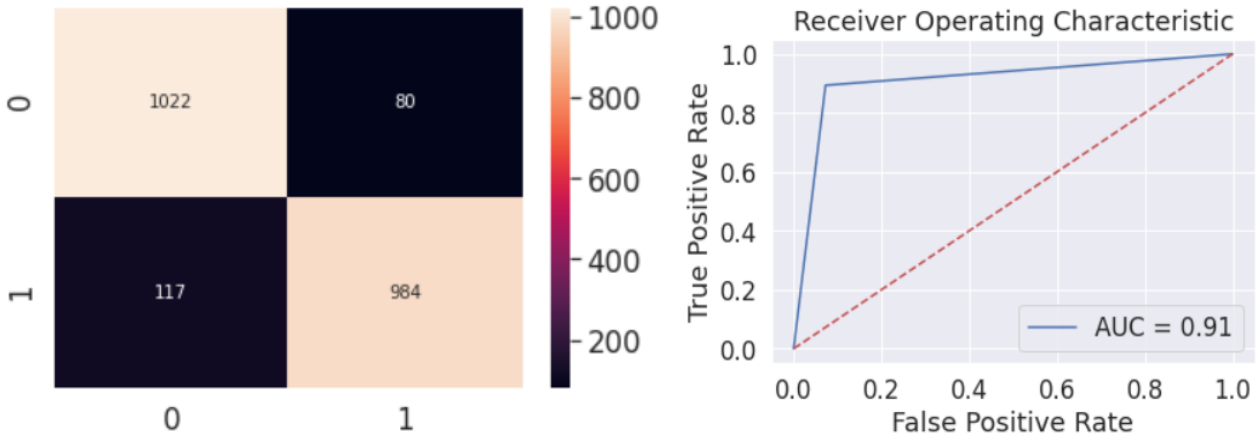
Support Vector Machine:

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N - the number of features) that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.
In case of binary classification Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. To select hyperplanes we consider support vectors.Here support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these support vectors, we maximize the margin of the classifier.

- Train score of support vector machine 96.23241034952338 %
- Accuracy of support vector machine : 91.05764866091693 %

Classification Report:

	precision	recall	f1-score	support
0	0.90	0.93	0.91	1102
1	0.92	0.89	0.91	1101
accuracy			0.91	2203
macro avg	0.91	0.91	0.91	2203
weighted avg	0.91	0.91	0.91	2203



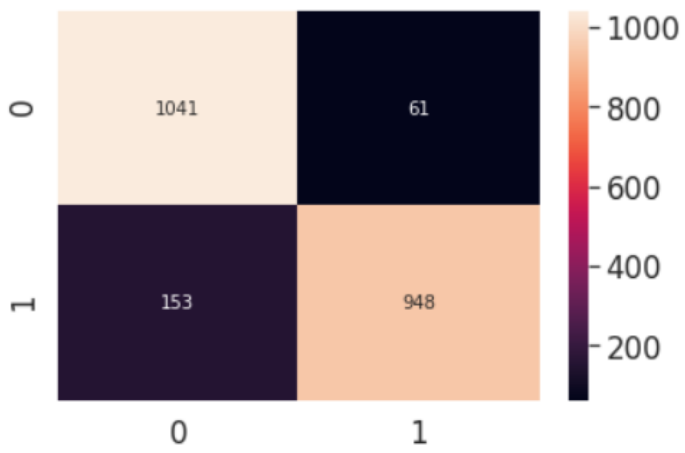
Random forest:

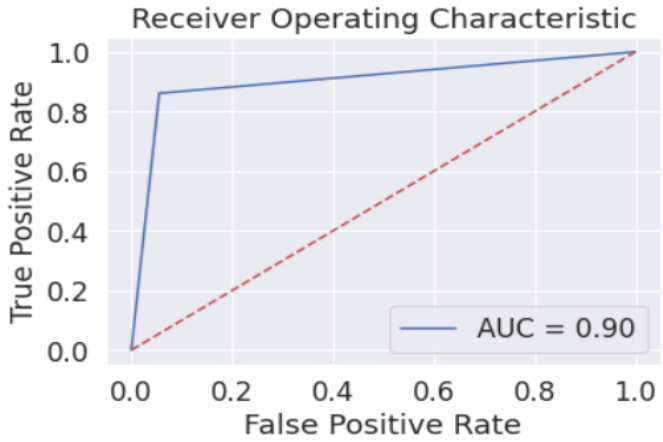
Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest consists of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets predictions from each prediction from each tree and selects the best solution and selects the best solution by means of voting. The individual decision trees are generated using an attribute selection indicator such as information gain, gain ratio, and Gini index for each attribute. Each tree depends on an independent random sample. In a classification problem, each tree votes and the most popular class is chosen as the final result.

