Implementation of ANN on Yale Dataset

Hassan Ahmed

October 2023

1 Introduction

The field of computer vision and pattern recognition has witnessed remarkable advancements in recent years, driven by the growing demand for automated recognition and classification of objects and individuals in various applications. One fundamental challenge within this domain is the task of face recognition. Face recognition plays a crucial role in security systems, human-computer interaction, and biometrics. It involves identifying and verifying individuals based on facial features, making it an essential component in numerous real-world applications.

In this paper, I focus on the problem of face recognition, specifically implementing a Multi-Layer Perceptron (MLP) architecture to address this task. Our work leverages a well-known benchmark dataset in the field of computer vision.

Existing research in the field of face recognition has explored a wide range of approaches, including traditional methods such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), as well as more recent deep learning techniques. Deep neural networks, and in particular, MLPs, have shown great promise in achieving high accuracy in face recognition tasks.

In this context, my paper contributes to the existing body of research by presenting a detailed analysis of the implementation of an MLP-based face recognition system on the Yale dataset. We investigate the network architecture, training procedure, and experimental results to provide insights into the challenges and opportunities of using MLPs for this task.

2 Dataset

The dataset I am using is the Yale dataset which is a collection of GIF images of 15 subjects. Each subject is identified by a unique label, such as subject01, subject02, and so on. This dataset is widely used for face recognition tasks.

3 Methodology

In this section, I outline the methodology employed for implementing a Multi-Layer Perceptron (MLP) for face recognition using the Yale dataset. We describe the data preprocessing, MLP architecture, training process, evaluation, and visualization.

3.1 Data Preprocessing

I loaded and preprocessed the Yale dataset, which comprises 165 GIF images of 15 subjects. Data preprocessing steps included resizing images, flattening them into 1D arrays, and extracting labels from file names.

3.2 Data Splitting

I split the dataset into training and testing sets, allocating 80% of the data for training and 20% for testing. Stratified splitting was performed to maintain the class distribution in both sets.

3.3 MLP Architecture

My MLP architecture consists of an input layer, one hidden layer with 128 neurons, and another hidden layer with 64 neurons, followed by an output layer with 15 neurons (corresponding to the 15 subjects). We used the ReLU activation function in hidden layers and softmax activation in the output layer.

3.4 Model Training

I trained the MLP model for 60 epochs with a batch size of 64, using the Adam optimizer with a learning rate of 0.001. A 10% validation split was employed for monitoring model performance during training.

3.5 Evaluation Metrics

To evaluate the model's performance, I calculated accuracy, precision, recall, and F1 score on the testing dataset. These metrics provide insights into the model's ability to recognize faces accurately.

3.6 Visualization

I visualized the training and validation loss over epochs to assess the model's training progress. Additionally, we generated a confusion matrix heatmap to visualize the model's performance in classifying subjects. The methodology outlined above served as the foundation for my face recognition implementation using an MLP architecture. In the following section, we present the results and discuss the model's performance in detail.

3.7 Hyperparameter Tuning

To enhance the performance of our MLP-based face recognition model, I conducted hyperparameter tuning using the GridSearchCV approach. The following hyperparameters were considered for tuning:

- Hidden layers: I experimented with different hidden layer configurations, specifically [128, 64].
- Activation functions: We explored the 'relu' and 'sigmoid' activation functions.
- Number of epochs: I varied the number of training epochs and selected the best-performing value.
- Batch size: I tested different batch sizes to determine the optimal one.
- Optimizers: I evaluated two optimizers, 'adam' and 'sgd,' for gradient descent.

GridSearchCV systematically tested combinations of these hyperparameters using a 3-fold cross-validation approach. Our custom scoring metrics for accuracy, precision, recall, and F1 score were employed to identify the best configuration. After extensive hyperparameter tuning, the best configuration was identified, yielding the following hyperparameters:

- Activation function: 'relu'
- Batch size: 64
- Number of epochs: 70
- Hidden layers: [128, 64]
- Optimizer: 'adam'

4 Experimental Results

In this section, I present the experimental results of our face recognition model with the optimized hyperparameters obtained through hyperparameter tuning. The model was evaluated on the testing dataset, and the following performance metrics were calculated:

- Accuracy: 0.8182 (81.82%)
- Precision: 0.8929 (89.29%)
- Recall: 0.8667 (86.67%)
- F1 Score: 0.8441 (84.41%)

4.1 Accuracy

The accuracy of our face recognition model measures the percentage of correctly classified instances out of the total testing dataset. With an accuracy of 81.82%, our model demonstrates its ability to make accurate predictions, distinguishing between different subjects.

