Analysis Report Group Project – Group 6

Contents

1. Data preparation

We work with the UMIST Face Database (umist_cropped.mat), which contains cropped grayscale face images. Each image is flattened into a vector of 10 304 pixel intensities and stored in a matrix of shape (10 304, 575). Person identifiers are attached to these feature vectors to build a Pandas DataFrame that combines the image and its label.

Class balance overview:

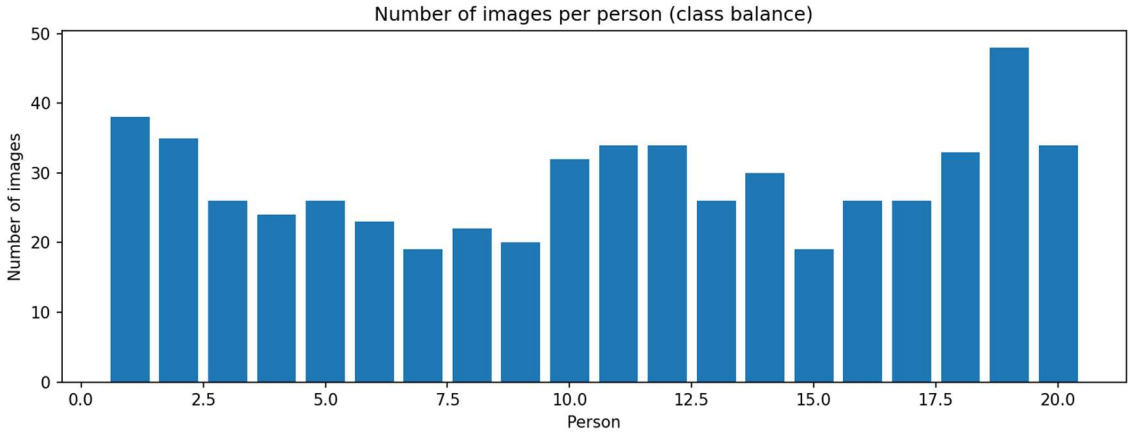
Total images: 575

Total persons: 20

Average images per person: 28.75

Standard deviation: 7.12

Minimum per person: 19, maximum: 48



2. Data splitting

To evaluate the models properly, the dataset is split into three disjoint subsets using a stratified split so that each person keeps approximately the same proportion of images in every subset. This avoids bias toward classes with more images and enables a fair evaluation.

After splitting we obtain:

Training set: 402 images

Validation set: 86 images

Test set: 87 images

Stratification is done on the person identifier so that the class distribution is preserved across the three splits.

Normalization:

Before any dimensionality reduction or neural network training, all pixel features are standardized with StandardScaler. The scaler is fit on the training data and then applied to the validation and test data. This centers each feature to zero mean and unit variance, which helps gradient based optimization converge and prevents features with larger numeric ranges from dominating the training process.

3. Dimensionality reduction

The raw feature space has 10 304 dimensions, which is large relative to the number of images and contains strongly correlated pixels. To obtain more compact and informative representations we use two unsupervised feature extractors: Principal Component Analysis and a neural network autoencoder.

3.1 Principal Component Analysis (PCA)

PCA is a linear method that projects the standardized images onto orthogonal directions that capture the maximum variance in the data. The eigenvalues indicate how much variance each principal component explains, and the cumulative explained variance curve helps decide how many components to keep.

Variance threshold: 95 percent.

We select the smallest number of principal components that explain at least 95 percent of the total variance. This results in 106 components and reduces the dimensionality from 10 304 to 106 while preserving most of the information. The summary stored in outputs/data_outputs/pca_summary.txt confirms that 95.01 percent of the variance is retained.

To visualize what these components capture, we reshape each principal component back into the original image shape and plot them as so called eigenfaces.

Figure 2: Shows a grid of the first eigenfaces.

The first components represent coarse, global patterns such as average lighting and pose, while later components capture finer person specific details.

3.2 Autoencoder

In addition to the linear PCA representation we train a fully connected autoencoder to learn a non linear compressed representation of the faces. The autoencoder receives the standardized pixel vectors as input and is trained to reconstruct them at the output.

