HACETTEPE UNIVERSITY

AIN433

ASSIGNMENT 1

Hüseyin Eren DOĞAN – 2210765009

INDEX

1 - Introduction

1.1 – Dimension Reduction

Dimension reduction is a technique used in machine learning and statistics to reduce the number of input variables in a dataset. The primary goal is to simplify the dataset while retaining its important features, thereby improving computational efficiency, reducing noise, and often enhancing the performance of machine learning models.

1.2 – PCA Algorithm

Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms the original variables of a dataset into a new set of uncorrelated variables called principal components. The algorithm works by finding the directions (principal components) along which the data varies the most.

Here are the steps of the algorithm:

-Contstructing Matrix: Getting the matrix by stacking the columns (image vectors).

-Normalization: Normalizing the data with substracting mean vector of all of the matrix.

-Eigenvalue Decomposition: Finding the eigenvalues and eigenvectors of the covariance matrix. The eigenvalues represent the variance of the data along the corresponding eigenvectors' directions.

-Sorting Eigenvalues: The eigenvalues are sorted in descending order. The larger the eigenvalue, the more variance it represents. The idea is that the principal components associated with higher eigenvalues capture more of the total variance in the data.

-Principal Components: After sorting the eigenvalues, we select the top k eigenvectors (principal components) corresponding to the k largest eigenvalues to form a new feature subspace. The value of k is chosen based on the desired dimensionality of the reduced dataset.

-Projection: The selected eigenvectors are then used to form a projection matrix. This matrix is used to project the original data onto a lower-dimensional subspace. Finally, the original data is multiplied by the projection matrix to obtain the transformed dataset with reduced dimensions.

1.3 – Eigenvalues and Eigenvectors

Eigenvalues are scalar values that represent the amount of variance captured by each eigenvector in a transformation. In the context of PCA, eigenvalues tell us how much each principal component contributes to the overall variance in the data.

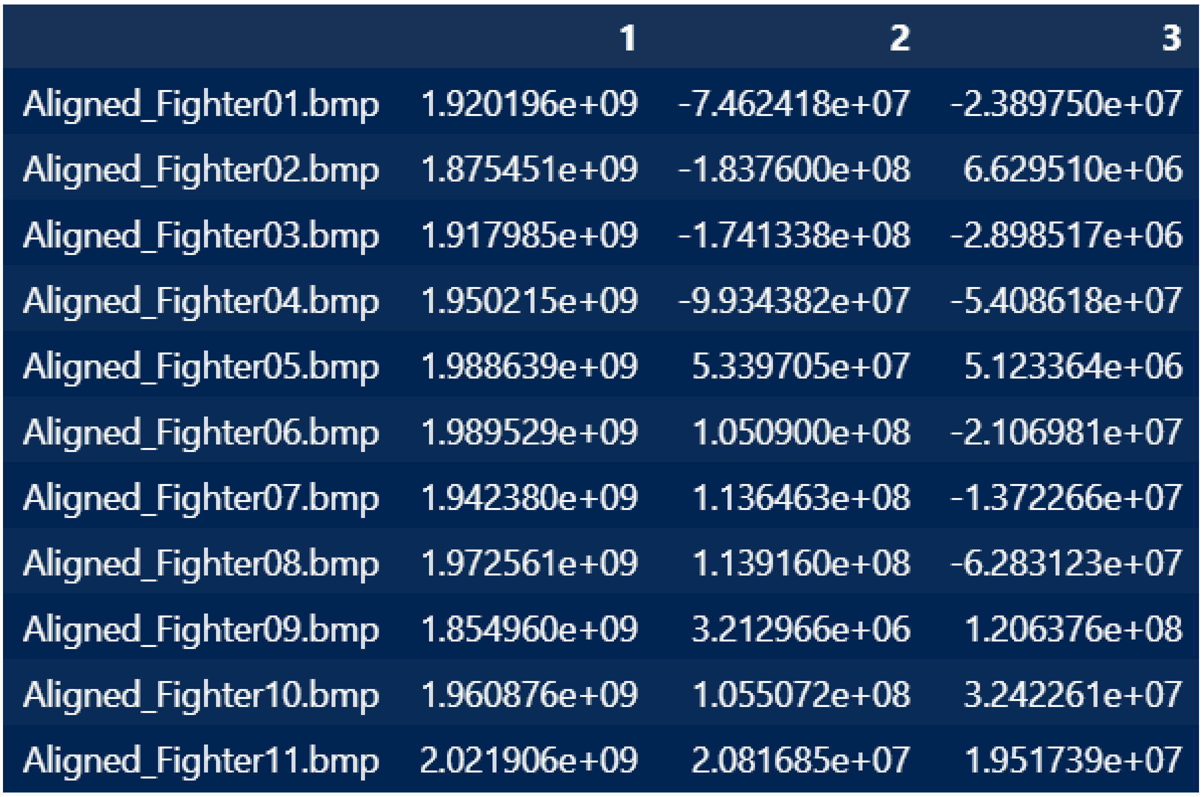
Eigenvectors are the corresponding non-zero vectors associated with eigenvalues. In the context of PCA, eigenvectors represent the directions or axes along which the data varies the most. Each eigenvector corresponds to a principal component, and their combination forms a new coordinate system.

