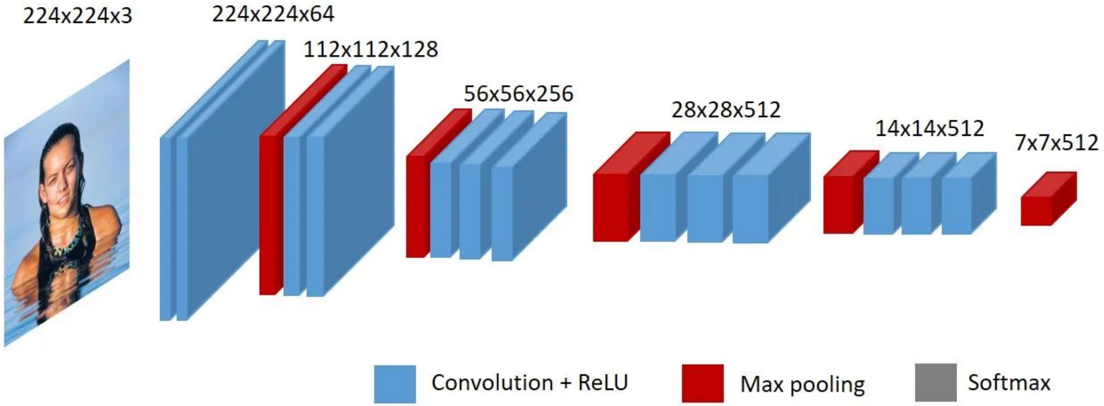
Images Preprocessing and Dimensionality reduction (PCA)

Francesco Villi

Introduction

- Dataset: Labeled Faces in the Wild
- Standardize or not standardize?
- Extract features: VGGFace
- Apply PCA
- Retrieval task using Faiss library
- Face recognition task using KKN neighborhood



Principal component analysis

- **Train**: Given sample

$$D = \{x_1, \ldots, x_n\}, x_i \in \mathbb{R}^n$$

  - Compute:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T$$

  - Compute eigenvalues and eigenvectors of Σ, where:

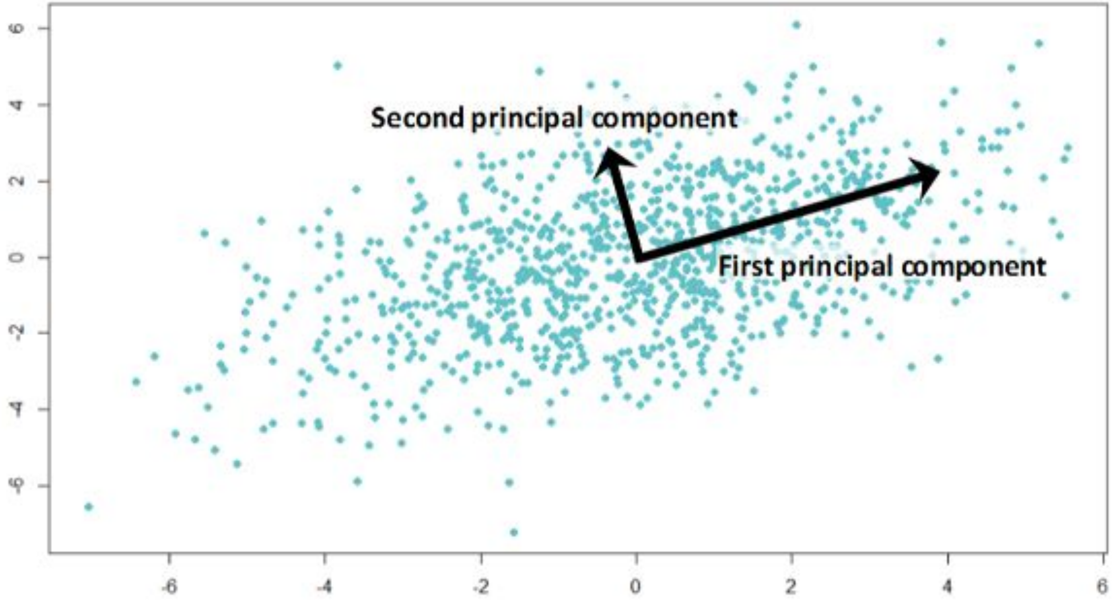
$$\Sigma = \Phi\Lambda\Phi^T,$$
$$\Lambda = diag(\sigma_1^2, \ldots, \sigma_n^2),$$
$$\Phi^T\Phi = I$$