The architecture is:

Input: 10 304 standardized pixel features

Encoder: Dense(512, activation ReLU) followed by Dense(64, linear activation) as the latent layer

Decoder: Dense(512, activation ReLU) followed by Dense(10 304, linear activation) as the output

Loss: Mean squared error between input and reconstruction

Optimizer: Adam with learning rate 0.001

Training: 20 epochs, batch size 64, with shuffled mini batches

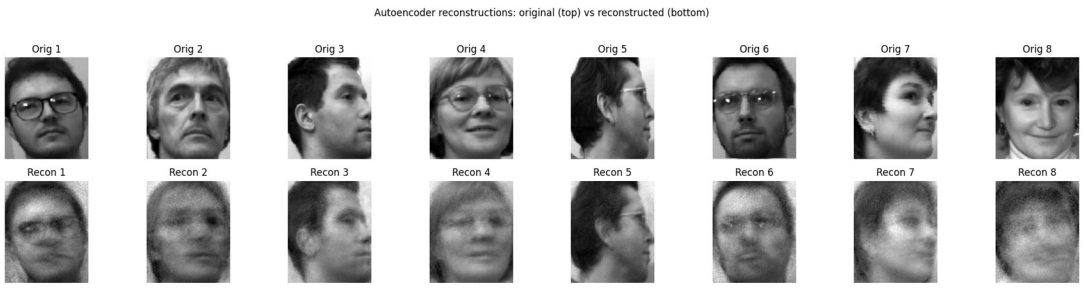
Activation functions: ReLU is used in the hidden layers because it is simple, non saturating, and works well for dense networks on standardized continuous inputs. The latent and output layers

use linear activation so that the network can freely represent positive and negative reconstruction corrections without clipping at a nonlinearity.

Loss and optimizer choice: The goal of the autoencoder is to reconstruct the original pixel values as closely as possible, so mean squared error is an appropriate reconstruction loss. Adam with learning rate 0.001 is chosen as a robust default optimizer that adapts the effective step size for each parameter and tends to work well for a wide range of deep learning problems.

Training results: the final training and validation reconstruction losses are approximately 0.097 and 0.193, as reported in outputs/data_outputs/metrics.txt. The validation loss is higher than the training loss, which is expected because the model must generalize to unseen images.

Figure 3: Illustrates the reconstruction quality by showing pairs of original and reconstructed faces.



The reconstructed faces are clearly recognizable but look smoother and slightly blurrier, which indicates that the autoencoder has learned to capture the main facial structure while discarding high frequency noise.

4. Clustering in learned feature spaces

We next explore how well unsupervised clustering can group images of the same person together. We apply three clustering algorithms on low dimensional representations and evaluate them with silhouette scores and cluster purity with respect to the true person labels. We selected KMeans and hierarchical clustering as standard, widely used methods for partitioning data in PCA space, and a Gaussian mixture model in the autoencoder latent space to capture potentially more complex cluster shapes.

