

CSL2050
Pattern Recognition and Machine Learning
Facemask Detector Using Machine Learning

Jahnavi Juluri(B19CSE109)
Buruka Nikhitha (B19EE019)

Loading the data

```
!7z x '/content/drive/MyDrive/data/Real-World-Masked-Face-Dataset-master.zip'
```

Details-

```
Folders: 29  
Files: 1222  
Size:      187996668  
Compressed: 187072300
```

```
!7z x '/content/drive/MyDrive/data/self-built-masked-face-recognition-dataset.zip'
```

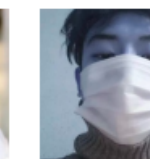
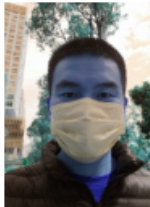
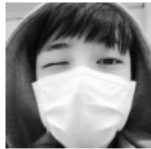
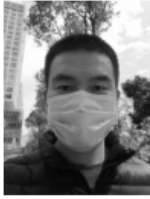
Details-

```
Folders: 987  
Files: 92672  
Size:      627948526  
Compressed: 636274634
```

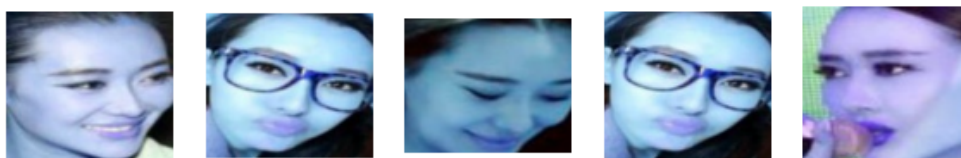
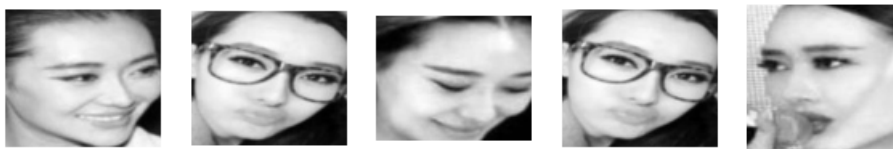
This command is used to load all the files from the zip file present in the drive into the content folder in the colab files section so as to read the data and use them in the project.

Data Visualization

- With Mask



- Without Mask

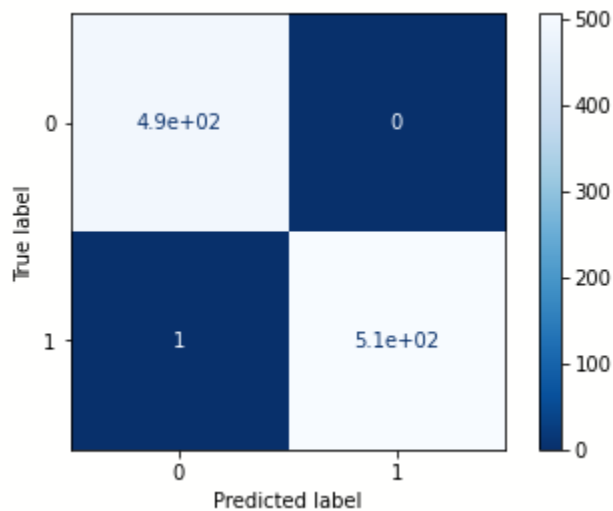


Training the Models

1) Support Vector Machine

```
svm_param = {'kernel': ('linear','poly', 'rbf'), 'C':  
(0.01,0.1,1,10)}  
grid = GridSearchCV(SVC(), svm_param)  
grid.fit(x_train,y_train.values.ravel())  
svm = grid.best_estimator_  
svm.fit(x_train,y_train.values.ravel())
```

