Mask or No-Mask Classification

Harsh Rajiv Agarwal (B19EE036), Saptashrungi Birajdar (B19CSE076)
Github Repo: *PRML: Mini-Project*

Index Terms

Image Classification, Region of Interests, Neural Network, Random Forest, Bayes Classification, Support Vector Machine, Light Gradient Boosting Machine, Resnet, MobileNet

I. Introduction

COVID-19 pandemic has affected our daily routine, world trade, work-life, education etc. Wearing a protective face mask has become a new normal, whenever we step out of the house or even within the house sometimes. Nowadays it is mandatory to wear masks as a precautionary measure, given the rate of spread of virus. Therefore, face mask detection has become a crucial task to help global society, to warn those who do not do so. This report explains various ways to classify whether a person is wearing mask or not, using TensorFlow, Keras, OpenCV and Scikit-Learn. Classifiers like Random forest, Support Vector Machine, Gaussian Naive Bayes and Light Gradient Boosting were used and compared. Further neural networks like CNN, and pretrained CNN models like ResNet50 and MobileNet were used to detect masks.

II. Data Description and Preprocessing

The dataset contains a number of face images of people wearing masks and without masks.

Fig. 1. Faces with Mask
The images contained masks of varied colors, shapes and sizes. Commonly found types were N-95, Surgical and Cotton Mask.

Fig. 2. Faces without Mask
The dataset contained images shot from different profiles and each image has a different facial posture.

Preprocessing

The first step was to remove null elements from the dataset, although there were no missing elements. The masked images dataset contained some corrupt images that had to be discarded from the dataset. The non-masked images contained no corrupt images and hence, none of them were removed.
The preprocessed and transformed the data was used in two ways. In the first method the images were converted into flattened arrays which were then fed to the classifiers like Random Forest, Support Vector Machine, LightGBM and Gaussian NB. These images were augmented so as to equalize the proportions of masked and non-masked images and also random noises were introduced in the dataset. In the latter method, the images were scaled into (height,width,3) dimensions so as to apply varied neural networks on them.
We trained various machine learning models and deep neural networks to classify these images into mask or no mask.

III. Machine Learning Models

A. Random Forest

The training algorithm for random forests applies the general technique of bootstrap aggregating to tree learners. Given a training set X with responses Y, bagging repeatedly selects a random sample with replacement of the training set and fits trees to these samples. After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x or by taking the majority vote in the case of classification trees. This bootstrapping procedure leads to better model performance because it decreases the variance of the model,without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Random forest was fitted with data without noise where the accuracy was close to 81%. However on addition of noise to the data the accuracy reduced to 72.92% maximum cross-validation score with average 71%.

B. Gaussian Naive Bayes

Gaussian Naive Bayes was used to classify the images, in their array form. This model was also experimented with noise data, however it's accuracy reduced to 60% (for with noise dataset) from 75% (for without noise dataset). Comparitively other models performed better, though slower than Gaussian NB. For the image array, Gaussian Naive Bayes made naive assumptions which lead to its lesser accuracy.

C. Light Gradient Boosting - LGBM

LGBM classifier trained on usefulCount instead of the reviews to verify the trend and check the correlation. Remarks - Just usefulCount does not help in giving good predictions. The model gave overall accuracy of 65%, maximum accuracy 71.15% and minimum 54%. For dataset without noise it gave better accuracy, around 82%.

D. SVM

SVM was tried over multiple hyperparameter C values in the range of [10-20], howvever it did affect the accuracy much. For the dataset with noise the accuracy was close to 71%. SVM performed much better on dataset without noise, and gave accuracy 94%. It was observed to be too slow, making cross-validation time consuming to a large extent. Noise in the data, was affecting the SVM, as it sensitive to outliers if C is not optimal.

E. Analysis of Conventional Machine Learning Models

The advantages of linear regressions are their properties that make them easy to interpret ; while the weakness is that they only model linear relationships between dependent and independent variables. The strengths of gradient boosting solve linear regression's weakness. Gradient boosting can optimize on different loss functions and provides several hyperparameters tuning options that make the function fit very flexible. Where as the weaknesses of gradient boosting are computationally expensive and less interpretable. Random Forest despite being quite robust did not provide very accurate results. Naive Bayes despite the data not being scaled or gaussian in any way, with uneven distribution of ratings gave a fair accuracy on the test data.