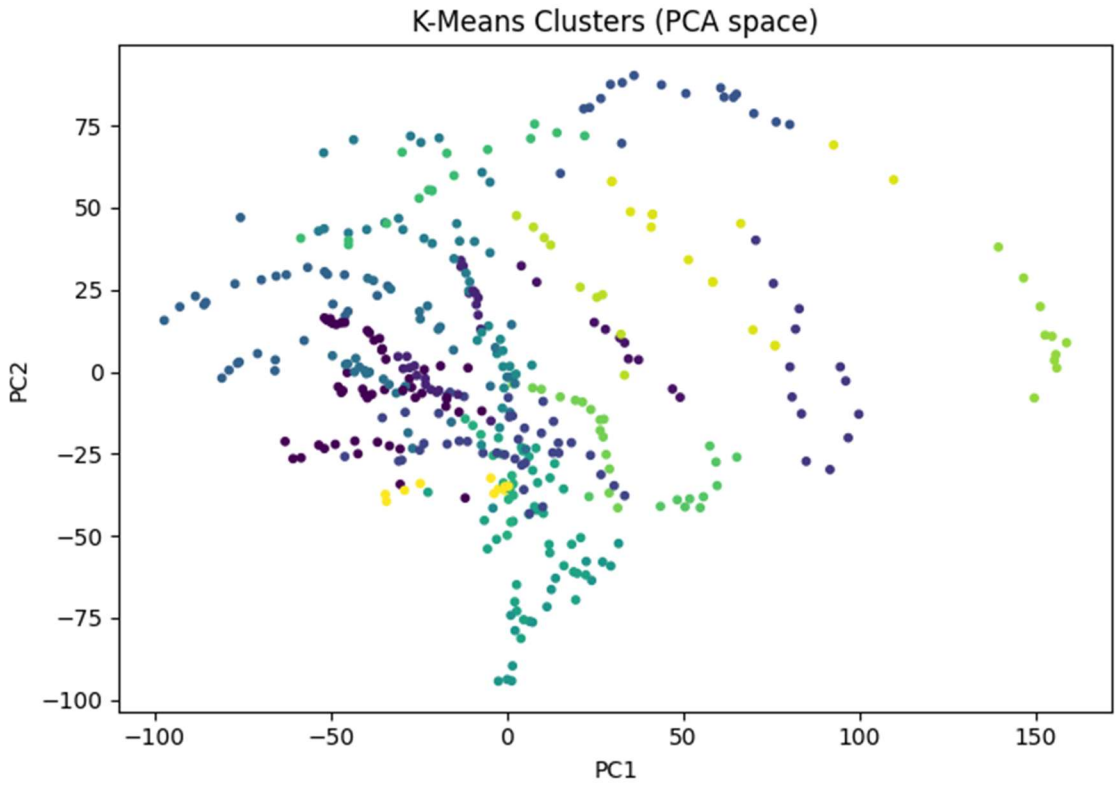
Directly clustering the original 10 304 dimensional pixel vectors is difficult because distances become less meaningful in such a high dimensional space and many features are highly correlated. By first applying PCA or the autoencoder we compress the data into a much lower dimensional representation that concentrates the important variation. This makes distance-based methods like KMeans and hierarchical clustering more stable and produces clusters that are easier to visualize and interpret

4.1 KMeans on PCA features

KMeans is applied to the PCA features with k equal to 20, which matches the number of persons. The algorithm partitions the data into 20 clusters by minimizing the within cluster sum of squared distances.

The resulting clustering obtains a silhouette score around 0.20 and an overall weighted purity around 0.54. These values show that clusters are somewhat meaningful but still overlapping, which is reasonable for a challenging face dataset with variations in pose, expression and lighting.

Figure 4: Visualizes the clusters in the first two PCA dimensions.



Images that belong to the same person tend to appear in nearby regions, although there is still noticeable mixing across some clusters.

4.2 Hierarchical clustering on PCA features

We also apply agglomerative hierarchical clustering with 20 clusters in the PCA space. This method starts with each point as its own cluster and iteratively merges the closest clusters according to a linkage criterion.

The hierarchical clustering achieves a silhouette score similar to KMeans but slightly higher overall purity, around 0.56. This suggests that, although the separation between clusters is still limited, the hierarchy can group some people slightly more coherently.