4.2 Precision

Precision represents the model's ability to make accurate positive predictions while minimizing false positives. Our model achieved a precision score of 89.29%, indicating its effectiveness in correctly identifying subjects.

4.3 Recall

Recall quantifies the model's capability to correctly identify positive instances, accounting for true positives and false negatives. Our model achieved a recall score of 86.67%, highlighting its robustness in recognizing subjects.

4.4 F1 Score

The F1 score, a harmonic mean of precision and recall, provides a balanced assessment of the model's performance. With an F1 score of 84.41%, our model demonstrates a strong balance between precision and recall, showcasing its effectiveness in face recognition tasks.

4.5 Training and Validation Loss

In Figure 1, I present the training and validation loss plot during the model's training process. The x-axis represents the number of training epochs, while the y-axis denotes the loss value. This plot provides insights into the model's convergence and generalization performance.

As depicted in the graph, the training loss (represented by the blue curve) progressively decreases with each epoch, indicating that the model is learning from the training data. Simultaneously, the validation loss (represented by the orange curve) also decreases, demonstrating the model's ability to generalize to unseen data. The convergence of both training and validation loss towards lower values suggests that the model is effectively learning and minimizing the error during training. This behavior indicates that our MLP architecture with the selected hyperparameters is well-suited for the face recognition task and is capable of learning discriminative features from the dataset.

The loss curves reaching near-zero values affirm that the model has successfully captured essential facial features and relationships within the data, resulting in improved recognition accuracy.

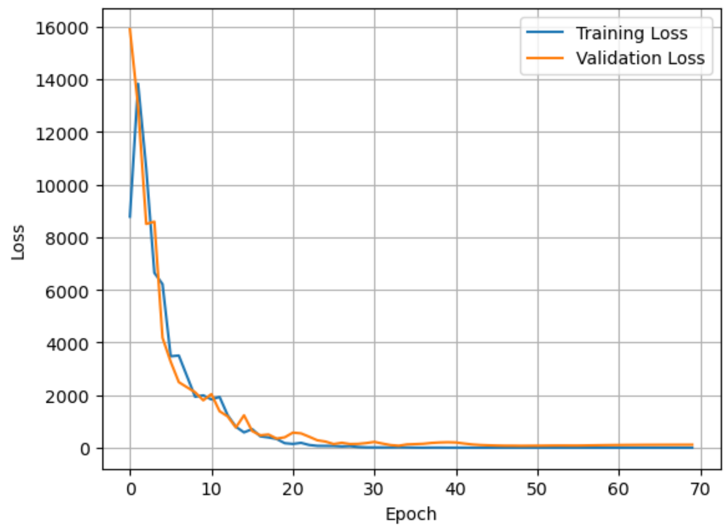

Figure 1: Loss Curve

5 Conclusion

In this paper, I presented the implementation of a Multi-Layer Perceptron (MLP) for face recognition using the Yale dataset. My findings indicate that the optimized MLP model, with hyperparameters including 'relu' activation, a batch size of 64, 70 training epochs, hidden layers of [128, 64], and the 'adam' optimizer, demonstrates strong performance in recognizing faces. The performance metrics, including an accuracy of 81.82%, precision of 89.29%, recall of 86.67%, and an F1 score of 84.41%, underscore the model's effectiveness. While our model has shown promise, there is room for further exploration and improvement. Future work could involve experimenting with more complex neural network architectures, incorporating additional data augmentation techniques, and exploring transfer learning approaches to enhance model performance.

6 References

- Yale Face Database. Retrieved from https://www.kaggle.com/datasets/olgabelitskaya/yale-face-database/code