IV. Deep Learning Models

A. Convolutional Neural Network

A convolutional neural network was defined using keras layers. After one layer of convolution, followed by BatchNormalization, Maxpooling and Dropout of 0.25. Batch normalization applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. The Dropout layer randomly sets input units to 0 at each step during training time, which helps prevent overfitting. The activation function for all layers was softmax(non linear) except the last hidden layer which utilized sigmoid activation function. It achieved a validation accuracy of 96.79%, with training accuracy of 98.73% and 0.5937 loss. The loss function used was binary crossentropy as it gave good results. To increase training speed the optimiser used was Adam instead of SGD.

Convolutional Neural Network

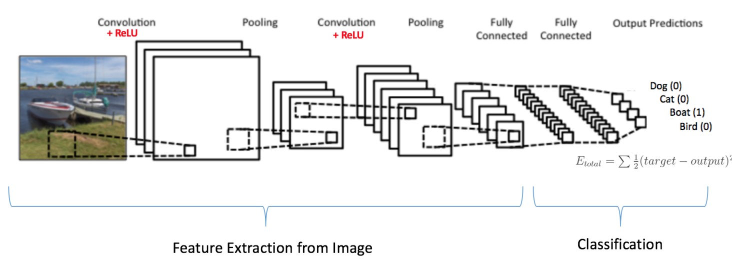


Fig. 3. CNN

B. ResNet50 - Residual Networks

Residual can be simply understood as subtraction of feature learned from input of that layer. ResNet does this using shortcut connections (directly connecting input of nth layer to some (n+x)th layer. It has proved that training this form of networks is easier than training simple deep convolutional neural networks and also the problem of degrading accuracy is resolved.

ResNet50

This is the fundamental concept of ResNet.

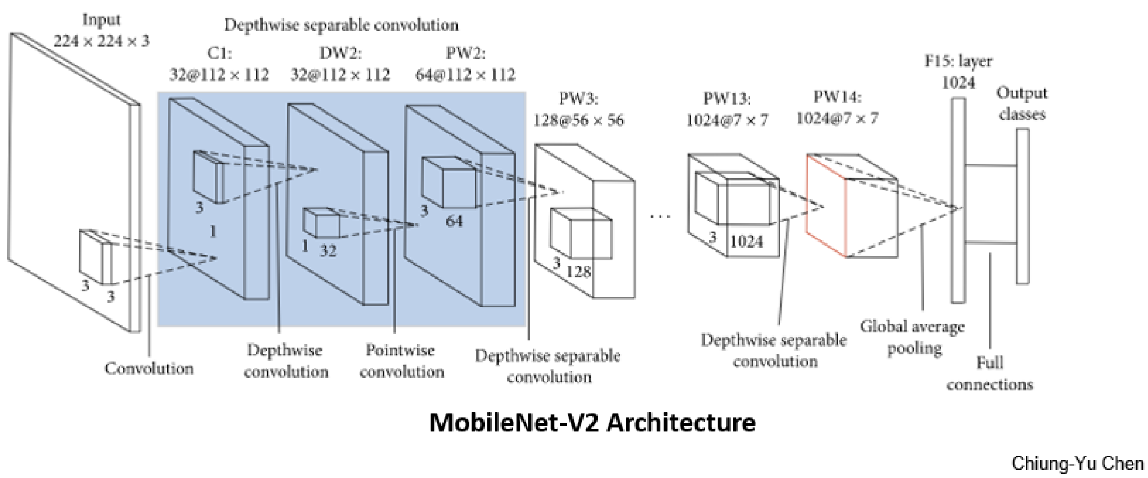
ResNet50 is a 50 layer Residual Network. There are other variants like ResNet101 and ResNet152 also. As we know, without adjustments, deep networks often suffer from vanishing gradients, that is, as the model backpropagates, the gradient gets smaller and smaller. ResNet50 gave maximum accuracy of 94% while training, however after 10 epochs, the accuracy varied between 91%-95%. It was very slow compared to other two deep-learning models, however it gave better accuracy

Fig. 4. ResNet50

compared to mobilenet and some ML classifiers like Gaussian naive Bayes or SVM even in the presence of noise in the data. This holds for CNN and MobileNet too, as their accuracy was also not much affected by noise.

C. MobileNet

MobileNet is an efficient and portable CNN architecture that is used in real world applications. MobileNets primarily use depthwise seperable convolutions in place of the standard convolutions used in earlier architectures to build lighter models. MobileNets introduce two new global hyperparameters(width multiplier and resolution multiplier) that allow model developers to trade off latency or accuracy for speed and low size depending on their requirements.A standard MobileNet has 4.2 million parameters which can be further reduced by tuning the width multiplier hyperparameter appropriately. The size of the input image is $224 \times 224 \times 3$.