Figure 5: The hierarchical structure can be inspected in a dendrogram

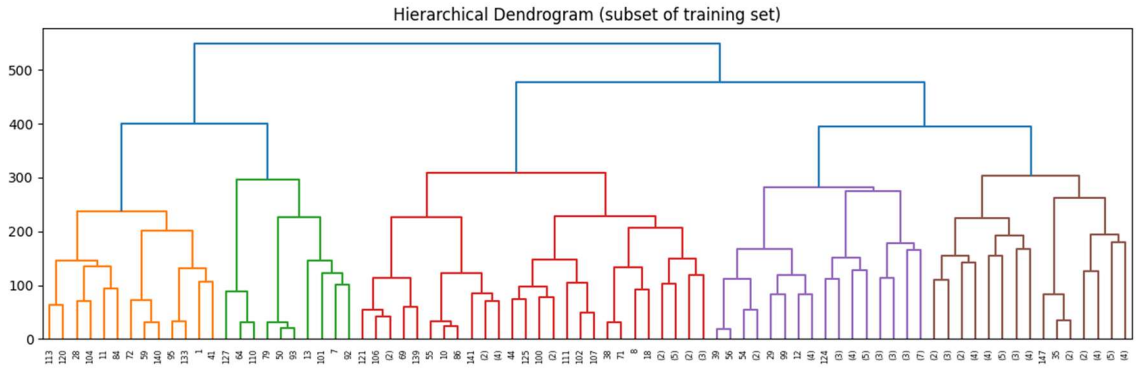
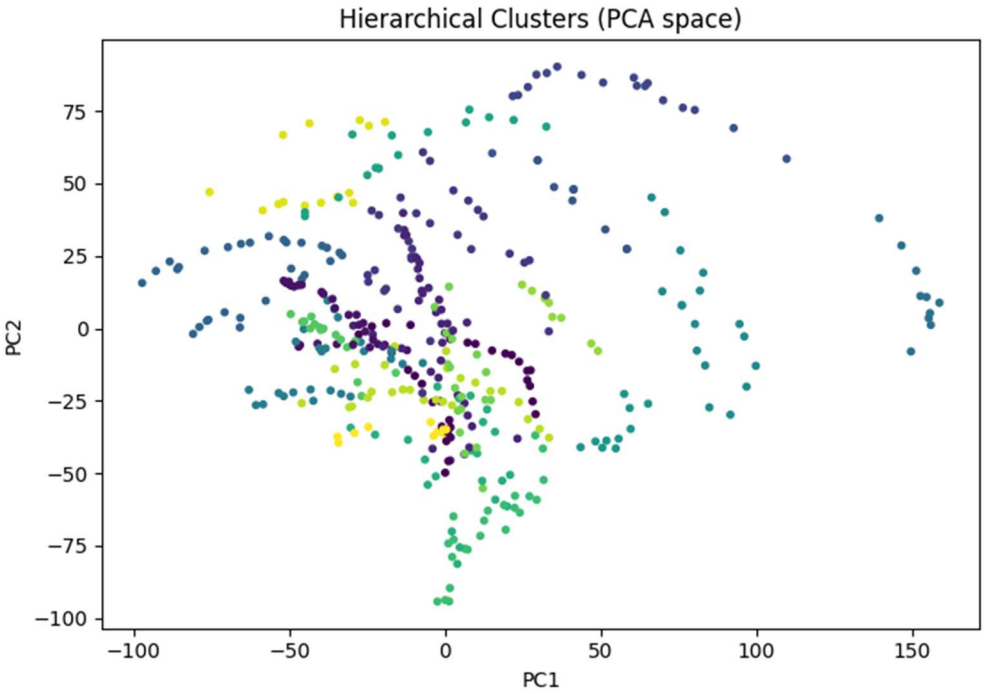


Figure 6: The final clusters can be projected into the first two PCA dimensions.



4.3 Gaussian mixture model on autoencoder latent space

To exploit the non linear representation learned by the autoencoder we fit a Gaussian mixture model with 20 components on the 64 dimensional latent vectors. This model assumes that the latent space is organized around 20 Gaussian like regions and performs soft clustering.

The GMM clustering in the latent space attains a silhouette score of roughly 0.20 and a purity around 0.50. The similarity of these scores to those obtained from PCA based clustering suggests that the autoencoder latent space has a similar degree of cluster structure with respect to the person identity.

4.4 Clustering hyperparameter choices

For KMeans and hierarchical clustering we set the number of clusters to 20 to match the number of individuals in the dataset, which allows a direct interpretation of clusters as approximate person groups. For hierarchical clustering we used Ward linkage, which tends to produce compact, spherical clusters and works well with Euclidean distances in PCA space. For the GMM we set the number of components to 20 for the same reason. We did not perform an extensive hyperparameter search over k or the number of components; instead, we focused on comparing these consistent settings across the three methods

5. Classification with neural networks

The final task is to classify each image into one of the 20 persons. To do this we train neural network classifiers on top of the unsupervised features. We consider two feature sets: PCA only and PCA combined with KMeans cluster labels as additional features.

