Pattern Recognition Coursework 2

Gustavo Brito Rodríguez
Imperial College London
01198785
gb1216@ic.ac.uk

Alberto García Matachana
Imperial College London
01205901
ag4116@ic.ac.uk

Abstract

This report compares the performance of different distance metrics with different data pre-processing. It also explores different feature representations such as histograms and cluster-based representations and evaluates their performance and possible reasons behind them.

1. Question 1: Distance Metrics

1.1. Data

The data set provided (face.mat) consists of 10 raster-scanned 46x56 (2576 pixels) grey-scale face images for each of 52 individuals. All the 520 images are frontal, but the facial expressions and directions of gazes are varied.

Each image $n \in N = \{1, 2, ..., 520\}$, was represented as a single vector $x_n \in \mathbb{R}^D : D = 2576$. The whole data set matrix was therefore $X \in \mathbb{R}^{N \times D}$. The L_2 normalised version of X is $\bar{X}$. The standardised version of X is $\tilde{X}$, where all elements of X have been scaled to have unit variance and 0 mean.

The data was separated into training split, defined as:

$$X_{tr} \in \mathbb{R}^{N_{tr} \times D} : N_{tr} = 320 \tag{1}$$

and test split defined as:

$$X_{tst} \in \mathbb{R}^{N_{tst} \times D} : N_{tst} = 200 \tag{2}$$

1.2. Baseline

This subsection focuses in establishing a performance baseline for the standard non-learned distance metrics discussed in the lectures, defined in Appendix A.1. The values were calculated on the raw (X_{tst}), normalised ($\bar{X}_{tst}$) and standardised ($\tilde{X}_{tst}$) test sets.

The metrics used are the Rank1 and Rank10 accuracies, where rank$k$ is the probability that an image of the same label is within the nearest k images, and the mean average precision (mAP) for Rank10. The best percentage accuracy results, obtained from the standardised data, can be seen on Table 1. On Appendix A.1, percentage results for unmodified and normalised data can be found; Table 8 and Table 9, respectively.

Metric	Standardised		
	Rank1	Rank10	mAP
L_1	76.0	95.8	0.527
L_2	70.0	94.5	0.471
L_∞	57.0	82.0	0.170
Chi-Square	70.0	94.0	0.480
Cosine	68.5	96.0	0.476
Correlation	68.5	96.0	0.476

Table 1: Baseline performance on $\tilde{X}_{tst}$

The table above shows how processing the data normally results in an improved performance. For example, Minkowski-Form distances (L_1, L_2 and L_∞) benefit from processing greatly as all features are weighted equally, which results in larger values dominating the evaluations even if they are not representative of the class. The uniform range of data (in normalisation) and unit variance (in standardisation) improves performance. Nevertheless, higher-order Minkowski-Form distances performed worse than their lower-order counterparts, as in higher dimensions, the concept of proximity, distance and nearest neighbor are less qualitatively meaningful[1].

In general, standardisation resulted in slightly better performance than normalisation as it preserves the outliers in the data.

In the case of the Cosine and Correlation distance, there is no surprise that the accuracy is equal for processed and unmodified data as the difference in direction between points should not change post-normalisation, only the vector lengths. Additionally, the results for Cosine and Correlation distance on the standardised data are equal as their formulas are equivalent at 0 mean.

Finally, Chi-Square is a weighted variant of L_2, which is more robust to outlying data points. It weighs the inverse of the species totals. Therefore, despite being more robust to outlying data, it also tends to exaggerate the weights on less common features. It performs better than Cosine and Correlation distances because it does not consider the variance in vector angle as a whole, but, the difference of each feature. However, when data is normalised, Chi-Square performs worse since this processing technique gets rid of outliers.

1.3. Experiment 1. Histogram of Pixel Intensities

It is also possible to use a histogram of the non-normalised pixel intensities with different bin sizes as the image representation. As in 1.2, we can evaluate the non-learned distance metrics defined in Appendix A.1. A sample histogram for the first 3 images of X_{tst} and bin size 1 can be found in Figure 3.

The respective Rank1/Rank10 accuracy along with the Rank10 mAP are shown on Table 2 below .

Metric	Histogram		
	Rank1	Rank10	mAP
Bhattacharyya	23.5	63.5	0.207
Chi-Square	26.5	63.0	0.224
Intersection	20.0	60.5	0.187
Correlation	17.0	56.5	0.122

Table 2: Baseline performance on pixel intensity histogram with bin size 1

When calculating non-learned distance metrics on a histogram, it is important to achieve a good balance between robustness and distinctiveness, and this is done by varying the bin size. The different metrics were also calculated with 50 and 20 bin sizes, the results of which can be found in Appendix A.2.