  - Order eigenvalues $\sigma_1^2 > \ldots > \sigma_n^2$
  - Select K eigenvalues and eigenvectors

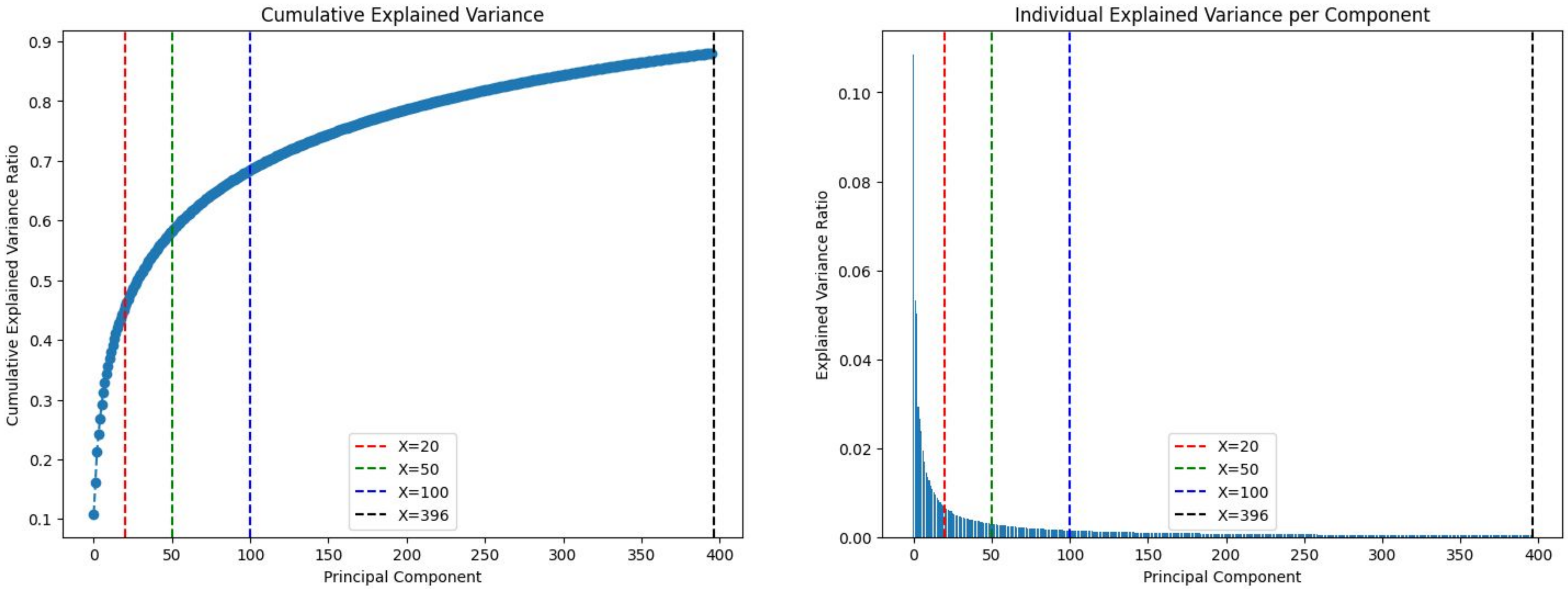
- **Test**: Given principal components $\phi_i, i \in 1, \ldots, k$ and test sample $T = \{t_1, \ldots, t_n\} \in \mathbb{R}^d$
  - Subtract mean from each point $t_i' = t_i - \hat{\mu}$
  - Project onto eigenvector space $y_i = At_i'$ where

$$A = \begin{pmatrix} \phi_1^T \\ \vdots \\ \phi_k^T \end{pmatrix}$$

  - Use $T' = \{y_1, \ldots, y_n\}$



Choose number of components



FAISS library

Key points:

- Exhaustive search with IndexFlatL2 and IndexFlatIP
- Flat indexes just encode the vectors into codes of a fixed size and store them in an array.
  - At search time, all the indexed vectors are decoded sequentially and compared to the query vectors.
  - IndexFlat: the vectors are stored without compression
- IndexFlatIP can be used as cosine similarity

Cosine Similarity and L2 Dist. for Normalized Vectors

Cosine Similarity with Normalized Vectors:

$$\text{cosine\_similarity}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} = \mathbf{a} \cdot \mathbf{b}$$