5.1 Feature design

The PCA only representation uses the 106 principal component scores as input to the classifier. For the PCA plus clusters representation we build a one hot encoding of the KMeans cluster labels with 20 dimensions and concatenate it to the PCA features. This yields an augmented feature vector that includes both continuous PCA information and a discrete indication of which cluster the image belongs to.

We did not include distances to cluster centroids as additional features in order to keep the feature vector compact and to avoid overemphasizing the KMeans geometry. Instead, we focus on comparing a simple PCA baseline against the PCA plus one hot cluster labels variant.

5.2 Network architecture and training strategy

Both classifiers share the same fully connected architecture:

Input: PCA features alone or PCA plus one hot KMeans cluster labels

Hidden layer 1: Dense(256, activation ReLU)

Dropout layer with dropout rate 0.3

Hidden layer 2: Dense(128, activation ReLU)

Dropout layer with dropout rate 0.3

Output layer: Dense(20, activation softmax)

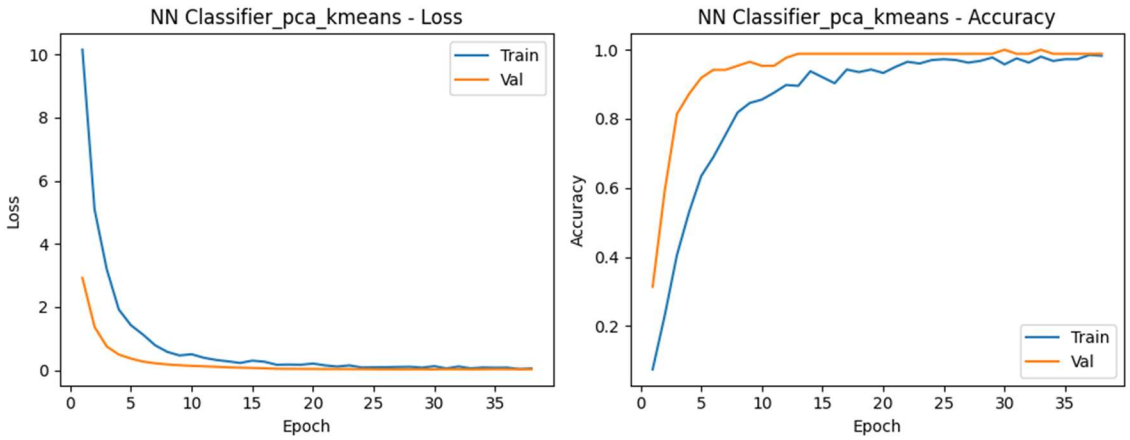
Activation functions: ReLU is used in the hidden layers because it encourages sparse activations and avoids the saturation issues of sigmoids or tanh. The output layer uses softmax because it converts the 20 outputs into a probability distribution over the 20 classes and is the standard choice for multi class classification.

Loss and optimizer: The classifiers are trained with sparse categorical cross entropy, which is appropriate when the ground truth labels are integer encoded class indices. We keep using the Adam optimizer with learning rate 0.001, since it provided stable and fast convergence during preliminary experiments on this dataset.

Training strategy: We train for up to 50 epochs with batch size 64 and use early stopping on the validation loss with patience 5 while restoring the best weights. This strategy limits overfitting and automatically selects an effective number of epochs depending on the model and features. Dropout with rate 0.3 after each hidden layer acts as regularization by randomly zeroing a

fraction of activations during training, which helps the network generalize better to unseen images.

Figure 7: Shows the training and validation loss and accuracy curves for the PCA plus KMeans classifier.



The curves indicate that the model quickly reaches high accuracy and that the validation and training curves stay close, which suggests limited overfitting.

5.3 Hyperparameter tuning

We performed a small hyperparameter search for the classifier using the PCA plus KMeans feature set. The following configurations were compared:

- hp1: learning rate 0.001, dropout 0.3, hidden units [256, 128]
- hp2: learning rate 0.0005, dropout 0.4, hidden units [512, 256]
- hp3: learning rate 0.001, dropout 0.5, hidden units [256, 64]

For each configuration we trained a classifier with the same training strategy and evaluated it on the validation set. The metrics are recorded in outputs/data_outputs/metrics.txt under the section Classifier Hyperparameter Evaluation (PCA + KMeans features). Configuration hp3 performed significantly worse, while hp1 and hp2 achieved similar validation accuracy but hp1 provided a slightly better balance of precision, recall and F1. We therefore selected hp1 as the final configuration and reused it for both the PCA only and the PCA plus KMeans classifiers.