As was expected, a decrease in the number of bins reduced distinctiveness, i.e. Rank1, and increased the robustness, i.e. Rank10. This is because, as you increase bin size, less details in the image are considered, but, is more immune to the noise in the images.

Regarding time and storage complexity, both metrics were lower due to dimensionality reduction.

Overall, reducing the dimensionality achieved lower accuracies than the full image feature space metrics. The maximum mAP achieved was 0.231 with the Chi-Square Histogram distance with 50 bins. This value is considerably lower than the best achieved accuracy in 1.2 of 0.527, obtained by applying Manhattan Distance to the Standardised data set.

1.4. Experiment 2. Mahalanobis Distance and Dimensionality Reduction

The Mahalanobis Distance metric attempts to address the problems that Euclidean Distance suffers from in high dimensions, by transforming the data points into uncorrelated variables[2][1]. Furthermore, Mahalanobis Distance uses the covariance matrix of the data in order to improve performance on non-spherically distributed data[3]. Therefore, one would expect the performance of the Mahalanobis Distance to be an improvement on Minkowski-Form distances given that the Mahalanobis Distance is a generalised version of the Euclidean Distance. The equation for the Mahalanobis Distance between two points can be found in Appendix A.3. During experiments, the covariance matrix was found not to be positive semi-definite so the pseudo-inverse was used instead.

Additionally, Mahalanobis Distance can also be used as a way to reduce the dimensionality in the same way that PCA does, hence, the accuracy of the method is expected to reduce with the number of eigenfaces used. Tests were carried out with 16,32,64,128 and 256 eigenvectors, the results of which, can be seen on Table 3 below.

Eigenvector number	Unmodified		
	Rank1	Rank10	mAP
16	58.3	80.2	0.154
32	65.6	92.7	0.356
64	70.2	94.4	0.473
128	77.4	97.1	0.550
256	77.4	97.1	0.550

Table 3: Mahalanobis Distance performance on X_{tst}

As expected, the use of Mahalanobis Distance with high number of eigenvectors results in improved performance on the Euclidean Distance metric. This change can be more dramatically seen on the non-normalised as the uniform range of data (in normalisation) and unit variance (in standardisation) works against the Mahalanobis Distance in this case. Nevertheless, there is still an improvement on the processed data as Mahalanobis Distance also takes correlation between dimensions into account (which normalisation does not).

1.5. Experiment 3. Dimensionality reduction with PCA and LDA

Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are linear transformation methods used for dimensionality reduction. LDA is a method whose objective is to find the set of directions $W = [w_1, w_2, ..., w_M] \in \mathbb{R}^{D \times M}$ (known as Fisherfaces) that optimally separate data of different classes. This is fundamentally different from PCA, whose objective is to find the direction of maximum data variance regardless of image labels.

In this section, PCA and PCA-LDA were applied on the feature vectors described in 1.1 and 1.3 and observe how the use of histograms as a feature space compare with standardised pixel values as a feature space. For the tests performed, 50 bins were used in the histogram feature space as it yielded the best performance in 1.3. Additionally, the value of M_{LDA} from Equation (16) in Appendix A.4 used was 31 (the rank of the Between-class scatter matrix) as in Coursework 1, the closer M_{LDA} was to the Between-class scatter matrix rank, the higher the classification per-

formance was. For PCA, given that the there was no fallout in accuracy with 128 and 256 eigenfaces in 1.4, the number of eigenfaces used for this part was 128.

The rank10 mAP results for different distance metrics can be found on the table below:

Metrics	PCA		PCA-LDA	
	Pixels	Hist	Pixels	Hist
L_1	0.560	0.323	0.652	0.486
L_2	0.552	0.311	0.631	0.471
Chi-Square	0.526	0.298	0.622	0.420
Correlation	0.503	0.277	0.606	0.394

Table 4: PCA and LDA perfromed on $\tilde{X}_{tst}$ and histogram of pixel identity with 50 bins.

The PCA method meant a significant improvement on the baseline metrics, but, was completely outclassed by PCA-LDA which yielded much better results in every metric. The performance of the L_2 PCA pixel feature representation was very similar to that of the Mahalanobis Distance metric in 1.4, which is unsurprising given how both use the eigenvectors of the covariance matrix (or its inverse) and then use Euclidean distance.