- Train score of random forest 100.0 %
- Accuracy of random forest : 90.2859736722651 %

Classification Report:

	precision	recall	f1-score	support
0	0.87	0.94	0.91	1102
1	0.94	0.86	0.90	1101
accuracy			0.90	2203
macro avg	0.91	0.90	0.90	2203
weighted avg	0.91	0.90	0.90	2203





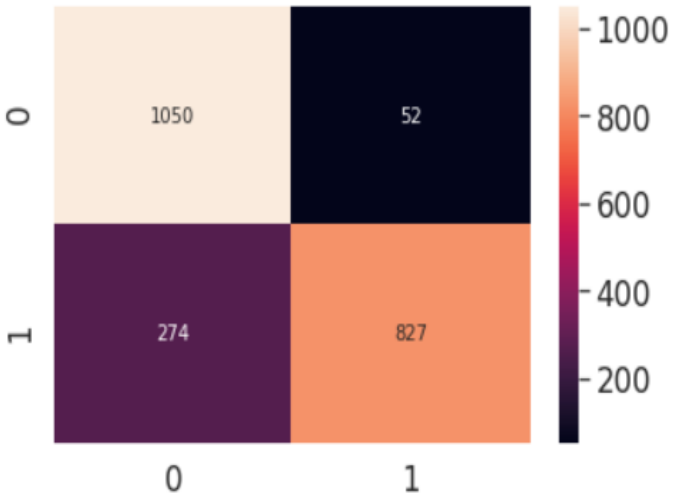
KNN:

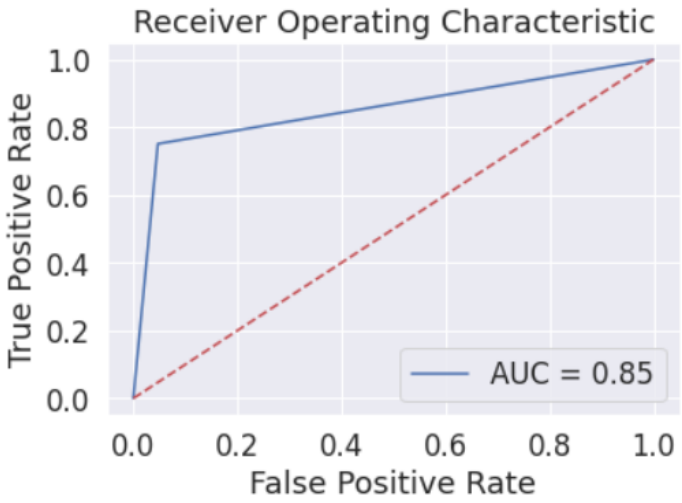
K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

- Train score of knn classifier 89.9682251475261 %
- Accuracy of knn classifier : 85.20199727644122 %

Classification Report:

	precision	recall	f1-score	support
0	0.79	0.95	0.87	1102
1	0.94	0.75	0.84	1101
accuracy			0.85	2203
macro avg	0.87	0.85	0.85	2203
weighted avg	0.87	0.85	0.85	2203





Recursive Feature Elimination:

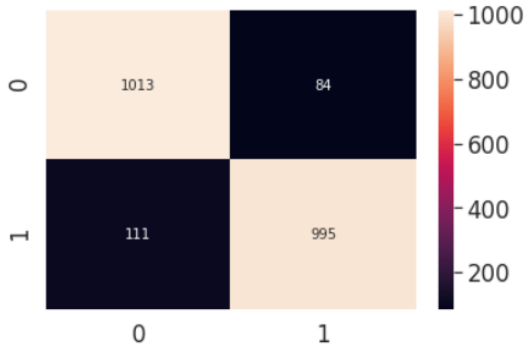
RFE is a wrapper-type feature selection algorithm. This means that a different machine learning algorithm is given and used in the core of the method, is wrapped by RFE, and used to help select features. This is in contrast to filter-based feature selections that score each feature and select those features with the largest (or smallest) score. We reduced the dimensionality of our dataset from 784 to 100 and got comparatively better results than before feature elimination.

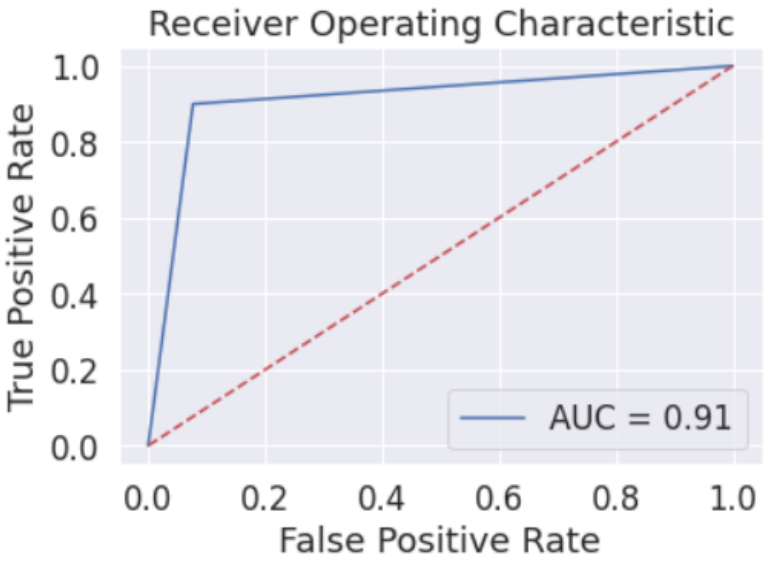
selected features : [3, 5, 6, 7, 8, 9, 10, 11, 12, 15, 17, 20, 24, 179, 180, 181, 182, 183, 184, 185, 204, 207, 208, 209, 210, 211, 212, 213, 214, 216, 293, 297, 326, 327, 328, 406, 407, 426, 427, 428, 429, 433, 434, 435, 436, 440, 455, 462, 466, 468, 469, 496, 497, 498, 525, 551, 568, 570, 571, 572, 573, 574, 575, 576, 578, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 626, 627, 628, 629, 630, 631, 632, 633, 658, 683, 684, 685, 686, 687, 688, 689, 690, 711, 712, 713, 714, 715, 716, 717]

