A face verification system

acp24kp

University of Sheffield

Abstract

This report on the Face Verification System explores the development of a machine learning model from the provided dataset **Labeled Faces in the Wild** to verify whether the input pair of images belong to the same or different persons. It includes the the usage of techniques like augmentation, training a classifier, feature standardization, scaling, dimensionality reduction, and hyperparameter tuning along with a detailed performance analysis on various available classifiers. Later, it evaluates the performance of the trained model using statistical performance metrics, such as confusion matrix, ROC curve etc.

1. Introduction

Development of the verification system commences with the analysis of the **Labeled Faces in the Wild** dataset. The dataset has 2200 rows, where each row represents a pair of images, flattened into array and there is a corresponding label indicating if the image pairs represent same or different individuals. Additional evaluation data is provided, which consists of 1000 rows where each row similarly represents the pairs of images and its labels.

The balanced nature of the dataset (with equal number of labels from both the classes) can be confirmed through the plot of frequency of unique values in the target/label column. The aim is to design an ML model (of joblib format) which can outperform the base model performance which has an accuracy of 56.3%. The development path will initiate from data analysis to model selection, training and evaluation. The results of the analysis will be discussed and eventually, a trained model with better predictive capacity will be achieved.

2. System Description

The provided dataset 'train.joblib' has keys 'data' and 'target', whose value correspond to features and labels respectively. A few random image pairs are plotted after reshaping into 62x47 pixels for each image. As the dataset is relatively small in size, the augmentation is necessary to be implemented to increase its size, from 2200 to 8800 rows, it can be achieved with more augmentations techniques like rotation of the pair of images by certain angles, flipping them horizontally or vertically, adding random Gaussian noise, shear transformations, performing cutout or random erasing on images, the idea is to bring variety in the dataset and increase the data multi-fold.

Once the dataset is sufficiently augmented to ensure a robust model can be generated with sufficient data, and which can generalize well (without overfitting) to slight variations in the input; preprocessing is carried out. Standardization is applied to scale the image pixels to a smaller range, making the dataset consistent and allowing algorithms to treat all features equally, preventing models from placing undue importance on features with larger values.

Although there are multiple techniques to find the minimum number of features, responsible for maximum variance in the dataset, Here, **PCA (Principal Component Analysis)** [1] is employed to reduce the dimensionality of dataset. It reduces the number of highly correlated features, removes the redundant data, it also speeds up the computation and training process. Besides this, It assists find the hidden patters in the data, which is difficult to notice when the features are large in number.

After pre-processing and feature engineering, the next action is to find the correct algorithms for the dataset. A pipeline is created which starts from image augmentation, followed by standardization, feature engineering, model selection, hyperparameter tuning and model training.

Since, there is a wide range of classification algorithms available to train, a few of them namely, KNN, Random Forest, Support Vector machine, are implemented. The augmented data is split into 75% for training and 25% for testing before training begins.

Hypertuning of the parameters is performed using the GridSearchCV/RandomisedSearchCV technique on models, accompanied by cross validation to find the model that can offer best prediction capacity on given dataset. Ultimately, the Multilayer Perceptron (MLP) classifier, which achieves the highest test accuracy, is selected for further study. Note, that the optimal value of parameters varies from one data to another.

3. Experiments

The Neural Network is selected as the final algorithm and hyper-tuned using 5-fold cross-validation to identify the best set of parameters. The table 1, shows best hyper parameters for MLP classifier. To validate the selected model, a paired t-test

Table 1: *Various model accuracy on the evaluation dataset*

MLP Parameters	Best value
max_iter	1000
solver	adam
early_stopping	True
activation_function	ReLU
PCA_n$_components$	400
hidden_layer_sizes	(800, 400, 200, 100, 50, 20)

is conducted, confirming that the Multilayer Perceptron (MLP) classifier is the most suitable choice for the system. The model architecture and layers are further optimized through hyperparameter tuning (GridSearchCV), trained, and saved, the model is pickled as **model.joblib**. Run the command below to evaluate its performance on provided training dataset (**eval1.joblib**) using script (**evaluate.py**):

```
python evaluate.py model.joblib
```

Above command gives more than 60 % accuracy on unseen data (eval1.joblib). Various metrics such as Accuracy, ROC AUC