The different metrics used produced results similar to those in 1.2 which further proves the observations made in that section.

As observed in 1.3, the histograms produced worst results than the full image, but, the time taken to run the PCA algorithm for the pixel feature representation was 3.5 seconds versus 0.8 seconds for the histogram feature vectors.

1.6. Experiment 4. Mahalanobis Metric Learning

In this section, Relevant Component Analysis (RCA) was performed on the data and its performance compared to the Mahalanobis Distance from 1.4 [4].

RCA attempts to discard the variability in the data which is not relevant to retrieval and deteriorates performance. This irrelevant data is defined as data which is maintained in the data set but is not correlated with the specific task at hand[4].

The rank10 mAP of this experiment for *chunklets* of size 4, 6 and 10 can be found below:

Chunklet size	RCA mAP
4	0.561
6	0.584
10	0.594
15	0.579

Table 5: RCA performed on X_{tst} with different *chunklet* size.

This algorithm clearly outperformed the regular Mahalanobis Distance metric, which is to be expected, given that it also did in [4] because of the use of chunklets to remove irrelevant data. Nevertheless, it performed worse than PCA-LDA in 1.5, given that PCA-LDA is a supervised method.

Increasing chunklet size also improved performance reaching its maximum when the chunklet size is equal to the class size. This differs from supervised learning, however, in the fact that despite class size being known, it is not known which labels correspond to each centroid during learning. This means that the irrelevant data is discarded amongst images of the same chunklet, which should make them easier to classify.

2. Question 2: Cluster based representations

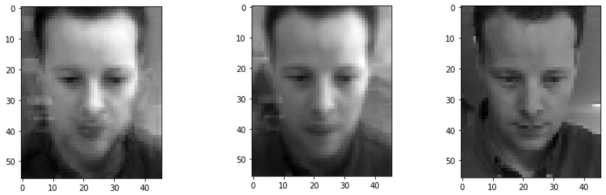
2.1. Experiment 5. Clustering training data

In Experiment 5, the objective was to obtain centroids for each class in the training data, X_{tr}, through both K-means and Agglomerative clustering in order to later obtain new feature representations for the data.

Functions *KMeans* and *AgglomerativeClustering* were imported from the *sklearn.cluster* library. For K-means clustering, the function was used with number of clusters $= 32$, a tolerance of 0.0001 and 1000 iterations. For Agglomerative Clustering, the parameters used were euclidean affinity, ward linkage and 32 clusters.

Both imports had a built-in function which returned figurative labels for each image (i.e. labels had no meaning apart from indicating which images were in the same cluster). The centroids were obtained by taking the average image from each of these clusters. Once the centroids were obtained, they were properly labelled with a Hungarian algorithm. Its cost function was calculated by taking the euclidean distance between the obtained centroids and the average image from the training data labels.

Figure 1 below compares the centroid with label 31 assigned by the Hungarian algorithm, the average label 31 image drawn from the training set and an individual sample image with label 31.



(a) Centroid label 31 (b) Label 31 average (c) Label 31 sample

Figure 1: Centroid comparison

One can see how Figures 1a and 1b are similar which would explain how they got assigned the same label. The results for labeling accuracy are shown on Table 6 below.

Metrics	Clustering	
	K-Means	Agglomerative
Manhattan	39.38	43.75
Euclidean	33.12	37.50
Chebyshev	26.88	32.27
Cosine	32.34	37.50
Correlation	35.96	40.62

Table 6: Comparison of K-means and agglomerative clustering labeling performances

Agglomerative outperforms K-means. In K-Means, cluster centers are drawn towards denser regions of the sample distribution, which results in tight clusters around denser regions and vice versa. This is amplified in high dimensions for reasons similar to the ones in 1.2. K-means is also influenced by within-class outliers which force the centroids towards them.

An additional experiment was carried out on the agglomerative method in order to observe how labeling accuracy changes with the number of clusters. If this number was unknown to us, a method such as the elbow method could be used to determine a guideline. Nevertheless, it known that the training set contains 32 different labels and hence 32 clusters. The increase in performance (which later quickly leads to a decrease) seen in Figure 2 is due to overfitting of stochastic noise in the training data.