Given the size of the dataset, this search is intentionally small. A larger grid over learning rates, hidden sizes and regularization parameters could further refine the model, but the current settings already achieve strong performance with reasonable complexity.

Other hyperparameters, such as the PCA variance threshold (95 percent), the autoencoder latent dimension (64), the maximum number of epochs (20 for the autoencoder and 50 for the

classifier) and the batch size (64), were kept fixed based on common practice and preliminary experiments rather than being tuned systematically.

5.4 Impact of cluster labels as additional features

To assess whether including cluster labels as additional features helps classification we compare the performance of two classifiers that share the same architecture and training procedure but differ in their input features.

- PCA only classifier: uses only the 106 PCA components.
- PCA plus KMeans classifier: uses the 106 PCA components concatenated with a 20 dimensional one hot encoding of the KMeans cluster label.

Model	Split	Accuracy	Precision (macro)	Recall (macro)	F1 (macro)
PCA plus KMeans	Validation	0.9884	0.9900	0.9833	0.9844
PCA plus KMeans	Test	0.9655	0.9733	0.9650	0.9643
PCA only	Validation	0.9767	0.9800	0.9708	0.9717
PCA only	Test	0.9770	0.9817	0.9733	0.9743

On the validation set the PCA plus KMeans classifier reaches slightly higher performance, but on the test set the two models achieve very similar accuracy and macro F1. This indicates that the one hot cluster labels do not provide a clear improvement in generalization. In our runs the PCA only model is at least as good as the PCA plus KMeans model on unseen data, which suggests that the PCA representation already captures most of the discriminative structure.

6. Quantitative evaluation and confusion matrix

Beyond aggregate metrics, it is useful to inspect how errors are distributed across classes. We plot the confusion matrices for the PCA only and PCA plus KMeans classifiers on the test set.

Figure 8: Can show the confusion matrix of the PCA plus KMeans classifier.