L2 Distance with Normalized Vectors:

$$\begin{aligned} \text{L2\_distance}(\mathbf{a}, \mathbf{b}) &= \|\mathbf{a} - \mathbf{b}\|_2 \\ &= \sqrt{(\mathbf{a} - \mathbf{b}) \cdot (\mathbf{a} - \mathbf{b})} \\ &= \sqrt{\mathbf{a} \cdot \mathbf{a} - 2\mathbf{a} \cdot \mathbf{b} + \mathbf{b} \cdot \mathbf{b}} \end{aligned}$$

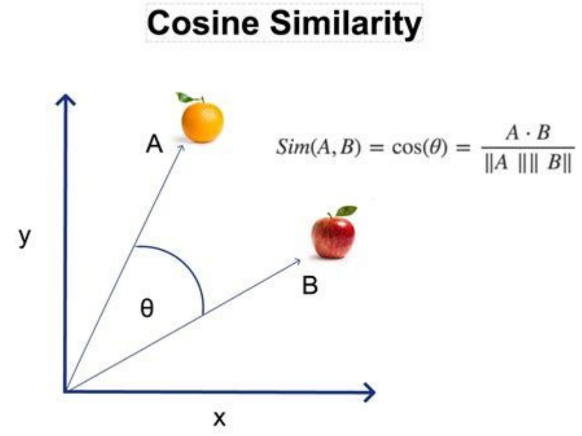
3.1. Substitute $\|\mathbf{a}\| = \|\mathbf{b}\| = 1$:

$$\begin{aligned} &= \sqrt{1 - 2(\mathbf{a} \cdot \mathbf{b}) + 1} \\ &= \sqrt{2 - 2(\mathbf{a} \cdot \mathbf{b})} \end{aligned}$$

$$\text{cosine\_similarity}(\mathbf{a}, \mathbf{b}) = \mathbf{a} \cdot \mathbf{b}$$

$$\text{L2\_distance}(\mathbf{a}, \mathbf{b}) = \sqrt{2 - 2(\mathbf{a} \cdot \mathbf{b})}$$

We observe that both expressions are equivalent. Hence, for normalized vectors, cosine similarity is indeed equivalent to L2 distance.



Cosine distance is not a true distance metric!!

Project structure

- **Get face images**: sklearn LFW datasets with RGB images where the lowest number of samples was 55 (9 different faces).
- **Balancing data (opt.)**: Selecting the minimum number of available samples for the underrepresented class.
- **Split dataset**: I partitioned the dataset into training and testing sets
- **Pixel standardization (opt.)**: I scaled pixel values of train set to have a zero mean and unit variance
- **Resize images**: Resized images to 224x224 size
- **Extracting features**: I used VGGFace without the 3 fully connected layers at the top of the network. So for each sample, I got a tensor with shape (7, 7, 512).
- **Reduction**: I applied PCA reductions to the features extracted from VGGFace.

Tests

- Standardization or not standardization?
- PCA or not PCA?

- TESTS:
  - **Retrieval task**
    - RAW features vs PCA (with 20, 50,100, 396) vs PCA (excluding first 1,2,3 components out of 20, 50, 396)
    - Computing the mean over 9 differents splits of train-test set, I measure the precision at various level (@5, @10, @20, @ALL)
    - Everything tested with standardized and non-standardized images
  - **Recognition task**
    - K-Nearest Neighbors
      - In a tie, sum "distances" per label, favor the greatest similarity
      - Tested with std and non-std, balanced and unbalanced dataset

Results retrieval task

$ID\ Image$	$P@10 - STD$				
	RAW	20	50	100	396
0	**0.97**	0.96	0.96	0.96	0.96
1	**0.97**	0.92	0.93	0.93	0.93
2	**0.94**	0.93	0.92	0.91	0.91
3	**0.92**	0.85	0.85	0.85	0.84
4	**0.86**	0.83	0.82	0.80	0.79
5	**0.95**	0.94	**0.95**	**0.95**	0.94
6	**0.91**	0.86	0.86	0.86	0.87
7	**0.99**	0.98	0.98	0.98	0.98
8	**0.86**	0.84	0.84	0.85	0.84
AVG	**0.93**	0.90	0.90	0.90	0.90

Table 1. Table displaying P@10 scores over 9 runs with different split test-train. Comparing PCA with 20,50,100,396 and 7x7x512 components with image standardization.