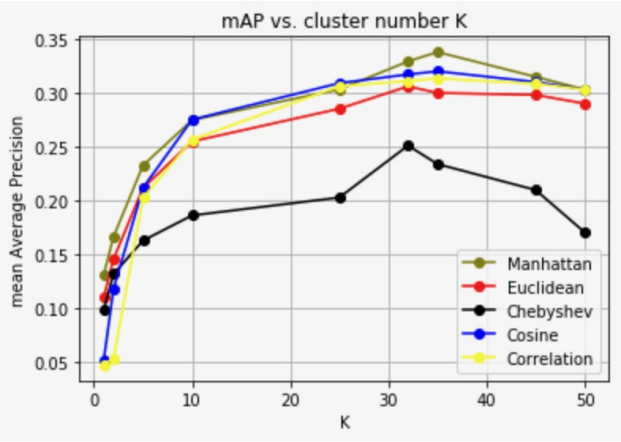


Figure 2: Labeling performance against number of clusters

2.2. Experiment 6. New Feature Representations

For this section, three new feature representations were obtained from the Agglomerative centroids obtained in 2.1, as they outperformed K-means-obtained centroids.

- *Centroid Distance (CD)*
- *Inverse Softmax (IS)*
- *GMM-FisherFaces (GMM)*

The first representation, *Centroid Distance*, uses as a feature representation the Euclidean distance from the data point to each centroid.

Inverse Softmax uses the softmax probabilities of inverse distances to cluster centres as the vector representation.

Finally, *GMM-FisherFaces* calculates fisher vectors from Gaussian Mixture Models, which build on the *Inverse Softmax* by using it as the association strength.

Their rank10 mAP can be compared on Table 7 below and their rank1 and rank10 breakdowns can be found in B.1

Metrics	CD	IS	GMM
Manhattan	0.314	0.335	0.406
Euclidean	0.287	0.313	0.327
Chebyshev	0.256	0.271	0.295
Cosine	0.307	0.311	0.332
Correlation	0.303	0.307	0.311

Table 7: mAP comparison of Centroid Distance, Inverse Softmax and GMM FisherFaces, respectively.

The table above shows GMM-FisherFace outperforms the 2 other representations. The Fisher method uses a Gaussian Mixture Model to construct a visual word dictionary in the low level feature space. We represent the image as the gradient of the log likelihood with respect to the parameters of the model. The Fisher Vector is just the concatenation of these partial derivatives, therefore, it describes in which direction the parameters of the model should be modified to best fit the data.

The use of cluster-based representations also results in an improvement on test time complexity given that once the centroids have been calculated, one NN is performed on K number of centroids rather than N points.

In the end, the maximum performance was achieved with the PCA-LDA method from 1.5. The use of a higher dimensional plane allows for better accuracy despite its added time and storage complexity.

References

[1] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space, 2001.

[2] P. Mahalanobis. On tests and measures of group divergence. 1. theoretical formulae. *Jour And Proc Asiatic Soc Bengal*, 26(4):541–588, 1930( ), 1933.

[3] D. M. A. K. M. M.Brindha, Dr.G.M.Tamilselvan. A comparative study of face authentication using euclidean and mahalanobis distance classification method, 2015.

[4] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis, 05 2002.

A. Question 1

A.1. Non-learned Distance Metrics

This section defines the baseline distance metrics between two points, $P = \{p_1, p_2, ..., p_D\}$ and $Q = \{q_1, q_2, ..., q_D\}$, that are referred throughout the report and whose results are in 1.2.

The Minkowski-Form distance of order z is defined as:

$$L_z(P,Q) = \Big[\sum_{i \in D} |p_i - q_i|^z\Big]^{\frac{1}{z}} \tag{3}$$

The City Block distance, also known as Manhattan distance, is the first-order Minkowski-Form distance:

$$L_1(P,Q) = \sum_{i \in D} |p_i - q_i| \tag{4}$$

The Euclidean distance is the second-order Minkowski-Form distance:

$$L_2(P,Q) = \sqrt{\sum_{i \in D} |p_i - q_i|^2} \tag{5}$$

The Chebyshev distance, also known as the Chessboard distance is the Minkowski-Form distance when $z \to \infty$:

$$L_\infty(P,Q) = \lim_{z \to \infty} \Big[\sum_{i \in D} |p_i - q_i|^z\Big]^{\frac{1}{z}} \tag{6}$$

The Chi-Square distance is defined as:

$$\chi^2(P,Q) = \sqrt{\frac{1}{2}\sum_{i \in D} \frac{(p_i - q_i)^2}{p_i + q_i}} \tag{7}$$

