Assignment 1: Design of Machine Learning Solution for Biometric Recognition Task

Student name: Biral Ratna Dhakhwa

ID: 2133088

Email: [biralr.dhakhwa@patancollege.edu.np](mailto:biralr.dhakhwa@patancollege.edu.np)

Unit Code: CIS007-2

Unit Coordinator: Dr Vitaly Schetinin

Table of Contents

[Table of Figures 2](#_Toc128685053)

[Introduction 3](#_Toc128685054)

[Designing solution 3](#_Toc128685055)

[Cleaning data 3](#_Toc128685056)

[Datasets 3](#_Toc128685057)

[Implementing PCA to datasets 4](#_Toc128685058)

[Classification 5](#_Toc128685059)

[Evaluation 5](#_Toc128685060)

[Tuning 5](#_Toc128685061)

[Experiments 5](#_Toc128685062)

[Multi-layer Perceptron classifier 5](#_Toc128685063)

[Decision Tree classifier 7](#_Toc128685064)

[Support Vector Machine Classifier 7](#_Toc128685065)

[Conclusion 9](#_Toc128685066)

[References 10](#_Toc128685067)

[Appendix 10](#_Toc128685068)

# Table of Figures

[Figure 1 PCA of dataset 4](#_Toc128727593)

[Figure 2Variance of each PCA 4](#_Toc128727594)

[Figure 3 Accuracy obtained by MLP 6](#_Toc128727595)

[Figure 4 Cross Validate MLP Classifier 6](#_Toc128727596)

[Figure 5 Performance of MLP Classifier 7](#_Toc128727597)

[Figure 6 Accuracy obtained by Decision Tree 8](#_Toc128727598)

[Figure 7 Cross Validate Decision Tree 8](#_Toc128727599)

[Figure 8 Accuracy obtained by SVM 8](#_Toc128727600)

[Figure 9 Cross validate SVM Classifier 9](#_Toc128727601)

[Figure 10 Performance of SVM classifier 10](#_Toc128727602)

[Figure 11 Comparison of SVM, MLP and Decision Tree classifier 11](#_Toc128727603)

# Introduction

AI stands for Artificial Intelligence. It is the science and engineering of making intelligent machines, especially intelligent computer programs. It is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable (McCarthy, 2007).

Machine Learning, or ML, is a field of research that is concerned with giving computers the ability to learn and make predictions without being explicitly programmed. ML is considered to be one of the most exciting and promising technologies of our time, as it has the potential to revolutionize the way we interact with machines and the world around us. (GeeksforGeeks, 2023).

ML has a wide range of applications, from speech and image recognition to natural language processing, predictive analytics, and more. It is used in industries such as finance, healthcare, retail, and entertainment, among others.

Facial recognition is a way of identifying or confirming an individual’s identity using their face. Facial recognition systems can be used to identify people in photos, videos, or in real-time. In order to ensure that face recognition systems can accurately identify individuals despite noise, corruptions, and variations in facial images, it is important to use robust classification systems. Some examples of such classifiers include MLP, Pairwise, CNN, and Decision Tree classifiers (J. Uglov, 2007).

# Designing solution

The first assignment included two machine learning scripts, namely "process\_yale\_images" and "classify\_yale", along with a dataset consisting of 30 images of different people captured under 50 different conditions. "process\_yale\_images" and "classify\_yale" scripts were executed in the Python programming language using Google Colab. Libraries such as sklearn, matplotlib, numpy, pandas. Initially, the images were converted into a vectorized format and then split into two files, namely "yaleExtB\_target.npy" and "yaleExtB\_data.npy".

## Cleaning data

The first step in face recognition is to clean the data from yaleExtB\_target.npy and yaleExtb\_data.npy. Cleaning data refers to the process of identifying and correcting or removing errors, inconsistencies, and inaccuracies in datasets. This can include tasks such as removing repeated data, filling in missing values etc.

## Datasets

Once the data gets cleaned, it is divided into two distinct datasets: training data and testing data. The split between these two datasets is made with a ratio of 33% for the testing data and 67% for the training data according to the test size of 0.33. The data was split using the train\_test\_split function in python.

## Implementing PCA to datasets

PCA is a technique for reducing the dimensionality of the data while retaining the most important information. By applying PCA to the facial images, the high-dimensional image data is transformed into a lower-dimensional space that captures the most important variations between the images. This lower-dimensional feature representation is then used as input to the MLP classifier to train and test the face recognition system (Cadima, 2016).

Text

Description automatically generated

Figure PCA of dataset

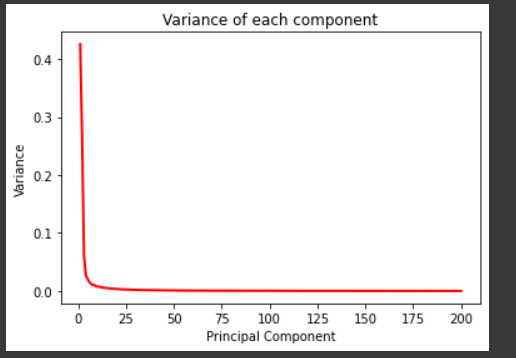


Figure Variance of each PCA

The plot shows that the first few principal components capture the majority of the variance in the data, while the later components capture less and less variance. In this case, it appears that the first 10 principal components capture a significant amount of the variance, while the remaining components capture relatively little. This information can be used to decide how many principal components to retain in order to represent the data accurately while reducing its dimensionality.

