Mask vs No-Mask Classification

Harish Kumar *(B19CSE035)* and Jayesh Khandelwal *(B19EE040)*

Abstract

Suppose that we are given an image of an person and we want to find out if the person in the image is wearing mask or not. For a human this a very easy job to accomplish but the question is can we achieve the same through an automated system. That what we tried to accomplish in here. We tried various Machine Learning and Deep Learning Models and compared them to give a insight on which algorithm fits best for the job. We aslo applied the same models on standardized data. We further did dimensionality reduction to check the gains and losses in time and efficiency respectively. The results of all the comparisons and information regarding the dataset is also mentioned.

I. Introduction

A. Dataset

So, before talking about what we have done. Lets, first talk about the used dataset we have used for all the analysis. We have used a dataset from the given resource Link to the dataset. The dataset has images of human faces which appear to be real but they aren't, they are generated by Deep Learning Model. Each of these faces was generated by a GAN.

B. Preprocessing

.

For Normal Classification models like RF, KNN, MLP, SVM we used the latent vectors as a set of features and Mask , No-Mask as the target variable 1 , 0. The latent vectors are genetared by extrating the features from the given image dataset.The total dataset used has 4406 images(2203 for mask and 2203 for no-mask). We make the sets, Training (2203 images/vectors) , Validation (1101 images/vectors) , Testing (1102 images/vectors). All three sets have their natural use.

For CNN , Images were used and they were Normalised and were divided into batches of size 32 with each reducing pixels to [28,28].Using **ImageDataGenerator**, Images were sheared, zoomed and rotated to obtain more accuracy to the model .

C. Models Trained

So, now we look at the various models we have implemented and the results obtained with their help. Later we will compare each result.

*1) **Random Forest (RF)**:* .

Random forest classifiers fall under the broad umbrella of ensemble based learning methods. They are simple to implement, fast in operation, and have proven to be extremely successful in a variety of domains. The key principle underlying the random forest approach comprises the construction of many "simple" decision trees in the training stage and the majority vote (mode) across them in the classification stage. Among other benefits, this voting strategy has the effect of correcting for the undesirable property of decision trees to overfit training data.

(a) Mask

(b) No Mask

Fig. 1: Images in the Dataset

In the training stage, random forests apply the general technique known as bagging to individual trees in the ensemble. Bagging repeatedly selects a random sample with replacement from the training set and fits trees to these samples. Each tree is grown without any pruning. The number of trees in the ensemble is a free parameter which is readily learned automatically using the so-called out-of-bag error. .

2) *Multi-layer Perceptron(MLP):* .

A multilayer perceptron (MLP) is a deep, artificial neural network. It is composed of more than one perceptron. They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP. MLPs with one hidden layer are capable of approximating any continuous function.

Multi-layer perceptrons are often applied to supervised learning problems: they train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs. Training involves adjusting the parameters, or the weights and biases, of the model in order to minimize error. Back propagation is used to make those weigh and bias adjustments relative to the error, and the error itself can be measured in a variety of ways, including by root mean squared error (RMSE).

Feed-forward networks such as MLPs are like tennis, or ping pong. They are mainly involved in two motions, a constant back and forth. You can think of this ping pong of guesses and answers as a kind of accelerated science, since each guess is a test of what we think we know, and each response is feedback letting us know how wrong we are.

In the forward pass, the signal flow moves from the input layer through the hidden layers to the output layer, and the decision of the output layer is measured against the ground truth labels.

In the backward pass, using back propagation and the chain rule of calculus, partial derivatives of the error function w.r.t. the various weights and biases are back-propagated through the MLP. That act of differentiation gives us a gradient, or a landscape of error, along which the parameters may be adjusted as they move the MLP one step closer to the error minimum. This can be done with any gradient-based optimisation algorithm such as stochastic gradient descent. The network keeps playing that game of tennis until the error can go no lower. This state is known as convergence.

3) *K Nearest Neighbors (KNN):* .

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point. Technique is non-parametric , it means that it does not make any assumptions on the underlying data distribution. KNN is also a lazy algorithm (as opposed to an eager algorithm).which means, there is no explicit training phase or it is very minimal. This also means that the training phase is pretty fast . Lack of generalization means that KNN keeps all the training data. To be more exact, all (or most) the training data is needed during the testing phase.KNN Algorithm is based on feature similarity.So, we applied KNN on our model.

4) *Support Vector Machine (SVM):* .

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.

5) *Convolutional Neural Network(CNN):* .

A convolutional neural network, or CNN, is a deep learning neural network sketched for processing structured arrays of data such as portrayals. CNN are very satisfactory at picking up on design in the input image, such as lines, gradients, circles, or even eyes and faces. This characteristic that makes convolutional neural network so robust for computer vision. CNN can run directly on a underdone image and do not need any prepossessing. A convolutional neural network is a feed forward neural network, seldom with up to 20. The strength of a convolutional neural network comes from a particular kind of layer called the convolutional layer. CNN contains many convolutional layers assembled on top of each other, each one competent of recognizing more sophisticated shapes. With three or four convolutional layers it is viable to recognize handwritten digits and with 25 layers it is possible to differentiate human faces. The agenda for this sphere is to activate machines to view the world as humans do, perceive it in a alike fashion and even use the knowledge for a multitude of duty such as image and video recognition, image inspection and classification, media recreation, recommendation systems, natural language processing, etc.

II. Data Standardization

The dataset is a latent vector, so we can standardize it to make the variance of every feature as 1. We used ’standarscalar’, it subtracts the mean and divide by the standard deviation for each feature.