The Cosine distance is defined as:

$$\text{Cosine}(P,Q) = 1 - \frac{P^T Q}{||P||_2 ||Q||_2} \tag{8}$$

Finally, the Correlation distance is defined as:

$$\text{Correlation}(P,Q) = 1 - \frac{(P - \bar{P})^T (Q - \bar{Q})}{||P - \bar{P}||_2 ||Q - \bar{Q}||_2} \tag{9}$$

Metric	Unmodified		
	Rank1	Rank10	mAP
L_1	31.0	59.0	0.227
L_2	21.5	51.0	0.177
L_∞	15.0	48.2	0.152
Chi-Square	74.0	95.5	0.490
Cosine	68.0	94.5	0.460
Correlation	70.0	96.5	0.485

Table 8: Baseline performance on X_{tst}

Metric	Normalised		
	Rank1	Rank10	mAP
L_1	75.5	95.5	0.515
L_2	68.0	94.5	0.460
L_∞	55.0	80.1	0.157
Chi-Square	70.0	94.0	0.480
Cosine	68.0	94.5	0.460
Correlation	70.0	96.5	0.485

Table 9: Baseline performance on $\bar{X}_{tst}$

A.2. Histograms

The following describe distance metrics to compare two histograms $H_1 = \{H_1(1), H_1(2), ..., H_1(n)\}$ and H_2 where N is the number of bins.

The Bhattacharyya Distance between two histograms is defined as:

$$B(H_1,H_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{H}_1 \bar{H}_2 N^2}} \sum_i \sqrt{H_1(i) \times H_2(i)}} \tag{10}$$

where $\bar{H}_n$ is the average bin value of histogram H_n.

The Chi-Square Distance between two histograms is defined as:

$$\chi^2(H_1,H_2) = \sum_i \frac{(H_1(i) - H_2(i))^2}{H_1(i)} \tag{11}$$

The Correlation Distance between two histograms is defined as:

$$\text{Corr}(H_1,H_2) = \frac{\sum_i (H_1(i) - \bar{H}_1)(H_2(i) - \bar{H}_2)}{\sqrt{\sum_i (H_1(i) - \bar{H}_1)^2 \sum_i (H_2(i) - \bar{H}_2)^2}} \tag{12}$$

Finally, the Intersection Distance between two histograms is defined as:

$$\text{Inter}(H_1, H_2) = \sum_i \min(H_1(i), H_2(i)) \quad (13)$$

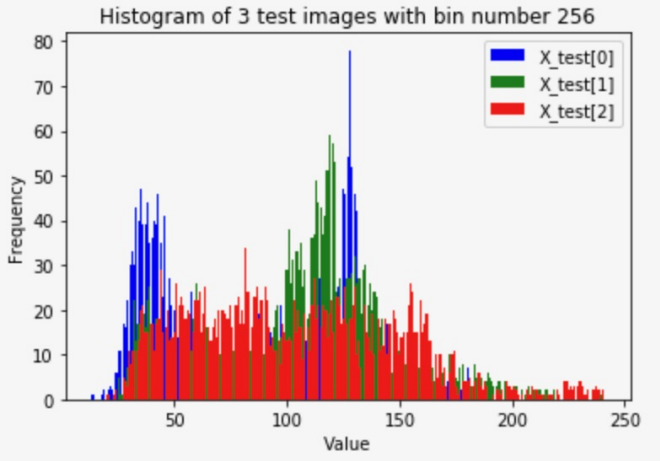


Figure 3: Histogram of pixel intensity for first 3 test images with 256 bins.

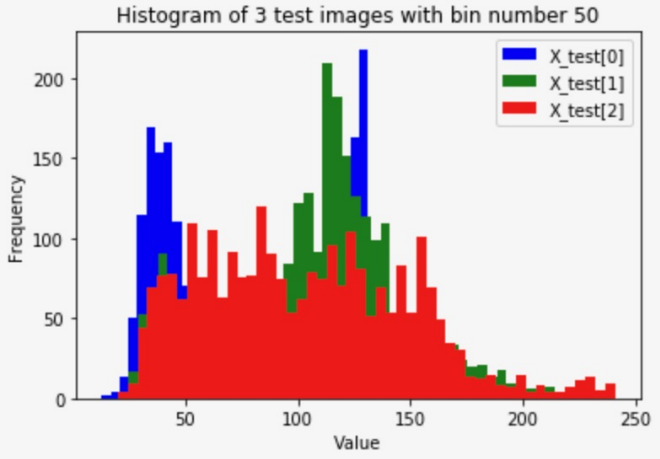


Figure 4: Histogram of pixel intensity for first 3 test images with 50 bins.