Confusion Matrix-



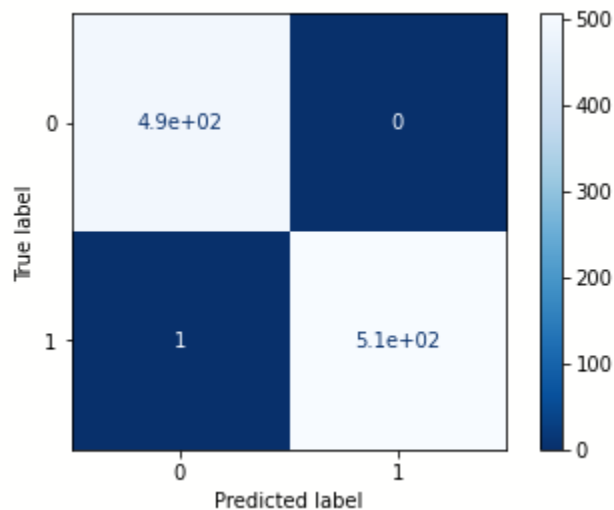
Classification Report-

	precision	recall	f1-score	support
0	1.00	1.00	1.00	493
1	1.00	1.00	1.00	507
accuracy			1.00	1000
macro avg	1.00	1.00	1.00	1000
weighted avg	1.00	1.00	1.00	1000

2) KNN Classifier

```
knn_param = {'n_neighbors':[2,3,4,5], 'leaf_size':[3,6,5,4]}  
grid2 = GridSearchCV(KNeighborsClassifier(), knn_param)  
grid2.fit(x_train,y_train.values.ravel())  
knn = grid2.best_estimator_  
knn.fit(x_train,y_train.values.ravel())
```

Confusion Matrix-



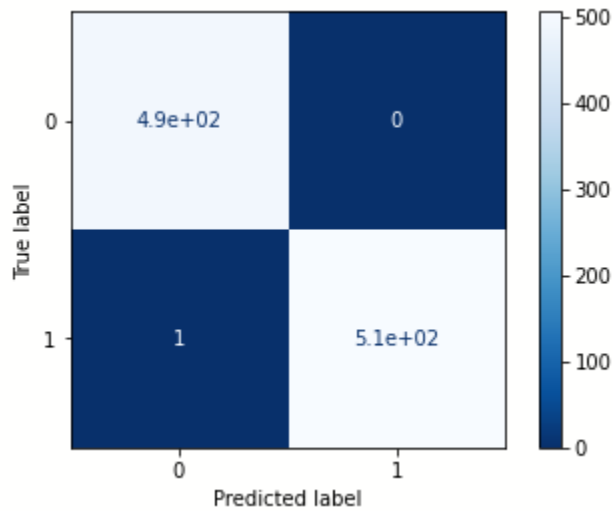
Classification Report-

	precision	recall	f1-score	support
0	1.00	1.00	1.00	493
1	1.00	1.00	1.00	507
accuracy			1.00	1000
macro avg	1.00	1.00	1.00	1000
weighted avg	1.00	1.00	1.00	1000

3) MLP Classifier

```
mlp = MLPClassifier(random_state=1, max_iter=300).fit(x_train,  
y_train.values.ravel())
```