[MobileNet]

While fitting the model of the given dataset it was observed to attain an accuracy of 81.5% in 20 epochs. Hence the number of epochs were increased to 50, and accuracy reached 91.55% which is much better. Also it is faster than ResNet50, however ResNet50 reached accuracy of 81+ % in 5 epochs while MobileNet took 20 epochs. Therefore MobileNet was faster though it required a few more epochs.

V. Classifier Comparision

We compare the 4 ML (Random forest, SVM, Gaussian NB, LightGBM) models with 3 DL models. For the 4 ML classifiers along with accuracy, measures like precision, recall, F1-score were compared. The comparision is shown below:

Type	Classifier	Accuracy%	Precision	Recall	F1-Score
ML	Random Forest	72.92	0.68	0.80	0.74
	Gaussian Naive Bayes		43	43	439
	LightGBM	66.3	43	43	43
	SVM	79.235	43	43	43

Type	NN Model	Accuracy%	Val-Accuracy	Epochs	Loss
DL	CNN	98.73	96.79	50	0.5937
	MobileNet	91.55	91.85	50	0.2699
	ResNet	93.5%	90%	20	0.19

Comparison of accuracies

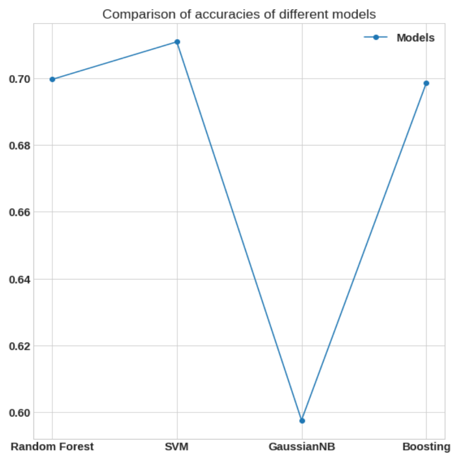


Fig. 5. The comparison of accuracies of various ML models on Noisy dataset.

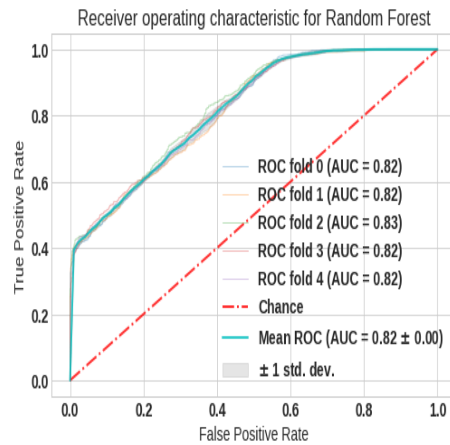


Fig. 6. ROC for Random Forest on cross-validation

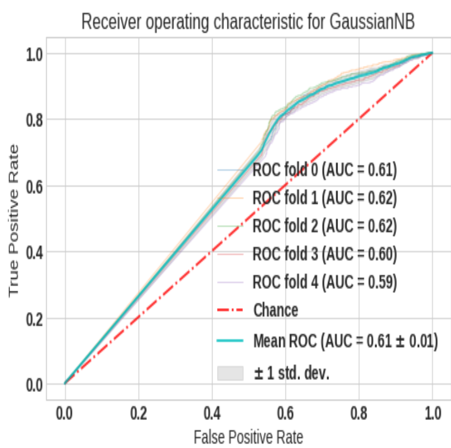


Fig. 7. ROC for Gaussian Naive Bayes on cross-validation

VI. Conclusion

In this project, classification was performed on the image dataset of faces with and without mask using various conventional and deep learning models to evaluate their performance. It was observed that the deep learning models performed better and were not affected much by noise or outliers. The group images of people with or without masks were also considered. In such cases, the ROI i.e. the Region Of Interest was found, cropped and classified using the models which trained on individual face dataset. Hence the models were generalised and can tackle group images or live video feed too.

Contribution

- Harsh Rajiv Agarwal(B19EE036) - Worked on iterating different preprocessing aspects and best in-depth exploratory data analysis. He implemented models of MobileNet alongwith all the ML classifiers. He analysed the classifiers, plotting ROCs and finding metrics like Precision, Recall and F1-Score. He also analyzed the models on Real World Dataset.
- Saptashrungi Birajdar (B19CSE076) - Worked on exploratory data analysis. He worked extensively on configuring CNN model and finding appropriate hyperparameters for it. She also worked on ResNet50, a pretrained CNN model, and analysed it's performance compared to other DL models.