## Classification

Classification is a crucial step in face recognition, where the extracted features from the pre-processing step are used to identify the person in the image. In this step, a machine learning model such as an MLP classifier is trained using the pre-processed and labeled data to classify each image into a specific person's identity.

## Evaluation

Once the classifier has been trained on the training data, it is necessary to evaluate its performance on new data that the model has not seen before. This is an essential step in ensuring that the model can generalize well to new data and is not overfitting to the training data.

One common approach to evaluate the performance of the classifier is to use a technique called cross-validation. In cross-validation, the data is divided into multiple subsets or folds, and the model is trained and tested on different combinations of these subsets. This helps to ensure that the model is not biased towards a specific subset of the data and provides a more robust estimate of the model's performance.

## Tuning

If the performance of a classifier is not satisfactory, it is important to adjust the parameters of the classifier to improve its accuracy. The parameters for MLP that can be tuned include the number of hidden nodes, the solver, the size of the hidden layer, the activation function, and the batch size.

It is important to understand the role of each parameter and how it affects the classifier. For example, increasing the hidden node can increase the performance of the model but can lead to overfitting, which leads to poor generalization on new data. Similarly choosing the right solver is important as different solver have different weakness and strengths. This process is repeated until a satisfying result is met.

# Experiments

## Multi-layer Perceptron classifier

The Multi-layer Perceptron classifier(MLP) is a type of artificial neural network that is commonly used for classification tasks in machine learning. A MLP is a type of neural network that includes an input layer, one or more hidden layers, and an output layer. The nodes or neurons in each layer are connected to the nodes in the adjacent layers by weighted connections, and during training, the weights of these connections are adjusted using the backpropagation algorithm to optimize the performance of the model.

The backpropagation algorithm involves calculating the error between the predicted output and the true output, and then propagating this error backwards through the network to adjust the weights of the connections. This process is repeated iteratively over multiple epochs until the network reaches convergence or a stopping criterion is met. The goal of this process is to train the MLP to learn complex patterns in the input data and make accurate predictions on new data (Zola, 2022). To prepare data for training and testing the MLP classifier, Principal Component Analysis (PCA) was used.

The overall performance of 0.95 in MLP face recognition was achieved by using specific parameter values, including 200 hidden nodes, the "lbfgs" solver, the "tanh" activation function, a batch size of 256, and setting the verbose and early stopping parameters to true.

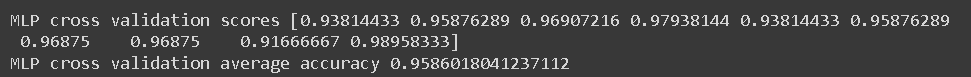


Figure Accuracy obtained by MLP

The optimal results were obtained using these parameters because the “lbfgs” solver is a quasi-Network method that provides fast convergence for small datasets. The “tanh” Activation function is good for MLP classifiers as it can produce non-linear decision boundaries, allowing the classifier to learn complex relationships between input features and target classes.

A batch size of 256 is a good compromise between computational efficiency and accuracy, as it balances the benefits of mini-batch training with the benefits of full-batch training. Verbose helps in more detailed output during training while early stop helps to prevent overfitting by stopping the training when the performance is marginal or degrades.

Chart, line chart

Description automatically generated

Figure Cross Validate MLP Classifier

The depicted graph illustrates the performance of the MLP classifier using 10-fold cross-validation. Initially, the accuracy increased steadily until it reached its peak of 0.98 in the third fold. However, the accuracy decreased in the fourth fold to 0.94. In folds 5 to 7, the accuracy remained stable at around 0.96 to 0.97. In the eighth fold, there was a sharp decline in accuracy to 0.92, while the ninth fold achieved the highest accuracy of 0.99.

A picture containing table

Description automatically generated

Figure Performance of MLP Classifier

## Decision Tree classifier

Decision Tree classifier belongs to a family of supervised learning algorithms, unlike other supervised learning algorithm it can be used for solving regression and classification problems (Chauhan, 2022). In classification, the decision tree is used to predict the class label of a new data point. In regression, the decision tree is used to predict a numerical value for a new data point.

Highest accuracy achieved overall was 0.55. this was obtained by setting the random state to 55.