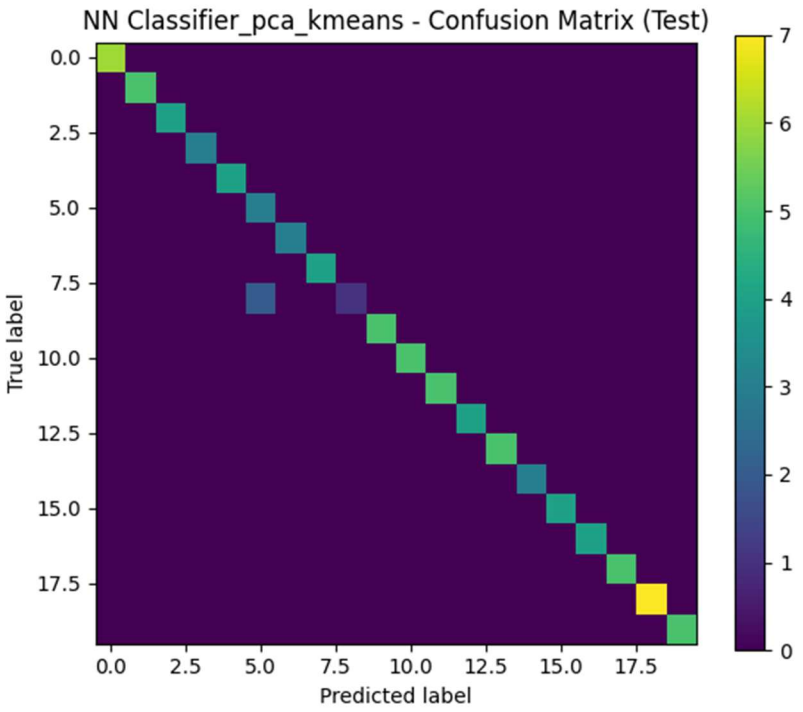
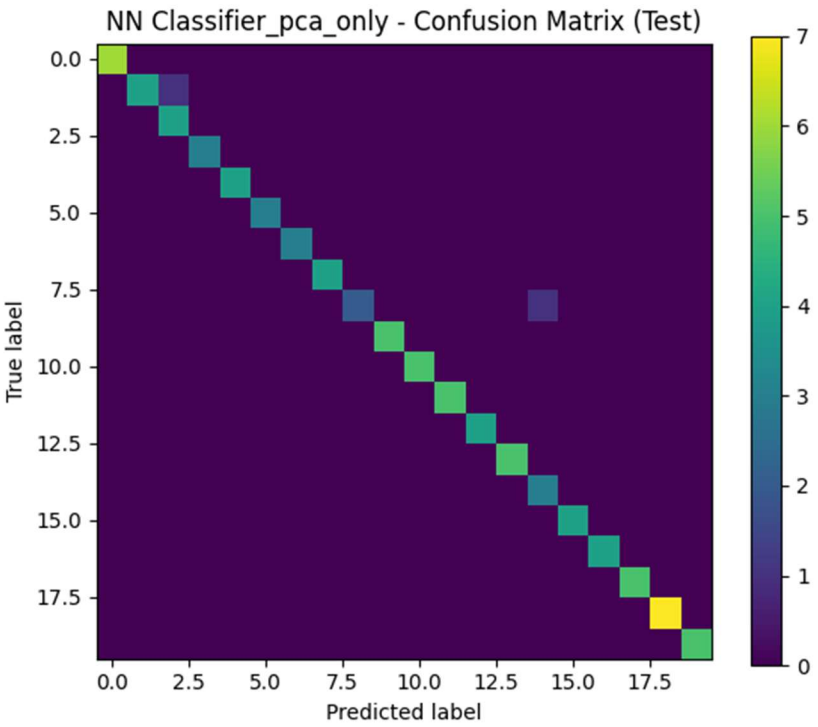


Figure 9: Can show the confusion matrix of the PCA only classifier.

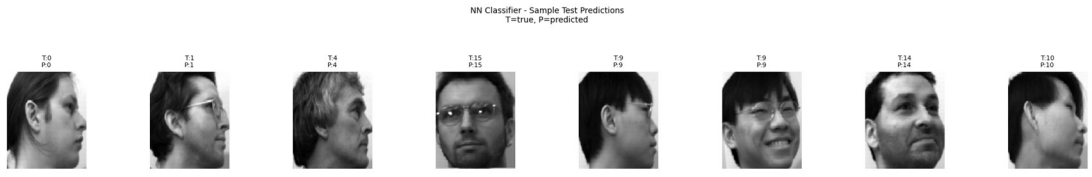


In both cases the diagonal dominates, which means that most images are correctly classified. Misclassifications are concentrated among a few pairs of people whose faces are visually similar or appear in similar poses. This matches the intuition that some individuals are harder to distinguish based on these grayscale images.

7. Qualitative analysis of sample predictions

To complement the numeric metrics, we examine a small set of concrete predictions made by the classifier.

Figure 10: Shows a grid of test images with their true and predicted labels using the PCA plus KMeans model.



Most examples in the grid are correctly classified, and the predicted label usually matches the person's identity. The few mistakes correspond to faces that either look very similar to another person or appear in unusual poses. Overall, the qualitative behaviour is consistent with the high-test accuracy and macro F1 reported earlier.

8. Conclusions

This project combined several unsupervised learning techniques to build a compact and informative representation of faces and then used these representations for supervised classification. PCA provided a strong linear baseline, and the autoencoder offered a non linear alternative with similar clustering structure in its latent space.

Clustering experiments with KMeans, hierarchical clustering and Gaussian mixture models revealed moderate cluster structure aligned with person identity. When the learned features were fed into a neural network classifier, high classification performance was achieved on both validation and test sets.

Adding KMeans cluster labels as extra features did not significantly improve generalization compared to using PCA alone, which suggests that the main discriminative information is already captured by the PCA components. The final models demonstrate that combining unsupervised feature extraction with a relatively small supervised network is a powerful approach for face recognition on this dataset.