By selecting the top k eigenvectors (those associated with the largest eigenvalues), we can reduce the dimensionality of the dataset while retaining the most significant features. This is crucial for applications like PCA, where the goal is often to represent the data in a lower-dimensional space.

Eigenvectors with smaller eigenvalues capture less variance and may correspond to noise or less significant patterns in the data. By sorting and selecting based on eigenvalues, we focus on the dominant structures and reduce the impact of noise.

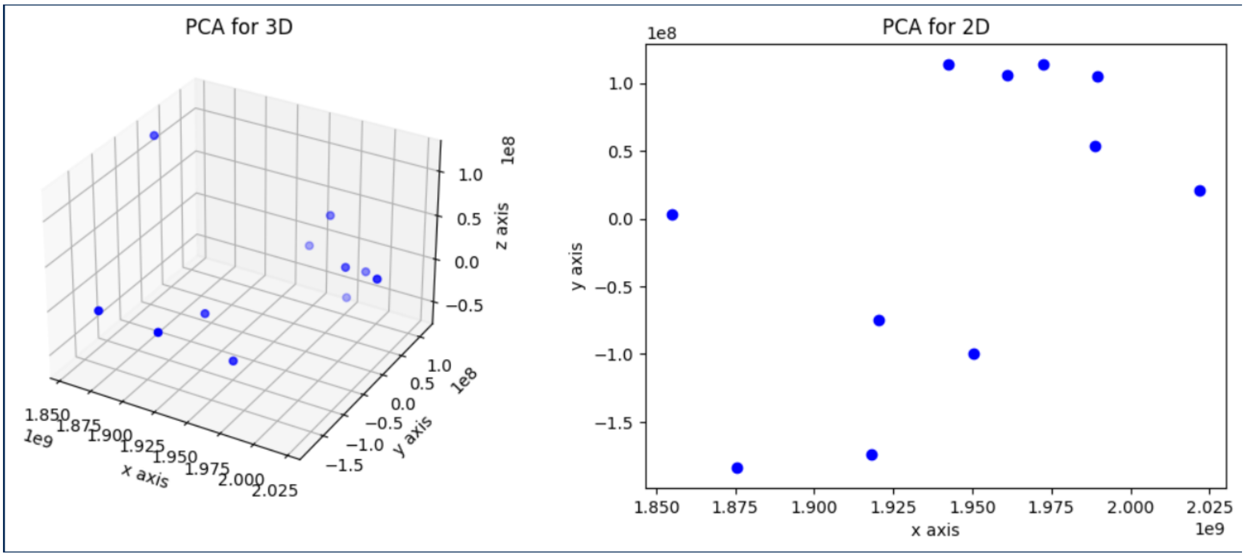
1.4 – Dimensionality Reduction with PCA

Results of PCA in part-1:



