Glass vs No-Glass Classification

Gourav Kumar *(B19CSE034),* Harshit Gupta *(B19CSE108),* 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 glasses 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 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 a course on Deep Learning in Washington University. Link to the dataset. The dataset has images of humans 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. The vector that generates these faces is contained in 512 values labeled v1-v512. Using this latent vector the image is then produced. The image and/or GAN vector can both be used to predict if the individual is wearing glasses. Images from the data set are at 1024x1024 resolution. As a part of training data we had 4500 label images/vectors and 500 images/vectors as testing data.

B. Preprocessing

.

For Normal Classification models like KNN, MLP, SVM we used the latent vectors as a set of features and Glass , No-glass as the target variable 1 , 0. We make the sets, Training (3600 images/vectors) , Validation (900 images/vectors) , Testing (500 images/vectors). Here the validation is used to test the model accuracy and test set to predict labels.

For CNN , Images were used and they were Normalised and were divided into batches of size 32 with each reducing pixels to [64,64].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) 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.

2) 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.

3) 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.

4) 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. 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.

III. Feature Selection

Feature Selection is one of the core concepts in machine learning which hugely impacts the performance of your model. The data features that you use to train your machine learning models have a huge influence on the performance you can achieve.

Irrelevant or partially relevant features can negatively impact model performance. Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in.

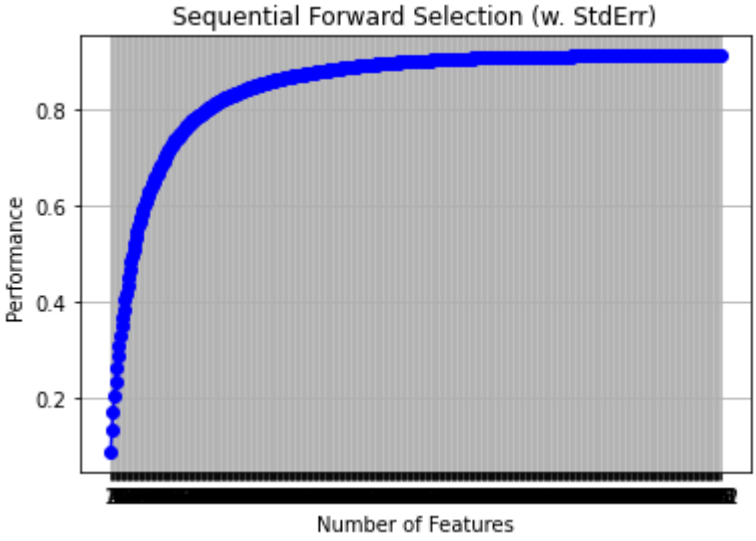
Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features. The advantges are listed below:

1) Reduces Overfitting: Less redundant data means less opportunity to make decisions based on noise.

2) Improves Accuracy: Less misleading data means modeling accuracy improves.

3) Reduces Training Time: fewer data points reduce algorithm complexity and algorithms train faster.

Applying the feature selection algorithms on our dataset and then evaluating the performance at different number of selected features resulted in the shown graph.



We can see that approximately 150 top feature (latent vectors) would result similar performance with all 512 latent vectors.

IV. Results

A. Multi-layer Perceptron (MLP)

Multi-layer Perceptron (MLP)			
Performance	Normal Dataset	Dimensionality Reduction	Feature Selection
Precision	0.984375	0.991258	0.9792746
Recall	1.0	1.0	1.0
F1-Score	0.992128	0.995610	0.9895287
Accuracy	0.99	0.994444	0.9866667
Confusion matrix	[[324 9] , [0 567]]	[[328 5] , [0 567]]	[[321 12] , [0 567]]

B. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN)			
Performance	Normal Dataset	Dimensionality Reduction	Feature Selection
Precision	0.989528	0.998239	-
Recall	1.0	1.0	-
F1-Score	0.994736	0.999118	-
Accuracy	0.9933333	0.998889	-
Confusion matrix	[[327 6] , [0 567]]	[[332 1] , [0 567]]	-

C. Support Vector machine (SVM)

Support Vector machine (SVM)			
Performance	Normal Dataset	Dimensionality Reduction	Feature Selection
Precision	0.982668	0.994736	-
Recall	1.0	1.0	-
F1-Score	0.991258	0.997361	-
Accuracy	0.9888889	0.996667	-
Confusion matrix	[[323 10] , [0 567]]	[[330 3] , [0 567]]	-

Due to the lack of the required computation power and limited resources we couldn't run the code for feature selection for KNN and SVM as it was extending over 12hours of time. Their code of these are present in the .ipynb submitted

D. Convolutional Neural Network

Convolutional Neural Network (CNN)			
Epoch	Loss	Accuracy	Validation Accuracy
1	0.5916	0.6767	0.8521
2	0.3870	0.8348	0.8509
3	0.3357	0.8506	0.8732
4	0.3250	0.8753	0.8710
5	0.3376	0.8642	0.8699
6	0.3209	0.8751	0.8788
7	0.2961	0.8800	0.8788
8	0.3028	0.8724	0.8732
9	0.2924	0.8866	0.8710
10	0.3094	0.8798	0.8799

V. Conclusion

In this project, we implemented multiple classification model on an image data set. We classified image using the image itself and its latent vector. From the results we can see that

1) model accuracy of K-Nearest Neighbors(KKN) $>$ Multi-layer Perceptron(MLP) $>$ Support Vector machine(SVM)
2) The best accuracy is of K-Nearest Neighbors (KNN) with Dimensionality Reduction (LDA) model, **99.89 %**.

Another major conclusion we can draw from this project is using CNN on images proved to be computationally costly where models like MLP, KNN , SVM all resulted in approximate 99 % accuracy with latent vector. Thus we prefer use of latent vector over images. (NOTE: Converting Image into Latent Vectors is not a computationally east task.)

VI. Contributions

- Gourav Kumar(B19CSE034) - Data Preprocessing and Multi-layer Perceptron(MLP)
- Harshit Gupta(B19CSE108) - K Nearest Neighbors (KNN) and Convolutional Neural Network(CNN):
- Jayesh Khandelwal(B19EE040) - Data Preparation and Support Vector Machine (SVM)

Acknowledgment

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

References

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