$ID\ Image$	$P@10$				
	RAW	20	50	100	396
0	**0.96**	0.86	0.87	0.87	0.86
1	**0.93**	0.86	0.86	0.87	0.86
2	**0.87**	0.83	0.83	0.84	0.83
3	**0.81**	0.75	0.75	0.75	0.75
4	**0.79**	0.66	0.68	0.70	0.67
5	0.85	0.85	**0.86**	**0.86**	0.84
6	**0.85**	0.82	0.83	0.84	0.84
7	**0.98**	0.97	0.96	0.96	0.96
8	**0.82**	0.77	0.77	0.76	0.76
AVG	**0.87**	0.82	0.82	0.83	0.82

Table 2. Table displaying P@10 scores over 9 runs with different split test-train. Comparing PCA with 20,50,100,396, 7x7x512 components without image standardization

Results retrieval task

ID Image	*STD − P@10*								
	−1/20	−1/50	−1/396	−2/20	−2/50	−2/396	−3/20	−3/50	−3/396
0	0.95	**0.96**	**0.96**	0.93	0.94	0.94	0.94	0.88	0.88
1	0.87	**0.88**	0.87	0.84	0.86	0.84	0.84	0.84	0.83
2	0.92	0.91	0.91	**0.94**	0.93	0.93	0.93	**0.94**	**0.94**
3	**0.81**	**0.81**	0.80	0.80	**0.81**	0.79	0.79	0.78	0.76
4	**0.82**	0.80	0.79	**0.82**	0.81	0.80	0.80	**0.82**	**0.82**
5	**0.92**	**0.92**	0.91	0.91	0.91	0.90	0.90	0.91	0.90
6	0.85	0.86	0.87	**0.88**	0.87	**0.88**	**0.88**	0.85	0.83
7	**0.98**	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.96
8	0.80	0.80	0.80	**0.82**	0.81	0.79	0.79	0.81	0.79
AVG	**0.88**	**0.88**	0.87	**0.88**	**0.88**	0.87	0.87	0.86	0.86

Table 9. Table displaying rounded Precision@10 scores for different PCA settings; ”-X/K” denotes the removal of the first X out of K components in PCA reconstruction. Mean computed over 9 runs.

ID Image	*P@10*								
	−1/20	−1/50	−1/396	−2/20	−2/50	−2/396	−3/20	−3/50	−3/396
0	0.87	**0.88**	**0.88**	0.83	0.84	0.84	0.84	0.84	0.85
1	0.84	**0.85**	0.85	0.83	**0.85**	0.84	0.80	0.81	0.79
2	0.86	0.86	0.85	0.87	0.87	0.87	**0.91**	**0.91**	0.90
3	0.80	0.78	0.79	**0.82**	0.79	0.81	0.75	0.68	0.71
4	0.71	0.73	0.69	0.71	0.72	0.69	0.73	**0.74**	0.73
5	0.87	**0.88**	**0.88**	0.85	0.86	0.86	0.86	0.86	0.84
6	0.86	**0.89**	0.88	0.83	0.86	0.85	0.86	**0.89**	0.86
7	**0.97**	**0.97**	0.96	0.96	0.96	0.96	0.95	0.95	0.96
8	0.80	**0.81**	**0.81**	0.70	0.71	0.72	0.70	0.69	0.70
AVG	0.84	**0.85**	0.84	0.82	0.83	0.83	0.82	0.82	0.82

Table 10. Feature extracted has been standardized before to extract features and applying PCA. ”-X/K” denotes the removal of the first X out of K components in PCA reconstruction. Mean computed over 9 runs.

Results retrieval task

STD PCA red.	P@5	P@10	P@20	P@ALL
20	0.94	0.90	0.84	**0.33**
50	0.94	0.90	0.82	0.28
100	0.94	0.90	0.82	0.27
396	0.94	0.90	0.81	0.27
0	**0.96**	**0.93**	**0.83**	0.14
-1/20	0.93	0.88	0.78	0.28
-2/20	0.92	0.88	0.79	0.29
-3/20	0.91	0.87	0.77	0.25
-1/50	0.93	0.88	0.76	0.24
-2/50	0.93	0.88	0.77	0.23
-3/50	0.92	0.86	0.74	0.20
-1/396	0.93	0.87	0.74	0.23
-2/396	0.93	0.87	0.74	0.22
-3/396	0.92	0.86	0.70	0.19

Table 3. Feature extracted has been standardized before to extract features and apply PCA.Precision as a mean value between all classes at different cut-off values for PCA components. Mean computed over 9 runs. 0 means no reduction

PCA red.	P@5	P@10	P@20	P@ALL
20	0.87	0.82	0.73	0.22
50	0.89	0.82	0.72	0.20
100	0.89	0.83	0.71	0.20
396	0.89	0.82	0.70	0.20
0	**0.93**	**0.87**	0.72	0.12
-1/20	0.88	0.84	**0.76**	**0.23**
-2/20	0.88	0.82	0.71	0.21
-3/20	0.87	0.82	0.71	0.21
-1/50	0.89	0.85	0.74	0.21
-2/50	0.89	0.83	0.69	0.18
-3/50	0.88	0.82	0.67	0.18
-1/396	0.90	0.84	0.72	0.20
-2/396	0.90	0.83	0.66	0.17
-3/396	0.89	0.82	0.64	0.17

Table 4. Precision as a mean value between all classes at different cut-off values for PCA components. Mean computed over 9 runs. 0 means no reduction.