	1	2	3
Aligned_Fighter01.bmp	1.920196e+09	-7.462418e+07	-2.389750e+07
Aligned_Fighter02.bmp	1.875451e+09	-1.837600e+08	6.629510e+06
Aligned_Fighter03.bmp	1.917985e+09	-1.741338e+08	-2.898517e+06
Aligned_Fighter04.bmp	1.950215e+09	-9.934382e+07	-5.408618e+07
Aligned_Fighter05.bmp	1.988639e+09	5.339705e+07	5.123364e+06
Aligned_Fighter06.bmp	1.989529e+09	1.050900e+08	-2.106981e+07
Aligned_Fighter07.bmp	1.942380e+09	1.136463e+08	-1.372266e+07
Aligned_Fighter08.bmp	1.972561e+09	1.139160e+08	-6.283123e+07
Aligned_Fighter09.bmp	1.854960e+09	3.212966e+06	1.206376e+08
Aligned_Fighter10.bmp	1.960876e+09	1.055072e+08	3.242261e+07
Aligned_Fighter11.bmp	2.021906e+09	2.081685e+07	1.951739e+07

3D and 2D scatter plots of the results:



In these plots, every dot represents an image, with coordinates corresponds their PCA features; x-axis is PC1, y-axis is PC2, z-axis is PC3 values of them.

We can say that PC1 captures the majority of the variance in data, so we may be more interested in that principal components because it is likely to be more critical for representing the dataset in lower-dimensional spaces.

2 – Image Retrieval with Color Histogram

2.1 – Color Histogram

A color histogram is a representation of the distribution of colors in an image. It provides a summary of how many pixels in the image fall into different color intensity or color value bins.

Here are the steps of the algorithm:

-Image Preparation: Loading the image and divide it into smaller regions or pixels.

-Color Quantization: Converting the image from its original color space (e.g., RGB) to a simpler color space by splitting color channels or reduce the number of possible colors to a predefined set.

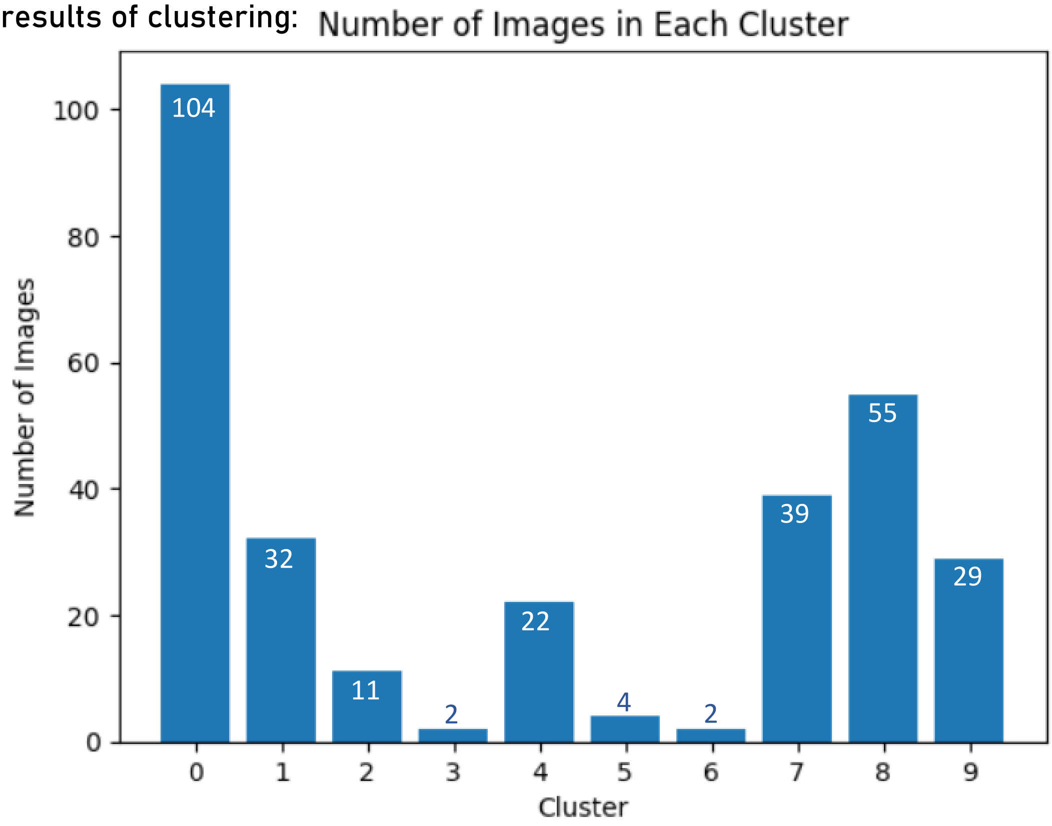
-Histogram Bins: Defining a set of bins for each color channel and counting the number of pixels that fall into each bin for each color channel. This involves iterating through the image and incrementing the corresponding bin counts.