Confusion Matrix-

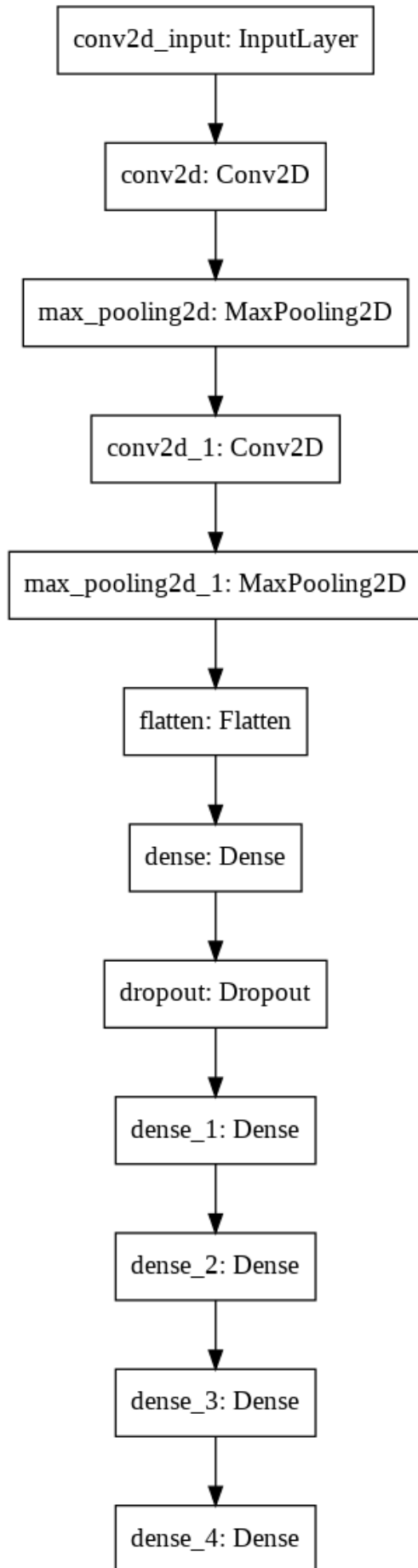


Classification Report-

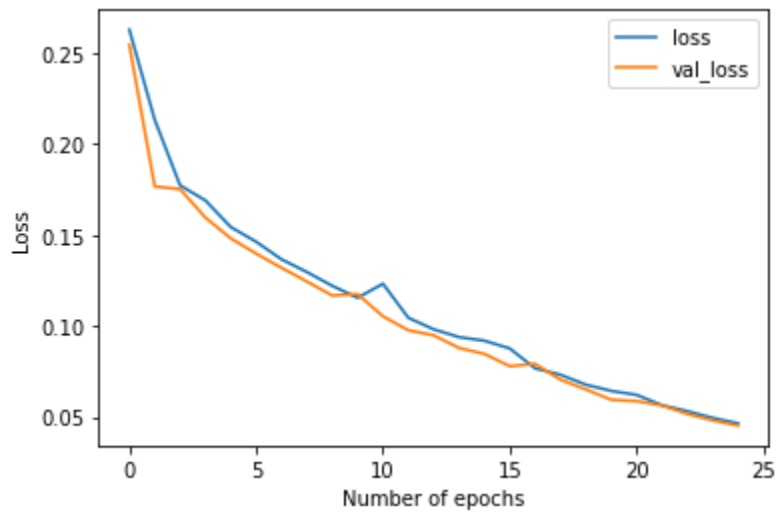
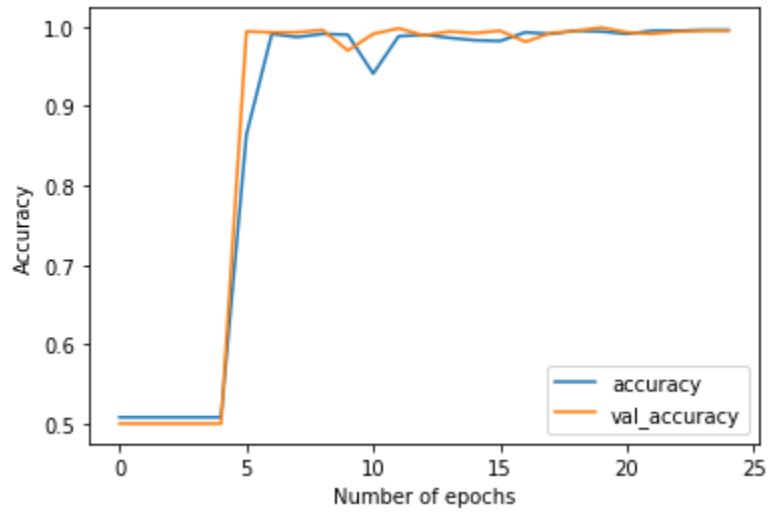
	precision	recall	f1-score	support
0	1.00	1.00	1.00	493
1	1.00	1.00	1.00	507
accuracy			1.00	1000
macro avg	1.00	1.00	1.00	1000
weighted avg	1.00	1.00	1.00	1000

4) CNN

The model-




```
cnn.fit(x_train1,y_train1,batch_size=128,epochs=25,  
        validation_data=(L, P_),  
        callbacks=[e_stop])
```



Classification Report-

	precision	recall	f1-score	support
0	1.00	0.49	0.66	1000
1	0.00	0.00	0.00	0
accuracy			0.49	1000
macro avg	0.50	0.25	0.33	1000
weighted avg	1.00	0.49	0.66	1000