hugomcp@di.ubi.pt, 2026/27

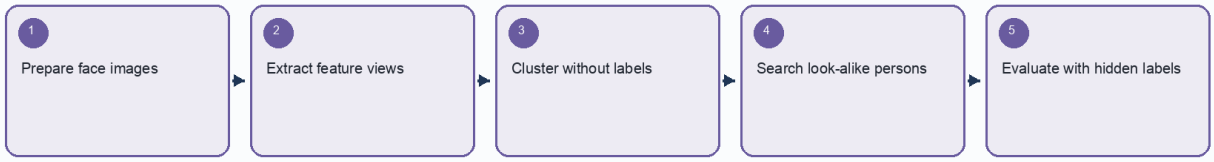
Machine Learning

Practical Project 3 - Unsupervised Learning

Dataset	AR face dataset from the course page.
Main question	Can unsupervised models discover identity groups and visually similar people without using labels during training?
Core methods	PCA/eigenfaces, local descriptors, autoencoder or pretrained embeddings, K-Means, agglomerative clustering, DBSCAN/HDBSCAN, and SOM.
Important constraint	Identity labels may be used only after clustering, for evaluation and error analysis.

Project 3 scheme

Unsupervised face grouping and look-alike search



Central idea: compare what the model learns from data alone with what changes when representation, objective, or physics constraints are changed.

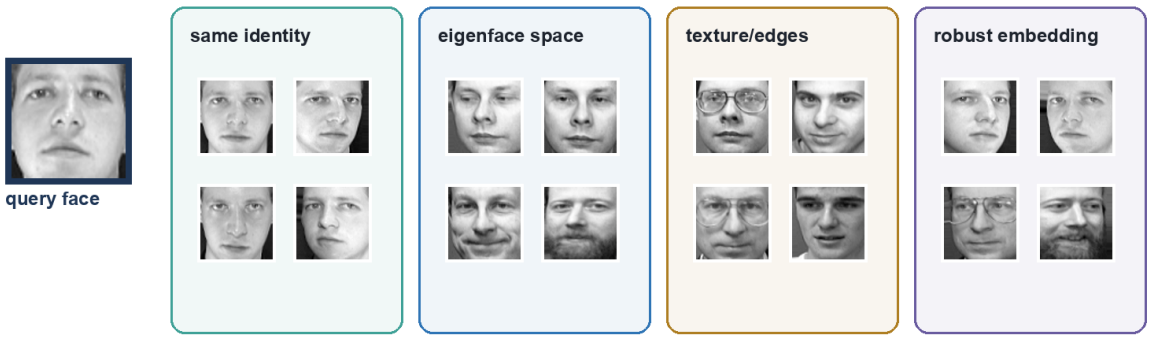
Face image examples: same person can vary in expression, lighting, and detail



The labels are hidden during clustering.

- Raw pixels are sensitive to illumination.
- Eigenfaces/PCA capture global structure.
- Texture/edge descriptors can emphasize local detail.
- Deep embeddings may group visually similar persons.

Look-alike people: similarity depends on the feature perspective



Major project idea

This project separates identity recognition from visual similarity. You should discover when two images cluster together because they are the same person, and when they are grouped because lighting, expression, accessories, pose, or facial texture make them look alike.

Major goals

- Cluster face images without using identity labels during model fitting.
- Compare multiple feature perspectives: raw pixels, eigenfaces/PCA, texture/edge descriptors, and learned embeddings.
- Build a look-alike search tool that retrieves visually similar persons according to selected feature spaces.
- Evaluate clustering quality quantitatively and interpret the most important errors visually.

Minor goals / checkpoints

- Detect, crop, resize, and normalize faces consistently; document any images that fail preprocessing.
- Compare grayscale and color preprocessing when the AR dataset is used.
- Study the number of PCA components/eigenfaces and visualize the first eigenfaces.
- Apply at least three clustering algorithms and justify their hyperparameters.
- Use silhouette/Davies-Bouldin without labels and ARI/NMI/homogeneity after labels are revealed.
- Create nearest-neighbour panels for identity similarity and look-alike similarity.
- Discuss privacy, bias, and the difference between educational analysis and deployment.

Implementation requirements

1. Create a preprocessing.py module that loads images, normalizes size/intensity, and creates feature matrices.
2. Create a features.py module with at least three feature views: raw pixels, PCA/eigenfaces, and one local or learned descriptor.
3. Create a clustering.py module comparing K-Means, agglomerative clustering, and one density-based or SOM method.
4. Create a lookalike_search.py module returning top-k visually similar images for a query face under different feature perspectives.
5. Keep ground-truth identities hidden until the final evaluation stage.

Experiments to report

- Cluster count sensitivity: evaluate several values of k and compare internal/external metrics.
- Feature sensitivity: explain which feature space best recovers identity and which best retrieves look-alikes.
- Robustness: study changes caused by lighting, expression, sunglasses/scarf, or occlusion if available.
- SOM or 2D manifold visualization: show whether identities or visual attributes occupy coherent regions.
- Error analysis: show false merges, false splits, and convincing look-alike examples.

What to submit

- Python scripts/notebook for preprocessing, feature extraction, clustering, and retrieval.
- A report with clustering metrics, visual cluster maps, and look-alike panels.
- A short ethical note explaining limitations and acceptable use of face clustering.
- A README with dataset choice, preprocessing decisions, and reproduction instructions.

Suggested data sources

AR Face Database reference: https://www.di.ubi.pt/~hugomcp/ml/AR_jpg.zip

Suggested 4-week workflow

- Prepare face images and feature matrices; define evaluation protocol with labels hidden during clustering.
- Implement PCA/eigenfaces and at least two clustering baselines.
- Add local/learned feature view, look-alike search, SOM or density-based clustering, and robustness tests.
- Compare metrics, assemble visual cluster/look-alike panels, write ethics note and final report.