-Normalization: Normalizing each pixel value with dividing the value of pixel by sum of all pixels.

-Histogram Representation: The resulting histogram is a vector where each element corresponds to the frequency of pixels in a particular color bin. The concatenation of histograms for all color channels forms the final color histogram for the entire image.

2.2 – Color Histogram Retrieval Results

After applying color histogram features to K-Means Clustering algorithm, we get the results of clustering:



If we look at the plot, we can see cluster at index 0 has more than 100 images while clusters at indexes 3, 5, 6 have few. It shows that we may have low accuracy in this clustering if we think about that we have 30 images for each class. It may be caused by the model is overfitted the data or there may be too much noise in the data so it affects the distances between images. Maybe we can solve this issue with normalizing the cluster bins in histogram.

2.3 – MAP (Mean Average Precision) Calculation Results

For evaluating the performance of the retrieval algorithm, color histogram, MAP metric calculation has been used. For first, the distance between all query images and all class images has been calculated. Then class images has been sorted by their distance to the image and the closest 10 image selected for each query image.

Here are the classes of query images and class names of images which are in their top 10 ranked list:

QUERY IMAGES	1	2	3	4	5	6	7	8	9	10
airplane	blimp	ibis	airplane	bear	blimp	bonsai	ibis	dog	goose	goose
airplane	airplane	goat	goat	cactus	cactus	bonsai	airplane	goose	ibis	dog
bear	cactus	goat	cactus	goat	goose	cactus	cactus	goat	goat	dog
bear	goose	bear	goat	ibis	goat	ibis	dog	goose	goose	ibis
blimp	bear	dog	ibis	airplane	bonsai	airplane	iris	ibis	airplane	cactus
blimp	dog	cactus	bear	cactus	bonsai	dog	goat	goose	ibis	bonsai
bonsai	bonsai	airplane	dog	goat	dog	bear	cactus	goat	dog	dog
bonsai	bonsai	iris	bonsai	cactus	cactus	cactus	ibis	bear	goose	bear
cactus	cactus	dog	iris	bear	goat	cactus	goose	goat	bonsai	bear
cactus	bonsai	dog	goat	goat	goose	ibis	dog	ibis	goat	airplane
dog	bonsai	blimp	blimp	bonsai	bear	dog	bonsai	blimp	goat	blimp
dog	cactus	ibis	cactus	cactus	goose	bonsai	dog	iris	goat	cactus
goat	iris	bonsai	iris	iris	ibis	cactus	iris	cactus	airplane	bear
goat	ibis	goose	dog	bear	dog	goat	ibis	bear	iris	goat
goose	airplane	iris	cactus	dog	dog	goat	bear	goat	goose	bear
goose	iris	iris	iris	bonsai	bonsai	bonsai	ibis	bear	cactus	goose
ibis	goat	bear	goat	goat	goat	dog	cactus	bear	goose	cactus
ibis	bonsai	iris	bonsai	bear	iris	cactus	goose	bonsai	goat	iris
iris	iris	bear	iris	cactus	iris	goat	bonsai	ibis	goose	bonsai
iris	iris	bonsai	cactus	goat	dog	cactus	iris	goat	bear	iris