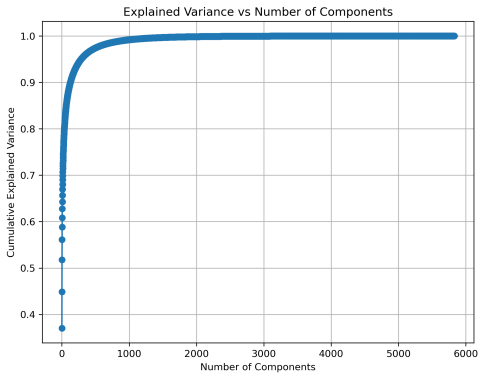

Figure 1: *The variation of cumulative variance with the number of PCA components*

score and confusion matrix are plotted. Since, an AUC score of 0.67 entitles the model with intermediate performance level [2]. The AUC score value more than 0.5 indicates the trained model performs clearly better than random guessing [2]. Other metrics FPR (False Positive Rate) and TPR (True Positive Rate) variations can be observed on the figure 3.

Paired t-test (on list of cross validation accuracies of all models) can also help to understand how the selected model is significantly different from other models. The formula to find the t-statistic value for the paired t-test is given by the equation as follows:

$$t = \frac{\bar{d}}{SE(d)} \tag{1}$$

where:

- $\bar{d}$ is mean difference,
- $SE(d)$ is the standard error of the mean difference,

4. Results and Analysis

Table 2: *Various models accuracy on the evaluation dataset*

Models	% test accuracy	% Eval accuracy
KNN	78.77	52.80
Random Forest	72.32	56.89
SVM	78.5	58.59
MLP	78.13	62.2

The trained model **model.joblib** showed a performance accuracy of nearly 78% on the test data set, but it performs differently on unseen data (eval1.joblib), where the accuracy is around 60%, During the model training, the neural network identifies the optimal parameters, suggesting that a deeper hidden layer architecture improves the performance, besides this, when the number of features is significantly reduced from 5828 to merely 500 using PCA (a dimensionality reduction technique), the accuracy improved significantly. The predictions poential of the neural network depends upon several factors, such as, the number of hidden layers, the number of neurons in them and the activation function used in each hidden layer.

The figure 2 shows a confusion matrix on the unseen data, it indicates how accurately the trained/selected model predicts the

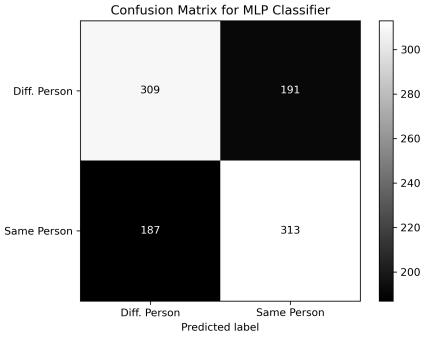

Figure 2: *The confusion matrix when the model model.joblib predicts the dataset eval1.joblib*

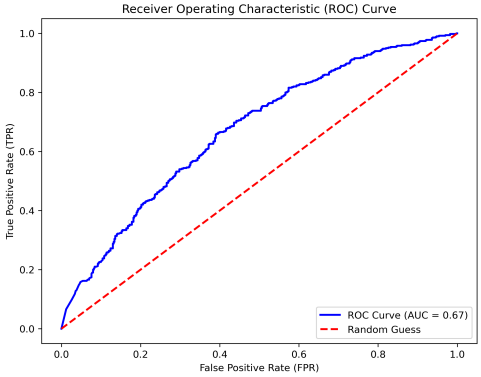

Figure 3: *The ROC AUC curve of the trained model model.joblib*

true labels. The y axis represents the true labels while the x axis indicates the predicted ones. If FP is more harmful, improve Precision, while when there is a focus on missing positive cases, the recall must be improved. It depends entirely on the use case and the criticality whether there is a improvement needed in FP or FN.

The figure 3 represents the area under the ROC, indicating how better the classification model performs. Since, the AUC score is 0.67, it validates the moderate predictive capability of the selected MLP Classifier. This graph also helps decide the threshold to favour higher TPR or to give preference to lower FPR, depends upon the requirements and the use case.

5. Discussion and Conclusions

The MLP Classifier model is developed, which performs moderately on the unseen dataset. If more time allowed, the image verification system can be designed to benefit from additional augmentation techniques and using more powerful CNNs models/pre-trained models. There is also potential to expand the practical applications of image verification in the different business sectors.

6. References

[1] P. Samuels and M. Gilchrist, "Paired samples t-test," 04 2014.

[2] A. A. H. e. a. de Hond, "Interpreting area under the receiver operating characteristic curve," *The Lancet Digital Health*, vol. 4, no. 8, pp. e853–e855, Dec. 2022.