Results recognition task

ID test	Accuracy w/ n components STD				
	RAW	20	50	100	396
test 1	0.979	**0.989**	**0.989**	0.969	0.979
test 2	**0.989**	0.949	0.939	0.959	0.969
test 3	**1.000**	0.989	0.989	**1.000**	0.989
test 4	0.989	0.989	0.989	**1.000**	0.979
test 5	**1.000**	0.979	0.989	0.989	**1.000**
test 6	**0.989**	0.949	0.959	0.959	0.969
test 7	**0.979**	0.969	0.969	0.969	0.969
test 8	0.989	**1.000**	0.979	0.989	**1.000**
test 9	**0.979**	**0.979**	**0.979**	**0.979**	0.969
AVG	**0.988**	0.977	0.976	0.979	0.980

Table 6. Accuracy in face recognition with different train-test splits and using KKN with k=9. All images have been pre-processed with standardization and a balanced dataset

ID test	Accuracy w/ n components				
	RAW	20	50	100	396
test 1	**0.985**	0.971	0.964	0.967	0.975
test 2	**0.989**	0.960	0.964	0.975	0.978
test 3	**0.978**	0.971	0.971	0.964	0.971
test 4	**0.989**	0.946	0.953	0.957	0.967
test 5	**0.982**	0.943	0.950	0.960	0.960
test 6	**0.982**	0.950	0.957	0.950	0.957
test 7	**0.982**	0.946	0.953	0.960	0.957
test 8	**0.982**	0.957	0.960	0.967	0.971
test 9	**0.989**	0.960	0.971	0.975	0.978
AVG	**0.984**	0.956	0.960	0.964	0.968

Table 5. Accuracy in face recognition with different train-test splits and using KKN with k=9

Results recognition task

ID test	Accuracy w/ n components STD 30				
	RAW	20	50	100	396
test 1	0.965	0.901	0.941	0.960	0.960
test 2	0.970	0.911	0.926	0.936	0.950
test 3	0.990	0.931	0.960	0.960	0.970
test 4	0.980	0.931	0.950	0.960	0.965
test 5	0.975	0.955	0.965	0.960	0.960
test 6	0.985	0.965	0.960	0.965	0.980
test 7	0.980	0.926	0.975	0.980	0.985
test 8	0.990	0.936	0.965	0.970	0.980
test 9	0.980	0.960	0.955	0.960	0.975
AVG	**0.979**	0.934	0.955	0.961	0.970

Table 8. Accuracy in face recognition with different train-test splits and using KKN with k=9. All images have been pre-processed w standardization and used a balanced dataset where min_faces_per_person=30

ID test	Accuracy w/ n components Unb. STD 30				
	RAW	20	50	100	396
test 1	0.970	0.924	0.945	0.959	0.959
test 2	0.983	0.949	0.966	0.978	0.983
test 3	0.978	0.926	0.953	0.959	0.962
test 4	0.978	0.917	0.947	0.962	0.964
test 5	0.978	0.932	0.953	0.968	0.970
test 6	0.987	0.945	0.955	0.968	0.968
test 7	0.985	0.947	0.949	0.955	0.970
test 8	0.976	0.928	0.951	0.953	0.964
test 9	0.966	0.907	0.938	0.949	0.951
AVG	**0.978**	0.930	0.951	0.961	0.966

Table 9. Accuracy in face recognition with different train-test splits and using KKN with k=9. All images have been pre-processed w standardization and used an unbalanced dataset where min_faces_per_person=30

Conclusion

- Feature Extraction using VGGFace and PCA:
  - Introduction of image standardization enhances algorithm performance.
  - Consistency in maintaining all components yields superior results across scenarios.
  - PCA offers a solution for resource-constrained environments
- Impact of Removing PCA Components:
  - Best results achieved without standardization, removing the first component.
  - However, degradation in overall performance observed compared to standard method.
- Face Recognition using KNN with PCA:
  - Standardized images generally outperform non-standardized ones in recognition tasks.
  - Using unbalanced datasets doesn't decrease performance in recognition