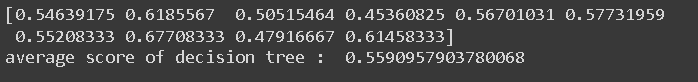


Figure Accuracy obtained by Decision Tree

Chart, line chart

Description automatically generated

Figure Cross Validate Decision Tree

The graph represents the cross-validation performance of the Decision Tree Classifier, where the accuracy is plotted against the different folds. The accuracy initially increased to 0.61 in the first fold, but then dropped to 0.54 in the third fold. However, the accuracy showed a steady increase from the fourth to the seventh fold, reaching the highest accuracy of 0.67. Subsequently, the accuracy decreased to 0.47 in the eighth fold, and increased again to 0.61 in the ninth fold.

## Support Vector Machine Classifier

SVM also known as Support Vector Machine Classifier is a type of machine language that can be used to analyze and classify data. It works by finding the best possible decision boundary (or hyperplane) that separates data points of different classes in a high-dimensional space (Kumar, 2022).

Using the SVM classifier the average accuracy was 0.95. it was obtained by using the following parameters, kernel 'linear', C1 and gamma 'scale'.

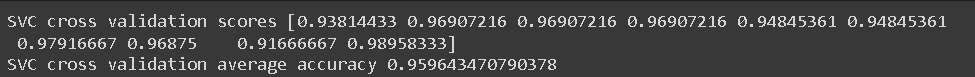


Figure Accuracy obtained by SVM

The 'linear' kernel was chosen as it simplifies complex problems like face recognition by creating a linear decision boundary. The parameter 'C' was used to balance between maximizing the margin and minimizing classification error, and C1 was found to be the best value for this dataset.

Chart, line chart

Description automatically generated

Figure Cross validate SVM Classifier

The graph depicts the performance of SVM classifier with 10-fold cross-validation. The accuracy remains steady at 0.97 from the first to the third fold. It declines to 0.95 in the fourth and fifth folds but increases to 0.98 in the sixth fold. However, it then declines and reaches its lowest point of 0.91 in the eighth fold. The accuracy is highest in the ninth fold, reaching 0.989.

A picture containing table

Description automatically generated

Figure Performance of SVM classifier

# Conclusion

A face recognition solution was created that utilizes different Machine Learning techniques. First, the provided datasets were transformed into vectors using PCA. Then, MLP, SVM, and Decision Tree classifiers were used to classify different images. Cross-validation was employed to train and test the classifiers, resulting in the final results. The highest average accuracy was obtained using the MLP classifier.

Chart, line chart

Description automatically generated

Figure Comparison of SVM, MLP and Decision Tree classifier

Upon comparing the performance of all three classifiers, it was noted that both MLP(Orange) and SVM(Blue) exhibited comparable levels of accuracy, whereas the performance of Decision Tree(Green) was considerably lower than the other two. SVM and MLP are better in this case because, they are highly flexible and powerful classifier while handling complex and high dimensional data, such as images. SVM is a powerful classifier that can handle both linear and nonlinear data by finding the optimal hyperplane that can maximize the margin between classes. MLP on the other hand, is a versatile neural network that can learn complex patterns and features in the data through multiple layers of neurons. In contrast, Decision Tree may not be as effective in handling complex data because it constructs a tree like model which may not capture the full complexity of the data. Additionally, Decision Tree is more susceptible to overfitting and may not generalize well to new data and it only used the parameter random state. To summarize, the SVM classifier achieved a slightly higher accuracy compared to the MLP classifier with an average accuracy of 0.959 and 0.958 respectively.

# References

Cadima, I. T. J. a. J., 2016. *Principal component analysis: a review and recent developments.* [Online]   
Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4792409/  
[Accessed 28 FEB 2023].

Chauhan, N. S., 2022. *Decision Tree Algorithm, Explained.* [Online]   
Available at: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html  
[Accessed 1 Mar 2023].

GeeksforGeeks, 2023. *Machine Language.* [Online]   
Available at: https://www.geeksforgeeks.org/machine-learning/  
[Accessed 2 Mar 2023].

J. Uglov, L. J. V. S. a. C. M., 2007. EURASIP Journal on Advances in Signal Processing. *Comparing Robustness of Pairwise and Multiclass,* 2008(16 june 2007), p. 7.

Kumar, A., 2022. *SVM Classifier using Sklearn: Code Examples.* [Online]   
Available at: https://vitalflux.com/svm-classifier-scikit-learn-code-examples/#:~:text=SVC%2C%20or%20Support%20Vector%20Classifier,the%20data%20into%20two%20classes.  
[Accessed 1 Mar 2023].

McCarthy, J., 2007. WHAT IS ARTIFICIAL INTELLIGENCE?. *WHAT IS ARTIFICIAL INTELLIGENCE?,* p. 15.

Zola, A., 2022. *backpropagation algorithm.* [Online]   
Available at: https://www.techtarget.com/searchenterpriseai/definition/backpropagation-algorithm#:~:text=Backpropagation%2C%20or%20backward%20propagation%20of,data%20mining%20and%20machine%20learning.  
[Accessed 28 FEB 2023].

# Appendix

Code and Video Link:

<https://drive.google.com/drive/folders/1ZehKRjbUIe9J5fyiOlSj9ouVTGuBK2dd?usp=share_link>