III. Dimensionality Reduction

In machine learning classification problems, there are often too many factors on the basis of which the final classification is done. These factors are basically variables called features. The higher the number of features, the harder it gets to visualize the training set and then work on it. Sometimes, most of these features are correlated, and hence redundant. This is where dimensionality reduction algorithms come into play. Dimensionality reduction is the process of reducing the number of random variables under consideration, by obtaining a set of principal variables.

1) Linear Discriminant Analysis(LDA): .

LDA is a type of Linear combination, a mathematical process using various data items and applying a function to that site to separately analyze multiple classes of objects or items.First general steps for performing a Linear Discriminant Analysis

1) Compute the d-dimensional mean vector for the different classes from the dataset.
2) Compute the Scatter matrix (in between class and within the class scatter matrix)
3) Sort the Eigen Vector by decrease Eigen Value and choose k eigenvector with the largest eigenvalue to from a d x k dimensional matrix w (where every column represent an eigenvector)
4) Used d * k eigenvector matrix to transform the sample onto the new subspace.

IV. Results

A. Random Forest (RF)

Multi-layer Perceptron (MLP)			
Performance	Normal Dataset	Standardized Data	Dimensionality Reduction
Precision	0.89182	0.89182	0.83018
Recall	0.678244	0.678244	0.884826
F1-Score	0.770508	0.770508	0.85663
Accuracy	0.799273	0.799273	0.852861
Confusion matrix -	[[509 45] , [176 371]]	[[509 45] , [176 371]]	[[455 99] , [63 484]]
Cross validation score (CVS)	[0.828 0.787 0.763 0.827 0.827]	[0.828 0.787 0.763 0.827 0.827]	[0.850 0.886 0.840 0.904 0.840]
Mean CVS	0.806713	0.806713	0.864784
Standard Deviation CVS	0.0265787	0.026578	0.026107

B. Multi-layer Perceptron (MLP)

Multi-layer Perceptron (MLP)			
Performance	Normal Dataset	Standardized Data	Dimensionality Reduction
Precision	0.881918	0.912476	0.840909
Recall	0.873857	0.89579	0.879341
F1-Score	0.877869	0.904059	0.85969
Accuracy	0.879200	0.90554	0.85740
Confusion matrix -	[[[490 64] , [69 478]]	[[507 47] , [57 490]]	[[463 91] , [66 481]]
Cross validation score (CVS)	[0.850 0.796 0.836 0.881 0.795]	[0.886 0.904 0.881 0.890 0.859]	[0.8461 0.877 0.854 0.904 0.854]
Mean CVS	0.832139	0.884734	0.867523
Standard Deviation CVS	0.033027	0.014957	0.0213088

C. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN)			
Performance	Normal Dataset	Standardized Data	Dimensionality Reduction
Precision	0.936619	0.930394	0.846153
Recall	0.72943	0.73308	0.864716
F1-Score	0.82014	0.82004	0.85533
Accuracy	0.84105	0.840145	0.854677
Confusion matrix -	[[527 27] , [148 399]]	[[524 30] , [146 401]]	[[468 86] , [74 473]]
Cross validation score (CVS)	[0.823 0.841 0.813 0.809 0.831]	[0.805 0.837 0.804 0.795 0.836]	[0.850 0.859 0.859 0.886 0.831]
Mean CVS	0.82394	0.815779	0.857535
Standard Deviation CVS	0.011847	0.033027	0.017587

D. Support Vector machine (SVM)

Support Vector machine (SVM)			
Performance	Normal Dataset	Standardized Data	Dimensionality Reduction
Precision	0.92105	0.921200	0.846702
Recall	0.89579	0.89762	0.86837
F1-Score	0.90824	0.90925	0.857400
Accuracy	0.91008	0.91099	0.856494
Confusion matrix -	[[512 42] , [57 490]]	[[512 42] , [56 491]]	[[468 86] , [72 475]]
Cross validation score (CVS)	[0.918 0.891 0.881 0.886 0.923]	[0.918 0.891 0.881 0.881 0.918]	[0.846 0.877 0.854 0.913 0.854]
Mean CVS	0.9001727	0.898354	0.8693418
Standard Deviation CVS	0.017035	0.01671	0.0245339

E. Convolutional Neural Network

Convolutional Neural Network (CNN)			
Epoch	Loss	Accuracy	Validation Accuracy
1	0.3605	0.9356	1.0000
2	1.0283e-04	1.0000	1.0000
3	8.8778e-09	1.0000	1.0000
4	7.8197e-12	1.0000	1.0000
5	6.7781e-14	1.0000	1.0000
6	7.4703e-16	1.0000	1.0000
7	1.0728e-15	1.0000	1.0000
8	2.8077e-14	1.0000	1.0000
9	2.0256e-15	1.0000	1.0000
10	1.0059e-16	1.0000	1.0000

V. Conclusion

We can see that SVM for Standardized Data gives the best accuracy **91.09%**. We can also concluded that for image features extration, standardiztion of data does not improve the accuracy much.

Dimensionality reduction of Data doesn't improve the accuracy, rather decreases in some cases.

The CNN model gives us the best accuracy, i.e. **100%**.

VI. Contributions

- Harish Kumar(B19CSE035) - Data Loading, Data Preprocessing (CNN), Support Vector Machine (SVM) and Convolutional Neural Network(CNN)
- Jayesh Khandelwal(B19EE040) - Data Preprocessing , Multi-layer Perceptron(MLP) and K Nearest Neighbors (KNN) and Random Forest (RF)

Acknowledgment

The authors would like to thank Dr. Richa Singh, for their guidance in the entire span on course which lead to making this project successful.

References

[1] Data Preprocessing Stack overflow resources.
[2] For RF, MLP , KNN , SVM used SKlearn Documentation.
[3] For CNN used Udemy Course - Deep Learning AtoZ.