SVM After feature Selection

- Train score of support vector machine after applying feature selections : 93.5088515660463 %
- Accuracy of support vector machine after applying feature selections : 91.1484339536995 %

	precision	recall	f1-score	support
0	0.90	0.92	0.91	1097
1	0.92	0.90	0.91	1106
accuracy			0.91	2203
macro avg	0.91	0.91	0.91	2203
weighted avg	0.91	0.91	0.91	2203

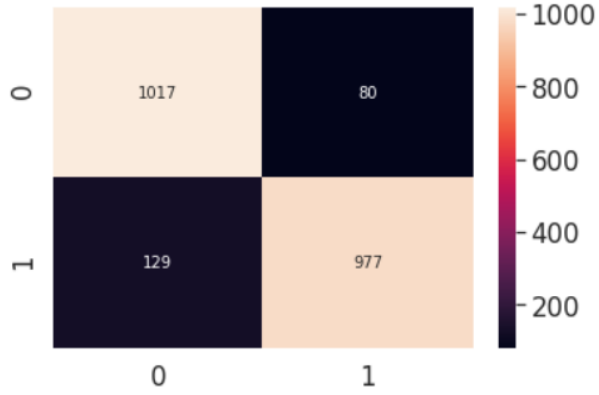




Random Forest After Feature Selection

- Train score of random forest after applying feature selections : 100.0 %
- Accuracy of random forest after applying feature selections : 90.51293690422152 %

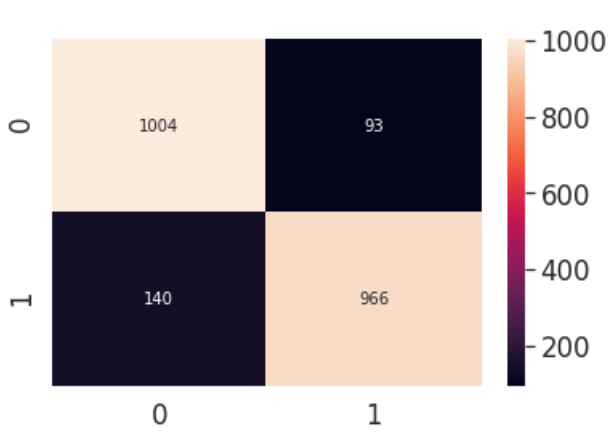
	precision	recall	f1-score	support
0	0.89	0.93	0.91	1097
1	0.92	0.88	0.90	1106
accuracy			0.91	2203
macro avg	0.91	0.91	0.91	2203
weighted avg	0.91	0.91	0.91	2203

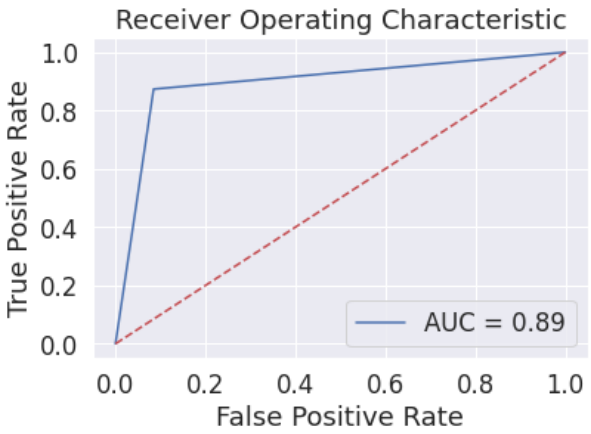


KNN After Feature Selection

- Train score of knn classifier after applying feature selections : 92.55560599182933 %
- Accuracy of knn classifier after applying feature selections : 89.42351339083069%

	precision	recall	f1-score	support
0	0.88	0.92	0.90	1097
1	0.91	0.87	0.89	1106
accuracy			0.89	2203
macro avg	0.89	0.89	0.89	2203
weighted avg	0.89	0.89	0.89	2203





Contribution of Each Member :

1) Aniket Wani (B19EE009) : Build the code for preprocessing and visualization and error correction in code and training Support Vector Machine with and without feature selection and KNN after feature selection. Build report

2) Harpal Sahil Santosh (B19CSE107) : Preprocessing of code, Random forest classifier with and without feature selection, KNN before feature selection, build report,