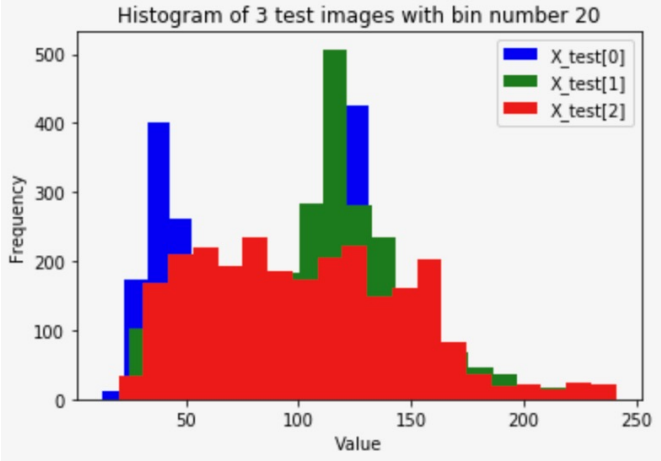


Figure 5: Histogram of pixel intensity for first 3 test images with 20 bins.

Metric	Histogram		
	Rank1	Rank10	mAP
Bhattacharyya	23.5	62.5	21.0
Chi-Square	27.5	65.0	23.1
Intersection	2.0	13.0	1.7
Correlation	2.0	16.5	2.2

Table 10: Baseline performance on pixel intensity histogram with 50 bins

Metric	Histogram		
	Rank1	Rank10	mAP
Bhattacharyya	20.5	65.5	20.8
Chi-Square	25.0	66.5	22.3
Intersection	1.0	13.5	1.4
Correlation	0.5	15.0	1.7

Table 11: Baseline performance on pixel intensity histogram with 20 bins

A.3. Mahalanobis Distance

The formula for the Mahalanobis Distance between points P and Q is as follows:

$$M(P, Q) = \sqrt{(P-Q)^T S^{-1} (P-Q)} \quad (14)$$

where S is the covariance matrix.

Eigenvector number	Normalised		
	Rank1	Rank10	mAP
16	55.3	76.5	0.113
32	61.1	88.3	0.318
64	68.2	93.0	0.443
128	76.1	97.1	0.531
256	76.1	97.1	0.531

Table 12: Mahalanobis Distance performance on $\bar{X}_{tst}$

Eigenvector number	Standardised		
	Rank1	Rank10	mAP
16	56.3	77.0	0.123
32	63.3	90.5	0.329
64	69.8	93.6	0.451
128	76.2	96.1	0.535
256	76.2	96.1	0.535

Table 13: Mahalanobis Distance performance on $\tilde{X}_{tst}$

A.4. PCA-LDA

The objective of PCA-LDA is to find the fisherfaces W_{OPT} defined as:

$$W_{Opt} = W_{PCA}W_{LDA} \in \mathbb{R}^{D \times M_{LDA}} \tag{15}$$

where:

$$W_{LDA} = \underset{W}{\operatorname{argmax}} \frac{|W^T W_{PCA}^T S_B W_{PCA} W|}{|W^T W_{PCA}^T S_W W_{PCA} W|} \tag{16}$$

and

$$W_{PCA} = \underset{W}{\operatorname{argmax}} |W S_T W| \in \mathbb{R}^{D \times M_{PCA}} \tag{17}$$

B. Question 2

B.1. Cluster-based representations

Metric	Rank1	Rank10	mAP
Manhattan	41.5	79.0	0.314
Euclidean	36.0	75.0	0.287
Chebyshev	31.0	67.1	0.256
Cosine	42.0	79.0	0.307
Correlation	40.5	78.0	0.303

Table 14: Performance breakdown for Centroid-Distance representation

Metric	Rank1	Rank10	mAP
Manhattan	44.0	83.0	0.335
Euclidean	42.0	80.5	0.313
Chebyshev	35.3	74.5	0.271
Cosine	41.1	76.2	0.311
Correlation	41.0	75.0	0.307

Table 15: Performance breakdown for Inverse-Softmax representation

Metric	Rank1	Rank10	mAP
Manhattan	56.0	87.0	0.406
Euclidean	41.0	81.5	0.327
Chebyshev	37.0	76.5	0.295
Cosine	41.5	84.5	0.332
Correlation	41.0	79.5	0.311

Table 16: Performance breakdown for GMM-Fisherfaces representation