After that step, we calculate the MAP values for analyzing the retrieval, here are MAP values for each class:

CLASSES	airplane	bear	blimp	bonsai	cactus	dog	goat	goose	ibis	iris
MAP	0.1496	0.2360	0.0592	0.3361	0.2540	0.2130	0.2178	0.1431	0.1581	0.2571

If we look at the results we can see that the classes which have higher MAP values has relevant images in higher ranks in their ranked list. For instance, the class who has the highest MAP value is ‘bonsai’, if we look at two bonsai images’ ranked lists there are 3 bonsai images ranked as 1st, 1st and 3rd. And the lowest MAP value award goes to ‘blimp’ class with 0.0592. If we take a look at blimp query images’ ranked lists, there is no image which has ‘blimp’ class. As a result, higher MAP values is better retrieval accuracy.

3 – Classification

3.1 – Logistic Regression Algorithm

Logistic Regression is a classification algorithm used for binary classification problems, where the output variable is a categorical variable with two classes. The key idea is to model the relationship between the input features and the binary outcome in a way that produces a probability distribution.

Here are steps of logistic regression algorithm:

-Sigmoid Function: Logistic Regression uses the logistic function (also called the sigmoid function) to model the relationship between the independent variables and the probability of the output belonging to a particular class. The sigmoid function is an S-shaped curve that maps any real-valued number to a value between 0 and 1.

-Linear Combination: The linear combination is calculated as the dot product of the feature values and their corresponding weights plus a bias term.

-Probability Prediction: The sigmoid function is applied to linear combination to obtain a probability that the output belongs to the positive class.

-Decision Boundary: A decision boundary is established by choosing a threshold probability (often 0.5). If P(class 1) is greater than the threshold, the instance is classified as class 1; otherwise, it is classified as class 0.

-Training: During the training phase, the algorithm aims to find the optimal values for the weights and the bias term by minimizing a cost function. The cost function measures the difference between the predicted probabilities and the actual class labels.

-Gradient Descent: Gradient descent or other optimization algorithms are used to iteratively update the weights and bias to minimize the cost function.

3.2 – Binary Classification with Logistic Regression

For this part, since i have used histogram features which i exctracted in part 2, i have selected the two classes according to MAP values which has been calculated in second part, ‘bonsai’ and ‘iris’. It is not so important but i thought it may be more accurate in this way. Anyways, i have taken 60 images of bonsai and iris classes for training the model, then tested it on 4 query images of that two classes. Here is predictions and accuracy of the model:

Predictions: bonsai, iris, iris, iris

True classes: bonsai, bonsai, iris, iris Accuracy = 75%

The accuracy is acceptable as good, but we should not forget this model trained and especially tested on a small data, so it may not be give a proper idea for the model’s performance at all.

3.3 – 10-Class Classification with Logistic Regression (Bonus Part)

For this part, i have looped the binary classification method which i used in part 3 over all the classes and trying to get predictions for the class which has highest probability.

If we talk about the results of this part, the model predicted every image as airplane while there is 20 images of 10 classes in true classes. I think the model overfitted the data but i could not fix this issue.

Predictions: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

True classes: 0 0 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9

Accuracy = 10%